جهاد دانشگاهی
۱۳۵۹
واحد تهران
دانشکده فنی

چاپ هشتم

محاسبات عددی

آنالیز عددی کاربردی

ترجمه و تالیف:

علی محمد پورپاک

دانشکده فنی
دانشگاه تهران

C.GERALD
P.WHEATLEY

CD
اختیاری

محاسبات عددی

آنالیز عددی کاربردی

برای رشته های مهندسی و علوم

همراه ۳۵۰ برنامه کامپیوتری به زبان های
MATLAB,FORTRAN,PASCAL,C
شامل ۳۰۰ مسأله با جواب

مؤلفین: جرالد/ ویتلی

ترجمه وتالیف:

علی محمد پورپاک

عضو هیأت علمی دانشکده فنی
دانشگاه تهران

فهرست

پیشگفتار ۱
هدف ۱
راهنمای کتاب ۱
فصل ۱ ..... ۳
محاسبات عددی و کامپیوتر ۳
۱.۱ مقدمه فصل اول ۴
۲.۱ کاربرد یک کامپیوتر برای انجام آنالیز عددی ۶
Basic و حتی زبان ماشین نوشته شود. ۶
۳.۱ یک مثال نمونه ۷
۴.۱ بررسی روش نصف کردن فاصله ۹
۵.۱ محاسبات کامپیوتری و خطا ۱۱
خطای برش ۱۱
خطای گرد کردن ۱۲
خطای ناشی از داده‌های اولیه ۱۲
اشتباه ۱۳
انتشار خطاها ۱۳
محاسبه ممیز سیّار و برآورد خطا ۱۴
سیستم‌های کامپیوتری ۱۶
۵.۱ محاسبه و خطا ۱۷
خطا در تبدیل مبنای اعداد ۱۹
ماشین *EPS* ۲۰
خطای مطلق، خطای نسبی، ارقام با معنی ۲۰
ارقام با معنای صحیح ۲۱
قضیه ۱.۱ ..... ۲۲
قضیه ۲.۱ ..... ۲۳
انتشار خطا در توابع ۲۳
۶.۱ موضوعات تئوری ۲۸
۷.۱ پردازش موازی ۲۸
تمرینات فصل اول ۳۰

فصل ۲ ..... ۳۳
حل معادلات غیر خطی ۳۳
۱.۲ روش نصف کردن فاصله‌ها ۳۴
۲.۲ روش درون یابی خطی ۳۷
۳.۲ روش تکرار تابعی، $x = g(x)$ ۴۳
تسریع همگرایی ۴۶
۴.۲ روش نیوتن ۴۸
۵.۲ همگرایی روش نیوتن ۵۱
۶.۲ روش نیوتن برای چند جمله‌ایها ۵۲
روش هورنر ۵۳
۷.۲ روش بیرستو برای عامل‌های درجه دوم ۵۴
۸.۲ روش مولر ۵۸
۹.۲ روش گرافی ۶۱
۱۰.۲ برنامه‌های کامپیوتر ۶۲
برنامه ۱.۲ روش مولر ۶۳
برنامه ۲.۲ روش بیرستو ۶۹
برنامه ۳.۲ روش نیوتن ۷۵
برنامه ۴.۲ روش تکرار تابعی ۷۸
توضیح ۸۰
تمرینات فصل دوم ۸۰
مسائل کاربردی و پروژه‌ها ۹۱
فصل ۳ ..... ۹۷
۱.۳ ولتاژ و جریان در یک شبکه ۹۸
۲.۳ محاسبهٔ نیروها در یک خرپای مسطح ۹۹
۳.۳ روش حذفی گاوس – جردن ۱۰۱
۴.۳ روش تجزیه LU ۱۰۳
۵.۳ تجزیه چولسکی ۱۰۷
۶.۳ شرایط ناهنجار دستگاه معادلات خطی ۱۰۸
۱.۷.۳ روش ژاکوبی ۱۱۰
۲.۷.۳ روش گاوس – سیدل ۱۱۲
۸.۳ دستگاه معادلات غیر خطی (روش نیوتن) ۱۱۵
برنامه ۱.۳ حل دستگاه معادلات خطی ۱۱۹

برنامه ۲.۳ حل دستگاه معادلات غیر خطی ۱۲۲
برنامهٔ ۳.۳حل دستگاه معادلات به روش تجزیه LU ۱۲۵
تمرینات فصل سوم ۱۲۹
فصل ۴ ..... ۱۳۴
مقدمه ۱۳۵
۱.۴ خواص مقادیر ویژه و بردارهای ویژه یک ماتریس ۱۳۹
۲.۴ روش لوریه - فادیو ۱۴۰
۳.۴ روش توانی ۱۴۲
اثبات همگرائی ۱۴۳
۴.۴ روش تکراری معکوس ۱۴۶
اثبات همگرائی: ۱۴۷
۱.۴ قضیه گرچ گورین ۱۴۸
۵.۴ روش صفر کردن مقادیر ویژه ماتریس ۱۴۹
۶.۴ روش تبدیل ماتریس برای تعیین مقادیر ویژه ۱۵۲
۷.۴ روش ژاکوبی (برای ماتریس متقارن) ۱۵۳
۸.۴ روش گیونز(برای ماتریس‌های متقارن) ۱۵۵
۹.۴ معکوس کردن ماتریس ۱۵۹
برنامه ۱.۴ روش توانی ۱۶۲
تمرینات فصل چهارم ۱۶۶
فصل ۵ ..... ۱۶۹
۱.۵ مسأله درونیابی ۱۷۰
۲.۵ چندجمله‌ای‌های لاگرانژ ۱۷۰
۳.۵ روش نیویل ۱۷۴
۴.۵ تفاضل‌های محدود ۱۷۷
تفاضل‌های محدود برای تابع $f(x)$ بصورت یک چند جمله‌ای ۱۸۰
خطای درونیابی ۱۸۱
تخمین خطا وقتی $f(x)$ ناشناخته است - قاعده جمله بعدی ۱۸۲
درونیابی نزدیک انتهای یک جدول ۱۸۳
دادهای متساوی الفاصله ۱۸۳
تفاضل‌های تابع بجای تفاضل‌های محدود ۱۸۶
چند جمله‌های درونیاب بر مبنای جدول‌های تفاضلی ۱۸۷
۵.۵ درونیابی با اسپلاین درجه سه ۱۸۹

۶.۵ خم‌های بی‌زیّر *Bezier* و خم‌های *B*- اسپلاین ۲۰۰
خم‌های *B*- اسپلاین ۲۰۵
۷.۵ تقریب چند جمله‌ایی سطوح ۲۱۰
بکار بردن اسپلاین درجه سه، سطوح بی‌زیّر، و سطوح B- اسپلاین ۲۱۳
۸.۵ روش تقریب حداقل مربعات ۲۱۷
داده‌های غیر خطی ۲۲۰
درجه چند جمله‌ای مناسب چیست؟ ۲۲۵
۹.۵ برنامه‌های کامپیوتری ۲۲۸
۱.۵ لیست برنامه تفاضل‌های محدود به زبان *C* ۲۲۸
۲.۵ لیست برنامه *B*-اسپلاین درجه سوم به زبان *PASCAL* ۲۳۱
۳.۵ لیست برنامه حداقل مربعات به زبان *PASCAL* ۲۳۵
تمرینات فصل پنجم ۲۳۹
فصل ۶ ..... ۲۵۲
۱.۶ محاسبه مشتقات ۲۵۳
۲.۶ مشتقات از چند جمله‌ای‌های درونیاب ۲۵۳
۳.۶ مشتقات مراتب بالاتر ۲۵۸
۴.۶ نمودار لوزی برای مشتقات ۲۶۰
۵.۶ فنون برونیابی ۲۶۵
۶.۶ گردکردن و دقت مشتقات ۲۶۸
۷.۶ برنامه کامپیوتری برای محاسبه مشتق ۲۷۲
برنامه ۱.۶ لیست برنامه محاسبه مشتقات به زبان *BASIC* ۲۷۲
تمرینات فصل ششم ۲۷۴
فصل ۷ ..... ۲۷۷
۱.۷ فرمول‌های انتگرال گیری نیوتن- کوتز ۲۷۸
۲.۷ روش ذوزنقه‌ای ۲۸۰
۳.۷ روش سیمپسون ۲۸۳
۴.۷ روش انتگرال گیری رامبرگ ۲۸۸
۵.۷ روش‌های دیگر برای پیدا کردن فرمول‌های انتگرال گیری ۲۹۱
۶.۷ روش انتگرال‌گیری گاوس ۲۹۳
۷.۷ انتگرال‌های ناویژه و انتگرال‌های نامعین ۲۹۸
۸.۷ کاربردهای توابع اسپلاین درجه سوم ۳۰۲
۹.۷ انتگرال‌های چندگانه ۳۰۴

۱۰.۷ خطا در انتگرال‌های چندگانه ۳۰۹
۱۱.۷ انتگرال‌گیری چند گانه با حدود متغیر ۳۱۰
۱۲.۷ برنامه‌های کامپیوتری ۳۱۴
۱.۷ لیست برنامه انتگرال دوگانه به‌روش رامبرگ به زبان *FORTRAN* ۳۱۴
۲.۷ لیست برنامه انتگرال گیری به روش سیمپسون به زبان *PASCAL* ۳۱۷
۳.۷ لیست برنامه انتگرال گیری به روش گاوس به زبان *C* ۳۲۰
تمرینات فصل هفتم ۳۲۳
فصل ۸ ..... ۳۲۹
حل عددی معادلات دیفرانسیل معمولی ۳۳۰
۱.۸ مسأله ارتعاش جرم - فنر ۳۳۱
۲.۸ حل معادلات دیفرانسیل با استفاده از سری تیلر ۳۳۲
۳.۸ روش اویلر ۳۳۴
۴.۸ روش هیون ۳۳۵
۵.۸ حل دستگاه معادلات دیفرانسیل ۳۳۵
۶.۸ خطای موضعی و ناحیه‌ای ۳۳۹
۷.۸ روش‌های رانج - کوتا ۳۴۰
روش رانج - کوتای مرتبه دوم ۳۴۱
روش رانج کوتای مرتبه چهارم ۳۴۳
۸.۸ روش‌های چندگامه ۳۴۴
۹.۸ روش میلن - سیمپسون یا روش پیش بینی - تصحیح ۳۴۴
۱۰.۸ روش پیش بینی - تصحیح آدامز (مولتن) ۳۴۶
۱۱.۸ الگوریتم آدامز - مولتن ۳۴۷
۱۲.۸ روش رانج - کوتا - فیلبرگ ۳۴۹
۱۳.۸ حل دستگاه معادلات دیفرانسیل ۳۵۲
۱۴.۸ روش کول - نیومروف ۳۵۳
۱۵.۸ برنامه‌های کامپیوتری ۳۵۵
برنامه ۱.۸ لیست برنامه کامپیوتری آدامز - مولتن به زبان *BASIC* ۳۵۵
برنامه ۲.۸ لیست برنامه روش رانج - کوتا- فیلبرگ به زبان *PASCAL* ۳۵۸
برنامه ۳.۸ روش رانج - کوتا- فیلبرگ ۳۶۱
تمرینات فصل هشتم ۳۶۵
منابع ۳۷۲

بسم الله الرحمن الرحیم

پیشگفتار

محاسبات عددی (آنالیز عددی کاربردی) از یکطرف با ریاضیات محض و از طرف دیگر با محاسبات مهندسی در رابطه است.

می‌توان گفت که آنالیز عددی علمی است ریاضی و هنری است محاسباتی. کامپیوتر در محاسبات کاربرد بسیار وسیعی دارد و با توجه به سرعت زیاد عملیات در کامپیوتر است که برنامه‌های مختلف و پیچیده برای حل مسائل ریاضی نوشته و اجراء می‌شوند.

آنالیز عددی کاربردی برای دانشجویان و کارشناسان مهندسی، علوم، ریاضیات، و علوم کامپیوتر طبق سرفصل‌های پیشنهادی شورای عالی برنامه‌ریزی وزارت علوم، تحقیقات و فن آوری نوشته شده است. این مطالب به عنوان کتاب مرجع برای مهندسین و دانشمندانی که به روش‌های عددی برای حل مسائل نیازمندند با ارزش می‌باشد. باعث خوشحالی خواهد شد که دیده شود بسیاری از افرادی که این کتاب را تهیه می‌کنند آنرا جزء ثابت و دائمی کتابخانه خود قرار دهند و این به دلیل قابل درک و فهم بودن این کتاب می‌باشد و اجازه می‌دهد که شخص با مطالعه فردی معلوماتش را افزایش دهد.

هدف

هدف از چاپ این کتاب، ارائه مطالب کلی در رابطهٔ با آنالیز عددی و تأکید بر کاربردهای عملی آن به جای تئوری می‌باشد. و در این حال روش‌های گوناگون مقایسه می‌شوند، اشکالات هر کدام بررسی می‌شود، ارتباط بین قضایای اساسی ریاضی و شیوه‌های مختلف بیان می‌شود و در نتیجه یک درک واقعی از مباحث حاصل می‌گردد. شرح مفصل مباحث با استفاده از مثال و هماهنگی‌های منطقی بین آنها حاصل می‌گردد. شرحی که دانشجو هر چه بیشتر و بهتر در کاربرد این شیوه مهارت پیدا کند. بهمین دلیل این کتاب را «آنالیز عددی کاربردی» یا «محاسبات عددی» می‌توان نامید.

راهنمای کتاب

آنالیز عددی کاربردی، (محاسبات عددی) نتایج ارزشمندی را بدست می‌دهد که در چندین مبحث مهم در این کتاب طرح و بررسی می‌شود. تعداد زیاد تمرینها به استاد اجازه می‌دهد مناسبترین آنها را بر طبق علاقه و استعداد دانشجویان انتخاب کند و این موضوع باعث می‌شود که

قسمت‌های مختلف هر فصل در اینگونه مسائل گردآوری شوند. وقتی خواننده این کتاب را به صورت خود آموز استفاده می‌کند تعدادی از تمرینات مهم برای آزمایش او خواهد بود. به علاوه در تمرینات عملی هر بخش مسایل کاربردی و پروژه‌هایی وجود دارد که دانشجو را به تلاش وامی‌دارد و با حل مسایل مشابه به روش‌های گوناگون، ارتباط بین کارآیی و تأثیر روش‌های متفاوت به خوبی برای او روشن می‌گردد.

توضیح مختصری در مورد مطالب هر فصل در آغاز آن می‌آید که تصور ذهنی از مطالب آن ارائه می‌دهد.

بیشتر فصلها با یک مثال ساده معرفی می‌شوند که در آن موضوعات هر فصل بکار می‌رود. این موضوع به دانشجو انگیزه می‌دهد و نشان می‌دهد که مباحث و مطالب گنجانده شده در واقع مفید می‌باشند. و عملاً مورد استفاده قرار می‌گیرند

در پایان هر فصل برنامه‌های کامپیوتری به زبان‌های *C* و *Pascal*؛ *Fortran* یا *Basic* آورده شده است که این برنامه‌ها، روشها و الگوریتم‌های مهم را تکمیل می‌کنند و در فهم مثالها و درک اینکه کامپیوتر چگونه می‌تواند محاسبات را انجام دهد مؤثر می‌باشد. ادعا نمی‌شود که این برنامه‌ها در سطح برنامه‌های حرفه‌ای هستند زیرا قصد آنها وضوح و روشن کردن موضوع است. بعضی برنامه‌ها به چند زبان *C* ، زبان *FORTRAN*، زبان *BASIC* و *PASCAL* آماده شده است. و برای آسانی ورود به کامپیوترهای شخصی روی دیسکت ذخیره شده‌اند.

اکثر برنامه‌های پایان هر فصل دارای خروجی‌های ساده‌ای هستند، تا خواننده بتواند خود را بیازماید.

نکته‌ای که در اجرای برنامه ها باید توجه داشت و اغلب فراموش می‌شود، دقت در محاسبات عددی است. عددهائی را که در محاسبات بکار می‌بریم معمولاً عددهائی تقریبی هستند و نه تحقیقی.

چه بسا در یک برنامه صدها بلکه هزاران بار جمع، تفریق، ضرب یا تقسیم روی این اعداد تقریبی انجام پذیرد. در اینصورت ممکن است دقت در نتایج حاصل بتدریج کم شود، در اینصورت باید در محاسباتی که با کامپیوتر انجام می‌دهیم نیز دقت کافی نموده تا بتوانیم به جواب بدست آمده اطمینان کنیم.

در محاسبات عددی (آنالیز عددی کاربردی) ضمن آنکه باید سعی شود هر چه ممکن است زمان اجرای برنامه در کامپیوتر کاهش یابد و یا بسهولت بتوان با ماشین حساب جواب را بدست آورد، باید همواره به دقت در محاسبات نیز توجه بسیار گردد تا هر چه بیشتر جواب نهائی به جواب تحقیقی نزدیک شود و خطا از حد معین پیش بینی شده کمتر باشد.

علی محمد پورپاک

فصل ۱

محاسبات عددی و کامپیوتر

جان فون نیومان
John Von Neumann
(۱۹۵۷-۱۹۰۳)

هرجا بحثی از کامپیوترهای نوین می‌شود اشاره به ریاضیدان بزرگ مجارستانی، جان فون نیومان، مناسبت دارد. او مسؤولیت به کار انداختن اولین ماشین حساب الکترونیکی کامل و اندیشه کامپیوتر رقمی حافظه دار را به عهده داشت.

نیومان در سال ۱۹۰۳ در بوداپست به دنیا آمد وی دکترایش را در سال ۱۹۲۶ در بوداپست گرفت. در سال ۱۹۳۰ به امریکا مهاجرت کرد. وی قبلاً به خاطر سهمش در نظریهٔ عملگرها، نظریه کوانتوم، و نظریهٔ بازی شهرتی جهانی داشت. او نقش زیادی در جهت دهی به مقدار وسیعی از ریاضیات قرن بیستم داشت. طی جنگ جهانی دوم به کارهای علمی و اداری ساختن بمب‌های اتمی و هیدروژنی اشتغال داشت.

موضوعات این فصل

* مقدمه
* کاربرد یک کامپیوتر برای انجام آنالیز عددی
* یک مثال نمونه
* بررسی روش نصف کردن
* محاسبه کامپیوتری و خطا
* موضوعات تئوری
* پردازش موازی

۱.۱ مقدمه فصل اول

آنالیز عددی راهی برای انجام مسایل ریاضیات عالی با کامپیوتر می‌باشد، روشی که بطور گسترده توسط دانشمندان و مهندسین برای حل مسایل آنان بکار می‌رود. فایده اصلی آنالیز عددی این است که حتی زمانی که مسأله دارای جواب «تحلیلی» نیست، جواب عددی می‌تواند بدست آید. بعنوان مثال، انتگرال زیر که طول قوسی از منحنی $y = sin(x)$ را می‌دهد، دارای تابع اولیه نیست:

$$\int_0^{\pi} \sqrt{1 + \cos^2(x)}\, dx. \qquad (1.1)$$

محاسبات عددی می‌تواند طول قوس این منحنی را با روش‌های استاندارد محاسبه کند و هرگز نیازی به تغییر متغیر یا محاسبه انتگرال به روش جزء به جزء نیست.

بعلاوه تنها عملیات ریاضی لازم، جمع، تفریق، ضرب و تقسیم و مقایسه کردن می‌باشد.

مهم است که درک کنیم یک جواب آنالیز عددی همیشه عدد است. روش‌های تحلیلی معمولاً نتیجه‌ای بر حسب توابع ریاضی می‌دهند که می‌توانند برای مقادیر خاص تعیین مقدار شوند.

نتیجه تحلیلی دارای این فایده است، که رفتار و خواص تابع اغلب معلوم است. این خاصیت برای جواب‌های عددی محض وجود ندارد. بهر حال، نتایج عددی می‌تواند رسم شود و بعضی از رفتارهای جواب را نشان دهد.

فرق مهم دیگر اینست که نتیجه آنالیز عددی یک مقدار تقریبی است، اما نتیجه می‌تواند تا حد دلخواه دقیق شود. برای رسیدن به دقت بالا، عملیات جداگانه بسیار زیادی باید انجام گیرد، اما کامپیوترها بقدری سریع و بدون اشتباه آنرا انجام می‌دهند که هیچ مشکلی وجود ندارد. در واقع، تعین مقدار یک نتیجه تحلیلی برای بدست آوردن جواب عددی برای کاربردی خاص نظیر همان خطاها را دارا می‌باشد.

در اینجا بعضی از عملیاتی که آنالیز عددی می‌تواند انجام دهد و در این کتاب شرح داده شده است، ذکر می‌شود.

* حل ریشه‌های یک معادله غیر خطی.
* حل دستگاه‌های بزرگ معادلات خطی.
* بدست آوردن جواب‌های یک دستگاه معادلات غیر خطی.
* تعیین مقادیر ویژه و بردارهای ویژه یک ماتریس.
* برازش خم به داده‌ها با روش‌های متنوع.
* پیدا کردن سودمند و مؤثر تقریب توابع.
* برازش جهت بدست آوردن مقادیر در جدول داده‌ها.
* مشتقات از هر مرتبه برای توابع حتی زمانی که تابع فقط بعنوان جدولی از مقادیر شناخته

شده باشد.

* انتگرال هر تابع حتی زمانی که تابع فقط بعنوان جدولی از مقادیر، شناخته می‌شود. انتگرال‌های چندگانه نیز بدست می‌آیند.

* حل معادلات دیفرانسیل معمولی زمانی که مقادیر اولیه برای متغیرها داده شده‌اند. این معادلات می‌توانند از هر مرتبه و پیچیدگی باشند.

۲.۱ کاربرد یک کامپیوتر برای انجام آنالیز عددی

روش‌های عددی چنان به عملیات محاسباتی تکراری و ملال آوری نیاز دارند که فقط زمانی که ما دارای یک کامپیوتر برای انجام این تعداد زیاد عملیات هستیم حل مسایل به این طریق عملی می‌گردد. بشر بقدری زیاد اشتباه می‌کند که اطمینان کمی به نتیجه این محاسبات خواهد داشت. بعلاوه نیروی انسانی را نمی‌توان صرف چنین عملیات طولانی کرد.

البته یک کامپیوتر اساساً هیچ اطلاع و ذکاوتی ندارد و برای هر مرحله باید جزئیات و دستورات کامل داده شود تا اجراء شود. به بیان دیگر، یک برنامه کامپیوتری باید نوشته شود تا کامپیوتر بتواند آنالیز عددی را انجام دهد. همچنانکه این کتاب را مطالعه می‌کنید، شما به حد کافی درباره تعدادی از روش‌های عددی موجود خواهید آموخت و قادر خواهید بود برنامه‌هائی بنویسید و به اجراء در آورید.

زبان برنامه نویسی خیلی مهم نیست؛ برنامه می‌تواند به زبان *C* و *Pascal* و *Fortran* یا *Basic* و حتی زبان ماشین نوشته شود.

اکثر روشها بطور کامل بصورت شبه کد شرح داده خواهد شد به شکلی که ترجمه این کد به یک برنامه نسبتاً بطور مستقیم انجام می‌شود. تعدادی از مثال برنامه‌ها برای نمایش طرز عمل داده شده‌اند. این برنامه‌ها طراحی شده‌اند تا بسهولت خوانده شوند، بجای آنکه مثالهایی از برنامه‌های حرفه‌ای باشند.

در واقع نوشتن برنامه‌ها همیشه ضروری نیست. روش‌های عددی بقدری مهم هستند که بسته‌های نرم افزاری کاملی آماده شده‌اند.

(International Mathematical and Statistical Libraray) IMSL

math/library/ دارای صدها برنامهٔ کارآمد با راندمان بالا است که به زبان *FORTRAN* و *C* نوشته شده‌اند و روشها را اجراء می‌کنند.

اخیراً، *LAPACK (Linear Algebra Package)* به قیمت نازلی آماده ارائه شده است. این بسته از برنامه‌های *FORTRAN* از برنامه‌های فرعی که در بسته‌های قبلی *LINPACK* و *EISPACK* بودند، تشکیل شده است.

حالت دیگر بکار بردن یک برنامه نوشته شده به یکی از زبان‌های سطح بالا می‌باشد. یک نوع دیگر نرم افزار که گاهی برنامه جبری سمبلیک *(Symbolic algebra program)* نامیده می‌شود، وجود دارد. این برنامه‌ها طریقی که انسان مسایل ریاضی را حل می‌کند را تقلید می‌نمایند. چنین برنامه‌ای طرح شده است که نوع تابع نشان داده شده را بشناسد (چند جمله‌ای، غیر جبری و غیره) و سپس عملیات ریاضی خواسته شده را روی تابع یا عبارت اجراء کند. آن برنامه با مراجعه به جدول عبارت‌های جدید که نتیجه انجام عمل است یا بوسیله یک مجموعه از قوانین از پیش ساخته شده، انجام می‌گردد. بعنوان مثال، یک برنامه می‌تواند قوانین معمولی را برای پیدا کردن مشتقات

بکار برد، جدول انتگرالها را برای انتگرال گیری در اختیار بگیرد، و تجزیه چند جمله‌ای به عاملها یا یک مجموعه از عاملها را بسط دهد. اینها فقط چند تا از امکانات هستند. اگر جواب تحلیلی نتواند داده شود، این برنامه‌ها به کاربر اجازه می‌دهند یک جواب با روش‌های عددی بدست آورند.

تعدادی از چنین برنامه‌های جبری سمبلیک در دسترس می‌باشند، از جمله:

MAPLE ،*MATLAB* ، *MATHCAD* ، *DERIVE* ، *MATHEMATICA* *MACSYMA.*

یک صورت خاص در اکثر این برنامه‌ها قدرت آنها در اجرای تعداد زیادی عملیات با محاسبه دقیق است. یک مثال جالب اینست که عدد π تا *100* رقم اعشاری نمایش داده می‌شود. معمولاً زمانی که یک کامپیوتر معمولی را بکار می‌بریم، ما باید با تعداد محدودی از ارقام دقیق قانع شویم.

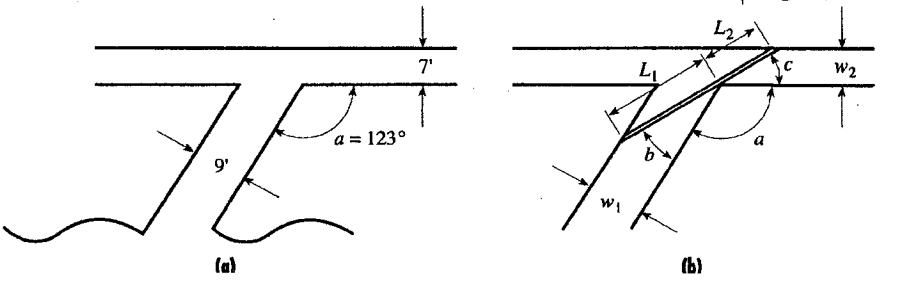
۳.۱ یک مثال نمونه

ما موضوع محاسبات عددی را با نشان دادن یک مثال که بطور عددی حل خواهد شد، نشان می‌دهیم. اگر شما برای یک شرکت معدن کار کنید. مسأله زیر ممکن است مطرح شود.

مثال ۱.۱

دو تونل معدن موجودند که به زاویهٔ $123°$ یکدیگر را قطع می‌کنند، مطابق شکل ۱-۱. تونل مستقیم دارای عرض *7* فوت است، در صورتی که تونل ورودی دارای عرض *9* فوت است. بزرگترین طول نردبانی که می‌توان آن را از پیچ عبور داد چه اندازه است؟ شما می‌توانید از ضخامت قسمت‌های نردبان صرف نظر کنید.

مسأله را در حالت کلی حل می‌کنیم زاویه پیچ را و عرض‌های تونل را متغیرهای w_1 و w_2 انتخاب می‌کنیم.



شکل ۱.۱

همچنانکه تجزیه و تحلیل زیر نشان می‌دهد، برای حل این مسأله باید یک معادله غیر جبری را برای یافتن مقدار C حل کنیم، و سپس مقدار C را جایگزین کنیم.

پیدا کردن جواب یک معادله جبری یا غیر جبری، همچنانکه در اینجا مورد نظر است

موضوع اصلی فصل بعدی می‌باشد.

اینجا یک روش برای تجزیه و تحلیل این مسأله وجود دارد. نردبان را در موقعیت های متوالی‌های که به گوشه‌ای تکیه دارد مورد بررسی قرار می‌دهیم، یک موقعیت ویژه موجود خواهد بود که در آن دو انتهای نردبان دیوارها را لمس می‌کنند در حالی که یک نقطه در امتداد نردبان به گوشه‌ای تکیه دارد که در آنجا دو تونل یکدیگر را قطع می‌کنند. (نگاه کنید به شکل *(b)* ۱.۱) فرض کنیم C زاویه بین نردبان و دیوار در این موقعیت بحرانی باشد.

یک سری از خطوط را که در این موقعیت رسم شده‌اند در نظر بگیرید. طول نردبان بستگی به زاویه C داشته و با آن تغییر می‌کند و روابط زیر برقرار می‌باشند، زوایا برحسب اندازه رادیان می‌باشند:

ماکزیمم طول نردبان که می‌تواند پیچ را طی کند، برابر می‌نیمم L است که بعنوان تابعی از زاویه C می‌باشد؛ با تعیین زاویه C، طول نردبان بدست می‌آید.

از این رو قرار می‌دهیم:

$$L_1 = \frac{w_1}{\sin(b)}, \qquad L_2 = \frac{w_2}{\sin(c)}$$

$$b = \pi - a - c,$$

$$L = L_1 + L_2 = \frac{w_1}{\sin(\pi - a - c)} + \frac{w_2}{\sin(c)}.$$

$$\frac{dL}{dC} = 0$$

در فصل بعد ما روشهایی را برای پیدا کردن ریشه‌های یک معادله (مانند این مثال) مطالعه می‌کنیم.

1: `w1 : = 9` فوت

2: `w2 : = 7` فوت

3: $$\frac{\texttt{w1}}{\texttt{SIN }(\pi - \texttt{a} - \texttt{c})} + \frac{\texttt{w2}}{\texttt{SIN }(\texttt{c})}$$

4: $$\frac{\texttt{d}}{\texttt{dc}}\left[\frac{\texttt{w1}}{\texttt{SIN }(\pi - \texttt{a} - \texttt{c})} + \frac{\texttt{w2}}{\texttt{SIN }(\texttt{c})}\right]$$

5: $$-\frac{\texttt{w1 COS }(\texttt{a} + \texttt{c})}{\texttt{SIN }(\texttt{a} + \texttt{c})^2} - \frac{\texttt{w2 COS }(\texttt{c})}{\texttt{SIN }(\texttt{c})^2}$$

6: $$\frac{9\ \texttt{COS }(2.147 + \texttt{c})}{\texttt{SIN }(2.147 + \texttt{c})^2} \quad \frac{7\ \texttt{COS }(\texttt{c})}{\texttt{SIN }(\texttt{c})^2}$$

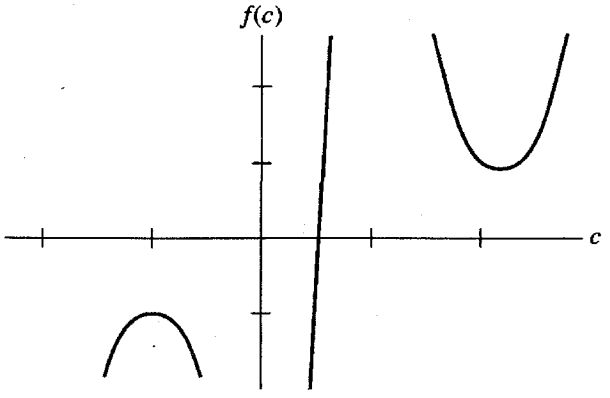
شکل (۲.۱)

بعنوان مثال برای بدست آوردن جواب‌های چند جمله‌ای‌های درجه دو، می‌توان از یک فرمول ساده که با آن آشنا هستیم استفاده کنیم. برای چند جمله‌ای‌های درجه سه و چهار، فرمولهایی وجود دارند، اما بعلت پیچیدگی بندرت بکار می‌روند. برای معادلات با درجات بالاتر مانند چند جمله‌ای درجه پنج ثابت می‌شود که پیدا کردن جواب بصورت یک فرمول غیر ممکن است. اکثر معادلات غیر جبری (شامل توابع نمایی و مثلثاتی) همچنان پیچیده هستند.

اگرچه مشکل است، اما اگر ممکن باشد که جواب چنین معادلاتی را به فرم صحیح نشان دهیم، آنالیز عددی راههایی را ارائه خواهد داد که از طریق آنها احتمالاً یک جواب پیدا می‌شود. احتمال دارد جواب را با دقتی که مایلیم بدست آوریم. بسیاری از این روش‌های عددی حاصل یک سری از تقریبات متوالی محسوب می‌گردند که با این روشها با تکرارهای کافی جواب تقریبی که از جواب صحیح اختلافی کمتر از خطای دلخواه قابل اغماض *(tolerance)* دارد، بدست می‌آید. بنابراین دیده می‌شود که روش‌های عددی مانند مفهوم حد در آنالیز ریاضی می‌باشند.

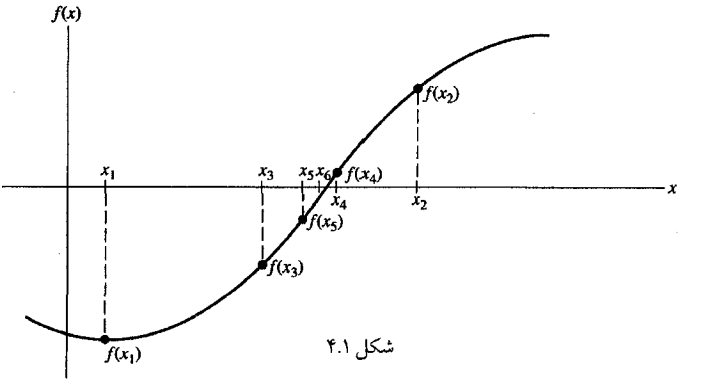
۴.۱ بررسی روش نصف کردن فاصله

برای حل این مسأله روش نصف کردن فاصله را بکار می‌بریم. هرچند در فصل بعدی روش‌های مختلف حل معادلات جبری و غیر جبری مورد بحث قرار می‌گیرند. برای حل معادله $f(x)=0$ با دو مقدار که ریشه معادله بین آنها قرار دارد شروع می‌کنیم، بطوریکه تابع $y=f(x)$ در فاصله $[x_1, x_2]$ پیوسته است و اگر $f(x_1)\ f(x_2)<0$ باشد طبق قضیه بولزانو $f(x)$ در این فاصله حداقل دارای یک ریشه است شکل ۳.۱ یک ریشه را بین 0 و 1 نشان می‌دهد.



شکل ۳.۱

لیکن در شکل ۴.۱ روش تکرار عملیات نمایش داده شده است. ما قصد داریم یک



شکل ۴.۱

برنامه کامپیوتری برای این الگوریتم بنویسیم. قبل از آن این الگوریتم را بصورت شبه کد بیان می‌کنیم. بدین طریق منطق برنامه که مستقل از زبان برنامه نویسی است شرح داده می‌شود.

آلگوریتم برای روش نصف کردن

برای تعیین ریشه $f(x)=0$ یا خطای قابل اغماض معین، بطوریکه $f(x_1)*f(x_2)<0$

REPEAT

set $x_3 = (x_1+x_2)/2.$

IF $f(x_3)*f(x_1) < 0$:

SET $x_2 = x_3.$

ELSE Set $x_1 = x_3.$

ENDIF

UNTIL ($|x_1 - x_2|$ < خطای قابل اغماض) یا $f(x_3)=0$

جواب تقریبی x_3 می‌باشد و خطا از $\frac{1}{2}|x_1 - x_2|$ کوچکتر است.

زمانی که آلگوریتم بصورت شبه کد بیان شده است نوشتن برنامه کامپیوتری مشکل نیست. شکل ۵.۱ برنامه به زبان *BASIC* را برای مثال نردبان در معدن نشان می‌دهد.

```
' USES BISECTION METHOD TO FIND A ROOT OF F(X) = 0
' GETS X1, X2, TOLERANCE VALUE FROM USER
' CHANGE DEF FNX() TO MATCH REQUIRED F(X)
DEF FNX (C) = 9 * COS(A + C) / SIN(A + C) ^ 2 + 7 * COS(C) / SIN(C) ^ 2
A = 123 * 3.14159 / 180
LIMIT = 50
INPUT "ENTER VALUES FOR X1, X2, TOLERANCE ", X1, X2, TOL
' DO A HEADING
PRINT "ITER NO", "X1", "X2", "X3", "F(X3)": PRINT
ITER = 1
DO
  F1 = FNX(X1)
  F2 = FNX(X2)
  IF F1 * F2 > 0 THEN
    PRINT "VALUES DO NOT BRACKET A ROOT": EXIT DO
  END IF
  X3 = (X1 + X2) / 2
  F3 = FNX(X3)
  PRINT ITER, X1, X2, X3, F3
  IF F3 * F1 < 0 THEN X2 = X3 ELSE X1 = X3
  ITER = ITER + 1
LOOP UNTIL ABS(X1 - X2) / 2 < TOL OR F3 = 0 OR ITER > LIMIT
```

شکل ۵.۱ برنامه برای روش نصف کردن

شکل ۶.۱ خروجی برنامه را بعد از اجراء نشان می‌دهد. جواب مسأله برابر *0.4678* رادیان یا *(26.8°)* می‌باشد. ما می‌توانیم ماکزیمم طول نردبان را از معادله ۲.۱ بدست آوریم. از این رابطه *L=33.42* فوت بدست می‌آید.

در برنامه تعداد تکرار عملیات کنترل می‌شود اگرچه در شبه کد نشان داده نشده است. امّا همیشه بهتر است این تست گنجانده شود تا از حلقه بدون پایان اجتناب شود.

ITER NO	X1	X2	X3	F(X3)
1	.4	.5	.45	4.661504
2	.45	.5	.475	-1.893359
3	.45	.475	.4625	1.364828
4	.4625	.475	.46875	-.2675005
5	.4625	.46875	.465625	.5476606
6	.465625	.46875	.4671875	.1398538
7	.4671875	.46875	.4679688	-6.388076E-02
8	.4671875	.4679688	.4675781	3.797662E-02
9	.4675781	.4679688	.4677734	-1.295546E-02

شکل ۶.۱ خروجی برنامه برای $x_1=0.4$ و $x_2=0.5$ و خطای قابل اغماض $1E-4$

۵.۱ محاسبات کامپیوتری و خطا

قبلاً مشاهده کردیم که ضروری است به خطای محاسبات عددی در روش‌های مورد نظرمان توجه کنیم، برای چند روش، تحلیل کردیم که چگونه در تکرارهای متوالی خطاها کاهش می‌یابند (و یا افزایش می‌یابند اگر روش واگرا شود).

وقت آن رسیده که نظر عمیقتری به چند منبع خطا بیفکنیم که دقت ونتیجه را تحت تأثیر قرار می‌دهند. می‌توانیم این چند منبع را به صورت زیر فهرست نماییم.

خطای برش

این نام به خطاهایی اطلاق می‌شود که خود روش، مسبب آنست (این نام از این حقیقت ناشی می‌شود که روش‌های عددی را می‌توان معمولاً با یک سری تیلور بریده شده مقایسه نمود) و این خطایی است که تاکنون بیشترین توجه را به آن معطوف داشته‌ایم. بعنوان مثال ممکن است e^x را با یک چند جمله‌ای درجه سه تقریب کنیم.

بهر حال می‌دانیم که برای محاسبه e^x واقعاً یک سری طولانی نامحدود لازم است:

$$p_3(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}.$$

توجه شود که مقدار تقریبی e^x با چند جمله‌ای درجه سه جواب دقیق را نمی‌دهد. خطا مربوط به برش سری بوده و ارتباطی به کامپیوتر و ماشین حساب ندارد.

$$e^x = p_3(x) + \sum_{n=4}^{\infty} \frac{x^n}{n!}.$$

در روش‌های تکراری در صورت همگرایی، خطای برش با ادامه تکرار کاهش می‌یابد.

خطای گرد کردن

کلیه دستگاه‌های محاسباتی اعداد را تا حدودی غیر دقیق نمایش می‌دهند، به جزء اعداد صحیح. در کامپیوترهای عددی که تقریباً همیشه از اعداد بصورت ممیز سیّار *(floating point)* با طول ثابت استفاده می‌کنند، مقادیر حقیقی بطور دقیق بیان نمی‌شوند. ما این خطا را خطای گرد کردن می‌نامیم، فرقی نمی‌کند کسر اعشاری بعد از آخرین رقم گرد شده یا بریده شده باشد، این را با تفصیل بیشتری در زیر بحث خواهیم کرد.

عددهایی را که کاملاً می‌توان با نمادی مشخص نمود، عددهای حقیقی می‌نامیم مانند $679,5,3$ یا 0.169 یا 2.15، لیکن بعضی از اعداد را هر چند می‌توان بصورت نمادی مشخص کرد مانند $\frac{1}{3}$، ولی این نماد در ماشین بصورت 0.3333333 که یک عدد اعشاری است نمایش داده شده و بکار می‌رود و مسلماً با مقدار تحقیقی اختلاف جزیی دارد، یا عدد حقیقی $\frac{37}{99}$ را می‌توان نوشت:

$$\frac{37}{99} = 0.37373737...$$

با توجه به محدودیتی که در تعداد ارقام در ماشین‌های محاسب وجود دارد مثلاً می‌توان تا 7 رقم اعشار آنرا بطور تقریب بکار برد. البته باید توجه داشت اکثر ماشین‌های محاسب عدد را گرد کرده و بصورت 0.3737374 تبدیل می‌کند.

چنانچه $\frac{37}{99}$ را در نظر بگیریم $\frac{37}{99} = 0.37373737$ اگر در کامپیوتر بصورت اعشاری تا 7 رقم اول نمایش داده شود، $0.3737373 = 37/99$، در اینصورت این عدد دارای خطایی برابر 10^{-7} × $0.7373...$ می‌باشد. برای آنکه این خطا را برای اعداد به حداقل ممکن برسانیم در محاسبات به اعداد مورد نظر مقدار 0.5×10^{-7} افزوده و سپس از رقم هشتم به بعد صرفنظر می‌کنیم. مانند:

$$0.3737373737...$$

$$0.00000005$$

$$0.3737374237...$$

با این روش، عدد 0.3737374 به عنوان گرد شده عدد مورد نظر بکار می‌رود. که خطای آن همواره از 0.5×10^{-7} کمتر است. و انجام این عمل در اکثر کامپیوترها پیش‌بینی شده است.

خطای ناشی از داده‌های اولیه

مسایل دنیای واقعی که در آن یک وضعیت فیزیکی موجود فرضی بوسیله یک معادله ریاضی مدل سازی می‌شود. اغلب دارای ضرایبی هستند که بطور کامل شناخته شده نیستند. دلیل، مشکلاتی است که مربوط به دقت در اندازه‌گیری است. تحلیل‌گر سیستم برای غلبه بر چنین خطاهایی هر روشی را انتخاب کند هیچ کاری نمی‌تواند انجام دهد، اما او لازم است از چنین عدم دقت‌هایی آگاه باشد، بخصوص ممکن است لازم باشد که آزمونهایی انجام دهد تا ببیند نتایج تا چه

اندازه نسبت به تغییرات اطلاعات ورودی حساس هستند. چون دلیل تشکیل محاسبات آن است که تصمیماتی را اتخاذ کنیم که در جهان واقعی معتبر باشند، تحلیل حساسیت از اهمیت فوق العاده‌ای برخوردار است. همانطور که هامینگ *(Hamming)* می‌گوید: «مقصود از محاسبه پیدا کردن بینش است، نه عدد.»

اشتباه

پیش بینی می‌شود که شما یک کامپیوتر عددی (یا حداقل یک حسابگر الکترونیکی) را در حرفه خودتان برای آنالیز عددی استفاده خواهید کرد. محتملاً چنین وسایل محاسباتی را هنگامی که در حال فراگیری موضوعات بحث شده در این کتاب هستید، بطور وسیعی بکار خواهید برد. چنین دستگاههایی به ندرت اشتباه می‌کنند، امّا چون فرد برنامه نویس در حال آماده کردن عملیات و تعبیر پاسخها است، در حال برنامه نویسی، اشتباه و یا خطاهای فاحش تا حدی بیشتر از آنچه مورد نظر است رخ می‌دهند. راه حل در اینجا توجه بیشتر همراه با آزمایش دقیق نتایج است تا ببینیم که معقول و مستدل هستند یا خیر. گاهی اوقات انجام یک آزمایش که نتایج آن را می‌دانیم با ارزش است، اما این هیچ تضمینی برای عاری بودن از خطایی تعجب انگیز نیست.

انتشار خطاها

این خطا مستلزم دقت بیشتر نسبت به سایر خطاها است. انتشار خطا مقصود خطایی است که در هر مرحله از پیشرفت عمل و در وابستگی خطای مرحله قبلی پیش می‌آید. این خطا علاوه بر خطای موضعی بوجود آمده در آن مرحله است و آن چیزی شبیه به خطا در شرایط اولیه است. در بعضی از روش‌های پیدا کردن ریشه، با تغییر تابع و برداشتن اولین ریشه از آن می‌توان سایر ریشه‌ها را از معادله جدید بدست آورد. این روش کاهش معادله نامیده می‌شود.

در اینجا معادله کاهش یافته خطاهای مراحل قبلی را منعکس می‌کند. در مثال‌های عددی که در فصل آتی بحث می‌شود. انتشار خطاها از اهمیت خاص برخوردار هستند. اگر خطاها با ادامه روش بطور پیوسته بزرگ شوند، سرانجام مقدار حقیقی را کاملاً تحت الشعاع قرار می‌دهند و آنرا از اعتبار ساقط می‌کنند، ما چنین روشی را ناپایدار نامیم. برای روش پایدار - نوع مطلوب - خطاهای ایجاد شده در نقاط قبلی با ادامه روش از بین می‌روند. این مطلب بطور کاملتر در فصل‌های بعدی خواهد آمد.

هریک از این انواع خطا، در حالیکه تا اندازه‌ای در رابطه با یکدیگر عمل می‌کنند، ممکن است حتی در صورت عدم وجود سایر خطاها رخ دهند. بعنوان مثال مانند یک روش تحلیلی، خطای گردکردن ممکن است رخ دهد حتی اگر خطای برش وجود نداشته باشد. بطور مشابه، خطای برش می‌تواند سبب عدم دقت شود حتی اگر بتوانیم دقت کافی را در محاسبه حفظ کنیم.

محاسبه ممیز سیّار و برآورد خطا

برای اینکه گردکردن عدد را به تفصیل مورد بررسی قرار دهیم، ضروری است بدانیم کمیت‌های عددی چگونه در کامپیوترها نمایش داده می‌شوند. تقریباً در تمام حالات اعداد بصورت کمیت‌های با ممیز سیّار ذخیره می‌شوند که بسیار شبیه به نماد علمی هستند. بعنوان مثال، عدد اعشاری ثابت 13.524 برابر عدد ممیز سیّار 0.13524×10^2 می‌باشد که اغلب بصورت $0.13524E2$ نمایش داده می‌شود. مثال دیگر عدد $0.0442-$ برابر $0.442E-1-$ می‌باشد.

کامپیوترهای مختلف روش‌های کمی متفاوت را استفاده می‌کنند، اما طرز عمل کلی مشابه است.

ما می‌توانیم یک عدد ممیز سیّار را به فرم کلی زیر بنویسیم.

$$0.\ d_1 d_2 ... d_p * B^e$$

بطوری که d_i ها اعداد یا بیت‌ها با مقادیر صفر تا $B-1$ می‌باشند.

B = عددی در مبنای معمولاً 2 یا 16 و یا 10 می‌باشد.

P = تعداد بیت‌های با معنی (ارقام با معنی)، یعنی دقت.

e = نمای صحیح در محدوده مشخص تعریف شده روی فاصله $[E_{min}, E_{max}]$ (دامنه شامل مقادیر منفی چون مقادیر مثبت می‌باشد).

ارقام با معنی (بیت‌ها) قسمت کسری عدد را تشکیل می‌دهند. تقریباً در تمام موارد، اعداد نرمال می‌شوند، یعنی که ارقام کسری انتقال پیدا می‌کنند و نما تغییر می‌کند، بطوریکه d_1 غیر صفر می‌شود. عدد صفر حالت خاص است، آن معمولاً قسمت کسری صفر و نمای صفر دارد. این نوع از صفر عدد نرمال نیست و نمی‌تواند باشد. این تقریباً در تمام سیستم‌های کامپیوتری متناظر استاندارد است.

برای محاسبات دستی مبنای B معمولاً 10 می‌باشد. کامپیوترها اغلب بر اساس مبنای 2 می‌باشند، امّا مبنای دیگر، مثل 16 گاهی اوقات استفاده می‌شود.

اکثر کامپیوترها دو یا حتی سه نوع عدد اعشاری را اجازه استفاده می‌دهند: دقت ساده که معادل 6 تا 7 ارقام اعشاری با معنی است، دقت مضاعف، معادل 16 رقم اعشاری با معنی است؛ و دقت گسترده، که ممکن است معادل 19 تا 20 رقم اعشاری باشد.

مثالهائی از اعداد همانگونه که در کامپیوتر نشان داده می‌شوند، ذکر می‌گردد.

کار با ارقام دوتائی یا شانزده تائی با دست ناخوشایند است، بنابراین ما با مبنای آشناتر 10 مثالها را شروع می‌کنیم. فرض می‌کنیم $B=10$ و $P=4$. سپس این اعداد نشان داده می‌شوند:

$$27.39 \quad \rightarrow \quad +.2739*10^2;$$

$$-0.00124 \rightarrow \quad -.1240*10^{-2};$$

$$37000 \quad \rightarrow \quad +.3700*10^5$$

مشاهده می‌کنیم که کسر نرمال شده است و اولین عنصر d_1 غیر صفر است. اگر مبنا 2 باشد، بدین

معنی است که اولین بیت کسر همیشه *1* می‌باشد. بعضی سیستم‌های کامپیوتری از این حقیقت سود برده‌اند و اولین بیت را ذخیره نمی‌کنند و یک بیت دقت بیشتر دارند. (در این حالت به آن بعنوان بیت پنهان اشاره می‌شود).

اگر چه اعداد حقیقی روی هر فاصله حقیقی نامحدود هستند، اما این برای اعداد با ممیز سیّار صحت ندارد. چون تعداد بیت‌های بکار رفته برای عدد ممیز سیّار ثابت است، تعداد محدودی عدد از مقادیر مجزا در سیستم اعداد کامپیوتر وجود دارد –در مغایرت عظیم با سیستم اعداد حقیقی. از اینرو، فاصله‌هائی در بین سیستم اعداد کامپیوتری وجود دارند.

جهت نمایش، یک حالت بسیار ساده شده وقتی که $B=2$ و $P=2$ و $-2 \le e \le 3$ می‌باشد، بکار می‌بریم. برای این سیستم، تمام اعداد نرمال شده به یکی از اشکال زیر می‌باشند:

$$\pm\ 0.10_2 * 2^e \quad \text{یا} \quad \pm\ 0.11_2 * 2^e\ ,\ -2 \le e \le 3$$

زیرا که برای کسرها در مبنای 2 و $0.10_2 = \frac{1}{2}, 0.11_2 = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$

در دامنه $6+$ یا $6-$ $(0.11 \times 2^3 = 6,\ -0.11_2 \times 2^3 = -6)$ و یک لیست از همه اعداد مثبت در این سیستم برابرند با:

$$.10_2 * 2^{-2} = \frac{1}{8}, \qquad .11_2 * 2^{-2} = \frac{3}{16},$$

$$.10_2 * 2^{-1} = \frac{1}{4}, \qquad .11_2 * 2^{-1} = \frac{3}{8},$$

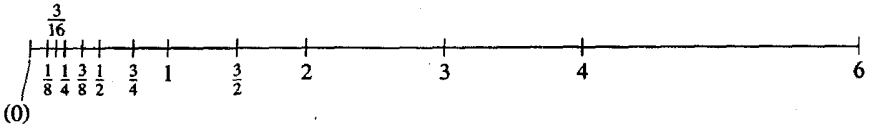
$$.10_2 * 2^{0} = \frac{1}{2}, \qquad .11_2 * 2^{0} = \frac{3}{4},$$

$$.10_2 * 2^{1} = 1, \qquad .11_2 * 2^{1} = \frac{3}{2},$$

$$.10_2 * 2^{2} = 2, \qquad .11_2 * 2^{2} = 3,$$

$$.10_2 * 2^{3} = 4, \qquad .11_2 * 2^{3} = 6.$$

تصویر زیر توزیع این مقادیر مثبت را نشان می‌دهد. اعداد منفی بطور مشابه توزیع می‌شوند. فاصله‌های خالی از نظر اندازه متفاوت هستند. این فواصل خالی و فواصل ته ریز و سرزیر شرح داده خواهند شد.



مشاهده می‌شود که صفر نشان داده نشده است. برای نمایش صفر، یک حالت خاص را تعریف می‌کنیم. صفر مقداری است که کلیهٔ مقادیر اعشاری آن صفر هستند (نرمال نیز نمی‌شود). همچنین نمای آن صفر هست و این استاندارد در تمام سیستم‌های کامپیوتری وجود دارد. به نتیجه مهم این

فواصل در کلیه سیستم‌های عددی کامپیوتر توجه می‌کنیم. در سیستم کوچک که در تصویر قبلی نمایش داده شد، مقدار *2.3* برابر *2* ذخیره می‌شود؛ دو مقدار *2.2* و *2.4* دقیقاً یکسان هستند. این برای کلیه مقادیر بین *2* و *3* صادق است.

این شرح می‌دهد که چرا در برنامه نصف کردن حلقه نامحدود خواهد بود اگر خطای قابل اغماض بسیار کوچک باشد. در یک حالت، بعد از *21* تکرار $x_1 = 0.4677237$ $x_2 = 0.4677238$ و $x_3 = 0.4677238$ ؛ کامپیوتر نمی‌تواند بین x_2 و x_3 فرق قائل شود. بنابراین دیگر بیشتر از این نمی‌توان به مقدار دقیق ریشه نزدیک شد.

سیستم‌های کامپیوتری

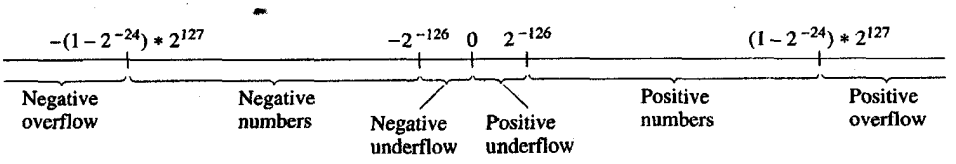
در سیستم‌های کامپیوتری موجود اعداد ممیز سیّار، به شکلی نیست که به آسانی فراگرفته شود، زیرا که مبنا عدد *10* نیست.

جدول ۱.۱ ترکیب سه سیستم مختلف را مقایسه می‌کند. حقیقتاً این سیستمها یکسان نیستند و این معنی را می‌دهد که کامپیوترهای مختلف می‌توانند نتایج مختلفی از همان مجموعه محاسبات بدهند، در اینصورت برنامه‌های ایجاد شده همیشه از نظر دقت قابل انتقال نیستند.

شکل ۷.۱ دامنه‌های اعداد قابل نمایش در *IEEE* استاندارد در جدول ۱.۱ را نشان می‌دهد.

جدول ۱.۱ مقایسه چند روش برای ذخیره اعداد ممیز سیّار

Method	Total length	Bits in fraction	Bits in exponent	Bias value††	Base	Max. exponent	Min. exponent	Largest number	Smallest number	Approx. precision, no. decimal digits
IEEE										
single	32	23*	8	127	2	127	−126	1.701E38	1.755E-38	7
double	64	52*	11	1023	2	1023	−1022	8.988E307	2.225E-308	16
extended	80	64	15	16383	2	16383	−16382	6E4931	3E-4931	19
VAX										
single	32	23	8	127	2	127	−127	1.701E38	5.877E-39	7
double-1	64	55	8	127	2	127	−127	1.701E38	5.877E-39	16
double-2	64	52	11	1023	2	1023	−1023	8.988E307	1.123E-308	15
extended	128	112	15	16383	2	16383	−16383	6E4931	1E-4931	33
IBM										
single	32	24	7	63	16	63	−64	7.237E75	8.636E-78	7
double	64	56	7	63	16	63	−64	7.237E75	8.636E-78	16
extended	128†	112	7	63	16	63	−64	7.237E75	8.636E-78	33



شکل ۷.۱ اعداد در حالت استاندارد *IEEE*

۵.۱ محاسبه و خطا

این برای ما سخت است که در مبنای بجز *10* فکر کنیم، بنابراین در بحث حساب دقت عملیات ممیز سیّار در مبنای *10* نرمال را مورد استفاده قرار می‌دهیم.

در سایر مبناها که واقعاً در کامپیوتر بکار می‌رود رفتار بطور مشابه است. برای ساده کردن بحث، سه رقم در قسمت کسر و یک رقم برای نما در نظر می‌گیریم. علامت کسر و نما را جداگانه در نظر می‌گیریم. برای این مثالها، $B=10$ و $P=3$ و $-9 \le e \le 9$ و گردکردن و برش دادن را مقایسه می‌کنیم.

وقتی دو عدد ممیز سیّار جمع یا تفریق می‌شوند، ارقام کسری عدد با نمای کوچکتر باید انتقال پیدا کند (نماهای مساوی) و نقاط اعشاری زیر هم قرار گیرند، ممکن است لازم باشد انتقال داده شده و نما برای نرمال کردن نتیجه تنظیم شود.

این انتقال می‌تواند بعضی از ارقام با معنی آن را از دست بدهد. ممکن است لازم باشد ارقام نتیجه را انتقال دهیم تا نما تغییر و عدد نرمال شود. بعضی کامپیوترها بطور اتوماتیک جواب نهائی را گرد می‌کنند، سایر کامپیوترها ارقام اضافی بعد از دقت سیستم را برش می‌دهند. موقع ضرب (تقسیم) اعداد، قسمت‌های کسر فقط ضرب می‌شوند (تقسیم می‌شوند) و نماها جمع (تفریق) می‌گردند و نتیجه نرمال می‌شود. در هر یک از عملیات ملاحظه می‌شود بخاطر محدودیت در ذخیره تعداد ارقام اعشاری که در نتیجه باعث گرد کردن یا برش قسمت کسری می‌گردد، منبع اصلی خطا بوجود می‌آید.

بعضی مثالها در ادامه می‌آیند.

مثال ۲.۱ محاسبه کنید:

$$1.37 + .0269 = .137 * 10^1 + .269 * 10^{-1}.$$

$$\begin{array}{r} .137 \quad * 10^1 \\ +\ .00269 * 10^1 \\ \hline .13969 * 10^1 \end{array} \quad \text{Align decimal points.}$$

برش دادن → $.139 * 10^1.$

گرد کردن → $.140 * 10^1.$

گرد کردن، یک عمل اضافه لازم دارد، که ممکن است در سخت افزار یا در برنامه نرم افزار انجام گیرد.

مثال ۳.۱ محاسبه کنید:

$$4850 - 4820 = .485 * 10^4 - .482 * 10^4.$$

$$\begin{array}{r} .485 * 10^4 \\ -\ .482 * 10^4 \\ \hline .003 * 10^4 \\ .300 * 10^2 \quad \text{Normalize.} \end{array}$$

برش دادن → $.300 * 10^2.$

گرد کردن → $.300 * 10^2.$

در مثال ۳.۱ مشاهده می‌شود که واقعاً فقط یک رقم دقت در نتیجه وجود دارد اگرچه تفاضل نشان می‌دهد که صفرهای انتهائی با معنی هستند. این از دست دادن دقت وقتی دو عدد مساوی از هم کم می‌شوند یک منبع اصلی خطا در عملیات ممیز سیّار است. در این حالت، گردکردن و برش دادن به یک جواب نهائی می‌رسند.

مثال ۴.۱ محاسبه کنید: $.3780 - .321 = .378 * 10^4 - .321 * 10^0.$

$$\left.\begin{array}{r} .378 * 10^4 \\ -\ .0000321 * 10^4 \end{array}\right\} \text{Align decimal points.}$$
$$.3779679 * 10^4$$

برش دادن → $.377 * 10^4.$
گرد کردن → $.378 * 10^4.$

در مثال ۴.۱ داده انتقال یافته و نقاط اعشاری زیر هم قرار می‌گیرند و ارقام با معنی تفاضل حذف می‌شوند.

مثال ۵.۱ محاسبه کنید: $.403000 * .0197 = .403 * 10^6 * .197 * 10^{-1}.$

$$\left.\begin{array}{r} .403 \\ *.197 \end{array}\right\} \text{نرمال کردن} \qquad \left.\begin{array}{r} 6 \\ -1 \end{array}\right\} \text{جمع نماها}$$
$$.079391 \qquad 5$$

نرمال کردن $.079391 * 10^5$
Normalize. $.79391 * 10^4$

برش دادن → $.793 * 10^4.$
گرد کردن → $.794 * 10^4.$

در ضرب دو عدد n رقمی، نتیجه تا $2n$ رقم خواهد بود. در اینصورت رجیستر (ثبات گر) با طول دو برابر بکار می‌رود (یا دو ثبات‌گر با طول معمولی بکار می‌رود.) جواب نهائی به طول یک رجیستر گرد شده یا برش داده شده می شود.

مثال ۶.۱ محاسبه کنید: $.0356/1560 = .356 * 10^{-1} / .156 * 10^4.$

$$\left.\begin{array}{r} .356 \\ \div\ .156 \end{array}\right\} \qquad \left.\begin{array}{r} -1 \\ -4 \end{array}\right\}$$
$$2.28205 \qquad -5$$

نرمال کردن $2.28205 * 10^{-5}$
Normalize. $.228205 * 10^{-4}$

برش دادن → $.228 * 10^{-4}$
گرد کردن → $.228 * 10^{-4}$

زمان لازم برای تشکیل عملیات مختلف محاسبات متغیر است. ضرب تا *10* برابر کندتر از جمع و تفریق است. تقسیم ممیز سیّار از همه کندتر است و تا *25* برابر کندتر از جمع است. این نتایج مربوط به کامپیوترهای سال‌های ۹۰ می‌باشند.

در بعضی از این کامپیوترها، ضرب و تقسیم با تشکیل برنامه‌های نرم افزاری انجام می شدند که این اختلاف زمانی بیشتر می‌شود.

امروزه در کامپیوترهای شخصی با استفاده از دستگاه «هم پردازشگر» ریاضی اختلاف زمان محاسبه بسیار کمتر می‌شود.

هم پردازشگر ریاضی نشان می‌دهد که زمان ضرب تقریباً 3 درصد بیشتر از جمع و تقسیم 9 درصد بیشتر از جمع است.

خطا در تبدیل مبنای اعداد

چون اعداد اعشاری موقع ذخیره شدن بصورت اعداد ممیز سیّار در مبنای 10 تبدیل می‌شوند این اعداد به عددی دیگر معمولاً در مبنای 2 تبدیل می‌شوند، تعداد ارقام اعشاری با معنی، معادل با یک کسری از یک عدد صحیح نیست. این تبدیل خود باعث خطا می‌شود. سازندگان کامپیوترهای مختلف به مسئله نمایش ممیز سیّار با تغییرات وسیعی از طول بیت‌ها با طول مختلف پرداخته‌اند به طوریکه دقت فراهم شده بسیار متفاوت است (به عنوان مثال در دقت ساده معادل 6 تا 16 رقم اعشاری است) تبدیل اعداد به مبنای عددی داخل کامپیوتر اغلب باعث خطایی خواهد شد. کسرهای اعشاری با ارقام متناهی ممکن است در مبنای دو و شانزده دارای تعداد ارقام نامتناهی گردند. $[(0.100110011001...)_2 = (0.6)_{10}]$. بعلاوه در مجموعه عددی که از یک طول متناهی از بیت‌ها تشکیل می‌شود جاهای خالی بسیاری در یک مجموعه عددی خود دارد، یعنی تعداد مقادیر مختلف قابل نمایش یک مجموعه متناهی است در نتیجه، مجبوریم مجموعه نامتناهی اعداد ریاضی را به یک مجموعه متناهی از اعداد کامپیوتر بنگاریم. برای یک مثال ساده شده کسرهای سه رقمی، فقط 900 مقدار کسری مختلف وجود دارد. کلیه اعداد ریاضی بین 0.1 و 1 باید به یکی از این 900 مقدار نگاشته شوند. بازای هر دهه، (یکان، یا دهگان یا صدگان یا،...) بوسیله یک مقدار ثابتی از نما نمایش داده می‌شود. (0.234×10^2) بنابراین فضای بین مقادیر در هر دهه با دهه دیگر مختلف است. صفر یک موقعیت خاصی را در میان اعداد با ممیز سیّار دارد. البته نمی‌توان آنرا نرمال نمود. بنابراین قراردادهای مخصوصی پذیرفته می‌شود. در اکثر دستگاهها، همه ارقام کسر صفر هستند. نما نیز باید کوچکترین مقدار منفی قرار داده شود، اگر این چنین نشود، برای زیر هم قرار گرفتن نقاط اعشاری در موقع جمع، ارقام با معنی از عدد افزوده شده به خارج انتقال می‌یابد. صفر تا حدودی از سایر اعداد مجزاست. در دستگاه مثال ساده مذکور، نزدیکترین همسایگی به صفر $\pm0.1\times10^{-9}$ می‌باشند. کوشش برای نمایش هر مقدار مطلق کوچکتر از این سبب یک خطای برنامه به‌نام ته‌ریز شدن نما از حد مجاز می‌شود در بعضی زبان‌های برنامه نویسی هرگاه عدد از حد مجاز کمتر شود عدد مذکور بجای آن جایگزین می‌شود و اجرای برنامه ادامه می‌یابد. بطور مشابه، کوشش برای نمایش اعدادی با مقدار مطلق بزرگتر از 0.999×10^9 در دستگاه باعث سرریز شدن نما می‌شود. معمولاً این سبب خاتمه اجرا می‌شود، اما بعضی دستگاهها بزرگترین مقدار با ممیز سیّار را

ممکن است جایگزین این عدد کرده و عمل را ادامه دهند. در محاسبه ممیز سیّار چیزهای عجیبی اتفاق می‌افتد. به عنوان مثال اگر هزار بار *0.001* را با خودش جمع کنیم برابر *1.0* نمی‌شود. در بعضی موارد، حاصلضرب یک عدد در عدد یک، آن عدد را دوباره نمی‌دهد. در بسیاری محاسبات، تغییر ترتیب عملیات محاسبات، نتایج را عوض می‌کند.

ماشین *EPS*

یک خصوصیت مهم در حساب کامپیوتر اینست که کامپیوتر تا چه اندازه اختلاف کوچک بین دو اندازه مختلف را می‌شناسد. این کمیت کامپیوتر *eps* اپسیلن نامیده می‌شود، بطوری که *"eps"* مخفف حرف یونانی اپسیلن *epsilon* می‌باشد. این اندازه از دقت ماشین، استاندارد شده است تا کوچکترین عدد ممیز سیّار پیدا شود تا وقتی به ممیز سیّار *1.000* اضافه شود، نتیجه‌ای غیر از *1.000* را بدهد. (اعداد کوچکتر از *eps* در کامپیوتر تأثیر صفر را دارند.) در تمرینات، ممکن است از شما سؤال شود اندازه اپسیلن *eps* ماشین در دسترس خود را معین کنید.

تعیین *eps* ممکن است حتی به زبان کامپیوتری که یک برنامه نوشته شده است بستگی داشته باشد، زیرا که بعضی از زبانها فقط دقت مضاعف را برای اعداد ممیز سیّار خود بکار می‌برند.

خطای مطلق، خطای نسبی، ارقام با معنی

خطای مطلق نتیجه محاسبات اغلب برای اندازه‌گیری دقت بکار می‌رود.

دقت در هر محاسبه همیشه اهمیت زیادی دارد. دو روش معمولی برای بیان اندازه خطا در نتیجه محاسبه وجود دارد: خطای مطلق و خطای نسبی.

$$\text{خطای مطلق} = |\text{مقدار تقریبی} - \text{مقدار تحقیقی}|$$

$$\text{خطای مطلق} = |Q - \bar{Q}|$$

بنابراین مقدار تحقیقی (Q) با اضافه کردن خطای مطلق به مقدار تقریبی بدست می‌آید، یک خطای داده شده معمولاً وقتیکه مقدار تحقیقی کوچک است بسیار جدّی تلقی می‌شود. بعنوان مثال: 1036.52 ± 0.010 تا پنج رقم با معنی صحیح است و غالباً دارای دقت کافی می‌باشد، در صورتیکه داشته باشیم 0.010 ± 0.005 به وضوح یک خطای فاحش است. بکار بردن خطای نسبی راهی برای جبران این مشکل است.

$$\text{خطای نسبی} = \frac{|Q - \bar{Q}|}{Q}$$

$$\text{خطای نسبی} = \frac{|\text{مقدار تقریبی} - \text{مقدار تحقیقی}|}{|\text{مقدار تحقیقی}|}$$

خطای نسبی، اغلب دقت را بهتر نمایش می‌دهد. خطای نسبی از واحد اندازه گیری تقریباً مستثنی است که این یکی از مطلوبترین ویژه‌گی‌های آن می‌باشد. وقتی مقدار تحقیقی صفر است، خطای نسبی تعریف نشده است. خطای گردکردن مربوط به طول کسر متناهی در اعداد با ممیز سیّار وقتی بصورت نسبی بیان شود، تقریباً ثابت است، توجه کنید که وقتی تفاضل دو عدد ممیز سیّار تقریباً مساوی را بدست می‌آوریم، خطای نسبی فاحشی بوجود می‌آید.

هرچه خطای نسبی کوچکتر باشد مبین آنستکه اندازه‌گیریها و محاسبات دقیقتر هستند. چون به مقدار تحقیقی Q دسترسی نداریم از ΔQ که در رابطه زیر صدق می‌کند و حد بالای خطای مطلق نامیده می‌شود استفاده می‌کنیم.

$$|Q - \bar{Q}| \leq \Delta Q$$

و همچنین برای تعیین خطای نسبی خواهیم داشت:

$$|Q| = |\bar{Q} - (\bar{Q} - Q)| \geq |\bar{Q}| - |\bar{Q} - Q| \geq |\bar{Q}| - \Delta Q$$

پس حد بالای خطای نسبی برابر است با:

$$r \text{ خطای نسبی} \leq \frac{\Delta Q}{|\bar{Q}| - \Delta Q}$$

ارقام با معنای صحیح

نتیجه محاسبه ممکن است عددی تقریبی باشد و با مقدار تحقیقی واقعی تفاوتی داشته باشد که همان خطای محاسبه است. بر حسب اندازه این خطا، تعداد معینی از ارقام عدد تقریبی برای نشان دادن عدد تحقیقی بکار می‌رود. مثلاً اگر نتیجه محاسبات بطور تقریب $\frac{5}{6}$ باشد در اینصورت ممکن است این سوال پیش آید که کدامیک از دو عدد زیر تقریب بهتری دارند.

$$\frac{5}{6} \approx 0.8333333 \qquad \frac{5}{6} \approx 0.83965931$$

عدد اول دارای 7 رقم با معنی و خطای آن کمتر از 0.5×10^{-7} می‌باشد، در صورتیکه عدد دوم با اینکه دارای 9 رقم با معنی است، لیکن خطای آن حتی از 0.5×10^{-3} بیشتر است، زیرا که عدد اول دارای 7 رقم با معنی صحیح و عدد دوم دارای دو رقم با معنی صحیح می‌باشد. بطور کلی عدد تقریبی $f \times 10^{e}$ دارای n رقم با معنی صحیح است اگر از رقم $(n+1)$ ام ببعد f صرفنظر کنیم مقدار خطای مطلق از $5. \times 10^{-n}$ کمتر باشد و بالعکس اگر خطای مطلق کمتر از $5. \times 10^{-n}$ باشد در اینصورت f تا n رقم با معنی صحیح است. در ماشین‌های کامپیوتری که قسمت کسری اعداد بصورت ممیز سیّار دارای 7 رقم با معنی هستند، خطای مطلق قسمت کسری f کمتر از $5. \times 10^{-7}$ می‌باشد.

از حد بالای خطای نسبی یک عدد می‌توان حداقل تعداد رقم‌های با معنی صحیح آنرا محاسبه نمود و برعکس چنانچه تعداد رقم‌های صحیح یک عدد رابدانیم، می‌توانیم حد بالای خطای نسبی را محاسبه نمائیم. فرض کنیم عدد 263.782 کلیه ارقامش با معنای صحیح است. حد بالای خطای نسبی آنرا بدست می‌آوریم.

می‌دانیم که خطای مطلق آن از نصف ارزش واحد آخرین رقم بیشتر نیست.

$$\Delta Q \le \frac{1}{2} \times 10^{-3}$$

حدبالای خطای نسبی $r = \dfrac{\Delta Q}{|\bar{Q}| - \Delta Q} \le \dfrac{.0005}{263.782 - 0.0005} = \dfrac{.5}{263782 - 0.5}$

$$r < \frac{1}{527563}$$

هر چند که این محاسبات نتیجه دقیقی را می‌دهد لیکن روشی خسته کننده است.

اما اگر به حد بالای خطای نسبی به صورت 5×10^{-n} توجه کنیم، خواهیم داشت:

$$r < \frac{1}{527563} < \frac{1}{500000} < \frac{1}{200000} = 5 \times 10^{-6}$$

که معرف آنستکه خطای نسبی عددی با 6 رقم با معنای صحیح کمتر از 5×10^{-6} می‌باشد. در حالت اول نشان دادیم که خطای نسبی از $\frac{1}{527563}$ کمتر است در صورتیکه اکنون از خطای نسبی کمتر از $\frac{1}{200000}$ استفاده می‌کنیم. در حالت اوّل، اطلاعات اضافی بدست آمده، دارای ارزش عملی زیادی نمی باشد، بنابراین برای برآورد خطای نسبی و همچنین تعداد ارقام با معنی صحیح روش ساده‌تری را بر می‌گزینیم، هر چند که تااندازه‌ای از دقت کاسته می‌شود، لیکن دارای ارزش محاسباتی بسیار زیادی می‌باشد.

قضیه ۱.۱ اگر قسمت کسری عددی دارای n رقم با معنی صحیح باشد، خطای نسبی آن عدد از 5×10^{-n} کمتر است. به جز در حالتی که این عدد برابر است با: $0.1000...000 \times 10^{e}$

اثبات: اگر قسمت کسری Q را تا رقم n ام گرد کنیم،

ملاحظه می‌کنیم: $\Delta Q \le .5 \times 10^{-n} \times 10^{e}$

اکنون حد بالای خطای نسبی را برای Q بدست می‌آوریم:

$$\frac{\Delta Q}{|Q|} \le \frac{\Delta Q}{|\bar{Q}| - \Delta Q} = \frac{10^{n-e}\,\Delta Q}{10^{n-e}|\bar{Q}| - 10^{n-e}\Delta Q} = \frac{.5}{10^{n-e}|\bar{Q}| - .5}$$

امّا طبق فرض $|\bar{Q}| \ne .1 \times 10^{e}$ ، پس $|\bar{Q}| = .1 \times 10^{e} + 10^{-n} \times 10^{e}$

کوچکترین مقدار $\dfrac{\Delta Q}{|Q|} < \dfrac{.5}{10^{n-e} \times .1 \times 10^{e} + 1 - .5}$

$$\frac{\Delta Q}{|Q|} < \frac{.5}{.1 \times 10^{n}} = 5 \times 10^{-n}$$

در حالت خاصی که $\bar{Q} = .1 \times 10^{n}$ خطای نسبی را محاسبه می‌کنیم:

n	$\bar{Q}$	خطای نسبی
1	1	$\frac{.5}{.5} = 1$
2	10	$\frac{.5}{9.5} = \frac{1}{1.9 \times 10^{1}} > \frac{1}{2 \times 10^{1}}$
3	100	$\frac{.5}{99.5} = \frac{1}{1.99 \times 10^{2}} > \frac{1}{2 \times 10^{2}}$
4	1000	$\frac{.5}{999.5} = \frac{1}{1.999 \times 10^{3}} > \frac{1}{2 \times 10^{3}}$

مشاهده می‌شود که در سایر حالات قضیه ۱.۱ تقریب بسیار خوبی از حد بالای خطای نسبی را می‌دهد.

اکنون به طریق عکس، اگر خطای نسبی عددی را بدانیم، می‌خواهیم تعداد رقم‌های با معنی صحیح آن عدد را تخمین بزنیم.

قضیه ۲.۱ اگر خطای نسبی یک عدد تقریبی کوچکتر یا مساوی $5 \times 10^{-(n+1)}$ باشد، در اینصورت عدد تا n رقم با معنی صحیح است.

اثبات خطای نسبی بیشتر از $5 \times 10^{-(n+1)}$ نمی باشد، در اینصورت داریم:

$$\frac{\Delta Q}{|Q|} \leq 5 \times 10^{-(n+1)}$$

$$\frac{\Delta f}{|f|} \leq 5 \times 10^{-(n+1)}$$

$$\Delta f \leq 5 \times 10^{-(n+1)} \ |f|$$

فرض کنیم $Q = f \times 10^{e}$ و $0.1 \leq |f| < 1$

$$\Delta f < 5 \times 10^{-(n+1)} = .5 \times 10^{-n}$$

در اینصورت عدد تقریبی Q تا n رقم با معنی صحیح است.

انتشار خطا در توابع

فرض کنیم برای مقدار دقیق x، $f(x)$را بتوان دقیقاً معین نمود، در اینصورت چنانچه x دارای خطا باشد، می‌خواهیم خطای $f(x)$را مشخص نمائیم. البته در صورتیکه $f(x)$به ازای مقدار معین x، دارای خطائی حداکثر برابر r باشد، این خطا به نتیجه حاصل از انتشار خطا افزوده می‌شود. لیکن در مواردی که خطای برش و خطای گرد کردن در مقایسه با سایر منابع خطا بسیار کوچک باشند، می‌توان از آنها صرفنظر کرد.

گیریم $y = f(x_1,x_2,...,x_n)$ که y مقدار دقیق تابع fدر نقطهٔ $(x_1,x_2,...,x_n)$ می‌باشد. اگر تغییر کوچکی در $x_1,x_2,...x_n$رخ دهد این تغییرات را بترتیب با $\Delta x_1,\Delta x_2,...,\Delta x_n$ نمایش می‌دهیم. در اینصورت خواهیم داشت:

$$dy = \frac{\partial f}{\partial x_1}\Delta x_1 + \frac{\partial f}{\partial x_2}\Delta x_2 + ... + \frac{\partial f}{\partial x_n}\Delta x_n \qquad (1)$$

کمیت dy تقریباً برابر Δy یعنی انتشار خطا در y می‌باشد.

$$\Delta y \approx \frac{\partial f}{\partial x_1}\Delta x_1 + \frac{\partial f}{\partial x_2}\Delta x_2 + ... + \frac{\partial f}{\partial x_n}\Delta x_n \qquad (2)$$

مثال ۷.۱ تابع $u = x\,y^2\,z^3$ را بازاء $x=37.1$, $y = 9.87$, $z = 6.052$ حساب کنید در صورتیکه می‌دانیم: $\Delta x = .3$, $\Delta y = .11$, $\Delta z = .016$

لگاریتم طرفین تساوی را بدست می‌آوریم:

$$Ln(u) = Ln(x) + 2Ln(y) + 3Ln(z)$$

$$\frac{du}{u} = \frac{dx}{x} + 2\frac{dy}{y} + 3\frac{dz}{z}$$

$$\frac{\Delta u}{u} \approx \frac{\Delta x}{x} + 2\frac{\Delta y}{y} + 3\frac{\Delta z}{z}$$

$$r_u \approx r_x + 2r_y + 3r_z$$

$$r_x = \frac{3}{371} = .0081$$

$$r_y = \frac{11}{987} = 0.0112$$

$$r_z = \frac{16}{6052} = 0.0026$$

$$r_u = .0081 + .0224 + .0078 = .0383$$

$$r_u = 3.8 \times 10^{-2} < 5 \times 10^{-2}$$

پس تعداد ارقام با معنای صحیح u را می‌توان 2 درنظر گرفت.

$$u = xy^2y^3 = .801 \times 10^6$$

می‌توانیم بجای بدست آوردن لگاریتم مستقیماً خطای مطلق و خطای نسبی را بدست آوریم:

$$du = y^2z^3\,\Delta x + 2xyz^3\,\Delta y + 3xy^2z^2\,\Delta z,$$

$$\frac{\Delta u}{u} \approx \frac{\Delta x}{x} + 2\frac{\Delta y}{y} + 3\frac{\Delta z}{z}$$

قبلاً ملاحظه کردیم که حتی اعداد تحقیقی نیز در ماشین در اثر گردشدن تغییر کرده و خطا ایجاد می‌گردد. تمام کمیت‌هائی که دارای خطا هستند، خواه ثابت یا متغیر، باید بصورت متغیر بکار برده شوند و سپس در محاسبهٔ خطا مورد استفاده قرار گیرند. البته در بسیاری از مسائل خطای برش و خطای گرد کردن درمقایسه با سایر منابع دیگر خطا بسیار کوچک هستند و می‌توان از آنها صرفنظر کرد.

مثال ۸.۱ تابع توانی $y = x^a$ مفروض است. خطای مطلق وخطای نسبی تابع y را حساب کنید.

$$dy = ax^{a-1}\,\Delta x$$

$$\frac{\Delta y}{y} = a\frac{\Delta x}{x}$$

$$r_y = a\,r_x$$

در اینصورت درتابع $y = x^3$ داریم $r_y = 3r_x$ یعنی خطای نسبی متغیر تابع سه برابر خطای نسبی متغیر xاست.

مثال ۹.۱ تابع نهائی $y = a^x$ مفروض است $(a>0)$ خطای مطلق و خطای نسبی y را محاسبه کنید.

$$dy = a^x \, Ln\,(a)\, dx$$

$$\Delta y = y \, Ln\,(a)\, \Delta x$$

$$\frac{\Delta y}{y} = Ln\,(a)\, \Delta x$$

$$r_y = Ln\,(a)\, \Delta x$$

و برای $a = e$ خواهیم داشت:

$$r_y = \Delta x$$

یعنی خطای نسبی متغیر تابع y برابر خطای مطلق متغیر مستقل x می‌باشد.

مثلاً برای $x = 85$ خطای مطلق برابر $\Delta x = 0.5$، و خطای نسبی $y=e^x$ مساوی است با:

$r_y = .5 = 5 \times 10^{-1}$ و مقدار y دارای حداکثر یک رقم با معنی صحیح می‌باشد.

در صورتی که $x=85.00001$ خطای مطلق $\Delta x = 0.5 \times 10^{-5}$ ، $r_y = 5 \times 10^{-6}$ و مقدار y دارای 6 رقم با معنی صحیح می‌باشد

مثال ۱.۱۰ تابع $y = a \sin(b)$ با فرض آنکه مقادیر $a = 30.0$ ، $b = .45$ تا رقم‌های با معنای صحیح نشان داده شده‌اند را محاسبه کرده و خطای مطلق و خطای نسبی y را حساب کنید.

$$dy = \frac{\partial y}{\partial a}\Delta a + \frac{\partial y}{\partial b}\Delta b$$

$$\Delta a = .5 \times 10^{-1} \quad , \quad \Delta b = .5 \times 10^{-2}$$

$$\Delta y \approx \sin(b)\, \Delta a + a\cos(b)\, \Delta b$$

$$\Delta y \approx (.4349656)\,(.05) + (30.0)(.9004471)(.005) = .1568152$$

$$\Delta y \approx .16$$ خطای مطلق

$$y = a\sin(b) = 13.048968 = 13$$ y دو رقم با معنی صحیح دارد

$$\frac{\Delta y}{y} = \frac{.16}{13} = .0123076 = 1.2 \times 10^{-2} < 5 \times 10^{-2}$$ زیرا که

مثال ۱.۱۱ مطلوبست محاسبه

$$I_r = \int_0^1 e^{4/3(x-1)}\, x^{r+3}\, dx$$

بازای $r = 0,1,...,15$

در صورتیکه روش جزء بجزء را بکار بریم.

$$I_r = \left[\frac{e^{4/3(x-1)}\, x^{r+3}}{(4/3)}\right]_0^1 - \int_0^1 (r+3)\, \frac{e^{4/3(x-1)}\, x^{r+2}}{(4/3)}\, dx$$

که یک رابطه بازگشتی بدست می‌آید.

$$\boxed{I_r = \frac{3}{4} - \frac{3}{4}(r+3)\, I_{r-1}}$$

I_0 را می‌توان به روش جزءبجزء یا با استفاده از بسط تابع به سری محاسبه کرد.

$$I_0 = \int_0^1 e^{4/3(x-1)}\, x^3\, dx = e^{-4/3} \int_0^1 e^{(4/3)x}\, x^3\, dx$$

$$I_0 = e^{4/3} \int_0^1 [1 + \frac{4}{3}x + (\frac{4}{3}x)^2 / 2! + (4/3\,x)^3 /3! + \ldots]\, x^3 dx$$

$$I_0 = e^{-4/3} \left[\frac{x^4}{4} + \frac{4}{3}\frac{x^5}{5} + (\frac{4}{3})^2(\frac{1}{2!})\frac{x^6}{6} + (\frac{4}{3})^3(\frac{1}{3!})\frac{x^7}{7} + \ldots\right]_0^1$$

$$I_0 = e^{-4/3} \left[\frac{1}{4} + \frac{4}{3}\frac{1}{5} + (\frac{4}{3})^2(\frac{1}{12}) + (\frac{4}{3})^3\frac{1}{4^2} + (\frac{4}{3})^4(\frac{1}{4!})(\frac{1}{8}) + \ldots\right]$$

$$I_0 = .1957351919$$

(البته می‌توانیم انتگرال را به روش‌های مختلف عددی که در فصل‌های بعدی ذکر می‌شود، محاسبه کنیم.)

برای محاسبه I_r عملیات را با I_0 شروع می‌کنیم:

$$\bar{I}_0 = fl(I_0) = I_0 + \varepsilon_0$$

ε_0 خطای گرد کردن

$fl(I_0)$ بصورت ممیز سیّار

$$\bar{I}_1 = fl(\frac{3}{4} - \frac{3}{4}(4)\quad \bar{I}_0) = \frac{3}{4} - 3\ \bar{I}_0 + \varepsilon'_1$$

که از ε'_1 خطای گرد کردن عبارت صرفنظر می‌کنیم در اینصورت:

$$\bar{I}_1 = \frac{3}{4} - 3\,(I_0 + \varepsilon_0)$$

$$\bar{I}_1 = \frac{3}{4} - 3I_0 - 3\,\varepsilon_0$$

$$\bar{I}_1 = I_1 + \varepsilon_1$$

$$\bar{I}_2 = fl\,(\frac{3}{4} - \frac{3}{4} \times 5 \times \bar{I}_1) = \frac{3}{4} - \frac{3}{4} \times 5 \times \bar{I}_1 + \varepsilon'_2$$

از ε'_2 صرفنظر می‌کنیم.

$$\bar{I}_2 = \frac{3}{4} - \frac{3}{4} \times 5(I_1 + \varepsilon_1) = I_2 + \frac{45}{4}\varepsilon_0$$

اگر بهمین نحو عملیات را ادامه دهیم خواهیم داشت:

$$\bar{I}_r = fl\,[\frac{3}{4} - \frac{3}{4}(r+3)\quad \bar{I}_{r-1}]$$

اگر از ε'_r خطای گردکردن این مرحله صرفنظر کنیم:

$$\boxed{\bar{I}_r = \frac{3}{4} - \frac{3}{4}(r+3)\quad \bar{I}_{r-1}}$$

بطوری که

$$I_r = \bar{I}_r + \varepsilon_r$$

$$I_{r-1} = \bar{I}_{r-1} + \varepsilon_{r-1}$$

با جایگذاری در رابطه اصلی داریم: $\bar{I}_r + \varepsilon_r = \frac{3}{4} - \frac{3}{4}(r+3)\ (\bar{I}_{r-1} + \varepsilon_{r-1})$

پس از ساده کردن: $\bar{I}_r + \varepsilon_r = \frac{3}{4} - \frac{3}{4}(r+3)\ \bar{I}_{r-1} - \frac{3}{4}(r+3)\ \varepsilon_{r-1}$

$$\varepsilon_r = -\frac{3}{4}(r+3)\ \varepsilon_{r-1}$$

از این رابطه بازاء $r=1,2,...,15$ خواهیم داشت:

$$\varepsilon_1 = -\frac{3}{4}(1+3)\ \varepsilon_0$$

$$\varepsilon_2 = -\frac{3}{4}(2+3)\ \varepsilon_1$$

$$\vdots$$

$$\varepsilon_{15} = -\frac{3}{4}(15+3)\ \varepsilon_{14}$$

پس از ضرب طرفین تساوی ε_r ,..., ε_2, ε_1, ε_0 و ساده کردن داریم:

$$\varepsilon_{15} = -(\frac{3}{4})^{15} \times \frac{18!}{6}\varepsilon_0$$

$$\bar{I}_{15} = I_{15} + (-\frac{3}{4})^{15} \times \frac{18!}{6}\varepsilon_0$$

که $\bar{I}_{15}$ بر حسب خطای اولیه ε_0 تعیین گردیده است و از خطاهای گردکردن بعدی نیز صرفنظر شده است.

$$\bar{I}_{15} = I_{15} + (-.143 \times 10^{14})\ \varepsilon_0$$

$$Max\ |\varepsilon_0| = .5 \times 10^{-10}$$

$$|\bar{I}_{15} - I_{15}| = |-.143 \times 10^{14} \times .5 \times 10^{-10}|$$

$$|\bar{I}_{15} - I_{15}| = .715 \times 10^{3}$$

همانطور که ملاحظه شد در محاسبات این مسأله خطای حاصل در مرحله‌ای، انتشار یافته و در مراحل بعدی رشدی بیش از اندازه معین نموده است، چنین محاسبه عددی را «ناپایدار» می‌نامیم. در مثال فوق چنانچه ابتدا I_{15} را محاسبه می‌نمودیم

با استفاده از این رابطه بازگشتی $I_{r-1} = \frac{1}{(r+3)} - \frac{4}{3(r+3)} I_r$

می‌توانستیم به ازای $r = 15,14,...,1$ به نتایج مورد نظر دست یابیم، محاسبات مسأله پایدار است، زیرا که در هر مرحله، خطای گردکردن اولیه بر عددی بزرگتر از یک $\frac{3}{4}(r+3)$ تقسیم گشته و خطا در نتیجه نهایی به حداقل ممکن کاهش می‌یابد.

برای محاسبه $I_{15} = \int_0^1 e^{4/3(x-1)}\ x^{18}\ dx$ می‌توانیم از بسط سری تابع $e^{4/3(x)}$ یا از ماشین حساب استفاده کنیم که $I_{15}=0.0493332$ بدست می‌آید و مسلماً با استفاده از این روش نیازی به محاسبه مستقیم انتگرال‌های بعدی نیست و رابطه فوق بدین منظور ارایه شده است.

۶.۱ موضوعات تئوری

هر کاربر برنامه ریاضی باید در ارتباط با تئوری‌های پشتیبان آن باشد. زیرا که تئوریها محدودیت برنامه و شرایطی که برای برنامه باید درست باشد تا جواب‌های قابل قبول بدست آید را مورد بحث قرار می‌دهند. در این کتاب، ضمن آنکه تئوری در داخل هر فصل بحث و بررسی می‌گردد، یک شرح کامل از تئوری آن موضوع نیز در پایان فصل خواهد آمد.

در این فصل، فقط یک روش شرح داده شده است - روش نصف کردن - بنابراین، این قسمت مختصر است. بهرحال، وقتی یک روش عددی بکار می‌رود معمولاً انواع پرسشها درباره تئوری مطرح می‌گردد. به علاوه، در این قسمت بعضی نکات که چطور تئوری ارائه خواهد شد، بحث می‌گردد.

اکثر روش‌های آنالیز عددی تکراری هستند و جواب‌های تقریبی با یک دنباله از تخمین‌ها بهبود می‌یابند.

چهار پرسش وجود دارند که ما همیشه درباره یک روش تکراری سؤال می‌کنیم:

۱ - تحت چه شرایطی روش بکار می‌رود؟ برای چه نوع توابعی روش بکار می‌رود، و چطور می‌دانیم که شرایط صادق هستند؟

۲ - آیا روش همگراست؟ آیا تقریبات متوالی به جواب صحیح با دقت داده شده می‌رسند؟

۳ - خطای هر تخمینی در چه دامنه‌هائی می‌تواند قرار گیرد؟

آیا می‌توانیم از پیش ماکزیمم خطا را بعد از تعداد معینی تکرار بدانیم؟

۴ - با چه سرعتی خطاهای تخمین‌های متوالی کاهش می‌یابند؟ بعنوان مثال، آیا خطاها متناسب با تعداد تکرارها کاهش می‌یابند یا دقت سریع‌تر از خطی است که در این حالت جواب بهبود می‌یابد.

۷.۱ پردازش موازی

یک کامپیوتر معمولاً دستورات را بطور متوالی اجراء می‌کند - یکی بعد از دیگری - اما روش دیگری در حال پدیدار شدن است.

در سراسر تاریخ توسعه کامپیوترها، ماشین‌های سریع و سریعتری ساخته شده‌اند، اما امروزه به حدّ بهبود سرعت رسیده‌ایم. سریعترین ماشینها اکنون می‌توانند با ساعت زمان تقریباً ۱ تا ۳ *nanoseconds* عمل کنند و 10^9 مرتبه عمل ممیز سیّار در هر ثانیه (*"flops"*) را انجام دهند.

ماشینهائی با چنین کیفیت بالایی امروزه سوپر کامپیوترهای خیلی گران قیمت هستند و

دارای سرعت بسیار بالایی می‌باشند و فکر با این سرعتها به وحشت می‌افتد، اما برای مسایل با اندازه بزرگ و واقعی همانند پیش بینی هوا، شبیه سازی برای پیش‌بینی کیفیت آیرودینامیکها، پردازش تصویری و هوش مصنوعی این ماشینها دارای سرعت کافی نیستند. بسیاری از این کاربردها درگیر حل روش‌های عددی مجموعه‌های بزرگ دستگاه معادلات مشابه هستند.

یکی از اولین فنون افزایش سرعت عملیات کامپیوتر «خط لوله‌ای» *(pipelining)* می‌باشد -که تشکیل دومین دستورالعمل در *CPU* می‌باشد، قبل از آنکه دستور العمل اول کامل شده باشد. چون مراحل خاص برای حل دستگاه معادلات درگیر ضرب‌های بسیاری از ضرب یک بردار در یک بردار دیگر می‌شود، این ماشینها، بهبود سرعت معینی را پیشنهاد می‌کنند، اما فقط با یک ضریب *5* یا *10*، برای این خصوصیت هزینه کامپیوترهای بزرگ بطور چشمگیری افزایش می‌یابد.

روش امروزه بکار بردن پردازش موازی است، به معنی اینکه، چندین ماشین روی تنها یک مسأله کار می‌کنند و تقسیم مراحل پردازش حل، به تعدادی مراحل که بطور همزمان می‌توانند تشکیل شوند.

همه مسایل اجازه چنین عملیات موازی را نمی‌دهند، اما تعداد مهمی از مسایل ریاضیات کاربردی می‌توانند این چنین انجام شوند. بدست آوردن تعداد زیادی یا حتی چند سوپرکامپیوتر، هزینه‌ای بسیار زیاد دارد. روش دیگر بکار بردن تعداد زیادی از میکروپراسسورهای کم قیمت است. (تعداد *1024* ریزپردازنده عدد کاربردی است).

اگر چه سرعت خاص میکروپراسسور برابر با یک سوپر کامپیوتر نیست، اختلاف سرعت توسط تعداد زیادی از ماشینها که ترکیب می‌شوند جبران می‌گردد.

تمرینات فصل اول

۱ - جواب مسائل زیر را بصورت ممیز سیّار با گردکردن و برش دادن بدست آورید.

a. $12.3+ 0.0234$ b. $12.3 + 0.0234$

c. $12.3 -0.0234$ d. $-321+ 32.1$

e. $132*0.987$ f. $-2.14/.000137$

g. $(-.111 + .222)* .00111/999$

۲ - خطای مطلق و خطای نسبی هر یک از نتایج مسأله (۱) را بدست آورید.

۳ - یک برنامه بنویسید و بطور تجربی سرعت‌های نسبی جمع، تفریق، ضرب و تقسیم را روی کامپیوتر در اختیار خودتان را معین کنید. تعداد تکرارها را به حد کافی زیاد بگیرید.

۴ - برای کامپیوتری که در دسترس دارید، بگویید وقتی نما سرریز یا ته ریز می‌شود ماشین چه انجام می‌دهد؟ آیا نتیجه به برنامه بستگی دارد؟

۵ - یک برنامه بنویسید که *eps* ماشین را تعیین کند. آیا این مقدار که بصورت تجربی تعیین شده است برابر است با آنچه شما از سیستم عددی ماشین انتظار داشته‌اید؟

۶ - مقادیر توابع زیر را به ازای مقادیر x داده شده محاسبه کنید و خطای مطلق و خطای نسبی نتایج رابدست آورید. داده‌های اولیه تا رقم‌های با معنی صحیح داده شده‌اند.

الف - $y = x^3 Sin(x)$ به ازای $x = \sqrt{2}$ ، $\sqrt{2} \approx 1.414$

ب - $y = x Ln(x)$ به ازای $x = \pi$ ، $\pi \approx 3.142$

پ - $y = e^x Cos(x)$ به ازای $x = \sqrt{3}$ ، $\sqrt{3} \approx 1.732$

ت - $y = a.Sin(b).Ln(c)$ به ازای $a = 21.00$ ، $b = .86$ ، $c = 5.6$

۷- مقادیر توابع زیر را به ازای مقادیر متغیرهای داده شده محاسبه کنید، و خطای مطلق و خطای نسبی نتایج را بدست آورید. داده‌های اولیه تا ارقام با معنی صحیح داده شده‌اند.

الف - $y = Ln(x_1+x^2_2)$ ، $x_1 = .97$ ، $x_2 = 1.132$

ب - $y = \frac{x_1+x^2_2}{x_3}$ ، $x_1 =3.28$ ، $x_2 =.932$ ، $x_3 =1.132$

پ - $y = x_1 x_2 +x_2 x_3 +x_3 x_1$ ، $x_1 =2.104$ ، $x_2=1.935$ ، $x_3 =.845$

۸- مطلوبست محاسبه $y= 1-Cos1^\circ$ در صورتیکه یک ماشین حساب با دقت چهار رقم اعشار داشته باشیم. خطای نتیجه را برآورد کنید. برای دقت بیشتر با همین ماشین حساب چه روشی

را پیشنهاد می‌کنید.

راهنمایی: برای دقت بیشتر از رابطه $cos(2\alpha)=1-sin^2(\alpha)$ استفاده کنید

۹- دستگاه معادلات زیر را با بکاربردن $d=2$ بصورت ممیز سیّار حل کنید و نتیجه را با جواب‌های دقیق مساله مقایسه نمائید.

$$\begin{cases} x + \frac{y}{2} = 2 \\ \frac{x}{2} + \frac{y}{3} = \frac{7}{6} \end{cases}$$

۱۰- مقدار دقیق $\frac{\pi}{4}$ برابر است با مجموع سری:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \ldots$$

الف - تعداد جملاتی را که باید در نظر گرفت تابا انتخاب دو رقم، بصورت ممیز سیّار، جملات بعدی تاثیری بر جواب نداشته باشند، را تعیین کنید.

ب - اگر عبارت زیر رابکار بریم، بطوریکه اولین جمله $(1 - \frac{1}{3})$ و دومین جمله $(\frac{1}{5} - \frac{1}{7})$ باشد، تعداد جملات را تعیین نمائید، تا نتیجه قسمت (الف) بدست آید.

$$\frac{\pi}{4} = (1-\frac{1}{3}) + (\frac{1}{5}-\frac{1}{7}) + (\frac{1}{9} - \frac{1}{11}) + \ldots$$

۱۱ - برنامه شکل ۵.۱ را با کامپیوتر اجراء کنید. آیا جوابها همانند شکل ۶.۱ می‌باشند؟ اگر یکسان نیستند، اختلاف را شرح دهید.

۱۲ - برنامه شکل ۵.۱ را به زبان دیگر ترجمه کنید و برنامه را اجراء کنید. اگر جوابها همانند ۶.۱ نیستند اختلاف را شرح دهید.

۱۳ - برنامه شکل ۵.۱ را اصلاح کرده و آزمایش تعداد ماکزیمم تکرار را حذف کنید. با چه سطح خطائی برنامه حلقه را پایان می‌دهد؟ چطور برنامه را پایان می‌دهید وقتی این تغییر در حلقه رخ می‌دهد؟

۱۴ - برنامه شکل ۵.۱ را اصلاح کنید و ریشه‌های معادلات زیر را بدست آورید.

a) $e^x - 3x^2 = 0$ در فاصله $[-2\ ,\ 0]$

b) $x^3 - 2x^2 + 1.1 = 0$ در فاصله $[0.1\ ,\ 5]$

c) $x - cos[x/(x-2)] = 1$ در فاصله $[-3\ ,\ 1]$

جواب: a) $-.458962$

۱۵ - با روش برش بعد از سه رقم و با روش گردکردن تا سه رقم چند جمله‌ای $2.75x^3 - 2.95x^2 + 3.16x - 4.67$ را برای $x = 1.07$ از چپ به راست و جمله به جمله حساب کنید. خطای نسبی را پیدا کنید.

جواب: مقدار تحقیقی برابر است با -1.297387

با گردکردن نتیجه برابر است با -1.29

و خطای مطلق برابر 0.00739 و خطای نسبی برابر 0.00569 می‌باشد.

۱۶ – تعیین مقدار چند جمله‌ای بصورت آشیانه‌ای بسیار مناسب‌تر است. صورت آشیانه‌ای تمرین ۱۵ برابر است با:

$$((2.75x - 2.95)x + 3.16)x - 4.67$$

نتیجه تمرین ۱۵ را از این رابطه بدست آورید.

جواب: با گردکردن تا سه رقم در هر مرحله نتیجه برابر است با -1.3

۱۷ – تمرین شماره ۵ برای تعیین اپسیلن *eps* ماشین را برای دقت مضاعف *double* و دقت توسعه یافته *extended* انجام دهید.

۱۸ – در روش نصف کردن خطا بعد از n تکرار بصورت زیر است. خطا با افزایش n بصورت یکنواخت کاهش می‌یابد. نشان دهید که این همیشه صحیح نیست. $\text{خطا} = (\frac{1}{2^n}) \mid b-a \mid$

۱۹ – چطور می‌توان تشخیص داد یک پردازش موازی می‌تواند اجرای عددی تکراری را سرعت بخشد.

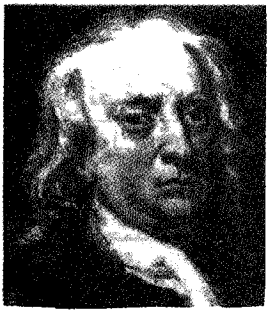
۲۰ – انتگرال معین $I_n = \int_0^1 x^n e^{x-1} dx$

را باین ترتیب بررسی نمائید که رابطه ای بازگشتی بین I_n, I_{n-1} را بدست آورید و سپس با محاسبه I_0 مقادیر I_1,... I_{20} را محاسبه کنید.

بار دیگر با محاسبه I_{20} مقادیر I_{19}, I_{18},...,I_0 را بدست آورید و جوابها را مقایسه کرده و توضیح دهید. به این منظور برنامه‌ای به یک زبان دلخواه بنویسید و آنرا اجرا نمائید.

۲۱ – انتگرال معین $I_n = \int_0^1 x^n \, Sin \, \pi x \, dx$

را مطابق برنامهٔ قبلی برای I_0 تا I_{20} محاسبه کنید و پایداری روشی را که بکار می‌برید، توضیح دهید و نیز پس از محاسبه I_{20} مقادیر I_{18} و I_{16} و... I_0 را بدست آوردید.



ایزاک نیوتن
NEWTON
(۱۷۲۷ - ۱۶۴۲)

نیوتن در سال ۱۶۴۲ در روستای وولز تورپ در انگلستان به دنیا آمد. نیوتن آزمایشگری ماهر و تحلیل گری عالی بود، به طور قطع همه او را در زمرهٔ بزرگترین ریاضیدانانی می‌دانند که تا کنون جهان به خود دیده است. این سخن از لاگرانژ است که نیوتن بزرگترین نابغه‌ای بوده که در جهان زیسته است و نیز خوشبخت ترین آنان، زیرا تنها یک بار می‌توان دستگاه جهان را تأسیس کرد. مهم‌ترین اثر نیوتن کتاب اصول ریاضی فلسفه طبیعی اوست (۱۶۸۷) که در آن برای اولین بار یک دستگاه دینامیکی کامل و یک صورتبندی ریاضی از پدیده‌های اصلی زمینی و سماوی حرکت ظاهر می‌شود. و از سایر آثار او مانند نور شناسی و منحنی‌های درجه سوم (۱۷۰۴) و روش فلوکسیونها و سری‌های نامتناهی و دروس حساب عمومی می‌توان نامبرد. نیوتن روش فلوکسیون‌های خود را که امروزه تحت عنوان حساب دیفرانسیل شناخته می‌شود در موارد متعدد و به طرز جالبی بکار برد، ماکزیمم و مینیمم، مماس بر منحنی‌ها، زنجیرهای منحنی‌ها، نقاط عطف، تحدب و تعقر منحنی‌ها را تعین کرد و در انتگرال گیری برخی معادلات دیفرانسیل توانایی فوق العاده از خود نشان داد. در ستایش او گفته‌اند: طبیعت و قوانین طبیعت در ظلمت نهفته بودند؛ ذات باری فرمود «نیوتن به وجود آید» و همه چیز روشن شد.

فصل ۲

حل معادلات غیر خطی

موضوعات این فصل

* روش نصف کردن فاصله‌ها
* روش درون یابی خطی
* استفاده از روش $x = g(x)$
* روش نیوتن
* همگرایی روش نیوتن
* روشهایی برای چند جمله‌ایها (روش هورنر)
* روش مولر
* روش بیرستو
* روش‌های دیگر برای چند جمله‌ایها روش گرافی Graeffe
* برنامه نویسی برای جواب‌های عددی
* برنامه مولر به زبان C بنام PMULLER.C
* برنامه بیرستو به زبان پاسکال بنام BRSTOW.PAS
* برنامه نیوتن به زبان C بنام PNEWTHR.C
* برنامه تکرار تابعی به زبان پاسکال بنام PXGIT.PAS
* تمرینات فصل دوم

۱.۲ روش نصف کردن فاصله‌ها

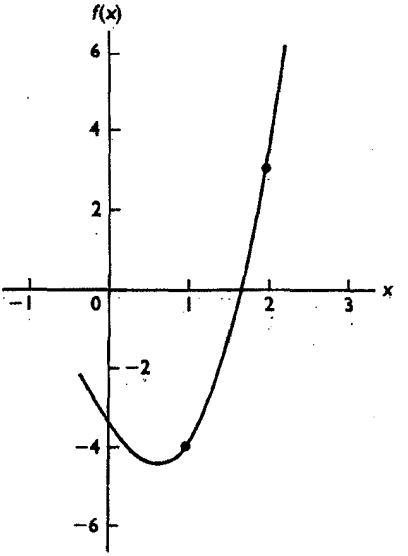
اولین روش عددی که مطالعه می‌کنیم روش نصف کردن فاصله است[۱]،

$$f(x) = x^3 + x^2 - 3x - 3 = 0$$

مثال ۱.۲ چند جمله‌ای درجه سوم زیر را بررسی کنید.

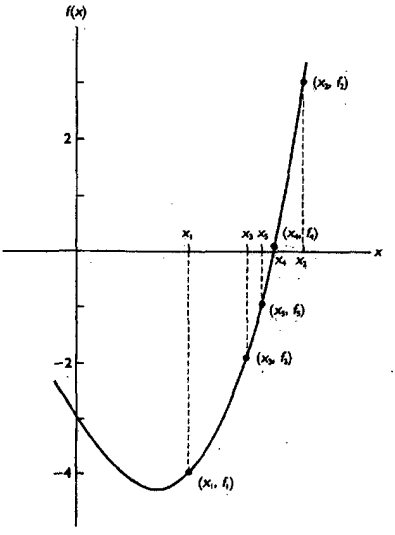
مقدار f در $x = 1$، برابر -4 است و در $x = 2.2$ مقدار آن $+3$ می‌باشد. چون تابع پیوسته می‌باشد، طبق قضیه بولزانو تغییر علامت تابع بین $x=1$ و $x=2$ حداقل یک ریشه روی فاصله $(1,2)$ را نشان می‌دهد. شکل ۱.۲ را ببینید.

فرض کنید ما مقدار تابع را در $x=1.5$ تعیین کنیم و نتیجه را با مقادیر تابع در $x=1$ و $x=2$ مقایسه کنیم. چون تابع بین $x=1.5$ و $x=2$ تغییر علامت می‌دهد، یک ریشه بین این مقادیر واقع است. ما می‌توانیم روش نصف کردن فاصله را برای تعیین یک فاصله کوچکتر که در آن باید یک ریشه واقع باشد، ادامه دهیم. برای این مثال، ادامه عمل در آخر منتهی به یک تقریب برای ریشه در $x=\sqrt{3}=1.7320508075$ می‌شود. جریان عمل در شکل ۲.۲ نشان داده شده است.



شکل ۱.۲

۱ - این روش بعنوان بولزانو شناخته شده است.



شکل ۲.۲

اگر ما فقط یک جواب تقریبی را بخواهیم، یک روش ترسیمی در شکل ۲.۲ نشان داده شده است که ممکن است مناسب باشد. برای داشتن دقت بیشتر لازم است یک قاعدهٔ ریاضی برای انجام آن داشته باشیم. همچنین ما باید آلگوریتم مورد نظر را بیان کنیم تا اجرای روش را با کامپیوتر آسان نماید. ما سبکی از بیان آلگوریتم را که روی ساختار مرتبی تأکید می‌کند، آغاز می‌کنیم.

روش نصف کردن فاصله

برای تعیین ریشه $f(x) = 0$، با دقت معین ابتدا مقادیر x_1 و x_2 را مفروض می‌گیریم بطوریکه $f(x_1)$ و $f(x_2)$ دارای علامت مخالف باشند. سپس:

```
REPEAT
    set x3 = (x1+x2)/2.
    IF f(x3)*f(x1) < 0:
        SET x2 = x3.
            ELSE Set x1 = x3.
            ENDIF
```

مقدار نهایی x_3 ریشه تقریبی است، این خطا بیشتر از $\frac{1}{2} |x_2 - x_1|$ نیست.

توجه شود اگر $f(x)$ در فاصله $[x_2, x_1]$ ناپیوسته باشد، این روش ممکن است یک ریشه نادرست بدهد.

با بکار بردن این روش برای $f(x) = x^3 + x^2 - 3x - 3 = 0$، نتایج جدول ۱.۲ بدست می‌آید. انجام مکرر این آلگوریتم و تقریب‌های متوالی، یک روش «تکراری» نامیده می‌شود. جدول ۱.۲ ضرورت نمایش مقادیر تقریبی متغیر x و تابع $f(x)$ را نشان می‌دهند بطوریکه عملیات ممیز سیّار در کامپیوترهای عددی فقط تعداد محدودی از ارقام اعشاری را مورد استفاده قرار می‌دهد. بی‌دقتی مشابهی در کار ما وجود دارد، زیرا که کامپیوترها تنها تعداد محدودی از ارقام با معنی را نگهداری می‌کنند. توجه شود که نه فقط در روش‌های عددی بلکه در کلیه محاسبات این چنین است. ما به چنین «خطای گرد کردن» توجه خواهیم کرد. فرق بین روش‌های عددی و آنالیز عددی این است که در آنالیز عددی بررسی خطاهای روش بکار رفته ضروری است.

مسلماً استفاده چشم بسته از هر روش بدون نگرانی در مورد دقت آن، غیر منطقی است. اینکه عددی را گرد می‌کنیم یا ارقام اضافی را حذف می‌کنیم مسلماً ایجاد خطا خواهند کرد. در جدول ۱.۲ ارقام بعد از پنج رقم قطع شده‌اند و نتیجه محاسبه حداکثر تا پنج رقم با معنی صحیح خواهد بود.

علاوه بر محدودیت در دقت، چون ما فقط تعداد محدودی از ارقام را در کار خود نگهداری می‌کنیم، اگر عمل را خیلی زود خاتمه دهیم این محدودیت، عدم دقت نتیجه را نشان می‌دهد، یک امتیاز مهم روش نصف کردن فاصله، غیر از سادگی آن، آگاهی ما از دقت تقریب به ریشه در هر مرحله است.

جدول ۱.۲ روش نصف کردن فاصله برای $f(x) = x^3 + x^2 - 3x - 3 = 0$

تعدادتکرارها	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	حداکثرخطادر x_3
1	1	2	1.5	−4.0	3.0	−1.875	0.5
2	1.5	2	1.75	−1.875	3.0	0.17187	0.25
3	1.5	1.75	1.625	−1.875	0.17187	−0.94335	0.125
4	1.625	1.75	1.6875	−0.94335	0.17187	−0.40942	0.0625
5	1.6875	1.75	1.71875	−0.40942	0.17187	−0.12478	0.03125
6	1.71875	1.75	1.73437	−0.12478	0.17187	0.02198	0.015625*
7	1.71875	1.73437	1.72656				0.0078125
	⋮	⋮	⋮			⋮	
∞			1.73205			−0.00000 ...	

خطای واقعی در x_3 بعد از پنج تکرار برابر با -0.01330 می‌باشد

چون ریشه باید بین دو مقدار متفاوت x باشد بطوری که تابع بازاء این دو مقدار تغییر علامت می‌دهد،[1] خطا در آخرین تقریب نمی‌تواند بیش از نصف آخرین فاصله‌ای باشد که این تقریب نقطه میانی آن فاصله است. این فاصله دقیقاً شناخته شده است زیرا که اختلاف اولیه، $|x_2 - x_1|$ در هر مرحله نصف شده است. در سایر روشها، تعیین دقت به این سادگی نخواهد بود.

۱- توجه کنید که اگر تابع ناپیوسته باشد. $f(x)$ ممکن است بدون داشتن ریشه‌ای در فاصله تغییر علامت بدهد. برای محاسبه ریشه هر تابع ناشناخته باید پیوستگی آن معین شود.

جدول ۲.۲ روش نصف کردن فاصله برای $f(x) = e^x - 3x = 0$

تعدادتکرارها	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	حداکثرخطادر x_3
1	1.0	2.0	1.5	−0.28172	1.38906	−0.01831	0.5
2	1.5	2.0	1.75	−0.01831	1.38906	0.50460	0.25
3	1.5	1.75	1.625	−0.01831	0.50460	0.20342	0.125
4	1.5	1.625	1.5625	−0.01831	0.20342	0.08323	0.0625
5	1.5	1.5625	1.53125	−0.01831	0.08323	0.03020	0.03125
6	1.5	1.53125	1.51562	−0.01831	0.03020	0.00539	0.015625*
	⋮	⋮	⋮				
∞			1.51213				

خطای واقعی در x_3 بعد از پنج تکرار برابر با -0.01912 می‌باشد

دقت یک مقدار محاسبه شده معمولاً با خطای مطلق (تفاضل مقدار تحقیقی از مقدار تقریبی) یا خطای نسبی (خارج قسمت خطای مطلق بر مقدار تحقیقی) بیان می‌شود. خطای نسبی اغلب مقیاس بهتری از دقت برای مقادیر خیلی کوچک یا خیلی بزرگ می‌باشد.

گاهی اوقات دقت بر حسب تعداد ارقام صحیح بیان می‌شود، و در حالت‌های دیگر برحسب تعداد ارقام صحیح بعد از نقطه اعشاری بیان می‌شود. وقتی مقدار تحقیقی شناخته شده نیست، غیر ممکن است دقت را بطور دقیق بیان کنیم، و دقت تقریبی باید مشخص شود.

روش نصف کردن فاصله برای معادلات غیر جبری بهمان اندازه سایر روش‌های این فصل مناسب است. وقتی این روش را برای $f(x) = e^x - 3x = 0$ بکار می‌بریم، دارای ریشه بین $x = 1$ و $x = 2$ می‌باشد، جدول ۲.۲ نتایج حاصل را نشان می‌دهد.

در روش نصف کردن فاصله، قبل از آن که روش را شروع کنیم، لازم است که مقادیر اولیه بدست آمده باشند. این برای پیداکردن ریشه در اکثر روشها صادق است، این مقادیر شروع می‌توانند با رسم نه چندان دقیق یا محاسبات آزمایشی، یا با نوشتن یک برنامه در یک کامپیوتر یا ماشین حساب قابل برنامه ریزی بدست آیند. شاید بهترین روش از طریق برنامه‌های مخصوص جهت رسم توابع می‌باشد تا کامپیوتر خمها را در امتدادی که می‌خواهیم رسم کند و پارامترها را تغییر می‌دهیم تا مقادیر تقریبی ریشه‌ها بدست آیند.

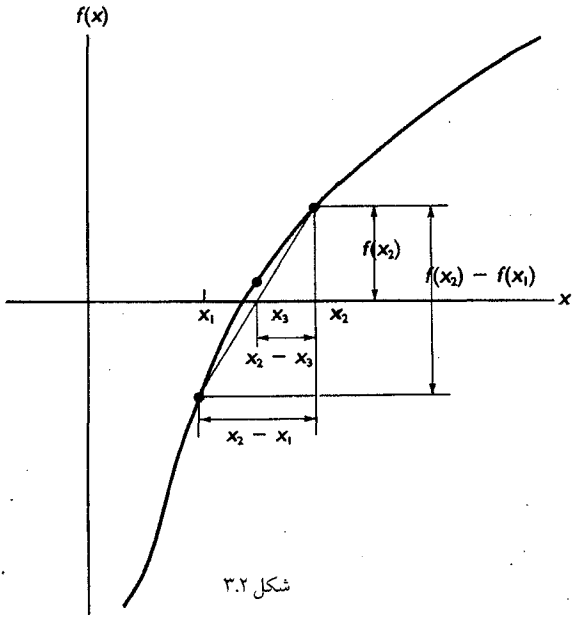
۲.۲ روش درون یابی خطی

هرچند روش نصف کردن فاصله آسان و تجزیه و تحلیل خطای آن ساده است. لیکن روش بسیار کارآمدی نمی‌باشد. برای اغلب توابع می‌توانیم سرعت همگرایی به ریشه را بهبود بخشیم. روش درون یابی یک چنین روشی است.

فرض کنیم تابع روی فاصله (x_1, x_2) بررسی می‌شود، که در آن $f(x_1)$ و $f(x_2)$ دارای علامت مخالف باشند از تشابه مثلثها در شکل ۳.۲ می‌توانیم بنویسیم:

$$\frac{x_2 - x_3}{x_2 - x_1} = \frac{f(x_2)}{f(x_2) - f(x_1)},$$

$$x_3 = x_2 - \frac{f(x_2)}{f(x_2) - f(x_1)}(x_2 - x_1).$$



شکل ۳.۲

سپس $f(x_3)$را محاسبه می‌کنیم و دوباره با درون یابی خطی، بین مقادیری که تابع تغییر علامت می‌دهد، مقدار جدید x_3 بدست می‌آید. تکرار این عمل تخمین‌های بهبود یافته ریشه را خواهد داد.

مثال ۲.۲ این روش را برای چند جمله‌ای درجه سوم در مثال ۱.۲ به کار برید.

جدول ۳.۲ نتایج این روش را برای همان چند جمله‌ای که شرح داده شد، نشان می‌دهد. بنظر می‌رسد این روش قدری از روش نصف کردن فاصله سریعتر باشد، بعد از سه مرحله نتیجه‌ای تقریباً با همان دقت بدست می‌آیند که در روش قبلی بعد از هفت مرحله بدست آمد. بطور ذاتی مشهود است سرعتی که تقریبات متوالی بسمت صفر تابع (ریشه $f(x)=0$) میل می‌کند به درجه‌ای که تابع از یک خط مستقیم در فاصله مورد بررسی دور می‌شود بستگی دارد. به بیان دیگر درجه همگرایی بستگی به سرعت تغییر ضریب زاویه خم دارد که با بزرگی قدر مطلق مشتق مرتبه دوم اندازه گیری می‌شود، یک بیان آلگوریتم این روش در زیر نشان داده شده است.

روش درون یابی خطی

تعیین یک ریشه $f(x) = 0$ وقتیکه مقادیر x_1 و x_2 طوری داده شده‌اند که $f(x_1)$ و $f(x_2)$ دارای علامت مخالف باشند

REPEAT

$$set\ x_3 = x_2 - f(x_2)\ \frac{x_2 - x_1}{f(x_2) - f(x_1)}.$$

IF $f(x_3)$ مختلف العلامه $f(x_1)$

SET $x_2 = x_3$.

ELSE Set $x_1 = x_3$.

ENDIF

UNTIL $|x_1 - x_2| <$ (خطای قابل اغماض) یا $|f(x_3)| < \varepsilon$

(ε مقدار قابل اغماض است)

توجه: این روش ممکن است یک ریشه نادرست بدهد اگر $f(x)$ روی فاصله $[x_1$ و $x_2]$ پیوسته نباشد.

جدول ۳.۲ یک عیب جدی روش درون یابی را آشکار می‌کند: نزدیک شدن به جواب یکطرفه است همچنانکه در شکل ۴.۲ نشان داده شده است. اگر $f(x)$ دارای انحنای بزرگی بین x_1 و x_2 باشد، این امر باعث می‌شود که با سرعت کمی به جواب نزدیک شویم، یک علاج این روش درون یابی اصلاح شده است. مقدار $f(x)$ در نقطه انتهایی که حرکت نمی‌کند را با $f(x)/2$ جایگزین می‌کنیم.[1] این روش همچنانکه شکل ۵.۲ نشان می‌دهد به‌حل مشکلات کمک می‌کند.

جدول ۳.۲ روش درون‌یابی خطی برای $f(x)=x^3+x^2-3x-3$

تعداد تکرارها	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$
1	1.0	2.0	1.57142	−4.0	3.0	−1.36449
2	1.57142	2.0	1.70540	−1.36449	3.0	−0.24784
3	1.70540	2.0	1.72788	−0.24784	3.0	−0.03936
4	1.72788	2.0	1.73140	−0.03936	3.0	−0.00615
5	1.73140	2.0	1.73194*			

۱- نصف کردن عرض در انتهای دیگر فاصله را، موقعی که در یک طرف ریشه قرار می‌گیریم، متوقف می‌کنیم، همچنانکه در سومین تکرار در شکل ۵.۲ رخ می‌دهد.

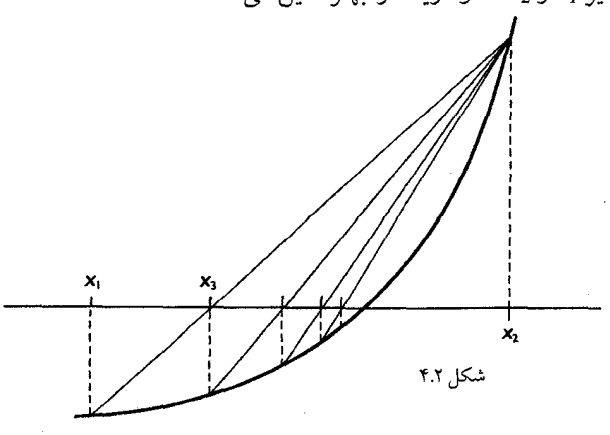
یک آلگوریتم برای اصلاح روش درون یابی خطی در زیر نشان داده شده است.

روش درون یابی اصلاح شده

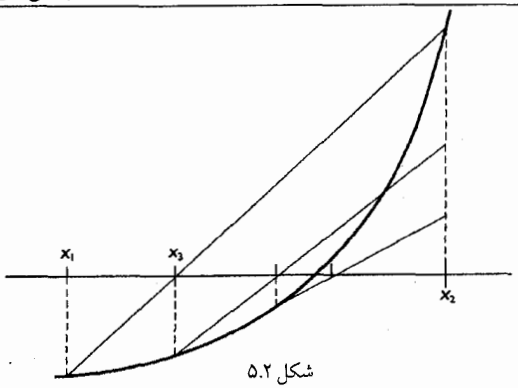
تعیین یک ریشه $f(x) = 0$ وقتیکه مقادیر x_1 و x_2 طوری داده شده‌اند که $f(x_1)$ و $f(x_2)$ دارای علامت مخالف باشند

Set SAVE $= f(x_1)$; *set F1* $= f(x_1)$; *set F2* $= f(x_2)$
REPEAT
  Set $x_3 = x_2 - F2\frac{x_2 - x_1}{F2 - F1}$.
  IF $f(X_3)$ مختلف العلامه با *F1*
    Set $x_2 = x_3$
      Set $F2 = f(x_3)$
    IF $f(X_3)$ هم علامت *SAVE:*
      Set $F1 = F1/2$.
    ENDIF.
ELSE Set $x_1 = x_3$,
    Set $F_1 = f(x_3)$
    IF $f(X_3)$ هم علامت *SAVE:*
      Set $F2 = F2/2$,
    ENDIF
  ENDIF
Set SAVE $= f(x_3)$
UNTIL $|f(x_3)| < \varepsilon$ (خطای قابل اغماض ε)

بهر حال تغییر مقادیر x_1 و x_2 حدود ریشه را بهتر معین می‌کند.



شکل ۴.۲



شکل ۵.۲

روش دیگری وجود دارد که می‌توانیم روش درون یابی خطی را بهبود بخشیم. بجای آنکه لازم باشد برای دو مقدار بکار رفته در درون یابی، تابع دارای علائم مختلف باشد، می‌توانیم دو مقدار مختلف که به ریشه نزدیکتر هستند را انتخاب کنیم و از این نقاط درون یابی یا برون یابی می‌نمائیم. معمولاً نزدیکترین مقادیر به ریشه دو مقدار آخری هستند که محاسبه شده‌اند. این روش فاصله تحت بررسی جهت بدست آوردن ریشه را کوتاه‌تر خواهد ساخت و از اینرو این روش بهبود می‌یابد و تابع می‌تواند با خطی که از این دو نقطه می‌گذرد نمایش داده شود.

جدول ۵.۲ محاسباتی را که بر طبق این روش بنام روش وتر مشهور است نشان می‌دهد این مثال سرعت همگرایی بیشتری را نشان می‌دهد. x_6 در این روش دارای دقت بیشتری از روش درون یابی خطی می‌باشد.

توسعه و تبیین آلگوریتم روش وتر را بعنوان تمرین برای دانشجویان باقی می‌گذاریم.

جدول ۴.۲ درون یابی خطی اصلاح شده برای $f(x) = x^3 + x^2 - 3x - 3 = 0$

تعدادتکرارها	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	$SAVE$
1	1.0	2.0	1.57142	−4.0	3.0	−1.36449	−4.0
2	1.57142	2.0	1.77557	−1.36449	1.5*	0.42369	−1.36449
3	1.57142	1.77557	1.72720	−1.36449	0.42369	−0.04576	0.42369
4	1.72720	1.77557	1.73191	−0.04576	0.42369	-1.332×10^{-3}	−0.04576
5	1.73191	1.77557	1.732183†	-1.332×10^{-3}	0.21184*		

جدول ۴.۲ روش اصلاح شده را که برای مسأله چند جمله‌ای مثال قبلی بکار رفته است، نشان می‌دهد. در حالیکه روش اصلاح شده معمولاً سریعتر همگراست

جدول ۵.۲ روش وتر برای $f(x) = x^3 + x^2 - 3x - 3 = 0$

تعدادتکرارها	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$
1	1.0	2.0	1.57142	−4.0	3.0	−1.36449
2	2.0	1.57142	1.70540	3.0	−1.36449	−0.24784
3	1.57142	1.70540	1.73513	−1.36449	−0.24784	0.02920
4	1.70540	1.73513	1.73199	−0.24784	0.02920	−0.0005755
5	1.73513	1.73199	1.73205*			

خطای واقعی در x_3 بعد از پنج تکرار برابر با -0.00011 می‌باشد

البته باید توجه شود موقعی که اطلاع کافی نداریم که ریشه حقیقی در مجاورت دو نقطه که مقادیر تابع در آنها هم علامت هستند، وجود دارد، نباید برای آن ریشه برون یابی کنیم. شکل ۶.۲ بیهودگی جستجو برای یک ریشه را که وجود ندارد، نشان می‌دهد. به علاوه، ملاحظه خواهد شد که در روش وتر وقتی $f(x_1) = f(x_2)$ بشود تقسیم بر صفر بوجود می‌آید. همچنین ممکن است منحرف شده و ریشه دیگری بجز آنچه انتظار داشته‌ایم بدست آوریم.

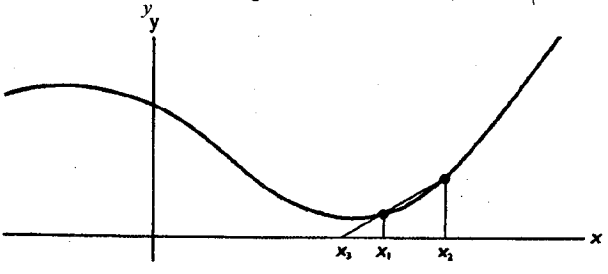
البته روشهایی که بر مبنای درون یابی خطی هستند محدود به چند جمله‌ایها نیستند.

مثال ۳.۲ با هریک از روشهایی که تاکنون بحث شد ریشه معادله زیر را به دست آورده و آنها را مقایسه کنید.

$$3x + sinx - e^x = 0$$

در جدول ۶.۲ نتایج آورده شده است، البته مقدار x برحسب رادیان در جمله مثلثاتی بکار برده می‌شود. هر روش با $x_2 = 0$ و $x_1 = 1$ شروع می‌شود.

توجه کنید که در کلیه روشهایی که بکار برده شده‌اند تخمین اولیه ریشه مورد محاسبه ضروری است. اغلب همان قدر فکر و کوشش برای بدست آوردن مقدار شروع مطلوب لازم است که برای تصحیح کردن آن با دقت قابل قبول لازم می‌باشد. گاهی اوقات با توجه به اطلاع از یک مسئله فیزیکی، یک مقدار شروع را پیشنهاد می‌کنیم، ولی اگر مقدور نباشد، معمولاً با محاسبات آزمون و خطا، یا با رسم نه چندان دقیق تابع، مقادیر اولیه را پیدا می‌کنیم. ما بعداً بعضی روشها را مورد بحث قرار خواهیم داد که برای چند جمله‌ایها خود آغاز هستند.



شکل ۶.۲

جدول ۶.۲ مقایسه روشها برای پیداکردن ریشه‌های معادله $f(x) = 3x + sin(x) - e^x = 0$

	تعدادتکرارها		نصف‌کردن		درون یابی		درون‌یابی‌اصلاح شده		روش‌وتر	
	x_3	$f(x_3)$	x_3	$f(x_3)$	x_3	$f(x_3)$	x_3	$f(x_3)$	x_3	$f(x_3)$
1	0.5	0.330704	0.470990	0.265160	0.470990	0.265160	0.470990	0.265160		
2	0.25	−0.286621	0.372277	0.029533	0.372277	0.029533	0.372277	0.029533		
3	0.375	0.036281	0.361598	2.94×10^{-3}	0.351514	−0.022356	0.359904	-1.29×10^{-2}		
4	0.3125	−0.121899	0.360538	2.90×10^{-4}	0.360727	7.64×10^{-4}	0.360424	5.53×10^{-6}		
5	0.34375	−0.041956	0.3604334	2.93×10^{-5}	0.3604224	1.80×10^{-6}	0.3604218	2.13×10^{-7}		
		0.01667		-1.17×10^{-5}		-7×10^{-7}		-1×10^{-7}		

۳.۲ روش تکرار تابعی، $x = g(x)$ (روش نقطه ثابت*)

اکنون روش دیگری را مورد بحث قرار می‌دهیم که قابلیت کاربرد عمومی دارد و به ما اجازه می‌دهد تئوری لازم را بیان کنیم. با معادله $f(x)=0$ شروع می‌کنیم، و آن را به صورت یک عبارت هم‌ارز $x=g(x)$ می‌نویسیم.

به طوری که اگر $f(r) = 0, r = g(r)$

تحت شرایط مناسب، که در زیر بیان می‌کنیم، الگوریتم

$$x_{n+1} = g(x_n), \qquad n = 1, 2, 3, \ldots$$

به یک صفر از $f(x)$ همگرا خواهد بود.

مثال ۲.۴: ریشه‌های معادله زیر را به روش تکرار تابعی بدست می‌آورید

$$f(x) = x^2 - 2x - 3 = 0$$

واضح است که ریشه‌ها $x=3$، $x=-1$ می‌باشند.

اگر با ترتیب دیگر بنویسیم، داریم: $x = \sqrt{2x+3}$

به طوری که $g(x) = \sqrt{2x+3}$ اگر با $x_1=4$ شروع کنیم، داریم.

$$x_2 = \sqrt{11} = 3.316$$
$$x_3 = \sqrt{9.632} = 3.104,$$
$$x_4 = \sqrt{9.208} = 3.034,$$
$$x_5 = \sqrt{9.068} = 3.011,$$
$$x_6 = \sqrt{9.022} = 3.004,$$

این فقط یکی از چند حالت ممکن برای نوشتن دوباره جملات معادله است. در نتیجه تکرارهای مختلف به ریشه $x_\infty = 3$ همگرا می‌شود.

معادله $f(x) = x^2 - 2x - 3 = 0$ را می‌توان به ترتیب دیگری نیز نوشت،

به عنوان مثال $x = \dfrac{3}{(x-2)}$ که ترتیب دیگری از نوشتن فرم $x=g(x)$ می‌باشد اگر $x_1=4$ آنگاه:

$$x_2 = 1.5,$$
$$x_3 = -6,$$
$$x_4 = -0.375,$$
$$x_5 = -0.263,$$
$$x_6 = -0.919,$$
$$x_7 = -1.028,$$
$$x_8 = -3.991,$$
$$x_9 = -1.003,$$

اما توجه شود این دنباله به ریشه $x = -1$ همگراست و تکرارها این بار بجای یکنواخت بودن بطور نوسانی همگرا می‌شوند.

سومین ترتیب نوشتن را در نظر می‌گیریم. از

* *fixed ponit mrthod*

$$x = \frac{x^2 - 3}{2}$$

برای $x_1 = 4$ بدست می‌آوریم

$$x_2 = 6.5,$$
$$x_3 = 19.635,$$
$$x_4 = 191.0$$

که به وضوح واگراست.

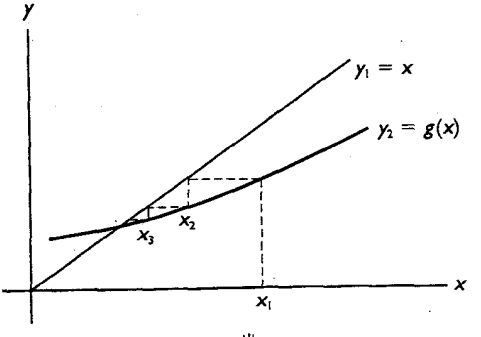
شکل ۷.۲ چند حالت مختلف را نشان می‌دهد.

الف: همگرایی یکنواخت را نشان می‌دهد.

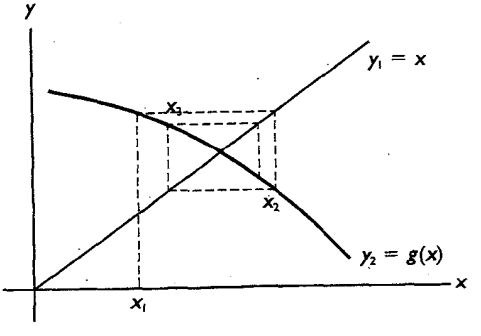
ب: همگرایی نوسانی را نشان می‌دهد.

ج: واگرایی را نشان می‌دهد.

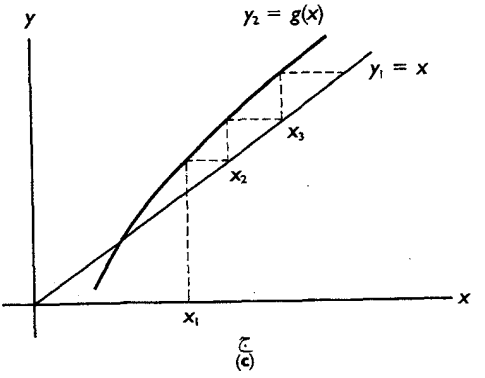
برای معادله $x = g(x)$، جواب، محل تقاطع خط $y_1 = x$ با خم $y_2 = g(x)$ می‌باشد. به طور عمودی به طرف خم و سپس به طور افقی به طرف خط حرکت می‌کنیم و این عمل را تکرار می‌کنیم، تا به جواب برسیم.



(الف) (a)



(ب) (b)



شکل ۷.۲

در زیر آلگوریتم این روش نشان داده شده است.

تکرار با فرم $x=g(x)$

برای تعیین یک ریشه از $f(x)=0$ با فرض یک مقدار x_1 که به ریشه نزدیک می‌باشد.

معادله را دوباره به صورت هم‌ارز دیگری به فرم $x=g(x)$ جایگذاری کنید.

$x=g(x)$

$set\ x_2 = x_1$

REPEAT

$set\ x_1 = x_2$

$x_2 = g(x_1)$

UNTIL $|x_2 - x_1| < \varepsilon$ (ε مقدار خطای قابل اغماض است)

توجه: این روش ممکن است به ریشه‌ای غیر از ریشه مورد انتظار همگرا شود، یا ممکن است واگرا گردد. ترتیب‌های متفاوت با تعداد تکرار مختلف همگرا خواهد شد.

اکنون شرایطی را مطالعه می‌کنیم که برای همگرایی لازم هستند. با

$$x_{n+1} = g(x_n)$$

تکرار را انجام می‌دهیم.

فرض کنیم $x=r$ جوابی برای $f(x)=0$ باشد، بنابراین $f(r)=0$ و $r=g(r)$ با کم کردن r و $g(r)$ از طرفین $x_{n+1} = g(x_n)$ و تقسیم کردن به $x_n - r$ اگر $g(r)$ و $g(x)$ در فاصله بین r و x پیوسته باشند، بنابر قضیه مقدار میانگین می‌توان نوشت:

$$x_{n+1} - r = g'(\xi_n)(x_n - r),$$

که در آن ξ_n بین x_n و r قرار دارد اگر، خطای تکرار nام را به‌صورت $e_n = x_n - r$ و $e_{n+1} = x_{n+1} - r$ تعریف می‌کنیم، می‌توانیم بنویسیم.

$$e_{n+1} = g'(\xi_n)\, e_n$$

با گرفتن قدر مطلق از طرفین داریم:

$$|e_{n+1}| = |g'(\xi_n)| * |e_n|$$

اکنون فرض می‌کنیم که $|g'(x)| \leq k < 1$ به ازاء همه مقادیر x در یک فاصله به شعاع h در نزدیکی r بر قرار باشد، اگر x_1 در این فاصله انتخاب شود، x_2 نیز در این فاصله خواهد بود و الگوریتم همگرا خواهد شد زیرا که:

$$|e_{n+1}| \leq k\,|e_n| \leq k^2\,|e_{n-1}| \leq \ldots \leq (k^n)\,|e_1|$$

به طور خلاصه، اگر $g(x)$ و $g'(x)$ روی یک فاصله حول یک ریشه r از معادله $x=g(x)$ پیوسته باشند، و اگر برای کلیه مقادیر x دراین فاصله $|g'(x)| \leq 1$، در این صورت، $x_{n+1} = g(x_n)$ $n = 1,2,3,\ldots$ به سمت ریشه $x=r$ همگراست مشروط بر اینکه x_1 در این فاصله انتخاب شده باشد. توجه شود که این یک شرط کافی است، زیرا که برای بعضی معادلات همگرایی برقرار است اگرچه کلیه شرایط صادق نیستند.

بجای آزمون تحلیلی $|g'(\xi)|$، یک آزمون عملی آن است که صرفاً ملاحظه کنیم که مقادیر متوالی همگرا هستند. در یک برنامه کامپیوتری تعیین اینکه $|\xi_3-\xi_2| > |\xi_2-\xi_1|$ چنین ارزشی را دارد.

چون توضیحات بالا نشان می‌دهد که $e_{n+1} = g'(\xi_n) * e_n$، روشن است که سرعت همگرایی سریع است اگر $|g'(x)|$ در فاصله مذکور کوچک باشد. اگر مشتق منفی باشد، خطاها تغییر علامت می‌دهند و همگرایی نوسانی خواهد بود، همچنان که با انجام تکرارها x_n به ریشه نزدیک می‌شود، مقادیر $g'(\xi_n)$ به مقدار ثابت $g'(r)$ میل می‌کند زیرا که ξها در فاصله کوچکتری حول r تجمع می‌کنند، در حد هر خطا متناسب با خطاهای قبلی می‌شود. بدین دلیل، گاهی اوقات این روش «تکرار خطی» نامیده می‌شود.

تسریع همگرایی

اگر چه تناسب میان خطاهای متوالی فقط در وضعیت حدی درست است. اگر فرض کنیم که هر یک از خطاها متناسب با خطای قبلی است، می‌توانیم یک فن تسریع را بیان کنیم که تسریع ایتکن نامیده می‌شود که اغلب مفید است:

$$e_n = x_n - r = K^{n-1}e_1$$

فرض می‌کنیم:

$$x_n = r + K^{n-1}e_1.$$

یا

$$e_{n+1} = x_{n+1} - r = K^n e_1,$$

به همین ترتیب اگر:

$$e_{n+2} = x_{n+2} - r = K^{n+1}e_1,$$

پس داریم:

$$x_{n+1} = r + K^n e_1,$$

و

$$x_{n+2} = r + K^{n+1}e_1.$$

این جملات را در رابطه زیر جایگذاری می‌کنیم.

نتیجه می‌شود که:

$$\frac{x_n x_{n+2} - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n}.$$

$$\frac{x_n x_{n+2} - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n} = \frac{(r + K^{n-1}e_1)(r + K^{n+1}e_1) - (r + K^n e_1)^2}{(r + K^{n+1}e_1) - 2(r + Ke_1) + (r + K^{n-1}e_1)}$$

$$= \frac{r(K^{n+1} - 2K^n + K^{n-1})e_1}{(K^{n+1} - 2K^n + K^{n-1})e_1} = r.$$

چون به ریشه r دسترسی نداریم در واقع مقدار جدید x_{n+3} بطور تقریبی جواب است.

از سه برآورد متوالی ریشه، x_1, x_2, x_3 به یک برآورد اصلاح شده برون یابی می‌کنیم. چون فرض اینکه خطاهای متوالی نسبت ثابت دارند معمولا صحیح نیست، مقدار برون یافته دقیق نیست، اما معمولا اصلاح شده است، با استفاده از این فن از x_1 شروع کرده و دو مقدار را از روش تکراری محاسبه می‌کنیم و برون یابی کرده و مقدار جدید را محاسبه می‌کنیم و مجدداً برون یابی را ادامه می‌دهیم.

برای اجتناب از مشکل خطای گرد کردن که در تفاضل دو عدد بزرگ رخ می‌دهد، صورت دیگری را که مفید است، توضیح می‌دهیم:

تعریف می‌کنیم:

$$\Delta x_i = x_{i+1} - x_i,$$

$$\Delta^2 x_i = \Delta(\Delta x_i) = \Delta(x_{i+1} - x_i) = x_{i+2} - 2x_{i+1} + x_i.$$

دستورالعمل تسریع به صورت زیر خواهد شد.

$$r = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n} = \boxed{\frac{x_n x_{n+2} - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n}.}$$

این تفاضل‌ها به سادگی در یک جدول محاسبه می‌شوند.

مثال ۵.۲: معادله زیر را به روش آتکین حل کنید.

$$f(x) = x^2 - 2x - 3 = 0,$$
$$x_{n+1} = \sqrt{2x_n + 3}, \quad x_1 = 4,$$

x	Δx	$\Delta^2 x$
$x_1 = 4.000$	0.684	
$x_2 = 3.316$	0.212	0.472
$x_3 = 3.104$		

برآوردی که با تسریع بدست آمده عبارت است از

$$r = 4.000 - \frac{(0.684)^2}{0.472} = 3.009.$$

تقریباً دو تکرار جلو افتاده‌ایم، و اغلب روش تسریع آیتکن سریعتر است.

اغلب روش تسریع ایتکن پرش بزرگتر از این را می‌دهد: در حقیقت یک اندازه‌گیری جالب برای تعیین این پرش وجود دارد.

فرض کنید برای n داریم، x_n، x_{n+1}، x_{n+2}، x_{n+3} سپس مقدار C را تعیین کنیم:

$$C = \frac{\sum x_i x_{i+1} - \frac{1}{3}\sum x_i \sum x_{i+1}}{\sqrt{\left(\sum x_i^2 - \frac{1}{3}\left(\sum x_i\right)^2\right)\left(\sum x_{i+1}^2 - \frac{1}{3}\left(\sum x_{i+1}\right)^2\right)}},$$

به‌طوری که جمع $i = n$ تا $i = n+1$ می‌باشد. اگر C نزدیک به ± 1 باشد، در اینصورت شتاب ایتکن بیشترین تاثیر را دارد. در مثال حاضر مقادیر زیر را داریم.

بازاء $n=0$ در فرمول داریم:

$$x_0 = 4.000,$$
$$x_1 = 3.316,$$
$$x_2 = 3.104,$$
$$x_3 = 3.034.$$

برای اطلاعات بیشتر جهت بهبود روش تسریع به *Jones(1982)* مراجعه کنید.

۴.۲ روش نیوتن

یکی از روشهایی که بطور وسیعی در حل معادلات بکار برده می‌شوند روش نیوتن

است.[۱] شکل ۸.۲ یک شرح ترسیمی را ارائه می‌دهد. با شروع از تخمین اولیه که چندان از یک ریشه x، دور نیست در طول مماس به سمت نقطه تقاطع آن با محور x درون یابی می‌کنیم و این نقطه را به عنوان تقریب بعدی انتخاب می‌نماییم. این عمل ادامه می‌یابد تا اینکه مقادیر متوالی x به قدر کافی بهم نزدیک شوند، یا مقدار تابع به قدر کافی به صفر نزدیک گردد.[۲]

طرز محاسبه بلافاصله از مثلث قائم الزاویه در شکل ۸.۲ بدست می‌آید که زاویه شیب خط مماس در $x = x_1$ به عنوان یکی از زوایای حاده آن است:

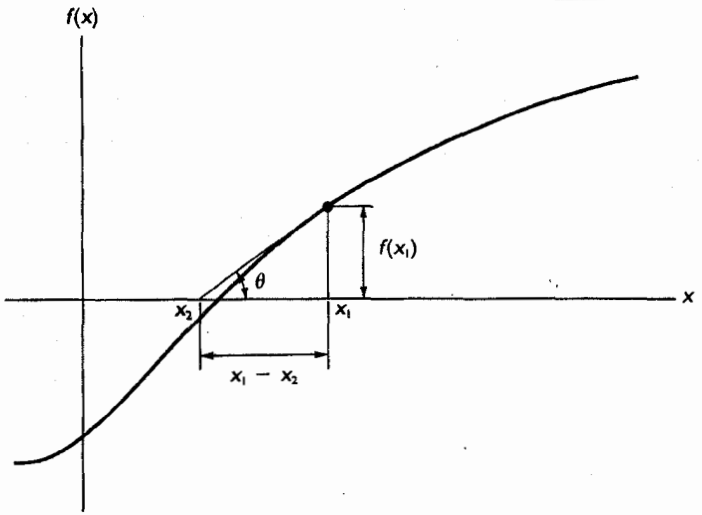
$$\tan \theta = f'(x_1) = \frac{f(x_1)}{x_1 - x_2}, \qquad x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

روش محاسبه را به صورت زیر ادامه می‌دهیم:

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)},$$

یا با جمله‌ای عمومی‌تر

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 1, 2, 3, \ldots$$



شکل ۸.۲

آلگوریتم نیوتن کاربرد وسیعی دارد، زیرا که حداقل در همسایگی یک ریشه، از هر روشی که تاکنون بحث شده است دارای همگرایی سریعتری است.

در بخش بعدی نشان می‌دهیم که این روش همگرائی از مرتبه دو دارد، بدین معنی که خطای

۱- نیوتن بحث مفصل این روش را چاپ نکرد، اما یک معادله درجه سه را در *Principia (1687)* حل کرد. صورتی از این روش که در اینجا داده شده است نسبت به مثال اصلی وی بطور قابل توجهی پیشرفت کرده است.

۲- اینکه کدام معیار باید بکار برده شود اغلب به مسأله فیزیکی خاص بستگی دارد که معادله برای آن بکار می‌رود. به طور عادی مطابقت مقادیر x متوالی با یک خطای قابل اغماض ضروری است.

هر گام مضربی از مربع خطای گام قبلی می‌باشد. بهر حال تعیین مقدار دو تابع $f(x_n)$ و $f'(x_n)$ در هر مرحله لازم است.

مثال ۶.۲ روش نیوتن برای $f(x) = 3x + sinx - e^x = 0$ بکار برید.

محاسبات زیر را خواهیم داشت:

$$f(x) = 3x + sinx - e^x,$$
$$f'(x) = 3 + cosx - e^x$$

اگر با $x_1 = 0.0$ شروع کنیم، داریم:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.0 - \frac{-1.0}{3.0} = 0.3333;$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 0.33333 - \frac{-0.068418}{2.54934} = 0.363017;$$

$$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)} = 0.363017 - \frac{-0.279\times10^{-4}}{2.50226} = 0.363217;$$

بعد از سه تکرار، ریشه تا هفت رقم با معنی صحیح می‌باشد. با مقایسه این نتیجه با نتایج جدول ۶.۲ مشاهده می‌شود که روش نیوتن بطور قابل ملاحظه‌ای سریعتر از روش‌های قبلی می‌باشد. لیکن برای مقایسه روش‌های عددی، معمولاً تعداد دفعاتی که توابع باید تعیین مقدار شوند، شمارش می‌شود. چون در روش نیوتن در هر گام تعیین مقدار دو تابع لازم است. آنطور که در ابتدا بنظر می‌رسید در مقایسه با سایر روشها این روش از سرعت بیشتری برخوردار نیست. در سه تکرار با روش نیوتن، تعیین مقدار شش تابع لازم است. در پنج تکرار با روش‌های قبلی تعیین مقدار هفت تابع لازم می‌شود. یک بیان قرار دادی الگوریتم روش نیوتن، که برای اجرای برنامه کامپیوتری مناسب است، در زیر نشان داده شده است.

روش نیوتن

برای تعیین ریشه $f(x) = 0$، یک مقدار به طور مناسب نزدیک به ریشه، برای x انتخاب می‌کنیم.

محاسبه $f(x_1)$, $f'(x_1)$

set $x_2 = x_1$

IF $(f(x_1) \neq 0)$ AND $(f'(x_1) \neq 0)$

REPEAT

SET $x_1 = x_2$.

Set $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$

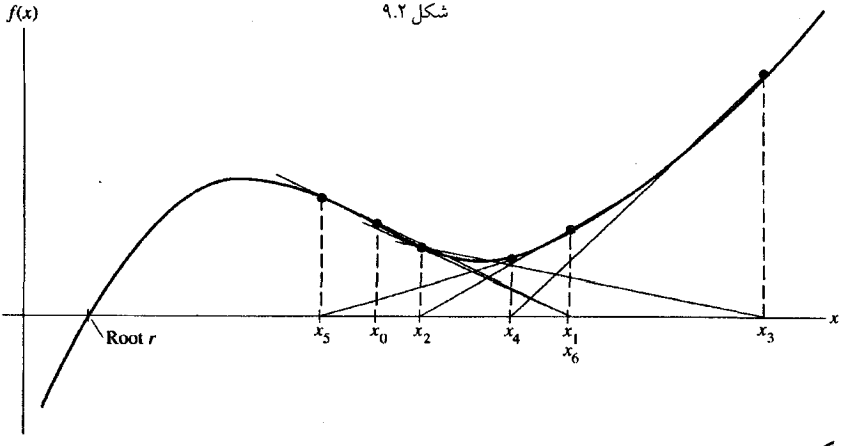
UNTIL $(|x_1 - x_2| < \varepsilon_1$ یا (ε_1 مقدار قابل اغماض است)

$|f(x_2)| < \varepsilon_2$ (ε_2 مقدار قابل اغماض است)

تذکر: اگر مقدار شروع به حد کافی نزدیک به ریشه نباشد. این روش ممکن است به یک ریشه غیر از آنکه انتظار داریم میل کند یا «واگرا» شود.

روش نیوتن می‌تواند برای توابع چند جمله‌ای بکار رود، البته با روش‌های خاص چنین کاربردی را تسهیل می‌کنند. در بخش بعدی این فصل این موضوع را بررسی می‌کنیم.
در برخی حالات روش نیوتن همگرا نخواهد شد، شکل ۹.۲ این وضعیت را روشن می‌کند. با شروع از نقطه x_1، به ریشه r هرگز نمی‌توان رسید. شرط تحلیلی همگرایی را در بخش بعدی بیان خواهیم کرد و نشان خواهیم داد که روش نیوتن در اکثر موارد همگرایی مرتبه دومی دارد.

شکل ۹.۲



۵.۲ همگرایی روش نیوتن

اکنون نتیجه بخش قبلی را بکار می‌بریم تا معیاری را برای همگرایی روش نیوتن نشان دهیم. الگوریتم

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 1, 2, 3, \ldots$$

به صورت $x_{n+1} = g(x_n)$ می‌باشد. تکرارهای متوالی همگرا هستند اگر $|g'(x)| < 1$ باشد.
چون:

$$g(x) = x - \frac{f(x)}{f'(x)},$$

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

بنابراین اگر:

$$\left|\frac{f(x)f''(x)}{[f'(x)]^2}\right| < 1$$

در یک همسایگی ریشه r، این روش با هر مقدار اولیه در فاصله مذکور، همگراست. این شرط فقط کافی است. و پیوستگی $f(x)$ و $f'(x)$ ضروری است. توجه شود که $f'(x)$ نباید صفر شود.

حالا نشان می دهیم که روش نیوتن دارای همگرائی مرتبه دوم می‌باشد. چون r یک ریشه از $f(x)=0$ می‌باشد پس $r=g(r)$. به علت اینکه $x_{n+1} = g(x_n)$، می توان نوشت.

$$x_{n+1} - r = g(x_n) - g(r)$$

فرض کنید $g(x_n)$ را به صورت یک بسط تیلور بر حسب $(x_n - r)$ تا جمله مشتق دوم (به عنوان باقیمانده) بنویسیم.

$$g(x_n) = g(r) + g'(r)(x_n - r) + \frac{g''(\xi)}{2}(x_n - r)^2,$$

که در آن ξ در فاصله x_n تا r قرار دارد.

$$g'(r) = \frac{f(r)f''(r)}{[f'(r)]^2} = 0$$

زیرا که $f(r)= 0$ (r یک ریشه است) لذا داریم.

$$g(x_n) = g(r) + \frac{g''(\xi)}{2}(x_n - r)^2.$$

با قرار دادن $e_n =(x_n - r)$ بدست می‌آوریم.

$$e_{n+1} = x_{n+1} - r = g(x_n) - g(r) = \frac{g''(\xi)}{2} e_n^2$$

هر خطا متناسب با مربع خطای قبلی (در حد) است، یعنی روش نیوتن همگرایی مرتبه دوم دارد.[1]

۶.۲ روش نیوتن برای چند جمله‌ایها

توابع چند جمله‌ای اهمیت خاصی را دارا می‌باشند. در ادامه این کتاب خواهیم دید که بسیاری از روش‌های عددی با ارزش، بر مبنای چند جمله‌ایها هستند.

این نقش مهم توابع چند جمله‌ای مربوط به رفتار «زیبای» آنهاست. آنها همه جا پیوسته هستند، آنها هموار هستند، مشتقات آنها نیز پیوسته و هموار هستند، آنها سریعاً تعیین مقدار می‌شوند. چند جمله‌ای‌ها مخصوصاً با کامپیوتر خوب مطابقت داده می‌شوند زیرا عملیات ریاضی که برای محاسبه لازم است، فقط جمع، تفریق، ضرب و تقسیم می‌باشند، که همگی عملیات سریعی در کامپیوتر هستند.

بخاطر این اهمیت خاص چند جمله‌ایها، اکنون روش پیدا کردن ریشه‌ها را در مورد آنها بررسی می‌کنیم. برای اکثر روشهائی که قبلا بحث شد نکته جدیدی برای گفتن نیست، اما برای روش نیوتن ایده‌های جدید و با ارزشی وجود دارد.

۱- اگر $f(x)$ در $x = r$ دارای ریشه تکراری باشد، سرعت همگرایی از مرتبه دوم نخواهد بود، برای ریشه مکرر مرتبه دو، می‌توان نشان داد که همگرایی خطی است.

روش هورنر[1]

در بکار بردن روش نیوتن برای چند جمله‌ای‌ها، سودمندترین راه آن است که $f(x)$ و $f'(x)$ را با استفاده از تقسیم ترکیبی بدست آوریم. این روش همچنین مفیدترین راه برای محاسبه چند جمله‌ایها و مشتقات آنها در یک برنامه کامپیوتری می‌باشد.

چند جمله‌ای درجه nام زیر را در نظر می‌گیریم:

$$P_n(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n.$$

$P_n(x)$ را بر عامل $(x-x_1)$ تقسیم می‌کنیم تا یک چند جمله‌ای کاهش یافته $Q_{n-1}(x)$ از درجه $(n-1)$ و یک باقیمانده b_n که ثابت است بدست آید:

$$\frac{P_n(x)}{x - x_1} = Q_{n-1}(x) + \frac{b_n}{x - x_1}.$$

به ترتیب دیگر بدست می‌آید.

$$P_n(x) = (x - x_1)Q_{n-1}(x) + b_n.$$

توجه شود که در $x=x_1$ داریم

$$P_n(x_1) = (0)[Q_{n-1}(x_1)] + b_n,$$

طبق قضیه باقیمانده، در تقسیم $P_n(x)$ بر $(x-x_1)$ باقیمانده برابر با مقدار چند جمله‌ای در $x = x_1$، $P_n(x_1)$، می‌باشد.

اگر از معادله آخر مشتق بگیریم، داریم

$$P'_n(x) = (x - x_1)Q'_{n-1}(x) + (1)Q_{n-1}(x) + 0.$$

فرض می‌کنیم $x = x_1$ داریم:

$$P'_n(x_1) = Q_{n-1}(x_1).$$

مقدار این چند جمله‌ای Q_{n-1} را با یک تقسیم دومی که باقیمانده‌اش برابر $Q_{n-1}(x_1)$ است، تعیین می‌کنیم، الگوریتم تقسیم ترکیبی را با نوشتن $Q_{n-1}(x)$ به شکل مشابه $P_n(x)$ ادامه می‌دهیم.

$$\begin{aligned} P_n(x) &= a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n \\ &= (x - x_1)Q_{n-1}(x) + b_n \\ &= (x - x_1)(b_0x^{n-1} + b_1x^{n-2} + \cdots + b_{n-2}x + b_{n-1}) + b_n. \end{aligned}$$

با انجام عملیات ضرب و مساوی قرار دادن جملات مشابه نسبت به x بدست می‌آوریم.

x^n: $a_0 = b_0$		$b_0 = a_0$
x^{n-1}: $a_1 = b_1 - b_0x_1$	یا or	$b_1 = a_1 + b_0x_1$
x^{n-2}: $a_2 = b_2 - b_1x_1$		$b_2 = a_2 + b_1x_1$
x: $a_{n-1} = b_{n-1} - b_{n-2}x_1$		$b_{n-1} = a_{n-1} + b_{n-2}x_1$
: $a_n = b_n - b_{n-1}x_1$		$b_n = a_n + b_{n-1}x_1.$

1- Horner

در حالت کلی داریم $b_i = a_i + b_{i-1}x_1$ که کلیه مقادیر b بجز b_0 محاسبه می‌شوند. برای محاسبه مشتق، مقادیر c از مقادیر b محاسبه می‌شوند. به همان روشی که مقادیر b از a بدست می‌آیند.
تقسیم ترکیبی همچنین به روش «ضرب آشیانه‌ای» برای تعیین مقادیر چند جمله‌ای مشهور است چند جمله‌ای درجه پنج زیر را در $x=x_1$ در نظر بگیرید.

$$a_0x_1^5 + a_1x_1^4 + a_2x_1^3 + a_3x_1^2 + a_4x_1 + a_5.$$

می‌توانیم مجدداً بنویسیم:

$$((((a_0x_1 + a_1)x_1 + a_2)x_1 + a_3)x_1 + a_4)x_1 + a_5.$$

در حالت کلی به تعداد $15=2+3+4+5$ ضرب و پنج جمع لازم می‌شود. در شکل آشیانه‌ای، فقط پنج ضرب باضافه پنج جمع لازم می‌شود، بدیهی است که این روش مناسب‌تر است.
مقایسه ضرب آشیانه‌ای با معادلات $b_i = a_i+b_{i-1}x_1$ و $b_2 = a_2+b_1x_1$ برای تقسیم ترکیبی، نشان می‌دهد که جملات متوالی دقیقاً به یک شکل تشکیل می‌شوند، به‌طوریکه تقسیم ترکیبی وضرب آشیانه‌ای دو نام برای یک روش هستند.

بعد از اینکه اولین ریشه پیدا شد، معمولا برای تعیین سایر ریشه‌ها از چندجمله‌ای کاهش یافته $Q_{n-1}(x)$ استفاده می‌کنیم، این روش محاسبات را کوتاهتر می‌کند.
اگر معادله کاهش یافته، درجه دوم باشد، بنابراین فرمول درجه دوم بکار برده می‌شود، اما اگر یک چند جمله‌ای از درجه بالاتر را حل می‌کنیم، روش نیوتن با به کار بردن تقسیم ترکیبی بکار گرفته می‌شود تا با تخمین اولیه، ریشه دوم را بدست آوریم و این روند تکرار می‌شود تا اینکه معادله کاهش یافته از درجه دوم شود.
باید توجه نمود که استفاده از توابع کاهش یافته می‌تواند دستخوش خطاهای غیر منتظره شود. اگر اولین ریشه به طور تقریبی تعیین شود و اگر ضرایب معادله کاهش یافته دارای دقت کافی نباشند، ریشه‌های بعدی نه تنها دستخوش خطاهای گرد کردن هستند، بلکه همچنین دستخوش خطاهایی هستند که در اثر وجود ضرایب غیر دقیق حاصل می‌شوند. برخی توابع، فوق‌العاده به این وضع حساس هستند و تغییرات کوچکی در تعدادی ضرایب باعث اختلاف‌های بزرگی در ریشه‌ها می‌شود که اصطلاحاً «ناهنجار» یا ناجور نامیده می‌شوند. در این روش برای دقت بیشتر و اجتناب از حالت «ناجوری» *(Ill-condition)* دقت مضاعف می‌تواند کمک کند و همچنین می‌توان از بزرگترین ریشه از نظر قدر مطلق شروع کرده، به ترتیب جوابها را بدست آورد.

۷.۲ روش بیرستو برای عامل‌های درجه دوم

استفاده از روشهائی که تاکنون بحث شد برای پیدا کردن ریشه مختلط یک چند جمله‌ای مشکل است هرچند که روش نیوتن را اگر با یک تخمین اولیه مختلط شروع کنیم رضایت بخش است اما در محاسبه با دست، انجام ضرب و تقسیم اعداد مختلط چندان روش مناسبی نیست و

اجرای برنامه کامپیوتری نیز نامناسب است. مخصوصا در این روش به این نکته که مزدوج هر ریشه نیز در معادله صدق می‌کند توجهی نکرده‌ایم و از آن استفاده نشده است.

برای چند جمله‌ایی‌ها اگر ضرایب حقیقی باشند ریشه‌های مختلط به صورت جفت‌های مزدوج رخ می‌دهند. در این حالت، اگر فاکتورهای درجه دومی که حاصلضرب دو ریشه مزدوج هستند استخراج کنیم می‌توان از محاسبات مختلط پرهیز کرد زیرا که چنین فاکتورهای درجه دومی دارای ضرایب حقیقی هستند. ابتدا آلگوریتم تقسیم ترکیبی را با چند جمله‌ای آزمایشی $x^2 - r\,x - s$ بسط می‌دهیم.

$$\begin{aligned} p(x) &= a_0x^n + a_1x^{n-1} + \ldots + a_n \\ &= (x - rx - s)\,Q_{n-2}(x) + \text{باقیمانده} \\ &= (x^2 - rx - s)(b_0x^{n-2} + b_1x^{n-3} + \ldots + b_{n-3}x + b_{n-2}) \\ &\quad + b_{n-1}(x - r) + b_n. \end{aligned}$$

(باقیمانده $b_{n-1}\,(x-r) + b_n$ است، و بدین صورت نوشته می‌شود تا بعدا موجب سادگی شود. اگر $x^2 - rx - s$ مقسوم علیه دقیق $P_n(x)$ باشد، b_{n-1}, b_n هر دو صفر هستند.)

با انجام اعمال ضرب و مساوی قرار دادن ضرایب توان‌های مشابه x، داریم.

$$\begin{array}{lll} a_0 = b_0 & & b_0 = a_0 \\ a_1 = b_1 - rb_0 & & b_1 = a_1 + rb_0 \\ a_2 = b_2 - rb_1 - sb_0 & & b_2 = a_2 + rb_1 + sb_0 \\ a_3 = b_3 - rb_2 - sb_1 & \text{or} & b_3 = a_3 + rb_2 + sb_1 \\ \vdots & & \vdots \\ a_{n-1} = b_{n-1} - rb_{n-2} - sb_{n-3} & & b_{n-1} = a_{n-1} + rb_{n-2} + sb_{n-3} \\ a_n = b_n - rb_{n-1} - sb_{n-2} & & b_n = a_n + rb_{n-1} + sb_{n-2} \end{array} \qquad (1.1)$$

مایلیم که b_n و b_{n-1} هر دو صفر شوند چون این نشان آن است که $x^2 - rx - s$ یک عامل درجه دوم از چند جمله‌ای می‌باشد. بار اول معمولا چنین نخواهد بود. مگر آنکه r و s را به طور مناسبی تغییر دهیم و باقیمانده را صفر کنیم یا حداقل خطای ضرایب آن را کوچک کنیم.

بدیهی است که b_n و b_{n-1} هر دو تابعهایی از دو پارامتر r و s هستند.

با بسط b_n و b_{n-1} به‌صورت سری تیلور یک تابع دو متغیره را برحسب $(r^* - r)$ و $(s^* - s)$ بدست می‌آوریم که در آن $(r^* - r)$ و $(s^* - s)$ آنقدر کوچک فرض می‌شوند به‌طوری‌که جملات مرتبه بالاتر از یک، قابل صرف نظر کردن باشند.

$$b_{n-1}(r^*,s^*) = b_{n-1}(r,s) + \frac{\partial b_{n-1}}{\partial r}(r^* - r) + \frac{\partial b_{n-1}}{\partial s}(s^* - s) + \cdots,$$

$$b_n(r^*,s^*) = b_n(r,s) + \frac{\partial b_n}{\partial r}(r^* - r) + \frac{\partial b_n}{\partial s}(s^* - s) + \cdots.$$

فرض کنیم (r^*, s^*) نقطه‌ای باشد که در آن باقیمانده صفر می‌شود، و

$$r - r^* = \Delta r, \qquad s - s^* = \Delta s$$

(Δr و Δs نموهایی هستند که به مقادیر اصلی r و s افزوده می‌شوند تا مقادیر جدید s^* و r^* بدست آیند که بازاء آنها باقیمانده چند جمله‌ای حاصل صفر است). پس

$$b_{n-1}(r^*,s^*) = 0 \simeq b_{n-1} + \frac{\partial b_{n-1}}{\partial r}\Delta r + \frac{\partial b_{n-1}}{\partial s}\Delta s,$$

$$b_n(r^*,s^*) = 0 \simeq b_n + \frac{\partial b_n}{\partial r}\Delta r + \frac{\partial b_n}{\partial s}\Delta s.$$

این دو معادله را به طور همزمان نسبت به مجهول‌های Δr و Δs حل می‌کنیم، بنابراین لازم است که مشتقات نسبی را محاسبه کنیم.

بیرستو نشان داد که می‌توان مشتقات نسبی لازم را توسط تقسیم ترکیبی دیگری بر عامل $x^2 - rx - s$ یافت. درست به همان طریقی که bها از aها بدست آمدند، یک مجموعه از cها را با روابطی که در سمت چپ زیر نشان داده شده تعریف کرده، و با مشتقات نسبی در ستون‌های سمت راست مقایسه می‌کنیم:

$c_0 = b_0$	$\frac{\partial b_0}{\partial r} = \frac{\partial a_0}{\partial r} = 0$	$\frac{\partial b_0}{\partial s} = \frac{\partial a_0}{\partial s} = 0$
$c_1 = b_1 + rc_0$	$\frac{\partial b_1}{\partial r} = r\frac{\partial b_0}{\partial r} + b_0 = b_0 = c_0$	$\frac{\partial b_1}{\partial s} = \frac{\partial a_1}{\partial s} + r\frac{\partial b_0}{\partial s} = 0$
$c_2 = b_2 + rc_1 + sc_0$	$\frac{\partial b_2}{\partial r} = r\frac{\partial b_1}{\partial r} + b_1 = c_1$	$\frac{\partial b_2}{\partial s} = r\frac{\partial b_1}{\partial s} + s\frac{\partial b_0}{\partial s} + b_0 = b_0 = c_0$
$c_3 = b_3 + rc_2 + sc_1$	$\frac{\partial b_3}{\partial r} = r\frac{\partial b_2}{\partial r} + b_2 + s\frac{\partial b_1}{\partial r} = b_2 + rc_1 + sc_0 = c_2$	$\frac{\partial b_3}{\partial s} = r\frac{\partial b_2}{\partial s} + s\frac{\partial b}{\partial s} + b_1 = b_1 + rc_0 = c_1$
$\vdots$	$\vdots$	$\vdots$
$c_{n-1} = b_{n-1} + rc_{n-2} + sc_{n-3}$	$\frac{\partial b_{n-1}}{\partial r} = r\frac{\partial b_{n-2}}{\partial r} + b_{n-2} + s\frac{\partial b_{n-3}}{\partial r} = b_{n-2} + rc_{n-3} + sc_{n-4} = c_{n-2}$	$\frac{\partial b_{n-1}}{\partial s} = r\frac{\partial b_{n-2}}{\partial s} + s\frac{\partial b_{n-3}}{\partial s} + b_{n-3} = b_{n-3} + rc_{n-4} + sc_{n-5} = c_{n-3}$

از این‌رو مشتقات نسبی که مورد نیاز است با cهای متناظر خاص مساوی هستند. تساوی‌های به طور همزمان نشان می‌دهند.

$$-b_{n-1} = c_{n-2}\,\Delta r + c_{n-3}\,\Delta s$$
$$-b_n = c_{n-1}\,\Delta r + c_{n-2}\,\Delta s$$

مقادیر Δr و Δs بدست می‌آیند.

$$\Delta r = \frac{\begin{vmatrix} -b_{n-1} & c_{n-3} \\ -b_n & c_{n-2} \end{vmatrix}}{\begin{vmatrix} c_{n-2} & c_{n-3} \\ c_{n-1} & c_{n-2} \end{vmatrix}},$$

$$\Delta s = \frac{\begin{vmatrix} c_{n-2} & -b_{n-1} \\ c_{n-1} & -b_n \end{vmatrix}}{\begin{vmatrix} c_{n-2} & c_{n-3} \\ c_{n-1} & c_{n-2} \end{vmatrix}}.$$

یک بیان از آلگوریتم برای روش بیرستو در بخش‌های بعدی این فصل ارائه می‌شود.

مثال ۷.۲ عامل‌های درجه دوم $x^4 - 1.1x^3 + 2.3x^2 + 0.5x + 33 = 0$ را بدست آورید.

از $x^2 + x + 1$ به عنوان عامل شروع ($r=-1$ و $s=-1$) استفاده کنید. (اغلب از $r=s=0$ به عنوان مقادیر شروع استفاده می‌شود مگر اطلاع دیگری برای یک عامل تقریبی داده شده باشد). معادلات (۱.۲) به یک دستورالعمل تقسیم ترکیبی دوگانه مطابق زیر تبدیل می‌شوند.

	a_0	a_1	a_2	a_3	a_4
	1	−1.1	2.3	0.5	3.3
$r = -1$		−1.0	2.1	−3.4	0.8
$s = -1$		—	−1.0	2.1	−3.4
	1	−2.1	3.4	(−0.8)	(0.7)
		−1.0	3.1	−5.5	
		—	−1.0	3.1	
	1	(−3.1)	(5.5)	(−3.2)	
		c_{n-3}	c_{n-2}	c_{n-1}	

b_{n-1} ، b_n

توجه شود که معادلات b_1 و c_1 بر حسب s نیستند. خط‌های تیره در جدول بالا نمایش عامل‌هایی است که وجود ندارند. سپس داریم:

$$\Delta r = \frac{\begin{vmatrix} 0.8 & -3.1 \\ -0.7 & 5.5 \end{vmatrix}}{\begin{vmatrix} 5.5 & -3.1 \\ -3.2 & 5.5 \end{vmatrix}} = \frac{2.23}{20.33} = 0.11, \qquad r^* = -1 + 0.11 = -0.89,$$

$$\Delta s = \frac{\begin{vmatrix} 5.5 & 0.8 \\ -3.2 & -0.7 \end{vmatrix}}{20.33} = \frac{-1.29}{20.33} = -0.06, \qquad s^* = -1 - 0.06 = -1.06.$$

آزمایش دوم نتیجه می‌دهد.

	1	−1.1	2.3	0.5	3.3
−0.89		−0.89	1.77	−2.68	0.06
−1.06		—	−1.06	2.11	−3.17
	1	−1.99	3.01	−0.07	0.17 ← b's
		−0.89	2.56	−4.01	
		—	−1.06	3.05	
		−2.88	4.51	−1.03	← c's

$$\Delta r = \frac{\begin{vmatrix} 0.07 & -2.88 \\ -0.17 & 4.51 \end{vmatrix}}{\begin{vmatrix} 4.51 & -2.88 \\ -1.03 & 4.51 \end{vmatrix}} = \frac{-0.175}{17.374} = -0.010, \qquad r^* = -0.89 - 0.010 = -0.900,$$

$$\Delta s = \frac{\begin{vmatrix} 4.51 & 0.07 \\ -1.03 & -0.17 \end{vmatrix}}{17.374} = \frac{-0.694}{17.374} = -0.040, \qquad s^* = -1.06 - 0.040 = -1.100.$$

عامل‌های دقیق برابرند با $(x^2+0.9x+1.1)(x^2-2x+3)$.

۸.۲ روش مولر

هر یک از روش‌های پیدا کردن ریشه‌ها که تاکنون بررسی کرده‌ایم تقریب تابع در همسایگی ریشه با یک خط مستقیم می‌باشد.

روش مولر[1] یک روش درون یابی است که روش درون‌یابی رتبه دو را به جای درون یابی خطی در بخش‌های قبلی بکار می‌برد. یک چند جمله‌ای درجه دو که از سه نقطه نزدیک یک ریشه می‌گذرد، $[x_0, f(x_0)]$، $[x_1, f(x_1)]$ و $[x_2, f(x_2)]$، برای تابع $f(x)$ بدست می‌آوریم. با استفاده از فرمول ریشه‌های معادله درجه دوم، ریشه مناسب این سه جمله‌ای درجه دوم به عنوان برآورد اصلاح شده ریشه بدست می‌آید. با استفاده از نزدیکترین سه نقطه به‌ریشه‌ای که محاسبه می‌شود، عمل را ادامه می‌دهیم. طرز عمل برای روش مولر با نوشتن یک معادله درجه دوم به شکل av^2+bv+c که از سه نقطه نزدیک به ریشه می‌گذرد، بیان می‌شود (شکل ۹.۲ را ببینید). اگر محور را طوری تبدیل کنیم که از نقطه میانی بگذرد، یعنی با فرض $v = x - x_0$ بیان را ساده می‌کنیم.

فرض کنیم $h_1 = x_1 - x_0$ و $h_2 = x_0 - x_2$ با تعیین مقدار $P_2(v)$ در این سه نقطه، ضرایب را محاسبه می‌کنیم.

$$v = 0: \quad a(0)^2 + b(0) + c = f_0;$$
$$v = h_1: \quad ah_1^2 + bh_1 + c = f_1;$$
$$v = -h_2: \quad ah_2^2 - bh_2 + c = f_2.$$

۱- *Muller*

از معادله اول $c = f_0$ با فرض $h_2/h_1 = \gamma$، می‌توانیم دو معادله دیگر را نسبت به a و b حل کنیم.

$$a = \frac{\gamma f_1 - f_0(1 + \gamma) + f_2}{\gamma h_1^2(1 + \gamma)}, \qquad b = \frac{f_1 - f_0 - ah_1^2}{h_1}.$$

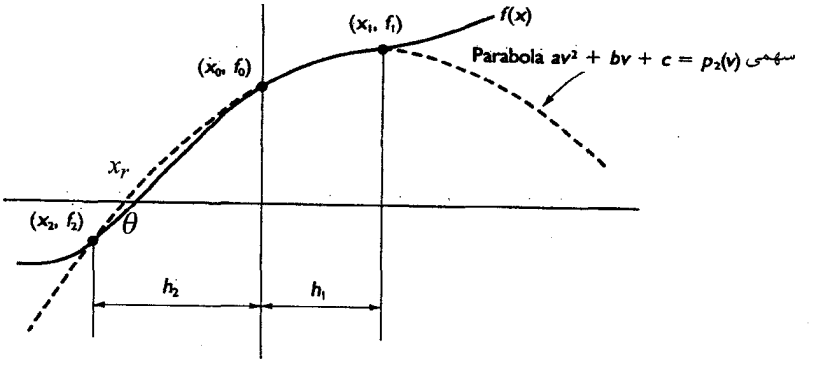
بعد از محاسبه $a,\ b,\ c$ ریشه‌های معادله درجه دوم $av^2 + bv + c = 0$ را بدست می‌آوریم، و نزدیکترین ریشه به نقطه میانی را انتخاب می‌کنیم. این مقدار برابر است با:
با انتخاب $x = x_o + v$

$$\text{ریشه} = x_0 - \frac{2c}{b \pm \sqrt{b^2 - 4ac}},$$

علامتی را در مخرج انتخاب می‌کنیم که بزرگترین مقدار مطلق را در مخرج داشته باشیم (یعنی، اگر $b>0$ آنگاه علامت مثبت، اگر $b<0$ آنگاه علامت منفی، اگر $b=0$ هر یک از علایم را می‌توان انتخاب کرد.)

ریشه چند جمله‌ای را به عنوان یک نقطه از سه نقطه برای تقریب بعدی اختیار می‌کنیم، که عبارت است از سه نقطه‌ای که نزدیکترین فاصله را دارند (یعنی اگر ریشه سمت راست x_0 باشد، x_0 و خود x_1 و ریشه را اختیار می‌کنیم، اگر سمت چپ آنست، x_0 و x_2 و خود ریشه را اختیار می‌کنیم.)

همیشه نامگذاری اندیس متغیرها را تغییر داده تا اینکه x_0 نقطه میانی سه مقدار باشد.



شکل ۹.۲

یک آلگوریتم برای روش مولر در زیر نشان داده شده است:

روش مولر

نقاط داده شده x_2 و x_0 و x_1 به ترتیب می‌باشند.

۱- مقدار تابع‌های در نظر گرفته شدهٔ f_2، f_0 و f_1 را حساب می‌کنیم.

۲- مقدار ضرایب سهمی را به وسیله سه نقطه پیدا می‌کنیم.

۳- دو ریشه معادله سهمی را حساب می‌کنیم.

۴- نزدیکترین ریشه به x_0 را انتخاب کرده و آن را در x_r قرار می‌دهیم.

۵- x_1 و x_0 و x_2 به‌ترتیب x_1 و x_r و x_0 جایگذاری کنید $IF x_r > x_0$ *THEN*

x_1 و x_0 و x_2 به ترتیب x_0 و x_r و x_2 جایگذاری کنید *ELSE*

۶- $IF\ |f(x_r)| < FTOL,\ THEN\ RETURN(x_r)$

ELSE go to 1.

مثال ۸.۲: یک ریشه $f(x)=\sin x - x/2$ را از روش مولر پیدا کنید.

فرض کنید

$$x_0 = 2.0, \quad f(x_0) = -0.09070, \quad h_1 = 0.2;$$
$$x_1 = 2.2, \quad f(x_1) = -0.29150, \quad h_2 = 0.2;$$
$$x_2 = 1.8, \quad f(x_2) = 0.07385, \quad \gamma = 1.0.$$

سپس

$$a = \frac{(1.0)(-0.29150) - (-0.09070)(2.0) + 0.07385}{(1.0)(0.2)^2(2.0)} = -0.45312,$$

$$b = \frac{-0.29150 - (-0.09070) - (-0.45312)(0.2)^2}{0.2} = -0.91338,$$

$$c = -0.09070;$$

ریشه $$2.0 - \frac{2(-0.09070)}{-0.91338 - \sqrt{(0.91338)^2 - 4(-0.45312)(-0.09070)}} = 1.89526.$$

برای تکرار بعدی داریم

$$x_0 = 1.89526, \quad f(x_0) = 1.9184 \times 10^{-4}, \quad h_1 = 0.10474;$$
$$x_1 = 2.0, \quad f(x_1) = -0.09070, \quad h_2 = 0.09526;$$
$$x_2 = 1.8, \quad f(x_2) = 0.07385, \quad \gamma = 0.9095.$$

$$a = \frac{(0.9095)(-0.09070) - (1.9184 \times 10^{-4})(1.9095) + 0.07385}{(0.9095)(0.10474)^2(1.9095)} = -0.47280,$$

ریشه $$b = \frac{-0.09070 - 1.9184 \times 10^{-4} - (-0.47280)(0.10474)^2}{0.10474} = -0.81826,$$

$$c = 1.9184 \times 10^{-4};$$

$$\text{ریشه} = 1.89526 - \frac{(2)(1.9184 \times 10^{-4})}{-0.81826 - \sqrt{(0.81826)^2 - 4(-0.47280)(1.9184 \times 10^{-4})}}$$

$$= 1.895494. \quad \blacksquare$$

بعد از اولین تکرار، ریشه تا هفت رقم با معنی بدست آمده است. تجربه نشان می‌دهد که روش مولر مشابه روش نیوتن تقریبا دارای همگرایی مرتبه دوم می‌باشد، لیکن محاسبه مشتقات لازم نیست و (بعد از آنکه مقادیر شروع بدست آمدند) برای هر تکرار فقط یکبار تعیین مقدار تابع لازم است.

۹.۲ روش گرافی[۱]

می‌توان مقادیر ریشه‌های یک چند جمله‌ای را مستقیماً از ضرایب آن بدست آورد، بدون اینکه نیازی به مقادیر شروع باشد. این روش بر اساس این حقیقت است که اگر کلیّه ریشه‌ها متفاوت بوده و اختلاف آنها از یکدیگر زیاد باشد، آنگاه برای چند جمله‌ای ریشه‌ها مشخص می‌شوند.

$$p_n(x) = a_1x^n + a_2x^{n-1} + \ldots + a_nx + a_{n+1},$$

$$r_1 = -\frac{a_2}{a_1}, r_2 = -\frac{a_3}{a_2}, \ldots, r = -\frac{a_{n+1}}{a_n}$$

برای جدا کردن ریشه‌های چند جمله‌ائی مفروض از یکدیگر، آنرا به چند جمله‌ائی دیگری با همان درجه تبدیل می‌کنیم که ریشه‌هایش مربعات ریشه‌های چند جمله‌ائی اصلی است.

با بیان دیگر روش گرافی چند جمله‌ای $P_n(x)$ را به چند جمله‌ای دیگری از همان درجه تبدیل می‌کند. بطوری که ریشه‌هایش مربع ریشه‌های معادله اصلی باشد. اگر ریشه‌های $P_n(x)$ حقیقی و مجزا باشند، ریشه‌های معادله جدید بطور وسیع‌تری نسبت به معادله اولیه از هم جدا می‌شوند. این مخصوصا برای ریشه‌های بزرگتر از یک از نظر قدر مطلق صادق می‌باشد. این عمل را ادامه می‌دهیم تا ریشه‌ها واقعا از هم دور شوند، در این صورت می‌توانیم ریشه‌ها را مستقیما از ضرایب چند جمله‌ای بدست آوریم.

مثال ساده زیر نشان می‌دهد چطور چند جمله‌ای جدید را وقتی چند جمله‌ای اولیه از مرتبه سه می‌باشد، بدست می‌آوریم.

$$p(x) = (x - 1)(x + 2)(x - 3)$$

سپس

$$p(-x) = (-x - 1)(-x + 2)(-x - 3)$$
$$= (-1)^3(x + 1)(x - 2)(x + 3)$$

با ضرب این دو

$$p(-x) * p(x) = (-1)^3(x^2 - 1^2)(x^2 - 2^2)(x^2 - 3^2)$$
$$= (-1)^3(x^2 - 1)(x^2 - 4)(x^2 - 9)$$
$$= (-1)^3(z - 1)(z - 4)(z - 9)$$

۱- *Graeffe*

و یک چند جمله‌ای بر حسب $x^2 = z$ داریم که ریشه‌هایش مربع ریشه‌های معادله $p(x)$ می‌باشد. [توان $(-1)^n$ یعنی n درجه چند جمله‌ای می‌باشد]. البته ما هرگز $p(x)$ را بشکل تجزیه شده نداریم، اما نتیجه یکسان است.

مشاهده می شود که با عمل مربع کردن علامت ریشه‌ها را از دست می‌دهیم بطوری که قدر مطلق ریشه‌ها مجزا می‌شوند.

فرض کنیم معادله اصلی را k بار در هم ضرب کنیم. می‌توانیم ریشه‌ها را با k بار جذر گرفتن (ریشه 2^k ام) عبارت زیر تخمین بزنیم.

$$\left|\frac{a_j}{a_{j-1}}\right| \quad j = 1, \ldots, n.$$

n درجه چند جمله‌ای است.

مثال ۹.۲ روش مربع ریشه‌ها را روی چند جمله‌ای $x^3 - 3x^2 - 6x + 8$ بکار برید.

سه چند جمله‌ای جدید (پس از جایگذاری x بجای x^2) بدست می‌آید:

$$\begin{aligned}
k = 1: &\quad x^3 - 21x^2 + 84x - 64,\\
k = 2: &\quad x^3 - 273x^2 + 4368x - 4096,\\
k = 3: &\quad x^3 - 65793x^2 + 16843008x - 16777216.
\end{aligned}$$

(با استفاده از نرم افزارهای محاسباتی می‌توان این نتایج را بدست آورد).

تخمین جواب‌ها از اولین چند جمله برابر است با

$$\sqrt{\frac{64}{84}} = 0.8729, \qquad \sqrt{\frac{84}{21}} = 2, \qquad \sqrt{\frac{21}{1}} = 4.5826.$$

از معادله دوم، بدست می‌آید.

$$\sqrt[4]{\frac{4096}{4368}} = 0.9841, \qquad \sqrt[4]{\frac{4368}{273}} = 2, \qquad \sqrt[4]{\frac{273}{1}} = 4.0648.$$

از معادله سوم داریم:

$$\sqrt[8]{\frac{16777216}{16843008}} = 0.9995, \qquad \sqrt[8]{\frac{16843008}{65793}} = 2, \qquad \sqrt[8]{\frac{65793}{1}} = 4.0020.$$

مقادیر جواب 1 و -2 و 4 هستند. باید علامت ریشه‌های بدست آمده را با جایگذاری در معادله تعیین کنیم. اگر علامت ریشه مثبت باشد، نتیجه جایگذاری تقریبا صفر است، والا ریشه منفی است.

۱۰.۲ برنامه‌های کامپیوتر

از این به بعد، در هر فصل چند برنامه به عنوان ضمیمه می‌آوریم تا بعضی از آلگوریتم‌های بحث شده در آن فصل را اجرا کنیم. ما سعی نداریم برای هر آلگوریتم، برنامه آن را بنویسیم،

دانشجویان با برنامه‌هائی که خودشان می‌نویسند می‌توانند تجربه لازم را بدست آورند. این مهارت برای شکوفائی آنالیز عددی ضروری است، اگرچه ممکن است بسته‌های نرم افزاری اکثر برنامه‌هائی حرفه‌ای لازم در این زمینه را فراهم کرده باشند.

ولی شخص نه برنامه را می‌تواند بخواند و نه تصمیم بگیرد کدامیک برای کاری که در دست دارد مناسب‌تر است. مگر آنکه حداقل زمینه قبلی برای بوجود آوردن برنامه‌های ساده را داشته باشد. ما همچنین انواع زبان‌های برنامه نویسی را در این برنامه‌ها استفاده می‌کنیم و تأکید می‌کنیم انتخاب زبان خیلی مهم نیست.

برنامه ۱.۲ روش مولر

در این قسمت لیست یک برنامه به نام *PMULLER* به زبان *C* آورده شده است که از روش مولر استفاده می‌کند.

F تابعی که ریشه صفر آن پیدا می‌شود. این تابع در برنامه فرعی جداگانه تعریف شده است.

XR مقدار ریشه که برگشت داده می‌شود. همچنین برای انتقال یک تخمین اولیه ریشه‌ها به برنامه فرعی استفاده می‌شود.

H اندازه تغییر مکان از تخمین اولیه که برای ساختن اولین تقریب درجه دوم به تابع استفاده می‌شود.

XTOL معیار توقف در مقادیر X یا خطای قابل اغماض برای x

FTOL معیار توقف در مقادیر $f(x)$ یا خطای قابل اغماض برای $f(x)$

NLIM محدودیت ماکزیمم تعداد تکرارها که در برنامه فرعی بکار می‌رود.

```
/*
   ****************************
   **                        **
   **       PMULLER.C        **
   **                        **
   ****************************

 APPLIED  NUMERICAL  ANALYSIS

 ------------------------------------------------------------------

   This program uses Muller's method to solve the equation

   f(x) = 3*x + sin(x) - exp(x) = 0.
*/

#include <stdio.h>
#include <math.h>

  float xr,        /* on input the initial guess, on output the root */
        h,         /* the original step size to determine x1 and x3  */
        xTOL,
        fTOL;      /* stopping criteria */
  int   nLIM;      /* maximum number of iterations allowed */

  float x1, x2, x3,
        f1, f2, f3,
        h1, h2,          /* x2 - x1,  and x3 - x2 respectively  */
        a, b, c,         /* the coefficients of the quadratic equation  */
        discriminant,    /* the discriminant of the quadratic */
        g,               /* the ratio h1/h2  */
        fr,              /*  f(xr)           */
        delx;
  int   j;

/*
 ------------------------------------------------------------------
   Evaluate the function f(x) = EXP(x) - 3x
*/

float f(x)
  float x;
{
   return(3.0*x + sin(x) - exp(x));
}       /* end of f */
/*
 ------------------------------------------------------------------
      Set initial values into the x's and the corresponding
      function values.
*/

setINITIALvalues()
{
   x1 = xr - h;   x2 = xr;     x3 = xr + h;
```

```
   f1 = f(x1);    f2 = f(x2);  f3 = f(x3);
}
/*
---------------------------------------------------------------------
   Compute the values needed for the Quadratic Equation.
   a*v*v + b*v + c = 0
*/

compute_values_for_QUADRATIC()
{
   h1 = x2 - x1;   h2 = x3 - x2;   g = h1/h2;
   a = ( f3*g - f2*(1.0 + g) + f1) / (h1*(h1 + h2));
   b = ( f3 - f2 - a*h2*h2 ) / h2;
   c = f2;

   discriminant = sqrt(b*b - 4.0*a*c);
   if (b < 0.0)
     discriminant = -discriminant;
}
/*
---------------------------------------------------------------------
   Find the root of the quadratic: a*v*v + b*v + c = 0
*/

float rootOFequation()
{
   return(-2.0 * c / (b + discriminant));
}
/*
---------------------------------------------------------------------
   Check STOPPING criteria on xTOL and fTOL
*/
int fTOL_check()
{
   if (fabs(fr) < fTOL)
     return(1);
   else
     return(0);
}
int xTOL_check()
{
   if (fabs(delx) < xTOL)
     return(1);
   else
     return(0);
}
/*
---------------------------------------------------------------------
   Select the THREE points for the next iteration. When delx > 0, choose
   x2, x3, and xr, BUT when delx < 0 choose x1,x2,xr.

   We enter the set so that they are in ascending order.
*/

select_next_three_points()
```

```
{
   if (delx > 0.0)
     {
      x1 = x2; f1 = f2;
      if (delx > h2)
        {
         x2 = x3; f2 = f3;
         x3 = xr; f3 = fr;
        }
      else
        {
         x2 = xr; f2 = fr;
        }
     }
   else
     {
  x3 = x2; f3 = f2;
      if (fabs(delx) > h1)
        {
         x2 = x1; f2 = f1;
         x1 = xr; f1 = fr;
        }
      else
        {
         x2 = xr; f2 = fr;
        }
     }
}

/*
---------------------------------------------------------------------
   ********************Output procedures**********************
*/

          /* Print if fTOL or xTOL conditions met */
print_fTOL_condition_met()
{
   printf("\n\n");
   printf(" fTOL condition met in %d iterations.\n ",j);
   printf("   x =  %.5f  and f(xr) =  %.5f\n", xr, fr);
   printf("\n\n");
}
print_xTOL_condition_met()
{
   printf("\n\n");
   printf(" xTOL condition met in %d iterations.\n", j);
   printf("   x =  %.5f  and f(xr) =  %.5f\n", xr, fr);
   printf("\n\n");
}
          /* Print the current values: x versus f(x) */
print_current_values()
{
   printf(" At iteration %d,  x =  %.5f,  f(x) =  %.5f\n", j, xr, fr);
   printf("\n");
}
```

```
            /* nLIM iterations and NO convergence */
print_NO_convergence()
{
   printf("\n");
   printf(" No convergence after %d iterations.\n", nLIM);
   printf(" x = %.5f, f(x) = %.5f\n\n", xr, fr);
}

/*
-----------------------------------------------------------------
   This procedure uses MULLER'S method to find a root of f(x) = 0.
   The procedure used here can only find a real root of the function
   entered.

   xr - on INPUT the initial guess
        on OUTPUT the solution

   h  - the original step size to give us the three points for the
        quadratic interpolation.
x1 = xr - h; x2 = xr; x3 = xr + h;
        As the process continues, the h will change.

*/

muller(xr, h, xTOL, fTOL, nLIM)
 float *xr, h, xTOL, fTOL;
 int   nLIM;

{
      /* *****PROCEDURE   MULLER*****  */

   setINITIALvalues();

   j = 1;

   while (j < nLIM)
     {
      compute_values_for_QUADRATIC();
      delx = rootOFequation();
      *xr = x2 + delx;   fr = f(*xr);

      print_current_values();

      if (fTOL_check())
        {
           print_fTOL_condition_met();
           j = j + nLIM;
        }
      else
       if (xTOL_check())
        {
           print_xTOL_condition_met();
           j = j + nLIM;
        }
```

```
      select_next_three_points();
      j = j + 1;
      }

   if (j ==   nLIM)
     print_NO_convergence();
}        /* PROCEDURE   MULLER   */

main()     /* MAIN  */
{
   xr = 0.5;  h = 0.2;  xTOL = 0.0001;  fTOL = 0.00001;  nLIM = 50;
   muller(&xr,  h, xTOL,   fTOL,   nLIM);
}

   initializeDATA();
   newton(&x,  ftol,  nlim,  &i);
}                    /* END  OF  MAIN  */
```

برنامه ۲.۲ روش بیرستو

در این قسمت برنامه پاسکال را برای اجرای روش بیرستو بکار می‌بریم و یک چند جمله‌ای را به چند عامل درجه دو تفکیک می‌کنیم.

در این برنامه ده برنامه فرعی وجود دارد که قبل از برنامه اصلی قرار دارند.

یک خصوصیت زبان از پاسکال که خطای برنامه‌نویسی را کاهش می‌دهد، لزوم معرفی کلیه متغیرها قبل از استفاده از آنها می‌باشد.

متغیرهای سراسری برای برنامه بیرستو در زیر خلاصه شده‌اند:

a	یک رشته که ضرایب چند جمله‌ای اصلی را ذخیره می‌کند. فضا برای حداکثر 12 ضریب می‌باشد.
b	این رشته وقتی چند جمله‌ای (با ضرایب حاصل از a) به یک عامل درجه دوم آزمایشی تقسیم می‌شود. ضرایب چند جمله‌ای جدید را ذخیره می‌کند.
c	این رشته که مشتقات جزئی را ذخیره می‌کند وقتی چند جمله‌ای با ضرایب حاصل از b به یک عامل درجه دوم آزمایشی تقسیم می‌شود ضرایب چند جمله‌ای جدید است.
r و s	پارامترها در عامل آزمایشی می‌باشند $x^2 - rx - s$
$delr$, $dels$	نمو، برای بهبود r و s
tol	معیار توقف یا خطای قابل اغماض
$denom$	متغیری که در محاسبه $delr$ و $dels$ استفاده می‌شود.
n	درجه چند جمله‌ای
$nplus2$, $i/j/h$	متغیرهای کنترل حلقه‌ها
$continue$	یک متغیر منطقی برای کنترل حلقه‌ها

برنامه

در اینجا یک لیست از برنامه‌های فرعی در *BRSTOW* آورده می‌شود:

initilize DATA: ضرایب چند جمله‌ای (رشته a) و مجموعه مقادیر شروع برای r و s را معین می‌کند. برنامه برای چند جمله‌ای درجه پنج زیر انجام می‌شود.

این برنامه مقادیر اولیه b و c را تعیین می‌کند.

print HEADING: یک تیتر می‌نویسد و چند جمله‌ای اصلی را نشان می‌دهد.

compurterbandcarrays	محاسبات b و c را انجام می‌دهد.
:FindRands	محاسبات $delr$ و $dels$ و تغییر r , s را انجام می‌دهد.
:tolMET	می‌بینید که آیا به معیار دقت رسیده است.
:ChangeRandSandk	مجدداً به r و s و k مقدار اولیه می‌دهد تا عامل بعدی بدست آید.
:Print QUADRATIC equation	عامل درجه دوم را نمایش می‌دهد.
:Vesuee POLYMOMIAL	مقادیر b را جایگزین a می‌کند تا ضرایب چند جمله‌ای جدید بدست آید.
:chock TOLMET	نمایش یک عامل وقتی عملیات همگرا می‌شود و مقادیر جدید r و s بدست می‌آیند

این برنامه پاسکال به وضوح نشان می‌دهد چطور یک برنامه ساخت یافته بدست می‌آید و فواید آن چیست. برنامه‌های فرعی بکار رفته، در برنامه اصلی صدا زده می‌شوند. خروجی برنامه در پایان لیست برنامه نشان داده شده است.

برای استفاده برنامه *BRSTOW* با چند جمله‌ای‌های مختلف، فقط مقدار داده اولیه در برنامه فرعی *initialize DATA* تغییر خواهد کرد. با تغییر این برنامه مقادیر اولیه هربار خوانده می‌شود.

```
          PROGRAM   BRSTOW(INPUT,OUTPUT);
(*
          APPLIED  NUMERICAL   ANALYSIS

-----------------------------------------------------------------
  This program uses Bairstow's Method of extracting the
quadratic factors of a polynomial.
*)
Const
  tol    =  0.005;
  limit  =  40;
Var
  a,                    (* The original polynomial *)
  b, c:                 (* Working polynomials *)
                Array[0..12] OF Real;
  r,s,                  (* Estimates for quadratic factor *)
  delr, dels,           (* Changes in r and s from previous
*)
  denom:          Real;

  n,                    (* Degree of the given Polynomial, a
*)
  nPLUS2, i,j,k:     Integer;
  f:                TEXT;
  continue:          BOOLEAN;

Procedure initializeDATA;
(*
-----------------------------------------------------------------
  Here we enter the given matrix in descending order as well as
  as well as initializing the working variables.
*)
Var
  i :   Integer;
Begin
  n := 5; nPLUS2 := n + 2;
  a[2] := 1.0; a[3] := -17.8;
  a[4] := 99.41; a[5] :=  -261.218;
  a[6] :=  352.611;  a[7] := -134.105;

  r := 0.0; s := 0.0;
(*
  Set up B and C arrays
*)
  FOR i := 0 TO 12 Do Begin
    b[i] := 0.0;
    c[i] := 0.0     End;
End;  (* initializeDATA  *)

Procedure printHEADING;
```

```
(*
  ---------------------------------------------------------
    Print out a heading and the coefficients of the
    given polynomial.
*)
Var
   i : Integer;
Begin
   Assign(f, 'Bairstow.out');
   ReWrite(f);
     WriteLn(f); WriteLn(f,' QUADRATIC FACTORS BY BAIRSTOW
METHOD. ');
   WriteLn(f); WriteLn(f);
   WriteLn(f,' THE ORIGINAL POLYNOMIAL IS: ');
   WriteLn(f); WriteLn(f,' POWER OF X ', ':5, ' COEFFICIENT');

   FOR i := 2 TO nPLUS2 Do
      WriteLn(f, (nPLUS2-i):6, ':10, a[i]:10:3);
   WriteLn(f); WriteLn(f);
   WriteLn(f,'*****THE FACTORS OF THE POLYNOMIAL ARE:*****');
   WriteLn(f)
End;  (* printHEADING  *)

Procedure computeBandCarrays;
(*
  ---------------------------------------------------------
    We compute the B and C arrays which are needed to
    find the Quadratic factor of the polynomial stored
    in the vector, A.
*)
Var
   j : Integer;
Begin
   FOR j := 2 TO nPLUS2 Do Begin
      b[j] := a[j] + r*b[j-1] + s*b[j-2];
      c[j] := b[j] + r*c[j-1] + s*c[j-2]  End
End;  (* computeBandCarrays  *)

Procedure findRandS;
(*
  ---------------------------------------------------------
    R and S are the coefficients of the Quadratic equation we
    are trying to factor from the given polynomial.
*)
Begin
   delR := (-b[n+1]*c[n] + b[n+2]*c[n-1])/denom;
   delS := (-c[n]*b[n+2] + c[n+1]*b[n+1])/denom;
   r := r + delR;
   s := s + delS
End;   (* findRandS  *)

FUNCTION tolMET: BOOLEAN;
```

```
(*
  ---------------------------------------------------------
    Check whether the accuracy condition is met.
  *)
  Begin
    tolMET := (ABS(delR) + ABS(delS)) < tol;
  End; (* tolMET *)

  Procedure changeRandSandK;
  (*
  ---------------------------------------------------------
    Change R and S values and start over when the denominator
    is zero.
  *)
  Begin
    r := r + 1.0; s := s + 1.0; k := 1
  End; (* changeRandSandK *)

  Procedure printLINEARequation;
  (*
  ---------------------------------------------------------
    Print a linear equation.
  *)
  Begin
    WriteLn(f,b[n+1]:11:5, ' X + ', b[n+2]:10:5);
    continue := FALSE
  End; (* printLINEARequation *)

  Procedure printQUADRATICequation;
  (*
  ---------------------------------------------------------
    Print a quadratic equation;
  *)
  Begin
    WriteLn(f, b[n]:11:5,'X**2 + ', b[n+1]:10:5,' + ', b[n+2]:10:5);
    continue := FALSE
  End; (* printQUADRATICequation *)

  Procedure reducePOLYNOMIAL;
  (*
  ---------------------------------------------------------
    Get new polynomial after the quadratic equation
    has been factored out.
  *)
  Var
    i : Integer;
  Begin
    nPLUS2 := n + 2; k := 1;
    FOR i := 2 TO nPLUS2 Do a[i] := b[i]
  End; (* reducePOLYNOMIAL *)

  Procedure checkTOLMET;
```

```
(*
  If the tolerance condition is met we can factor out
  the quadratic equation and check on the degree of the
  remaining polynomial.
*)
Begin
  IF tolMET THEN                                    Begin
    WriteLn(f,'    X**2 - ', r:10:5, ' X - ', s:10:5);
    n := n-2;

    CASE n OF                          (* Reduced Equation is of
*)
     1    : printLINEARequation;       (*      Degree ONE
*)
     2    : printQUADRATICequation;    (*      Degree TWO
*)
     3..12 : reducePOLYNOMIAL          (*      Degree > TWO
*)
                                  End (* CASE *) End
  Else (* not tolMET *)
    k := k + 1
End; (* checkTOLMET *)

Begin (* MAIN *)
  initializeDATA;
  printHEADING;
  k := 1;
  continue := (k < limit);
  WHILE continue Do                              Begin (* WHILE *)
    computeBandCarrays;
    denom := c[n]*c[n] - c[n+1]*c[n-1];
    IF (denom <> 0.0) THEN       Begin (* denom IF *)
      findRandS;
      checkTOLMET                End (* denom <> 0 *)

    ELSE (* denom = 0 *)
      changeRandSandK;           (* denom IF *)

  continue := (k < limit ) AND continue       End;    (* WHILE *)
  WriteLn(f);
  Close(f)
End.
```

برنامه ۳.۲روش نیوتن

لیست برنامه به نام *PNEWTN.C* به زبان *C* که برای حل معادلهٔ $f(x) = 3x + sin(x) - exp(x) = 0$ آورده شده است.

```
/*
   ******************************
   **                          **
   **        PNEWTN.C          **
   **                          **
   ******************************

 APPLIED  NUMERICAL  ANALYSIS
 ------------------------------------------------------------------------
   This program uses the Newton's method to find the root of a given
   function, f(x) = 0 starting from a suitable initial guess.

   Solve f(x) = 3*x + sin(x) - exp(x) = 0.
*/
#include  <stdio.h>
#include  <math.h>

 float x, ftol;
 int   i, nlim;

/*
 ------------------------------------------------------------------
   FUNCTION  fcn COMPUTES  THE  VALUES  FOR F(X).
*/
float fcn(x)
 float x;

{
   return(3.0*x  +  sin(x)  -  exp(x));
}           /* end of fcn */
/*
 ------------------------------------------------------------------
   FUNCTION  fcnPRIME  EVALUATES  THE DERIVATIVE  F'(X).
*/
float fcnPRIME(x)
 float x;
{
   return(3.0  +  cos(x)  -  exp(x));
}          /* end of fcnPRIME  */

/*
   PROCEDURE  NEWTON
*/

newton(x,  ftol,  nlim,  i)
 float  *x, ftol;
 int    *i, nlim;
```

```
/*
-----------------------------------------------------------------
   x    : initial value of X. It should be near the ROOT.
          On OUTPUT  x returns the value of the root.
   ftol : stopping criteria on the x's and f's respectively.
   nlim : stopping criterion. Absolute limit on iterations.
   i : indicates how the procedure terminated.
      i =  2  =>  ftol criterion met.
      i = -1  =>  nlim exceded.
*/
{
   float fx, delx;
   int   counter;              /* KEEPS TRACK OF ITERATIONS   */
   int   CONTINUE;             /* SEARCH COMPLETED OR NOT     */

   fx = fcn(*x);
   counter = 1;
   CONTINUE  = 1;
   while (CONTINUE)                    /* WHILE  */
     {
      delx = fx/fcnPRIME(*x);
      *x = *x - delx;                  /* FIND NEW X */
      fx = fcn(*x);
  printf(" AT ITERATION    %d   X =   %10.5f   F(X) =  %10.5f\n",
            counter,  *x, fx);

      if (fabs(fx) < ftol)             /* IF FABS(fx)  */
/*
-----------------------------------------------------------------
      SOLUTION FOUND. PRINT RESULTS AND RETURN TO MAIN
                  PROGRAM.
*/
        {
         *i = 2;

         printf("\n\n");
         printf("\t\t\t**********\n\n");
          printf(" F TOLERANCE  MET IN %d ITERATIONS:   X = %10.7f
F(X) =  %10.7f\n",
               counter,  *x, fx);
         printf("\n");

         CONTINUE  = 0;
        }

     else                             /* ELSE  */
/*
-----------------------------------------------------------------
        SOLUTION NOT FOUND. CONTINUE SEARCH USING THE
SECANT
      METHOD. INCREASE COUNTER BY ONE.
*/
```

```
      {
       counter++;
       if (CONTINUE   && (counter  < nlim))
         CONTINUE   = 1;
       else
         CONTINUE   = 0;
      }                          /* END  OF  ELSE  */

   }                             /* END  OF  WHILE  */

/*
-------------------------------------------------------------------
      CHECK  TO  SEE  IF  NLIM  WAS  EXCEEDED.
*/
  if (counter  >=  nlim)
    {
     *i  =  -1;
     printf("\n");
       printf(" TOLERANCE   NOT  MET  AFTER  %d  ITERATIONS:   X =
%.5f F(X)  =  %.6f\n",
            nlim,  *x, fx);
     printf("\n");
    }                        /* IF  COUNTER   >  NLIM  */

}                            /* END  OF  NEWTON   PROCEDURE   */

/*
-------------------------------------------------------------------
      SET  UP  INITIAL   VALUES
*/

initializeDATA()
{
  x  =  0.5;                 /* INITIAL   GUESSES  */
  ftol  =  0.00001; nlim  =  50;    /* STOPPING  CRITERIA   */
}
main()              /* MAIN  */
{
  printf("\n\n");
     printf(" SOLUTION   OF  F(X)  =  0  USING  THE  NEWTON'S
METHOD.\n\n");
```

برنامه ۲.۴روش تکرار تابعی

لیست برنامه به نام *PXGXIT.PAS* به زبان پاسکال که برای حل معادلهٔ $f(x) = 3x + sin(x) - exp(x) = 0$ آورده شده است.

```
Program pxgxit(INPUT,OUTPUT);
(*$C+,U+*)
(*
 APPLIED NUMERICAL ANALYSIS
-----------------------------------------------------------------
 This program solves f(x) = 0, by solving g(x) = x by iteration.
 In this example the function f(x) = 3x + sin(x) - exp(x) = 0 is
 changed to g(x) = x = ( exp(x) - sin(x) )/ 3 with a
 starting value of x = 0.5.
*)
Var
   x,       (* on input the original guess For the root,
               on output the solution to f(x) = 0 *)
   xTOL     (* stopping criterion: when ABS(g(x) - x) < xTOL *)
        : Real;
   nLIM     (* limit to the number of iterations *)
        : Integer;
(*
-----------------------------------------------------------------
 Define the function g(x) = ( exp(x) - x )/ 2
*)
FUNCTION g(x : Real ): Real;
Begin
    g := ( EXP(x) - SIN(x) ) / 3.0
End;
(*
-----------------------------------------------------------------
--------
   The Procedure that iterates on g(x) until it converges to the
solution
   or diverges.
*)

Procedure xgxit(Var x : Real; xTOL : Real; nLIM : Integer );
Var
  j : Integer;
  xSAVE,
  delta1, delta2 : Real;
(*
-----------------------------------------------------------------
   ******************Output Procedures*********************
   ABS(x - g(x)) seem to diverge
*)
Procedure printERRORmessage;    (* iterates seem to diverge *)
Begin
   WriteLn; WriteLn;
```

```
   WriteLn('  The last three values  indicate DIVERGENCE.');
   WriteLn;  WriteLn
End;

     (* No convergence after nLIM iterations *)
Procedure print_nLIM_EXCEEDED;
Begin
   WriteLn;  WriteLn;
   WriteLn('  Method did NOT converge in ', nLIM:2,  ' iterations.');
   WriteLn('  Final values are x = ', xSAVE:12:5,  ' g(x) = ',
           x:12:5);
   WriteLn;  WriteLn
End;

     (* Print the current x versus g(x)  *)
Procedure printCURRENTanswer;
Begin
   WriteLn('  At iteration', j:3,',  x = ', xSAVE:7:4,
           ', g(x) = ',x:7:4)
End;

     (* Convergence has been attained. Print FINAL result  *)
Procedure printFINALanswer;
Begin
   WriteLn;
   WriteLn('  xTOL  MET  IN ', j:2,' iterations.  x = ', xSAVE:8:5,
           '  g(x) = ', x:8:5)
End;
Begin     (* pxgxit *)
   xSAVE  := x;   x := g(x);  j := 1; delta1 := ABS(x  - xSAVE);

   WHILE  ( j < nLIM ) AND  (delta1 > xTOL)  DO Begin
     printCURRENTanswer;
     j := j + 1;  xSAVE  := x;  x := g(x);  delta2 := ABS(x  -
xSAVE);
     IF delta2 > delta1  THEN
        Begin  printERRORmessage;    j := j + nLIM   End

        ELSE    delta1 := delta2                End;

  IF j < nLIM  THEN  printFINALanswer
     ELSE  IF j = nLIM  THEN  print_nLIM_EXCEEDED
        ELSE
          printERRORmessage
End;

Begin  (* PXGXIT  *)
   x := 0.5;   nLIM  := 20;  xTOL  := 0.0001;
   xgxit(x, xTOL,  nLIM  )
End.
```

توضیح:

بدون انجام تمرینات متعدد یادگیری و تجزیه و تحلیل محاسبات عددی غیر ممکن است. برای شما در این کتاب شرایطی فراهم آورده‌ایم تا بتوانید با انجام تمرینات متعدد این موضوع را فرا بگیرید. روش ما شامل دو مرحله است. یک نوع آن عبارت است از تمریناتی که مشابه آنها به شما داده می‌شود تا با ماشین حساب محاسبه کنید این کار سبب می‌شود تا بتوانید به راحتی مسائل مشکل‌تر را حل کنید. بهتر است خودتان برنامه‌هایی برای حل مسائل بنویسید که خود حاکی از فهم کامل شما نسبت به الگوریتم است. نوشتن یک برنامه مشابه با برنامه‌های داده شده تمرین خوبی است، و بهتر است از آن برنامه‌ها به زبان *FORTRAN*، *PASCAL*، *BASIC*، *C* یک نسخه داشته باشید. شما می‌توانید با استفاده از برنامه‌های موجود در کتابخانه‌های استاندارد مانند *IMSL* تجربیات زیادی کسب کنید. یا از بسته‌های نرم افزاری علمی *MATHEMATICA*، *MATALB* یا *MAPLE* استفاده کنید. بستگی به علاقه شما و نظر استاد درس می‌توانید از هر یک از این روشها یا نرم‌افزارها که بخواهید استفاده کنید.

تمرینات فصل دوم

۱- معادله $e^x - 3x$ دارای یک ریشه $r=0.61906129$ است. در دامنه $(1$ و $0)$ با 6 بار تکرار به روش نصف کردن این ریشه را بیابید. چند تکرار برای پیدا کردن ریشه تا چهار رقم معنی‌دار نیاز است؟ یعنی $|x-r| < .00005$ برای هشت رقم معنی‌دار چندبار تکرار لازم است.

جواب: بعد از 6 بار تکرار جواب برابر $x_3=.609375$ می‌باشد. بعد از $n=15$ بار تکرار جواب با 5 رقم با معنی صحیح بدست می‌آید. و برای 8 رقم دقت باید $n=28$ بار تکرار انجام شود.

۲- معادله درجه دو $(x-0.6)(x-0.4) = x^2-x+0.24$ دارای دو ریشه $(x=0.4)$ و $(x=0.6)$ می‌باشد. تحقیق کنید که دو کران دامنه $(1$ و $0)$ برای روش نصف کردن مناسب نیست. تابع را رسم کنید و کران‌های موجود را طوری تغییر دهید که با استفاده از روش نصف کردن بتوان آنها را همگرا کرد. فرض کنید که دامنه $(1$ و $0.5)$ باشد. بعد از پنج تکرار چقدر خطا خواهیم داشت و بعد از پنج تکرار، مقدار خطای واقعی را حساب کنید.

جواب: فواصل برابر $[0.6, -\infty]$ و $[0.4, +\infty]$ می‌باشند. از فاصله $[0.5, 1.0]$ بعد از 5 تکرار $x=0.60937$ و خطا برابر $(\frac{0.5}{2})^5 = 0.015625$ می‌باشد. خطای واقعی برابر -0.009375 می‌باشد.

۳- روش نصف کردن برای هر تابع پیوسته‌ای بکار می‌رود و فقط مختص چند جمله‌ای‌ها نیست تحقیق کنید که از تقاطع $y=L_n(x)$ و $y=x-2$ ریشه $L_n x-x+2 = 0$ تا ۴ رقم با معنی صحیح حاصل می‌شود.

جواب: محل تقاطع در $x=0.1586$ و $x = 3.1462$ می‌باشند.

۴- با روش نصف کردن کوچکترین ریشه مثبت معادلات زیر را بدست آورید. در هر مورد ابتدا یک دامنه مناسب پیدا کنید. سپس ریشه را با دقت نسبی 0.005 محاسبه نمایید.

$$e^x - x - 2 = 0 \quad (a$$

$$x^3 - x^2 - 2x + 1 = 0 \quad (b$$

$$2e^{-x} - \sin(x) = 0 \quad (c$$

$$x^3 + 4x^2 - 7x + 1 = 0 \quad (d$$

جواب:

$$1.146 \quad (a$$

$$0.445 \quad (b$$

$$0.9210 \quad (c$$

$$1.175 \quad (d$$

۵- چند جمله‌ای $x^3-x^2-3x-3=0$ به عنوان مثالی در بخش‌های قبلی آمده است که دارای ریشه تقریبی $x=\sqrt{3}$ می‌باشد همچنین دارای دو ریشه $x=-1$ و $x=-\sqrt{3}$ است. با دو مقدار مناسب که $x=-\sqrt{3}$ را محدود می‌کند شروع کنید. نشان دهید که روش درون یابی خطی به ریشه همگراست.

جواب: چون $f(-2)f(1.5)<0$ می‌باشد. طبق قضیه بولزانو تابع در فاصله $[-2,1.5]$ حداقل دارای یک ریشه است. از تکرارهای متوالی روش درون یابی خطی نتیجه می‌شود:

$$-1.63636$$

$$-1.69825$$

$$-1.72085$$

$$-1.72842$$

$$-1.73088$$

$$\vdots$$

$$-1.73205$$

۶- در تمرین ۵ اگر بخواهیم با مقادیر $x = -1.5$ و $x = -1.7$ شروع کنیم، تابع تغییر علامت نخواهد داد. بنابراین این دو مقدار برای روش درون یابی خطی مناسب نیستند. با این حال در روش «وتر» می‌توان با این مقادیر شروع نمود. از این ریشه‌ها به روش وتر شروع کنید. چند تکرار برای تخمین تا چهار رقم اعشار لازم است؟ فرض کنید. مقادیر آغازی -1.5 و -1.1 باشند. کدام ریشه با روش وتر بدست می‌آید؟ اگر با -1.5 و -1.25 شروع کنیم چه ریشه‌ای بدست می‌آید؟

جواب: بعد از سه تکرار $x_3=1.73201$ حاصل می‌شود.
با استفاده از فاصله $[-1.5, -1.1]$ ریشه -1 بدست می‌آید.
فاصله $[-1.5, -1.25]$ همان ریشه را می‌دهد.

۷- تحقیق کنید معادله درجه سوم $y=x^3-x+1$ در کجا سهمی $y=2x^2$ را قطع می‌کند. دو منحنی رسم کنید که محل تقاطع را نشان دهد. سپس با استفاده از روش درون یابی خطی مقادیر x را به روش قدم به قدم و نقطه‌یابی پیدا کنید.
جواب: محل تقاطع در $x=-0.80194$ و $x=0.55496$ و $x=2.24697$ می‌باشند.

۸- از روش درون یابی خطی برای حل معادلات تمرین استفاده کنید.
قسمت a و b را در مورد همگرایی مقایسه کنید.

	فاصله شروع	تعداد تکرار برای درون یابی خطی	تعداد تکرار برای درون یابی اصلاح شده
a)	*(1.0,1.2)*	*4*	*4*
	(1,2)	*14*	*4*
b)	*(0.4,0.6)*	*3*	*4*
	(0,1)	*4*	*4*
c)	*(0.8,1.0)*	*4*	*4*
	(1,2)	*5*	*5*
d)	*(1.0,1.2)*	*4*	*4*
	(1,2)	*19*	*7*
	(0,2)	*25*	*11*

۹- یک آلگوریتم برای روش درون یابی خطی بنویسید.
x_1 و x_2 را نزدیک به ریشه انتخاب می‌کنیم و یک مقدار برای خطای قابل اغماض مقدار تابع در نظر می‌گیریم. (ε) مقدار نهایی $x3$ ریشه تقریبی را می‌دهد. اما همگرائی تضمین نمی‌شود. مگر آنکه قضیه بولزانو مورد استفاده قرار گیرد و در هر بار تکرار $f(x_1).f(x_2)<0$ باشد.

```
compute f(x1) and f(x2).
Repeat
    Let x3 = x2 - f(x2)(x2 - x1)/(f(x2) - f(x1))
  compute f(x3)
x3 = x2
    if abs(f(x1)) > abs(f(x2) then x2 = x1
Until f(x3) < tolerance value.
```

۱۰ - ریشه نزدیک به $x=-.5$ مربوط به $e^x-3x^2=0$ را به روش نیوتن تا دقت 6 رقم پیدا کنید.

جواب: $x=-0.458962$

۱۱ - معادله $e^x-3x^2=0$ نه تنها دارای یک ریشه نزدیک به $x=.5$ است بلکه یک ریشه نزدیک به $x=4$ نیز دارد. با استفاده از روش نیوتن ریشه مثبت را بیابید.

جواب: $x=3.73307$

۱۲ - با استفاده از روش نیوتن یکی از معادلات تمرین ۴ را حل کنید. برای رسیدن به دقت مورد نظر چندبار تکرار لازم است؟

جواب: a) با $x_0=1.000$ ریشه بعد از دو تکرار برابر 1.146 می‌باشد.

b) با $x_0=0.5$ ریشه بعد از یک تکرار برابر 0.444 می‌باشد.

c) با $x_0=1.0$ ریشه بعد از دو تکرار برابر 0.921 می‌باشد.

d) با $x_0=1.0$ ریشه بعد از دو تکرار برابر 0.178 می‌باشد.

۱۳- الف) از روش نیوتن برای حل معادله $x^2=N$ استفاده کنید و به یک آلگوریتم برای ریشه دوم N برسید. $x_{i+1}=\frac{1}{2}(x_i+N/x_i)$ که در آن x_0 یک تقریب اولیه برای $\sqrt{N}$ می‌باشد.

ب) از فرمول مشابهی برای ریشه سوم و چهارم و nام استفاده کنید.

جواب: الف با فرض $f(x)=x^2-N$ داریم:

$$x_{n+1}=x_n-(x^2-N)/\ 2x_n\ =\ (x_n+N/x_n)/\ 2$$

ب - با فرض $f(x)=x^k-N$ داریم:

$$x_{n+1}=\frac{1}{k}((k-1)x_n+N/x_n^{k-1})$$

(شرط همگرایی را بنویسید).

۱۴ - اگر آلگوریتم تمرین قبل را دوبار بکار بریم نشان دهید که

$$N^{1/2} \approx \frac{(A\ +\ B)}{4} + \frac{N}{(A\ +\ B)}$$

که در آن $N=A*B$

جواب: با فرض $x_1=A$ داریم $B=\frac{N}{x_1}$ و $x_2=(A+B)/\ 2$ یا $x_2=(x_1+\frac{N}{x_1})/\ 2$ و در نتیجه داریم: $x_3=\frac{1}{2}((A+B)/\ 2+2N/(A+B))=(A+B)/\ 4+N\ /(A+B)$

۱۵ - نشان دهید که خطای نسبی مسأله قبل تقریباً برابر است با

$$\frac{1}{8}(\frac{A-B}{A+B})^4$$

۱۶ - $f(x)\ =\ (x-1)^3(x-2)$ دارای یک ریشه $x=2$ و ریشه $x=1$ می‌باشد. از $x=2.1$ شروع

نموده و یک بار با استفاده از روش نیوتن سرعت همگرائی را بررسی کنید. سپس با $x=0.9$ شروع کنید و توجه کنید که چقدر همگرایی به ریشه دیگر کند می‌شود در حالی که خطای شروع در هر دو مرحله فقط 0.1 است.

جواب: با $x_0=2.1$ بعد از یک تکرار برابر $x_0=2.02$ و بعد از سه تکرار خطا کمتر از $2E-6$ می‌گردد.

با $x_0=0.9$ بعد از یک تکرار برابر $x_1=0.932$ و بعد از 29 تکرار خطا کمتر از $2E-6$ خواهد شد.

۱۷- از روش مولر برای حل معادلات زیر استفاده کنید. چهار تکرار انجام دهید و تشخیص دهید که چگونه خطاهای متوالی به همدیگر مربوط هستند. از یک مقدار معین شروع کنید و با اختلاف 0.2 تکرار کنید.

الف) ریشه نزدیک 1 (مقدار تحقیقی 1.07816259) $2x^3+4x^2-2x-5=0$

ب) ریشه نزدیک 4 (مقدار تحقیقی 3.73307903) $e^x-3x^2=0$

ج) ریشه نزدیک 1 $e^x-3x^2=0$

د) ریشه نزدیک 1.1 $\tan(x)-x-1=0$

جواب: دقت مضاعف را به کار می‌بریم و نتایج را به دست می‌آوریم.

الف: با اعداد $(0.8, 1.0, 1.2)$ شروع می‌کنیم و تکرارها برابرند با:

$1.07777432, 1.07816216, 1.07816256, 1.07816280$

و خطا برابر $2.12E-7$ می‌باشد.

ب) با اعداد $(3.4, 4.0, 4.2)$ شروع می‌کنیم و تکرارها برابرند با:

$3.72291424, 3.73324320, 3.7330784, 3.7330784$

و خطا برابر $6.3E-7$ می‌باشد.

ج) با اعداد $(0.8, 1.0, 1.2)$ شروع می‌کنیم و تکرارها برابرند با:

$0.9095711, 0.91000832, 0.91000800, 0.91000794$

د) با اعداد $(1.0, 1.2, 1.41)$ شروع می‌کنیم و تکرارها برابرند با:

$1.15555216, 1.13339224, 1.1322832, 1.3226784$

۱۸- روش مولر دارای یک فرم «خود شروع» است. در این روش به جای آنکه مقدار اولیه را با یک مقدار نزدیک به ریشه شروع کنیم، الگوریتم را به طور دلخواه با $x_0=0$ و $x_1=0.5$ و $x_2=-0.5$ شروع می‌کنیم. از این مقادیر اولیه استفاده کنید و معادلات مسأله ۱۷ را حل کنید. با این فن می‌توان نزدیکترین ریشه به مبداء را پیدا کرد. آیا این بیان همواره صحیح است؟

جواب:

الف) ریشه 1.315449- به دست می‌آید، که نزدیکترین ریشه به مبدأ نیست.

ب. ج) ریشه $0.458962-$ به دست می‌آید، که نزدیکترین ریشه به مبدأ می‌باشد.
د) نزدیکترین ریشه به مبدأ 1.3226 بعد از 15 بار تکرار با خطای کمتر از $1E-4$ به دست می‌آید.

۱۹ - تعمیم روش «خود شروع» در تمرین ۱۸ عبارت است از کاهش تابع بعد از آنکه اولین ریشه پیدا شد. *(deflating)* یعنی درست کردن یک تابع جدید که دارای همان ریشه‌های معادله اصلی است. به استثنای ریشه $x=r$ که این تابع جدید در تقسیم $f(x)$ بر $(x-r)$ بدست می‌آید. این روش را با مقایسه رسم $f(x)=x(x-3)(x-1)$ و $g(x)=x(x-3)$ و $j(x)=(x-1)(x-3)$ $f(x)=x(x-1)$ ، بکار برید از این روش برای پیدا کردن ریشه‌های معادلات در تمرین ۱۷ استفاده کنید و مانند تمرین ۱۸ شروع کنید.
جواب: الف) سایر ریشه‌ها برابر $1.31545-$ و $1.76271-$ می‌باشند.
ب. ج) ریشه سوم برابر $0.458962-$ می‌باشد.
د) تعداد زیادی ریشه وجود دارند: $4.533,\ -4.286,\ -7.706,$.

۲۰- از آنجایی که روش مولر مقادیر تابع را که ممکن است خیلی بهم نزدیک باشند از هم کم می‌کند احتمال دارد که خطای نسبی بزرگی بوجود آید.
در بعضی از معادلات تمرین ۱۷ از مقادیری که خیلی به همدیگر نزدیک باشند (اما به ریشه نزدیک نیست) استفاده می‌شوند.
پیچیدگی زمانی مشاهده می‌شود که $f(x)$ خیلی آهسته به ریشه نزدیک می‌شود.
تأثیر این مورد در تابع چند جمله‌ای که دارای ریشه تکراری است رخ می‌دهد.
جواب: مانند $f(x)=(x-1)^3$ که نشان می‌دهد تقسیم بر صفر بوجود می‌آید، قبل از آنکه به ریشه رسیده باشیم. (جواب 1.00567 با دقت مضاعف).

۲۱- اگر از چهار عمل اصلی مختلط برای محاسبه ریشه‌های مختلط استفاده شود، روش مولر مناسب است. ریشه مختلط $x^4+4x^3+21x^2+4x+20=0$ را پیدا کنید. روش نیوتن می‌تواند برای پیدا کردن ریشه مختلط مورد استفاده قرار گیرد. می‌توانید روشها را مقایسه کنید. محاسبه ریاضی با دست کار زیادی می‌برد ممکن است مجبور شوید در حل این مسئله از برنامه کامپیوتری استفاده کنید.
جواب: ریشه‌ها برابر $0 \pm i,\ -2 \pm 4i$ می‌باشند.

۲۲- معادله $f(x)=e^x-3x^2=0$ دارای سه ریشه است. یکی از حالت‌های واضح این معادله $x=\pm\sqrt{e^x/3}$ می‌باشد با قرار دادن $x=0$ نشان دهید که ریشه به سمت $-.5$ همگراست، در صورتی که مقدار منفی استفاده شود و در صورتی که از یک مقدار مثبت استفاده شود به ریشه 1 همگرا است. نشان دهید حتی اگر با تقریب کمی به جواب سوم که 4 می‌باشد شروع کنیم به این جواب همگرا نمی‌شود.

جواب: $x=+\sqrt{e^x/3}$ و $x=0$ بعد از 15 تکرار جواب $x=0.91002$ با خطای قابل اغماض $1E-5$ به دست می‌آید. با $x=-\sqrt{e^x/3}$ و $x_0=0$ بعد از 9 بار تکرار $x=-0.45896$ با خطای $1E-5$ به دست می‌آید.

با $x=Ln(3x^2)$ و $x_0=4$ به جواب $x=3.73308$ همگراست.

۲۳- یکی از ریشه‌های معادله درجه دوم x^2+x-1 عبارت است از $x=0.6180$ اگر با $x_1=1$ شروع کنیم عبارت معادل $x=1/(x+1)$ به این ریشه همگرا می‌شود، چند مرحله برای رسیدن به ریشه لازم است (تا ۴ رقم اعشار با روش تکرار خطی).

جواب: $x=g(x)=\frac{1}{x+1}$ بعد از 10 تکرار به جواب $x=0.6181$ همگراست. با روش تکرار خطی 8 تکرار لازم است.

۲۴- جواب‌های معادله $f(x)=x-cos(x)$ را به روش‌های تکرار تابعی، نیوتن، درون یابی خطی بدست آورید.

۲۵- نشان دهید که معادله $Ln(x)-2+x=0$ دارای یک جواب در فاصله $[1,2]$ می‌باشد. روش تکرار تابعی را بکار برده و جواب را تا چهار رقم با معنی صحیح بدست آوردید.

۲۶- معادله درجه سه $2x^3+4x^2-2x-5=0$ دارای ریشه‌ای نزدیک به $x=1$ است. حداقل سه نمونه دیگر پیدا کنید که از $x=1.5$ شروع شده و به این ریشه همگرا می‌شوند.

جواب:

$$g(x)=((2x+5)/(2X+4))^{1/2};$$
$$g(x)=((-2x^3+2X+5)/4)^{1/2};$$
$$g(x)=((-4x^2+2X+5)/2)^{1/3}.$$

حالت سوم سرعت همگرایی کمی دارد.

۲۷- اگر $f(x)=0$ دارای ریشه مضاعف در $x=r$ باشد آنگاه $f'(r)=0$ و بنابراین شرط همگرائی ممکن است برای هر دامنه‌ای که شامل آن باشد صدق نکند. با توجه به شرط همگرائی.

$$\frac{|f(x)f''(x)|}{f'^2(x)}<1$$

بنابراین در روش نیوتن اطمینانی برای بدست آوردن ریشه x^2 نخواهیم داشت. معادله ساده $x^2-4x+4=0$ یا $(x-2)^2=0$ دارای یک ریشه مضاعف $x=2$ است و $f'(x)=2x-4$ در $x=2$ صفر می‌شود. اگر با هر مقدار معین x_0 شروع کنیم، روش نیوتن همگرا خواهد بود. این واقعیت را با بحث همگرایی، دوباره بررسی کنید.

جواب: شرط همگرایی برقرار است:

$$\left|\frac{f(x)f''(x)}{f'^2(x)}\right| = \left|\frac{2(x-2)^2}{4(x-2)^2}\right| = \frac{1}{2} \quad \langle 1$$

از رابطه نیوتن‌داریم:

$$x_{r+1}=x_r-\frac{f(x_r)}{f'(x_r)}=x_r-\frac{(x_r-2)^2}{2(x_r-2)}=x_r-\frac{1}{2}x_r+1=\frac{1}{2}x_r+1$$

$$x_0=3,\ x_1=2.5,\ x_2=2.25,\ x_3=2.125,\ x_4=2.063,\ x_5=2{,}032,\ x_n=2.0000$$

۲۸ - در معادله درجه دوم x^2-4x+4 که دارای ریشه مضاعف در $x=2$ می‌باشد با $x_0=1$ شروع کنید و با استفاده از روش نیوتن و تقریب‌های مناسب معادله را بدست آورید. خطاها را در هر مرحله با مرحله بعدی مقایسه کنید. آیا در این حالت روش نیوتن به صورت مرتبه دو یا فقط خطی همگرا است؟

سرعت همگرائی را چگونه می‌توان محاسبه نمود؟

$$x_{r+1}=\frac{1}{2}x_r+1 \quad , x_r=\frac{1}{2}x_{r-1}+1$$

جواب:

$$|x_{r+1}-x_r| = \left|\frac{1}{2}x_r+1-\frac{1}{2}x_{r-1}-1\right| = \frac{1}{2}|x_r-x_{r-1}|$$

$e_{r+1}=Ke_r$ و روش به صورت خطی همگراست.

اگر با $x_0=1$ شروع کنیم.

$$x_0=1,\ x_1=1.5,\ x_2=1.75,\ x_3=1.875,\ \ x_n=2.0000$$

$$e_0=1,\ e_1=0.5,\ e_2=0.2,\ e_3=0.125,\ \ldots\ e_n\to 0$$

۲۹ - اگر $f(x)=0$, $f'(x)=0$ اما $f''(x)\neq 0$، نشان دهید که

$$x_{n+1}=x_n-2f(x_n)/f'(x_n)$$

اگر x_n در همسایگی جواب باشد همگرایی مرتبه دو خواهد داشت.

روش نیوتن به یک ریشه از ریشه‌های مکرر مرتبه m تعمیم می‌یابد، بطوری که:

$$x_{n+1}=x_n-m\,f(x_n)/f'(x_n)$$

نشان دهید با محدودیت‌های مناسب روی $f(x)$، همگرائی مرتبه دو می‌باشد.

جواب:

$$g(x)=x-\frac{2f(x)}{f'(x)}$$

$$g'(x)=1-2\frac{f'^2(x)-f(x)f''(x)}{f'^2(x)}=1-2+\frac{2f(x)f''(x)}{f'^2(x)}$$

$$g'(x)=-1+2\frac{f(x)f''(x)}{f'^2(x)}=-1+2f''(x)\frac{f(x)}{f'^2(x)}$$

در همسایگی جواب $x=r$ داریم $f(r)=0$ و $f'(r)=0$ و از قاعده هوپیتال استفاده می‌کنیم.

$$\lim_{x\to r}g'(x)=-1+2\lim_{x\to r}f''(x)\frac{f'(x)}{2f'(x)f''(x)}=-1+1=0$$

و برای تابع تکراری نیز طبق قاعد هوپیتال داریم.

$$g(x)\underset{x\to r}{=}\lim_{x\to r}x-2\frac{f(x)}{f'(x)}=\lim_{x-r}x-2\frac{f'(x)}{f''(x)}=x=r$$

در صورتی که $f(x)=0$ دارای ریشه مکرر مرتبه m باشد، از رابطه زیر استفاده می‌کنیم:

$$x_{n+1}=x_n-m\frac{f(x_n)}{f'(x_n)}$$

بطوری که $f^{(m)}(r)\neq 0,\ f^{(m-1)}(r)=0,\ \dots,\ f''(r)=0,\ f'(r)=0$

$$g(x)=x-m\frac{f(x)}{f'(x)}$$

$$g'(x)=1-m\frac{f'^2-ff''}{f'^2}=1-m+m\frac{f}{f'^2}f''$$

از قاعده هوپیتال استفاده می‌کنیم:

$$\lim_{x\to r}g'(x)=1-m+m\lim_{x\to r}\frac{f'f''+ff'''}{2f'f''}$$

$$\lim_{x\to r}g'(x)=1-m+m\left[\frac{1}{2}+\lim_{x\to r}\frac{ff''}{2f'f''}\right]=1-\frac{1}{2}m+\frac{m}{2}\lim_{x\to r}\frac{ff'''}{f'f''}$$

و در رابطه تکراری نیوتن داریم:

$$x_{n+1}=x_n-m\frac{f(x_n)}{f'(x_n)}$$

اگر $x_n \to r$: در حد داریم:

$$x_{n+1}=\lim_{x_n\to r}x_n-m\lim_{x_n\to r}\frac{f'(x_n)}{f''(x_n)}=r-m\lim\frac{f^{(m-1)}(x_n)}{f^{(m)}(x_n)}=r$$

$$\lim_{x\to r}g'(x)=1-\frac{1}{2}m+\frac{m}{2}\lim_{x\to r}\frac{f}{f'}\times\lim_{x\to\infty}\left[\frac{f'''}{f''}\right]$$

$$=1-\frac{1}{2}m+\frac{m}{2}\lim_{x\to r}\frac{f''}{f'''}\times\lim_{x\to\infty}\frac{f'''}{f''}=1-\frac{1}{2}m+\frac{m}{2}=1$$

۳۰- لازم است جواب‌های حقیقی و مثبت معادله $e^{-x}\ cos(x) = k$ بازاء مقادیر مختلف k $x>0,\ k$ را بدست آوریم. نشان دهید که اگر $k > 1$ یا $k<-exp(-3\ \pi/4)/2\sqrt{2}$، معادله دارای جواب نیست. در صورتی که برای سایر مقادیر k معادله دارای جواب است. و چنانچه $k = 2.$ ، معادله دارای یک جواب است. جواب تقریبی را پیدا نمائید و معین کنید که کدامیک از دو عمل تکراری زیر به این جواب همگراست. جواب را تا چهار رقم با معنی بدست آورید.

$x_{r+1} = cos^{-1}(ke^{x_r})$

$x_{r+1} = Ln(cosx_r)-Ln(k)$

۳۱- معادله $x = Ln(x) - 2$ دارای جوابی نزدیک به $x = 1.5$ می‌باشد.

a- این جواب را با بکار بردن روش نیوتن پیدا نمائید.

b- این جواب را با بکار بردن روش درون یابی خطی بدست آورید.

با شمارش کل تعداد اعمال لازم در محاسبه، با توجه به اینکه زمان محاسبه در ماشین برای آنها تقریباً برابر است با: جمع و تفریق 4، ضرب 10، تقسیم 29، $[log]$ لگاریتم 170 (برحسب میلیونیم ثانیه). مشخص نمائید که روش نیوتن با صرفه‌تر است یا روش درون یابی خطی. برای یک مسئله کلی $g(x)=0$ چه موقع روش خطی به روش نیوتن ارجحیت دارد.

۳۲- اگر در روش تکراری نیوتن $x_r = \Theta + \varepsilon_r$ و $x_{r+1} = \Theta + \varepsilon_{r+1}$ بطوری که Θ جواب معادلهٔ $f(x) = 0$ باشد،

a- نشان دهید که:

$$\varepsilon_{r+1} \cong \frac{\varepsilon_r^{\,2} f''(\Theta)}{2!\, f'(\Theta)}$$

b- از *a* استفاده کرده نشان دهید که:

$$\varepsilon_r \cong \left[\frac{f''(\Theta)}{2!\, f'(\Theta)}\right]^{(2^r-1)}\varepsilon_0^{(2^r)}$$

و نتیجه بگیرید که روش نیوتن همگراست مشروط بر آنکه:

$$\left|\frac{\varepsilon_0 f''(\Theta)}{2! f'(\Theta)}\right| < 1$$

۳۳- با رسم توابع e^x و $\frac{x+1}{x-1}$ نشان دهید که معادله $e^x = \frac{x+1}{x-1}$ تنها دارای دو ریشه حقیقی است. با استفاده از روش نیوتن ریشه مثبت آن را تا دو رقم با معنای صحیح بدست آورید.

۳۴- معادله $x^4 + 2x^3 - 7x^2 + 3 = 0$ دارای چهار ریشه است که دوتای آنها مثبت است. با روش نیوتن و با استفاده از ماشین حساب این ریشه‌ها را تا 7 رقم با معنی صحیح بدست آورید. سپس بقیه ریشه‌ها را با کاهش معادله بدست آورید.

جواب: یک ریشه را از روش نیوتن به دست می‌آوریم مانند $r=1.61803$ و سپس با تقسیم $p_4(x)$ بر $x-r$ از چند جمله‌ای خارج قسمت $p_3(x)$ ریشه دیگری را به دست می‌آوریم و عملیات را ادامه می‌دهیم تا کلیه ریشه‌ها به دست آیند.

$.791287, -0.618033, -3.79128$

۳۵- الف) اگر $p(x)$ دارای ریشه مکرر از مرتبه m در $x=r$ باشد نشان دهید $q(x) = p(x)/p'(x)$ دارای همان ریشه‌های $p(x)$ است. اما فقط یک ریشه ساده در $x=r$ دارد.

ب) نشان دهید که در حقیقت همه ریشه‌های $q(x)$ ساده هستند.

جواب : الف)

$p(x)=(x-r)^m R(x)$

$p'(x)=m(x-r)^{m-1}R(x)+(x-r)^m R'(x)=(x-r)^{m-1}[mR(x)+(x-r)R'(x)]$

$q(x)=p(x)/p'(x)=(x-r)R(x)/[mR(x)+(x-r)R'(x)]$

ب) زیرا که $R(x)$ دارای ریشه مکرر نیست و $q(x)$ نیز ریشه مکرر ندارد.

۳۶- معادله $x^4 + 4.6x^3 + 6.6x^2 - 11x - 14$ مفروض است. با استفاده از تقسیم ترکیبی مقدار $p(-1)$ را حساب کنید. $p(x)$ را به صورت حاصلضرب دو چند جمله بنویسید، که یکی با درجه سوم باشد. ریشه حقیقی مثبت این چند جمله‌ای درجه سوم را پیدا کنید و سپس از فرمول معادله درجه دوم برای یافتن دو ریشه دیگر استفاده کنید. به این روش، روش کاهش یا *Deflation* می‌گویند.

جواب: $P(x)=(x+1)(x-1.4)(x^2+5x+10)$

۳۷- فرض کنید $p(x)= a_1x^n + a_2x^{n-1} + \dots a_nx^1 + a_{n+1}$ یک برنامه کامپیوتری (به هر زبان) بنویسید و $p(x)$ و $p'(x)$ را به ازاء مقدار معین x به روش تقسیم ترکیبی حساب کنید.

جواب: درجه N و ضرایب $a(1), \dots, a(N+1)$ را می‌خوانیم:

```
input x.
FOR i = 2 to N+1 DO
    a(i) = X*a(i-1) + a(i) ENDDO
FOR i = 2 to N DO
    a(i) = X*a(i-1) + a(i) ENDDO
P'(X) = a(N).
```

۳۸- از عامل ضرب آزمایشی $x^2 - 4x + 5$ شروع نموده و با استفاده از روش بیرستو چند جمله‌ای $x^4 + 3.1x^3 - 2.1x^2 + 1.1x - 5.2$ را به عوامل ضرب تبدیل نموده و چهار

ریشه آن را بدست آورید.

جواب: عوامل ضرب برابرند با: $(x^2-4.1x+5.2)$, (x^2+x+1)

۳۹- ریشه‌های معادله مسئله ۳۶ را با روش بیرستو بدست آورید.

جواب $(x^2-0.4x-1.4)$, $(x^2+5x+10)$

۴۰- هیچکدام از ریشه‌های این چند جمله‌ای درجه چهار، حقیقی نیستند ریشه‌های مختلط این معادله را با تبدیل آن به عوامل ضرب بدست آورید.

$$x^4 + 4x^3 + 21x^2 + 4x + 20 = 0$$

جواب: عوامل ضرب برابرند با: x^2+1 $x^2+3.99997x+19.99988$

۴۱- روش بیرستو را بکار برده و ریشه‌های معادله زیر را بدست آورید. کلیه ریشه‌ها مختلط خواهند بود.

$$x^4 - 3.1x^3 + 2.1x^2 - 1.1x + 5.2=0$$

جواب: $(-0.5\pm0.866i)$, $(2.05\pm0.9987i)$

مسائل کاربردی و پروژه‌ها

۴۲- فرض کنید

$$\begin{cases} x(0)=x'(0)=y(0)=0 \\ x'' - x + y = g(t) \\ x'' + x + 2y' + y = f(t) \end{cases}$$

برای حل دستگاه معادله دیفرانسیل درجه دوم به روش تبدیل لاپلاس لازم است که عبارت را به فرم زیر در آوریم $(s^2+1)(s) - (2s+1)(s^2-1) = s^3 + s^2 + 3s + 1$

بنابراین از عوامل ضرب نیز می‌توان برای تبدیل عکس هم استفاده کرد. عوامل ضرب چه هستند.

جواب: $(s-1.4812)(s+0.8111)(s+2.1701)$

۴۳- دسانتیس *Desantis (1976)* رابطه‌ای برای تراکم پذیری گازهای حقیقی به فرم زیر بدست آورد:

$$z = \frac{1 + y + y^2 - y^3}{(1 - y)^3}$$

که در آن $y=\frac{b}{4v}$ و b عبارت است از تصحیح والس *waals* و v حجم مولار است. اگر $z=0.892$ باشد مقدار y چیست؟

جواب: پس از ساده کردن رابطه داریم:

$$f(y)=1+y+y^2-y^3+0.892(y^3-3y^2+3y-1)=0$$

چون $f(0)\ f(1)<0$ و طبق قضیه بولزانو در فاصله $[0,1]$ دارای حداقل یک ریشه است. با استفاده از روش نیوتن یا درون یابی خطی جواب برابر $y=-0.028997$ به دست می‌آید.

۴۴- در مطالعات انرژی خورشیدی با تمرکز نور روی آینه‌های مسطح در یک جمع کننده مرکزی، وانت هال *Hull(1976) - vant* معادله‌ای برای ضریب C به عنوان «ضریب تمرکز هندسی» بدست آورده است:

$$C=\frac{\pi(h/cosA)^2F}{0.5\ \pi\ D^2(1+sinA\ -\ 0.5\ cosA)}$$

که در آن A عبارت است از زاویه لبه میدان و F، همگرایی شکست (فاصله کانونی) مربوط به میدان آینه است. D قطر میدان و h ارتفاع آن می‌باشد. در صورتی که $h=300$ و $C=1200$ و $F=0.8$ و $D=14$ باشد، A را پیدا کنید.

جواب: زاویه A را بین 0، $\frac{\pi}{2}$ در نظر می‌گیریم. از روش نیوتن یا روش نصف کردن مقدار A محاسبه می‌شود.

$A=0.1176$ رادیان $(A=6.74$ درجه$)$

۴۴- «لی» و «دافی» *(1976 - lee & Daffy)* ضریب اصطکاک را برای جریان محلول الیاف به عدد رینولدز *Reynolds* با استفاده از این معادله:

$$\frac{1}{\sqrt{f}}=(\frac{1}{k})\ L_n(RE\sqrt{f})\ +\ (14\ -\ \frac{5.6}{k})$$

بدست آوردند.

در این رابطه f، ضریب اصطکاک، RE عدد رینولدز و k ثابتی است که بستگی به غلظت محلول دارد محلولی که دارای غلظت 0.08% است و $k=0.28$ می‌باشد. مقدار f را بدست آورید در صورتی که $RE=3750$

جواب: با استفاده از روش نیوتن یا روش درون یابی خطی می‌توان مقدار $f=5.12E-3$ را به دست آورد.

۴۵- معادله *"Redlich - Kwong"* عبارت است از

$$P=\frac{RT}{v-b}-\frac{A(T)}{v(v+b)}$$

با اندازه‌های $p=87.3$ و $T=486.9$ و $v=12.005$ ساخته شده است. می دانیم که تحت این شرایط $A(t)=0.0837$ می‌باشد. $R = 1.98$ یک ثابت است. مقدار b را که در رابطه صدق می‌کند بدست آورید.

جواب: $b=0.9620$

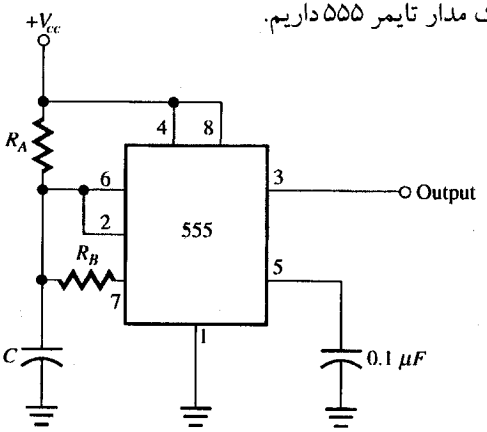
۴۶- بر اساس کار فرانک کامنتسکی *(Kamenetski)* در سال *1955* دما در داخل یک ماده که در تماس منبع حرارتی است با حل معادله زیر بدست می‌آید.

$$e^{-(½)t}\ \cosh^{-1}(e^{-(½)t}) = \sqrt{½L_{cr}}$$

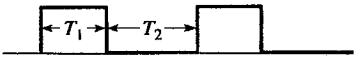
$L_{cr} = 0.088$ مقدار t را محاسبه کنید.

جواب: $t=4.7576$

۴۷- فرض کنید که در یک مدار تایمر ۵۵۵ داریم.



شکل موج خروجی آن عبارت است از:



که در آن $T_1 + T_2 = \frac{1}{f}$ و f فرکانس است.

$$Duty\ cycle = \frac{T_1}{T_1+T_2} \times 100\%$$ ، می‌توان نشان داد $T_1 = R_A C\ L_n(2)$

$$T_2 = \frac{R_A R_B c}{R_A+R_B} + Ln\left(\left|\frac{R_A-2R_B}{2R_A-R_B}\right|\right)$$

فرض کنید: $R_A = 8670$ و $C = 0.01\times10^{-6}$, $T_2 = 1.4\times10^{-4}$

الف) سیکل فرضی *Duty cycle* و f , T_1 را بیابید.

ب) R_B را با استفاده از یک برنامه کامپیوتری درون یابی خطی پیدا کنید.

ج) یک f و یک سیکل فرضی *Duty cycle* را انتخاب کرده و T_2 و T_1 را بدست آورید.

جواب: الف)

$T_1=6.0096E-6,\ f=6848.9,\ Duty\ cycle=4.12\%$

ب) $R_B=15531$ (همچنین 20629)

ج) برای $T_2=1.8E-4,\ T_1=2E-5,\ Duty\ cycle=10\%\ ,\ f=5000$

۴۸- حل یک مسئله مقدار مرزی با کمک روش سری فوریه به صور تحلیلی اغلب منجر به پیدا کردن ریشه‌های معادلات غیر جبری می‌گردد. مثلا:

$$y''+\lambda y = 0 \qquad y(0) = 0\ ,\ y(1) = y'(1)$$

منجر به حل $\tan(z) = z$ می‌گردد. سه مقدار z جز $z = 0$ را پیدا کنید.

جواب: از روش تکرار تابعی و روش نیوتن جواب‌ها به دست می‌آیند:

1.5707, 4.7123, 7.7252

۴۹- همه نقاط ماکزیمم و می‌نیمم تابع

$$f(x) = [\sin(x)]^6 * e^{20x} * \tan(1-x)$$

را روی فاصله $(0,1)$ پیدا کنید. (به معایب استفاده از $f'(x)=0$ در روش نیوتن توجه کنید).

جواب: از روش درون یابی خطی یا روش نصف کردن جواب $x=0.95991$ به دست می‌آید.

۵۰- در فصل‌های بعد روش محاسبه انتگرال معین یک تابع بنام گاوس ارائه خواهد شد. برای بسط فرمول‌های آن لازم است که صفرهای چند جمله‌ای «لژاندر» را بدست آوریم صفرهای چند جمله‌ای لژاندر درجه شش زیر را حساب کنید.

$$P_6(x) = \frac{1}{48}(693x^6-945x^4+315x^2-15)$$

(توجه: صفرهای چند جمله‌ای‌های لژاندر از نظر قدر مطلق کمتر از یک هستند.)

جواب: $\pm0.2386,\ \pm0.6612,\ \pm0.9325$

۵۱- چند جمله‌ای لژاندر به عنوان یک مجموعه از چند جمله‌ای‌هایی است که به آنها چند جمله‌ای‌های «متعامد» می‌گویند. به دسته دیگری از چند جمله‌ای‌ها لاگوئر *(Laguerre)* می‌گویند.

صفرهای معادلات زیر را بدست آورید.

a) $L_3(x) = x^3-9x^2+18x-6$,

b) $L_4(x)=x^4-16x^3+72x^2-96x+24$

جواب: a) صفرها برابرند با: 6.2899, 0.41579, 2.2942

b) صفرها برابرند با: 1.7457, 0.32255, 9.3949, 4.5367

۵۲- به یک دسته از چند جمله‌ای‌های «متعامد» چند جمله‌ای‌های چبیشف می‌گویند.

ریشه‌های این معادله را بدست آورید $T_6(x)=32x^6-48x^4+18x^2-1=0$

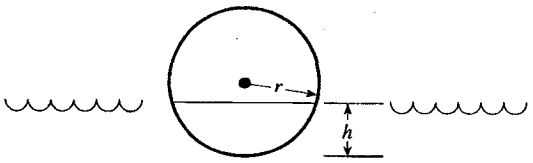
همه ریشه‌های چند جمله‌ای‌های چبیشف نیز کمتر از یک هستند.

جواب: صفرهای تابع برابرند با:

$\pm 0.26433,\ \pm 0.70711,\ \pm 0.96593$

۵۳- یک کره با چگالی d و شعاع r دارای وزن $\frac{4}{3}\pi r^3 d$ می‌باشد حجم یک قسمت از کره $\frac{1}{3}\pi(3rh^2-h^3)$ است.

ارتفاع قسمتی از کره با چگالی 0.6 که در آب قرار گرفته بر حسب شعاعش بدست آورید.



جواب: هر جسمی وارد مایعی شود به اندازه وزن مایع هم حجمش از وزن آن کاسته می‌شود.

پس:

$$\frac{4}{3}\pi r^3\times 0.6=\frac{1}{3}\pi(3rh^2-h^3)$$

$$2.4r^3=3rh^2-h^3,$$

$$h^3-3rh^2+2.4r^3=0,$$

$$\left(\frac{h}{r}\right)^3-3\left(\frac{h}{r}\right)^2+2.4=0,$$

$$x^3-3x+2.4=0$$

$$x=\frac{h}{r}=1.1341.$$

۵۴- الف) برنامه *NEWTN* را برای محاسبه ریشه‌های مختلط مجدداً بنویسید و آن را تست کنید.

ب) همین برنامه را برای *MULLER* انجام دهید.

۵۵) برنامه‌ای بنویسید که از روش درون یابی خطی برای پیدا کردن اولین تقریب به ریشه $f(x)=0$ استفاده کند و سپس برنامه فرعی *NEWTN* را برای تصحیح آن صدا کنید.

فصل ۳

حل دستگاه معادلات خطی

کارل فریدریش گاوس
GAUSS
(۱۷۷۷ - ۱۸۵۵)

گاوس در سال ۱۷۷۷ در برونسویک آلمان بدنیا آمد. وی را بزرگترین ریاضیدان قرن نوزدهم و همراه با ارشمیدس و نیوتن جز بزرگترین ریاضیدانان همهٔ اعصار بر شمرده‌اند.

گاوس رسالهٔ دکترایش را در دانشگاه هلمشتد در سن بیست سالگی نوشته است، وی اولین برهان قضیه اساس جبر (یک معادلهٔ چند جمله‌ای با ضرایب مختلط و از درجهٔ $n>0$ حداقل یک ریشه مختلط دارد) را ارائه داد. برای اثبات این قضیه قبلاً از سوی نیوتن، اویلر، دالامبر و لاگرانژ تلاش‌های بی‌حاصلی به عمل آمده بود.

گاوس سهم ارزنده‌ای در نجوم، و الکتریسیته دارد. در سال ۱۸۱۲، در مقاله‌ای دربارهٔ سری فوق هندسی، اولین تحقیق اصول خود را دربارهٔ همگرایی سری‌ها به عمل آورد.

بزرگترین اثر منتشر شدهٔ گاوس تحقیقات حسابان اوست، اثری که در نظریهٔ نوین اعداد دارای اهمیّت اساسی است. یافته‌های گاوس در مورد ساختمان چند ضلعی‌های منظم در این اثر ظاهر می‌شود. شاهکار گاوس دربارهٔ نظریه سطوح، تحقیقات کلی دربارهٔ رویه‌های منحنی در سال ۱۸۲۷ منتشر شده. این گفته معروف گاوس است: «ریاضیات ملکهٔ علوم، و نظریهٔ اعداد ملکهٔ ریاضیات است».

موضوعات این فصل

* ولتاژ و جریان در یک شبکه
* محاسبه نیروها در یک خرپای مسطح
* روش حذفی گاوس - جردن
* روش تجزیه LU
* روش تجزیه چولسکی
* شرایط ناهنجار دستگاه معادلات خطی
* روش‌های تکراری
* حل دستگاه معادلات غیر خطی
* برنامه حل دستگاه معادلات خطی به روش گاوس - سیّدل به زبان پاسکال بنام GSITRM.PAS
* برنامه حل دستگاه معادلات غیر خطی به روش نیوتن به زبان پاسکال بنام NLSYST.PAS
* برنامه حل دستگاه معادلات به‌روش تجزیه LU به‌زبان C بنام PDEC.C
* تمرینات فصل سوم

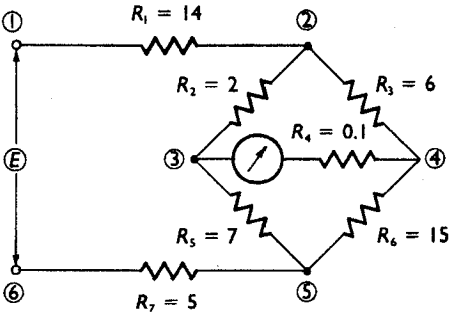
۱.۳ ولتاژ و جریان در یک شبکه

مهندسین برق اغلب باید عبور جریان و ولتاژ موجود در یک شبکه پیچیده متشکل از مقاومتها را پیدا کنند.

در اینجا یک مسأله در این مورد مطرح شده است.

هفت مقاومت مطابق شکل (۱.۳) بسته شده است و ولتاژ اعمال شده برابر با ولتاژ نقاط *6,1* است.

این شبکه را بعنوان تغییرات، روی پل وتستون می‌شناسیم.



شکل ۱.۳

فرض کنیم که می‌خواهیم جریانی که بین نقاط *4,3* ایجاد شده است را با استفاده از قوانین فیزیک بدست آوریم.

روش محاسبات می‌تواند ولتاژ را در هر نقطه (گره) بدهد.

قانون کیرشف: مجموع جریانهائی که از هر گره عبور می‌کند، صفر است.

قانون اهم: جریانی که از یک مقاومت می‌گذرد مساوی ولتاژ دو سر مقاومت تقسیم بر مقاومت آن است.

برای چهار ولتاژ و هفت جریان می‌توانیم با استفاده از این قوانین یازده معادله با یازده مجهول تشکیل دهیم.

جریانهائی که از چهار گره عبور می‌کند:

$$i_{12} - i_{23} - i_{24} = 0 \text{ (node 2)}$$
$$i_{23} - i_{34} - i_{35} = 0 \text{ (node 3)}$$
$$i_{24} - i_{34} - i_{45} = 0 \text{ (node 4)}$$
$$i_{35} - i_{45} - i_{56} = 0 \text{ (node 5)}$$

جریانهائی که از مقاومتها می‌گذرند:

$$i_{12} = \frac{5 - V_2}{14}$$

$$i_{23} = \frac{V_2 - V_3}{2}$$

$$i_{24} = \frac{V_2 - V_4}{6}$$

$$i_{34} = \frac{V_3 - V_4}{0.1}$$

$$i_{35} = \frac{V_3 - V_5}{7}$$

$$i_{45} = \frac{V_4 - V_5}{15}$$

$$i_{56} = \frac{V_5 - 0}{5}$$

در این فصل روشهایی را برای بدست آوردن متغیرها با استفاده از کامپیوتر شرح می‌دهیم. بدون یک کامپیوتر انجام کار ملال آور است و به آسانی خطای محاسباتی بوجود می‌آید. ما معمولاً معادلات را به فرم استاندارد در می‌آوریم، بهر حال نتیجه چنین است:

$$\begin{aligned}
i_{12} - i_{23} - i_{24} &= 0\\
i_{23} - i_{34} - i_{35} &= 0\\
i_{24} + i_{34} - i_{45} &= 0\\
i_{35} + i_{45} - i_{56} &= 0\\
14i_{12} + V_2 &= 5\\
2i_{23} - V_2 + V_3 &= 0\\
6i_{24} - V_2 + V_4 &= 0\\
0.1i_{34} - V_3 + V_4 &= 0\\
7i_{35} - V_3 + V_5 &= 0\\
15i_{45} - V_4 + V_5 &= 0\\
5i_{56} - V_5 &= 0
\end{aligned}$$

۲.۳ محاسبهٔ نیروها در یک خرپای مسطح

خرپا ساختاری است که برای تحمل کردن بارها همانند پلها و ساختمانها بکار می‌رود. یک خرپا از اعضای مستقیمی ترکیب شده است که تشکیل یک یا چند مثلث را می‌دهند و باعث پایداری سازه می‌شوند.

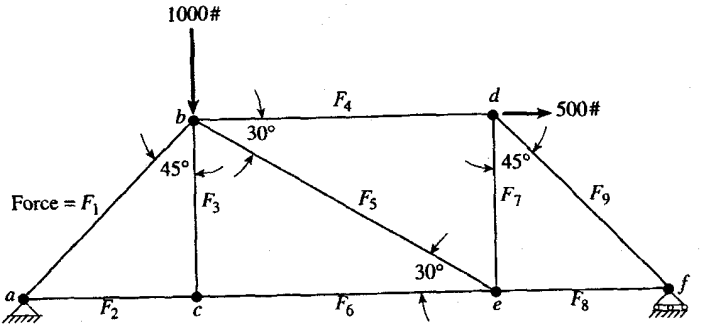
در خرپا این خصوصیات وجود دارد:

۱) تمام گره‌ها بصورت اتصالات لولایی هستند. (لولا *(Pin)*، گره‌ای می‌باشد که اتصال آن فاقد اصطکاک هست ولی می‌تواند آزادانه بچرخد از اینرو نمی‌تواند گشتاور خمشی به عضو متصل به خود وارد کند).

۲) همهٔ بارها فقط در گره‌ها وارد می‌شوند. *[همینطور عکس العملها]*

۳) از وزن اعضای خرپا صرف نظر می‌شود.

واضح است که خرپا فقط در بعضی از گره‌ها باید بار تحمل کند. بارها ممکن است ثابت باشند یا امکان انتقال در یک جهت را بدهند.



شکل (۲.۳) نشان دهنده یک خرپاست.

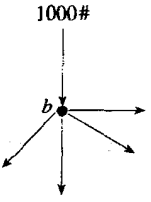
در این خرپا، شش گره و نه عضو وجود دارد. هر گره دارای دو درجه آزادی است (بدین معنی که دو نیروی مستقل در آنجا عمل می‌کند.). نیروی وارد در هر گره را به مؤلفه‌های x و y تجزیه می‌کنید.

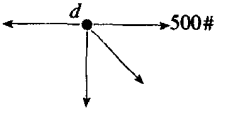
نیروهائی که در خرپا عمل می‌کنند در امتداد عضوها وارد می‌شوند. مرسوم است که نیروها را در گره در امتداد اعضا بررسی می‌کنیم.

چون نه عضو وجود دارد، ما دارای نه نیروی عضوی هستیم $F_i,\ i=\ 1,2,...,9$.

چون سیستم در حال تعادل است، با صفر قرار دادن مجموع همه نیروهای عمل کنندهٔ افقی و عمودی در هر گره می‌توان این نیروها را معین کرد.

بعنوان مثال، در گره‌های b و d داریم:





اگر از گره b شروع کنیم و در مورد سایر گره‌ها ادامه دهیم، می‌توانیم 9 معادله بنویسیم.

$$\begin{aligned}
\cos(45°)*F_1 - F_4 - \cos(30°)*F_5 &= 0\\
\sin(45°)*F_1 + F_3 + \sin(30°)*F_5 &= -1000\\
F_2 - F_6 &= 0\\
-F_3 &= 0\\
F_4 - \sin(45°)*F_9 &= 500\\
F_7 + \cos(45°)*F_9 &= 0\\
\cos(30°)*F_5 + F_6 - F_8 &= 0\\
\sin(30°)*F_5 + F_7 &= 500\\
F_8 + \cos(45°)*F_9 &= 0
\end{aligned}$$

این 9 معادله دقیقاً برای بدست آوردن نه مجهول نیروهای عضوی کافی است. یک چنین خرپائی «پایدار معین» نامیده می‌شود. اگر عضو دیگری از گره c به d متصل کنیم، در اینصورت معادلات کافی برای حل دستگاه نخواهیم داشت.

این حالت «پایدار نامعین» نامیده می‌شود.

۳.۳ روش حذفی گاوس - جردن

این روش بسیار ساده است و توضیح آنرا با یک مثال آغاز می‌کنیم.

مثال ۱.۳ مطلوبست حل دستگاه زیر:

$$\begin{bmatrix}1 & 3 & 4\\ 2 & 7 & 18\\ 7 & 1 & 3\end{bmatrix} x = \begin{bmatrix}2\\1\\5\end{bmatrix}$$

سطر اول را در ۲- ضرب کرده و متناظراً با سطر دوم جمع می‌کنیم همچنین سطر اول را در 7- ضرب و با سطر سوم جمع می‌کنیم، نتیجه چنین می‌شود:

$$\begin{bmatrix}1 & 3 & 4\\ 0 & 1 & 10\\ 0 & -20 & -25\end{bmatrix} x = \begin{bmatrix}2\\-3\\-9\end{bmatrix} \qquad \begin{cases}1x_1 + 3x_2 + 4x_3 = 2\\ 0x_1 + 1x_2 + 10x_3 = -3\\ 0x_1 - 20x_2 - 25x_3 = -9\end{cases}$$

اگر سطر دوم را در ۲۰ ضرب و با سطر سوم جمع کنیم خواهیم داشت.

$$\begin{bmatrix}1 & 3 & 4\\ 0 & 1 & 10\\ 0 & 0 & 175\end{bmatrix} x = \begin{bmatrix}2\\-3\\-69\end{bmatrix} \qquad \begin{cases}1x_1+3x_2+4x_3 = 2\\ 0x_1+1x_2+10x_3 = -3\\ 0x_1+0x_2+175x_3 = -69\end{cases}$$

از معادله سوم داریم:

$$x_3 = \frac{-69}{175}$$

از معادله دوم داریم: $x_2 = -3 - 10x_3 = -3 + \frac{690}{175} = \frac{165}{175}$

و از معادله اول x_1 بدست می‌آید. $x_1 = 2 - 3x_2 - 4x_3 = \frac{131}{175}$

این روش بنام روش حذفی می‌باشد و از دقت کافی برخوردار نیست. مگر آنکه دستگاه معادلات را طوری مرتب کنیم که در هر مرحله کلیه مضاربی که در صفر کردن عناصر یک ستون (یا سطر) بکار می‌روند کوچکتر یا مساوی یک باشند در اینصورت این روش پایدار می‌باشد (تقریباً جواب صحیح نتیجه می‌شود).در غیر اینصورت ممکن است عمل ناپایدار شود، یعنی خطای گرد کردن در معادله ای، در سایر مراحل انتشار یافته و با ضرب معادلات در عامل‌های انتخابی تغییر زیادی را بوجود می‌آورد.

روش حذفی گاوس با توضیح مختصر فوق روشن می‌گردد. در این روش با انتخاب و تغییر سطر ها چنان از خواص دترمینان‌ها (مضربی از یک معادله را به معادله دیگر افزوده و نتیجه را بجای آن قرار می‌دهیم) بهره گرفته و عناصر آنرا صفر می‌کنیم که دستگاه بصورت بالا مثلثی تبدیل شود.

$$\begin{bmatrix} \alpha_{1,1} & \cdots & & \alpha_{1,n} \\ & \alpha_{2,2} & \cdots & \alpha_{2,n} \\ & & \alpha_{3,3} & \cdots \ \alpha_{3,n} \\ O & & & \alpha_{n,n} \end{bmatrix} \underline{x} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

از معادله nام، x_n و از معادله $n-1$ام که بر حسب x_n و x_{n-1} می‌باشد، x_{n-1} و بهمین ترتیب سایر جوابها را تا x_1 بدست می‌آوریم.

فرض کنیم $a_{1\,1}$ بزرگترین عنصر از نظر قدر مطلق در اولین ستون باشد. (اگر چنین نباشد جای سطرها (معادلات) را عوض می‌کنیم). در اینصورت مضاربی از $a_{1\,1}$ را چنان انتخاب می‌کنیم که اگر با $a_{2\,1}$, $a_{3\,1}$,...,$a_{n\,1}$ جمع کنیم کلیه این عناصر صفر شوند. بدین معنی که در هر بار مضربی از سطر اول را با سطر دوم و سوم و... و nام جمع می‌کنیم، در اینصورت خواهیم داشت:

$$\begin{bmatrix} \alpha_{11} & \alpha_{1\,2} & \cdots & \alpha_{1\,n} \\ 0 & \alpha_{2\,2} & & \alpha_{2\,n} \\ \vdots & \vdots & \vdots & \\ 0 & \alpha_{n\,2} & & \alpha_{n\,n} \end{bmatrix} \quad x = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

مجدداً فرض می‌کنیم $\alpha_{2\,2}$ بزرگترین عنصر از نظر قدر مطلق در ستون دوم و زیر قطر باشد (اگر چنین نباشد جای سطرهای زیر قطر را عوض می‌کنیم). و مانند قسمت قبل عمل می‌کنیم تا کلیه عناصر زیر این عنصر در ستون دوم صفر شوند و اگر بهمین نحو در مورد سایر ستونها عمل کنیم ماتریس A به یک ماتریس بالا مثلثی تبدیل می‌شود.

مثال ۲.۳ مطلوبست حل دستگاه زیر:

$$\begin{bmatrix} .6 & 3.8 & 7 \\ 2.6 & 3.1 & 5 \\ 3.7 & 5.8 & 2.9 \end{bmatrix} \underline{x} = \begin{bmatrix} 5.7 \\ 2.6 \\ 3.4 \end{bmatrix}$$

$$\begin{bmatrix} 3.7 & 5.8 & 2.9 \\ 2.6 & 3.1 & 5 \\ .6 & 3.8 & 7 \end{bmatrix} \underline{x} = \begin{bmatrix} 3.4 \\ 2.6 \\ 5.7 \end{bmatrix}$$

بزرگترین عنصر در ستون اول 3.7 می‌باشد، پس جای سطر اول و سوم را عوض می‌کنیم و سطر اول را در $-\frac{2.6}{3.7} = -.703$ ضرب و با سطر دوم جمع می‌کنیم.

همچنین سطر اول را در $-\frac{.6}{3.7} = -.162$ ضرب و با سطر سوم جمع می‌کنیم و خواهیم داشت:

$$\begin{bmatrix} 3.7 & 5.8 & 2.9 \\ 0 & -977 & 2.961 \\ 0 & 2.860 & 6.53 \end{bmatrix} x = \begin{bmatrix} 3.4 \\ .210 \\ 5.149 \end{bmatrix}$$

در این مرحله بزرگترین عنصر درستون دوم که در زیر قطر یا روی آن قرار گرفته است برابر 2.860 می‌باشد. پس سطر دوم و سوم را با یکدیگر عوض می‌کنیم و سپس سطر دوم را در $-\frac{.977}{2.860} = -.342$ ضرب و با سطر سوم جمع می‌کنیم.

در این‌حال خواهیم داشت:

$$x_3 \frac{1.971}{5.194} = = .379$$

$$x_2 = \frac{5.149 - 6.530\, x_3}{2.860} = .935$$

$$x_1 = \frac{3.4 - 5.8\, x_2 - 2.9\, x_3}{3.7} = -.844$$

جوابها تا سه رقم اعشار صحیح می‌باشند.

توجه شود که: $Ax=b$

اگر $b \neq 0$ و اگر $det\ A \neq 0$ باشد، x وجود دارد و یکتاست

اگر $b=0$ باشد، در اینصورت $x = 0$ همیشه یک جواب است.

اگر $A=0$ det باشد، ممکن است جواب نهائی وجود داشته باشد.

۴.۳ روش تجزیه[1] LU

فرض می‌کنیم روش حذفی گاوس رادر مورد دستگاه معادلات $Ax=b$ بکار برده و دستگاه بصورت بالا مثلثی تبدیل گشته است $Ux=d$ اگر ماتریس واحد پائین مثلثی[2] زیر را در نظر بگیریم:

1- Decomposition 2- Lower Triangular

$$L = \begin{bmatrix} m_{1,1} & & 0 \\ m_{2,1} & m_{2,2} & \\ m_{3,1} & m_{3,2} & m_{3,3} \\ \vdots & & \\ m_{n,1} & m_{n,n} & \cdots m_{n,n} \end{bmatrix}$$

$\{m_{r,s}\}$ مضاربی می‌باشند که در روش حذفی گاوس بکار رفته اند. ($m_{r,s}$ ضریبی است که در تفریق s امین سطر معادله از r امین سطر معادله بکار رفته است، تا عنصر (r,s)ام از A را صفر کند)

$$U = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,n} \\ & \alpha_{2,2} & \cdots & \alpha_{2,n} \\ & & \ddots & \vdots \\ 0 & & & \alpha_{n,n} \end{bmatrix}$$

می‌توان نشان داد که: $LU = A$

اکنون مستقیماً تجزیه LU را تعیین می‌نمائیم. فرض کنیم:

$$\begin{bmatrix} \ell_{11} & 0 & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} & 0 \\ \ell_{41} & \ell_{42} & \ell_{43} & \ell_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}.$$

حال ماتریس‌های بالا مثلثی U و پایین مثلثی L را چنان تعیین می‌کنیم تا داشته باشیم:
$LU = A$

باید توجه نمود که در ماتریس‌های مثلثی L , U داریم:

$L_{ik} = 0 \quad k > i$
$U_{kj} = 0 \quad k > j$

حاصل ضرب سطر اول L در ستون‌های U را به دست می‌آوریم:

$$\ell_{11} u_{12} = a_{12}, \qquad \ell_{11} u_{13} = a_{13}, \qquad \ell_{11} u_{14} = a_{14},$$

مقادیر سطر اول U به دست می‌آیند:

$$u_{12} = \frac{a_{12}}{\ell_{11}}, \qquad u_{13} = \frac{a_{13}}{\ell_{11}}, \qquad u_{14} = \frac{a_{14}}{\ell_{11}}.$$

حاصل ضرب سطرهای L در ستون دوم U

$$\ell_{21} u_{12} + \ell_{22} = a_{22},$$
$$\ell_{31} u_{12} + \ell_{32} = a_{32},$$
$$\ell_{41} u_{12} + \ell_{42} = a_{42},$$

مقادیر ستون دوم L به دست می‌آیند

$$\ell_{22} = a_{22} - \ell_{21} u_{12},$$
$$\ell_{32} = a_{32} - \ell_{31} u_{12},$$
$$\ell_{42} = a_{42} - \ell_{41} u_{12}.$$

اکنون مقادیر سطر دوم ماتریس U را نیز به دست می‌آوریم:

$$u_{23} = \frac{a_{23} - \ell_{21} u_{13}}{\ell_{22}}, \qquad u_{24} = \frac{a_{24} - \ell_{21} u_{14}}{\ell_{22}},$$

حاصل ضرب سطر سوم L در ستون‌های U

$$\ell_{23} = a_{33} - \ell_{31} u_{13} - \ell_{32} u_{23}, \qquad \ell_{43} = a_{43} - \ell_{41} u_{13} - \ell_{42} u_{23},$$

مقادیر ستون سوم ℓ به دست می‌آیند.

$$u_{34} = \frac{a_{34} - \ell_{31} u_{14} - \ell_{32} u_{24}}{\ell_{33}},$$

$$\ell_{44} = a_{44} - \ell_{41} u_{14} - \ell_{42} u_{24} - \ell_{43} u_{34}.$$

و در حالت کلی داریم:

$$\ell_{ij} = a_{ij} - \sum_{k=1}^{j-1} \ell_{ik} u_{kj}, \qquad j \le i, \qquad i = 1, 2, \ldots, n,$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} \ell_{ik} u_{kj}}{\ell_{ii}}, \qquad i \le j, \qquad j = 2, 3, \ldots, n.$$

می‌توانیم عبارات فوق را با دقت کافی محاسبه نمائیم.

اگر $Ax = b$ ، و $A = LU$ در اینصورت : $LUx = b$

اگر $Ux=z$ را در نظر بگیریم، خواهیم داشت: $Ux = b$

در اینصورت از $Lz = b$ بردار z و از $Ux = z$ بردار x یعنی جواب دستگاه بدست می‌آید.

مثال ۳.۳ ماتریس A را به صورت LU تجزیه کنید.

$$A = \begin{bmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{bmatrix}.$$

معادلات را برای مقادیر ℓ و u را مطابق زیر بدست می‌آوریم:

$$\ell_{11} = 3, \qquad \ell_{21} = 1, \qquad \ell_{31} = 2; \qquad u_{12} = -\tfrac{1}{3}, \qquad u_{13} = \tfrac{2}{3}.$$

$$\ell_{22} = 2 - (1)(-\tfrac{1}{3}) = \tfrac{7}{3}, \qquad \ell_{32} = -2 - (2)(-\tfrac{1}{3}) = -\tfrac{4}{3}.$$

$$u_{23} = \frac{3 - (1)(\frac{2}{3})}{\frac{7}{3}} = 1, \qquad \ell_{33} = -1 - (2)(\tfrac{2}{3}) - (-\tfrac{4}{3})(1) = -1.$$

$$L = \begin{bmatrix} 3 & 0 & 0 \\ 1 & \frac{7}{3} & 0 \\ 2 & -\frac{4}{3} & -1 \end{bmatrix}, \qquad U = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{2}{3} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

اگر L و U را به صورت فشرده بنویسیم، داریم:

$$LU = \begin{bmatrix} 3 & -\frac{1}{3} & \frac{2}{3} \\ 1 & \frac{7}{3} & 1 \\ 2 & -\frac{4}{3} & -1 \end{bmatrix} \begin{matrix} \leftarrow ② \\ \\ \leftarrow ④ \end{matrix}$$

$$\begin{matrix} \uparrow & \uparrow & \uparrow \\ ① & ③ & ⑤ \end{matrix}$$

مثال ۴.۳ مثال ۲.۳ را با استفاده از روش تجزیه LU حل نمائید.

با توضیحات قبلی خواهیم داشت:

$$A = \begin{bmatrix} 3.7 & 5.8 & 2.9 \\ .6 & 3.8 & 7.0 \\ 2.6 & 3.1 & 5.0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ .162 & 1 & 0 \\ .703 & .342 & 1 \end{bmatrix} \begin{bmatrix} 3.7 & 5.8 & 2.9 \\ 0 & 2.86 & 6.53 \\ 0 & 0 & 5.194 \end{bmatrix} = LU$$

$Ax=b \qquad b^T = [3.4 \;\; 5.7 \;\; 2.6]$

$Lz = b \quad , \quad z_1 = 3.4$

$.162\, z + z_2 = 5.7 \quad , \quad z_2 = 5.149$

$.703\, z_1 - .342\, z_2 + z_3 = 2.6 \quad , \quad z_3 = 1.971$

$z^T = [3.4 \;\; 5.149 \;\; 1.971]$

$Ux = b$

$5.194\, x_3 = 1.971 \qquad x_3 = .379$

$2.86\, x_2 + 6.53\, x_3 = 5.149 \qquad x_2 = .935$

$3.7\, x_1 + 5.8\, x_2 + 2.9\, x_3 = 3.4 \quad x_1 = -.844$

روش تجزیه LU بیشتر در مواردی مفید است که تعدادی از دستگاه‌های مختلف دارای ضرایب عددی مساوی باشند مانند:

$Ax_1 = b_1 \qquad Ax_2 = b_2 \qquad Ax_3 = b_3 \ldots \; Ax_n = b_n$

که A در کلیه دستگاه معادلات فوق یکسان است. و بعد از تجزیه $A=LU$ در تمامی این موارد می‌توان از آن استفاده نمود.

۵.۳ تجزیه چولسکی[1]

این روش خاص برای تجزیه A در حالتی که ماتریس متقارن است، مورد استفاده قرار می‌گیرد. پس اگر A ماتریسی متقارن باشد، $A=A^T$ در اینصورت نشان می‌دهیم که: $A=LL^T$.

زیرا که

$$A=LU$$

$$A^T=(LU)^T = U^TL^T = LU$$

در نتیجه

$$U^T=L$$

$$U=L^T$$

$$A = LL^T$$

$$L^T=\begin{bmatrix} \ell_{11} & \ell_{21} & \cdots & \ell_{n1} \\ & \ell_{22} & \cdots & \ell_{n2} \\ & & \ddots & \vdots \\ O & & & \ell_{nn} \end{bmatrix} \qquad L=\begin{bmatrix} \ell_{11} & O & \\ \ell_{21} & \ell_{22} & \\ \vdots & \ddots & \\ \ell_{n1} & \cdots & \ell_{nn} \end{bmatrix}$$

توجه داریم که در این حالت برای ذخیره اطلاعات در مورد L,U تنها به نصف حالت قبلی حافظه احتیاج است و زمان محاسبات نیز به همین نسبت کم خواهد شد. برای محاسبه عناصر ماتریس L داریم:

$$\ell_{1,1}^2 = a_{1,1}$$

$$\ell_{1,1}\,\ell_{2,1} = a_{1,2}$$

$$\vdots$$

$$\ell_{1,1}\,\ell_{n,1} = a_{1,n}$$

$$\ell_{2,1}^2 + \ell_{2,2}^2 = a_{2,2}$$

$$\ell_{2n}^2\ell_{22}^2 = a_{22}$$

$$\ell_{21}\ell_{31}+\ell_{22}\ell_{32} = a_{23}$$

$$\vdots$$

و در حالت کلی خواهیم داشت:

$$\ell_{21}\,\ell_{n1} + \ell_{22}\ell_{n2} = a_{2n}$$

$$\ell_{r,1}^2 + \ell_{r,2}^2 + \ell_{r,3}^2 + \ldots + \ell_{r,r}^2 + \ldots + \ell_{r,n}^2 = a_{r,r}$$

$$\ell_{r,1}\,\ell_{s,1} + \ell_{r,2}\,\ell_{s,2} + \ldots + \ell_{r,r}\,\ell_{s,r} + \ldots + \ell_{r,n}\,\ell_{s,n} = a_{r,s}$$

در نتیجه رابطه کلی زیربدست می‌آید:

1- **Choleski Decomposition**

$$\ell_{r,r} = \left(a_{r,r} - \sum_{k=1}^{r-1} \ell_{r,k}^2\right)^{1/2}$$

$$\ell_{r,s} = \left(a_{r,s} - \sum_{k=1}^{r-1} \ell_{r,k}\ \ell_{s,k}\right)/\ \ell_{r,r}$$

$$r = 1,2,\ldots,\ n \qquad s = r+1, r+2,\ \ldots, n$$

a) – ممکن است که ماتریس L بصورت مختلط باشد.اما اگر ماتریس A بصورت «معین مثبت» باشد چنین حالتی رخ نخواهد داد.

b) –این عمل پایدار است اگر و فقط اگر ماتریس A بصورت «معین مثبت» باشد.پس در عمل روش کلی تجزیه را بکار برده و در هر مرحله ای که به عنصری از L بصورت مختلط رسیدیم عملیات را متوقف می‌کنیم و این نشان آنست که ماتریس A «معین مثبت» نیست.

بعد از محاسبه L برای حل معادله $Ax=b$ خواهیم داشت:

$$A = LL^T \qquad LL^T x = b$$

و از $Lz = b$ بردار z واز $L^T x = z$ بردار جواب x بدست می‌آید.

توضیح: گوئیم ماتریس A بصورت «معین مثبت»[1] است اگر:

i) زیر دترمینان‌های اصلی A همه مثبت باشند.

$$\det\ [a_{1,1}]\ ,\quad \det \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}\ ,\quad \det \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}\ ,\ \ldots$$

یا

ii) کلیه مقادیر ویژه A مثبت باشند.

یا

iii) $\forall\ y$ مقدار $y^T Ay$ مثبت باشد.

۶.۳ شرایط ناهنجار[2] دستگاه معادلات خطی

a) – چنانچه تغییر کوچکی در مقادیر سمت راست یک دستگاه معادلات خطی بوجود آید چه رخ می‌دهد؟

۱– **Positive Definite**

۲– *ill – condition*

$Ax=b$ (مقدار ثابت b)

δb تغییری است که در b بوجود آمده است.

$$A(x + \delta x) = b + \delta b$$

$$Ax + A\delta x = b + \delta b$$

$$A\ \delta x = \delta b$$

$$\delta x = A^{-1}\ \delta b$$

δx تغییری است که در اثر تغییرات b بوجود آمده است. در اینصورت اگر A^{-1} دارای عناصر بزرگی باشد مقدار δx تغییرات جواب، نیز بزرگ خواهند بود و مساله در شرایطی ناهنجار (ناجور) واقع می‌گردد.

b) - چنانچه تغییر کوچکی در مقادیر سمت چپ یک دستگاه معادلات خطی بوجود آید چه رخ خواهد داد؟ (مقدار ثابت A) $Ax=b$

اگر تغییر کوچکی در عناصرماتریس A بوجودآید، داریم:

$$(A+\ \delta A)(x\ +\ \delta x) = b$$

مسلماً تغییری در جواب دستگاه ایجاد خواهد شد. (δx)

برای محاسبه این تغییر می‌بینیم که: $Ax\ +\ A\ \delta x\ +\ \delta Ax\ +\ \delta A\ .\ \delta x\ =\ b$

یا

$$A\ \delta x\ +\ \delta A\ x\ +\ \delta A\ \delta x\ =\ 0$$

چنانچه از δx δA صرفنظر کنیم خواهیم داشت:

$$A\ \delta x\ +\ \delta Ax\ =\ 0$$

$$\delta x\ =\ -A^{-1}\ .\ \delta A\ .x$$

پس δx تغییری که در جوابها بوجود می‌آید، در صورتی بزرگ می‌باشد که A^{-1} دارای عناصر بزرگی باشد و در این حالت مساله در شرایطی ناهنجار واقع است.

پس از قسمت‌های (a),(b) نتیجه می‌شود که یک دستگاه معادلات خطی در صورتی در شرایط ناهنجار قرار دارد که A^{-1} دارای عناصری بزرگ باشد. اما این نتیجه جنبه کاربرد عملی نخواهد داشت، زیرا با توجه باینکه $A^{-1} = \frac{adj\ A}{det\ A}$ نتیجه می‌شود که:

A^{-1} دارای عناصر بزرگ است اگر $det\ A$ کوچک باشد. ولی در این روش، محاسبه دترمینان‌های مورد نظر و در نتیجه محاسبه A^{-1} و تعیین عناصر آن پر خرج است. (تعریف دیگری از شرایط ناهنجاری آنستکه نرم $\|A^{-1}\|$ بزرگ باشد) چنانچه داشته باشیم: $A=LU$

در حالتی‌که L یک ماتریس پائین مثلثی و U یک ماتریس بالا مثلثی است.

$$det\ A\ =\ det\ (LU)\ =\ det\ L\ .det\ U$$

$$det\ L=1$$

$$det\ A\ =\ u_{1\,1}\ .\ u_{2\,2}\ .\ u_{3\,3}\ \ldots\ u_{n\,n}$$

نتیجه می‌شود که یک دستگاه معادلات خطی، ناهنجار است اگر حاصلضرب عناصر قطر ماتریس بالا مثلثی U کوچک باشند.

توجه – روشهائی که تا اینجا مورد بررسی قرار گرفتند یعنی:

۱– روش حذفی گاوس– جردن ۲– تجزیه LU ۳– تجزیه چولسکی را روش‌های مستقیم می نامیم. اکنون روش‌های غیر مستقیم یا روش‌های تکراری را بررسی می‌نمائیم.

۱.۷.۳ روش ژاکوبی[1]

فرض کنیم معادلات را طوری بنویسیم که عناصر متعلق به قطر ماتریس A جدا شده باشند.

$$x_1 = \frac{1}{a_{1\,1}} (b_1 - a_{1\,2}\, x_2 - a_{1\,3}\, x_3 - \ldots - a_{1\,n}\, x_n)$$

$$x_2 = \frac{1}{a_{2\,2}} (b_2 - a_{2\,1}\, x_1 - a_{2\,3}\, x_3 - \ldots - a_{2\,n}\, x_n)$$

$$\vdots$$

$$x_n = \frac{1}{a_{n\,n}} (b_n - a_{n\,1}\, x_1 - a_{n\,2}\, x_2 - \ldots - a_{n\,n-1}\, x_{n-1})$$

چنانچه یک مقدار تقریبی برای بردار جواب x انتخاب نموده $[x_1, x_2, \ldots, x_n]^T$ و در طرف راست معادلات فوق جایگزین کنیم، مقدار تقریبی دیگری برای جواب بدست می‌آید. انتخاب اولیه را $x^{(0)}$ و جواب بدست آمده را $x^{(1)}$ نامگذاری می‌نمائیم. اگر مجدداً $x^{(1)}$ را در طرف راست معادلات فوق قرار دهیم جواب تقریبی دیگر که با $x^{(2)}$ نمایش می‌دهیم بدست می‌آید و ممکن است که با ادامه عملیات، $x^{(r)}$ بسمت جواب نهائی دستگاه معادلات خطی همگرا باشد.

پس می‌توان نوشت:

$$\lim_{r\to\infty} \underline{x}^{(r)} \to \underline{x}$$

$$x_1^{(k+1)} = \frac{1}{a_{1,1}} (b_1 - a_{1,2}x_2^{(k)} - a_{1,3}x_3^{(k)} - \ldots - a_{1,n}x_n^{(k)})$$

$$x_2^{(k+1)} = \frac{1}{a_{2,2}} (b_2 - a_{2,1}x_1^{(k)} - a_{2,3}x_3^{(k)} - \ldots - a_{2,n}x_n^{(k)})$$

$$\vdots$$

$$x_n^{(k+1)} = \frac{1}{a_{n,n}} (b_n - a_{n,1}x_1^{(k)} - a_{n,2}x_2^{(k)} - \ldots - a_{n,n-1}x_{n-1}^{(k)}$$

1 –Jocobi's Method

شرط همگرائی را در قسمت بعدی بیان و اثبات می‌کنیم.

مثال ۵.۳ مطلوبست حل دستگاه معادلات خطی زیر:

$$\begin{bmatrix} 7 & -4 & 0 \\ -4 & 12 & -6 \\ 0 & -6 & 14 \end{bmatrix} \underline{x} = \begin{bmatrix} 12 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{cases} 7x_1 - 4x_2 + 0x_3 = 12 \\ -4x_1 + 12x_2 - 6x_3 = 0 \\ 0x_1 - 6x_2 + 14x_3 = 0 \end{cases}$$

$$x_1^{(k+1)} = \frac{1}{7}\left(12 + 4x_2^{(k)}\right)$$

$$x_2^{(k+1)} = \frac{1}{12}\left(4x_1^{(k)} + 6x_3^{(k)}\right)$$

$$x_3^{(k+1)} = \frac{1}{14}\left(6x_2^{(k)}\right)$$

با انتخاب $\underline{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ عملیات را آغاز می‌کنیم:

$$\begin{cases} x_1^{(1)} = \frac{1}{7}(12+4) = 2.2857 \\ x_2^{(1)} = \frac{1}{12}(4+6) = .8333 \\ x_3^{(1)} = \frac{1}{14}(6) = .4286 \end{cases}$$

مجدداً عملیات را تکرار می‌کنیم:

$$\begin{cases} x_1^{(2)} = \frac{1}{7}(12+4\times.8333) = 2.1905 \\ x_2^{(2)} = \frac{1}{12}(4\times2.2857+6\times.4286) = .9762 \\ x_3^{(2)} = \frac{1}{14}(6\times2.8333) = .3571 \end{cases}$$

$$\begin{cases} x_1^{(3)} = \frac{1}{7}(12+4\times.9762) = 2.2721 \\ x_2^{(3)} = \frac{1}{12}(4\times2.1905+6\times.3571) = .9087 \\ x_3^{(3)} = \frac{1}{14}(6\times.9762) = .4184 \end{cases}$$

بهمین طریق عملیات را ادامه داده تا به جواب نهائی برسیم.

$x_1 = 2.264$ $\qquad$ $x_2 = .960$ $\qquad$ $x_3 = .411$

۲.۷.۳ روش گاوس - سیدل[1]

در اینجا همان روابط اساسی روش ژاکوبی را بکار می‌بریم اما در هر مرحله آخرین تقریبی را که برای هر x_i تاآن مرحله بدست آورده‌ایم، مورد استفاده قرار می‌دهیم:

$$\begin{cases} x_1^{(k+1)} = \dfrac{1}{a_{1,1}}\left(b_1 - a_{1,2}\,x_2^{(k)} - a_{1,3}\,x_3^{(k)} - \cdots - a_{1,n}\,x_n^{(k)}\right) \\ x_2^{(k+1)} = \dfrac{1}{a_{2,2}}\left(b_2 - a_{2,1}\,x_1^{(k+1)} - a_{2,3}\,x_3^{(k)} - \cdots - a_{2,n}\,x_n^{(k)}\right) \\ x_3^{(k+1)} = \dfrac{1}{a_{3,3}}\left(b_3 - a_{3,1}\,x_1^{(k+1)} - a_{3,2}\,x_2^{(k+1)} - a_{3,4}\,x_4^{(k)} - \cdots - a_{3,n}\,x_n^{(k)}\right) \\ \vdots \\ x_n^{(k+1)} = \dfrac{1}{a_{n,n}}\left(b_n - a_{n,1}\,x^{(k+1)} - a_{n,2}\,x_2^{(k+1)} - a_{n,3}\,x_3^{(k+1)} - \cdots - a_{n,n-1}\,x_{n-1}^{(k+1)}\right) \end{cases}$$

مثال ۶.۳ مثال قبلی را با روش فوق حل کنید:

$$\begin{cases} x_1^{(k+1)} = \dfrac{1}{7}\left(12 + 4\,x_2^{(k)}\right) \\ x_2^{(k+1)} = \dfrac{1}{12}\left(4x_1^{(k+1)} + 6x_3^{(k)}\right) \\ x_3^{(k+1)} = \dfrac{1}{14}\left(6x_2^{(k+1)}\right) \end{cases}$$

با $\underline{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ عملیات را شروع می‌کنیم:

$$\begin{cases} x_1^{(1)} = \dfrac{1}{7}(12+4) = 2.2857 \\ x_2^{(1)} = \dfrac{1}{12}(4\times 2.2857 + 6) = 1.2619 \\ x_3^{(1)} = \dfrac{1}{14}(6\times 1.2619) = 05408 \end{cases}$$

$$\begin{cases} x_1^{(2)} = \dfrac{1}{7}(12 + 4\times 1.2619) = 2.4354 \\ x_2^{(2)} = \dfrac{1}{12}(4\times 2.4354 + 6\times .5408) = 1.0822 \\ x_3^{(2)} = \dfrac{1}{14}(6\times 1.0822) = .4638 \end{cases}$$

1-Gauss - Seidel

$$\begin{cases} x_1^{(3)} = \frac{1}{7}(12+4\times1.0822) = 2.333 \\ x_2^{(3)} = \frac{1}{12}(4\times2.333 + 6\times .4638) = 1.0096 \\ x_3^{(3)} = \frac{1}{14}(6\times1.0096) = .4327 \end{cases}$$

در این مثال می‌بینیم که سرعت همگرائی در روش گاوس - سیدل نسبت به روش ژاکوبی بیشتر می‌باشد.

در اینجا لازم است که چند قضیه را متذکر شویم.

قضیه ۱.۳ اگر ماتریس A بصورت «قطری مسلط»[1] باشد.

روش ژاکوبی و گاوس - سیدل برای هر مقدار انتخابی اولیه $x^{(0)}$ همگرا می‌باشند.

$$\left[|a_{i,i}| \geq \sum_{j=1}^{n} |a_{i,j}| \qquad \text{برای هر } i \right]$$

بطور مثال ماتریس A بطور قطری مسلط می‌باشد.

$$A = \begin{bmatrix} 7 & -4 & 0 \\ -4 & 12 & -6 \\ 0 & -6 & 14 \end{bmatrix}$$

$$|7| \geq |-4| + |0| = 4$$
$$|12| \geq |-4| + |-6| = 10$$
$$|14| \geq |-6| + |0| = 6$$

اثبات: برای معادله iام از دستگاه $Ax=b$ داریم:

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ii}x_i + \ldots + a_{in}x_n = b_i$$

با فرض آنکه $a_{ii} \neq 0$ نتیجه می‌شودکه: $j \neq i$,

$$x_i = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{n} a_{ij}x_j \right]$$

انتخاب اولیه x در حالت کلی نسبت به جواب تحقیقی دارای اختلافی می‌باشد و اگر هر عضو آن یا x_j را بصورت $x_j + \varepsilon_j$ نشان دهیم، خواهیم داشت:

$$x_i + e_i = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{n} a_{ij}(x_j + \varepsilon_j) \right] , \quad j \neq i$$

1- Diagonality Dominant

$$x_i + e_i = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{n} a_{ij} x_j - \sum_{j=1}^{n} a_{ij}\, \varepsilon_j \right] \quad , \quad j \neq i$$

$$e_i = - \sum_{j=1}^{n} a_{ij}\, \varepsilon_j / a_{ii} \quad , \quad j \neq i$$

$$|e_i| = \frac{\left| \sum_{j=1}^{n} a_{ij} \varepsilon_j \right|}{|a_{ii}|} \leq \frac{\sum_{j=1}^{n} |a_{ij}|\, |\varepsilon_j|}{|a_{ii}|}$$

$$i = 1, 2, \ldots, n$$

$$j \neq i$$

$$\varepsilon = \text{Max} \{ |\varepsilon_1|, |\varepsilon_2|, \ldots, |\varepsilon_n| \}$$

$$|e_i| \leq \frac{\sum_{j=1}^{n} \varepsilon\, |a_{ij}|}{a_{ii}} \leq \frac{\sum_{j=1}^{n} |a_{ij}|}{|a_{ii}|}\, \varepsilon \quad , \quad j \neq i$$

اگر ماتریس ضرایب مجهولات قطری مسلط باشد، نتیجه می‌شود که:

$$|a_{ii}| \geq \sum_{j=1}^{n} |a_{ij}|$$

$$j \neq i$$

$$i = 1, 2, \ldots, n$$

$$K = \frac{\sum_{j=1}^{n} |a_{ij}|}{|a_{ii}|}$$

$$e \leq K \varepsilon \quad , \quad K < 1$$

در اینصورت در هر مرحله خطا نسبت به مرحله قبل کاسته می‌شود ودر ادامه عملیات در مرحله nام، خطا از مقدار خطای پیش بینی شده قابل اغماض کمتر خواهد شد و روش همگرا می‌شود.

قضیه ۲.۳ اگر ماتریس A از معادله $Ax=b$ معین مثبت باشد، روش گاوس - سیدل مستقل از مقدار اولیه x_0 بوده و همگراست.

قضیه ۳.۳ اشتین و روزنبرگ[1]

اگر عناصر غیر از قطر ماتریس در هر سطر یا ستون دارای علامتی مخالف با عنصر واقع بر قطر باشند، در اینصورت روش گاوس - سیدل بسیار سریعتر از روش ژاکوبی همگرا خواهد بود.

۸.۳ دستگاه معادلات غیر خطی (روش نیوتن)

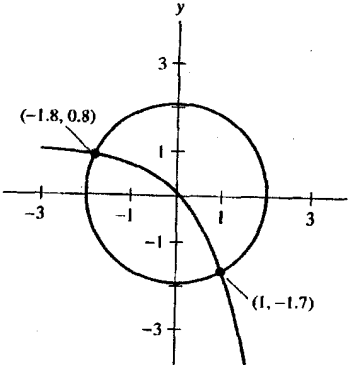
همانطور که قبلاً ذکر گردید، پیدا کردن جواب دستگاه معادلات غیر خطی بسیار مشکل‌تر از دستگاه معادلات خطی است. (در حقیقت، بعضی دستگاهها جواب حقیقی ندارند).

مثال ۷.۳ دستگاه معادلات غیر خطی زیر را بررسی کنید:

$$\begin{cases} x^2+y^2 = 4 \\ e^x+y = 1 \end{cases}$$

این دستگاه محل تقاطع دایره $x^2+y^2=4$ و خم $y = 1 - e^x$ می‌باشد که در شکل زیر نمایش داده شده است و جوابها نزدیک نقاط $(-1.8, 0.8)$ و $(1, -1.7)$ می‌باشند. **روش نیوتن** همانند معادلات خطی می‌تواند برای دستگاهها نیز بکار برده شود. با شکل زیر شروع می‌کنیم.

$$\begin{cases} f(x,y) = 0 \\ g(x,y) = 0 \end{cases}$$



فرض می‌کنیم $x = r$ و $y = s$ یک ریشه باشند و دو تابع را بعنوان سری تیلور در همسایگی نقطه (x_i, y_i) برحسب $(s - y_i)$ و $(r - x_i)$ بسط می‌دهیم، بطوریکه (x_i, y_i) نقطه نزدیک ریشه باشد:

1-Stein & Rosenbirg

$$\begin{aligned} f(r,s) = 0 &= f(x_i,y_i) + f_x(x_i,y_i)(r-x_i) \\ &\quad + f_y(x_i,y_i)(s-y_i) + \cdots \\ g(r,s) = 0 &= g(x_i,y_i) + g_x(x_i,y_i)(r-x_i) \\ &\quad + g_y(x_i,y_i)(s-y_i) + \cdots \end{aligned}$$

بعد از برش سری داریم:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f(x_i,y_i) \\ g(x_i,y_i) \end{bmatrix} + \begin{bmatrix} f_x(x_i,y_i) & f_y(x_i,y_i) \\ g_x(x_i,y_i) & g_y(x_i,y_i) \end{bmatrix} \begin{bmatrix} r-x_i \\ s-y_i \end{bmatrix}.$$

این رابطه را بصورت یک دستگاه معادلات می‌نویسیم:

$$\begin{bmatrix} f_x(x_i,y_i) & f_y(x_i,y_i) \\ g_x(x_i,y_i) & g_y(x_i,y_i) \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = -\begin{bmatrix} f(x_i,y_i) \\ g(x_i,y_i) \end{bmatrix},$$

این دستگاه را به روش حذفی گاوس حل می‌کنیم و سپس قرار می‌دهیم:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix},$$

تخمین بهبود یافته ریشه (r, s) را بدست می‌آوریم. این عمل را با قرار دادن i بجای $i+1$ تکرار می‌کنیم تا f و g به صفر نزدیک شوند.

برنامه *NLSYST* در پایان فصل برای حل این معادلات می‌باشد.

$$f(x,y) = 4 - x^2 - y^2 = 0$$

$$g(x,y) = 1 - e^x - y = 0$$

$$f_x = -2x \quad , \quad f_y = -2y$$

$$g_x = -e^x \quad , \quad g_y = -1$$

با $x_0 = 1$ و $y_0 = -1.7$ شروع می‌کنیم، معادلات زیر را حل می‌کنیم.

$$\begin{bmatrix} -2 & 3.4 \\ -2.7183 & -1.0 \end{bmatrix} \begin{bmatrix} \Delta x_0 \\ \Delta y_0 \end{bmatrix} = -\begin{bmatrix} 0.1100 \\ -0.0183 \end{bmatrix},$$

بعد از حل دستگاه داریم:

$\Delta x = 0.0043$ و $\Delta y = -0.0298$ و در نتیجه $x_1 = 1.0043$

و $y_1 = -1.7298$. اگر یکبار دیگر عمل را ادامه دهیم.

$y_2 = -1.729637$ و $x_2 = 1.004169$

و مقادیر تابع در (x_2, y_2) تقریباً $0.0000001-$ و $0.0000001-$ می‌باشند.

برای حل دستگاه معادلات فوق می‌توانستیم از معکوس ماتریس ضرایب استفاده کنیم.

$$\begin{bmatrix}\Delta x_0\\ \Delta y_0\end{bmatrix} = -\begin{bmatrix}-2 & 3.4\\ -2.7183 & -1.0\end{bmatrix}^{-1}\begin{bmatrix}0.110\\ -0.0183\end{bmatrix} = \begin{bmatrix}0.0043\\ -0.0298\end{bmatrix};$$

$$\begin{bmatrix}x_1\\ y_1\end{bmatrix} = \begin{bmatrix}1.0\\ -1.7\end{bmatrix} + \begin{bmatrix}0.0043\\ -0.0298\end{bmatrix} = \begin{bmatrix}1.0043\\ -1.7298\end{bmatrix};$$

$$\begin{bmatrix}\Delta x_1\\ \Delta y_1\end{bmatrix} = -\begin{bmatrix}-2 & 3.4\\ -2.7183 & -1.0\end{bmatrix}^{-1}\begin{bmatrix}0.000827\\ 0.000196\end{bmatrix} = \begin{bmatrix}-0.000133\\ 0.000165\end{bmatrix};$$

$$x_2 = 1.004167, \qquad y_2 = -1.729635;$$

$$f(x_2, y_2) = -0.000011, \qquad g(x_2, y_2) = -0.000002.$$

در صورتی که بتوانیم دستگاه معادلات غیر خطی را به یک معادله غیر خطی برحسب یکی از متغیرها تبدیل کنیم، می‌توانیم روش نیوتن را مطابق فصل قبل بکار برده و یکی از جوابها را بدست آوریم.

مثال قبلی را در نظر می‌گیریم:

$$4 - x^2 - y^2 = 0$$

$$1 - e^x - y = 0$$

برای $y = 1 - e^x$ و جایگذاری در معادله بدست می‌آوریم،

$$4 - x^2 - (1 - e^x)^2 = 0$$

$$3 - x^2 + 2e^x - e^{2x} = 0$$

سپس روش‌های تکراری در فصل دوم را برای حل بکار می‌بریم.

نتیجه روش نیوتن برای یک دستگاه از n معادله غیر خطی برابر است با:

$$\begin{bmatrix} f_{1x} & f_{1y} & f_{1z} & \cdots \\ f_{2x} & f_{2y} & f_{2z} & \cdots \\ f_{3x} & f_{3y} & f_{3z} & \cdots \\ \vdots & \vdots & \vdots & \\ f_{nx} & f_{ny} & f_{nz} & \cdots \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \\ \vdots \end{bmatrix} = - \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{bmatrix}$$

در (x_i, y_i, z_i) مقادیر توابع را حساب می‌کنیم. بعد از حل دستگاه خواهیم داشت:

$$x_{i+1} = x_i + \delta x_i, \qquad y_{i+1} = y_i + \delta y_i, \qquad z_{i+1} = z_i + \delta z_i$$

در یک برنامه کامپیوتر، بسهولت نمی‌توانیم مشتقات توابع را حساب کنیم. روش دیگر تقریب مشتقات نسبی با نمو کوچک متغیرها و محاسبه مجدد تابع می‌باشد.

$$(f_1)_x \doteq \frac{f_1(x+\delta, y, z, \ldots) - f_1(x, y, z, \ldots)}{\delta},$$

$$(f_1)_y \doteq \frac{f_1(x, y+\delta, z, \ldots) - f_1(x, y, z, \ldots)}{\delta},$$

$$\vdots$$

$$(f_i)_{x_j} \doteq \frac{f_i(x, y, z, \ldots, x_j+\delta, \ldots) - f_i(x, y, z, \ldots, x_j, \ldots)}{\delta}.$$

مشتقات تقریبی با روش تفاضل‌های محدود در فصل‌های بعدی بحث خواهد شد.

سودمندی روش نیوتن همگرایی مرتبه دوم آنست، (حداقل وقتی نزدیک به ریشه هستیم) اما مشکل، تعدّد محاسبهٔ توابع است، در دستگاه 2×2 مربوط به مثال قبلی، در هر مرحله شش تابع باید محاسبه شود و برای یک دستگاه 3×3، باید دوازده تابع حساب شوند و برای n معادله مشابه، تعداد n^2+n تابع حساب می‌شوند.

مثال ۸.۳ مسأله دیگری را بررسی می‌کنیم:

$$e^x - y = 0$$
$$xy - e^x = 0$$

برای این دستگاه داریم:

$$\begin{bmatrix} e^{x_i} & -1 \\ y_i - e^{x_i} & -x_i \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = -\begin{bmatrix} f(x_i, y_i) \\ g(x_i, y_i) \end{bmatrix}.$$

با فرض $x_0 = 0.95$ و $y_0 = 2.7$ برای حل این دستگاه با استفاده از معکوس ماتریس، داریم:

$$\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \end{bmatrix} = -\begin{bmatrix} 2.5857 & -1 \\ 0.1143 & 0.95 \end{bmatrix}^{-1} \begin{bmatrix} -0.1143 \\ -0.0207 \end{bmatrix},$$

بدست می‌آید

مجدداً ادامه می‌دهیم:

$$x_1 = 1.00029, \qquad y_1 = 2.71575;$$

$$\begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix} = -\begin{bmatrix} 2.5857 & -1 \\ 0.1143 & 0.95 \end{bmatrix}^{-1} \begin{bmatrix} 0.003325 \\ -0.002533 \end{bmatrix},$$

$$x_2 = 1.000048, \qquad y_2 = 2.718445.$$

برای $n = 2$، ماتریس مشتقات نسبی را اصلاح می‌کنیم بطوریکه:

$$\begin{bmatrix} \Delta x_2 \\ \Delta y_2 \end{bmatrix} = -\begin{bmatrix} 2.7184 & -1 \\ 0.00003 & 1.0000 \end{bmatrix}^{-1} \begin{bmatrix} -0.000033 \\ -0.000163 \end{bmatrix},$$

$$x_3 = 1.000000, \qquad y_3 = 2.718282.$$

که با جواب دقیق $(1, e)$ مطابقت دارد.

******

برنامه ۳.۱حل دستگاه معادلات خطی

این قسمت لیست برنامه برای روش تکراری گاوس - سیدل به نام *GSITRN.PAS* می‌باشد و یک دستگاه معادلات خطی با این برنامه حل شده است.

```
    Procedure    Gauss_Seidel_iteration_on_A_and_b(VAR    iter    :
INTEGER);
    (*
                    GSITRN.PAS

    ------------------------------------------------------------
      This procedure obtains the solution to N linear Equations by
          the GAUSS-SEIDEL   ITERATION   METHOD.   An   initial
approximation is
      sent to the procedure in the vector x. The solution, as
      approximated by the maximum change in any x component is less
      than TOL. If this cannot be accomplished in NITER iterations, a
      message is printed and the current approximation is returned.

      The equations are to be arranged so as to have the LARGEST
      VALUES ON THE DIAGONAL,    i. e. diagonal dominance.

    ------------------------------------------------------------
      GLOBAL VARIABLES

        a      - coefficient matrix with largest values on the diagonal.
        b      - right hand side vector.
        x      - on INPUT the initial approximation to the solution,
                 on OUTPUT the solution.

        n      - the number of equations.
        nITER  - limit to the number of iterations.
        iter   - actual number of iterations to get final answer.

        tol    - test value to stop iterating on the solution.

    ------------------------------------------------------------
    *)
    VAR
      save, xMAX    : REAL;
      i, j          : INTEGER;
      continue      : BOOLEAN;

    BEGIN
    (*
    ------------------------------------------------------------
```

```
  Make the diagonal elements equal to 1.0
*)
  WriteLn(outfile);  WriteLn(outfile);
  For i := 1 TO n DO                              Begin
    save := a[i,i];
    b[i] := b[i]/save;
    For j := 1 TO n DO                          Begin
      a[i,j] := a[i,j]/save;
      If (j<>i) Then
          Write(outfile,  -a[i,j]:10:4)
        ELSE
          Write(outfile,  '  ******')   End;
    WriteLn(outfile,  b[i]:15:4)                End;
WriteLn(outfile);  WriteLn(outfile);
(*
-----------------------------------------------------------------
  Now we perform the iterations. Store max change on x values
for
  testing against tol. WHILE tests the current number of
iterations
  to nITER.
*)
  iter := 1;  continue := TRUE;

  WHILE (ITER < nITER) AND continue DO              Begin
    xMAX := 0.0;
    For i := 1 TO n DO                      Begin
      save := x[i];
      x[i] := b[i];
      For j := 1 TO n DO
        IF j<>i
          THEN x[i] := x[i] - a[i,j]*x[j];

      IF (ABS(x[i] - save) > xMAX ) THEN
        xMAX := ABS(x[i]-save)          End;

    iter := iter + 1;
    continue := xMAX > tol;
    For j := 1 TO n DO
        Write(outfile,  x[j]:9:5);
    WriteLn(outfile);                           End;
END;
(*
-----------------------------------------------------------------
  PRINT OUT THE MATRIX AND THE RIGHT HAND SIDE
*)
Procedure write_out_matrix_and_right_hand_side;
VAR
  i,j  : INTEGER;
BEGIN
  WriteLN(outfile);  WriteLN(outfile);
  WriteLN(outfile,'  The given matrix a and its right hand side b are: ');
```

```
    WriteLN(outfile);
    For i := 1 TO n DO                    BEGIN
      For j := 1 TO n DO
        Write(outfile,  a[i,j]:5:1);
      Write(outfile,  ' ':10, b[i]:8:5);
      WriteLN(outfile)                    END;

    WriteLN;  WriteLN
  END;
  (*
  ------------------------------------------------------------
    PRINT  OUT  THE  FINAL  ANSWER
  *)
  Procedure  write_final_answer;
  VAR
    i : INTEGER;
  BEGIN
    WriteLN(outfile,'   The  solution  to  the  equations  is: ');
    WriteLN(outfile);
    For i := 1 TO n DO
      WriteLN(outfile,'   ':10, x[i]:8:5);
    WriteLN(outfile)
  END;
  (*
  ------------------------------------------------------------
    PRINT  MESSAGE  If THERE  IS NO CONVERGENCE
  *)
  Procedure  write_no_convergence;
  Begin
    WriteLn(outfile);
    WriteLn(outfile,'  THE  TOLERANCE   OF ', tol:10:7, ' NOT  MET
IN',
            iter:3,' iterations.');
    WriteLn(outfile)
  End;
```

برنامه ۲.۳حل دستگاه معادلات غیر خطی

در این قسمت لیست برنامه *NLSYST.PAS* برای حل دستگاه معادلات غیر خطی آورده شده است و دستگاه معادلات زیر حل شده است.

$$f(x, y) = x^2 + y^2 - 5 = 0$$

$$g(x, y) = y - e^x - 1 = 0$$

```
PROGRAM   nlsyst(INPUT,OUTPUT);
(*
 APPLIED  NUMERICAL   ANALYSIS
-------------------------------------------------------------
   Unlike the Fortran version in the current text this program
   computes the numerical partial derivative only.
-------------------------------------------------------------
   This program solves the pair of equations:
     f(x,y) = x*x + y*y - 5 = 0,
     g(x,y) = y - exp(x) - 1 = 0.
   The starting value (2,4) is supplied along with the values
   given in the CONST section of the program.
*)

CONST
   xtol     = 1.0E-6;  (*                              *)
   ftol     = 5.0E-5;  (*   STOPPING  CRITERIA          *)
   maxcount = 30;      (*                              *)
   delta    = 0.000025; (* to compute numerical derivatives  *)
TYPE
   rhs = ARRAY[1..10]  OF REAL;
   vector = ARRAY[1..10]  OF INTEGER;
   matrix = ARRAY[1..10,1..10]  OF REAL;
VAR
   x, (* ARRAY  to hold the X-values. Contains the original
guesses. *)
   f  (* ARRAY  to hold the function values.
 *)
                : rhs;
   ipvt   (* used in ELIM  to keep  track of row changes for SOLVE
 *)
                : vector;
   value,temp3    : REAL;
   i,nequs,maxit  : INTEGER;
(*$I ELIM.PAS*)
(*
-------------------------------------------------------------
                COMPUTE  THE  NONLINEAR   EQUATIONS
*)
PROCEDURE   evalfcns( VAR  x,f : rhs);
(*
```

```
    This procedure evaluates the functions, F(X), we
    are trying to solve.
    f: the array that holds the values of the funtions.
*)
Begin
     f[1] := sqr(x[1]) + sqr(x[2]) - 5.0;
     f[2] := x[2] - exp(x[1]) - 1.0
End; (*evalfcns *)
(*
        PROCEDURE   solvNLS
*)
PROCEDURE   solvNLS(VAR   newx : rhs);
(*
    solNLS solves a system of nonlinear equations by Newton's
    method. The Partial derivatives of the functions are esti-
    mated by difference quotients when a variable is perturbed
    by an amount equal to delta. This is done for each variable
    in each function.
    The increments to improve the estimates for the x-values
    are computed from a system of equations using PROCEDURE
    elim.
    newx: this returns the solution to the system of equations.
*)
VAR
    tempx, tempf    : rhs;
    a               : matrix;
    i,j,icol,jrow,
    count           : INTEGER;
    ftolmet,
    xtolmet,
    tolmet          : BOOLEAN;
Begin (* solvNLS *)
count := 1; tolmet := FALSE;
WHILE (NOT tolmet) AND (count <= maxcount) Do Begin
   ftolmet := TRUE; xtolmet := TRUE;
   evalfcns(newx,f);
(*
-------------------------------------------------------------
        PRINT OUT CURRENT X AND F(X) VALUES
*)
WriteLn;
WriteLn('    X1 = ', newx[1],' ':5, ' X2 = ', newx[2]);
WriteLn('    F1 = ', f[1],  ' ':5, ' F2 = ', f[2]);
WriteLn;
(*
-------------------------------------------------------------
        CHECK ON SToPPING CRITERIA
*)
   FOR i := 1 To nequs Do IF (ABS(newx[i]) > xtol)
                    Then xtolmet := FALSE;
   FOR i := 1 To nequs Do IF (ABS(f[i]) > ftol)
                    Then ftolmet := FALSE;
```

```
      FOR i := 1 To nequs Do tempx[i] := newx[i]; (* SToRE IN
    TEMPX *)
     (*
     ------------------------------------------------------------
           FIND PARTIALS BY FORWARD DIFFERENCES
     *)
       FOR icol := 1 To nequs Do Begin
         tempx[icol] := newx[icol] + delta;
         evalfcns(tempx,tempf);
         FOR jrow := 1 To nequs Do (* COLUMN OF PARTIALS *)
               a[jrow,icol] := (tempf[jrow]-f[jrow])/delta ;
         tempx[icol] := newx[icol] End;
       elim(a, nequs,ipvt); (* LU DECOMPOSITION OF F'(X) *)
       FOR i := 1 To nequs Do tempx[i] := - f[i];
       solve(a,nequs,ipvt,tempx); (* SOLVE F'(X) DELX = - F(X) *)
    (*
    ------------------------------------------------------------
           UPDATE THE X VALUES
    *)
       FOR i := 1 to nequs Do newx[i] := newx[i] + tempx[i];
       count := count + 1; tolmet := ftolmet OR xtolmet
      End; (* WHILE loop *)
      WriteLn;
      WriteLn('      ----------------------------------------');
      WriteLn;
      IF (ftolmet) Then WriteLn(' ':7, ' FToL CONDITION MET IN ',
                        count:2, ' ITERATIONS.')
        Else IF (xtolmet) Then WriteLn(' ':7, ' XToL CONDITION
MET',
                        count:2, ' ITERATIONS.')
           Else WriteLn(' MAX NUMBER OF ITERATIONS
REQUIRED': 35)
     End; (* solvNLS *)
     Begin (* main *)
    (*
       INITIALIZE DATA
    *)
    nequs := 2; x[1] := 2.0; x[2] := 4.0;
    solvNLS(x);
    WriteLn;
    End. (* PROGRAM *)
```

برنامهٔ ۳.۳حل دستگاه معادلات به روش تجزیه *LU*

لیست این برنامه به نام *PDEC.C* در زیر آمده است. با برنامه فرعی *ELIM* ماتریس ضرایب را به صورت *LU* تجزیه می‌کنیم و سپس با برنامه فرعی *SOLVE* دستگاه تجزیه شده را حل می‌کنیم.

```
*
   ******************************
   **                          **
   **        PDEC.C              **
   **                          **
   ******************************

  APPLIED  NUMERICAL  ANALYSIS
  ------------------------------------------------------------
     This program solves a system of equations using Gaussian
  elimination with partial pivoting and back substitution according to the
  algorithm given in Section 3.
     The algorithm is implemented in two procedures: ELIM and
SOLVE.
     The first procedure returns the LU decompostion of the
  coefficient matrix, and the second solves the system using both
  forward and back substitution.
  */
  #include  <stdio.h>
  #include  <stdlib.h>
  #include  <math.h>

  float     **a,   /* set the matrix a as a pointer to a pointer  */
    **a1,      /* set the matrix a1 as a pointer to a pointer  */
    *b1;       /* set the matrix b1 as a pointer  */
  int *ipvt1;     /* set the matrix ipvt1 as a pointer to integer  */
  int   n1, i, j;
  main()                                    /* main proram
  */
  {
  printf("Please enter the size of your matrix: ");
  scanf("%i", &n1);
  a1 = malloc(sizeof(float  *) * n1);          /* create the
  matrix */
  for (i = 0; i < n1; i++)
    a1[i] = calloc(n1, sizeof(float));
  printf("\n\n");
  printf("Please enter the element of the matrix a.\n");
  for (i = 0; i < n1; i++)
  {
   for (j = 0; j < n1; j++)
```

```
  {
    printf(" a1[ %i][ %i] = ", i, j);
    scanf("%f", &a1[i][j]);
  }
}
b1 = malloc(sizeof(int)  * n1);
printf("\n\n");
printf("Please enter the element of the matrix b.\n");
for (i = 0; i < n1; i++)
{
  printf(" b1[ %i] = ", i);
  scanf("%f", &b1[i]);
}
ipvt1 = malloc(sizeof(int)  * n1);

 a = a1;
printf("\n");
/*
-------------------------------------------------------------------
    Write out the matrix of coefficients,a, and the right hand
    side.
*/
    if ( n1 > 7 )
    {
       printf("\t\t\t\tTHE  MATRIX  A\n");
       printf("\t\t\t\t------------\n\n");
    }
    else if ( n1 > 3 )
    {
      printf("\t\t\tTHE  MATRIX  A\n");
      printf("\t\t\t------------\n\n");
    }
    else if ( n1 <= 3 )
   {
      printf("\tTHE  MATRIX  A\n");
      printf("\t------------\n\n");
   }

    print_matrix_a  (&a, n1);

    printf("\n\n");
    printf("\t\t\tTHE  RIGHT  HAND  SIDE  B\n");
    printf("\t\t\t---------------------\n");
     print_matrix_b  (b1, n1);
    printf("\n\n");

/*
-------------------------------------------------------------------
    We find the LU decomposition  of the matrix A.
*/
   elim(&a, n1, ipvt1);
/*
```

```
-------------------------------------------------------------
   Procedure solve now gets solution by forward/back
   substitution.
 */
    solve(&a, n1, ipvt1, b1);
/*
-------------------------------------------------------------
   Write out the LU decomposition of A.

*/
   if ( n1 > 7 )
   {
     printf("\n\t\t\t THE LU DECOMPOSITION OF A\n");
     printf("\t\t\t ----------\n\n");
   }
   else if ( n1 > 3 )
   {
     printf("\n\tTHE LU DECOMPOSITION OF A\n");
     printf("\t----------\n\n");
   }
   else if ( n1 <= 3 )
   {
     printf("\n THE LU DECOMPOSITION OF A\n");
     printf(" ----------\n\n");
   }
   print_matrix_a (&a, n1);
   printf("\n\n");
   printf("\t\t\t THE SOLUTION VECTOR \n");
   printf("\t\t\t ---------- \n");
   print_matrix_b (b1, n1);
   printf("\n\n");
}
print_matrix_a (float ***a, int n2)
{
  int i, j;
   for ( i = 0; i < n2; i++ )
   {
     for ( j = 0; j < n2; j++ )
     {
       printf(" %8.4f",(*a)[i][j]);
     }
     printf("\n");
   }
}
print_matrix_b (float b[ ], int n)
{
  for ( i = 0; i < n; i++ )
  {
     printf("\t\t %10.4f ", b[i]);
     printf("\n");
  }
}
```

```
elim(float  ***a, int n, int ipvt[ ])
/*
-----------------------------------------------------------------
   ELIM:  this procedure  solves a set of linear equations  and gives
   an LU  decomposition  of the coefficient  matrix. The Gaussian
   elimination  method  is used, with partial pivoting  and back
   substitution.

   INPUT:   a  - the coefficient  matrix
            n  - the number  of equations
            ****************************
   OUTPUT:  a  - the LU  decomposition  of the matrix a
            ipvt  - a vector  containing  the order of the rows
                    of the rearranged  matrix due  to pivoting.
*/
{
   float save, ratio, value, det;
   int   i, ipvtemp,  nMinus1,  iPlus1, j, l, kcol, jcol, jrow,
         tempipvt;
/*
-----------------------------------------------------------------
   Begin  the  LU  decomposition
*/
   det  =  1.0;
   nMinus1  =  n-1;
   for ( i  =  0; i  <  n; i++  )
      ipvt[i]  =  i;
   for ( i  =  0; i  <  nMinus1;  i++  )
    {
      iPlus1  =  i+1; ipvtemp  =  i;
      for ( j  =  iPlus1; j  <  n; j++  )          /* find  the  */
      if (fabs((*a)[ipvtemp][i])  <  fabs((*a)[j][i]))   /* pivot  */
      ipvtemp  =  j;                                /* row  */
      if (ipvtemp  !=  i)           /* interchange  */
       {
         tempipvt  =  ipvt[i];
         ipvt[i]  =  ipvt[ipvtemp];          /* rows if a[i,i] */
         for ( jcol  =  0; jcol  <  n; jcol++  )          /* is not  the
max */
          {
            save  =  (*a)[i][jcol];          /* in column.  */
            (*a)[i][jcol]  =  (*a)[ipvtemp][jcol];
            (*a)[ipvtemp][jcol]  =  save;
          }                      /*  end  of for jcol loop  */
         ipvt[ipvtemp]  =  tempipvt;
         det  =  -det;
       }                                      /* end  of if
statement  */

/*
-----------------------------------
```

تمرینات فصل سوم

۱- روش حذفی گاوس با تغییر در سطرهارا برای پیدا کردن جواب معادلات خطی زیر بکار برید.

$$\begin{bmatrix} 2.4 & 6.6 & -2.7 \\ -2.1 & -2.7 & 5.9 \\ 3.0 & 5.0 & -4.0 \end{bmatrix} \quad \underline{x} \quad = \quad \begin{bmatrix} -.45 \\ 1.2 \\ .28 \end{bmatrix}$$

۲- نشان دهید که تعداد اعمال جمع و ضرب و تقسیم برای محاسبه دستگاه معادلات خطی با روش حذفی گاوس برابر است با:

برای تقسیم $\delta = \frac{n(n+1)}{2}$ و برای جمع و ضرب $\alpha = \beta = \frac{n(n-1)(2n+5)}{6}$

و در نتیجه زمان لازم برای این عملیات برابر است با:

$t = 4\,\alpha + 10\,\beta + 22\,\delta$ (t برحسب میلیونیم ثانیه)

۳- تجزیه LL^T چولسکی را برای ماتریس A بکار برده و جواب معادلات $Ax=b$ را بدست آورید.

$$\underline{A} = \begin{bmatrix} 4 & 2 & 6 \\ 2 & 10 & 6 \\ 6 & 6 & 14 \end{bmatrix} \qquad b = \begin{bmatrix} 2.6 \\ 4.0 \\ 6.0 \end{bmatrix}$$

جواب: $x_1=0.1,\ x_2=0.2,\ x_3=0.3$

۴- دستگاه معادلات زیر را به روش ژاکوبی و گاوس - سیدل می‌توان حل نمود.
جواب تقریبی را با تشکیل روابط هر یک و چهار بار تکرار بدست آورید.

$$\begin{bmatrix} 3 & 7 & 5 & 16 \\ 1 & 17 & 13 & 2 \\ 9 & 4 & 16 & 1 \\ 6 & 2 & 1 & 2 \end{bmatrix} \quad \underline{x} \quad = \quad \begin{bmatrix} 4.3 \\ 6.3 \\ 5.0 \\ 1.4 \end{bmatrix}$$

جواب: $x_1=0.1,\ x_2=0.2,\ x_3=0.2,\ x_4=0.1$

۵- دستگاه معادلات زیر را در نظر می‌گیریم

$$3x - 6y + 2z = 15,$$
$$-4x + y - z = -1.$$
$$x - 3y + 7z = 26.$$

a. روش حذفی گاوس را برای حل دستگاه بکار برید.
b. روش تجزیه LU را برای حل دستگاه بکار برید.

جواب: $x=-1,\ y=-2,\ z=3$

۶- دستگاه معادلات زیر را با استفاده از ماشین حساب و روش حذفی گاوس یا اجرای یک برنامه کامپیوتری مناسب حل کنید.

$$2.51x_1 + 1.48x_2 + 4.53x_3 = 0.05$$
$$1.48x_1 + 0.93x_2 - 1.30x_3 = 1.03$$
$$2.68x_1 + 3.04x_2 - 1.48x_3 = -0.53$$

جواب: تا شش رقم با معنی صحیح برابرند با:

$$x_1=1.45310 \quad , x_2 = -1.58919 \, , \quad x_3=-0.27489.$$

۷- دستگاه معادلات زیر را حل کنید:

$$9x + 4y + z = -17$$
$$x - 2y - 6z = 14$$
$$x + 6y = 4$$

a. روش گاوس - ژاکوبی را بکار برید.

b. روش گاوس - سیدل را بکار برید.

کدام سریعتر به جواب می‌رسند؟

جواب: $x=-2, \; y=1, \; z=-3$

۸- دستگاه معادلات زیر را در نظر می‌گیریم

$$6x_1 - 3x_2 + x_3 = 11,$$
$$2x_1 + x_2 - 8x_3 = -15,$$
$$x_1 - 7x_2 + x_3 = 10$$

a. جواب دستگاه را با استفاده از یک روش تکراری با شروع از $(0,0,0)$ حل کنید.

b. با اجرای برنامه کامپیوتری مناسب جواب دستگاه را بدست آورید.

جواب: ماتریس ضرایب باید قطری مسلط باشد.

$$\begin{bmatrix} 6 & -3 & 1 \\ 1 & -7 & 1 \\ 2 & 1 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 10 \\ -15 \end{bmatrix}$$

$$\begin{cases} x_1 = \frac{1}{6}(11 + 3x_2 - x_3) \\ x_2 = -\frac{1}{7}(10 - x_1 - x_3) \\ x_3 = \frac{1}{8}(15 + 2x_1 + x_2) \end{cases}$$

$$x^{(0)}=\begin{bmatrix}0\\0\\0\end{bmatrix}\quad x^{(1)}=\begin{bmatrix}\frac{11}{6}\\-\frac{10}{7}\\\frac{15}{8}\end{bmatrix}=\begin{bmatrix}1.82\\-1.42\\1.87\end{bmatrix}$$

$$x=\begin{bmatrix}1\\-1\\2\end{bmatrix}$$

عملیات را ادامه داده و جواب نهایی برابر است با:

۹ – دو محل تلاقی نزدیک به مبداء دو خم زیر را بدست آورید.

$$\begin{cases}x^2+x-y^2=1,\\ y-\sin(x^2)=0\end{cases}$$

$$\begin{cases}x=\pm(1+y^2+0.25)^{\frac{1}{2}}\\ y=\sin(x^2)\end{cases}$$

جواب:

$$(x,y)=(0.72595,0.5095),(x,y)=(-1.6701,0.34513)$$

۱۰ – دستگاه معادلات زیر را به روش نیوتن حل کنید

$$\begin{cases}x^3+3y^2=21,\\ x^2+2y+2=0.\end{cases}$$

جواب:

$$\begin{cases}x=(21-3y^2)^{1/3}\\ y=-½(2+x^2)\end{cases}$$

$(x,y) = (1.64303, -2.34978)$

$(x,y) = (-2.07929, -3.116174)$

۱۱ – دستگاه معادلات $Ax=b$ را بررسی کنید بطوریکه

$$A=\begin{bmatrix}3.01 & 6.03 & 1.99\\ 1.27 & 4.16 & -1.23\\ 0.987 & -4.81 & 9.34\end{bmatrix},\qquad b=\begin{bmatrix}1\\1\\1\end{bmatrix}.$$

a. به روش حذفی گاوس و تا سه رقم با معنی صحیح جواب دستگاه را بدست آورید.

b. با اجرای برنامه کامپیوتری با دقت معمولی جواب دستگاه را بدست آورید.

c. جواب دستگاه را با استفاده از دقت مضاعف بدست آورید.

d. فرض کنید دقت عناصر ماتریس A در بعضی عناصر آن تغییر کند و بجای $a_{11}=3.01$

عدد 3 قرار گیرد و بجای $a_{31}=0.987$ عدد 0.99 باشد. اکنون جواب را به دست آورید.

جواب‌های قسمت d,c,b,a را بدست آورید.

جواب: $(a$ $(-238,\ 94.5,\ 73.9)$

$(b$ $(-118,47.1,\ 37.0)$

$(c$ $(1592.71,\ -631.956,\ -493.653)$

$(d$ $(119.537,\ -47.1426,\ -36.84402)$

تغییرات x زیاد است و این دستگاه ناهنجار (ناجور) $ill-condition$ می‌باشد.

۱۲ - ماتریس هیلبرت مرتبه چهار یک مسأله حالت ناجور (ناهنجار) را نشان می‌دهد.

$$H=\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}.$$

دستگاه $Hx=b$ با انتخاب $b^T=[25/12,\ 77/60,\ 27/60,\ 319/420]$ را در نظر می‌گیریم.

a. دترمینان H را حساب کنید و نشان دهید ماتریس تقریباً منفرد است.

b. جواب دستگاه را تا سه رقم این در انجام محاسبات با ماشین حساب بدست آورید. و بار دیگر داده‌ها را در کامپیوتر تا هفت رقم با معنی مورد استفاده قرار می‌گیرند.

جواب دستگاه را با اجرای یک برنامه کامپیوتری بدست آورید.

جواب $(a:$ $det(H)=1.65E-5$

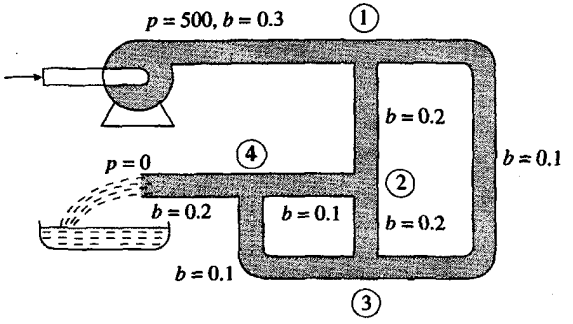
$(b$ $[0.988,\ 1.42,\ -0.428,\ 2.10]$

محاسبات تا سه رقم با معنی صحیح می‌باشد و دقت جواب مناسب نیست تا پنج رقم دقت جواب برابر است با: $[1.0000,\ 0.99995,\ 1.0017,\ 0.9990]$

۱۳ - جریان متلاطم مایع در یک شبکه در شکل زیر را در نظر می‌گیریم. سرعت جریان از یک گره تا گره دیگر متناسب با ریشه دوم تفاضل فشارها در دو گره می‌باشد. (جریان مایع از جریان الکتریکی در شبکه مجزا بوده و معادلات غیر خطی نتیجه می‌دهد.)

در کانال شکل زیر فشار را پیدا کنید.

مقادیر b نمایش فاکتور ضریب هدایت در رابطه $v_{ij}=b_{ij}(p_i-p_j)^{1/2}$ می‌باشند.



گره ۱: $0.3\sqrt{500 - p_1} = 0.2\sqrt{p_1 - p_2} + 0.1\sqrt{p_1 - p_3};$

گره ۲: $0.2\sqrt{p_1 - p_2} = 0.1\sqrt{p_2 - p_4} + 0.2\sqrt{p_2 - p_3};$

گره ۳: $0.1\sqrt{p_1 - p_3} + 0.2\sqrt{p_2 - p_3} = 0.1\sqrt{p_3 - p_4};$

گره ۴: $0.1\sqrt{p_2 - p_4} + 0.1\sqrt{p_3 - p_4} = 0.2\sqrt{p_4 - 0}.$

که یک دستگاه چهار معادله و چهار مجهول غیر خطی می‌باشد.

جواب: مقادیر صحیح جواب برابرند با: *[425,351, 346, 167]*

فصل ۴

مقادیر ویژه و بردارهای ویژه

موضوعات این فصل

* مقدمه
* یادآوری چند نکته درباره ماتریسها
* روش لوریه فادیو
* روش توانی
* روش تکراری معکوس
* روش صفرکردن مقادیر ویژه ماتریس
* روش تبدیل ماتریس برای تعیین مقادیر ویژه
* روش ژاکوبی
* روش گیونز
* معکوس کردن ماتریس
* برنامه روش توانی به زبان فرترن بنام PPOWER.C
* تمرینات فصل چهارم

کارل گوستاو ژاکوبی
Carl Gustav Jacobi
(۱۸۵۱-۱۸۰۴)

ژاکوبی در سال ۱۸۰۴ در پوتسدام به دنیا آمد. وی در دانشگاه برلین تحصیل کرد و در سال ۱۸۲۵ به اخذ درجه دکترا نایل آمد. محققین برجسته ریاضی بندرت معلم برجسته ریاضی از کار درآمدند، ولی ژاکوبی یکی از استثناها بود. بی‌چون و چرا او بزرگترین معلم ریاضی دانشگاه نسل خود بوده است. مشهورترین تحقیقات او در ریاضیات در زمینه توابع بیضوی است، توسط او بود که کلمه «دترمینان» پذیرش نهایی خود را یافت. وی همچنین در نظریهٔ اعداد، نظریه معادلات دیفرانسیل معمولی، معادلات با مشتقات جزیی، حساب تغییرات سهم داشت. به گفته وی «مقصود واقعی علم عظمت روح انسان است».

مقدمه

مثال ۱.۴ فرض کنیم یک دستگاه از اجسام مرتبط داشته باشیم بطوریکه A_1, B_1 حالت تعادل آنها باشد. اکنون فرض می‌کنیم این اجسام در دو جهت تغییر کوچکی یابند بطوریکه x_2,x_1 فاصله جدید آنها از حالت تعادل باشد.

می‌پذیریم که نیروها متناسب با x_2,x_1 عمل می‌نمایند تادو جسم ما به حالت تعادل اولیه درآیند. معادله حرکت این دو جسم چنین است.

$$\begin{cases} \dfrac{d^2x_1}{dt^2} = a_{11}\,x_1 + a_{12}\,x_2 \\ \dfrac{d^2x_2}{dt^2} = a_{21}\,x_1 + a_{22}\,x_2 \end{cases} \qquad (1)$$

جواب دستگاه فوق بشکل زیر خواهد بود.

$$\begin{cases} x_1 = \alpha\ Cos(\omega t+\phi) \\ x_2 = \beta\ Cos(\omega t+\phi) \end{cases} \qquad (2)$$

اگر این جواب را در دستگاه (۱) جایگزین کنیم خواهیم داشت:

$$\begin{cases} -\alpha\omega^2\ Cos(\omega t+\phi) = a_{11}\,\alpha\ Cos(\omega t+\phi)+a_{12}\,\beta\ Cos(\omega t+\phi) \\ -\beta\omega^2\ Cos(\omega t+\phi) = a_{21}\,\alpha\ Cos(\omega t+\phi)+a_{22}\,\beta\ Cos(\omega t+\phi) \end{cases} \qquad (3)$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = -\omega^2 \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \qquad (4)$$

در اینجا ω فرکانس نوسان برای x_1 و x_2، ϕ اختلاف فاز می‌باشد و α , β نیز دامنهٔ این نوسانها می‌باشند. معادلهٔ (4) «معادلهٔ جبری مقدار ویژه ماتریس» نامیده می‌شود و فرم کلی آن چنین است:

$$Ax = \lambda x \qquad (5)$$

که در آن A یک ماتریس $(n\times n)$ و x یک بردار $(n\times 1)$ و λ مقداری ثابت است. حال در نظر داریم تا معادلاتی مانند (5) را حل نمائیم و x (بردار ویژه)، λ (مقدار ویژه) راکه مربوط به ماتریس A می‌باشند بدست آوریم.

مشاهده می‌شود که $x = O$ یک جواب برای معادلهٔ $Ax = \lambda x$ می‌باشد، اما این جواب مطلوبی نیست زیرا که این جواب متناظر با عدم تغییر از حالت تعادل است و ما واقعاً به جواب غیر صفر نیازمندیم.

فرض کنیم معادله (5) را بصورت زیر بنویسیم:

$$(A - \lambda I)x = 0 \qquad (6)$$

معادلهٔ (6) نمایش یک دستگاه معادلات خطی همگن می‌باشد و در صورتی دارای جواب غیر صفراست که:

$$det(A - \lambda I) = 0 \qquad (7)$$

پس از بسط دترمینان داریم:

$$\lambda^n + b_{n-1}\lambda^{n-1} + \ldots + b_1\lambda + b_0 = 0 \qquad (8)$$

معادله (8) در حالت کلی دارای n جواب است، $\lambda_1, \lambda_2, \ldots, \lambda_n$ ، که آنها را مقادیر ویژه ماتریس A می‌نامیم.

اکنون بازاء هر مقدار λ معادله همگن (8) دارای جواب x خواهد بود و n بردار $x_1, x_2, \ldots, x_n$ متناظر با مقادیر ویژه $\lambda_1, \lambda_2, \ldots, \lambda_n$ بردارهای ویژه نامیده می‌شوند.

مثال ۲.۴ مطلوبست محاسبهٔ مقادیر ویژه و بردارهای ویژهٔ ماتریس A

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

در این معادله:

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} x = \lambda x$$

بردار ویژه $x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ یک جواب است که مورد نظر ما نیست.

پس از معادله فوق داریم:

$$Ax = \lambda x$$

$$(A - \lambda I)x = 0$$

$$\begin{bmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{bmatrix} x = 0$$

بنابراین باید داشته باشیم:

$$\det \begin{bmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{bmatrix} = 0$$

$$(1 - \lambda)(4 - \lambda) - 4 = 0$$

$$\lambda^2 - 5\lambda = 0$$

مقادیر ویژه عبارتند از:

$$\lambda_1 = 0$$

$$\lambda_2 = 5$$

برای بدست آوردن بردارهای ویژه داریم:

$$\lambda_2 = 5$$

$$(A - \lambda I)x = 0$$

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

x_1 بردار ویژه مربوط به مقدار ویژه λ_1 بازاء هر مقدار s می‌باشد.

$$\begin{cases} r + 2s = 0 \\ 2r + 4s = 0 \end{cases} \qquad \text{یا } r = -2s \qquad x_1 = \begin{bmatrix} -2s \\ s \end{bmatrix}$$

$$x_1 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} S$$

$$\lambda_2 = 5$$

$$(A-\lambda I)\underline{x} = 0 \rightarrow \begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} t \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} t$$

برای هر مقدار t ، x_2 بردار ویژه است .

فرض کنیم می‌خواهیم مقادیر ویژه و بردارهای ویژه را مطابق روش فوق برای یک ماتریس مرتبه n مانند A بدست آوریم در اینصورت:

i) از بسط $det(A - \lambda I)$ معادله چند جمله‌ای از درجه n را تشکیل می‌دهیم.

ii) جواب‌های این معادله $\lambda_1 \quad \lambda_2 \ldots \lambda_n$ را که از درجه nام می‌باشد بدست می‌آوریم.

iii) برای هر مقدار ویژه مانند $\lambda = \lambda_r$ از حل n معادله خطی، بردارهای ویژه را بدست می‌آوریم. (برای هر x_r) $\qquad (A - \lambda_r I)\, x_r = 0$

زمانی‌که درجه معادله مشخصه پائین باشد ازروش لوریه – فادیو با تشکیل معادله مشخصه ماتریس، مقادیر ویژه ماتریس را بدست می‌آوریم. روشهائی نیز وجود دارند که بدون تشکیل معادله مشخصه ماتریس می‌توان مستقیماً بعضی یا کلیه مقادیر ویژه را بدست آورد. در بخش‌های بعدی سعی می‌نمائیم که روشهائی دقیق و سریع را مورد مطالعه قرار دهیم.

مثال ۳.۴ در مسایل الاستیسیته (کش سان) و ارتعاش که از جمله کاربردهای معادلات دیفرانسیل در فیزیک مدرن هستند، به مسایل مقادیر مرزی یا مسایل مقدار ویژه برخورد می‌کنیم. (همچنین مسایلی در آمار به چنین حالتی تبدیل می‌شوند).

معادلهٔ دیفرانسیل همگن مرتبه دوم با شرایط مرزی زیر را در نظر می‌گیریم.

$$\frac{d^2y}{dx^2} + k^2 y = 0 \quad y(0)=0 \;,\; y(1) = 0$$

ابتدا مسأله را بصورت غیر عددی حل می‌کنیم. k^2 یک پارامتر است و جواب عمومی معادله دیفرانسیل برابر است با:

$$y = a \sin(kx) + b \cos(kx)$$

بآسانی تحقیق می‌شود که این جواب در معادله صدق می‌کند و شامل دو مقدار ثابت $a,\ b$ می‌باشد، که با استفاده از شرایط مرزی بدست می‌آیند.

$$0 = a \sin(0) + b \cos(0)$$

$$0 = a \sin(k) + b \cos(k)$$

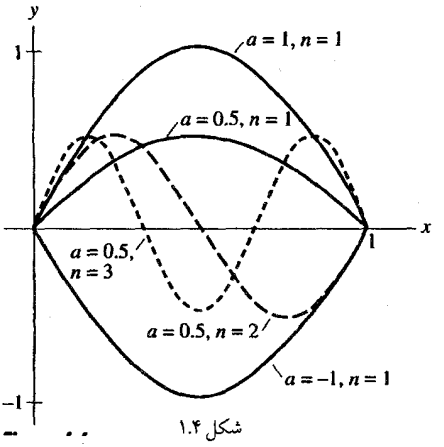
یک جواب معادله $a=0,\ b=0$ می‌باشد که مورد نظر ما نیست و جواب دیگر $b=0$

$$sin(k) = 0 \qquad \text{و} \qquad k=\pm\ n\ \pi \quad \text{و} \quad n = 1,2,...$$

این جواب بازاء مقادیر خاص k، مقدار ویژه سیستم است.

$$y = a\ sin(n\pi x).$$

شکل ۱.۴ چند جواب این معادله را نشان می‌دهد.



شکل ۱.۴

اگر با مسأله ارتعاش مواجه هستیم یک تابع ویژه $y(x)$ متناظر با مقدار ویژه وجود دارد و شکل منحنی الاستیک را وقتی سیستم متعادل است نشان می‌دهد. می‌توانیم معادله فوق را به طریق عددی حل کنیم.

معادله دیفرانسیل را با تفاضل‌های محدود جایگزین می‌کنیم. با استفاده از تقریب تفاضل مرکزی داریم:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + k^2 y_i = 0.$$

با انتخاب $h=0.2$ برای چهار نقطه درونی فاصله $0\ ,\ 1$ داریم.

$$\begin{aligned}
y_1 - (2 - 0.04k^2)y_2 + y_3 &= 0,\\
y_2 - (2 - 0.04k^2)y_3 + y_4 &= 0,\\
y_3 - (2 - 0.04k^2)y_4 + y_5 &= 0,\\
y_4 - (2 - 0.04k^2)y_5 + y_6 &= 0.
\end{aligned}$$

با شرایط مرزی $y_1 = y_6 = 0$، اگر این معادلات را به صورت ماتریس بنویسیم، خواهیم داشت:

$$\begin{bmatrix}
2 - 0.04k^2 & -1 & 0 & 0\\
-1 & 2 - 0.04k^2 & -1 & 0\\
0 & -1 & 2 - 0.04k^2 & -1\\
0 & 0 & -1 & 2 - 0.04k^2
\end{bmatrix}
\begin{bmatrix} y_2\\ y_3\\ y_4\\ y_5 \end{bmatrix} = 0. \qquad (7)$$

توجه شود که می‌توانیم این معادله را بصورت $(A-\lambda I)y=0$ بنویسیم، بطوری که:

$$A=\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}, \qquad \lambda = 0.04k^2.$$

چون $det(A-\lambda I)=0$ (دترمینان باید برابر صفر باشد.)

$$\mathrm{Det}\begin{bmatrix} 2-0.04k^2 & -1 & 0 & 0 \\ -1 & 2-0.04k^2 & -1 & 0 \\ 0 & -1 & 2-0.04k^2 & -1 \\ 0 & 0 & -1 & 2-0.04k^2 \end{bmatrix} = 0.$$

با بسط این دترمینان یک چند جمله‌ای درجه هشت بر حسب k که معادله مشخصه ماتریس نامیده می‌شود بدست می‌آید. فرض می‌کنیم: $z = 2 - 0.04k^2$

$$z^4 - 3z^2 + 1 = 0$$

$$z = 1.618 \quad , -1.618, \ 0.618, \ -0.618$$

و سپس مقادیر ویژه مشخص می‌گردند

$$k = \pm\ 3.09$$
$$k = \pm\ 5.88$$
$$k = \pm\ 8.09$$
$$k = \pm\ 9.51$$

این روش محاسبات، بسیار وقت گیر می‌باشد، خوشبختانه در اکثر مسایل کاربردی کوچکترین مقدار k مورد توجه می‌باشد؛ همچنین برای دقت بیشتر مقدار k، باید مقدار h را کوچک در نظر بگیریم. در اینصورت حجم عملیات بصورت غیر عادی زیاد می‌شود.

ما می‌خواهیم روش‌های ساده‌تری برای بدست آوردن مقادیر ویژه پیدا کنیم. که موضوع اساسی این فصل می‌باشد. همچنین در بسط دترمینان و بدست آوردن ریشه‌های یک معادله مشخصه با درجه بزرگ ممکن است خطاهای بزرگی بوجود آید و عملیات «ناجور» گردد. روش‌های عددی که در این فصل بیان می‌شود بمنظور بدست آوردن مقادیر ویژه و بردارهای ویژه با دقت کافی و کاهش زمان عملیات می‌باشند.

یادآوری:

۱.۴ خواص مقادیر ویژه و بردارهای ویژه یک ماتریس

a – بردارهای ویژه یکتا نیستند و در یک ضریب ثابت دلخواه تفاوت دارند.

b – در حالت کلی مقادیر ویژه و بردارهای ویژه یک ماتریس، حقیقی یا مختلط می‌باشند.

c – اگر A متقارن باشد، مقادیر ویژه و بردارهای ویژه آن حقیقی هستند.

d – اگر A متقارن باشد، بردارهای ویژه A متعامدند، یعنی $\lambda_r \neq \lambda_s$, $x_r^T . x_s = 0$

e- اگر کلیه مقادیر ویژه A متفاوت باشند، در اینصورت بردارهای ویژه A بطور خطی مستقل می‌باشند و ما می‌توانیم هر بردار دلخواه را بعنوان ترکیب خطی از این بردارهای ویژه بنویسیم.

بعنوان مثال اگر $A\underline{x} = \lambda \underline{x}$, $A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ خواهیم داشت:

$$\lambda_1 = 0 \, , \lambda_2 = 5 \quad , \quad x_1 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \qquad x_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

در اینصورت بردار دلخواه $z = \begin{bmatrix} 79 \\ 31 \end{bmatrix}$ را می‌توان بصورت ترکیب خطی از x_1, x_2 نوشت:

$$z = \alpha_1 x_1 + \alpha_2 x_2$$

$$\therefore \quad \alpha_1 \begin{bmatrix} -2 \\ 1 \end{bmatrix} + \alpha_2 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 79 \\ 31 \end{bmatrix}$$

$$\begin{bmatrix} -2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 79 \\ 31 \end{bmatrix}$$

$$\det \begin{bmatrix} -2 & 1 \\ 1 & 2 \end{bmatrix} = -5$$

چون دترمینان ضرایب مخالف صفر است، در نتیجه α_1 , α_2 (جواب دستگاه فوق) معین خواهند شد:

$$\alpha_1 = -\frac{127}{5}$$

$$\alpha_2 = \frac{141}{5}$$

f- اگر S ماتریسی غیر منفرد باشد در اینصورت:

مقادیر ویژه $\{\lambda_i\}$ و بردارهای ویژه $\{x_i\}$ از ماتریس A و مقادیر ویژه $\{\mu_i\}$ و بردارهای ویژه $\{y_i\}$ از ماتریس SAS^{-1} با هم مرتبط هستند، بطوری که:

$$\mu_i = \lambda_i \quad , \quad y_i = sx_i$$

اثبات:

$$Ax = \lambda x \quad , \quad SAx = \lambda Sx$$

$$SA(S^{-1}S)x = \lambda Sx \quad , \quad (SAS^{-1})\, Sx = \lambda Sx$$

که نتیجه مطلوب بدست آمده است. $(SAS^{-1})\, y = \lambda y$, $y = Sx$ و λ برابر μ مقدار ویژه ماتریس SAS^{-1} می‌باشد.

۲.۴ روش لوریه - فادیو[1]

می‌دانیم که مجموع عناصر قطر یک ماتریس مربع «اثر»[2] ماتریس خوانده می‌شود و برای ماتریس A داریم:

1- Leverrier-Faddeev 2- Trace

$tr\ (A) = a_{11} + a_{22} + a_{33} + \ldots + a_{nn}$

و ضرایب معادله درجه n امی‌که ریشه‌های آن مقادیر ویژه ماتریس A باشند، $C_1, C_2, \ldots, C_n$ از روبط زیر بدست می‌آیند.

$$A_1 = A \qquad C_1 = tr\ (A_1)$$

$$A_2 = A(A_1 - C_1 I) \qquad C_2 = (\frac{1}{2})tr\ (A_2)$$

$$A_3 = A(A_2 - C_2 I) \qquad C_3 = (\frac{1}{3})tr\ (A_3)$$

$$\vdots \qquad\qquad \vdots$$

$$A_n = A(A_{n-1} - C_{n-1} I) \qquad C_n = (\frac{1}{n})tr\ (A_n)$$

$$\lambda^n - C_1 \lambda^{n-1} - C_2 \lambda^{n-2} - \ldots - C_{n-1} \lambda - C_n = 0$$

$A_n - C_n I = 0$ که A_n ماتریسی قطری است و داریم:

از این‌رو معکوس ماتریس A را نیز می‌توان نتیجه گرفت:

$$A^{-1} = (\frac{1}{C_n})(A_{n-1} - C_{n-1} I)$$

مثال ۴.۴ مقادیر ویژه ماتریس زیر را بدست آورید.

$$A = \begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix}$$

$$A = A_1$$

$$C_1 = \mathrm{tr}\, A = 6$$

$$A_2 = A(A_1 - C_1 I) = \begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix} \begin{bmatrix} -4 & 3 & 2 \\ 10 & -3 & 4 \\ 3 & 6 & -5 \end{bmatrix} = \begin{bmatrix} 28 & 9 & 6 \\ 2 & 45 & 12 \\ 51 & -3 & 25 \end{bmatrix}$$

$$C_2 = \frac{1}{2}\mathrm{tr}\, A_2 = 49$$

$$A_3 = A(A_2 - C_2 I) = \begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix} \begin{bmatrix} -21 & 9 & 6 \\ 2 & -4 & 12 \\ 51 & -3 & -24 \end{bmatrix} = \begin{bmatrix} 66 & 0 & 0 \\ 0 & 66 & 0 \\ 0 & 0 & 66 \end{bmatrix}$$

$$C_3 = \frac{1}{3}\mathrm{tr}\, A_3 = 66$$

معادله مشخصه ماتریس برابر است با: $\lambda^3 - 6\lambda^2 - 49\lambda - 66 = 0$

مقادیر ویژه یا ریشه‌های معادله فوق برابرند با: $\lambda_1 = -2,\ \lambda_2 = -3,\ \lambda_3 = 11$

و معکوس ماتریس برابراست با:

$$A^{-1}=\frac{1}{C_3}(A_2-C_2I)=\frac{1}{66}\begin{bmatrix}-21 & 9 & 6\\ 2 & -4 & 12\\ 51 & -3 & -24\end{bmatrix}$$

۳.۴ روش توانی[1]

از معادله $Ax = \lambda x$ مقادیر ویژه $\lambda_1, \lambda_2, \ldots, \lambda_n$ و بردارهای ویژه $x_1,x_2,\ldots,x_n$ مربوط به ماتریس A تعیین می‌گردد. بردار دلخواه $z^{(i)} \neq 0$ را اختیار کرده و عمل تکراری زیر را انجام می‌دهیم.

$$y = Az^{(i)}$$

d_i بزرگترین عنصر از نظر قدر مطلق را از بردار y انتخاب کرده و بردار جدید $z^{(i+1)}$ را تشکیل می‌دهیم:

$$z^{(i+1)} = \frac{y}{d_i},\ i = 0, 1, 2, \ldots n$$

دراینصورت اگر داشته باشیم: $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$

(λ_1 حقیقی فرض می‌شود)

خواهیم داشت: $d_i \to |\lambda_1|,\ z^{(i+1)} \to x_1$ وقتی $i \to \infty$

مثال ۵.۴ بزرگترین مقدار ویژه از نظر قدر مطلق را برای ماتریس A بدست آورید.

$$A=\begin{bmatrix}2 & 3 & 2\\ 10 & 3 & 4\\ 3 & 6 & 1\end{bmatrix}$$

بردار اولیه $Z^{(0)}=\begin{bmatrix}1\\1\\1\end{bmatrix}$ را انتخاب می‌کنیم، پس:

$$y^{(0)}=AZ^{(0)}=\begin{bmatrix}7\\17\\10\end{bmatrix},\quad d_0=17$$

$$Z^{(0)}=\frac{y^{(0)}}{d_0}=\begin{bmatrix}.4118\\1\\.5882\end{bmatrix}$$

$$y^{(1)}=AZ^{(1)}=\begin{bmatrix}5.000\\9.4708\\7.8236\end{bmatrix},\quad d_1=9.4708$$

$$Z^{(2)}=\frac{y^{(1)}}{d_1}=\begin{bmatrix}0.5279\\1\\0.8260\end{bmatrix}$$

1- Power Method

$$y^{(2)} = AZ^{(2)} = \begin{bmatrix} 5.708 \\ 11.583 \\ 7.062 \end{bmatrix} \quad , \quad d_2 = 11.583$$

$$Z^{(3)} = \frac{y^{(2)}}{d_2} \quad , \quad = \begin{bmatrix} .4927 \\ 1 \\ .6096 \end{bmatrix}$$

$$y^{(3)} = AZ^{(3)} = \begin{bmatrix} 5.205 \\ 10.365 \\ 8.088 \end{bmatrix} \quad , \quad d_3 = 10.365$$

$$Z^{(4)} = \frac{y^{(3)}}{d_3} \quad , \quad Z^{(4)} = \begin{bmatrix} .5021 \\ 1 \\ .7803 \end{bmatrix}$$

$$y^{(4)} = AZ^{(4)} = \begin{bmatrix} 5.565 \\ 11.142 \\ 8.287 \end{bmatrix} \quad , \quad d_4 = 11.142$$

وبا چند تکرار دیگر نتیجه می‌شود که:

$$\lambda_1 = 11.0000 \quad , \quad x_1 = \begin{bmatrix} .5000 \\ 1.0000 \\ 0.7500 \end{bmatrix}$$

اثبات همگرائی

فرض کنیم λn , ... , $\lambda 2$, $\lambda 1$ مجزا از یکدیگر بوده بطوریکه x_n , ... , x_2, x_1 بطور خطی مستقل باشند. در اینصورت می‌توان نوشت:

$$z^{(0)} = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n$$

و α_n , ... , α_2, α_1 مقادیر ثابتی هستند.

اکنون این روش را بکار می‌بریم.

$$y = Az^{(0)} = \alpha_1 A x_1 + \alpha_2 A x_2 + \ldots + \alpha_n A x_n \qquad (1)$$

اما می‌دانیم که:

$$Ax_1 = \lambda_1 x_1 \, , \, Ax_2 = \lambda_2 x_2 \, , \, \ldots \, , \, Ax_n = \lambda_n x_n \qquad (2)$$

با جایگزینی این عبارات در رابطه (1) خواهیم داشت:

$$y = \alpha_1 \lambda_1 x_1 + \alpha_2 \lambda_2 x_2 + \ldots + \alpha_n \lambda_n x_n \qquad (3)$$

بنابراین y را بر بزرگترین عنصر آن از نظر قدر مطلق تقسیم می‌کنیم

$$z^{(1)} = \frac{y}{d_0} \qquad (4)$$

$$z^{(1)} = \frac{1}{d_0}(\alpha_1 \lambda_1 x_1 + \alpha_2 \lambda_2 x_2 + \ldots + \alpha_n \lambda_n x_n) \qquad (5)$$

و بطور مشابه داریم:

$$z^{(2)} = \frac{1}{d_1 d_0}(\alpha_1 \lambda_1^2 x_1 + \alpha_2 \lambda_2^2 x_2 + \ldots + \alpha_n \lambda_n^2 x_n) \qquad (6)$$

زیرا که: $y = Az^{(1)}$، $z^{(2)} = \frac{y}{d_1}$ و با جایگزینی روابط (2) در آن، عبارت (6) بدست می‌آید.

اگر عملیات را بهمین طریق ادامه دهیم، در حالت کلی داریم:

$$z^{(i)} = \frac{1}{d_{i-1} d_{i-2} \ldots d_2 d_1 d_0}(\alpha_1 \lambda_1^i x_1 + \alpha_2 \lambda_2^i x_2 + \cdots + \alpha_n \lambda_n^i x_n) \qquad (7)$$

$$z^{(i)} = \frac{\lambda_1^i}{d_{i-1} d_{i-2} \ldots d_2 d_1 d_0}(\alpha_1 x_1 + \alpha_2 (\frac{\lambda_2}{\lambda_1})^i x_2 + \cdots + \alpha_n (\frac{\lambda_n}{\lambda_1})^i x_n) \qquad (8)$$

با فرض اینکه داشته باشیم: $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$

خواهیم داشت:

$$|\frac{\lambda_2}{\lambda_1}| < 1 , |\frac{\lambda_3}{\lambda_1}| < 1 , \ldots , |\frac{\lambda_n}{\lambda_1}| < 1 \qquad (9)$$

بنابراین برای مقادیر بزرگ i، نتیجه می‌شود.

$$(\frac{\lambda_2}{\lambda_1})^i \to 0 \quad , \quad \ldots \quad , \quad (\frac{\lambda_n}{\lambda_1})^i \to 0 \qquad (10)$$

$$z^{(i)} = \frac{\lambda_1^{(i)}}{d_{i-1} d_{i-2} \ldots d_1 d_0} \alpha_1 x_1 \qquad (11)$$

$z^{(i)} \to x_1$

از اینرو نتیجه می‌شود:

(زیرا که مضربی از یک بردار ویژه، خود برداری ویژه است)

در این حال تکرار بعدی را آزمایش می‌کنیم.

$$y = Az^{(i)} = \frac{1}{d_{i-1} d_{i-2} \ldots d_1 d_0} \lambda_1^i \alpha_1 Ax_1 \qquad (12)$$

$$y = \frac{1}{d_{i-1} d_{i-2} \ldots d_1 d_0} \lambda_1^i \alpha_1 \lambda_1 x_1 \qquad (13)$$

پس:

$$y \to \lambda_1 z^{(i)} \qquad \text{یا} \qquad y = Az^{(i)} = \lambda_1 z^{(i)} \qquad (14)$$

چون بزرگترین عنصر $z^{(i)}$ از نظر قدر مطلق یک می‌باشد در اینصورت بزرگترین عنصر از نظر قدرمطلق برای y ، λ_1 خواهد بود، یعنی: $d_i \to |\lambda_1|$

مشاهده می‌شود که سرعت همگرائی عملیات به $|\frac{\lambda_2}{\lambda_1}|$ بستگی دارد. اگر $|\frac{\lambda_2}{\lambda_1}| \approx 1$ در اینصورت سرعت همگرائی بسیار کم است و اگر $|\frac{\lambda_2}{\lambda_1}| \approx 0$ در اینصورت سرعت همگرائی بسیار زیاد است.

تاکنون λ_1 حقیقی فرض شده بود. این سئوال مطرح است که اگر λ_1 حقیقی نباشد نتیجه چگونه محاسبه می‌شود.

$$|\lambda_1| = |\lambda_2| > |\lambda_3| \geq |\lambda_4| \geq \ldots \geq |\lambda_n|$$

λ_2 ، λ_1 مختلط و مزدوج یکدیگرند.

مطابق علمیاتی که توضیح داده شد، بجای رابطه (11) خواهیم داشت:

$$z^{(i)} = \frac{1}{d_{i-1}\, d_{i-2} \ldots d_1 d_0} \left(\alpha_1 \lambda_1^{\,i} x_1 + \alpha_2 \lambda_2^{i} x_2 \right) \qquad (15)$$

بهر حال هنوز امکان دارد که از دنباله $\{z^{(i)}\}$ مقادیر λ_1 ، λ_2 ، x_1 ، x_2 را مطابق زیر بدست آوریم. فرض می‌کنیم λ_1 ، λ_2 دو جواب معادله $\lambda^2 + a\lambda + b = 0$ باشند.

اگر بتوانیم a ، b را پیدا نمائیم، در اینصورت λ_1 ، λ_2 بدست می‌آیند برای اینکار نشان می‌دهیم که:

$$\underset{i \to \infty}{d_{i+1}\, d_i}\, z^{(i+2)} + a\, d_i\, z^{(i+1)} + b\, z^{(i)} = 0 \qquad (16)$$

به این منظور رابطه (15) را بکار برده و در رابطه (16) جایگزین می‌نمائیم.

$$d_{i+1}\, d_i \left(\frac{\alpha_1 \lambda_1^{i+2} x_1 + \alpha_2 \lambda_2^{i+2} x_2}{d_{i+1} d_i \ldots d_1 d_0} \right) + a\; d_i \left(\frac{\alpha_1 \lambda_1^{i+1} x_1 + \alpha_2 \lambda_2^{i+1} x_2}{d_i\, d_{i+1} \ldots d_1 d_0} \right) \qquad (17)$$

$$+ b \left(\frac{\alpha_1 \lambda_1^{i} x_1 + \alpha_2 \lambda_2^{i} x_2}{d_{i-1}\, d_{i-2} \ldots d_1 d_0} \right) = 0$$

بعد از ساده کردن رابطه (17) خواهیم داشت:

$$\alpha_1 \lambda_1^{i} \left(\lambda_1^2 + a\lambda_1 + b \right) x_1 + \alpha_1 \lambda_2^{i} \left(\lambda_2^2 + a\lambda_2 + b \right) x_2 = 0 \qquad (18)$$

و طبق فرض $\lambda_2^{\,2} + a\lambda_2 + b = 0$ ، $\lambda_1^{\,2} + a\lambda_1 + b = 0$ پس رابطه (16) برقراراست یعنی:

$$d_{i+1}\, d_i\, z^{(i+2)} + a\, d_i\, z^{(i+1)} + b\, z^{(i)} = 0$$

این رابطه نمایش n معادله خطی با دو مجهول b,a می‌باشد. معمولاً برای حل آن، دو معادله را انتخاب کرده و b,a را از حل این دو معادله و دو مجهول بدست می‌آوریم.

پس از بدست آمدن b,a از معادله درجه دوم $\lambda^2 + a\lambda + b = 0$ می‌توان λ_1 و λ_2 را تعیین نمود.

در اینصورت رابطه (15) را مجدداً بکار می‌بریم.

$$z^{(i)} = \frac{1}{d_{i-1}\, d_{i-2} \cdots d_1 d_0} \left(\alpha_1 \lambda_1{}^i x_1 + \alpha_2 \lambda_2^i x_2 \right)$$

و می‌دانیم که: $y = Az^{(i)}$, $\frac{y}{d_i} = z^{(i+1)}$

$Az^{(i)} = d_i z^{(i+1)}$

پس: $$d_i z^{(i+1)} = \frac{1}{d_{i+1} \cdots d_1 d_0} \left(\alpha_1 \lambda_1^{i+1} x_1 + \alpha_2 \lambda_2^{i+1} x_2\right)$$

از اینرو داریم: $$d_i z^{(i+1)} - \lambda_1 z^{(i)} \to \left(\frac{\alpha_2 \lambda_2^{i+1} - \alpha_2 \lambda_1 \lambda_2^i}{d_{i-1} \cdots d_1 d_0}\right) x_2$$

یا: $d_i\ z^{(i+1)} - \lambda_1\, z^{(i)} \to x_1$

و $d_i\ z^{(i+1)} - \lambda_2\, z^{(i)} \to x_2$

۴.۴ روش تکراری معکوس

در این روش سعی می‌کنیم یکی از مقادیر ویژه را که به عددی مانند ρ نزدیکتر است، بدست آوریم.

بردار دلخواه $z^{(0)} \neq 0$ را انتخاب کرده و روش تکراری را برای معادلهٔ

$$(A - \rho I)y = z^{(i)} \qquad (1)$$

بکار می‌بریم. بعد از بدست آوردن y از این معادله، $z^{(i+1)}$ را نتیجه می‌گیریم:

$$z^{(i+1)} = \frac{y}{\mu_i} \qquad (2)$$

μ_i بزرگترین عنصر y از نظر قدر مطلق است و مجدداً با استفاده از $z^{(i+1)}$ و رابطه (۱)، y جدید را بدست آورده و عملیات را ادامه می‌دهیم.

ρ مقداری ثابت است و برای کلیه i ها $(i \neq k)$: $|\lambda_k - \rho| \leq |\lambda_i - \rho|$

$|\lambda_k - \rho|$ کوچکترین مقدار ویژه یا $\frac{1}{|\lambda_k - \rho|}$ بزرگترین مقدار ویژه برای $(A - \rho I)^{-1}$ می‌باشد. در اینصورت λ_k نزدیکترین مقادیر ویژه به ρ می‌باشد و داریم:

$$\mu_i \underset{i \to \infty}{\to} \frac{1}{\lambda_i - \rho} \quad , \quad z^{(i)} \underset{i \to \infty}{\to} x_i$$

مقدار ویژه $\lambda_i = \rho + \frac{1}{\mu_i}$

بردار ویژه x_i

اثبات همگرائی:

این روش همانند روش توانی است اما ماتریس A توسط $(A - \rho I)^{-1}$ جایگزین شده است.

می‌دانیم که در روش توانی d_i به سمت بزرگترین مقدار ویژه ماتریس A همگراست، اما در روش تکراری معکوس d_i به سمت بزرگترین مقدار ویژه $(A - \rho I)^{-1}$ همگراست. (از نظر قدر مطلق) زیرا:

$$Ax = \lambda x \quad ,$$

$$Ax - \rho x = \lambda x - \rho x$$

$$(A - \rho I)\, x = (\lambda - \rho)\, x$$

یا:
$$x = (\lambda - \rho)(A - \rho I)^{-1} x$$

یا:
$$(A - \rho I)^{-1} x = \frac{1}{(\lambda - \rho)} x$$

یعنی $(\lambda - \rho)^{-1}$ مقدار ویژه ماتریس $(A - \rho I)^{-1}$ می‌باشد

توجه شود که دستگاه $(A - \rho I)\, y = z^{(i)}$ می‌بایست با روش تجزیه LU یا روش چولسکی حل شود. (اگر $A - \rho I$ متقارن و معین مثبت باشد.)

ماتریس $A - \rho I$ می‌باید به LU یا LL^T تجزیه شود و سپس روش تکراری آغاز شده و برای هر $z^{(i)}$ دستگاه $(A - \rho I)\, y = z^{(i)}$ حل گردد. یعنی ابتدا x محاسبه شده و سپس y بدست می‌آید.

$$\begin{cases} Lx = Z^{(i)} \\ Uy = x \end{cases}$$

مثال ۶.۴ مطلوبست محاسبه یکی از مقادیر ویژه ماتریس A که به صفر نزدیک‌تر است.

$$A = \begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix}$$

با انتخاب $\rho = 0$, $z^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ داریم:

$$\begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix} \quad y = z^{(0)}$$

که یک دستگاه معادله خطی می‌باشد. بعد از محاسبه y خواهیم داشت:

$$y = \begin{bmatrix} -0.0909 \\ 0.015 \\ 0.3636 \end{bmatrix}$$

$$z^{(1)} = \frac{y}{d_0} = \begin{bmatrix} -.2500 \\ .4167 \\ 1.0000 \end{bmatrix} \qquad d_0 = .3636$$

و با ادامه عملیات نتیجه خواهد شد:

$$z^{(r)} \to \begin{bmatrix} -.۲۰۰۱ \\ -.۳۴۴۱ \\ ۱.۰۰۰۰ \end{bmatrix}$$

$$d_r \to -.5001 = \frac{1}{\lambda_r} \quad , \quad \lambda_r = -2.0000$$

۱.۴ قضیه گرچ گورین[۱]

گاهی اوقات از یک مسأله فیزیکی می‌توان یک مقدار ویژه آنرا پیش بینی کرد. روش توانی اکثراً با چند تکرار یک مقدار تقریبی را می‌دهد. قضیه گرچ گورین تخمین خوبی برای حدود تغییرات مقادیر ویژه می‌دهد. فرض می‌کنیم D_i دایره‌ای باشد که مرکز آن a_{ii} و شعاع آن

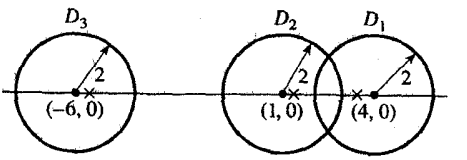
$$\Sigma |a_{ij}| \quad , j = 1, 2, \ldots, n \quad , j \neq i$$

۱ – طبق قضیه گرچ گورین هر مقدار ویژه A باید در میدان D باشد بطوری که

$$D = \bigcup_{i=1}^{n} D_i$$

۲ – طبق قضیه گرچ گورین اگر تعداد k دایره از مجموعه، $n-k$ دایره دیگر از این مجموعه را قطع نکنند، دقیقاً k مقدار ویژه در اجتماع این k دایره واقع هستند. برای ماتریس A به شکل زیر توجه کنید:

$$A = \begin{bmatrix} 4 & -1 & 1 \\ 1 & 1 & 1 \\ -2 & 0 & -6 \end{bmatrix}.$$



یک مقدار ویژه -5.76851 در D_3 صدق می‌کند و دو مقدار دیگر در $D_1 \cup D_2$ صادق هستند.

۱- *Gersch - Gorin*

۵.۴ روش صفر کردن مقادیر ویژه ماتریس

معادله ویژه $Ax = \lambda x$ را بررسی می‌کنیم که $\lambda_1, \lambda_2, \ldots, \lambda_n$ مقادیر ویژه و $x_1, x_2, \ldots, x_n$ بردارهای ویژه می‌باشند.

فرض می‌کنیم مقادیر ویژه حقیقی بوده و داشته باشیم $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$ اگر روش توانی رابرای ماتریس A بکار برده و λ_1 بزرگترین قدر مطلق مقادیر ویژه و همچنین بردار ویژه متناظر آن، x_1 را بدست آوریم، این سئوال مطرح می‌شود که چگونه سایر ریشه‌های این معادله را محاسبه نمائیم.

می‌توانیم ایدهٔ صفر کردن مقدار ویژه را بعنوان پاسخی برای این سئوال بکار بریم. یعنی ماتریس دیگری بنام A_1 بوجود آوریم که دارای کلیه مقادیر ویژه ماتریس A باشد بجز آنکه مقدار ویژه آن λ_1 نبوده و بجای آن صفر در مجموعه مقادیر ویژه جایگزین شده است.

حالت اول – A متقارن است.

فرض کنیم x_1 را نرمال کرده باشیم بطوریکه $x_1^T \cdot x_1 = 1$.

حال ماتریس زیر را بررسی می‌کنیم.

$$A_1 = A - \lambda_1 x_1 x_1^T$$

$$A_1 x_1 = (A - \lambda_1 x_1 x_1^T) x_1 = Ax_1 - \lambda_1 x_1 (x_1^T x_1)$$

چون

$$A_1 x_1 = Ax_1 - \lambda_1 x_1 = \lambda_1 x_1 - \lambda_1 x_1 = 0 . x_1$$

$$A_1 x_1 = 0 . x_1$$

بنابراین صفر مقدار ویژه و x_1 بردار ویژه A_1 است.

همچنین برای $r \neq 1$ داریم:

$$A_1 x_r = (A - \lambda_1 x_1 x_1^T) x_r$$
$$= Ax_r - \lambda_1 x_1 (x_1^T x_r)$$
$$= A x_r = \lambda_r x_r$$

بنابراین λ_r مقدار ویژه و x_r بردار ویژه A_1 می‌باشد بطوریکه اکنون می‌توانیم روش توانی را در مورد ماتریس A_1 بکار بریم و بزرگترین مقدار ویژه از نظر قدر مطلق λ_2 و بردار ویژهٔ نظیر آن x_2 را بدست آوریم. البته توجه کردیم که مقادیر ویژهٔ A_1 برابر با $0, \lambda_2, \ldots, \lambda_n$ بوده و بردارهای ویژه آن برابر $x_1, x_2, \ldots, x_n$ می‌باشد.

مجدداً می‌توانیم x_2 را بصورت نرمال درآوریم بطوریکه $x_2^T \cdot x_2 = 1$ و ماتریس A_2 را تشکیل دهیم:

$$A_2 = A_1 - \lambda_2 x_2 x_2^T$$

ماتریس A_2 دارای مقادیر ویژه $0, 0, \lambda_3, \ldots, \lambda_n$ و بردارهای ویژه $x_1, x_2, \ldots, x_n$ می‌باشد. در

اینصورت می‌توانیم روش توانی را بکار برده و λ_3 و x_3 را بدست آوریم و بهمین ترتیب سایر مقادیر ویژه و بردارهای ویژه محاسبه می‌شوند.

مثال ۷.۴ مقادیر ویژه ماتریس A و همچنین بردارهای ویژه آنرا از روش توانی بدست آورید.

$$A=\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}$$

با انتخاب $Z^{(0)}=\begin{bmatrix}1\\0\end{bmatrix}$ و $y = A\ z^{(0)}$ و محاسبه متوالی $z^{(r)}$ خواهیم داشت:

$$z^{(0)}=\begin{bmatrix}1\\0\end{bmatrix}\ ,\quad z^{(1)}=\begin{bmatrix}.5\\1.5\end{bmatrix}\ ,\quad z^{(2)}=\begin{bmatrix}1.0\\.8\end{bmatrix}$$

$$z^{(3)}=\begin{bmatrix}.928\\1.000\end{bmatrix}\ ,\quad z^{(4)}=\begin{bmatrix}1.000\\.975\end{bmatrix}\ ,\quad z^{(5)}=\begin{bmatrix}1\\1\end{bmatrix}$$

$$d_1 = 1\ ,\ d_2 = 2.5\ ,\ d_3 = 2.8\ ,\ d_4 = 2.925\ ,\ d_5 = 3.000$$

در نتیجه خواهیم داشت:

$$\lambda_1=3.00\ ,\quad x_1=\begin{bmatrix}1\\1\end{bmatrix}$$

برای نرمال کردن x_1 فرض می‌کنیم:

$$x_1=\alpha\begin{bmatrix}1.000\\1.000\end{bmatrix}$$

بطوریکه:

$$x_1^{T}\,x_1=1=\alpha\begin{bmatrix}1\\1\end{bmatrix}^{T}\alpha\begin{bmatrix}1\\1\end{bmatrix}=2\alpha^2=1$$

$$\alpha^2=\frac{1}{2}\quad,\ \alpha=\frac{\sqrt{2}}{2}\qquad x_1=\begin{bmatrix}\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}$$

$$A_1=A-\lambda_1\,x_1\,x_1^{T}=\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}-3\begin{bmatrix}\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}\begin{bmatrix}\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2}\end{bmatrix}$$

$$A_1=\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}-3\begin{bmatrix}\frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & \frac{1}{2}\end{bmatrix}=\begin{bmatrix}-\frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & -\frac{1}{2}\end{bmatrix}$$

از روش توانی استفاده می‌کنیم:

$$Z^{(r)} \rightarrow \begin{bmatrix} .5 \\ .0 \end{bmatrix}, \begin{bmatrix} 1.000 \\ -1.000 \end{bmatrix}, \begin{bmatrix} 1.000 \\ -1.000 \end{bmatrix}, \cdots$$

$$\lambda_2 = -1.000 \qquad x_2 = \begin{bmatrix} 1.000 \\ -1.000 \end{bmatrix}$$

حالت دوم – A نامتقارن است.

x_1 را طوری تغییر می‌دهیم که اولین عنصر آن 1 باشد. همچنین سایر بردارهای ویژه A را با ضرب درمقدار ثابت α طوری انتخاب می‌کنیم که اولین عنصرشان 1 باشد.

فرض می‌کنیم $A = \begin{bmatrix} a_1^T \\ B \end{bmatrix}$ بطوریکه a_1^T اولین سطر A و B سایر سطرهای A باشد.

حال ماتریس A_1 را تشکیل می‌دهیم:

$$A_1 = A - x_1 a_1^T$$

$$A_1 x_1 = (A - x_1 a_1^T) x_1 = Ax_1 - x_1(a_1^T x_1)$$

$$= A x_1 - \lambda_1 x_1$$

$$= \lambda_1 x_1 - \lambda_1 x_1 = 0.x_1$$

در روابط فوق از $a_1^T x_1 = \lambda_1$ استفاده نمودیم که با توجه به $Ax_1 = \lambda_1 x_1$ یا

$$Ax_1 = \begin{bmatrix} a_1^T \\ B \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \lambda_1 \begin{bmatrix} 1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

این نتیجه حاصل گردیده است.

همینطور از رابطه $Ax_r = \lambda_r x_r$

خواهیم داشت: $a_1^T x_r = \lambda_r$

مشاهده می‌شود که مقادیر ویژه A_1 برابرند با $\lambda_n, \ldots, \lambda_3, \lambda_2, 0$ و بردارهای ویژه آن، x_1 ، $x_1 - x_2, \ldots, x_1 - x_n$ زیراکه:

$$A_1 (x_1 - x_r) = (A - x_1 a_1^T)(x_1 - x_r)$$

$$= Ax_1 - x_1 (a_1^T x_1) - A x_r + x_1(a_1^T x_r)$$

$$= Ax_1 - \lambda_1 x_1 - A x_r + \lambda_r x_1$$

$$= \lambda_1 x_1 - \lambda_1 x_1 - \lambda_r x_r + \lambda_r x_1$$

$$= \lambda_r(x_1 - x_r) \qquad r \neq 1$$

در اینصورت اگر روش توانی را برای A_1 بکار بریم مقدار ویژه λ_2 و بردار ویژه‌ای که مضربی از $x_2 - x_1$ است، بدست می‌آید. یعنی:

$$x_1 - x_2 = C z_2$$

$$a_1^T (x_1 - x_2) = C a_1^T z_2$$

بنابراین:

$$a_1^T x_1 - a_1^T x_2 = C a_1^T z_2$$

$$\lambda_1 - \lambda_2 = C a_1^T z_2 \ ,$$

$$C = \frac{\lambda_1 - \lambda_2}{a_1^T z_2}$$

$$x_2 = x_1 - \frac{\lambda_1 - \lambda_2}{a_1^T z_2} z_2$$

بهمین ترتیب برای تعیین سایر مقادیر ویژه و بردارهای ویژه عمل می‌کنیم. اما لازم است توجه کنیم که این روش تنهابرای تعیین چند مقدار ویژه و بردار ویژه اولیه پایدار است.

۶.۴ روش تبدیل ماتریس برای تعیین مقادیر ویژه[1]

اگر بخواهیم کلیه مقادیر ویژه λ_n , ... , λ_1, λ_2 از ماتریس A را محاسبه کنیم $(Ax = \lambda x)$، روش‌های تکراری مذکور مناسب نیستند. در اینصورت روش‌های دیگری را بکار می‌بریم.

یادآوری می‌کنیم که اگر S، ماتریسی غیر منفرد باشد،

$$B = S^{-1} A S \qquad \text{(تبدیل متشابه)}$$

B دارای همان مقادیر ویژه A بوده، و بردارهای ویژه آن از حاصلضرب S^{-1} در بردارهای ویژه A بدست می‌آیند.

در اینجا این سوال پیش می‌آید که آیا می‌توانیم ماتریس S را بگونه ای پیدا کنیم که بعد از محاسبه B، معادله $By = \lambda y$ آسان حل شود.

$$Ax = \lambda x \qquad \text{(مشکل)} \quad , \quad By = \lambda y \qquad \text{(آسان)}$$

اگر ماتریس B قطری باشد، مقادیر ویژه آن همان عناصر قطرش می‌باشد، اما بهرحال بدست آوردن S نسبتاً مشکل است. دراینجا سعی می‌کنیم S را در چند مرحله بدست آوریم:

$$A_1 = S_1 A S_1^{-1}$$

$$A_2 = S_2 A_1 S_2^{-1}$$

$$A_3 = S_3 A_2 S_3^{-1}$$

$$\vdots$$

$$A_{n-1} = S_{n-1} A_{n-2} S_{n-1}^{-1}$$

1- **Transformation Method for Eigen values**

$$A_n = S_n A_{n-1} S_n^{-1}$$
$$B = A_n = S_n (S_{n-1} A_{n-2} S_{n-1}^{-1}) S_n^{-1} = \ldots$$
$$= (S_n S_{n-1} \ldots S_2 S_1) A (S_1^{-1} S_2^{-1} \ldots S_n^{-1})$$
$$B = (S_n S_{n-1} \ldots S_2 S_1) A (S_n S_{n-1} \ldots S_1)^{-1}$$
$$S = S_n S_{n-1} \ldots S_2 S_1$$
$$B = S A S^{-1} \quad ,$$

واضح است که $S_n, \ldots, S_2, S_1$ باید طوری انتخاب شوند که B ماتریس قطری شود و محاسبه معکوس آنها $S^{-1}_n, \ldots, S^{-1}_2, S^{-1}_1$ آسان باشد تا زمان محاسبه کوتاه شده و روش کم هزینه گردد. ماتریس زیر را بررسی می‌کنیم:

$$T_{p,q}(\Theta) = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & \cos\Theta & & \sin\Theta & & \\ & & & \ddots & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & -\sin\Theta & & \cos\Theta & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

کلیه عناصر ماتریس $S_{pq}(\Theta)$ بجز عناصر قطر آن و عنصر واقع بر سطر P ام، ستون q ام و سطر qام، ستون Pام برابر صفراست. بسادگی می‌توان نشان داد که:

$$SS^T = I \qquad \text{یا} \qquad S^T = S^{-1}$$

در اینصورت معکوس ماتریس S به آسانی قابل محاسبه می‌باشد.

۷.۴ روش ژاکوبی (برای ماتریس متقارن)

با توجه به توضیحات قبلی ماتریس A_n را به ماتریس A_{n+1} تبدیل می‌کنیم.

$$A_{n+1} = S_{n+1} A_n S^T_{n+1}$$

تا $B = A_{n+1}$ به سمت یک ماتریس قطری میل نماید.

توجه داریم که اگر A_n ماتریسی متقارن باشد، A_{n+1} نیز متقارن خواهد بود:

$$A^T_{n+1} = (S_{n+1} A_n S^T_{n+1})^T = (S^T_{n+1})^T A^T_n S^T_{n+1} =$$
$$S_{n+1} A_n S^T_{n+1} = A_{n+1}$$

با این روش می‌کوشیم که کلیه عناصر غیرواقع بر قطر ماتریس‌های $A_n, \ldots, A_2, A_1$ را در هر مرحله

صفر نمائیم. در مرحله nام فرض می‌کنیم عنصر سطر pام و ستون qام از ماتریس A_n بوده و بزرگترین عنصر غیر واقع بر قطر ماتریس از نظر قدر مطلق باشد.

Θ_r را طوری انتخاب می‌کنیم که عنصر نظیر a_{pq} در A_{n+1} صفر گردد.

$$\cos^2\Theta\ a_{pq} - \sin^2\Theta\ a_{pq} + \sin\Theta\ \cos\Theta(a_{qq} - a_{pp}) = 0$$

برای محاسبه Θ از رابطه فوق استفاده کرده و باتوجه به اینکه:

$$a_{pq} = a_{qp}$$

خواهیم داشت که:

$$tg\ 2\Theta = \frac{2a_{pq}}{a_{pp} - a_{qq}}$$

این عمل برای دنباله A_n بصورت نامحدود می‌باشد، زیرا عناصری که در مرحله A_{n+1} صفر می‌شوند، در مرحله بعدی مجدداً غیر صفر خواهند شد. اما بهرحال این عمل همگراست و درمرحله ای که عناصر غیر واقع بر قطر از نظر قدر مطلق از عدد مثبت ε کوچکتر بوده و قابل صرفنظر کردن باشند، عمل خاتمه می‌پذیرد.

مثال ۸.۴ مطلوبست تعیین مقادیر ویژه ماتریس A_1

$$A_1 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 1 \\ 3 & 1 & 1 \end{bmatrix}$$

بااستفاده از روش ژاکوبی داریم:

$$p = 1\ ,\ q = 3,\ a_{13} = 3\ ،\ tg\ 2\Theta_1 = \frac{2a_{13}}{a_{11} - a_{33}} = \frac{2 \times 3}{1 - 1} = \infty$$

$$\Theta_1 = \frac{\pi}{4}\ ,\quad S_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}\ ,\quad A_2 = \begin{bmatrix} 4 & 2.1213 & 0 \\ 2.1213 & 4 & -.7071 \\ 0 & -.7071 & -2 \end{bmatrix}$$

$$A_2 = S_1\ A\ S_1^T$$

$$P = 1 \ , q = 2 \ , tg\ 2\Theta_2 = \frac{2a_{12}}{a_{11} - a_{22}} = \frac{2 \times 2\ .1213}{4\ -4} = \infty$$

$$\theta_2 = \frac{\pi}{4}$$

$$, \ S_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \ , \ A_3 = \begin{bmatrix} 6.1213 & 0 & -.5 \\ 0 & 1.8787 & -.5 \\ -.5 & -.5 & -2 \end{bmatrix}$$

$$P = 1 \ , q = 3 \ , tg2\ \Theta_3 = \frac{2 \times (-.5)}{6.1213+2} = -\frac{1}{8.12113} = -.123133$$

$$\Theta_3 = -\frac{\pi}{50.02} = -06126$$

$$, S_3 = \begin{bmatrix} .998 & 0 & -.061 \\ 0 & 1 & 0 \\ .061 & 0 & .998 \end{bmatrix} \ , \ A_4 = \begin{bmatrix} 6.15 & .0305 & 0 \\ .0305 & 1.8787 & -.496 \\ 0 & -.496 & -2.025 \end{bmatrix}$$

$$P = 2 \ , q = 3 \ , tg2\ \Theta_4 = \frac{2 \times (-.496)}{1.8787 + 2.006} = -.2553607$$

$$\Theta_4 = -\frac{\pi}{25.13} = -.125$$

$$S_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & .992 & -.125 \\ 0 & .125 & .992 \end{bmatrix} \quad A_5 = \begin{bmatrix} 6.15 & .030 & .004 \\ .030 & 1.94 & .004 \\ .004 & .004 & -2.09 \end{bmatrix}$$

$$A_5 \approx \begin{bmatrix} 6.15 & 0 & 0 \\ 0 & 1.94 & 0 \\ 0 & 0 & -2.09 \end{bmatrix}$$

$$\lambda_1 = 6.15 \ , \lambda_2 = 1.94 \ , \ \lambda_3 = -2.09$$

۸.۴ روش گیونز[1](برای ماتریس‌های متقارن)

در روش ژاکوبی تعداد عملیاتی که لازمه رسیدن به جواب است مشخص نیست و باید عملیات راادامه داد تا عناصر غیر واقع بر قطر قابل صرفنظر کردن باشند. در روش گیونز، ابتدا با

۱- *Givens*

تعداد عملیات مشخصی، ماتریس را بصورت سه قطری[1] در می‌آوریم. اگر ماتریس از قبل بصورت سه قطری باشد این روش نسبت به روش ژاکوبی مناسبتر است.

در ماتریس تبدیل S، عناصر a_{pq} را با اندیس $(p\ ,\ q)$ بشکل زیر انتخاب می‌کنیم:

$$
\begin{array}{l}
(2,3)\ , \\
(2,4)\ \ , \ldots ,\ \ \ (2,n) \\
\qquad (3,4)\ \ , \ldots ,\ \ \ (3,n) \\
\qquad\ (4,5)\ \ ,\ldots ,\ \ (4,n) \\
\qquad\qquad\ \vdots \\
\qquad\qquad\ (n-1,n)
\end{array}
$$

و در هر مرحله بجای آنکه a_{pq} را صفر کنیم، Θ را طوری انتخاب می‌کنیم که عنصر a_{p-1q} از A_n صفر شود یعنی:

$$cos\Theta\ a_{p-1}\ q\ -\ sin\Theta\ a_{p-1}\ p=\ 0$$

$$\boxed{\tan\Theta=\frac{a_{p-1}{}^{q}}{a_{p-1}{}^{p}}}$$

در اینصورت با ادامه عملیات عناصر غیر واقع بر قطراصلی و دو قطر مجاور آن صفر خواهند شد و عملیات در هر مرحله بر عناصر صفر شده قبل تاثیری نخواهد داشت.

$$A=\begin{bmatrix}\times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\end{bmatrix}\qquad A_1=\begin{bmatrix}\times&\times&0&\times\\ \times&\times&\times&\times\\ 0&\times&\times&\times\\ \times&\times&\times&\times\end{bmatrix}$$

$$A_2=\begin{bmatrix}\times&\times&0&0\\ \times&\times&\times&\times\\ 0&\times&\times&\times\\ 0&\times&\times&\times\end{bmatrix}\qquad A_3=\begin{bmatrix}\times&\times&0&0\\ \times&\times&\times&0\\ 0&\times&\times&\times\\ 0&0&\times&\times\end{bmatrix}$$

ماترسی‌های فوق، صفرهائی که بترتیب ایجاد می‌شوند را نشان میدهند.

اکنون ببینیم چطور باید مقادیر ویژه ماتریس سه قطری متقارن را محاسبه کرد. برای بدست آوردن مقادیر ویژه سه قطری ماتریس B که در زیر نشان داده شده است

۱- *Three Diagonal*

$$B=\begin{bmatrix} \alpha_1 & \beta_2 & & 0 & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \beta_4 & \\ & & \beta_4 & \beta_{n-1} & \ddots\ \beta_n \\ 0 & & & \beta_n & \alpha_n \end{bmatrix}$$

معادله $By = \lambda y$ را حل می‌کنیم و داریم: $(B - \lambda I)y = 0$

$$B-\lambda I=\begin{bmatrix} \alpha_1-\lambda & \beta_2 & & & 0 \\ \beta_2 & \alpha_2-\lambda & \beta_3 & & \\ & \beta_3 & \alpha_3-\lambda & & \\ & & & \ddots & \beta_n \\ 0 & & & \beta_n & \alpha_n-\lambda \end{bmatrix}$$

اگر فرض کنیم $f_1, f_2, \ldots, f_n$ دترمینان‌های وابسته به قطراصلی را نشان می‌دهند، با توجه به اینکه برای محاسبه $det\ (B - \lambda I)$ کافی است آنرا نسبت به یکی از سطرها بسط دهیم، خواهیم داشت: $f_0 = 1$

$$f_1=\alpha_1-\lambda$$

$$f_2=(\alpha_2-\lambda)f_1-\beta_2^2$$

$$f_3=(\alpha_3-\lambda)f_2-\beta_3^2 f_1$$

$$f_4=(\alpha_4-\lambda)f_3-\beta_4^2 f_2$$

$$\vdots$$

با فرض $f_0=1$ ، $f_1=\alpha_1-\lambda$ در حالت کلی داریم:

$$f_n(\lambda)=(\alpha_n-\lambda)\, f_{n-1}(\lambda)-\beta_n^2\ f_{n-2}\ (\lambda)$$

در حالت کلی داریم:

$$\boxed{f_r(\lambda)=(\alpha_r-\lambda)\, f_{r-1}(\lambda)-\beta_r^2 f_{r-2}(\lambda)}$$

$$r=2,3,4,\ldots,n$$

مشاهده می‌شود که $f_n(\lambda) = det\ (B - \lambda I)$ و مقادیر ویژه ماتریس B، جواب‌های معادله $f_n\ (\lambda) = 0$ می‌باشند و می‌توانیم با بکاربردن یکی از روشهائی که در فصول گذشته ذکرشد (روش نصف کردن) این معادله را حل نمائیم.

مثال ۹.۴ مقادیر ویژه ماتریس سه قطری B را با استفاده از روش نصف کردن بدست آورید.

$$B = \begin{bmatrix} 1.2 & 2.2 & 0 \\ 2.2 & 1. & 2.3 \\ 0 & 2.3 & 1.2 \end{bmatrix}$$

$f_0 = 1$

$f_1 = 1.2 - \lambda$

$f_2 = (1 - \lambda) f_1 - (2.2)^2 f_0$

$f_3 = (1.2 - \lambda) f_2 - (2.3)^2 f_1$

البته در این مثال می‌توانیم معادله مشخصه را بدست آوریم:

$f_3(\lambda) = -\lambda^3 + 3.4\lambda^2 + 6.29\lambda - 10.716$

هرچند در این روش مقصود محاسبه معادله مشخصه نیست.

λ	$-3.$	-2.5	-2.0	-2.25	-2.125	-2.063	-2.094	-2.078	-2.086	-2.082
f_0	1	1	1	1	1	1	1	1	1	1
f_1	4.2	3.7	3.2	3.45	3.325	3.263	3.294	3.278	3.2862	3.282
f_2	11.92	8.11	4.76	6.373	5.551	5.155	5.352	5.253	5.3014	5.275
f_3	28.019	10.437	-1.946	3.736	$.868$	$-.441$	$.204$	$-.1213$	$.036$	$-.05$

با توجه به اینکه $f(-2.082)<0$ و $f(-2.086)>0$، طبق قضیه بولزانو $f_3(\lambda)$ دارای یک ریشه در فاصله $(-2.086, -2.082)$ می‌باشد: $\lambda_1 = -2.084$

چنانکه روش نیوتن را بکار می‌بردیم باید مشتقات $f_r(\lambda)$، $r = 0, 1, 2, \ldots, n$ را در هر مرحله محاسبه می‌کردیم، که در اینصورت عملیات بسیار طولانی‌تر می‌گردید.

لازم به یادآوری است که دنباله $f_0, f_1, \ldots, f_n$ از خاصیت دنباله اشترم[1] تبعیت می‌کند. بدین ترتیب که تعداد تساوی‌هائی که به ترتیب و بطور متوالی در علائم $f_0, f_1, \ldots, f_n$ بازاء مقدار انتخابی λ در هر ستون حاصل می‌شود، نشان دهنده تعداد مقادیر ویژه B می‌باشد که از λ بزرگترند.

در مثال فوق داریم:

$\lambda = -3 \quad , \quad \lambda_1, \lambda_2, \lambda_3 > -3$

$\lambda = -2.5 \quad , \quad \lambda_1, \lambda_2, \lambda_3 > -2.5$

$\lambda = -2 \quad , \quad \lambda_1, \lambda_2 > -2$

پس نتیجه می‌شود: $-2.5 < \lambda_3 < -2$

از این خاصیت نیز می‌توان برای تعیین فاصله‌ای که مقدار ویژه در آن قرار دارد، استفاده کرد.

۱- *sterm*

۹.۴ معکوس کردن ماتریس

در فصل ۳.۳ روش گاوس - جردن را برای حل n معادله n مجهول مورد مطالعه قرار دادیم.

$$\begin{cases} y = Ax \\ y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ y_3 = a_{31}x_1 + a_{32}x_2 + \ldots + a_{3n}x_n \\ \vdots \\ y_n = a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n \end{cases}$$

چنانچه $x_n , \ldots , x_2 , x_1$ را برحسب $y_n , \ldots , y_2 , y_1$ بدست آوریم ماتریس ضرایب y برابر A^{-1} معکوس ماتریس A ، می‌باشد.

$$Iy = A\,x$$
$$A^{-1}\,y = A^{-1}A\,x$$
$$A^{-1}\,y = I\,x$$

یعنی اگر بااستفاده از روش حذفی گاوس تغییراتی را که در ماتریس ضرایب A انجام می‌پذیرد، متناظراً در ماتریس یکه I انجام دهیم، زمانی که ماتریس ضرایب A به ماتریس یکه تبدیل می‌شود ماتریس I به ماتریس معکوس A^{-1}، تبدیل می‌گردد. به این منظور ماتریس $n \times 2n$ زیر را درنظر می‌گیریم.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & 1 & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & 0 & 1 & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

با استفاده از روش حذفی گاوس سعی می‌کنیم n سطر و n ستون سمت چپ ماتریس فوق را به یک ماتریس یکه تبدیل کنیم، با توجه به اینکه عملیات انجام گرفته متناظراً بر روی n سطر و n ستون سمت راست نیز موثر است، در اینصورت آنرا به ماتریس معکوس A تبدیل می‌کند.

توجه شود که در اینجا مقصود عملیات روی ضرایب مجهولات یک دستگاه معادلات می‌باشد که می‌توان جمع جبری دو معادله را به جای یکی از آن معادلات قرار داد و نام ماتریس را اصطلاحاً بجای آن بکار می‌بریم.

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & a_{1\,n+1} & a_{1\,n+2} & \dots & a_{12n} \\ 0 & 1 & 0 & \dots & 0 & a_{2\,n+1} & a_{2\,n+2} & \dots & a_{2\,2n} \\ 0 & 0 & 1 & \dots & 0 & a_{3\,n+1} & a_{3\,n+2} & \dots & a_{3\,2n} \\ \vdots & & & & & & & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 & a_{n\,n+1} & a_{n\,n+2} & \dots & a_{n\,2n} \end{bmatrix}$$

ممکن است معکوس ماتریس A که آنرا D می‌نامیم دارای دقت مطلوب نباشد یعنی $AD_1 \neq I$ یا $I - AD_1 = F_1$ و قدر مطلق عناصر F_1 از ε، عدد کوچک مثبت مفروض، کوچکتر نباشد. در اینصورت با استفاده ازروش تکراری، ماتریس معکوس را بهبود بخشیده تا به دقت مطلوب برسیم:

$$D_{i+1} = D_i\,(I + F_i) \qquad i = 1,2,3,\dots$$
$$I - AD_i = F_i$$

و لازم است که قدر مطلق عناصر F_i از واحد کوچکتر باشند.

مثال ۱۰.۴ معکوس ماتریس زیر را بدست آورید.

$$A = \begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix}$$

روش گاوس را بکار می‌بریم.

$$A_1 = \begin{bmatrix} 2 & 3 & 2 & 1 & 0 & 0 \\ 10 & 3 & 4 & 0 & 1 & 0 \\ 3 & 6 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1 & 3/2 & 1 & 1/2 & 0 & 0 \\ 10 & 3 & 4 & 0 & 1 & 0 \\ 3 & 6 & 1 & 0 & 0 & 1 \end{bmatrix}$$

در سطر اول از ۲ فاکتور گرفته‌ایم. اکنون سطر اول را در $-10.$ ضرب کرده و با سطر دوم جمع می‌کنیم و همینطور سطر اول را در ۳- ضرب کرده و با سطر سوم جمع می‌کنیم و A_3 بدست آید و سپس از ۱۲- در سطر دوم فاکتور می‌گیریم.

$$A_3 = \begin{bmatrix} 1 & 3/2 & 1 & 1/2 & 0 & 0 \\ 0 & -12 & -6 & -5 & 1 & 0 \\ 0 & 3/2 & -2 & -3/2 & 0 & 1 \end{bmatrix} \qquad A_4 = \begin{bmatrix} 1 & 3/2 & 1 & 1/2 & 0 & 0 \\ 0 & 1 & 1/2 & 5/12 & -1/2 & 0 \\ 0 & 3/2 & -2 & -3/2 & 0 & 1 \end{bmatrix}$$

در A_4 سطر دوم را در $-\frac{3}{2}$ ضرب کرده و با سطر اول و سوم جمع می‌کنیم و A_5 بدست می‌آید، سپس از $-\frac{11}{4}$ در سطر سوم فاکتور می‌گیریم تا A_6 بدست آید.

$$A_5 = \begin{bmatrix} 1 & 0 & 1/4 & -1/8 & +1/8 & 0 \\ 0 & 1 & 1/2 & 5/12 & -1/12 & 0 \\ 0 & 0 & -11/4 & -17/8 & 1/8 & 1 \end{bmatrix} \qquad A_6 = \begin{bmatrix} 1 & 0 & 1/4 & -1/8 & 1/8 & 0 \\ 0 & 1 & 1/2 & 5/12 & -1/12 & 0 \\ 0 & 0 & 1 & 17/22 & 1/22 & -4/11 \end{bmatrix}$$

در A_6 سطر سوم را در $\frac{1}{2}$ - ضرب و با سطر دوم جمع می‌کنیم و همین طور سطر سوم را در $-\frac{1}{4}$ ضرب و با سطر اول جمع می‌کنیم.

$$A_7=\begin{bmatrix}1 & 0 & 0 & -7/22 & 3/22 & 1/11\\ 0 & 1 & 0 & -1/33 & -2/33 & 2/11\\ 0 & 0 & 1 & 17/22 & -1/22 & -4/11\end{bmatrix} \qquad A^{-1}=\frac{1}{66}\begin{bmatrix}-21 & 9 & 6\\ 2 & -4 & 12\\ 51 & -3 & -24\end{bmatrix}$$

$$A^{-1}\approx\begin{bmatrix}-.31818182 & .13636364 & .09090909\\ .030303030 & -.060606061 & .18181818\\ .77272727 & -.045454545 & -.36363636\end{bmatrix} \quad , \quad AA^{-1}\approx I$$

برنامه ۱.۴روش توانی

در این قسمت لیست برنامه به نام *PPOWER.F* آورده شده است. برای بدست آوردن بزرگترین مقدار ویژه یک ماتریس و بردار ویژه متناظر به آن، از برنامه فرعی زیر بنام *POWER* استفاده می‌کنیم.

```
      PROGRAM  PPOWER
C
C ------------------------------------------------------------
C
APPLIED  NUMERICAL  ANALYSIS
C
C ------------------------------------------------------------
C
C    THE SUBROUTINE  POWER  THAT  IS  CALLED  HERE
IMPLEMENTS  THE BASIC
C    ITERATIVE  METHOD  FOR  FINDING  THE  LARGEST
EIGENVALUE  (IN
C MAGNITUDE)  AND  ITS CORRESPONDING  EIGENVALUE.
C

      REAL  A(10,10), X(10), TOL, C
      INTEGER  N, NDIM, NLIMIT
C
      DATA  A/7.0,2*1.0,3.0,6*0.0,8.0, 6.0,-2.0,4.0,6*0.0,
     +      6.0,-1.0,5.0,3.0,6*0.0,6.0,2*-2.0,4.0,66*0.0/
      DATA  NDIM, N, NLIMIT/10,  4, 15/
      DATA  X,TOL/1.0,  9*0.0, 0.001/
C
      CALL  PRINTA(A,  NDIM,  N)
C
      CALL  POWER(A,X,C,TOL,NLIMIT,N,NDIM)
C
C
      END
C
C ------------------------------------------------------
C    THIS  SUBROUTINE  PRINTS  OUT  THE  INPUT  N  BY  N
MATRIX  A
C ------------------------------------------------------
C
      SUBROUTINE  PRINTA(A,  NDIM,  N)
C
      INTEGER  N, NDIM,  I, J
      REAL  A(NDIM,  N)
C
      PRINT  100
      DO  10 I = 1, N
```

```
            PRINT  200, (A(I,J),  J = 1,N)
      10     CONTINUE
            PRINT  '(//)'
      100   FORMAT(//,  '  INPUT MATRIX  IS'/)
      200   FORMAT(10F7.2)
            RETURN
            END
      C
      C -------------------------------------------------------
      C  THIS SUBROUTINE  PRINTS  OUT  THE  SCALAR  VALUE
AND
      C  THE NORMALIZED  VECTOR.
      C -------------------------------------------------------
      C
            SUBROUTINE  PRINTVC(X,  C, NDIM,  N)
      C
            INTEGER  N, NDIM,  I
            REAL  X(NDIM),  C
      C
            PRINT  100, (X(I),  I = 1,N)
            PRINT  200, C
      100   FORMAT('   THE NORMALIZED  VECTOR  IS:  ', 10F7.4)
      200   FORMAT('   THE VALUE  FOR  C IS: ', F10.6/)
            RETURN
            END
      C
      C
```

```
      C
      C   SUBROUTINE  POWER  :
      C                    THIS SUBROUTINE  COMPUTES  THE
LARGEST  EIGEN-
      C    VALUE  AND  ITS CORRESPONDING  EIGENVECTOR  BY
THE POWER METHOD.
      C
      C -------------------------------------------------------------
            SUBROUTINE  POWER(A,X,C,TOL,NLIMIT,N,NDIM)
      C
      C   PARAMETERS  ARE  :
      C
      C   A     - AN N X N MATRIX  WHOSE  EIGENS  ARE  BEING
DETERMINED
      C   X     - ESTIMATE  FOR THE EIGENVECTOR  USED  TO
BEGIN  ITERATIONS.
      C          IF NO APPROXIMATION  TO THIS IS KNOWN,  THE
USUAL  CHOICE
      C          IS A VECTOR  WITH ALL COMPONENTS  EQUAL  TO
UNITY.  X ALSO
      C            RETURNS  THE FINAL  EIGENVECTOR  TO THE
CALLER.
      C   C     - RETURNS  THE VALUE  OF THE EIGENVALUE
      C    TOL    - TOLERANCE  VALUE  USED  TO DETERMINE
```

```
CONVERGENCE.  ITERATIONS
      C          CONTINUE  UNTIL  SUCCESSIVE  ESTIMATES  OF  THE
EIGENVALUE
      C             ARE  THE  SAME  WITHIN  TOL  IN  VALUE.
      C     NLIMIT  - LIMIT  TO  THE  NUMBER  OF  ITERATIONS  IF
NOT  CONVERGENT
      C     N       - SIZE  OF  THE  MATRIX  AND  THE  VECTOR  X
      C     NDIM    - FIRST  DIMENSION  OF  MATRIX  A  IN  THE
CALLING  PROGRAM
      C     XWRK    - VECTOR  USED  TO  STORE  INTERMEDIATE
VALUES.  MUST  BE
      C             DIMENSIONED  TO  HOLD  AT  LEAST  N  ELEMENTS
IN  THE
      C             MAIN  PROGRAM.
      C
      C  ---------------------------------------------------------------
      C
          INTEGER  N,NDIM,NLIMIT,IROW,ITER,JCOL
          REAL  A(NDIM,N),X(N),XWRK(10),C,TOL,SAVE
      C
      C  ---------------------------------------------------------------
      C
      C  BEGIN  THE  ITERATIONS.  GET  THE  PRODUCT  OF  A  AND
X,  THEN  NORMALIZE.
      C  THE  NORMALIZATION  FACTORS  SHOULD  CONVERGE  TO
THE  EIGENVALUE,  C,  ON
      C  REPEATED  MULTIPLICATIONS.  WE  STORE  THE  CURRENT
VALUE  OF  C  FOR
      C  COMPARISON  WITH  THE  NEXT  VALUE  TO  TEST
CONVERGENCE.  TO  BEGIN,  WE
      C  MAKE  SAVE  =  0.
      C
          SAVE  =  0.0
          DO  50  ITER  =  1,NLIMIT
            DO  20  IROW  =  1,N
            XWRK(IROW)  =  0.0
            DO  10  JCOL  =  1,N
                              XWRK(IROW)  =  XWRK(IROW)  +
A(IROW,JCOL)*X(JCOL)
         10   CONTINUE
         20   CONTINUE
      C
      C  ---------------------------------------------------------------
      C
      C  FIND  THE  LARGEST  ELEMENT  OF  THE  PRODUCT
VECTOR  FOR  NORMALIZING.
      C
          C  =  0.0
          DO  30  IROW  =  1,N
            IF  (  ABS(C)  .LT.  ABS(XWRK(IROW))  )  C  =  XWRK(IROW)
         30   CONTINUE
      C
```

```
C ------------------------------------------------------------------
C
C  NOW NORMALIZE  THE PRODUCT  VECTOR  AND  PUT
INTO X FOR NEXT ITERATION.
C
      DO 40 IROW = 1,N
        X(IROW)  = XWRK(IROW)   / C
   40 CONTINUE
C
      CALL PRINTVC(X,  C, NDIM,  N)
C
C ------------------------------------------------------------------
C
C SEE IF TOLERANCE  IS MET. IF SO, WE ARE DONE. IF NOT,
WE CONTINUE
C THE ITERATIONS.
C
C      IF ( ABS(C - SAVE)  .LE. TOL ) RETURN
       SAVE = C
   50 CONTINUE
C
C ------------------------------------------------------------------
C
C    IF  WE  DO  NOT  MEET  THE  TOLERANCE  FOR
CONVERGENCE,  PRINT A MESSAGE
C AND RETURN LAST VALUES CALCULATED.
C
      PRINT 200, NLIMIT
  200 FORMAT(/' CONVERGENCE  NOT REACHED  IN ',I5,'
ITERATIONS')
      RETURN
      END
```

تمرینات فصل چهارم

۱ - با استفاده از روش توانی بزرگترین قدر مطلق مقادیر ویژه ماتریس A را بدست آورید.
سپس با استفاده از روش تکراری معکوس کوچکترین قدر مطلق مقادیر ویژه ماتریس A را بدست آورید.

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$$

۲- کوچکترین و بزرگترین قدرمطلق مقادیر ویژه ماتریس B و بردار ویژه متناظر آنرا از روش توانی بدست آورید.

$$B=\begin{bmatrix}8 & -1 & -5\\ -4 & 4 & -2\\ 18 & -5 & -7\end{bmatrix}$$

$|\lambda_1| > |\lambda_2| > |\lambda_3|$

۳- بزرگترین مقدار ویژه از نظر قدر مطلق و بردار ویژه نظیر آن را برای ماتریس A به دست آورید.

$$A=\begin{bmatrix}4 & -1 & 1\\ 1 & 1 & 1\\ -2 & 0 & -6\end{bmatrix}$$

جواب: با شروع از $(1,1,1)^T$ داریم:

d	z
-8	$(-0.5,\ -0.375,\ 1)^T$
-5	$(-0.125,\ -0.025,\ 1)^T$
-6.25	$(-0.244,\ -0.176,\ 1)^T$
$\vdots$	$\vdots$
$\lambda_1=-5.76849$	$-0.1157,\ -0.1306,\ 1)^T$

۴- ماتریس C مفروض است.

$$C=\begin{bmatrix}3 & -4 & 3\\ -4 & 6 & 3\\ 3 & 3 & +1\end{bmatrix}$$

a) با بکار بردن روش توانی بزرگترین قدرمطلق مقادیر ویژه ماتریس C را بدست آورید. (λ_1)
$|\lambda_1| > |\lambda_2| > |\lambda_3|$

b) با بکار بردن روش صفر کردن مقادیر ویژه، مقدار ویژه λ_2 را در ماتریس C تعیین نمائید.

c) روش گیونز را برای تبدیل ماتریس C به ماتریس سه قطری بکار برده و مقدار ویژه سوم را بدست آورید.

۵- برای تعیین مقادیر ویژه، روش تبدیل ماتریس (روش ژاکوبی) را بکار برده و برنامه‌ای به زبان C برای آلگوریتم آن بنویسید و آنرا آزمایش کرده و مقادیر ویژه چند ماتریس مختلف را بدست آورید.

۶- روش ژاکوبی را بکار برده مقادیر ویژه ماتریس A را بدست آورید.

$$A=\begin{bmatrix} 1.0 & 1.0 & .5 \\ 1.0 & 1.0 & .25 \\ .5 & .25 & 2.0 \end{bmatrix}$$

جواب: $\lambda_1= 2.5365258 \qquad \lambda_2= .0166473 \qquad \lambda_3= 1.480125$

۷- در تمرین ۶ ماتریس A را تبدیل به ماتریس سه قطری B نمائید و سپس معادله مشخصه ماتریس A را تعیین نمائید. (روش گیونز)

$$B=\begin{bmatrix} 1 & \frac{\sqrt{5}}{2} & 0 \\ \frac{\sqrt{5}}{2} & 1.40 & .55 \\ 0 & .55 & 1.60 \end{bmatrix}$$

جواب: معادله مشخصه: $f(\lambda) = \lambda^3 - 4\lambda^2 + 3.6875\lambda + .625$

۸- فاصله‌ای که مقادیر ویژه ماتریس A به آن متعلق هستند تعیین نمائید و سپس بااستفاده از روش گیونز مقدار ویژه‌ای را که درفاصله $[10,20]$ قرار دارد بدست آورید.

$$A=\begin{bmatrix} 120.0 & -90.86 & 0 & 0 \\ -90.86 & 157.20 & 67.59 & 0 \\ 0 & 67.59 & 124.0 & -46.26 \\ 0 & 0 & -46.26 & 78.84 \end{bmatrix}$$

جواب: فاصله $I=[-1.25,315.65]$ و $\lambda = 16.013$ مقدار ویژه

۹- سه مقدار ویژه ماتریس A را با استفاده از روش توانی بدست آورید.

$$A=\begin{bmatrix} 7 & 3 & -2 \\ 3 & 4 & -1 \\ -2 & -1 & 3 \end{bmatrix}$$

جواب: $\lambda_1 = 9.606$ بزرگترین $\lambda_2 = 2.000$ کوچکترین و $\lambda_3 = 2.394$ نزدیکترین مقدار ویژه نسبت به $\rho = 5.803$

۱۰ - مقادیر ویژه ماتریس A زیر را بدست آورید.

سپس ماتریس A را معکوس کرده و مقادیر ویژه آنرا بدست آورید و نشان دهید که این مقادیر عکس مقادیر ویژه ماتریس A می‌باشد و بردارهای ویژه هر دو یکسان هستند.

$$A=\begin{bmatrix} -5 & 2 & 1 \\ 1 & -9 & -1 \\ 2 & -1 & 7 \end{bmatrix}$$

جواب:

مقادیر ویژه و بردارهای ویژه برابرند با:

$$-4.6242;\ [0.95724,\ 0.251543,\ -0.14308]$$

$$7.2024;\ [0.37006,\ 0.92577,\ 0.077473]$$

$$-9.5782;\ [0.41719,\ -0.90275,\ -0.10479]$$

مقادیر ویژه A^{-1} برابر است با:

$$-0.10440\ ,\ 0.13884,\ -0.21625$$

که عکس مقادیر ویژه ماتریس A می‌باشد.

۱۱ - *a)* معادله مشخصه و مقادیر ویژه ماتریس A را بدست آورید:

$$A=\begin{bmatrix} 5 & 3 & 2 \\ -2 & 6 & 3 \\ 3 & 2 & 4 \end{bmatrix}$$

b) اگر به هر عنصر ماتریس A، 0.1 اضافه کنیم، جواب را مجدداً بدست آورید.

جواب: معادله مشخصه برابر است با:

$$-\lambda^3 + 15.3\lambda^2 - 69.9\lambda + 100.2$$

و مقادیر ویژه برابرند با:

$$\lambda_1 = 8.39608$$

$$\lambda_2\ ,\ \lambda_3 = 3.45195 \pm 0.134597i$$

فصل ۵

درونیابی و برازش منحنی

موضوعات این فصل

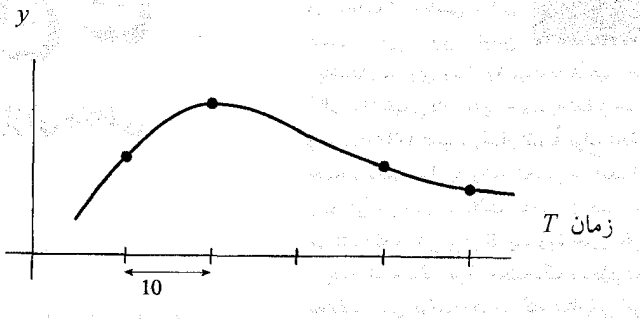
* مسأله درونیابی
* چند جمله‌ای‌های لاگرانژ
* تفاضل‌های محدود
* داده‌های متساوی الفاصله
* درونیابی با اسپلاین درجه سه
* خم‌های بی‌زیّر (Bezier) و B-اسپلاین (B-Spline)
* تقریب چند جمله‌ای سطوح
* تقریب حداقل مربعات
* برنامه‌های کامپیوتری
* لیست برنامه تفاضل‌های محدود به زبان C
* لیست برنامه B اسپلاین به زبان PASCAL
* لیست برنامه حداقل مربعات به زبان PASCAL
* تمرینات فصل پنجم

ژوزف لوئی لاگرانژ
LAGRANGE
(۱۷۳۶ - ۱۸۱۳)

لاگرانژ در سال ۱۷۳۶ در شهرتورن ایتالیا به دنیا آمد. کارهای لاگرانژ تأثیر عمیقی در تحقیقات ریاضی بعدی داشت، زیرا وی اولین ریاضیدان ممتازی بود که وضعیت کاملاً غیر رضایت‌بخش مبانی آنالیز را تشخیص داد و بدین جهت به تدقیق حسابان همت گمارد و در سال ۱۷۹۷ کتاب بزرگ او **نظریهٔ توابع تحلیلی شامل اصول حساب دیفرانسیل** به رشته تحریر در آمد. اثر دیگر لاگرانژ رساله‌ای در حل معادلات عددی از کلیه درجات در ۱۷۶۷ می‌باشد؛ که روشی برای تقریب ریشه‌های حقیقی یک معادله را می‌دهد. اثر ماندگار دیگر مکانیک تحلیلی (۱۷۸۸) می‌باشد و معادلات کلی حرکت یک دستگاه دینامیکی امروزه به **معادلات لاگرانژ** موسوم‌اند، که «همیلتن» آن را یک منظومهٔ علمی "scientific poem" توصیف کرده است.

۱.۵ مسأله درونیابی[۱]

فرض کنیم مقادیر بزرگی از داده‌ها در مورد حرکت راکت جدیدی را در اختیار داریم. سیگنال‌های مسافت سنج هر *10* ثانیه دریافت می‌شوند و موقعیت راکت و سایر اطلاعات را در اختیار ما می‌گذارند. چطور می‌توانیم موقعیت راکت را در زمان‌های میانی تعیین کنیم؟ این یک مسأله درونیابی است. نباید تصور کرد که موقعیت‌ها نسبت به زمان خطی هستند. گاهی نقاط شبیه شکل زیر رسم می‌شوند. همچنانکه مشاهده می‌شود، گاهی یک سیگنال را مانند نقطه سوم از دست می‌دهیم.



می‌خواهیم مقادیر نقاط میانی را بدست آوریم، حتی وقتی منحنی دارای ماکزیمم یا می‌نیمم است. علاوه بر این، روشی با کارآیی مطلوب مورد نیاز است، زیرا چندین حالت جهت انجام آن وجود دارند.

در این فصل فنون مؤثری جهت درونیابی بدست خواهیم آورد، مخصوصاً برای موقعیتهایی که داده‌ها خطی نیستند. اصلی که بکار برده خواهد شد برازش منحنی چند جمله‌ای به نقاط است.

بسیاری از بسط و بررسی این موضوع توسط نیوتن و کپلر انجام شده است، زیرا که آنان داده‌های موقعیت ستارگان را تجزیه و تحلیل می‌کردند.

۲.۵ چندجمله‌ای‌های لاگرانژ[۲]

در این بخش و سه بخش بعدی می‌پذیریم که داده‌ها دقیق هستند. جدول داده‌ها نشان دهنده مقادیر یک تابع ناشناخته هستند. اگر مایلیم یک چند جمله‌ای پیدا کنیم که از نقاط این تابع (x_i , f_i) بگذرد، می‌توانیم دستگاه معادلات ضرایب یک چند جمله‌ای را بکار بریم.

بعنوان مثال، فرض می‌کنیم می‌خواهیم یک چند جمله درجه سومی را به داده‌های زیر برازش کنیم:

۱ – **Interpolation problem**

۲– *Lagrangian Polynomial*

x	$f(x)$
3.2	22.0
2.7	17.8
1.0	14.2
4.8	38.3
5.6	51.7

ابتداء برای تعیین چند جمله‌ای به چهار نقطه نیاز داریم. فرض کنید اولین چهار نقطه را انتخاب می‌کنیم. اگر چند جمله‌ای درجه سوم برابر ax^3+bx^2+cx+d باشد، می‌توانیم چهار معادله با چهار مجهول d, c, b, a را بنویسیم:

$$x = 3.2: a(3.2)^3 + b(3.2)^2 + c(3.2) + d = 22.0,$$
$$x = 2.7: a(2.7)^3 + b(2.7)^2 + c(2.7) + d = 17.8,$$
$$x = 1.0: a(1.0)^3 + b(1.0)^2 + c(1.0) + d = 14.2,$$
$$x = 4.8: a(4.8)^3 + b(4.8)^2 + c(4.8) + d = 38.3.$$

این دستگاه را با یکی از روش‌های فصل سوم حل می‌کنیم و چند جمله‌ای را بدست می‌آوریم. سپس می‌توانیم مقدار تابع را در هر نقطه مانند $x = 3.0$ تخمین بزنیم.

بعنوان مثال، جواب دستگاه فوق برابر است با:

$$a = -0.5275$$
$$b = 6.4952$$
$$c = -16.1177$$
$$d = 24.3499$$

و چند جمله‌ای بدست می‌آید.

$$-0.5275x^3 + 6.4952x^2 - 16.1177x + 24.3499$$

در $x = 3.0$ مقدار تقریبی تابع برابر با 20.21 می‌باشد.

ما طریق بهتر وساده‌تری برای چنین چند جمله‌ای درونیاب پیدا می‌کنیم. اگر بخواهیم یک چند جمله‌ای تعیین کنیم که از نقطه پنجم $(5.6, 51.7)$ نیز بگذرد یا بجای چند جمله‌ای درجه سوم یک چند جمله‌ای درجه دوم عبور دهیم و بخواهیم با جواب قبلی مقایسه کنیم، کار انجام عملیات بسیار وقت‌گیر وکسل کننده است. به‌علاوه این فن گاهی منجر به‌یک دستگاه معادلات ناهنجار (ناجور) می‌گردد.

ابتداء به یک روش کاملاً مستقیم، یعنی چند جمله‌ای لاگرانژ توجه می‌کنیم. شاید روش چند جمله‌ای لاگرانژ ساده‌ترین راه برای نمایش وجود چند جمله‌ای درونیاب برای داده‌هائی با فواصل نامساوی باشد. فرض کنیم یک جدول از داده‌ها با مقادیر چهار نقطه (x_i , f_i) داشته باشیم.

x	$f(x)$
x_0	f_0
x_1	f_1
x_2	f_2
x_3	f_3

در اینجا مقادیر x به ترتیب خاصی مرتب نشده‌اند و لزومی ندارد فواصل مقادیر x برابر باشند. بهر حال مقادیر x جدا از یکدیگرند. می‌توانیم از این چهار زوج مقادیر یک چند جمله‌ای درجه سوم بگذرانیم. شکل لاگرانژ آن مطابق زیر است.

$$P_3(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} f_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} f_1 + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} f_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} f_3 \qquad (1.5)$$

توجه کنید که این معادله از چهار جمله تشکیل شده است که هر یک نسبت به x درجه سوم می‌باشند، از اینرو مجموع آنها نیز درجه سوم می‌باشد.

و داریم:

$$P_3(x_0) = f_0$$
$$P_3(x_1) = f_1$$
$$P_3(x_2) = f_2$$
$$P_3(x_3) = f_3$$

برای یک چند جمله‌ای درجه n لاگرانژ داریم:

$$P_n(x) = \sum_{i=0}^{n} \phi_i(x) f_i$$

بطوری که:

$$\phi_i(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})(x-x_{i+2})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$$

$$\begin{cases} \phi_i(x_i) = 1 \\ \phi_i(x_j) = 0 \qquad i \neq j \end{cases}$$

واضح است که چند جمله‌ای لاگرانژ از کلیه نقاط داده شده می‌گذرد.

$$P_n(x_i) = \phi_i(x_i) f_i = f_i \qquad \text{و } i = 0, 1, 2, \dots, n$$

مثال ۱.۵ یک چند جمله‌ای درجه سه که از چهار نقطه اول جدول قبل می‌گذرد برازش کنید و آنرا برای بدست آوردن درونیابی مقدار $x = 3.0$ بکار برید.

در مثال قبل بازاء $x = 3$ داریم:

$$P_3(3.0) = \frac{(3.0-2.7)(3.0-1.0)(3.0-4.8)}{(3.2-2.7)(3.2-1.0)(3.2-4.8)}(22.0)$$
$$+\frac{(3.0-3.2)(3.0-1.0)(3.0-4.8)}{(2.7-3.2)(2.7-1.0)(2.7-4.8)}(17.8)$$
$$+\frac{(3.0-3.2)(3.0-2.7)(3.0-4.8)}{(1.0-3.2)(1.0-2.7)(1.0-4.8)}(14.2)$$
$$+\frac{(3.0-3.2)(3.0-2.7)(3.0-1.0)}{(4.8-3.2)(4.8-2.7)(4.8-1.0)}(38.3).$$

مشاهده می‌شود که همان جواب قبلی بدست آمده است. $P_3(3.0) = 20.21$

هر چند انجام عملیات با ماشین حساب مناسب است، لیکن محاسبات خسته کننده است. می‌توان یک برنامه کامپیوتری برای این روش نوشت. وقتی یک چند جمله‌ای درونیاب که از کلیه نقاط می‌گذرد برای درونیابی بکار می‌بریم عموماً جواب دقیقی بدست نمی‌آید. چون اغلب داشتن $n+1$ نقطه دلیل استفاده از چند جمله‌ای درجه n نمی‌باشد.

در این فصل خواهیم دید که جمله خطا برابر است با

$$E(x) = (x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!} \qquad (2.5)$$

که ξ در کوچکترین فاصله‌ای است که شامل $\{x, x_0, x_1, \ldots x_n\}$ می‌باشد.

داشتن این عبارت خطا در معادله (2.5) جالب است ولی همیشه خیلی مفید نیست. زیرا تابعی که بازاء x_i مقدار f_i را تولید کرده است شناخته شده نیست و مشتق $n+1$ ام آن را نیز نمی‌شناسیم. بهرحال می‌توان نتیجه گرفت که اگر تابع «هموار»[1] باشد، باید یک چند جمله‌ای درجه n یا کمتر به آن نظیر شود و می‌توان نتیجه گرفت که برون یابی (بکار بردن چند جمله‌ای درونیاب برای خارج از دامنه‌ی مقادیر x که چند جمله‌ای ساخته شده است) دارای خطای بیشتری از درونیابی می‌باشد. همچنین اگر x به نقاط مرکزی x_i نزدیک باشد خطا کوچکتر است، زیرا که حاصلضرب جملات $(x-x_i)$ کوچکتر می‌باشند.

۱ – smooth

برای $n+1$ نقطه داده شده بازاء مقدار x مقدار $f(x)$ را از طریق درونیابی بدست آورید.

(x_i , f_i), $i = 0, \ldots , N$:

```
Set SUM = 0
DO FOR I = 0 to N:
  SET P = 1.
DO FOR j = 0 to N:
    IF J ≠ I:
       SET P = P*(x - x(J))/(x(I) - x(J)).
ENDDO (J).
  SET SUM = SUM + P*f_i
ENDDO (I).
SUM
```

مقدار جواب درونیابی

۳.۵ روش نیویل[1]

در روش چند جمله‌ای لاگرانژ چون درجه چند جمله‌ای را نمی‌دانیم، اگر درجه چند جمله‌ای خیلی پایین باشد، چند جمله‌ای درونیاب تخمین خوبی از $f(x)$ را نمی‌دهد. اگر درجه چند جمله‌ای خیلی بالا باشد، نوسان غیر قابل قبولی در مقادیر چند جمله‌ای رخ می‌دهد. (بخش خم‌های اسپلاین را ببینید.) روش نیویل بر این مشکل غلبه می‌کند. در این روش مقدار درونیابی با چند جمله‌ای‌های درجه بالاتر بطور متوالی محاسبه می‌شود، وقتی مقادیر متوالی به قدر کافی بهم نزدیک شدند عملیات را خاتمه می‌دهیم.

برای درونیابی خطی مقادیر تقریبی را بطور متوالی از نزدیک‌ترین نقاط جدول داده‌ها نسبت به x شروع به محاسبه می‌کنیم. در این صورت بازاء هر مقدار x برای محاسبه $f(x)$ مقادیر داده در جدول را برحسب نزدیک‌ترین مقدار x_i نسبت به x نامگذاری می‌کنیم بطوریکه:

$$|x_0 - x| \le |x_1 - x| \le \ldots \le |x_n - x|$$

و از معادله (2.5) جمله‌ی خطای درونیاب لاگرانژ مشاهده می‌شود که هرچه x_i ها به x نزدیکتر باشند، خطا کمتر است.

در چند جمله‌ای لاگرانژ برای درونیابی خطی با داشتن (x_1 , f_1) و (x_2 , f_2)، مقدار $f(x)$ بدست می‌آید.

$$f(x) = \frac{(x - x_2)}{(x_1 - x_2)} f_1 + \frac{(x - x_1)}{(x_2 - x_1)} f_2$$

می‌توانیم این رابطه را فشرده‌تر بنویسیم:

۱- Neville's Method

$$f(x) = \frac{(x - x_2)*f_1+(x_1 - x)* f_2}{x_1 - x_2} \qquad (3.5)$$

که این معادله در روش نیویل بکار می‌رود.

اگر مقادیر f_i را p_{i0} بنامیم، مقدار تخمین آن را در ستون‌های بعدی p_{ij} خواهد بود. بطوریکه p_{i1} نمایش چند جمله‌ای درجه یک و p_{i2} نمایش چند جمله‌ای درجه 2 و در حالت کلی P_{ij} برای محاسبه چند جمله‌ای درجه j بکار می‌رود. معادله *(4.5)* را ملاحظه کنید.

مثال ۲.۵ مقادیر داده‌های یک تابع در زیر داده شده است.

می‌خواهیم برای $x = 27.5$ درونیابی کنیم.

x	$f(x)$
10.1	0.17537
22.2	0.37784
32.0	0.52992
41.6	0.66393
50.5	0.63608

ابتداء مقادیر داده‌ها را برحسب نزدیکترین مقدار به $x = 27.5$ مرتب می‌کنیم.

i	$\lvert x - x_i \rvert$	x_i	$f_i = P_{i0}$
0	4.5	32.0	0.52992
1	5.3	22.2	0.37784
2	14.1	41.6	0.66393
3	17.4	10.1	0.17537
4	23.0	50.5	0.63608

در روش نیویل یک جدول می‌سازیم که ستون اول درونیابی خطی بین زوج مقادیر تابع برای $i=0,1$ و $i=1,2$ و $i=2,3$ و... می‌باشند. ستون بعدی جدول با درونیابی خطی از مقادیر ستون قبلی جدول بوجود می‌آید، بطوریکه $i = 0,2$ و $i=1,3$ و $i=2,4$ و... می‌باشند و همینطور در ستون بعدی با درونیابی ستون قبلی با انتخاب $i=0,3$ و $i=1,4$ و $i=2,5$ و... بوجود می‌آید تا آنکه به جواب دقیقی برسیم یا کلیه نقاط مورد استفاده قرار گیرند.

$$P_{i,j} = \frac{(x - x_i) * P_{i+1,j-1} + (x_{i+j} - x) * P_{i,j-1}}{x_{i+j} - x_i} \qquad (4.5)$$

i	x_i	P_{i0}	P_{i1}	P_{i2}	P_{i3}	P_{i4}
0	32.0	0.52992	0.46009	0.46200	0.46174	0.45754
1	22.2	0.37784	0.45600	0.46071	0.47901	
2	41.6	0.66393	0.44524	0.55843		
3	10.1	0.17537	0.37379			
4	50.5	0.63608				

مقادیر p_{01} و p_{11} از معادله *(4.5)* محاسبه می‌شوند.

$$P_{01} = \frac{(27.5 - 32.0)*0.37784 + (22.2 - 27.5)*0.52992}{22.2 - 32.0} = 0.46009,$$

$$P_{11} = \frac{(27.5 - 22.2)*0.66393 + (41.6 - 27.5)*0.37784}{41.6 - 22.2} = 0.45600.$$

بعد از محاسبه p_{i1}ها ستون بعدی را محاسبه می‌کنیم.

$$P_{22} = \frac{(27.5 - 41.6)*0.37379 + (50.5 - 27.5)*0.44524}{50.5 - 41.6} = 0.55843.$$

به عنوان مثال مقادیر جدول فوق از $sin(x)$ بازاء x برحسب درجه می‌باشد و مقدار صحیح آن بازاء $x=27.5$ برابر 0.46175 می‌باشد. در سطر اول جدول ملاحظه می‌شود درجه چند جمله‌ای مرتباً افزایش می‌یابد و مقدار تابع در x دقت بیشتری نسبت به جواب پیدا می‌کند و زمانی که تفاضل دو مقدار متوالی به حد کافی کوچک شود، عملیات را خاتمه می‌دهیم. در این مثال p_{i3} جواب مناسب است که یک چند جمله درجه سوم در همسایگی جواب را در x_0, x_1, x_2, x_3 نشان می‌دهد و این همان جوابی است که از چند جمله‌ای درجه سه لاگرانژ می‌توانستیم بدست آوریم.

فرمول کلی محاسبه مقادیر جدول مطابق معادله *(4.5)* می‌باشد.

و رابطه کلی بین مقادیر جدول در زیر نشان داده شده است.

i	x_i	$p_{i,0}$	$p_{i,1}$	$p_{i,2}$	$p_{i,3}$	$p_{i,4}$
0	x_0	$p_{0,0}$	$p_{0,1}$	$p_{0,2}$	$p_{0,3}$	$p_{0,4}$
1	x_1	$p_{1,0}$	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$	
2	x_2	$p_{2,0}$	$p_{2,1}$	$p_{2,2}$		
3	x_3	$p_{3,0}$	$p_{3,1}$			
4	x_4	$p_{4,0}$				

توجه شود که مقادیر $p_{i,1}$ نمایش یک چند جمله‌ای درجه یک می‌باشد که از دو نقطه متوالی می‌گذرد. (از معادله *(4.5)* استفاده می‌کنیم).

$$p_{i,1}(x_i) = p_{i,0}(x_i) = f_i$$

$$p_{i,1}(x_{i+1}) = p_{i+1,0}(x_{i+1}) = f_{i+1}$$

همچنین $P_{i,2}$ نمایش یک چند جمله‌ای درجه دوم می‌باشد که از سه نقطه متوالی می‌گذرد و با استفاده از معادله *(4.5)* خواهیم داشت:

$$p_{i,2}(x_i) = p_{i,1}(x_i) = f_i$$

$$p_{i,2}(x_{i+1}) = \frac{(x_{i+1} - x_i)*p_{i+1,1}(x_{i+1}) + (x_{i+2} - x_{i+1})*p_{i,1}(x_{i+1})}{x_{i+2} - x_i} = f_{i+1}$$

$$p_{i,2}(x_{i+2}) = p_{i+1,1}(x_{i+2}) = f_{i+2}$$

۴.۵ تفاضل‌های محدود[۱]

برای استفاده از روش چند جمله‌ای لاگرانژ جهت درونیابی باید به دو نکته توجه کرد. اول آنکه دارای عملیات محاسباتی بیش از روش تفاضل‌های محدود است که مورد بحث ما می‌باشد. دوم اگر مایل باشیم یک نقطه از جدول را کم یا زیاد کنیم باید مجدداً عملیات را شروع کنیم که بعلت افزایش محاسبات این موضوع اهمیت بیشتری دارد و در این صورت در هر دو روش چند جمله‌ای‌های لاگرانژ و روش نیویل باید کلیه محاسبات را برای درونیابی در نقطه جدید x تکرار کنیم. ولی در روش تفاضل‌های محدود کلیه این محاسبات غیر ضروری است.

در واقع، ما چند جمله‌ای دیگری متفاوت از روش لاگرانژ بدست نخواهیم آورد و در قسمت‌های بعد نشان خواهیم داد که از $n+1$ نقطه فقط یک چند جمله‌ای درجه n عبور می‌کند. لیکن در تفاضل‌های محدود فقط طریق بیان چند جمله‌ای متفاوت است.

برای بحث در جدول تفاضل‌های محدود فرض می‌کنیم تابع $f(x)$، برای چند مقدار x شناخته شده است.

x	$f(x)$
x_0	f_0
x_1	f_1
x_2	f_2
x_3	f_3

لزومی ندارد که xها هم فاصله باشند یا به ترتیب خاصی مرتب شده باشند. (هرچند ترتیب آنها یا هم فاصله بودن آنها ممکن است سودمند باشد.)

چند جمله درجه n زیر که به نحو خاصی نوشته شده است را بررسی می‌کنیم:

$$P_n(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1})a_n. \tag{5.5}$$

اگر مقادیر a_i را طوری انتخاب کنیم که در $n+1$ نقطه شناخته شده $(x_i, f_i),\ i = 0,\ldots,n$ داشته باشیم $P_n(x) = f(x)$، در اینصورت $p_n(x)$ یک چند جمله‌ای درونیاب می‌باشد. می‌خواهیم نشان دهیم که مقادیر a_i با استفاده از روشی معین می‌شوند که تفاضل‌های محدود مقادیر جدول شده نامیده می‌شود

یک علامت استاندارد خاص برای تفاضل‌های محدود بکار می‌رود.

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = (f_0^{[1]})$$

$$f[x_1, x_2] = \frac{f_2 - f_1}{x_2 - x_1} = (f_1^{[1]})$$

۱- Finite Differences

که اولین تفاضل محدود بین x_0 و x_1 یا بین x_1 و x_2 نامیده می‌شود.
(یک علامت غیر استاندارد که کوتاه‌تر است به آن افزوده‌ایم.) تابع

$$f[x_s, x_t] = \frac{f_t - f_s}{x_t - x_s}$$

اولین تفاضل محدود بین x_s و x_t می‌باشد. (توجه شود که ترتیب نقاط اهمیتی ندارد.)

$$f[x_s\ ,\ x_t] = \frac{f_t - f_s}{x_t - x_s} = \frac{f_s - f_t}{x_s - x_t} = f[x_t\ ,\ x_s].$$

دومین تفاضل محدود مرتبه بالاتر برحسب تفاضل محدود پایین‌تر تعریف می‌شود.

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = (f_0^{[2]});$$

و تفاضل محدود مرتبه n تعریف می‌شود:

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0} = (f_0^{[n]}).$$

این مفهوم حتی برای تفاضل مرتبه صفر بیان می‌شود:

$$f[x_s] = f_s = \left(f_s^{[0]}\right)$$

علامت استاندارد را بکار می‌بریم، جدول تفاضل محدود بصورت سمبلیک در جدول ۱.۵ نشان داده شده است. (این داده‌ها با داده‌های قسمت *2.5* برابرند.)

جدول ۱.۵

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_0	f_0			
x_1	f_1	$f[x_0, x_1]$		
x_2	f_2	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
x_3	f_3	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$
x_4	f_4	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_0, x_1, x_2, x_4]$

جدول ۲.۵

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, \ldots, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$	$f[x_i, \ldots, x_{i+4}]$
3.2	22.0				
		8.400			
2.7	17.8		2.856		
		2.118		−0.528	
1.0	14.2		2.012		0.256
		6.342		0.0865	
4.8	38.3		2.263		
		16.750			
5.6	51.7				

اکنون آماده‌ایم که مقادیر a_i از معادله *(5.5)* را با تفاضل‌های محدود بدست آوریم. معادله

(5.5) را بازاء $x = x_0$ و $x = x_1$ و ... $x = x_n$ می‌نویسیم.

$$
\begin{aligned}
x = x_0: &\quad P_n(x_0) = a_0,\\
x = x_1: &\quad P_n(x_1) = a_0 + (x_1 - x_0)a_1,\\
x = x_2: &\quad P_n(x_2) = a_0 + (x_2 - x_0)a_1 + (x_2 - x_0)(x_2 - x_1)a_2,\\
&\quad \vdots\\
x = x_n: &\quad P_n(x_n) = a_0 + (x_n - x_0)a_1 + (x_n - x_0)(x_n - x_1)a_2 + \cdots\\
&\qquad + (x_n - x_0)\ldots(x_n - x_{n-1})a_n.
\end{aligned}
$$

اگر $p_n(x)$ چند جمله‌ای درونیاب باشد، باید برای کلیه $n+1$ مقدار با جدول مطابقت داشته باشد.

$$p_n(x_i) = f_i \text{ برای } i = 0, 1, 2, \ldots, n$$

اگر بجای $p_n(x_i)$ در هر معادله f_i جایگزین شود، یک دستگاه که ضرایب آن ماتریس مثلثی است بدست می‌آید و هر a_i می‌تواند محاسبه شود.

از اولین معادله:

$$a_0 = f_0 = f[x_0] \qquad P_n(x_0) = f_0.$$

اگر

$$a_1 = f[x_0, x_1], \quad \text{پس}$$

$$P_n(x_1) = f_0 + (x_1 - x_0)\frac{f_1 - f_0}{x_1 - x_0} = f_1.$$

اگر

$$a_2 = f[x_0, x_1, x_2], \quad \text{پس}$$

$$
\begin{aligned}
P_n(x_2) &= f_0 + (x_2 - x_0)\frac{f_1 - f_0}{x_1 - x_0}\\
&\quad + (x_2 - x_0)(x_2 - x_1)\frac{(f_2 - f_1)/(x_2 - x_1) - (f_1 - f_0)/(x_1 - x_0)}{x_2 - x_0}\\
&= f_2.
\end{aligned}
$$

اگر $a_i = f[x_0, x_1, \ldots, x_n]$،

بطور مشابه هر $p_n(x_i)$ برابر f_i خواهد بود و چند جمله‌ای درونیاب که برازنده جدول تفاضل‌های محدود در $x = x_0, x_1, x_2, \ldots, x_n$ باشد برابر است با:

$$
\begin{aligned}
P_n(x) &= f_0^{[0]} + (x - x_0)f_0^{[1]} + (x - x_0)(x - x_1)f_0^{[2]}\\
&\quad + (x - x_0)(x - x_1)(x - x_2)f_0^{[3]} + \cdots\\
&\quad + (x - x_0)(x - x_1)\cdots(x - x_{n-1})f_0^{[n]}.
\end{aligned}
\tag{3.6}
$$

مثال ۳.۵ چند جمله‌ای درجه سومی بنویسید که برازنده داده‌های جدول ۲.۵ در کلیه نقاط از $x_0 = 3.2$ تا $x_3 = 4.8$ باشد.

$$P_3(x) = 22.0 + 8.400(x - 3.2) + 2.856(x - 3.2)(x - 2.7)$$
$$-0.528(x - 3.2)(x - 2.7)(x - 1.0).$$

اکنون چند جمله‌ای درجه چهارمی بنویسید که برازنده پنج نقطه جدول ۲.۵ می‌باشد. فقط کافی است یک جمله دیگر به $p_3(x)$ بیافزاییم:

$$P_4(x) = P_3(x) + 0.256(x - 3.2)(x - 2.7)(x - 1.0)(x - 4.8)$$

وقتی این روش برای درونیابی بکار می‌رود، مشاهده می‌شود که ضرب آشیانه‌ای می‌تواند مقدار عملیات محاسباتی را کاهش دهد، بعنوان مثال، برای $x=3$ داریم:

$$P_3(x) = \{[-0.528(3 - 1.0)+2.586](3-2.7)+8.400\}(3-3.2) + 22.0.$$

اگر در $x=3.0$ مقدار درونیابی برای چند جمله‌ای‌های درجه سوم در بخش‌های 2.5 و 3.5 را محاسبه کنیم، همان نتیجه $p_3(3.0)=20.21$ بدست می‌آید. تعجبی ندارد، زیرا که کلیه چند جمله‌ای‌های درجه سوم که از چهار نقطه معین عبور می‌کنند یکسان هستند. آنها ممکن است مختلف بنظر برسند، اما می‌توانند به یک شکل تبدیل شوند.

یک آلگوریتم برای ساختن یک جدول تفاضل محدود مطابق زیر است.

(x_i,f_i) و $i=0, 1, ..., n$ داده شده است.

DO FOR $i = 1$ *to* n:

  DO FOR $j = 0$ *to* $n-i$:

    محاسبه کنید $f_j^{[i]} = \left[f_{j+1}^{[i-1]} - f_j^{[i-1]}\right] / (x_{j+i} - x_j)$

    محاسبه و در ستون iام جدول وارد کنید

  ENDDO(j).

ENDDO (i).

مشاهده می‌شود که با پردازش موازی می‌توان تمام داده‌ها در ستون‌های متوالی را بطور همزمان محاسبه کرد.

تفاضل‌های محدود برای تابع $f(x)$ بصورت یک چند جمله‌ای

تفاضل‌های محدود برای تابع $f(x)=p_n(x)$ را بررسی می‌کنیم. فرض کنیم که $f(x)$ درجه سه باشد.

$$f(x) = 2x^3 - x^2 + x - 1$$

جدول تفاضل‌های محدود را می‌نویسیم:

x_i	$f(x_i)$	$f_i^{[1]}$	$f_i^{[2]}$	$f_i^{[3]}$	$f_i^{[4]}$	$f_i^{[5]}$
0.30	−0.7360	2.4800	3.0000	2.0000	0.0000	0.0000
1.00	1.0000	3.6800	3.6000	2.0000	0.0000	
0.70	−0.1040	2.2400	5.4000	2.0000		
0.60	−0.3280	8.7200	8.2000			
1.90	11.0080	21.0200				
2.10	15.2120					

مشاهده می‌شود که تفاضل‌های محدود ستون سوم همه یکسان هستند. (و تفاضل‌های محدود بعدی صفر هستند.) می‌توانیم از این واقعیت بهره‌مند شویم که برای چند جمله‌ای‌های درونیاب از ستون‌های بعدی استفاده نکنیم.

همچنین توجه به اینکه مشتق سوم یک چند جمله‌ای درجه سوم ثابت است بسیار مهم است، در این مثال ($p^{(3)}(x)=2*3!=12.$). رابطه بین تفاضل‌های محدود و مشتقات در فصل بعد در جزئیات توضیح داده می‌شود. در حال حاضر، فقط می‌توان بیان کرد که برای یک چند جمله‌ای درجه n، $p_n(x)$، a_n ضریب جمله‌ای است که بزرگترین توان را دارد، تفاضل‌های محدود مرتبه nام همیشه برابر a_n خواهد بود. چون مشتق nام این چند جمله‌ای برابر a_n $*n!$ می‌باشد، رابطه بین مشتقات و تفاضل‌های محدود $n!$ می‌باشد.

خطای درونیابی

جمله خطا برای یک چند جمله‌ای درونیابی که از جدول تفاضل محدود بدست می‌آید برابر خطای چند جمله‌ای لاگرانژ معادل آن است زیرا همانطور که قبلاً مشاهده شد، همه چند جمله‌ای‌های درجه n که از $n+1$ نقطه می‌گذرند، برابرند.

بدین معنی که جمله خطا برای چند جمله‌ای درجه nام $p_n(x)$ از معادله (5.5) برابر معادله (2.5) می‌باشد، که در اینجا تکرار می‌کنیم:

$$E(x) = (x - x_0)(x - x_1)\cdots(x - x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

این عبارت خطا هنوز مناسب استفاده نیست، زیرا مشتق f در آن مجهول است. بهر حال، اگر $f(x)$ تقریباً همان چند جمله‌ای درجه n باشد، (تفاضل‌های محدود مرتبه nام تقریباً ثابت هستند.) درونیابی با چند جمله‌ای درجه nام باید تقریباً دقیق باشد. بدین دلیل که مشتق $n+1$ام $f(x)$ تقریباً صفر است و خطای چند جمله‌ای درونیاب درجه nام بسیار کوچک خواهد بود.

اگر چند جمله‌ای درجه پایین‌تر بکار بریم، خطا بزرگتر خواهد شد. اگر $f(x)$ یک تابع شناخته شده باشد، می‌توانیم معادله 2.5 را جهت محدود کردن خطا استفاده کنیم.

مثال ۴.۵ جدول تفاضل محدود برای $f(x)=x^2e^{-x/2}$ برابر است با:

x_i	$f(x_i)$	$f_i^{[1]}$	$f_i^{[2]}$	$f_i^{[3]}$	$f_i^{[4]}$
1.10	0.6981	0.8593	−0.1755	0.0032	0.0027
2.00	1.4715	0.4381	−0.1631	0.0191	
3.50	2.1287	−0.0511	−0.0657		
5.00	2.0521	−0.2877			
7.10	1.4480				

خطای درونیابی برای $f(1.75)$ را با استفاده از چند جمله‌ای‌های درجه یک، دو و سه پیدا کنید.

نتایج در جدول ۳.۵ نشان داده شده‌اند و معادله (6.5) برای انجام درونیابی‌ها استفاده شده است. همانطور که انتظار داشتیم فرمول خطا مقادیر خطای واقعی را می‌دهد، در این حالت مشاهده می‌شود که با استفاده از چند جمله‌ای درجه سه دقت بهبود نمی‌یابد، زیرا که مقدار x بطور مناسب در مرکز مقادیر جدول قرار ندارد؛ همچنین مقدار مشتق نزولی نیست.

خطاهای درونیابی برای $f(1.75)$

درجه	مقدار درونیابی	مقدار واقعی	ماکزیمم	می‌نیمم	کناره بالا	کناره پایین
1	1.25668	0.01996	−0.3679	0.0594	0.0299	−0.00483
2	1.28520	−0.00856	−0.8661	0.1249	0.0059	−0.0408
3	1.28611	−0.00947	1.1398	−0.0359	0.0014	−0.0439

جدول (۳.۵)

تخمین خطا وقتی $f(x)$ ناشناخته است - قاعده جمله بعدی

تقریباً همیشه وقتی داده‌های تجربی را بکار می‌بریم، تابع مجهول است. هنوز راهی برای تخمین خطای درونیابی وجود دارد. زیرا که تفاضل محدود مرتبه nام تقریبی برای $\frac{f^{(n)}(x)}{n!}$ می‌باشد و در فصل بعد این را نشان خواهیم داد. بدین معنی که:

> $E_n(x)$ خطای درونیابی تقریباً مقدار جملهٔ بعدی است که باید اضافه شود!

این با ارزش‌ترین قاعده برای تخمین خطای درونیابی است و آن را قاعده جمله بعدی می‌نامیم.

در مثال قبلی داریم:

درجه	خطای واقعی	تخمین خطا از قاعده جمله بعدی
1	0.01996	0.02852
2	0.00856	0.00091
3	−0.00947	−0.00249

درونیابی نزدیک انتهای یک جدول

تا اینجا، پذیرفته‌ایم که مقادیر از بالا به پایین جدول اندیس گذاری می‌شود. این نشان می‌دهد در انتهای جدول، فرمولها برای ساختن چند جمله‌ای‌های درونیاب با استفاده از تفاضل‌های محدود جواب دقیق نمی‌دهد. بخاطر آورید، که ترتیب نقاط دلخواه است. ما می‌توانیم دقیقاً به همان روش از پایین جدول شروع کنیم و مقادیر را به بالا شماره گذاری کنیم، و تغییری در معادله *6.5* لازم نیست. بهر حال جدول تغییر نمی‌کند، به جز سمبل‌هایی که بکار می‌بریم. اکنون معادله *(6.5)* را با اندیس گذاری جدید استفاده می‌کنیم.

در جداول *4.5 (a)* و *(b)* دو طرح شماره گذاری مختلف را مقایسه می‌کنیم. مقادیر سطرهای جدول ۴.۵ *(b)* (قطر آن به سمت پایین است) دقیقاً همان اعداد جدول ۴.۵ *(a)* (قطر آن بسمت بالاست) می‌باشد.

جدول تفاضل – محدود ۴.۵ *(a)*

x_0	f_0	$f_0^{[1]}$	$f_0^{[2]}$	$f_0^{[3]}$	$f_0^{[4]}$
x_1	f_1	$f_1^{[1]}$	$f_1^{[2]}$	$f_1^{[3]}$	
x_2	f_2	$f_2^{[1]}$	$f_2^{[2]}$		
x_3	f_3	$f_3^{[1]}$			
x_4	f_4				

جدول تفاضل – محدود به سمت بالا اندیس گذاری شده‌است. ۴.۵ *(b)*

x_4	f_4				
x_3	f_3	$f_3^{[1]}$			
x_2	f_2	$f_2^{[1]}$	$f_2^{[2]}$		
x_1	f_1	$f_1^{[1]}$	$f_1^{[2]}$	$f_1^{[3]}$	
x_0	f_0	$f_0^{[1]}$	$f_0^{[2]}$	$f_0^{[3]}$	$f_0^{[4]}$

داده‌های متساوی الفاصله

اگر مقادیر تابع در فواصل متساوی الفاصله نسبت به متغیرهای مستقل داده شده باشد، مسألهٔ درونیابی از جدول داده‌ها بطور قابل ملاحظه‌ای ساده می‌شود.

ضروری است که ترتیب داده‌ها در جدول بر حسب مقادیر x نزولی باشد. برای ستون‌های مقادیر x جدول تفاضل‌های مقادیر تابع را می‌نویسیم. جدول ۵.۵ جدول تفاضل‌ها می‌باشد. هرستون سمت‌خ‌راست ستون $f(x)$ از تفاضل بین دو مقدار در ستون سمت چپ آن محاسبه می‌شود.

(بر تفاضل xها یعنی $x_{i+1} - x_i$ تقسیم نمی‌کنیم)

۵.۵ جدول تفاضل

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$
0.0	0.000				
		0.203			
0.2	0.203		0.017		
		0.220		0.024	
0.4	0.423		0.041		0.020
		0.261		0.044	
0.6	0.684		0.085		0.052
		0.346		0.096	
0.8	1.030		0.181		0.211
		0.527		0.307	
1.0	1.557		0.488		
		1.015			
1.2	2.572				

سمبل‌هایی که مقادیر را در جدول تفاضل‌ها نشان می‌دهند برای تعیین ضرایب چند جمله‌ای درونیاب مفید خواهند بود.

فرض می‌کنیم $h = x_{i+1} - x_i = \delta x$

با اندیس‌گذاری x_i ترتیب مقادیر x و $f(x)$ را نشان می‌دهیم، اولین تفاضل‌های تابع را تعریف می‌کنیم.

$$\Delta f_0 = f_1 - f_0, \quad \Delta f_1 = f_2 - f_1, \quad \Delta f_2 = f_3 - f_2, \quad \ldots, \quad \Delta f_i = f_{i+1} - f_i.$$

دومین و تفاضل‌های مرتبه بالاتر بطور مشابه تعریف می‌شوند. (تفاضل‌های پیشرو)[1]

$$\begin{aligned}
\Delta^2 f_1 &= \Delta(\Delta f_1) = \Delta(f_2 - f_1) = \Delta f_2 - \Delta f_1 = (f_3 - f_2) - (f_2 - f_1) \\
&= f_3 - 2f_2 + f_1, \\
\Delta^2 f_i &= f_{i+2} - 2f_{i+1} + f_i, \\
\Delta^3 f_1 &= \Delta(\Delta^2 f_1) = f_4 - 3f_3 + 3f_2 - f_1, \\
\Delta^3 f_i &= f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i, \\
&\vdots \qquad\qquad\qquad \vdots \\
\Delta^n f_i &= f_{i+n} - nf_{i+n-1} + \frac{n(n-1)}{2!} f_{i+n-2} - \frac{n(n-1)(n-2)}{3!} f_{i+n-3} + \cdots .
\end{aligned} \tag{7.5}$$

در معادلات (7.5) و سرتاسر این فصل مقدار تابع بصورت $f_3 = f(x_3)$ نشان داده می‌شود. الگوی ضرایب در معادلات (7.5) مشابه ضرایب بسط دو جمله‌ای $(x+1)^n$ می‌باشد. با این واقعیت می‌توانیم هر چه بیشتر قابل استفاده بودن روش‌های سمبلیک را اثبات کنیم.

معادلهٔ (7.5) نشان می‌دهد که هر تفاضل می‌تواند مستقیماً از مقادیر تابع بدست آید. جدول ۶.۵ جدول تفاضل‌ها را با استفاده نمایش سمبلیک نشان می‌دهد.

۱- **forward differences**

۶.۵ جدول تفاضل‌ها

s	x	$f(x)$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
-2	x_{-2}	f_{-2}				
			Δf_{-2}			
-1	x_{-1}	f_{-1}		$\Delta^2 f_{-2}$		
			Δf_{-1}		$\Delta^3 f_{-2}$	
0	x_0	f_0		$\Delta^2 f_{-1}$		$\Delta^4 f_{-2}$
			Δf_0		$\Delta^3 f_{-1}$	
1	x_1	f_1		$\Delta^2 f_0$		$\Delta^4 f_{-1}$
			Δf_1		$\Delta^3 f_0$	
2	x_2	f_2		$\Delta^2 f_1$		$\Delta^4 f_0$
			Δf_2		$\Delta^3 f_1$	
3	x_3	f_3		$\Delta^2 f_2$		
			Δf_3			
4	x_4	f_4				

ممکن است بطور معمول اولین مقدار x در جدول را، x_0 بنامیم، در اینصورت دسترسی مکرر به یک قسمت از داده‌ها باعث می‌شود اولین x معنی خود را از دست بدهد. در اینصورت بطور دلخواه مبداء را برای اندیس‌گذاری انتخاب می‌کنیم، زیرا که با استفاده از اندیس منفی به مقادیر x قبل از x_0 مراجعه می‌کنیم. وقتی $f(x)$ برای مجموعه داده‌ها مانند یک چند جمله‌ای رفتار می‌کند، جدول تفاضل‌ها خواص معینی دارد. در جدول ۷.۵ یک تابع روی میدان $x=1$ تا $x=6$ بصورت جدول داده شده است. و مانند x^3 رفتار می‌کند. ما فقط مقادیر جدول را برای $f(x)$ داریم.

جدول ۷.۵ جدول تفاضل برای تابع $f(x)$

x	$f(x)$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
0	0				
		1			
1	1		6		
		7		6	
2	8		12		0
		19		6	
3	27		18		0
		37		6	
4	64		24		0
		61		6	
5	125		30		
		91			
6	216				

در اینجا از تفاضل‌های پیشرو استفاده شده است. بعضی متن‌ها تفاضل‌های پس رو *backward differences* ($\nabla f_i = f_i - f_{i-1}$) و تفاضل مرکزی ($\delta f_i$) را تعریف می‌کنند. در این قسمت فقط تفاضل‌های پیشرو را استفاده می‌کنیم.

مشاهده می‌شود که سومین تفاضل‌ها ثابت هستند و در نتیجه چهارمین تفاضل و بالاتر صفر هستند و واقعیتی را نشان می‌دهد که تفاضل‌های مرتبه n از هر چند جمله‌ای درجه n ام ثابت است.

برای آنکه اثبات کنیم تفاضل‌های مرتبه nام ثابت هستند، ابتداء تفاضل‌های محدود ax^n را بدست می‌آوریم.

$$\begin{aligned}\Delta(ax^n) &= a(x+h)^n - ax^n \\ &= (ax^n + anx^{n-1}h + \cdots + ah^n) - ax^n\end{aligned}$$

$$= (anh)x^{n-1} +$$
$$\Delta(anh\,x^{n-1}) = an(n-1)h^2x^{n-2} + \quad \text{جملات با درجه کمتر}$$

سپس داریم:

$$\begin{aligned}
\Delta P_n(x) &= \Delta(a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n) \\
&= a_0nhx^{n-1} + \quad \text{جملات با درجه کمتر} \\
\Delta^2P_n(x) &= a_0n(n-1)h^2x^{n-2} + \quad \text{جملات با درجه کمتر} \\
&\vdots \\
\Delta^nP_n(x) &= a_0n(n-1)(n-2)\cdots(1)h^nx^{n-n} = a_0\,n!h^n. \qquad (8.5)
\end{aligned}$$

این نه فقط نشان می‌دهد که تفاضل‌های nام ثابت هستند، بلکه مقدار آن $a_0n!h^n$ می‌باشد. به تشابه بین تفاضل nام و مشق nام $P_n(x)$ توجه کنید.

$$p_n^{(n)}(x) = a_0n!$$

a_0 ضریب x^n می‌باشد.

تفاضل‌های تابع بجای تفاضل‌های محدود

جدول تفاضل‌های تابع که مورد بحث است وابستگی نزدیکی به جدول تفاضل‌های محدود بخش قبل دارد. بجز آنکه تفاضل‌های تابع و تفاضل‌های محدود مقادیر x در قسمت اخیر، در این دو جدول دارای مقادیر x متساوی الفاصله هستند.

برای روشن کردن موضوع، دو جدول (۸.۵) (a) و (۸.۵) (b) برای حالت ساده $f(x)=2x^3$ با $h=0.5$ مقایسه شده‌اند.

جدول ۸.۵ (a) — جدول تفاضل‌های تابع $f(x)=2x^3$ و $h=0.5$

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$	$\Delta^5 f_i$
0.00	0.00	0.25	1.50	1.50	0.00	0.00
0.50	0.25	1.75	3.00	1.50	0.00	0.00
1.00	2.00	4.75	4.50	1.50	0.00	
1.50	6.75	9.25	6.00	1.50		
2.00	16.00	15.25	7.50			
2.50	31.25	22.75				
3.00	54.00					

جدول تفاضل‌های محدود برای $f(x)=2x^3$ و $h=0.5$ جدول ۸.۵ (b)

x_i	f_i	$f_i^{[1]}$	$f_i^{[2]}$	$f_i^{[3]}$	$f_i^{[4]}$	$f_i^{[5]}$
0.00	0.00	0.50	3.00	2.00	0.00	0.00
0.50	0.25	3.50	6.00	2.00	0.00	0.00
1.00	2.00	9.50	9.00	2.00	0.00	
1.50	6.75	18.50	12.00	2.00		
2.00	16.00	30.50	15.00			
2.50	31.25	45.50				
3.00	54.00					

هم چنانکه انتظار داشتیم، ستون سوم تفاضل‌ها در هر دو جدول ثابت هستند، برای تفاضل‌های محدود برابر 2، که ضریب x^3 می‌باشد. برای جدول تفاضل تابع، این عدد در $(3!)(h^3)$ ضرب می‌شود $2*6*0.5^3=1.5$.

برای اولین ستون تفاضل‌ها، تفاضل‌های محدود برابر تفاضل‌های تابع تقسیم بر h (یا 0.5) هستند. دومین ستون تفاضل‌های محدود مساوی دومین تفاضل‌های تابع تقسیم بر $(h)(2h)$ (یا 0.5) می‌باشد. سومین ستون تفاضل محدود مساوی با سومین تفاضل‌های تابع تقسیم بر $(h)(2h)(3h)$ (یا 0.75) می‌باشد، که در حالت کلی نتیجه می‌شود:

$$f_i^{[n]}=\frac{\Delta^n f_i}{n!\, h^n}$$

اگر مقادیر متساوی الفاصله نباشند، مقایسه غیر ممکن است زیرا که جدول تفاضل‌های تابع تعریف نشده است.

اختلاف بین این دو نوع جدول دارای تأثیر بزرگی در رابطه بین تفاضل‌ها و مشتقات می‌باشد، موضوعی که در فصل بعد بحث و بررسی می‌گردد.

چند جمله‌های درونیاب بر مبنای جدول‌های تفاضلی

اگر تابعی که به صورت جدول می‌باشد رفتاری شبیه یک چند جمله‌ای داشته باشد (مشاهده می‌کنیم که تفاضل‌های مرتبه nام آن ثابت یا تقریباً ثابت هستند) می‌توانیم آنرا با یک چند جمله‌ای نظیر کنیم. جدول تفاضل‌های تابع طریق دیگری برای نوشتن چند جمله‌ای درجه n امی است که از $n+1$ نقطه (x_i,f_i), $i = 0, 1, 2, 3, \ldots n$ می‌گذرد. توجه شود که این یک چند جمله‌ای یکّه است.

یکی از راه‌های ساده نوشتن یک چند جمله‌ای که از یک گروه نقاط هم فاصله بگذرد، چند جمله‌ای‌نیوتن-گریگوری[1] پیشرو می‌باشد؛ و آنرا بر حسب اندیس s می‌نویسیم.

۱- Newton- Gregory

$$P_n(x_s) = f_0 + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_0 + \cdots \quad (9.5)$$
$$= f_0 + \binom{s}{1}\Delta f_0 + \binom{s}{2}\Delta^2 f_0 + \binom{s}{3}\Delta^3 f_0 + \binom{s}{4}\Delta^4 f_0 + \cdots$$

با مراجعه به جدول (۶.۵) مشاهده می‌شود که $P_n(x)$، در تمام نقاط $(x_i, f_i) i = 0, 1, 2, 3, \ldots n$ با جدول مطابقت دارد.

اگر $s = 0$

$$P_n(x_0) = f_0$$

اگر $s = 1$

$$P_n(x_1) = f_0 + \Delta f_0 = f_0 + f_1 - f_0 = f_1$$

اگر $s = 2$

$$P_n(x_2) = f_0 + 2\Delta f_0 + 2\Delta^2 f_0 = f_2$$

بطور مشابه $P_n(x)$ در کلیه نقاط مطابقت دارد، در معادله (9.5) تمام تفاضل‌ها در جدول ۶.۵ روی خط قطری به سمت پایین و شروع در f_0 قرار دارند.

قبلاً مشاهده کردیم که اگر در میدان از x_0 تا x_n، $p_n(x)$ و $f(x)$ دارای مقادیر مساوی در جدول مقادیر باشند، می‌پذیریم در نقاط درونی نیز تقریباً دارای مقادیر یکسان هستند. این مبنایی برای کاربرد $p_n(x)$ به عنوان چند جمله‌ای درونیاب می‌باشد. مجدداً تأکید می‌کنیم که $f(x)$ و $p_n(x)$ بطور کلی یکسان نیستند.

از اینرو در تخمین از چنین درونیابی وجود خطا را انتظار داریم. با فرض اینکه s مقداری غیر صحیح باشد از چند جمله‌ای معادله (9.5) بعنوان چند جمله‌ای درون‌یاب استفاده می‌کنیم. این بسط تعریف s می‌باشد، به طوری که برای هر مقدار x

$$s = \frac{x - x_0}{h}$$

مثال ۵.۵ چند جمله‌ای درجه 3 نیوتن - گریگوری را بنویسید که برازنده جدول ۵.۵ برای چهار نقطه $x = 1.0$ تا $x = 0.4$ باشد و از آن برای درونیابی $f(0.73)$ استفاده کنید.

برای ساختن چند جمله‌ای با $x_0 = 0.4$ خواهیم داشت.

$$f_0 = 0.423\ ,\ \Delta f_0 = 0.261,\ \Delta^2 f_0 = 0.085,\ \Delta^3 f_0 = 0.096.$$

s را محاسبه می‌کنیم.

$$s = \frac{x - x_0}{h} = \frac{0.73 - 0.4}{0.2} = 1.65$$

این مقادیر را در معادله (9.5) تا $\Delta^3 f_0$ به کار می‌بریم تا چند جمله‌ای درجه 3 به دست آید.

$$f(0.73) = 0.423 + (1.65)(0.261) + \frac{(1.65)(0.65)}{2}(0.085) \qquad (10.5)$$
$$+ \frac{(1.65)(0.65)(-0.35)}{6}(0.096)$$
$$= 0.423 + 0.4306 + 0.0456 - 0.0060$$
$$= 0.893.$$

تابع جدول شده در جدول ۵.۵ برابر $tan(x)$ می‌باشد، که مقدار واقعی در $x=0.73$ برابر 0.895 می‌باشد. چون تفاضل‌های سوم مقادیر ثابت نزدیک بهم نیستند، باید انتظار خطا داشته باشیم و چند جمله‌ای درجه سوم نمایش کامل تابع نیست، هر چند، این چند جمله‌ای تخمین نسبتاً خوبی است و مسلماً از درون‌یابی خطی که جواب 0.911 بدست می‌آید، بهتر است.

اگرچه تفاضل‌های چهارم نیز ثابت نیستند، امّا اگر $f(x)$ را با چند جمله‌ای درجه چهار تخمین بزنیم امید بهبود داریم. می‌توانیم فقط یک جمله به (10.5) اضافه کنیم.

$$\binom{s}{4}\Delta^4 f_0 = \frac{(1.65)(0.65)(-0.35)(-1.35)}{4!}(0.211) = 0.0045,$$
$$f(0.73) = 0.893 + 0.0045 = 0.898.$$

معمولاً چند جمله‌ای درجه بالاتر بهتر است، اما در این مثال، افزودن یک جمله تخمین را بهبود نبخشیده است بلکه اندکی آنرا بدتر کرده است. این مربوط به خطاهای گرد کردن در مقادیر اصلی این مثال است.

مجدداً یادآوری می‌کنیم که جمله خطا برابر خطای حاصل از روش لاگرانژ است. همچنین مشاهده می‌شود مقادیری که استفاده می‌کنیم باید طوری باشد که هرچه ممکن است مقدار x در درون مقادیری باشد که در ساختن چند جمله‌ای درونیاب استفاده می‌شوند.

همچنان که بیان شد، یک راه تخمین خطای مقدار درونیابی قاعده جمله بعدی می‌باشد. می‌توانیم با نگاه به معادله (10.5) این قاعده را آزمایش می‌کنیم، آخرین جمله افزوده شده برابر -0.006 است؛ و خطای چند جمله‌ای درجه دوم برابر 0.0456 می‌باشد.

۵.۵ درونیابی با اسپلاین درجه سه[۱]

راه دیگری برای برازش یک چند جمله‌ای به یک مجموعه از داده‌ها وجود دارد. اسپلاین درجه سه یک «خم هموار»[۲] را به نقاط داده شده برازش می‌کند.

این فکر از یک وسیله طراحی اسپلاین گرفته شده است. این دستگاه یک میله قابل انعطاف است و خم می‌شود تا بر نقاط منطبق شود.

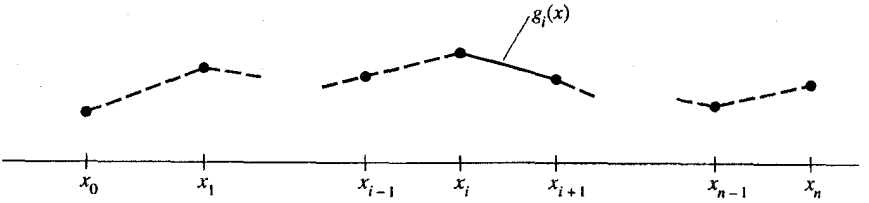
خم اسپلاین[۳] می‌تواند از هر درجه‌ای باشد. فرض می‌کنیم مجموعه $(n+1)$ نقطه داده

۱- Cubic Spline
۲- Smooth Curve
۳- Spline Curve

شده‌اند.

$$(x_i, y_i),\ i = 0, 1, 2, 3, \ldots n$$

بطورکلی، یک مجموعه چند جمله‌ای درجه n بین هر دو نقطه متوالی $g_i(x)$ از x_i به x_{i+1} برازش می‌کنیم. اگر درجه اسپلاین یک باشد. (یک خط مستقیم بین نقاط رسم می‌شود) خم مطابق شکل زیر است، مشکل «اسپلاین خطی» ناپیوستگی ضریب زاویه مشتق در نقاط داده شده است. یک چند جمله‌ای درجه n که از کلیه نقاط می‌گذرد دارای ضریب زاویه پیوسته در نقاط داده شده است. اما چند جمله‌ای درونیاب و درجات بالاتر دارای مشکل دیگری می‌باشد.



یک مثال از این وضعیت نامناسب را توضیح می‌دهیم:

فرض کنیم $f(x)=cos^{10}(x)$ روی فاصلهٔ $[-2, 2]$ باشد چند جمله‌ای‌های درجه $8,6,4,2$ را در نقاط متساوی الفاصله به آن برازش می‌کنیم. شکل ۱.۵ نشان می‌دهد که هیچکدام از چند جمله‌ای‌ها برازنده خوبی به $f(x)$ نیستند. مشکل این است که $f(x)$ تقریباً کشیدگی دارد به جزء برای برآمدگی بین $x=+1,\ x=-1$. در قسمت‌های هموار ضروری است که $p_n(x)$ صفرهایی خارج فاصله $[-1,+1]$ داشته باشد تا نوسان بوجود آید.

یک راه علاج مشکل، برازش چند جمله‌ای‌های مختلف در زیر فاصله‌های $f(x)$ است. برازش چند جمله‌ای‌های درجه اول معادل اسپلاین خطی می‌باشد.

شکل ۲.۵ یک برازش بهتر را با استفاده از چند جمله‌ای درجه دوم بین $x=0.65$ و $x=-0.65$ با $p(x)=0$ خارج این دامنه نشان می‌دهد. اگرچه این برازش بهتر است (و می‌توانیم آنرا بهبود بخشیم)، لیکن ناپیوستگی هایی در ضریب زاویه محل تلاقی چند جمله‌ای‌ها وجود دارد و $f(x)$ در این نقاط پیوسته است.

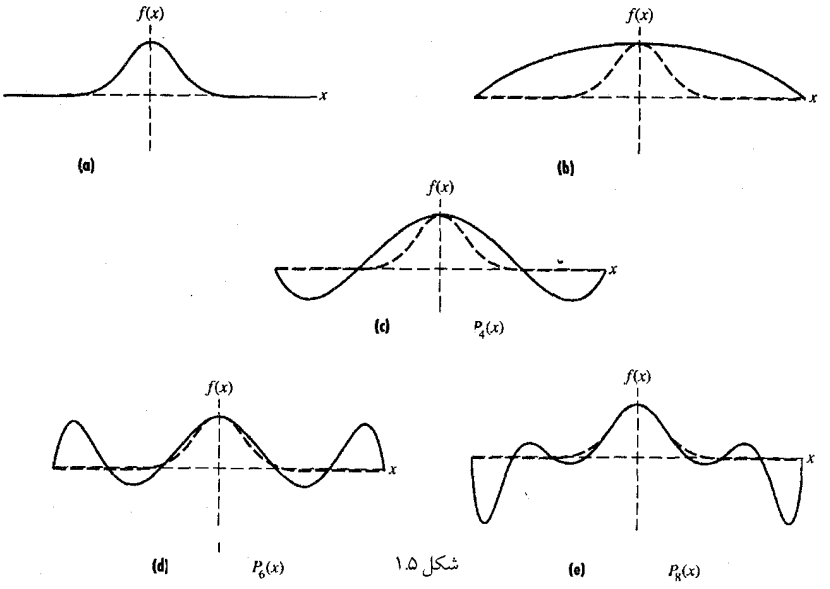
بر طبق قوانین خمش تیرها خم اسپلاین رسم می‌گردد بطوری که ضریب زاویه و انحناء پیوسته می‌باشد. خم اسپلاین در ریاضیات، چند جمله‌ای‌های درجه سوم (و بیشتر) بر طبق این شرایط می‌باشد. هرچند خم‌های اسپلاین می‌توانند از هر درجه باشند، اسپلاین‌های مرتبه سوم از همه بیشتر مورد استفاده قرار می‌گیرند و فقط این چند جمله‌ایها شرح داده خواهند شد.

ما اسپلاین‌های درجه سوم را بطور متوالی و در فواصل متوالی برازش خواهیم کرد.

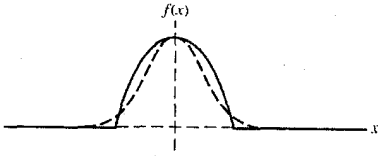
این چند جمله‌ای‌ها دارای ضریب زاویه و انحناء مساوی در نقاطی هستند که آنها را بهم متصل می‌کنند. نیازی نیست که فواصل دارای عرض یکسانی باشند.

در نقاط انتهایی مجموعه داده‌ها که خم‌های اسپلاین به $f(x)$ برازش می‌شوند، نقطه اتصال

به چند جمله دیگری وجود ندارد.



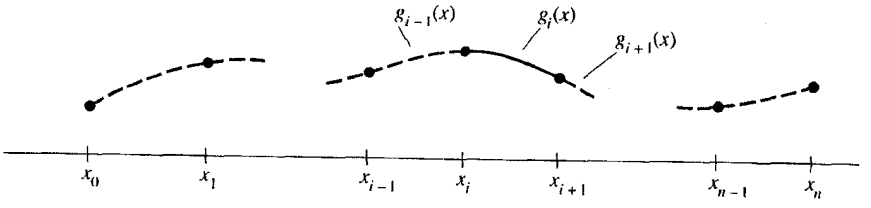
شکل ۱.۵



شکل ۲.۵

و بدین معنی است که ضریب زاویه و انحناء مقید نیست. این نقاط انتهایی بعداً در بسط موضوع مطالعه می‌گردند.

معادله و یک چند جمله‌ای درجه سه را در فاصله iام می‌نویسیم، که بین نقاط (x_i, y_i) و (x_{i+1}, y_{i+1}) قرار دارد. نتیجه شبیه قسمت زیر است:



و دارای معادله‌ای است برابر با:

$$g_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i. \qquad (11.5)$$

تابع اسپلاین درجه سه مورد نظر دارای شرایط زیر است:

$g(x) = g_i(x)$ روی فاصله (x_i, x_{i+1}) و $i = 0, 1, 2, \ldots n-1$

$$g_i(x_i) = y_i, \quad i = 0, 1, \ldots, n-1 \qquad g_{n-1}(x_n) = y_n; \tag{12.5)(a}$$

$$g_i(x_{i+1}) = g_{i+1}(x_{i+1}), \quad i = 0, 1, \ldots, n-2; \tag{12.5)(b}$$

$$g_i'(x_{i+1}) = g_{i+1}'(x_{i+1}), \quad i = 0, 1, \ldots, n-2; \tag{12.5)(c}$$

$$g_i''(x_{i+1}) = g_{i+1}''(x_{i+1}), \quad i = 0, 1, \ldots, n-2. \tag{12.5)(d}$$

[معادلات (12.5) برازش اسپلاین مرتبه سه به نقاط داده شده را نشان می‌دهند. (12.5)(a)، (12.5)(b) پیوستگی، و پیوستگی شیب و انحناء را (12.5)(c) و (12.5)(d) نشان می‌دهند.]

اگر $n+1$ نقطه وجود داشته باشد، تعداد فواصل و تعداد توابع $g_i(x)$ برابر n می‌باشد. در اینصورت $4n$ مجهول وجود دارد که $\{a_i, b_i, c_i, d_i\}$ بازای $i = 0, 1, 2, \ldots n-1$ می‌باشد.

معادلات (12.5)(a) فوراً نتیجه می‌دهد:

$$d_i = y_i, \; i = 0, 1, 2, \ldots, n-1 \tag{13.5}$$

معادله (12.5)(b) می‌دهد:

$$\begin{aligned} y_{i+1} &= a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + y_i \\ &= a_i h_i^3 + b_i h_i^2 + c_i h_i + y_i, \qquad i = 0, 1, \ldots, n-1. \end{aligned} \tag{14.5}$$

بازاء $h_i = x_{i+1} - x_i$ در ارتباط شیب و انحناء اتصال اسپلاین‌ها، از (11.5) مشتق می‌گیریم:

$$g'_i(x) = 3a_i h_i^2 + 2b_i h_i + c_i, \tag{15.5}$$

$$g''_i(x) = 6a_i h_i + 2b_i, \qquad i = 0, 1, \ldots, n-1. \tag{16.5}$$

اگر معادلات را بر حسب مشتقات مرتبه دوم بنویسیم، روش ریاضی آسان می‌شود - یعنی اگر، فرض کنیم

$$S_i = g''_i(x_i) \quad , \quad i = 0, 1, 2, \ldots, n-1$$

و

$$S_{i+1} = g''_i(x_{i+1})$$

از معادله (16.5) داریم.

$$\begin{aligned} S_i &= 6a_i(x_i - x_i) + 2b_i \\ &= 2b_i; \\ S_{i+1} &= 6a_i(x_{i+1} - x_i) + 2b_i \\ &= 6a_i h_i + 2b_i, \end{aligned}$$

از اینرو

$$b_i = \frac{S_i}{2}, \tag{17.5}$$

$$a_i = \frac{S_{i+1} - S_i}{6h_i}. \tag{18.5}$$

روابط (13.5)، (17.5) و (18.5) را برای d_i، b_i، a_i در رابطه (11.5) جایگزین می‌کنیم و

سپس آن را برای c_i حل می‌کنیم.

$$y_{i+1} = \left[\frac{S_{i+1} - S_i}{6h_i}\right] h_i^3 + \frac{S_i}{2} h_i^2 + c_i h_i + y_i;$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6}.$$

حال به دنبال شرایطی هستیم که شیب دو منحنی درجه سومی که در نقطه (x_i, y_i) با هم برخورد می‌کنند یکی باشند برای برابری در فاصلهٔ iام رابطه $(12.5)(c)$ به ازای $(x = x_i)$ مطابق زیر است:

$$y'_i = 3a_i(x_i - x_i)^2 + 2b_i(x_i - x_i) + c_i = c_i.$$

در فاصلهٔ قبلی از x_{i-1} تا x_i، شیب در انتهای سمت راست برابر خواهد بود با:

$$y'_i = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1}$$

$$= 3a_{i-1} h_{i-1}^2 + 2b_{i-1} h_{i-1} + c_{i-1},$$

با برابر قرار دادن و جایگذاری d, c, b, a با روابط آنها بر حسب S و y، به دست می‌آوریم:

$$y'_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6}$$

$$= 3\left(\frac{S_i - S_{i-1}}{6h_{i-1}}\right) h_{i-1}^2 + 2\left(\frac{S_{i-1}}{2}\right) h_{i-1} + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{2h_{i-1} S_{i-1} + h_{i-1} S_i}{6}.$$

با ساده کردن معادله داریم:

$$h_{i-1} S_{i-1} + (2h_{i-1} + 2h_i) S_i + h_i S_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)$$

$$= 6(f[x_i, x_{i+1}] - f[x_{i-1}, x_i]). \quad (3.19)^{(19.5)}$$

معادلهٔ (19.5) برای هر یک از نقاط داخلی از $i = 1$ تا $i = n-1$ بکار می‌رود و مجموعاً $n + 1$ نقطه خواهیم داشت.

به این صورت $n - 1$ معادله وابسته به $n + 1$ مقدار S_i می‌باشد.

هنگامی که موقعیت‌های مربوط به دو انتهای فاصله کل خم را مشخص کنیم دو معادله اضافی شامل S_0, S_n بدست می‌آیند.

برای بعضی موارد، این نقاط انتهایی اختیاری هستند.

و چهار پیشنهاد زیر غالباً مورد استفاده قرار می‌گیرند:*

---

*- شرط پنجم: در بعضی مواقع یک تابع متناوب است و داده‌ها یک تناوب کامل دارند. در این مورد: $S_0 = S_n$ می‌باشد و نیز شیبها در نقاط ابتدا و انتها یکی هستند.

۱ - قرار دهید: $S_0 = 0$ و $S_n = 0$، این انتخاب معادل آن است که فرض کنیم منحنی‌های درجه سوم انتهایی در حد نهایی خود به صورت خطی میل می‌کنند.

این شرایط، اسپلاین طبیعی نامیده می‌شود، و دقیقاً با وسیله رسم مطابقت دارد. این فن بطور فراوانی مورد استفاده قرار می‌گیرد.

۲ - در شرط دیگر شیب‌ها در هر یک از دو انتهای فاصله مقدار معینی قرار می‌دهیم. وقتی که اطلاعات کافی نباشد، شیب ممکن است در این نقاط تخمین زده شود. اگر $f'(x_0) = A$ و $f'(x_0) = B$ باشد، روابط زیر را بکار می‌بریم. [از تفاضل‌های محدود استفاده شده است]

در انتهای نقطه چپ $2h_0S_0 + h_1 S_1 = 6(f[x_0 , x_1] - A$

$h_{n-1}S_{n-1} + 2h_n S_n = 6(B - f[x_{n-1}, x_n])$.

۳ - فرض می‌کنیم $S_0 = S_1$ و $S_{n-1} = S_n$. این انتخاب معادل آن است که بپذیریم منحنی‌های درجه سوم انتهایی در حد نهایی خود به سمت سهمی میل می‌کنند.

۴ - S_0 را برونیابی خطی از S_1، S_2، در نظر می‌گیریم. S_n را یک برونیابی خطی از S_{n-1}، S_{n-2} فرض می‌کنیم. برای سری داده‌ها که تنها با یک منحنی درجه سه برازش شده‌اند. منحنی اسپلاین درجه سوم آنها همین منحنی خواهد بود.

سایر حالات این خصوصیت را ندارند.

روابط برای نقاط انتهای حالت چهارم مطابق زیر می‌باشند:

در نقطه انتهایی چپ:

$$\frac{S_1 - S_0}{h_0} = \frac{S_2 - S_1}{h_1}, \quad S_0 = \frac{(h_0 + h_1)S_1 - h_0 S_2}{h_1}$$

در نقطه انتهایی راست:

$$\frac{S_n - S_{n-1}}{h_{n-1}} = \frac{S_{n-1} - S_{n-2}}{h_{n-2}},$$

$$S_n = \frac{(h_{n-2} + h_{n-1})S_{n-1} - h_{n-1} S_{n-2}}{h_{n-2}}. \qquad (20.5)$$

رابطهٔ ۱ - در جایی که $S_0 = 0$، $S_n = 0$ باشد. اسپلاین طبیعی نامیده می‌شود. غالباً احساس می‌شود که این روش موجب می‌شود منحنی در نقاط انتهایی بیش از حد کشیده شود، با این وجود این روش اغلب به کار برده می‌شود.

رابطهٔ ۴ - این روش غالباً ما را دچار زحمت می‌سازد به این صورت که در فاصلهٔ دو انتها، خمیدگی منحنی را بیش از حد می‌نماید.

شاید بهترین حالت برای نقاط انتهایی همان حالت شمارهٔ دو باشد، به شرط آنکه ارائه

تخمین‌های منطقی برای مشتقات امکان‌پذیر باشد.

اگر معادله $S_1, S_2, \ldots, S_{n-1}$ (معادله 19.5) را به شکل ماتریسی بنویسیم خواهیم داشت:

$$\begin{bmatrix} h_0 & 2(h_0+h_1) & h_1 & & & & \\ & h_1 & 2(h_1+h_2) & h_2 & & & \\ & & h_2 & 2(h_2+h_3) & h_3 & & \\ & & & \ddots & & & \\ & & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \end{bmatrix} \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_{n-1} \\ S_n \end{bmatrix}$$

$$= 6 \begin{bmatrix} f[x_1,x_2] - f[x_0,x_1] \\ f[x_2,x_3] - f[x_1,x_2] \\ f[x_3,x_4] - f[x_2,x_3] \\ \vdots \\ f[x_{n-1},x_n] - f[x_{n-2},x_{n-1}] \end{bmatrix}.$$

در دستگاه ماتریسی بالا فقط $n-1$ معادله ولی $n+1$ مجهول وجود دارد. ما می‌توانیم دو مجهول S_0 و S_n را با استفاده از روابطی که متناظر باشرایط انتهائی می‌باشد حذف کنیم.

علاوه بر این ماتریس مذکور همیشه حتی در حالت چهارم نیز ماتریس سه قطری می‌باشد بنابراین دستگاه سریع‌تر حل می‌شود و ذخیره نمودن آن نیز اقتصادی می‌باشد.

یک برنامه در انتهای این فصل آورده شده است که این ماتریس سه قطری را می‌سازد و آن را با بردار سمت راست تلفیق می‌کند سپس آن را برای مقادیر s در هر فاصله حل می‌نماید.

بخاطر آورید که s مقدار مشتق دوم چند جمله‌ای درجه سه در هر فاصله می‌باشد، بنابراین برای بدست آوردن منحنی درجه سوم واقعی که خم برازنده را بدهد به کار بیشتری نیازمندیم.

شرایط انتهائی برای هر یک از حالات ماتریس ضرایب عبارتند از:

شرط اول $S_0 = 0 \ , \ S_n = 0$

$$\begin{bmatrix} 2(h_0+h_1) & h_1 & & & \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & h_2 & 2(h_2+h_3) & h_3 & \\ & & \ddots & & \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{bmatrix}$$

شرط دوم: $f'(x_0) = A \quad , \quad f'(x_n) = B$

$$\begin{bmatrix} 2h_0 & h_1 & & & \\ h_0 & 2(h_0+h_1) & h_1 & & \\ & h_1 & 2(h_1+h_2) & h_2 & \\ & & \ddots & & \\ & & & h_{n-2} & 2h_{n-1} \end{bmatrix}$$

شرط سوم: $S_0 = S_1$ و $S_n = S_{n-1}$

$$\begin{bmatrix} (3h_0 + 2h_1) & h_1 & & & \\ h_1 & 2(h_1 + h_2) & h_2 & & \\ & h_2 & 2(h_2 + h_3) & h_3 & \\ & & \ddots & \ddots & \\ & & & h_{n-2} & (2h_{n-2} + 3h_{n-1}) \end{bmatrix}$$

شرط چهارم: S_0 و S_n برونیابی خطی هستند:

$$\begin{bmatrix} \frac{(h_0 + h_1)(h_0 + 2h_1)}{h_1} & \frac{h_1^2 - h_0^2}{h_1} & & & \\ h_1 & 2(h_1 + h_2) & h_2 & & \\ & h_2 & 2(h_2 + h_3) & h_3 & \\ & & \ddots & \ddots & \\ & & \frac{h_{n-2}^2 - h_{n-1}^2}{h_{n-2}} & \frac{(h_{n-1} + h_{n-2})(h_{n-1} + 2h_{n-2})}{h_{n-2}} & \end{bmatrix}.$$

در شرط چهارم پس از حل کردن یک دستگاه از معادلات باید مقادیر S_0 و S_n را با استفاده از معادلهٔ *(20.5)* محاسبه نمائیم. برای شرط‌های *(3, 2, 1)* هیچ محاسبه‌ای لازم نیست. برای هر یک از سه حالت اول بردار سمت راست یکسان است و در معادله *(19.5)* داده شده است.

اگر داده‌ها متساوی الفاصله باشند ماتریس‌ها بصورت ساده در می‌آیند.

پس از آن‌که مقادیر s_i محاسبه شدند، ضرایب a_i، b_i، c_i، d_i برای منحنی درجه سه در هر فاصله بدست خواهند آمد. از این مقادیر می‌توانیم نقاطی را که روی منحنی درونیاب قرار می‌گیرند محاسبه کنیم.

$$a_i = \frac{S_{i+1} - S_i}{6h_i};$$

$$b_i = \frac{S_i}{2};$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6};$$

$$d_i = y_i.$$

مثال ۶.۵ داده‌های جدول (۹.۵) را بایک منحنی اسپلاین درجه سوم طبیعی برازش دهید. و مقدار اسپلاین $g(0.66)$ و $g(1.75)$ را حساب کنید.

$[f(x)=2e^x - x^2]$ مشاهده می‌شود که $h_0=1$، $h_1 = 0.5$، $h_2 = 0.75$ تفاضل‌های محدودی که برای بدست آوردن طرف راست معادلات می‌توانیم استفاده کنیم برابر است با:

$f[0,1] = 2.4366$، $f[1, 1.5] = 4.5536$ و $f[1.5,2.25]= 9.5995$

x	$f(x)$
0.0	2.0000
1.0	4.4366
1.5	6.7134
2.25	13.9130

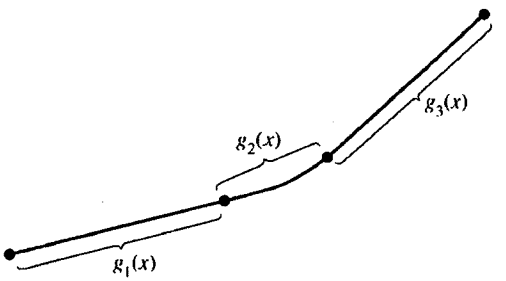
برای اسپلاین درجه سه طبیعی، شرط *1* را بکار می‌بریم.

$$\begin{bmatrix} 3.0 & 0.5 \\ 0.5 & 2.5 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} 12.7020 \\ 30.2754 \end{bmatrix},$$

بعد از حل دستگاه داریم $S_2 = 11.6518, S_1 = 2.2920$ (البته $S_0 = S_3 = 0$) این S ها را بکار می‌بریم و ضرایب اسپلاین‌های درجه سوم مورد نظر را محاسبه می‌کنیم.

i	فاصله	$g_i(x)$
0	[0.0, 1.0]	$0.3820(x-0)^3 + 0(x-0)^2 + 2.0546(x-0) + 2.0000$
1	[1.0, 1.5]	$3.1199(x-1)^3 + 1.146(x-1)^2 + 3.2005(x-1) + 4.4366$
2	[1.5, 2.25]	$-2.5893(x-1.5)^3 + 5.8259(x-1.5)^2 + 6.6866(x-1.5) + 6.7134$

شکل ۳.۵ خم اسپلاین درجه سوم را نشان میدهد. (تحقیق کنید که این معادلات در تمام شرایطی که برای خم‌های اسپلاین درجه سوم داده شده بود صدق می‌کنند.)



شکل (۳.۵)

$$g_1(x) = 0.3820(x-0)^3 + 0(x-0)^2 + 2.0546(x-0) + 2.0000$$
$$g_2(x) = 3.1199(x-1)^3 + 1.146(x-1)^2 + 3.2005(x-1) + 4.4366$$
$$g_3(x) = -2.5893(x-1.5)^3 + 5.8259(x-1.5)^2 + 6.6866(x-1.5) + 6.7134$$

g_0 را برای پیدا کردن $g(0.66)$ استفاده می‌کنیم، جواب *3.4659* می‌باشد. (مقدار دقیق *3.4340*)

g_2 را برای پیدا کردن $g(1.75)$ استفاده می‌کنیم، جواب *8.7087* می‌باشد. (مقدار دقیق *8.4467*)

بعضی نکات در مورد این مثال:

(a) چهار نقطه داده شده است که سه فاصله تعریف می‌شود.

(b) در هر یک از سه فاصله یک $g(x)$ تعریف شده است.

(c) چون هر $g(x)$ دارای چهار ضریب است، باید *12* ضریب مجهول محاسبه شود.

با معرفی Sها، فقط لازم است که دو معادله را حل کنیم.

مثال ۷.۵ داده‌های جدول زیر از مشاهدات نجومی نوعی از ستارگان چشمک زن بدست آمده است و تغییرات را با قدرمطلق روشنایی نسبت به زمان نشان می‌دهد.

زمان	0.0	0.2	0.3	0.4	0.5	0.6	0.7	0.8	1.0
قدرمطلق روشنایی	0.302	0.185	0.106	0.093	0.240	0.579	0.561	0.468	0.302

هر چهار شرط انتهایی برای محاسبه اسپلاین‌های درجه سه را بکار برید و مقادیر درونیابی بدست آمده در هر تابع اسپلاین را در فاصله زمانی 0.05 با یکدیگر مقایسه کنید:

ماتریس‌های بسط داده شده که در جدول ۱۰.۵ نشان داده شده‌اند، از حل آنها مقادیر $S_1, S_2, \ldots, S_7$ بدست می‌آیند، یک برنامه کامپیوتری در انتهای این فصل نتایج جدول (۱۱.۵) را می‌دهد.

جدول ۱۰.۵

شرط اول			
ضرایب ماتریس :			
—	0.60	0.10	−1.23
0.10	0.40	0.10	3.96
0.10	0.40	0.10	9.60
0.10	0.40	0.10	11.52
0.10	0.40	0.10	−21.42
0.10	0.40	0.10	−4.50
0.10	0.60	—	0.60

شرط دوم			
ضرایب ماتریس :			
—	0.40	0.10	0.00
0.20	0.60	0.10	−1.23
0.10	0.40	0.10	3.96
0.10	0.40	0.10	9.60
0.10	0.40	0.10	11.52
0.10	0.40	0.10	−21.42
0.10	0.40	0.10	−4.50
0.10	0.60	0.20	0.60
0.10	0.40	—	0.00

شرط سوم			
ضرایب ماتریس :			
—	0.80	0.10	−1.23
0.10	0.40	0.10	3.96
0.10	0.40	0.10	9.60
0.10	0.40	0.10	11.52
0.10	0.40	0.10	−21.42
0.10	0.40	0.10	−4.50
0.10	0.80	—	0.60

شرط چهارم

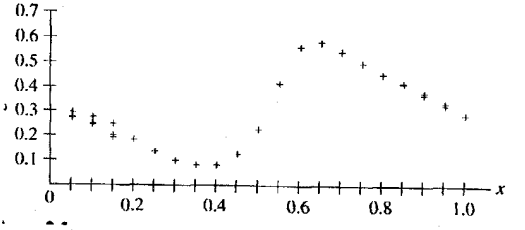
ضرایب ماتریس :			
—	1.20	−0.30	−1.23
0.10	0.40	0.10	3.96
0.10	0.40	0.10	9.60
0.10	0.40	0.10	11.52
0.10	0.40	0.10	−21.42
0.10	0.40	0.10	−4.50
−0.30	1.20	—	0.60

جدول (۱۱.۵)

t	مقادیر شرط اول	*مقادیر شرط دوم	مقادیر شرط سوم	مقادیر شرط چهارم
0.00	0.302	0.302	0.302	0.302
0.05	0.278	0.276	0.282	0.297
0.10	0.252	0.250	0.256	0.271
0.15	0.222	0.221	0.224	0.231
0.20	0.185	0.185	0.185	0.185
0.25	0.143	0.143	0.142	0.141
0.30	0.106	0.106	0.106	0.106
0.35	0.087	0.087	0.088	0.088
0.40	0.093	0.093	0.093	0.093
0.45	0.133	0.133	0.133	0.133
0.50	0.240	0.240	0.240	0.240
0.55	0.424	0.424	0.424	0.424
0.60	0.579	0.579	0.579	0.579
0.65	0.608	0.608	0.608	0.608
0.70	0.561	0.561	0.561	0.561
0.75	0.511	0.511	0.511	0.511
0.80	0.468	0.468	0.468	0.468
0.85	0.426	0.426	0.426	0.430
0.90	0.385	0.385	0.384	0.392
0.95	0.343	0.343	0.343	0.350
1.00	0.302	0.302	0.302	0.302

*توجه شود که در مقادیر شرط دوم از تفاضل‌های پیشرو و پسرو برای تقریب شیب در دو انتهای خم استفاده شده است یعنی:

$$v'(1.0) = -0.830\ ,\quad v'(0.0) = -0.585$$



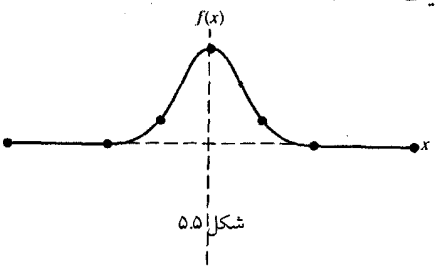
شکل ۴.۵

یک نمودار از چهار راه حل در شکل ۴.۵ داده شده است. نقاط چنان بهم نزدیک هستند که باید تقسیمات نزدیک انتها را بزرگ نمائیم تا تفاوتها مشخص شوند.

در بخش مرکزی خم ما بین زمان $t = 0.2$ و $t = 0.8$ اختلاف بیش از 0.001 نیست.

مثال ۸.۵ مطلوبست اسپلاین‌های درجه سه که برازنده تابع $f(x) = \cos^{10}(x)$ در $x = -2, -1, -0.5, 0, 0.5, 1, 2$ هستند.

در شکل ۵.۵ درونیابی اسپلاین درجه سه ساخته شده با تابع حقیقی مقایسه شده است. دو ترسیم به سختی قابل تفکیک هستند.



شکل ۵.۵

۶.۵ خم‌های بی‌زیّر *Bezier* و خم‌های *B*–اسپلاین

علاوه بر منحنی‌های اسپلاین که در بخش قبلی مطالعه کردیم، انواع مهم دیگری نیز وجود دارند. بخصوص خم‌های *B–spline, Bezier* که کاربرد وسیعی در گرافیک و طراحی کامپیوتری دارند.

این دو نوع از منحنی‌ها در واقع خم‌های اسپلاین درونیاب هستند، و معمولا از همهٔ نقاط عبور نمی‌کنند.

از این لحاظ آنها شباهت‌هایی را با منحنی‌های حداقل مربعات نشان می‌دهند که در فصل بعدی بحث خواهد شد.

هر دو منحنی *B–spline, Bezier* دارای این خاصیت مهم می‌باشند که در یک چند ضلعی که با نقاط داده شده ایجاد می‌شود باقی می‌مانند. این مطلب را بعداً توضیح خواهیم داد.

به علاوه این دو خم اسپلاین جدید دارای یک خاصیت هندسی جالب می‌باشند به این صورت که با تغییر یکی از نقاط منحنی‌های درجه سوم فقط یک قسمت از خم تحت تأثیر قرار می‌گیرد. *(Local)* در صورتیکه برای خم‌های اسپلاین درجه سه قسمت قبل، با تغییر فقط یک نقطه کلیه قسمت‌ها تحت تأثیر قرار می‌گیرند *(Global)*.

نتیجتاً برای منحنی‌های اسپلاین درجه سوم فقط نقاط داده‌ها مورد مطالعه قرار می‌گیرند.

در این منحنی‌ها که در این فصل آنها را مطالعه می‌کنیم، نقاطی که مورد سؤال هستند بیشتر شبیه به نقاط کنترل می‌باشند که برای تعیین شکل منحنی مورد نظر آنها را انتخاب کرده‌ایم و عمدتاً نوع درجه سوم این دو نوع منحنی را در نظر می‌گیریم.

در ادامه شکل پارامتری $y=f(x)$ را بیان می‌کنیم، شکل پارامتری آن با دو معادلهٔ: $y=f_2(u), x=f_1(u)$ رابطه‌ای بین x, y رابیان می‌کند.

متغیر مستقل u یک پارامتر نامیده می‌شود. برای مثال معادله دایره با پارامتر θ به صورت زیر نوشته می‌شود.

$$\begin{cases} x = r\cos(\theta) \\ y = r\sin(\theta) \end{cases}$$

هنگامی‌که x و y با یک پارامتر مثل u بیان شوند $(x(u),y(u))$ و $0 \le u \le 1$ یک سری از نقاط در ارتباط با مقادیر u مشخص می‌شوند. در مورد منحنی‌های بی‌زیّر قبل از منحنی‌های B - اسپلاین بحث خواهیم کرد.

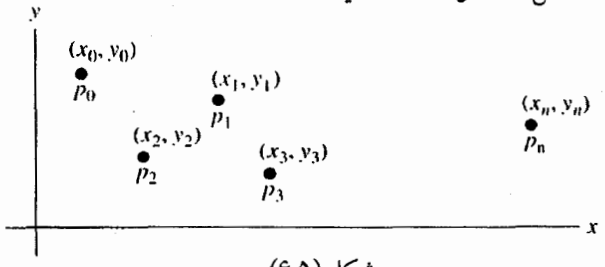
منحنی‌های بی زیّر بنام یک مهندس فرانسوی که در کارخانه اتومبیل سازی رنو کار می‌کرد نامگذاری شدند. او این منحنی ها را در اوایل سال *1960* برای برآوردن نیاز به منحنی هائی که شکل آنها باتغییر دادن چند پارامتر قابل کنترل بود گسترش داد.

کاربرد این منحنی‌های بی‌زیّر در ساخت سطوح مطلوب برای بدنه ماشین بود.

فرض کنیم یک مجموعه از نقاط کنترل داده شده‌اند $p_i=(x_i,y_i),\ i = 0, 1...,n$

(این نقاط، نقاط بی‌زیّر هم گفته می‌شوند)

برای مثال شکل (۶.۵) را ملاحظه کنید.



شکل (۶.۵)

این نقاط می‌توانند بر روی صفحه کامپیوتر با استفاده از ماوس *(mouse)* یا قلم نوری انتخاب شوند. جهت حرکت نقاط ، اختیاری است و اجباری در چپ یا به راست بودن آن نیست.

ما مؤلفه هر نقطه را بعنوان یک بردار دو بعدی بکار می‌بریم: $P_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$

یک سری از نقاط به شکل پارامتری بصورت زیر می‌باشند:

$$P(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix}, \quad 0 \le u \le 1$$

چند جمله‌ای درجه n بی زیّر با $n+1$ نقطه داده شده است.

$$p(u) = \sum_{i=0}^{n} \binom{n}{i} (1-u)^{n-i} u^i p_i ,$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

رابطه بالا دو معادلهٔ اسکالر را بیان می‌کند، یکی برای x_i و دیگری برای y_i.

برای $n = 2$ این رابطه یک معادله درجه دو را بیان می‌کند که با سه نقطهٔ p_0, p_1, p_2 مشخص می‌شود:

$$p(u) = (1)(1-u)^2 p_0 + 2(1-u)(u)p_1 + (1)u^2 p_2$$

همچنین برای $n = 2$ و $i = 0, 1, 2, \ldots$ داریم:

$$\binom{2}{0} = 1 \quad , \quad \binom{2}{1} = 2, \quad \binom{2}{2} = 1$$

معادله بالا دو معادلهٔ دیگر را بیان می‌کند:

$$x(u) = (1-u)^2 x_0 + 2(1-u)(u)x_1 + u^2 x_2$$
$$y(u) = (1-u)^2 y_0 + 2(1-u)(u)y_1 + u^2 y_2$$

مشاهده می‌کنید که اگر $u = 0$ باشد، $x(0)$ برابر x_0 و همینطور $y(0)$ برابر y_0 می‌باشد. اگر $u = 1$ باشد، مختصات نقطه (x_2, y_2) می‌شود. اگر u مقادیر بین صفر و 1 را اختیار کند، منحنی از نقطه اول تا نقطه سوم کشیده می‌شود.

بطور معمول منحنی از نقطه میانی آن سه نقطه عبور نمی‌کند (اگر نقاط روی خط مستقیم باشند، منحنی خط راستی است که از تمام آنها عبور می‌کند)، در حقیقت مؤلفه‌های نقاط منحنی درجه دوم «بی زیّر» دارای مختصاتی هستند که وزین شده و برابر مجموع مؤلفه‌های سه نقطه‌ای هستند که آن منحنی را بوجود آورده‌اند. به عبارت دیگر، معادلات «بی زیّر» را می‌توان مجموع سه چند جمله‌ای برحسب u در نظر گرفت که در آن عوامل وزنی، مؤلفه‌های سه نقطه می‌باشند.

ضمن کار برد معادله عمومی برای $n = 3$ ، چند جمله‌ای درجه سوم «بی‌زیّر» را در نظر گرفته و بعضی از جزئیات را مورد توجه قرار می‌دهیم.

خواص سایر چند جمله‌ای‌های «بی زیّر» مشابه منحنی‌های درجه سوم آن می‌باشد. در اینجا معادلات درجه سوم بی زیّر آورده شده‌اند:

$$x(u) = (1-u)^3 x_0 + 3(1-u)^2 u x_1 + 3(1-u)u^2 x_2 + u^3 x_3$$
$$y(u) = (1-u)^3 y_0 + 3(1-u)^2 u y_1 + 3(1-u)u^2 y_2 + u^3 y_3$$

مجدداً مشاهده می‌کنیم که: $p_0 = (x(0),y(0))$ ، $p_3 = (x(1),y(1))$ واینکه منحنی بطور معمول از نقاط ما بین آنها عبور نمی‌کند.

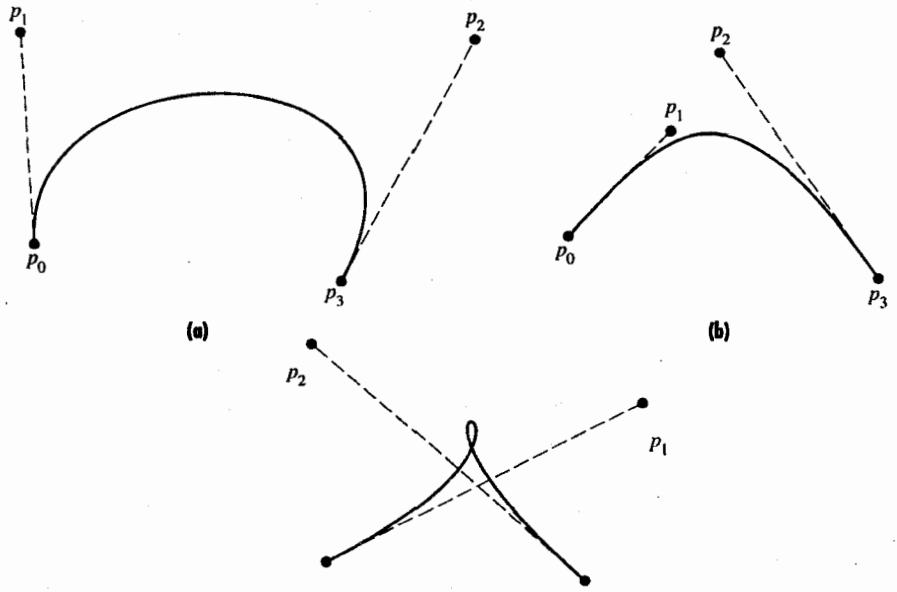
همانگونه که در منحنی‌های شکل (۷.۵) مشخص باشد، تغییر دادن نقاط کنترل میانی، شکل منحنی را عوض می‌کند، مثالها در شکل‌های (۷.۵) *(a)* تا (۷.۵) *(e)* می‌باشند.

شکل اول منحنی بی‌زیّر را با یک گروه متشکل از چهار نقطه نشان می‌دهد.

شکل‌های (۷.۵) *(d)* تا (۷.۵) *(e)* نشان می‌دهند که چگونه منحنی‌های بی زیر درجه سه فراتر از چهار نقطه ادامه می‌یابند.

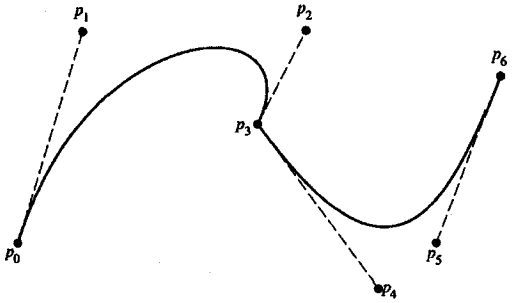
یکی از اشکال دقیقاً هفت نقطه p_6 تا p_0 را که به دو گروه چهارتائی تقسیم شده همراه با نقطه مرکزی p_3 نشان می‌دهد که این نقطه در هر دو مشترک می‌باشد.

شکل (۵.۵) *(e)* نشان می‌دهد که نقاط p_4, p_3, p_2 می‌بایستی بر روی یک خط راست قرار بگیرند که شیب در نقطه p_3 ناپیوسته و منقطع نگردد.



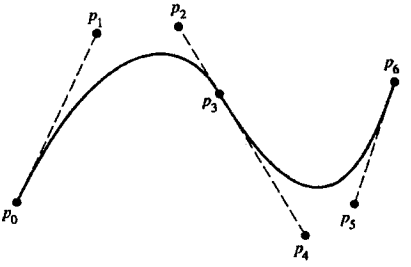
نقاط p_4, p_3, p_2 روی خط راست قرار دارند.

نقاط p_0 p_4, p_3, p_2 روی خط راست قرار ندارند.



(d)

نقاط p_4, p_3, p_2 روی خط راست قرار ندارند.



(e)

نقاط p_4, p_3, p_2 روی خط راست قرار دارند.

شکل (۷.۵)

فهرست خواص منحنی‌های بی زیّر *(Bezier)* جالب توجه میباشد:

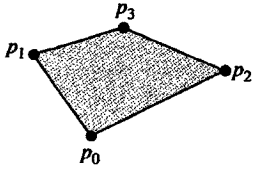
۱. $p(0) = p_0$ و $p(1)=p_3$

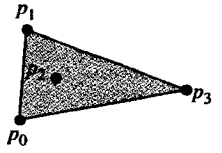
۲. با توجه به اینکه در $u=0$, $\frac{dx}{du}=3(x_1 - x_0)$, $\frac{dy}{du} = 3(y_1-y_2)$ شیب منحنی در $u = 0$ عبارتست از: $\frac{dy}{dx}=(y_1-y_0)/(x_1-x_0)$ که همان شیب خط گذرنده از p_0 و p_1 می‌باشد. بطور مشابه شیب منحنی در $u = 1$ نیز همان شیب خط گذرنده از دو نقطه آخر می‌باشد. در شکل این خطوط با خط چین مشخص شده‌اند.

۳. منحنی بی زیّر در داخل یک چند ضلعی محدب که با چهار نقطه مشخص شده احاطه گردیده است.

چند ضلعی محدب یک سری از نقاط، کوچکترین مجموعه محدبی است که تمام نقاط را شامل می‌شود.

طرح‌های مقابل نمونه‌های از چند ضلعی‌های محدب برای چهار نقطه هستند:





غالباً صحیح‌تر آن است که منحنی‌های بی زیّر را به شکل ماتریسی نمایش دهیم. به صورت زیر:

$$P(u) = [u^3, u^2, u, 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

$$= u^T M_2 p.$$

یک آلگوریتم برای رسم یک قطعه از خم بی زیّر مطابق زیر است.

چهار نقطه داده $(x_i, y_i) = 0, \ldots, 3;$

DO FOR $u = 0$ *TO* 1 *STEP* 0.01*:*

$$x = (1-u)^3 x_0 + 3(1-u)^2 u x_1 + 3(1-u)u^2 x_2 + u^3 x_3$$

$$y = (1-u)^3 y_0 + 3(1-u)^2 u y_1 + 3(1-u)u^2 y_2 + u^3 y_3$$

Plot. (x,y).

ENDDO

خم‌های B- اسپلاین

حال در مورد منحنی‌های B-اسپلاین بحث می‌کنیم. آنها شبیه به منحنی‌های بی زیر هستند از این لحاظ که معمولاً از همه نقاط داده شده نمی‌گذرند. آنها می‌توانند از هر درجه‌ای باشند، ولی ما بر روی نوع درجه سوم آنها تأکید می‌کنیم.

منحنی‌های درجه سوم B - اسپلاین مشابه منحنی‌های درجه سوم معمولی بخش قبل هستند از آن نظر که برای هر جفت از نقاط یک منحنی جدا نتیجه می‌شود. بهر حال در B - اسپلاین لازم نیست که منحنی از هیچ یک از نقاطی بگذرد که برای تعریف آن بکار رفته‌اند.

توصیف این منحنی‌ها را با بیان یک فرمول برای منحنی درجه سه B - اسپلاین که معادلات آن برحسب پارامتر u هستند آغاز می‌کنیم:

نقاط داده: $p_i = (x_i, y_i),\ i = 0, 1, \ldots n$

منحنی درجه سه B - اسپلاین برای فاصلهٔ $(p_i, p_{i+1})\ i = 1, 2, \ldots, n-1$

به صورت زیر است.

هنگامیکه

$$B_i(u) = \sum_{k=-1}^{2} b_k p_{i+k}, \quad \text{where}$$

$$b_{-1} = \frac{(1-u)^3}{6},$$

$$b_0 = \frac{u^3}{2} - u^2 + \frac{2}{3},$$

$$b_1 = -\frac{u^3}{2} + \frac{u^2}{2} + \frac{u}{2} + \frac{1}{6}, \qquad (21.5)$$

$$b_2 = \frac{u^3}{6}, \qquad 0 \le u \le 1.$$

همانطور که قبلاً گفته شد، p_i نقطه‌ای با مختصات $(x_i\ ,\ y_i)$ یا یک بردار با دو مؤلفه است. ضرایب b_k به عنوان مبنا هستند و با حرکت از یک نقطه به نقطه دیگر تغییر نمی‌کنند. مشاهده می‌شود که می‌توان آنها را به عنوان فاکتورهای وزندار برای مختصات یک مجموعه از چهار نقطه بکار برد. این مجموع وزندار با تغییر u بین صفر و یک، منحنی B-اسپلاین را تولید می‌کند.

اگر ضرایب معادلات را برای x و y از معادلهٔ 21.5 بنویسیم، خواهیم داشت:

$$x_i(u) = \frac{1}{6}(1-u)^3 x_{i-1} + \frac{1}{6}(3u^3 - 6u^2 + 4)x_i + \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1)x_{i+1} + \frac{1}{6}u^3 x_{i+2};$$

$$y_i(u) = \frac{1}{6}(1-u)^3 y_{i-1} + \frac{1}{6}(3u^3 - 6u^2 + 4)y_i + \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1)y_{i+1} + \frac{1}{6}u^3 y_{i+2}.$$

در اینجا به این نکته توجه کنید:

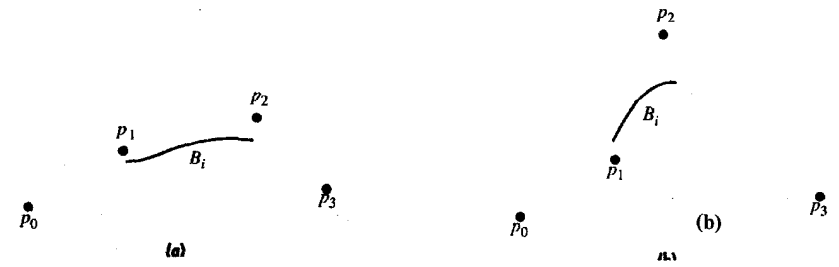
$x_i(u)$ یک تابع از u و همچنین x_i و y_i مؤلفه‌های نقطهٔ p_i می‌باشند. همانطور که گفته شد، منحنی‌های درجه سوم u به عنوان فاکتورهای وزندار روی مختصات چهار نقطه متوالی برای تولید منحنی عمل می‌کنند. برای مثال در $u=0$، $\frac{1}{6}$ و $\frac{2}{3}$ و $\frac{1}{6}$ و 0 می‌باشند. در $u = 1$ آنها 0، $\frac{1}{6}$ و $\frac{2}{3}$ و $\frac{1}{6}$ هستند. این مقادیر در فاصله $u = 0$ تا $u = 1$ تغییر می‌کنند. به عنوان یک تمرین این منحنی‌ها را رسم کنید. این عمل به شما یک درک از چگونگی تغییر فاکتورهای وزندار همراه با تغییر u را نشان می‌دهد.

اجازه دهید که به بررسی دو منحنی B – اسپلاین که دقیقاً با چهار نقطه مشخص می‌شوند بپردازیم. اشکال ۸.۵ *(a)* و ۸.۵ *(b)* اثر تغییر یک نقطه را نشان می‌دهند. همانطور که انتظار دارید وقتی p_2 به سمت بالا و چپ حرکت می‌کند، منحنی به دنبال آن کشیده می‌شود و در حقیقت در خلاف جهت p_1 کشیده می‌شود.

شما ممکن است از دیدن اینکه منحنی از دو موقعیت تقریباً نزدیک بهم شروع می‌گردد و پایان می‌یابد ولی منحنی هرگز به دو نقطه میانی خیلی نزدیک نمی‌شود، تعجب کنید.

اگر فرض کنیم یک منحنی که از تعریف معادله B_i ایجاد شده، از نزدیکی نقطهٔ p_1 تا نقطه p_2 حرکت می‌کند، مفید خواهد بود.

همچنین به خاطر بیاوریم که نقاط p_3, p_2, p_1, p_0 برای بدست آوردن B_i بکار می‌روند.



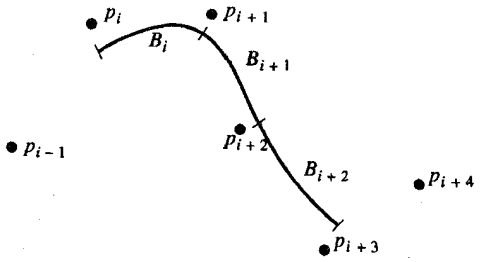
شکل (۸.۵)

از آنجایی که هر چهار نقطه در ارتباط با دو نقطه داخلی آن برای ایجاد فقط یک بخش از

منحنی B-اسپلاین مورد نیازند، باید چگونگی ایجاد منحنی B-اسپلاین را برای بیش از چهار نقطه در نظر بگیریم و علاوه بر این منحنی را برای ناحیه خارج از دو نقطه میانی توسعه دهیم.

روشی مشابه منحنی‌های درجه سه اسپلاین در بخش 5.5 را بکار می‌بریم، به این صورت که در یک زمان در امتداد یک نقطه حرکت می‌کنیم و یک مجموعه از چهار نقطه جدید را می‌سازیم. هنگامی‌که یک نقطه جدید را اضافه کردیم اولین نقطه از مجموعه قدیم را حذف می‌کنیم. شرایطی را که می‌خواهیم برای منحنی‌های B - اسپلاین برقرار گردد، دقیقاً مشابه منحنی‌های معمولی است یعنی پیوستگی منحنی و مشتقّات اول و دوم آن باید در هر نقطه برقرار باشد. نتیجه اینکه شرایط مورد نیاز این منحنی‌ها مشابه همان معادلات برای فاکتورهای وزندار (چند جمله‌ای‌های u و ضرایب b_k) می‌باشند.

شکل ۹.۵ سه قسمت متوالی یک منحنی B - اسپلاین را نشان می‌دهد.



B-اسپلاین‌های متوالی که بهم متصل گشته‌اند.

شکل (۹.۵)

خواص منحنی‌های B - اسپلاین را می‌توانیم به صورت زیر خلاصه کنیم:

۱ - همانند منحنی‌های درجه سوم اسپلاین بخش 5.5 منحنی‌های B-اسپلاین نیز به سه روش به هم متصل می‌شوند:

$$B_i(1) = B_{i+1}(0) = \frac{p_i + 4p_{i+1} + p_{i+2}}{6}, \qquad (a)$$

$$B_i'(1) = B_{i+1}'(0) = \frac{-p_i + p_{i+2}}{2}, \qquad (b)$$

$$B_i''(1) = B_{i+1}''(0) = p_i - 2p_{i+1} + p_{i+2}. \qquad (c)$$

اندیسها در اینجا به قسمت‌های منحنی و نقاط آنها در شکل (۹.۵) مربوط می‌شوند.

۲ - هر قسمت از منحنی با گروه‌های چهارتایی از نقاط مشخص می‌شوند و با چند ضلعی محدب از همین نقاط نیز احاطه می‌گردند.

حال چگونگی بدست آوردن انتهای منحنی‌های متصل بهم را بررسی می‌کنیم. اگر نقاط از p_0

تا p_n را داشته باشیم می‌توانیم منحنی‌های اسپلاین را از B_1 تا B_{n-2} بسازیم. حال نیاز به B_0 و B_{n-1} داریم. مشکل ما استفاده از طرز عملی است که توضیح داده شد و به نقاط اضافی، در خارج از محدوده نقاط داده شده نیازمندیم. شاید اگر بخواهیم نقاط ابتدایی و انتهایی از مجموعه نقاط داده شده را بهم متصل کنیم، ایده خوبی به‌نظر آید. اما چگونه می‌توانیم به‌این روش عمل کنیم؟

ابتدا، ما می‌خواهیم نقاط بیشتری را بدون بوجود آمدن حالت مصنوعی و ساختگی به نقاط اولیه اضافه کنیم به این صورت که نقاط اضافه شده در محدوده نقاط اولیه باشند. اگر ما بجای یک نقطهٔ مجازی دو نقطه مجازی یا اضافی را در هر انتها اضافه نمائیم، مشاهده می‌کنیم که منحنی‌های جدید نه تنها با بخش‌های ساخته شده قبلی بطور مناسبی متصل می‌شوند، بلکه ابتدا و انتهای حدود نقاط به همان شکلی است که مورد نظر ما می‌باشد.

بطور خلاصه – ما نقاط مجازی p_{-2} و p_{-1} را به p_0 و p_{n+1} و p_{n+2} را به p_n اضافه کرده‌ایم.
(روش‌های دیگری برای بکار بردن در بخش‌های ابتدایی و انتهایی منحنی‌های B–اسپلاین وجود دارند).

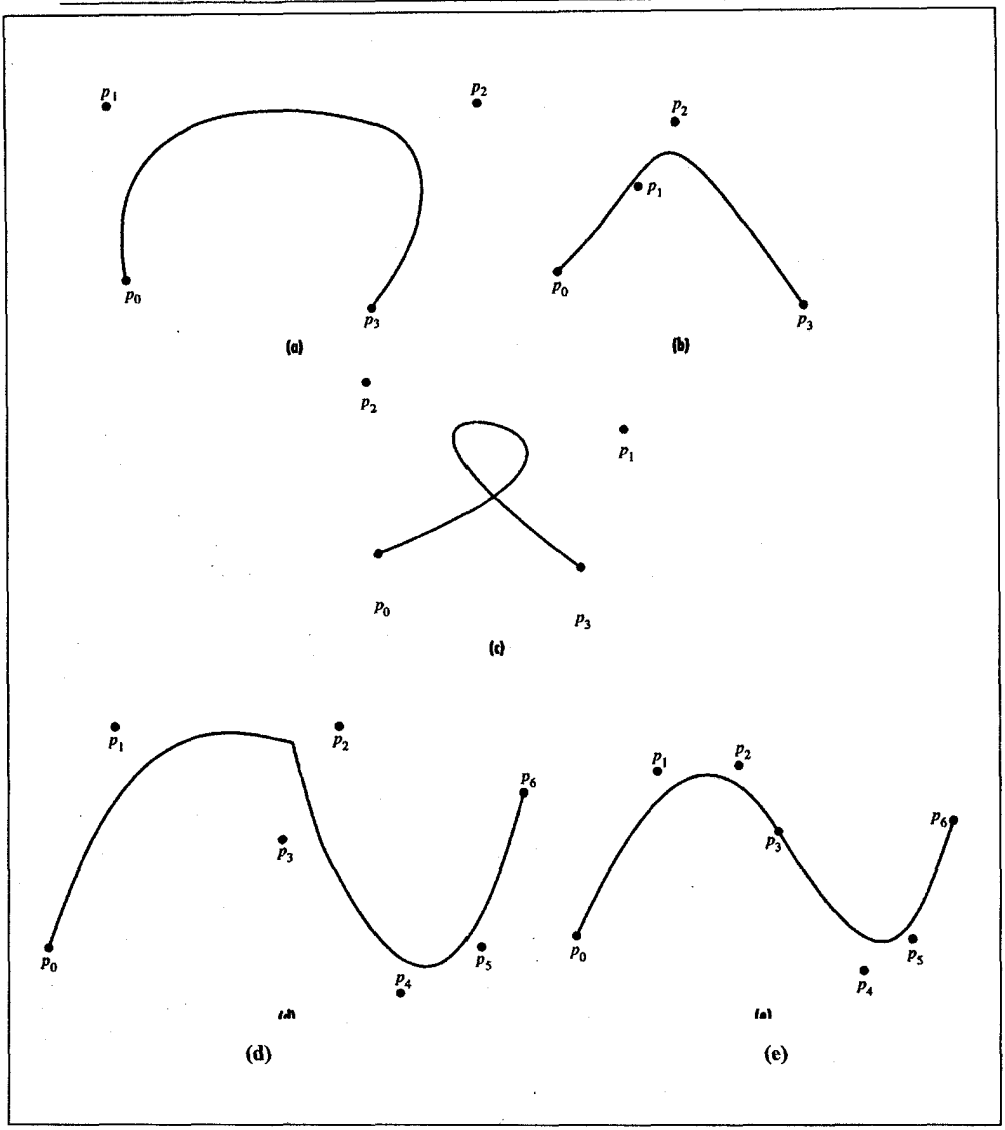
فرمول بصورت ماتریس برای منحنی‌های درجه سوم B–اسپلاین مفید بوده و به‌صورت مناسب زیر است:

$$B_i(u) = \frac{1}{6}[u^3, u^2, u, 1]\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix} = \frac{u^T M_b p}{6}. \qquad (22.5)$$

این فرمول در فاصلهٔ $[0, 1]$ و برای نقاط (p_i, p_{i+1}) بکار می‌رود.

ما این بخش را با نگاهی بر تعدادی از نمونه‌های منحنی‌های B – اسپلاین به پایان می‌بریم. پنج قسمت شکل ۱۰.۵ منحنی‌های B – اسپلاین را که با یک مجموعه از نقاط مشابه منحنی‌های بی‌زیر در شکل ۷.۵ ایجاد شده‌اند، نشان می‌دهد. با مقایسه اشکال ۱۰.۵ و ۷.۵ شما متوجه تفاوت‌های بارز بین این دو نوع منحنی خواهید شد.

۱ – برای یک منحنی B – اسپلاین، منحنی در نقاط حدی آغاز و پایان نمی‌یابد.
۲ – شیب‌های منحنی B – اسپلاین هیچ رابطه ساده‌ای با خطوط کشیده شده ما بین نقاط ندارند.
۳ – نقاط انتهایی منحنی‌های B – اسپلاین در مجاورت و نزدیکی دو نقطه میانی می‌باشند ولی نه x و نه y که مؤلفه‌های این نقاط انتهایی هستند، معمولاً با مؤلفه‌های نقاط میانی برابر نمی‌باشند.


p_1
p_2
p_0
p_3
(a)
p_2
p_1
p_0
p_3
(b)
p_2
p_1
p_0
p_3
(c)
p_1
p_2
p_6
p_3
p_0
p_5
p_4
(d)
p_1
p_2
p_3
p_6
p_0
p_5
p_4
(e)

شکل (۱۰.۵)

آلگاریتم رسم منحنی B - اسپلاین را از طریق زیر بدست می‌آوریم

```
n+1 نقطه‌داده شده‌است   p_i = (x_i, y_i), i = 0, . . . , n:
    Set p_{-1} = p_{-2} = p_0.
    Set p_{n+1} = p_{n+2} = p_n.
  DO FOR i = 0 to n − 1:
    DO FOR u = 0 TO 1 STEP 0.01:
    Compute
      x = (1 − u)^3/6x_{i−1} + (3u^3 − 6u^2 + 4)/6x_i
          + (−3u^3 + 3u^2 + 3u + 1)/6x_{i+1} + u^3/6x_{i+2},
      y = (1 − u)^3/6y_{i−1} + (3u^3 − 6u^2 + 4)/6y_i
          + (−3u^3 + 3u^2 + 3u + 1)/6y_{i+1} + u^3/6y_{i+2}.
    Plot (x, y).
    ENDDO (u).
  ENDDO (i).
```

۷.۵ تقریب چند جمله‌ایی سطوح

وقتی تابع z یک تابع چند جمله‌ای[1] از دو متغیر x و y باشد و نسبت به x از درجه سوم و نسبت به y از درجه دوم باشد، خواهیم داشت:

$$z = f(x,y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + a_6x^3 + a_7x^2y + a_8xy^2 + a_9x^3y + a_{10}x^2y^2 + a_{11}x^3y^2. \qquad (23.5)$$

یک چنین تابعی یک سطح را معرفی می‌کند که (x, y, z) یک نقطه روی آن می‌باشد. به نظر می رسد که این تابع دارای جملات بسیاری می‌باشد. اگر با چهار متغیر مستقل سرو کار داشته‌باشیم (فضای سه بعدی با متغیر زمان) در این صورت حتی چند جمله‌ای‌های با درجات پایین‌تر کاملاً پیچیده خواهند بود، به استثنای موارد خاص، مثلاً هنگامی که ما احتیاج به یک تصور روشن فرضاً از مشتق در یک نقطه دلخواه داشته باشیم. برای جلوگیری از چنین پیچیدگی‌هایی می‌توانیم هر متغیر را جداگانه بررسی کنیم.

این حالت را بررسی خواهیم کرد.

برای ساده شدن سریع معادلهٔ 23.5 توجه کنید که اگر y مقدار ثابتی را اختیار کند مثلاً $y = c$ و عامل y را با ضرائب معادله با هم در نظر بگیریم خواهیم داشت:

$$z|_{y=c} = b_0 + b_1x + b_2x^2 + b_3x^3$$

این طریق برای تقریب در نقطهٔ (a,b) خواهد بود. در یک جدول دو متغیره، یک متغیر را ثابت نگه می‌داریم مثلا $y = y_1$ و در نتیجه جدول تبدیل به یک مسئلهٔ یک متغیره خواهد شد.

روش بالا برای بدست آوردن مقدار $f(a, y_1)$ بکار می‌رود. اگر ما این عمل را با مقادیر

۱- ما یک تابع غیر چند جمله‌ای را با یک چند جمله‌ای که با این تابع مطابق می‌باشد تقریب می‌کنیم درست همانگونه که با یک تابع یک متغیر 0 عمل کردیم.

مختلف y، $y = y_2, y_3, \ldots, y_n$ تکرار نمائیم، یک جدول با x ثابت و مقدار $x=a$ با تغییر y خواهیم داشت. سپس درونیابی را در $y = b$ انجام می‌دهیم.

مثال: ۹.۵ مقدار $f(1.6, 0.33)$ را از داده‌های جدول ۱۲.۵ تخمین بزنید. برای x از درونیابی درجه دو و برای y از درونیابی درجه سه استفاده کنید.

یکی از متغیرها را برای ثابت نگهداشتن انتخاب می‌کنیم فرضاً x را (این انتخاب اختیاری است چون نتایج یکسانی را می‌دهد.) می‌خواهیم تقریب را برای y با سه ردیف جدول انجام دهیم 2 و 1.5 و $x=1.0$، زیرا مقدار خواسته شده $x=1.6$ به میانگین این سه عدد نزدیکتر است. مقدار y را 0.2 و 0.3 و 0.4 و 0.5 انتخاب می‌کنیم، زیرا میانگین آنها به $y=0.33$ نزدیکتر است.

محل سایه زده شده در جدول ۱۲.۵ ناحیه برازش چند جمله‌ای مورد نظر را نشان می‌دهد.

جدول ۱۲.۵ جدول تابع دو متغیره $z = f(x,y)$

x \ y	0.1	0.2	0.3	0.4	0.5	0.6
0.5	0.165	0.428	0.687	0.942	1.190	1.431
1.0	0.271	0.640	1.003	1.359	1.703	2.035
1.5	0.447	0.990	1.524	2.045	2.549	3.031
2.0	0.738	1.568	2.384	3.177	3.943	4.672
2.5	1.216	2.520	3.800	5.044	6.241	7.379
3.0	2.005	4.090	6.136	8.122	10.030	11.841
3.5	3.306	6.679	9.986	13.196	16.277	19.198

ممکن است از روش تفاضل‌های محدود استفاده کنیم یا جدول تفاضل‌ها را بکار برده و مقادیر درونیابی را بدست آوریم. در اینجا از روش دوم استفاده می‌کنیم، زیرا داده‌ها هم فاصله هستند.

	y	z	Δz	$\Delta^2 z$	$\Delta^3 z$
$x = 1.0$	0.2	0.640			
			0.363		
	0.3	1.003		−0.007	
			0.356		−0.005
	0.4	1.359		−0.012	
			0.344		
	0.5	1.703			
$x = 1.5$	0.2	0.990			
			0.534		
	0.3	1.524		−0.013	
			0.521		−0.004
	0.4	2.045		−0.017	
			0.504		
	0.5	2.549			
$x = 2.0$	0.2	1.568			
			0.816		
	0.3	2.384		−0.023	
			0.793		−0.004
	0.4	3.177		−0.027	
			0.766		
	0.5	3.943			

ما به یک جدول برای $y = 0.2$ تا $y = 0.5$ احتیاج داریم، چون برای هر درونیابی درجه سه احتیاج به چهار نقطه داریم، با استفاده از هر فرمول مناسب (درجه سه) نتیجه حاصل می‌شود، جدول ۱۴.۵ را بدست می‌آوریم.

	x	z	Δz	$\Delta^2 z$
$y = 0.33$	1.0	1.1108		
			0.5710	
	1.5	1.6818		0.3717
			0.9427	
	2.0	2.6245		

جدول ۱۴.۵

در جدول آخر یک رقم اعشار اضافه در نظر می‌گیریم که از خطای گردکردن جلوگیری کرده باشیم. از درونیابی دوباره $z=1.8406$ بدست می‌آید که آن را به صورت $z=1.841$ گرد می‌کنیم.

تابع جدول شده در جدول ۱۲.۵ تابع $f(x,y)=e^{x}\sin(y) + y-0.1$ می‌باشد. بنابراین مقدار حقیقی تابع $f(1.6,\ 0.33) = 1.8350$ می‌باشد. خطای $0.006-$ بدلیل آن است که درونیابی درجه دوم برای x بعلت بزرگی تفاضل‌های مرتبه دو کافی نیست. در نگاهی به گذشته می‌بینیم بهتر بود که برای y از درونیابی درجه دوم استفاده می‌کردیم، زیرا تفاضل‌های مرتبه سوم y کوچک هستند وهمچنین برای x درونیابی درجه سوم را بکار می‌بردیم.

مسئله مهم آن است که بدانیم کدامیک از مقادیر جدول ۱۲.۵ در محاسبات ما دخالت دارند. مستطیل سایه زده شده، این مقادیر را در بردارد. این سطح «ناحیه برازش» برای درونیابی چند جمله‌ای است که بکار برده‌ایم. اساس انتخاب مقادیر آن است که نقاطی که از آنها برای چند جمله‌ای درونیابی استفاده می‌شود مرکز ناحیه برازش هستند، که این مسئله به وضوح دقیقاً شبیه وضعیت جدول برای تابع یک متغیره بکار برده می‌شود. همچنین در جدول‌های سه و چهار متغیره نیز روش به همین صورت بکار برده می‌شود. البته درونیابی در این حالت به سرعت سخت و مشکل می‌شود.

یک ناحیه مستطیلی برازش تنها راه حل ممکن نیست. ما می‌توانیم درجه منحنی درونیابی را در هنگام جدول بندی سطرها و ستونها مختلف عوض کنیم. با درک مستقیم، بنظر می‌رسد بهتر باشد که برای سطرهای نزدیک به نقطه درونیابی از منحنی‌های با درجات بالاتری استفاده کنیم و هرچه که از آن نقطه دور می‌شویم درجه منحنی را کاهش دهیم. وقتی به این صورت عمل کنیم ضریب جمله خطا کاهش می‌یابد. برای درونیابی چند جمله‌ای‌ها جمله خطا کاملا پیچیده می‌شود. هنگامی‌که درجه تابع بکار رفته باین طریق کاهش می‌یابد ناحیه برازش به شکل لوزی در می‌آید.

ممکن است از چند جمله‌ای درونیابی لاگرانژ برای رویه‌ها استفاده کنیم. شاید این ساده ترین راه برای استفاده از روشی مشابه مثال بالا باشد. با ثابت در نظر گرفتن یک متغیر، چند جمله‌ای‌های لاگرانژ را برای درونیابی مقادیر داده شده نسبت به متغیر دیگر می‌نویسیم و بعد این

مقادیر را با چند جمله‌ای نهایی لاگرانژ ترکیب می‌کنیم. نتیجه نهایی یک چند جمله‌ای لاگرانژ می‌باشد که فاکتورهای تابع آن با چند جمله‌ای‌های لاگرانژ جایگزین شده‌اند.

نتایج بیان شده برای مثال بالا چنین است:

$$\frac{(y-0.3)(y-0.4)(y-0.5)}{(0.2-0.3)(0.2-0.4)(0.2-0.5)}$$
$$\times\left[\frac{(x-1.5)(x-2.0)}{(1.0-1.5)(1.0-2.0)}(0.640)+\frac{(x-1.0)(x-2.0)}{(1.5-1.0)(1.5-2.0)}(0.990)+\frac{(x-1.0)(x-1.5)}{(2.0-1.0)(2.0-1.5)}(1.568)\right]$$
$$+\frac{(y-0.2)(y-0.4)(y-0.5)}{(0.3-0.2)(0.3-0.4)(0.3-0.5)}$$
$$\times\left[\frac{(x-1.5)(x-2.0)}{(1.0-1.5)(1.0-2.0)}(1.003)+\frac{(x-1.0)(x-2.0)}{(1.5-1.0)(1.5-2.0)}(1.534)+\frac{(x-1.0)(x-1.5)}{(2.0-1.0)(2.0-1.5)}(2.384)\right]$$
$$+\frac{(y-0.2)(y-0.3)(y-0.5)}{(0.4-0.2)(0.4-0.3)(0.4-0.5)} \qquad (3.24)$$
$$\times\left[\frac{(x-1.5)(x-2.0)}{(1.0-1.5)(1.0-2.0)}(1.359)+\frac{(x-1.0)(x-2.0)}{(1.5-1.0)(1.5-2.0)}(2.045)+\frac{(x-1.0)(x-1.5)}{(2.0-1.0)(2.0-1.5)}(3.177)\right]$$
$$+\frac{(y-0.2)(y-0.3)(y-0.4)}{(0.5-0.2)(0.5-0.3)(0.5-0.4)}$$
$$\times\left[\frac{(x-1.5)(x-2.0)}{(1.0-1.5)(1.0-2.0)}(1.703)+\frac{(x-1.0)(x-2.0)}{(1.5-1.0)(1.5-2.0)}(2.549)+\frac{(x-1.0)(x-1.5)}{(2.0-1.0)(2.0-1.5)}(3.943)\right].$$

نوشتن معادله آسان است، ولی محاسبه آن با دست دشوار می‌باشد. برای نوشتن یک برنامه کامپیوتری برای درونیابی چند متغیره، روش لاگرانژ توصیه می‌شود. مزیت این روش آن است که لزومی به مساوی بودن فواصل مقادیر جدول نمی‌باشد. روش لاگرانژ شاید سرراست‌ترین و مستقیم‌ترین روش برای نوشتن چند جمله‌ای بصورت یک تابع صریح باشد. وقتی نقاط داده شده به فواصل مساوی قرار نداشته باشند، روش لاگرانژ یا روش تفاضل‌های محدود برای درونیابی بکار برده می‌شود. با روش دوم، دقیقاً اصل مشابهی بکار رفته است. یعنی یک متغیر را ثابت فرض می‌کنیم در حالیکه جدولی از مقادیر تفاضل‌های محدود ساخته می‌شوند، سپس مقادیر درونیابی از این جدول در جدول جدید ترکیب می‌شوند.

بکار بردن اسپلاین درجه سه، سطوح بی‌زیّر، و سطوح B- اسپلاین

پیشنهاد دیگر، بکار بردن منحنی‌های درجه سوم برای درونیابی در حالت‌های چند متغیره می‌باشد. شاید بهتر باشد که باز هم یک متغیر را ثابت در نظر بگیریم و منحنی‌های اسپلاین را در یک مسیر بسازیم، سپس نتایج را در مرحله دوم با هم ترکیب کنیم. بهر حال استفاده محاسباتی با اهمیت خواهد بود. درونیابی برای مقادیر توابع با دو متغیر مستقل می‌تواند بصورت ساختن یک سطحی در نظر گرفته شود که با تعدادی از نقاط داده شده تعریف شده است. بجای پیدا کردن مقادیر روی سطح که شامل نقاط داده شده می‌باشد، ما می‌توانیم سطوحی را بسازیم که به منحنی‌های بی زیّر و B- اسپلاین شبیه هستند بطوری که معمولا سطح شامل نقاط داده شده نمی‌باشد.

تاکنون ما قادر بوده‌ایم که در مواقعی که z بصورت تابعی از x و y داده شده است سطوح ساده را درونیابی کنیم، فرض کنیم که یک مجموعه از نقاط:

$$p_i = \{(x_i, y_i, z_i),\ i = 0, 1, \ldots n\}$$

داده شده‌اند ما می‌خواهیم که یک سطح را به آن نقاط تقریب کنیم. این مشابه آن است که بخواهیم یک کوه، یک هواپیما یا یک قوری چای را رسم کنیم.

ما ابتدا یک تصویر ذهنی از سطوح کلی‌تر را در نظر می‌گیریم. فرض می‌کنیم $p=(x,\ y,\ z)$ یک نقطهٔ دلخواه روی سطح باشد. بنابراین مؤلفه‌های هر نقطه مطابق معادلات زیر بیان می‌شوند:

$$x = x(u,v)$$
$$y = y(u,\ v)$$
$$z = z(u,\ v)$$

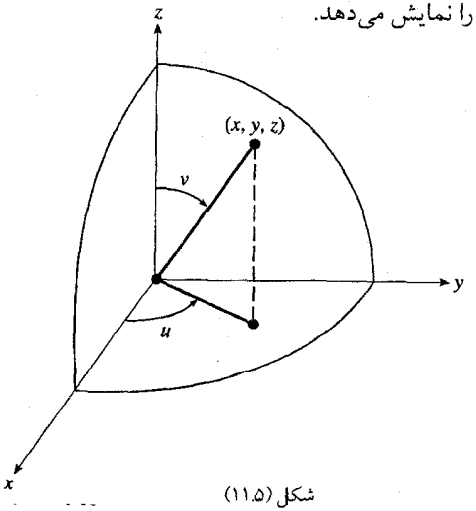
$u,\ v$ دو متغیر مستقل هستند که دامنه آنها بیش از مقادیر داده شده می‌باشد و $x,\ y,\ z$ متغیرهای وابسته هستند. این یک تغییر کوچک در نماد سازی قسمت اول این بخش می‌باشد.

یک مثال در این زمینه می‌تواند معادلات یک کره با شعاع r و به مرکز مبدأ مختصات باشد: $(0,\ 0,\ 0)$ هر نقطه روی سطح کره به این صورت داده می‌شود:

$$x = r\ cos(u)\ \ sin(v)$$
$$y = r\ sin(u)\ \ sin(v)$$
$$z = r\ cos(v)$$

دامنهٔ u از 0 تا 2π و دامنهٔ v از 0 تا π تغییر می‌کند.

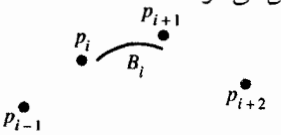
شکل ۱۱.۵ این موضوع را نمایش می‌دهد.



شکل (۱۱.۵)

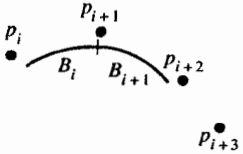
ما فقط ساختن سطح نوع B - اسپلاین را توضیح می‌دهیم. (جالبترین و منظم‌ترین توصیف سطح بی‌زیّر را می‌توان در "Crow ، 1984" یافت. همچنین Pokorny, Gerald، 1989 را ببینید.)

از بخش قبل آموختیم که یک قسمت منحنی درجه سوم B – اسپلاین که از نزدیکی نقطه p_i آغاز شده و تا نزدیکی نقطه p_{i+1} ادامه دارد با چهار نقطه مشخص می‌شود.



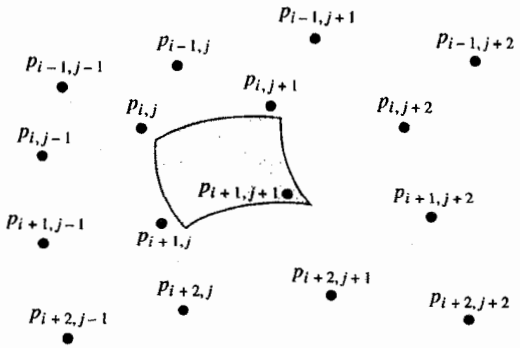
جائی که $p_i(u) = (x_i(u), y_i(u)$ در فضای دو بعدی یا $p_i(u) = (x_i(u), y_i(u), z_i(u)$ هنگام کار در فضای سه بعدی می‌باشد.

ادامه عمل با معرفی کردن نقطهٔ p_{i+3} و حذف p_{i-1} و ایجاد منحنی برای فاصله $0 \leq u \leq 1$ می‌باشد.



عملیات تا وقتی که B_{n-2} بدست آید، ادامه پیدا می‌کند و در پایان، اولین و آخرین قسمت‌های منحنی که با p_0 ، p_0 ، p_0 ، p_1 شروع و با p_{n-1} ، p_n ، p_n ، p_n به پایان می‌رسد، ایجاد خواهد شد.

در یک روش مشابه برای درونیابی یک تکّه از یک سطح B – اسپلاین به شانزده نقطه نیاز می‌باشد. مانند آنچه که در شکل ۱۲.۵ نمایش داده شده است.



شکل ۱۲.۵

در اینجا $p_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ یک نقطه در E^3 می‌باشد. این تکه سطح با محاسبه نقاط $p_{i,j}(u,v)$ برای فاصله $0 \leq u \leq 1$ و $0 \leq v \leq 1$ بوجود می‌آید. ما اندیس‌های نقاط $p_{i,j}$ را تغییر داده‌ایم تا برازش را به صورت ماتریس نمایش دهیم. برای سادگی کار، ما فقط مؤلفه x را در جزئیات بررسی می‌کنیم. معادل این فرمولها برای مؤلفه‌های y و z وجود دارند. ساده‌ترین فرمول

برای $x_{i,j}(u, v)$ بر اساس ماتریس معادله (22.5) نوشته می‌شود و بصورت زیر است:

$$x_{ij}(u,v) = \frac{1}{36}[u^3, u^2, u, 1]M_b X_{i,j} M_b^T \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}, \qquad (25.5)$$

ماتریس $x_{i,j}$ در آن یک ماتریس از مرتبه چهار می‌باشد.

$$\begin{bmatrix} x_{i-1,j-1} & x_{i-1,j} & x_{i-1,j+1} & x_{i-1,j+2} \\ x_{i,j-1} & x_{i,j} & x_{i,j+1} & x_{i,j+2} \\ x_{i+1,j-1} & x_{i+1,j} & x_{i+1,j+1} & x_{i+1,j+2} \\ x_{i+2,j-1} & x_{i+2,j} & x_{i+2,j+1} & x_{i+2,j+2} \end{bmatrix},$$

که دقیقاً مؤلفه‌های x شانزده نقطهٔ شکل ۱۲.۵ هستند.

ماتریس M_b همان ماتریس قبلی معادله 22.5 می‌باشد.

$$M_b = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}.$$

معادلات z,y با جایگزینی مناسب ماتریس‌های معادل $y_{i,j}$ ، $z_{i,j}$ که از ترکیب z، y برای شانزده نقطه بدست آمده‌اند، حاصل می‌شوند. هر یک از این معادلات بر حسب v, u از درجه سوم هستند و به آنها «معادلات درجه سوم دوگانه» گفته می‌شود.

مؤلفه‌های نقاط روی سطح به این صورت داده شده‌اند.

$$x(u,v) = \frac{1}{36}[u^3, u^2, u, 1]M_b X_{i,j} M_b^T [v^3, v^2, v, 1]^T,$$

$$y(u,v) = \frac{1}{36}[u^3, u^2, u, 1]M_b Y_{i,j} M_b^T [v^3, v^2, v, 1]^T,$$

$$z(u,v) = \frac{1}{36}[u^3, u^2, u, 1]M_b Z_{i,j} M_b^T [v^3, v^2, v, 1]^T,$$

دامنه u و v بین 0 و 1 می‌باشد.

به راحتی اثبات می‌شود که فاکتورهای مؤثر یا مجموعه فاکتورهای بکار رفته در هر نقطه عبارتند از:

در نقطهٔ $p_{i,j}(u,v)$ (برای $v = 0$, $u = 0$):

$$\begin{bmatrix} 1 & 4 & 1 & 0 \\ 4 & 16 & 4 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

در نقطه $p_{i,j}(u,v)$ (برای $v = 1$, $u = 1$):

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 4 & 16 & 4 \\ 0 & 1 & 4 & 1 \end{bmatrix}$$

هر جزء (i, j)ام ضریبی برای نقطه معادلش در شکل ۱۲.۵ می‌باشد. در واقع این ماتریسها

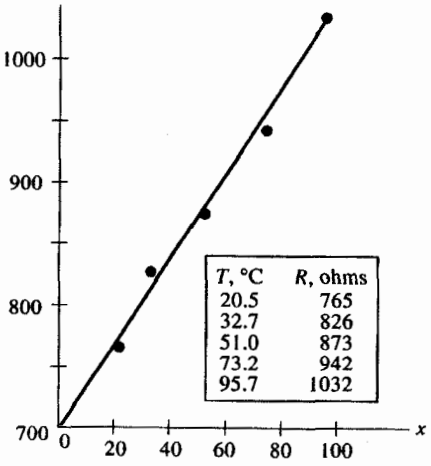
سطوحی هستند که نقاط نشان داده شده در شکل ۱۲.۵ را می‌پوشانند. بخشی از سطح با افزودن سطر یا ستون دیگری از نقاط و حذف نمودن یک سطر یا ستون معادل نقاط توسعه می‌یابد. باید تصدیق کنیم که بخش‌های قبلی و فعلی سطوح به ملایمت در طول لبهٔ اتصالشان به یکدیگر مرتبط شده‌اند. قسمت اولیه یا نهایی سطح مورد نظر با تکرار یک گوشه، همانگونه که برای منحنی‌های B-اسپلاین گفته شد قابل مشاهده خواهد بود. این ما را مطمئن خواهد کرد که آن قسمت سطح عملاً از یک نقطه آغاز شده پایان می‌یابد.

برای یک منحنی سه بار یک نقطه را تکرار می‌کنیم در صورتیکه برای یک سطح این تکرار نه بار خواهد بود. (برای دریافت جزئیات بیشتر بحث آموزشی درباره برازش منحنی‌ها و سطوح به *pokorny & Gerald (1988)* مراجعه نمایید.)

۸.۵ روش تقریب حداقل مربعات*

فرض کنید می‌خواهیم که یک منحنی را برداده‌های تقریبی مفروضی برازش کنیم. بعنوان مثال، داده‌های حاصل از تشخیص اثرات دما بر مقاومت که بوسیلهٔ یک دانشجو در آزمایشگاه فیزیک اندازه‌گیری شده است را در نظر می‌گیریم. دما و مقاومت اندازه‌گیری شده را مطابق شکل ۱۳.۵ ثبت کرده‌ایم. بطوری که نمودار یک رابطه خطی را پیشنهاد می‌کند. حال می‌خواهیم که بطور مناسبی ثابت‌های a,b را در رابطه بین مقاومت (R) و دما (T) تعیین کنیم،

$$R=aT + b \qquad (26.5)$$



شکل (۱۳.۵)

بطوریکه در استفاده‌های بعدی بتوانیم میزان مقاومت را با توجه به درجه حرارت پیش بینی کنیم. خطی که آنرا به چشم تشخیص می‌دهیم داده‌ها را نسبتاً خوب نشان میدهد. امّا اگر داده‌ها را

*— Least Square

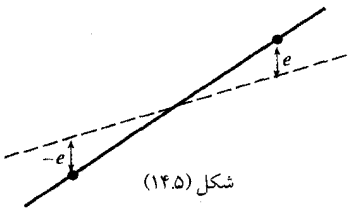
دوباره رسم کنیم و از شخص دیگری بخواهیم که خطی رسم کند، او به ندرت ممکن است همان خط را رسم کند.

یکی از خواسته‌های ما برای برازش خم بر داده‌ها آنست که فرایند، نامبهم و واضح باشد. ضمناً از سوی دیگر علاقه‌مندیم که انحراف نقاط از خط را به می‌نیمم خود برسانیم. که این انحراف با توجه به فاصله نقاط از خط، قابل اندازه‌گیری است. چگونگی محاسبهٔ این فاصله‌ها مربوط به آنست که آیا هر دو متغیّر تابع خطا هستند یا خیر.

فرض می‌کنیم خطای خواندن درجه حرارت در شکل ۱۳.۵ قابل اغماض است، بنابراین کلیهٔ خطاها مربوط به اندازه‌گیری مقاومت هستند و ماز فاصله عمودی استفاده خواهیم کرد (یعنی خطی قائم رسم می‌کنیم و فاصله نقطه تامحل تقاطع این خط و خط تقریب را بعنوان انحراف داده‌ها قلمداد می‌کنیم.) (چنانچه هر دو متغیر دارای خطا باشند، می‌بایست از فاصله عمودی استفاده کنیم، و در اینصورت مسئله بطور قابل توجّهی پیچیده خواهد شد، ولی ما با حالت ساده‌تر مطابق زیر عمل خواهیم کرد.)

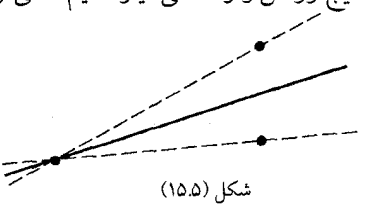
ممکن است ابتدا فکر کنیم که می‌توانیم انحراف‌ها را بامی‌نیمم کردن مجموع آنها به حداقل برسانیم ولیکن این روش معیار خوبی بدست نمی‌دهد. توجه خود را به حالتی که تنها دو نقطه داریم معطوف می‌کنیم. شکل ۱۴.۵

واضح است که بهترین خط، خط گذرنده از هر دوی آنهاست. ولی هر خطی که از نقطه میانی قطعه خط اتصال آن دو نیز بگذرد مجموع خطاها برابر صفر خواهد بود.



شکل (۱۴.۵)

پس در مورد می‌نیمم کردن مجموع قدر مطلق خطاها چه باید کرد؟ روش فوق در جایی که مطابق شکل (۱۵.۵) سه نقطه داشته باشیم نیز غیر کافیست. فرض کنید دو تا از این نقاط دارای مقدار x یکسان باشند (که البته این فرض دور از واقع نیست زیرا آزمایشات مکرر نتایج متفاوت می‌دهند.) واضح است که بهترین خط از میانگین نقاط آزمایش مجدد می‌گذرد و در حالیکه، هر خطی که در میان دو خط نقطه چین واقع شود، دارای مجموع یکسانی در مجموع قدر مطلق‌های فواصل عمودی خواهد بود، و از آنجا که ما به نتایج روشن و واضحی نیازمندیم، نمی‌توانیم از این روش در کارمان استفاده کنیم.


شکل (۱۵.۵)

ممکن است معیاری را برگزینیم که در آن اندازهٔ ماکزیمم قدر مطلق خطا را می‌نیمم کنیم (که معمولاً معیار *minimax* نامیده می‌شود.) لیکن در این مسائل معمولاً این روش به ندرت بکار می‌رود. این معیار نیز مطلوب نیست زیرا قدر مطلق تابع در مبدأ مشتق ندارد، و ضمناً احساس می‌شود که اهمیّت بیش از حدی به یک خطای بزرگ داده شده است. تخمین متداول آنست که مجموع مربعات خطاها را می‌نیمم کنیم که به قاعده حداقل مربعات موسوم است.

علاوه بر بدست آوردن یک نتیجه منحصر به فرد برای یک دسته از داده‌ها، روش حداقل مربعات با قانون ماکزیمم درست نمایی[1] در آمار نیز مطابقت دارد. اگر خطاهای اندازه‌گیری دارای توزیع نرمال باشند و اگر انحراف معیار در همهٔ دادها ثابت باشد، می‌توان نشان داد خطی که از می‌نیمم کردن حداقل مربعات حاصل می‌شود، دارای همان شیب و عرض از مبدایی است که روش ماکزیمم درست‌نمایی ایجاد می‌کند.

فرض کنید Y_i نمایانگر مقدار تقریبی است و y_i مقدار حاصل از معادلات زیر می‌باشد $y_i=ax_i+b$ ، بطوری که x_i بدون خطا فرض می‌شود، می‌خواهیم بهترین مقادیر را برای b,a معین کنیم تا yهای مقادیر تابع متناظر با مقادیر xرا بدست آوریم. فرض کنیم خطا برابر است با $e_i=Y_i - y_i$ و مجموع مربع خطاها باید مینیمم شود.

$$\begin{aligned} S &= e_1^2 + e_2^2 + \cdots + e_N^2 \\ &= \sum_{i=1}^{N} e_i^2 \\ &= \sum_{i=1}^{N} (Y_i - ax_i - b)^2 \end{aligned}$$

N تعداد زوج‌های (x, Y) است. با انتخاب مناسب پارامترهای a,b به مقدار می‌نیمم می‌رسیم، بنابراین b, a متغیّرهای مسئله هستند. وقتی S می‌نیمم است که دو مشتق جزئی $\frac{\partial s}{\partial b}\ \frac{\partial s}{\partial a}$ هر دو صفر باشند. از اینرو با توجّه به اینکه Y_i و x_i نقاط داده شده هستند و بستگی به انتخاب ما برای مقادیر a,b نخواهند داشت، داریم:

$$\frac{\partial S}{\partial a} = 0 = \sum_{i=1}^{N} 2(Y_i - ax_i - b)(-x_i),$$
$$\frac{\partial S}{\partial b} = 0 = \sum_{i=1}^{N} 2(Y_i - ax_i - b)(-1).$$

با تقسیم هر یک از معادلات فوق بر $2-$ و بسط مجموع آن‌ها معادلات نرمال زیر بدست می‌آیند.

$$\begin{aligned} a\sum x_i^2 + b\sum x_i &= \sum x_i Y_i, \\ a\sum x_i + bN &= \sum Y_i. \end{aligned} \qquad (27.5)$$

۱ – Maximum likekihood

کلیه جمع‌ها در معادلات (27.5) از $i=1$ تا $i=N$ می‌باشند. با حل این معادلات به مقادیر جواب برای شیب خط و عرض از مبداء a,b می‌رسیم.

برای داده‌های شکل (۱۳.۵) نتیجه می‌شود که.

$$N = 5, \quad \sum T_i = 273.1, \quad \sum T_i^2 = 18{,}607.27, \quad \sum R_i = 4438,$$

$$\sum T_i R_i = 254{,}932.5.$$

که در اینصورت معادلات نرمال عبارتنداز:

$$18607.27a + 273.1b = 254932.5$$

$$273.1a + 5b = 4438$$

که از این معادلات بدست می‌آید: $a = 3.395 \; ; \; b = 702.2$ و از اینرو معادلهٔ (26.5) را بصورت زیر می‌نویسیم: $R = 702 + 3.39T$.

داده‌های غیر خطی[1]

حقیقتاً، در بسیاری موارد، داده‌های آزمایش‌های تجربی خطی نیستند، بنابراین لازم است که تابع دیگری غیر از چند جمله‌ای درجه اوّل بر داده‌ها برازش گردد. شکل رایجی که مورد استفاده قرار می‌گیرد. فرم‌های نمایی است:

$$y=ae^{bx} \qquad ,y=ax^b$$

می‌توانیم به طور مشابه برای این حالات نیز معادلات نرمال را تعمیم دهیم و از روش حداقل مربعات استفاده نماییم و مشتقات جزئی را برابر صفر قرار می‌دهیم.

چنین معادلات غیر خطی بسیار دشوارتر از معادلات خطی قابل حل هستند، بدین منظور شکل نمایی را قبل از تعیین پارامترها به وسیلهٔ گرفتن لگاریتم از دو طرف به صورت معادلات خطی در می‌آوریم:

$$Ln\ y = Ln\ a + b\ Ln\ x\ ,\ Ln\ y = Ln\ a + bx$$

و اکنون ما تابع جدید $z=Ln\ y$ را بعنوان تابع خطی از x یا $Ln\ x$ برداده‌ها، مطابق روش ذکر شده، برازش می‌کنیم. البته این نکته قابل توجه است که در اینجا، مجموع مربعات انحراف Y از منحنی را می‌نیمم نمی‌کنیم، بلکه مجموع مربعات انحراف LnY را مینمم می‌کنیم، بنابراین همین مقادیر که برای مینیمم کردن درصد خطاها بکار رفته‌اند، خود ممکن است بعنوان صورت مطلوب در نظر گرفته شوند. یکی از مزایای شکل‌های خطی آنست که چنانچه داده‌ها را در صفحه لگاریتم (Ln یا log) رسم کنیم، بایک نگاه مشاهده می‌شود که آیا این شکل‌ها قابل تطبیقند و آیا خطی راست نشان دهنده داده‌هایی است که رسم می‌شود.

تغییر متغیرها برای ایجاد منحنی‌های خطی مانند تبدیل و رسم: $1/x$, $1/(ax+b)$, $1/x^2$ و

1- Nonlinear Data

دیگر شکل‌های غیر چند جمله‌ای ممکن است با تغییر کافی شیب اجازه رسم منحنی هـمواری را بدهند. خمهایی به شکل (S) را نمی‌توان به آسانی خطی ساخت؛ گاهی رابطهٔ گامپرتز[1]:

$$y=ab^{cx}$$

را بکار می‌گیرند که ثابت‌های c,b,a با بررسی خاصی تشخیص داده می‌شوند. رابطهٔ دیگری کـه داده‌ها را با منحنی شکل S تقریب می‌زند برابر است با:

$$\frac{1}{y}=a+be^{-x}$$

در حالت‌های نامناسب، می‌توان نواحی مورد نظر را به زیر ناحیه‌های مناسب تقسیم کرد و در هر زیر ناحیه یک برازش جداگانه بدست آورد.

مشکل روش‌های ترسیمی ارائه شدهٔ فوق عدم یکتایی جواب است. دو شـخص مـختلف هیچگاه منحنی یکسانی را از میان نقاط داده شده نخواهند گذراند. قضاوت آنان بواسطه نقاطی است که از بقیه داده‌ها بطور زیادی انحراف دارند. همچون زمانی که شخصی می‌خواهد در مقایسه با نقاط درون ناحیهٔ مرکزی داده‌ها توجه خود را بیشتر به کرانه‌ها و دو انتهای منحنی معطوف نماید.

مشکلات بیشتر، در هنگام انتگرال‌گیری و یا مشتق‌گیری از تابع رخ می‌دهند. بحث مـا در مورد چند جمله‌ای‌ها با حداقل مربعات خطا یکی از راه‌های حل این مشکل است.

از آنجا که چند جمله‌ایها به راحتی قابل استفاده و محاسبه هستند، یکی از روش‌های رایج برازش چنین توابع برداده‌هایی می‌باشند که قابل خطی کردن نیستند. اکنون این حـالت را بـررسی می‌کنیم. ضمناً خواهیم دید که برای این حالت معادلات نرمال بصورت خطی هستند، که این خود مزیت دیگر این روش است. در ادامه، از n به عنوان درجه چـند جـمله‌ای و از N بـعنوان تـعداد زوج‌های نقاط داده استفاده خواهیم کرد.

بدیهی است چنانچه $N = n+1$ باشد، چند جمله‌ای دقیقاً از کلیه نقاط داده شده خواهد گذشت و چند جمله‌ای در این حالت از روش مذکور در فصل سوّم بدست می‌آید، بنابراین از این به بعد همواره فرض می‌کنیم که $N>n+1$.

تابع را به صورت زیر می‌نویسیم:

$$y= a_0 + a_1x + a_2x^2+....+ a_nx^n= P_n(x) \qquad (28.5)$$

و خطا مطابق زیر تعریف می‌شود.

$$e_i = Y_i - y_i = Y_i - a_0 - a_1x_i - a_2x_i^2 - \cdots - a_nx_i^n.$$

Y_i را به عنوان نمایش مقدار نتیجه آزمایش مربوط به x_i در نظر می‌گیریم و y_i بدون خـطا فرض می‌شود.

اکنون ما مجموع مربعات خطاها را مینیمم می‌کنیم

$$S = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - a_0 - a_1x_i - a_2x_i^2 - \cdots - a_nx_i^n)^2.$$

۱- Gompertz

در نقطه مینیمم کلیه مشتقات جزئی: $\frac{\partial s}{\partial a_0}$, $\frac{\partial s}{\partial a_1}$, ..., $\frac{\partial s}{\partial a_n}$ صفر می‌گردند. که با نوشتن معادلات مربوط به آنها $n+1$ معادله بدست خواهد آمد:

$$\begin{aligned}
\frac{\partial S}{\partial a_0} &= 0 = \sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - \cdots - a_i x_i^n)(-1),\\
\frac{\partial S}{\partial a_1} &= 0 = \sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - \cdots - a_i x_i^n)(-x_i),\\
&\vdots \qquad\qquad \vdots\\
\frac{\partial S}{\partial a_n} &= 0 = \sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - \cdots - a_n x_i^n)(-x_i^n).
\end{aligned}$$

با تقسیم کلیه معادلات بر $2-$ و مرتب سازی آنها، به $n+1$ معادله نرمال توأم به صورت زیر می‌رسیم که همزمان حل خواهند شد.

$$\begin{aligned}
a_0 N + a_1 \sum x_i + a_2 \sum x_i^2 + \cdots + a_n \sum x_i^n &= \sum Y_i,\\
a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \cdots + a_n \sum x_i^{n+1} &= \sum x_i Y_i,\\
a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \cdots + a_n \sum x_i^{n+2} &= \sum x_i^2 Y_i,\\
&\vdots \qquad\qquad\qquad\qquad \vdots\\
a_0 \sum x_i^n + a_1 \sum x_i^{n+1} + a_2 \sum x_i^{n+2} + \cdots + a_n \sum x_i^{2n} &= \sum x_i^n Y_i.
\end{aligned} \tag{29.5}$$

پس از قرار دادن این معادلات به صورت ماتریس، یک الگوی جالب در ماتریس ضرایب بدست می‌آید.

$$\begin{bmatrix}
N & \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots \sum x_i^n\\
\sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \cdots \sum x_i^{n+1}\\
\sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \cdots \sum x_i^{n+2}\\
\vdots & & & & \vdots\\
\sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \sum x_i^{n+3} & \cdots \sum x_i^{2n}
\end{bmatrix} a = \begin{bmatrix}
\sum Y_i\\
\sum x_i Y_i\\
\sum x_i^2 Y_i\\
\vdots\\
\sum x_i^n Y_i
\end{bmatrix}. \tag{30.5}$$

کلیه جمع‌ها در معادلات *(29.5)* و *(30.5)* از *1* تا *N* هستند.

البته حل یک دستگاه بزرگ معادلات خطی خود کار ساده‌ای نیست، روش حل این نوع دستگاهها خود مبحث فصل سوم را تشکیل داده‌اند. این معادلات خاص یک مشکل دیگر را نیز به همراه دارند که آن عبارتست از یک سری خواص نامطلوب آنها که به نام ناهنجار (ناجور) موسومند. *(Ill-Condition)*

علت این پدیده آنست که به هنگام حل دستگاه، خطای‌های گردکردن باعث‌ایجاد خطاهای غیر عادی بزرگ در جواب‌ها می‌گردد که در مجهولات، یعنی ضرایب a_i در معادله *(28.5)* اثر

می‌گذارد، درجه چند جمله‌ای تا حدود $n=4$ و $n=5$ چندان مشکل آفرین نیست (یعنی استفاده از دقت مضاعف در کامپیوتر دلخواه بوده و ضروری نیست) ولی برای درجه بیش از این حالات روش‌های خاصی در حل مسئله لازم است. بعضی از روش‌های خاص از چند جمله‌ای‌های متعامد معادل با معادلهٔ *(28.5)* استفاده می‌نمایند. از نقطه نظر آزمایش کنندگان توابع پیچیده‌تر از چند جمله‌ای‌های درجهٔ چهار یا درجه پنج و بیشتر به ندرت مورد استفاده است. معمولا مسئله با برازش یک سری از چند جمله‌ایها در زیر مجموعه‌هایی از داده‌ها، قابل حل است.

ماتریس معادلات *(30.5)* ماتریس نرمال برای مسأله حداقل مربعات نامیده می‌شود. ماتریس دیگری که متناظر این ماتریس میباشد. ماتریس طراحی[۱] نامیده می‌شود و به شکل زیر است.

$$A = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_N \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_N^2 \\ \vdots & & & & \vdots \\ x_1^n & x_2^n & x_3^n & \dots & x_N^n \end{bmatrix}.$$

به آسانی نشان داده می‌شود که AA^T دقیقاً برابر ضرایب معادله *(30.5)* می‌باشد. همچنین Ay بطوری که y بردار ستون مقادیر Y می‌باشد و طرف راست معادله *(30.5)* را می‌دهد. این به معنی آنست که معادله *(30.5)* را مجدداً بشکل ماتریس می‌نویسیم.

$$AA^T a = Ba = Ay$$

معمولا مایلیم دستگاه را با روش حذفی گاوس حل کنیم اما بخاطر خواص B می‌توانیم سایر روشها را بکار بریم تا از مسأله ناهنجاری که قبلا به آن اشاره شد اجتناب کنیم.

۱ - ماتریس $B=AA^T$ متقارن و نیمه معین مثبت[۲] است. یک ماتریس $N*N$ مانند M نیمه معین مثبت گفته می‌شود اگر برای هر n عنصر بردار X داشته باشیم $X^TMX \geq 0$ اگر این شرط را اضافه کنیم که $X^TMX = 0$. فقط اگر X یک بردار صفر باشد، به M معین مثبت گفته می‌شود.

۲ - در جبر خطی، نشان داده می‌شود که B می‌تواند با ماتریس متعامد P قطری شود.

$$PBP^T = PAA^TP^T = D$$

بطوری که عناصر قطر D مقادیر ویژه B هستند. توجه شود که نتیجه تعامد $PP^T = I$ یک ماتریس یکّه است.

۳ - چون B نیمه معین مثبت است، تمام مقادیر ویژه B غیر منفی هستند. این بدین معنی است که ما می‌توانیم ماتریس S را تعریف کنیم که

$$S=\sqrt{D} \text{ یا } S^2 = D$$

عناصر قطری S مقادیر منفرد A نامیده می‌شود.

۴ - می‌توانیم معادلات *(30.5)* وجواب آنرا مطابق زیر بنویسیم:

۱ - Design matrix

۲- Positive semidefintie

$$AA^T a = P^T DPa = PS(PS)^T a = Ay$$
$$a = PD^{-1}P^T Ay.$$

توسعه این طریق، روش مهمی برای حل معادلات *(30.5)* که تجزیه مقدار منفرد[1] نامیده می‌شود، ارائه می‌کند.

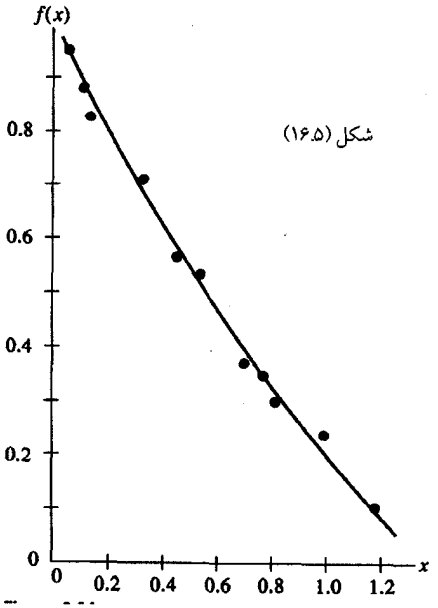
داده‌های زیر را برای روشن ساختن برازش منحنی در نظر می‌گیریم.

x_i	0.05	0.11	0.15	0.31	0.46	0.52	0.70	0.74	0.82	0.98	1.17
Y_i	0.956	0.890	0.832	0.717	0.571	0.539	0.378	0.370	0.306	0.242	0.104

$\sum x_i = 6.01$	$N = 11$
$\sum x_i^2 = 4.6545$	$\sum Y_i = 5.905$
$\sum x_i^3 = 4.1150$	$\sum x_i Y_i = 2.1839$
$\sum x_i^4 = 3.9161$	$\sum x_i^2 Y_i = 1.3357$

جدول (۱۵.۵)

کاربرد معادلات *(29.5)* را با برازش یک معادلهٔ درجهٔ دوم برداده‌های جدول (۱۵.۵) روشن می‌سازیم.



شکل (۱۶.۵)

شکل (۱۶.۵) یک ترسیم از داده‌ها را نشان می‌دهد. (در حقیقت داده‌ها دارای انحرافی از معادلهٔ درجهٔ دوم $y=1-x+0.2x^2$ هستند. و جالب خواهد بود که ببینیم چگونه بصورت مناسبی این تابع را تقریب می‌زنیم.)

برای بدست آوردن معادلات نرمال، به مجموع مقادیر در جدول (۱۵.۵) نیاز داریم. با یک ماشین حساب می‌توانیم نتیجه را حساب کنیم. یک برنامهٔ کامپیوتری در انتهای این فصل برای محاسبهٔ ماتریس ضرایب و بردار سمت راست آن طراحی شده است.

لازم است که دستگاه معادلات زیر را حل کنیم.

$$11a_0 + 6.01a_1 + 4.6545a_2 = 5.905,$$
$$6.01a_0 + 4.6545a_1 + 4.1150a_2 = 2.1839,$$
$$4.6545a_0 + 4.1150a_1 + 3.9161a_2 = 1.3357.$$

۱- singular value

نتیجه چنین خواهد بود: $a_0 = 0.995 ; a_1 = -1.018 ; a_2 = 0.225$ و بنابراین با بکارگیری روش حداقل مربعات خواهیم داشت.

$$y = 0.998 - 1.018x + 0.225x^2$$

با معادله $y = 1 - x + 0.2x^2$ مقایسه نمایید. به جهت خطاهای داده‌ها انتظار نداریم که ضرایب بطور کاملاً دقیق بدست آیند.

درجه چند جمله‌ای مناسب چیست؟

در حالت کلی ممکن است درجه چند جمله‌ای مناسب را ندانیم. هر چه که چند جمله‌ای با درجهٔ بالاتری را بکار بریم، ممکن است میزان انحراف نقاط از منحنی کاهش یابد، تا جایی که n یعنی درجهٔ چند جمله‌ای برابر با $N-1$ شود، که در آنصورت دقیقاً منحنی نظیر را بدست آورده‌ایم، (با این فرض که به ازای هر مقدار x تنها یک مقدار y داشته باشیم) و پاسخ تعیین درجه چند جمله‌ای در آمار یافت می‌شود. برای تعیین چند جمله‌ای مناسب باید افزایش درجهٔ چند جمله‌ای تقریب را تا آنجا ادامه دهیم که کاهش پراش σ^2 از نظر آماری با معنی باشد. برازش از رابطهٔ زیر محاسبه می‌گردد.

$$\sigma^2 = \frac{\sum e_i^2}{N - n - 1} \qquad (31.5)$$

در مثال مذکور، هنگامی که درجهٔ منحنی برازنده داده‌ها از 1 تا 7 تغییر می‌کند، نتایج در جدول (۱۶.۵) نشان داده شده است.

جدول (۱۶.۵)

درجه	معادله	σ^2	$\sum e^2$
1	$y = 0.952 - 0.760x$	0.0010	0.0092
2	$y = 0.998 - 1.018x + 0.225x^2$	0.0002	0.0018
3	$y = 1.004 - 1.079x + 0.351x^2 - 0.069x^3$	0.0003	0.0018
4	$y = 0.998 - 0.838x - 0.522x^2 + 1.040x^3 - 0.454x^4$	0.0003	0.0016
5	$y = 1.031 - 1.704x + 4.278x^2 - 9.477x^3 + 9.394x^4 - 3.290x^5$	0.0001	0.0007
6	$y = 1.038 - 1.910x + 5.952x^2 - 15.078x^3 + 18.277x^4 - 9.835x^5 + 1.836x^6$	0.0002	0.0007
7	$y = 1.032 - 1.742x + 4.694x^2 - 11.898x^3 + 16.645x^4 - 14.346x^5 + 8.141x^6 - 2.293x^7$	0.0002	0.0007

تخمین معادله (31.5) نشان می‌دهد درجهٔ چند جمله‌ای بهینه 2 می‌باشد و مهم است که بدانیم، صورت رابطه (31.5) یعنی مجموع مربعات انحراف نقاط از منحنی، با افزایش درجهٔ چند جمله‌ای می‌تواند مستمراً کاهش یابد. و این مخرج رابطهٔ (31.5) است که باعث افزایش σ^2 به هنگام گذشتن

از درجه اپتیمم می‌گردد. در این مثال این رفتار برای $n=3$ مشاهده می‌گردد. پس از $n=3$ خطای دیگری ظاهر می‌گردد. بر اثر حالت ناهنجاری، ضرایب چند جمله‌ای حداقل مربعات با دقت کمی بدست می‌آیند. و این باعث افزایش در مقدار σ^2 می‌گردد. قبل از خاتمهٔ این بخش، نشان می‌دهیم که این روشها را چگونه برای توابع پیچیده‌تر بکار می‌گیریم.

مثال ۱۰.۵ نتایج حاصل از آزمایش تونل باد در عبور جریان باد در انتهای بال یک هواپیما به صورت زیر بدست آمده است،

R/C: 0.73, 0.78, 0.81, 0.86, 0.875, 0.89, 0.95, 1.02, 1.03, 1.055, 1.135, 1.14, 1.245, 1.32, 1.385, 1.43, 1.445, 1.535, 1.57, 1.63, 1.755;

V_θ/V_∞: 0.0788, 0.0788, 0.064, 0.0788, 0.0681, 0.0703, 0.0703, 0.0681, 0.0681, 0.079, 0.0575, 0.0681, 0.0575, 0.0511, 0.0575, 0.049, 0.0532, 0.0511, 0.049, 0.0532, 0.0426;

بطوریکه R فاصله از مرکز انحناء (شعاع انحناء) و C طول قوس بال هواپیماست و V_θ نیز سرعت زاویه‌ای مماسی و V_∞ سرعت حد هواپیما در جریان آزاد است. اگر فرض کنیم $y=\frac{V_\theta}{V_\infty}$, $x=\frac{R}{C}$ معادله مورد نظر برابر است با:

$$g(x) = \frac{A}{x}(1-e^{-\lambda x^2})$$

و معادلات حداقل مربعات خواهند شد:

$$S = \sum_{i=1}^{21}(Y_i - g(x_i))^2$$
$$= \sum_{i=1}^{21}\left(Y_i - \frac{A}{x_i}(1 - e^{-\lambda x_i^2})\right)^2.$$

و با قرار دادن $\frac{\partial A}{\partial a} = \frac{\partial s}{\partial \lambda} = 0$ بدست می‌آید:

$$\sum_{i=1}^{21}\left(\frac{1}{x_i}\right)(1 - e^{-\lambda x_i^2})\left(Y_i - \frac{A}{x_i}(1 - e^{-\lambda x_i^2})\right) = 0,$$
$$\sum_{i=1}^{21} x_i(e^{-\lambda x_i^2})\left(Y_i - \frac{A}{x_i}(1 - e^{-\lambda x_i^2})\right) = 0.$$

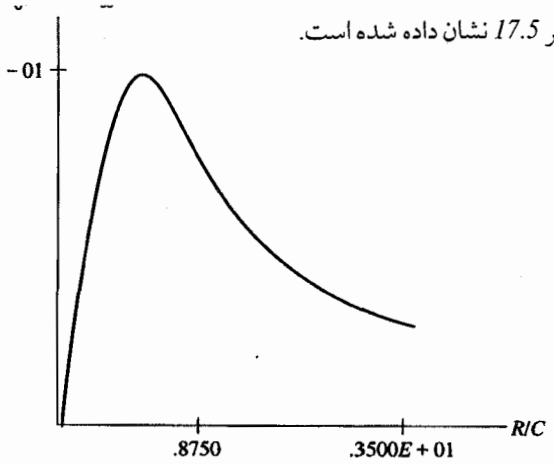
وقتی این دستگاه معادلات غیر خطی حل شوند، برای مقادیر A و λ خواهیم داشت

$$\boxed{g(x) = \frac{0.07618}{x}(1 - e^{-2.30574x^2}).}$$

بطوریکه:

$$S = 0.000016$$
$$A = 0.07618$$
$$\lambda = -2.30574$$

ترسیم این تابع در تصویر *17.5* نشان داده شده است.



شکل (۱۷.۵)

۹.۵ برنامه‌های کامپیوتری

۱.۵ لیست برنامه تفاضل‌های محدود به زبان C

این برنامه برای بدست آوردن چند جمله‌ای درونیاب می‌باشد که از مجموعه نقاط می‌گذرد.

```
/*
   ******************************
   **                          **
   **        DIVDIF.C          **
   **                          **
   ******************************


  APPLIED  NUMERICAL  ANALYSIS

  ------------------------------------------------------------------------

  This program uses the divided differences to get the interpolating
  polynomial that goes through a given set of data points.

    PROCEDURE  ddcoef finds the coefficents of the interpolating
    polynomial.

    FUNCTION  ddvalue uses the coefficients of the previous procedure
    to evaluate the polynomial at a give value: u.
*/

#include  <stdio.h>
#include  <math.h>
   float x[10], y[10],        /* the given data points */
         dd[10];              /* the vector of coefficients for p(x) */
   float u;
   int   i, j, n;
/*
       *** ddcoef ***
*/
ddcoef(x,  y, dd,  n)
  float x[], y[], dd[];
  int   n;

/*
-------------------------------------------------------------
      INPUT:  x,y -  the given data points.
              n   -  the number of data points
      OUTPUT:  dd -  the coefficients of p(x),  i.e.f[,,,]
*/
{
   int    i, j, k;
```

```
  float  temp1,  temp2;


  for ( i = 1; i <=  n; ++i )
     dd[i] = y[i];
  for ( j = 2; j <=  n; ++j )
  {
     temp1  = dd[j-1];
     printf("\n");
     for ( k = j; k <=  n; ++k )
     {
        temp2  = dd[k];
        dd[k] = (dd[k] - temp1)/(x[k]  - x[k-j+1]);
        temp1  = temp2;
     }        /* end of for k loop */
  }           /* end of for j loop */
}             /* end of ddcoef */
/*
     *** ddvalue ***
*/
float ddvalue(u)
  float u;
/*
-------------------------------------------------------------
     INPUT:   u - the x-value
     OUTPUT:  ddvalue - the corresponding y-value, i.e. p(u)
*/

{
  float sum;
  int   i;


  sum = 0.0;
  /* Compute  value by nested multiplication from highest term */
  for ( i = n; i >=  2; --i )
     sum = (sum + dd[i]) * (u - x[i-1]);
  sum = sum + dd[1];
  return(sum);
}       /* end of function  ddvalue */



main()          /* main program */
{
/*
-------------------------------------------------------------
  set up four data points
*/

  n = 5;
  x[1] = 3.2; x[2] = 2.7; x[3] = 1.0;
```

```
   x[4] = 4.8; x[5] = 5.6;
   y[1] = 22.0; y[2] = 17.8; y[3] = 14.2;
   y[4] = 38.3; y[5] = 51.7;

/*
---------------------------------------------------------------
     GENERATE  THE  COEFFICIENTS  FOR  THE  POLYNOMIAL
*/

   ddcoef(x, y, &dd, n);       /* compute coefficients */
   printf("\n");
   printf("The coefficients for the polynomial are:\n\n");
   for ( i = 1; i <= n; ++i )
      printf(" %.5f ", dd[i]);
   printf("\n\n");
   printf("\t\t\t**********\n\n");
   printf("\t\t  U\t\t  P(U)\t\t\n\n");

/*
---------------------------------------------------------------
     SET  UP  TABLE  OF  VALUES  FROM  1  TO  7
*/

   u = 1.0;
   do
    {
      printf(" \t\t%.3f\t\t %.5f\n", u, ddvalue(u));
      u = u + 0.2;
    }
   while ( u < 5.601 );
   printf("\n\n");
   printf("\t\t\t**********\n\n");
}
```

۲.۵ لیست برنامه B-اسپلاین درجه سوم به زبان *PASCAL*

این برنامه آلگوریتم هارینگون را به صورت ساده شده بکار می‌برد. در این برنامه از دستورات گرافیکی استفاده می‌شود.

```
        PROGRAM    cubic_B_spline(Input,Output,f);
(*
    ***********************************
    **                                 **
    **        BSPLINE.PAS              **
    **                                 **
    ***********************************

        APPLIED  NUMERICAL  ANALYSIS
-----------------------------------------------------------------

This program implements the Algorithm of S. Harrington, but in a
more simplified form. It is assumed that one has the graphics
commands, such as:

    Procedure draw_line(x1,y1,  x2,y2 : Integer);
 Procedure plotpOINT(x,y     :  Integer);
    GraphON;
    clearGRAPHICS;
*)

Const
    max_number_of_lines   = 20; (* Each curve segment consists of
20 straight lines *)
    max_number_of_points   = 20;
Type
    matrix  = Array[1..4,1..max_number_of_lines]   Of Real;
    vector  = Array[1..max_number_of_points]    Of Real;
    vector4  = Array[1..4] Of Real;
Var
    blend          : matrix;

    xsm,ysm         (* set of four points to create current
                       segment of B-spline curve          *)
        : vector4;
    ax,ay          (* data points as stored in the array  *)
                  : vector;
    x0,y0          : Real;
    lines_per_section,  i,
    number_of_points           : Integer;

    f  :  TEXT;  (* data file that stores the number of points
and
              the data points (x,y) in screen coordinates.  *)
(*
```

```
    INITIALIZE   THE  BLENDING   FUNCTIONS
-------------------------------------------------------------
This procedure evaluates the B-spline basis functions at the
number_of_points  of u on the interval [0,1].
*)
Procedure set_blending_functions(number_of_lines    : Integer);
Var
    i,j                 : Integer;
    u, u_cube, u_square,
    u_minus1_cube       : Real;
Begin
    For i := 1 To number_of_lines   Do              Begin
      u := i/number_of_lines;
      u_square  := u*u;
      u_cube    := u*u_square;
      u_minus1_cube  :=  (u - 1.0)*Sqr(u  - 1.0);
    blend[1,i] := u_minus1_cube/6.0;
    blend[4,i] := u_cube/6.0;
    blend[3,i] := (-u_cube/2.0  + u_square/2.0  + u/2.0 +
1.0/6.0);
    blend[2,i] := (u_cube/2.0  - u_square  + 2.0/3.0);
                                                End;
End;  (* Initialize  Blending  Functions   *)
(*
    Put_in_Sm
-------------------------------------------------------------
This procedure places a new sample point into the -sm arrays.
*)

Procedure put_in_sm(x,  y : Real);
Begin
    xsm[4] := x;  ysm[4] := y;
End;
(*
    Make_Curve
-------------------------------------------------------------
This procedure fills in a section of the curve.
*)
Procedure make_curve(Var    b : matrix);
Var
    i,j  : Integer;
    x,y  : Real;
Begin
    For j := 1 To lines_per_section     Do              Begin
      x := 0.0;  y := 0.0;
      For i := 1 To 4 Do          Begin
        x := x + xsm[i]*b[i,j];
        y := y + ysm{i]*b[i,j]   End;
      draw_line(Round(x0),  Round(y0),  Round(x),  Round(y)  );
      x0 := x;  y0 := y                             End;
End;  (* make_curve  *)
(*
```

```
    NEXT_SECTION
-------------------------------------------------------------
After a section of the curve has been drawn, we shift the sample
points
so that our blending functions can be applied to the next
section.
*)
Procedure next_section;
Var
    i : Integer;
Begin
    For i := 1 To 3 Do        Begin
      xsm[i] := xsm[i+1];
      ysm[i] := ysm[i+1]     End
End;
(*
    CURVE_ABS_2
-------------------------------------------------------------
This procedure extends the curve by taking a new sample point as
its argument and stores it into the _sm arrays.
*)
Procedure curve_abs_2(x,y  : Real);
Begin
    put_in_sm(x,y);
    make_curve(blend);
    next_section
End;              (* curve_abs_2 *)
(*
    START_B_SPLINE
-------------------------------------------------------------
We require the first four sample points to start the curve.
Procedure start_B_spline(ax,ay  : vector); (* first 4 points *)
Var
    i : Integer;
Begin
    For i := 1 To 3 Do        Begin
      xsm[i] := ax[1];
      ysm[i] := ay[1]       End;
    xsm[4] := ax[2]; ysm[4] := ay[2];
    make_curve(blend);
    next_section
End;              (* start_B_spline *)
(*
    END_B_SPLINE
-------------------------------------------------------------
This procedure terminates the B-spline curve
*)
Procedure end_B_spline;
Begin
    put_in_sm(ax[number_of_points],   ay[number_of_points]);
    make_curve(blend);
    next_section;
```

```
    put_in_sm(ax[number_of_points],   ay[number_of_points]);
    make_curve(blend)
End;                (* end_B_spline  *)
(*
    INITIALIZE
-------------------------------------------------------------
This procedure reads in the control points from a data file.
*)

Procedure initialize;
Var
    i : Integer;
Begin
    WriteLn;  WriteLn;
    WriteLn('  ':20, ' Control Points ');
    WriteLn;
    WriteLn('  X':24, ' Y':5);
    WriteLn;
    lines_per_section  := 10;
    ReadLn(f,  number_of_points);
    For i := 1 To number_of_points  Do       Begin
      ReadLn(f,  ax[i], ay[i]);
      WriteLn(ax[i]:25:0,  ay[i]:7:0)      End
    x0 := ax[1]; y0 := ay[i];
    For i := 1 To number_of_points   Do
                plotPoint(Trunc(ax[i]),  Trunc(ay[i]))
End;
Begin          (* MAIN   *)
    gotoXY(1,1);
    ClrScr;
    GraphON;  clearGRAPHICS;
    Assign(f,  'b:bspline2.dat');
    Reset(f);
    While Not Eof(f) Do                  Begin
      initialize;
(*
    Create a box about the display
*)
    draw_line(0,0,0,389);
    draw_line(0,389,511,389);
    draw_line(511,389,511,0);
    draw_line(511,0,0,0);
    set_blending_functions(lines_per_section);
    start_b_spline(ax,ay);
    For i := 3 To number_of_points  Do
      curve_abs_2(ax[i],  ay[i]);
    end_B_spline;                        End; (* While  *)
    Close(f)
End.
```

۳.۵ لیست برنامه حداقل مربعات به زبان *PASCAL*

این برنامه ضرایب یک چند جمله‌ای درجه m که برازنده یک مجموعه n نقطه از داده ها می‌باشد، محاسبه می‌کند.

```
Program least_squares(INPUT,OUTPUT);
(*
 APPLIED  NUMERICAL  ANALYSIS

-----------------------------------------------------------------
  This program is used in fitting a polynomial to a set of data
points (x[i], y[i]), i = 1..n by computing the coefficients of
the NORMAL  equations For the LEAST  SQUARES  METHOD.
*)
CONST
  MAX_n  = 100;
TYPE
  rhs = Array[1..MAX_n]  OF Real;
  vector = Array[1..MAX_n]  OF Integer;
  matrix = Array[1..10, 1..10] OF Real;
Var
  a : matrix;   (* the matrix of the normal equations  *)
  x,y,          (* the data points *)
  c,            (* coefficients  of the least squares equation  *)
  right_hand_side,
                (* right hand side of equations  *)
  xn  : rhs; (* Array to hold powers of the X values *)
  ipvt : vector; (* needed in ELIM  and SOLVE  *)
  beta,         (* sum of the errors squared *)
  sum : Real;
  ms, mf : 1..9; (* the range of the degree of Polynomials to be
                    computed        0<= ms< mf <=9    *)
  n,            (* the number  of data points *)
  mfPLUS1,  msPLUS1,
  iMINUS1,  iPt,
  icoeff,jcoeff,
  i,j           : Integer;
(*
-----------------------------------------------------------------
  Set up data values as well as ms, mf, and xn-Array.
  Check on the value of mf.
*)
PROCEDURE  initialize;
VAR
  i,j : Integer;
Begin
  For i := 1 TO  10 DO
    For j := 1 TO  10 DO  a[i,j] := 0.0;
  For i := 1 TO MAX_n  DO                    Begin
    x[i] := 0.0; y[i] := 0.0; c[i] := 0.0;
    right_hand_side[i]  := 0.0              End;
  n := 11;  ms := 3;  mf := 6;  mfPLUS1  := mf+1;  msPLUS1  :=
ms+1;
```

```
    x[1] := 0.05; x[2] := 0.11; x[3] := 0.15; x[4] := 0.31; x[5]
  := 0.46;
    x[6] := 0.52; x[7] := 0.70; x[8] := 0.74; x[9] := 0.82; x[10]
  := 0.98;
    x[11] := 1.17;
    y[1] := 0.956; y[2] := 0.890; y[3] := 0.832; y[4] := 0.717;
  y[5] := 0.571;
    y[6] := 0.539; y[7] := 0.378; y[8] := 0.370; y[9] := 0.306;
    y[10] := 0.242; y[11] := 0.104;
    IF mf > n-1 THEN Begin
        mf := n-1;
        WriteLn; WriteLn;
        WriteLn(' DEGREE OF POLYNOMIAL CANNOT EXCEED
N-1.');
        WriteLn(' REQUESTED MAXIMUM DEGREE TOO LARGE
- ');
        WriteLn(' REDUCED TO ', n-1:3); WriteLn End;
    For i := 1 TO n DO xn[i] := 1
  End; (* INITIALIZE *)
  (*
  ------------------------------------------------------------
  Build the normal matrix a and the right hand side which will be
  stored in the vector rhs.
  *)
  PROCEDURE setup_a_AND_rhs;
  VAR
    i,j : Integer;
  Begin
  (*
    Compute the first column of the a_matrix and the right hand
  side of the equations which will be stored in the vector c.
  *)
    For i := 1 TO mfPLUS1 DO Begin
      a[i,j] := 0.0;
      right_hand_side[i] := 0.0;
      For j := 1 TO n DO Begin
        a[i,1] := a[i,1] + xn[j];
        right_hand_side[i] := right_hand_side[i] + y[j]*xn[j];
        xn[j] := xn[j]*x[j] End
                                End;
  (*
  ------------------------------------------------------------
    Compute the last row of a.
  *)
    For i := 2 TO mfPLUS1 DO Begin
      a[mfPLUS1,i] := 0.0;
      For j := 1 TO n DO Begin
        a[mfPLUS1,i] := a[mfPLUS1,i] + xn[j];
        xn[j] := xn[j] * x[j] End
                                End;
  (*
    Now fill in rest of the a-matrix.
```

```
*)
  For j := 2 TO mfPLUS1 DO
    For i := 1 TO mf DO  a[i,j] := a[i+1,j-1]
End;
(*
-----------------------------------------------------------------
  Print out the normal matrix and the right hand side
*)
PROCEDURE  print_out_matrix_and_rhs;
VAR
  i,j  : Integer;
Begin
  For i := 1 TO mfPLUS1 DO          Begin
    For j := 1 TO mfPLUS1 DO
      Write( a[i,j]:6:1);
    WriteLn( '  *', right_hand_side[i]:6:1)  End
End;
(*$I ALIM.PAS  *)

(*
-----------------------------------------------------------------
  For each degree from ms to mf we find the coefficients of the
  least squares polynomial.
*)
PROCEDURE  find_new_polynomial(i  : Integer);
VAR
  j : Integer;
Begin
  For j := 1 TO i DO c[j] := right_hand_side[j];
  solve(a, i, c);
End;
(*
-----------------------------------------------------------------
 Print out the coefficients of the equation of degree i.
*)
PROCEDURE  print_equation(i  : Integer);
VAR
  j : Integer;
Begin
  WriteLn;
  WriteLn(' For DEGREE OF', i-1:2); WriteLn;
  WriteLn('      THE COEFFICIENTS ARE:');
  WriteLn; Write(':3);
  For j := 1 TO i DO
    Write(c[j]:8:3);
  WriteLn; WriteLn
End;
Begin  (* MAIN *)
  initialize;
  setup_a_AND_rhs;
  print_out_matrix_and_rhs;
  decompWITHOUTpivot(a,  mfPLUS1);
```

```
For i := msPLUS1 TO mfPLUS1 DO        Begin
    find_new_polynomial(i);
    print_equation(i)                 End
End.
```

تمرینات فصل پنجم

۱ - چهار نقطهٔ $(1, 0)$ و $(15, 2-)$ و $(9, 2+)$ و $(0, 1-)$ داده شده‌اند، از روش درونیابی لاگرانژ چند جمله‌ای درجه سومی بنویسید. که از آنها می‌گذرد و جملات را در هم ضرب کنید تا نتیجه استاندارد $ax^3 + bx^2 + cx + d$ بدست آید.

جواب:

$$y = p_3(x) = y_0\phi_0(x) + y_1\phi_1(x) + y_2\phi_2(x) + y_3\phi_3(x)$$

$$p_3(x) = -0.5x^3 + 4x^2 + 0.5 - 4.$$

۲ - اگر $e^{0.2}$ بوسیله درونیابی لاگرانژ در میان مقادیر $e^0 = 1$ و $e^{0.1} = 1.1052$ و $e^{0.3} = 1.3499$ تخمین زده شود، خطای تقریبی ماکزیمم و مینیمم را بیابید و با مقدار واقعی خطا مقایسه کنید.

جواب: مقدار تقریبی $e^{0.2} = 1.2218$ که دارای خطای $0.0004+$ می‌باشد. مینیمم خطای تخمینی $0.00033+$ و ماکزیمم آن $0.00045+$ می‌باشد.

۳ - جدول داده‌های زیر را در نظر می‌گیریم.

x	$f(x)$
2	3.4899
5	21.7889
6	31.3585
−1	0.8726
−2	3.4899

جدول نویل *(Neville)* را محاسبه کنید.

چند جمله‌ای‌های درجه دو، سه و چهار را بکار برده مقدار $f(3)$ را بدست آوردید.

۴ - جدول تفاضل محدود داده‌های زیر را بسازید و برای تقریب $f(0.15)$ از آن استفاده کنید:

x	$f(x)$
0.5	1.0025
−0.2	1.3940
0.7	1.0084
0,1	1.1221
0.0	1.1884

a: چند جمله‌ای درجه 2 که از سه نقطهٔ اول می‌گذرد.

b: چند جمله‌ای درجه 2 که از سه نقطهٔ پایانی می‌گذرد.

c: چند جمله‌ای درجه 3 که از چهار نقطهٔ اول می‌گذرد.

d: چند جمله‌ای درجه 3 که از چهار نقطهٔ آخر می‌گذرد.

e: چند جمله‌ای درجه 4 که از پنج نقطهٔ می‌گذرد.

جواب:

x	$f(x)$				
−0.2	1.3940				
		−0.5593			
0.5	1.0025		0.8676		
		−0.2990		−0.3557	
0.1	1.1221		0.5475		0.4892
		−0.1895		−0.2579	
0.7	1.0084		0.6764		
		−0.2571			
0.0	1.1884				

a: 1.09196
b: 1.09402
c: 1.09414
d: 1.09508
e: 1.09579

۵- در تمرین ۴، اگر بخواهیم مقادیر زیر را محاسبه کنیم، کدام سه نقطه برای ساختن چند جملهٔ درجه دو مناسب‌تر است.

a: $f(0.15)$ *b*: $f(-0.1)$ *c*: $f(1.2)$

جواب:

a: $(-0.2, 0.0, 0.1)$
b: $(-0.2, 0.0, 0.1)$
c: $(1.0, 0.5, 0.7)$

۶- یک جدول تفاضل‌های محدود را با داده‌های ذیل کامل کنید:

x	1.20	1.25	1.30	1.35	1.40	1.45	1.50
$f(x)$	0.1823	0.2231	0.2624	0.3001	0.3365	0.3716	0.4055

جواب:

x	$f(x)$						
1.2	0.1823						
		0.0408					
1.25	0.2231		−0.0015				
		0.0393		−0.0001			
1.3	0.2924		−0.0016		0.0004		
		0.0377		0.0003		−0.0007	
1.35	0.3001		−0.0013		−0.0003		−0.0011
		0.0364		0.0000		0.0004	
1.4	0.3365		−0.0013		0.0001		
		0.0351		0.0001			
1.45	0.3716		−0.0012				
		0.0339					
1.5	0.4055						

۷- در تمرین شمارهٔ ۶، چه درجه‌ای از چند جمله‌ای برای برازش دقیق هفت نقطه داده شده مورد نیاز است؟ کدام چند جمله‌ای با درجهٔ کمتر تقریباً برازنده خواهد بود؟ جوابتان را توجیه نمایید.

جواب: چند جمله‌ای درجه شش لازم است ولی یک چند جمله‌ای درجه سه برازنده نقاط است، زیرا که ستون سوم تفاضل‌ها تقریباً ثابت هستند یا به عبارتی ستون چهارم به سمت صفر میل می‌کند.

۸- با استفاده از چند جمله‌ای درجهٔ دو درونیابی نیوتن - گریگوری و جدول داده شدهٔ زیر، تقریب مقدار $f(0.158)$ را (با $x_0 = 0.125$) از چند جمله‌ای درجهٔ سوم تخمین بزنید. خطاهای هر یک را با استفاده از قانون جمله بعدی تخمین بزنید.

x	$f(x)$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
0.125	0.79168				
		−0.01834			
0.250	0.77334		−0.01129		
		−0.02963		0.00134	
0.375	0.74371		−0.00995		0.00038
		−0.03958		0.00172	
0.500	0.70413		−0.00823		0.00028
		−0.04781		0.00200	
0.625	0.65632		−0.00623		
		−0.05404			
0.750	0.60228				

جواب:

$$\frac{x-x_0}{h}=s\ ,\ \frac{x-0.125}{0.125}=s$$

$$s=\frac{0.158-0.125}{0.125}=\frac{0.033}{0.125}=0.264$$

$$p_3(0.158)=0.78801$$

۹- برای بدست آوردن مقدار $f(0.636)$ با انتخاب $x_0 = 0.375$ تمرین ۸ را تکرار کنید.

جواب:

$$s=\frac{x-x_0}{h}=\frac{0.636-6.375}{0.125}=\frac{0.261}{0.125}=2.088$$

$$p_3(0.636)=0.65178$$

۱۰- برای بدست آوردن مقدار y در $x = 0.58$ از داده‌های زیر استفاده کنید، و از یک چند جمله‌ای درجه 3 برای برازش مقادیر x برابر 0.3 و 0.5 و 0.7 و 0.9 استفاده نمایید.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
0.1	0.003			
		0.064		
0.3	0.067		0.017	
		0.081		0.002
0.5	0.148		0.019	
		0.100		0.003
0.7	0.248		0.022	
		0.122		0.004
0.9	0.370		0.026	
		0.148		0.005
1.1	0.518		0.031	
		0.179		
1.3	0.697			

جواب:

$x=0.54$

$y=0.1855$

۱۱ - کمترین درجه چند جمله‌ای که دقیقاً برازنده تمام هفت نقطه داده شده در تمرین ۱۰ می‌باشد، چیست؟

جواب: چون ستون سوم تفاضل‌ها ثابت است. بنابراین چند جمله‌ای درجه سوم مناسب است.

۱۲ - اگر مقادیر x دارای فواصل مساوی باشند، انجام درونیابی معکوس با استفاده از چند جمله‌ای گاوس پیشرو نسبت به استفاده از چند جمله‌ای با تفاضل‌های محدود آسانتر است، زیرا که تفاضل‌ها حول نقطه مرکز درونیابی معکوس باقی می‌مانند، بدون اینکه محدودهٔ چند جمله‌ای‌ها نیاز به تنظیم داشته باشند. با استفاده از تقریبات متوالی که براساس چند جمله‌ای پیشرو گاوس می‌باشد با داده‌های تمرین ۸ مقدار x را برای $f(x) = 0.2852$ بیابید (در واقع، داده‌های این جدول برای $f(x)= Lnx$ می‌باشد، جواب خود را با $e^{0.2852}$ مقایسه نمایید.

جواب:

۱۳ - در تابع $y = x^2$، بدیهی است که نقاط $(1, 1)$ و $(2,4)$ و $(3, 9)$ روی منحنی آن واقع هستند. x را بعنوان تابعی از y در نظر بگیرید $x = \sqrt{y}$ برای $y = 25$ داریم $x = 5$. اما ممکن است که x مربوط $y = 25$ را بوسیله درونیابی معکوس از سه نقطهٔ اول محاسبه نماییم. این عمل را با استفاده از چند جمله‌ای‌های لاگرانژ انجام دهید (البته، برونیابی کنید) چه خطائی را تخمین می‌زنید؟

۱۴ - تابع $f(x) = \frac{20}{1 + 5x^2}$ را در فاصله $[2, -2]$ در نظر بگیرید. منحنی چند جمله‌ای‌های درجه 1 و 2 و 3 و 4 و 5 برای برازش با $f(x)$ در نقاط متساوی الفاصله یکنواخت روی فاصله فوق را، محاسبه و رسم نمایید.

۱۵ - این عبارت را تایید کنید که: برای یک مجموعهٔ داده‌ها که دقیقاً یک چند جمله درجه سه به آن نظیر می‌شود مقادیر S در دو انتها مربوط به مقادیر مجاور S خطی خواهند بود، اگر منحنی اسپلاین و چند جمله‌ای درجه 3، یک تابع باشند. اگر مقدار انتهایی S را تغییر دهیم مانند S_0، آیا این عمل، قسمت‌های دیگر از منحنی اسپلاین را در فواصل غیر از فاصله اول، تغییر می‌دهد؟

۱۶ - ماتریس ضرایب و بردار سمت راست را برای اسپلاین درجه سوم نظیر به داده‌های زیر را بدست آورید. برای مقادیر انتهائی S، شرایط خطی را بکار برید.

x	$f(x)$
0.15	0.3945
0.76	0.2989
0.89	0.2685
1.07	0.2251
1.73	0.0893
2.11	0.0431

جواب:

$$A=\begin{bmatrix} 4.9523 & -2.7323 & 0 & 0 \\ 0.13 & 0.6200 & 0.18 & 0 \\ 0 & 0.18 & 1.6800 & 0.66 \\ 0 & 0 & 0.4412 & 2.6788 \end{bmatrix}$$

بردار سمت راست برابر است با: *(-.46275, -0.04359, 0.21212, 0.50507)*

۱۷ - دستگاه معادلات تمرین ۱۶ را حل کنید (شما ممکن است که برای این کار یک برنامهٔ کامپیوتری را مورد استفاده قرار دهید) و سپس ضرایب چند جمله‌ای‌های درجه سوم مختلف را معین نمایید.

داده‌ها، مربوط به نقاط تابع احتمال نرمال هستند. مقادیر درونیابی در نقاط $x = 0.30, 0.80, 1.50, 2.00$ با مقادیر تابع خطا $ERF(x)$ مقایسه کنید.

جواب:

x	مقدار تحقیقی	مقدار درون یابی
0.3	0.38139	0.38161
0.8	0.28969	0.28972
1.5	0.12952	0.13039
2.0	0.05399	0.05326

۱۸ - یک اسپلاین درجه سوم طبیعی برای $f(x) = \frac{20}{1+5x^2}$ در فاصله $[-2, 2]$ برازش کنید پنج نقطه با فواصل یکسان را روی تابع مورد استفاده قرار دهید. $[x = -2(1)2]$ شکل چند جمله‌ای را کشیده و با منحنی‌های تمرین ۱۴ مقایسه کنید.

جواب: بعضی از مقادیر برابر i با:

x	0.2	0.5	1.3	1.5	1.8	1.9
درون‌یابی	18.76	13.58	0.312	-0.281	0.188	0.557
مقدارتحقیقی	16.67	8.89	2.116	1.632	1.163	1.050

۱۹ - تمرین ۱۸ را تکرار کنید، لیکن از شرایط 2 و 3 در نقاط انتهایی استفاده نمایید. نتایج را با تمرین

۱۸ مقایسه کنید و دوباره با $f'(x_0) = +0.9$ و $f'(x_n) = -0.9$ تکرار نمایید.

جواب:

x	0.2	0.5	1.3	1.5	1.8	1.9
شرط اول	18.82	13.85	−0.548	−1.627	−0.984	−0.167
شرط دوم	18.95	14.35	−2.13	−4.11	−3.142	−1.488
مقداربه‌دست آمده	18.72	13.42	0.809	0.498	0.865	0.965

۲۰ – یک چند جمله‌ای درجه سوم اسپلاین با یک دورهٔ تناوب کامل از داده‌ها دارای مشتق‌های درجه اول و دوم یکسان در دو نقطه انتهایی خواهند بود. معادلاتی را که برای این حالت مقادیر S را می‌دهد، بسط دهید. آیا ماتریس قطری است؟

جواب:

$$s_0 - s_n = 0$$

$$-2h_0 s_0 + h_0 s_1 - 4h_{n-1} s_{n-2} + 3h_{n-1} s_{n-1} = 6\left(f\left[x_{n-2}, x_{n-1}\right] - f\left[x_0, x_1\right]\right)$$

۲۱ – در حقیقت، داده‌های مثال ۷.۵ برای یک پدیده متناوب هستند، اما راه حل‌ها این واقعیّت را ندیده گرفته‌اند. شرایط انتهایی را با یک تابع متناوب بکار برده و نتایج بدست آمده در مثال ۷.۵ را با منحنی اسپلاینی که بدین شکل بدست آمده است، مقایسه کنید.

جواب:

زمان	0.05	0.1	0.45	0.75	0.9	0.95
قدرمطلق روشنایی	0.280	0.253	0.133	0.511	0.386	0.345

۲۲ – نشان دهید که شکل‌های ماتریس معادلات بیزیّر *(Bezier)* و منحنی‌های B – اسپلاین *(B–Spline)*، معادل معادلات جبری داده شده در بخش ۶.۵ می‌باشند.

۲۳ – شکل‌های ماتریسی معادلات برای منحنی‌های بیزیّر *(Bezier)* و منحنی‌های B – اسپلاین *(B–Spline)* درجه چهارم را بنویسید.

جواب:

$$(u^4, u^3, u^2, u, 1)\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}(p_0, p_1, p_2, p_3, p_4)$$

B – اسپلاین

$$(u^4, u^3, u^2, u, 1)\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -6 & -6 & 6 & 0 \\ -4 & -12 & 12 & 4 & 0 \\ 1 & 11 & 11 & 1 & 0 \end{bmatrix}(p_0, p_1, p_2, p_3, p_4)$$

۲۴ – ثابت کنید که چند ضلعی محدب تمام نقاط منحنی‌های بیزیّر *(Bezier)* و منحنی‌های B – اسپلاین *(B–Spline)* را احاطه می‌کند.

راهنمایی: از این حقیقت استفاده کنید که هر نقطه p واقع در چند ضلعی محدب که بوسیلهٔ نقاط $\{p_0, p_1, \ldots, p_n\}$ تشکیل می‌شود می‌تواند بصورت $p = \sum_{i=0}^{n} \alpha_i p_i$ نوشته شود، بطوری که $\alpha_i \geq 0$ و $\sum_{i=0}^{n} \alpha_i = 1$.

۲۵ – آیا شیب در انتهای منحنی درجه سوم B – اسپلاین برابر شیب بین نقاط همجوار می‌باشد. آیا این مطلب برای منحنی B – اسپلاین درجات بالاتر هم صادق است؟

جواب:

$dy/ds = (dy/du)/(dx/du)$

$u = 0$, $dy/dx = (y_{i+1} - y_{i-1})/(x_{i+1} - x_{i-1})$ برای p_1.

$u = 1$, $dy/dx = (y_{i+2} - y_i)/(x_{i+2} - x_i)$ برای p_{i+1}.

۲۶ – یک مجموعهٔ نقاط برای ساختن منحنی‌های B – اسپلاین متصل بهم مورد استفاده قرار گرفته‌اند. یکی از نقاط تغییر داده می‌شود و منحنی‌ها دوباره محاسبه می‌گردند، کدام قسمت از منحنی‌ها تحت تأثیر قرار می‌گیرد؟ اگر منحنی‌های متصل بهم بی‌زیّر بودند، آیا همان نوع تأثیر مشاهده می‌گردید؟ اگر مجموعهٔ نقاط از منحنی‌های درجهٔ سوم اسپلاین بودند، چه تأثیری مشاهده می‌شد؟ آیا جمله کنترل محلی و جمله کنترل ناحیه‌ای در رابطه با پدیده‌ای که شما مشاهده می‌کنید، بکار می‌رود؟

۲۷ – استفاده از درجات بالاتر منحنی‌های B– اسپلاین یک امر طبیعی است، تقلیل درجه برای بدست آوردن یک B – اسپلاین درجهٔ سوم چطور؟ چه فرضیّاتی برای کاهش درجه، معقول هستند؟

۲۸ – منحنی‌های درجه سوم بی زیّر مربوط به دسته نقاط جدول زیر را محاسبه و رسم کنید:

نقطه	x	$f(x)$
1	100	100
2	50	150
3	200	150
4	50	50
5	100	200
6	50	100
7	200	100
8	50	50
9	100	200
10	100	100

جواب:

x	92.2	112.5	114.0	73.4	87.5	120.3	121.1	93.8
y	127.3	131.2	107.0	121.1	131.2	113.3	93.0	118.8

۲۹- تمرین ۲۸ را برای منحنی‌های درجه سوم B- اسپلاین تکرار نمایید.

جواب:

x	83.3	122.9	83.3	83.3	77.1	122.9	83.3	91.7
y	141.7	146.9	91.7	158.3	146.9	101.0	83.3	158.3

۳۰- در ۷.۵، اثبات گردید که رعایت ترتیب در درونیابی تأثیری نخواهد داشت، این حقیقت را با انجام درونیابی برای داده‌های جدول ۱۲.۵ بمنظور یافتن مقادیر در $y = 0.33$ نشان دهید. (در سطری با ثابت x در 1 و 1.5 و 2)، فرمول‌های درونیابی درجهٔ سوم جهت برازش $y = 0.2, 0.3, 0.4, 0.5$ مقادیر جدول را مورد استفاده قرار دهید. سپس بین این سه مقدار درونیابی را انجام دهید و $f(1.6, 0.33)$ را تعیین کرده و آنرا با مقدار بدست آمده در 1.841 مقایسه نمایید.

جواب: مقدار 1.8341 به دست می‌آید. برای $x=1.6$ مقادیر میانی برابرند با:

y	0.2	0.3	0.4	0.5
درون یابی	1.08736	1.66888	2.23572	2.78396

۳۱- بعد از اتمام مثال ۹.۵ مشاهده شد که مقادیر درجه سوم x و درجهٔ دوم y مناسب می‌باشند. $f(1.6, 0.33)$ را با انجام این عمل بدست آورده و با مقدار دقیق 1.8350 مقایسه نمایید. بهترین ناحیه برازش را استفاده کنید.

جواب: مقدار $z=1.8328$ به دست می‌آید. مقادیر میانی برای $x=1.6$ برابر است با:

y	0.2	0.3	0.4
درون یابی	1.08268	1.66193	2.23647

۳۲ – در مثال ۹.۵ از یک ناحیه مستطیلی برازش استفاده شده است در حالی که ناحیه دایره‌ای مناسب‌تر است. مقدار $f(1.62,\ 0.31)$ را با انجام درونیابی داده‌های جدول ۱۲.۵ بدست آورید، در این رابطه از چند جمله‌ای‌هایی که در جدول بازاء 1.6 و 1.5 $= x$ وقتی که 0.2 $= y$ و 0.4 $= y$ می‌باشد و همچنین در 2.5 $(0.5)\ x=0.5$ زمانی که 0.3 $= y$ است، استفاده نمایید این کار را با تشکیل یک سری جداول تفاضل محدود انجام دهید. در این مثال اگر در ابتدا x را ثابت گرفته درونیابی کنیم، انجام محاسبه مشکل است، لیکن شروع با y ثابت هیچگونه اشکالی ندارد.

جواب: $z=1.7524$ (مقدار تحقیقی $z=1.7515$) و مقادیر میانی برای $x=1.6$ برابر است با:

$$1.12872,\ 1.69268,\ 2.31668$$

۳۳ – برای محاسبه $f(3.55,\ 0.53)$ از جدول داده‌های زیر با استفاده از چند جمله‌ای درجه سوم در دو جهت درونیابی نمایید.

x \ y	0.1	0.4	0.6	0.9	1.2
1.1	1.100	0.864	0.756	0.637	0.550
3.0	8.182	6.429	5.625	4.737	4.091
3.7	12.445	9.779	8.556	7.205	6.223
5.2	24.582	19.314	16.900	14.232	12.291
6.5	38.409	30.179	26.406	22.237	19.205

جواب: $f=8.9106$. مقادیر میانی برای $x=3.55$:

y	0.1	0.4	0.6	0.9
درون یابی	11.4564	9.0022	8.8572	6.6327

۳۴ – مقدار تابع را بازاء 3.7 $= x$ و 0.6 $= y$ روی سطح B– اسپلاین از 16 نقطه در گوشهٔ سمت چپ بالایی جدول داده‌های تمرین ۳۳ تشکیل شده است، بدست آورید.

جواب: در $u=0.8736$ و $v=0.9325$ و $x=3.70$ و $y=0.60$ و $f=8.8535$

۳۵ – $S.H.P.Chen$ و $S.C.Saxean$ داده‌های تجربی خود را در مورد تشنج تنگستن برحسب تابعی از دما گزارش نمودند. این داده‌ها در جدول زیر داده شده‌اند. آنها دریافتند که معادله $e(T) = 0.02424\left(\frac{T}{303.16}\right)^{1.27591}$ داده‌ها را تا سه رقم با معنی صحیح صدق می‌کنند. تعیین کنید چند جمله‌ای درونیاب برای برقراری رابطه بین نقاط میانی دماهای مندرج در جدول از چه درجه‌ای خواهد بود؟

$T,°K$	300	400	500	600	700	800	900	1000	1100
e	0.024	0.035	0.046	0.058	0.067	0.083	0.097	0.111	0.125

$T,°K$	1200	1300	1400	1500	1600	1700	1800	1900	2000
e	0.140	0.155	0.170	0.186	0.202	0.219	0.235	0.252	0.269

جواب: برای $T=700$ جواب از فرمول برابر 0.0705 می‌باشد و در جدول برابر 0.067 می‌باشد.

۳۶- در مطالعات پولیمریزاسیون تشعشع القایی، یک منبع اشعهٔ گاما برای اندازه‌گیری مقدار تشعشع بکار گرفته شده است. مقدار انرژی با توجه به موقعیتش در دستگاه، تغییر می‌کند، و این اعداد ثبت می‌شوند.

وضعیت از مبدا	0	0.5	1.0	1.5	2.0	3.0	3.5	4.0
مقدار تشعشع برحسب 10^5 rads/hr	1.90	2.39	2.71	2.98	3.20	3.20	2.98	2.74

با استفاده از درونیابی چند جمله‌ای‌ها با درجات مختلف، برای داده‌ها اطلاعات ثبت نشده را بدست آورید. فکر می‌کنید، بهترین تخمین برای مقدار انرژی در فاصله 2.5 اینچ چیست؟

جواب: در فاصله 2.5 اینچ مقدار تشعشع برابر 3.27 می‌باشد. این جواب برای چند جمله‌ای درجه دوم و سوم می‌باشد.

۳۷- *M.S.Selim* و *R.C.Seagraves* تبادل یونی رزین‌ها را در نظریهٔ جنبشی ترکیبات مس مورد مطالعه قرار دادند، نرمالیته مایع شستشو مهمترین عامل تعیین انتشار و توزیع آن بوده است. داده‌ها از مقادیر مناسب نرمالیته حاصل شده است، جدولی از مقادیر صحیح نرمالیته N مورد نظر است $(N = 0.1,2,3,4,5)$. داده‌های زیر را بکار برده و مقادیر D را بدست آورده و در یک جدول بنویسید.

N	$D \times 10^6$, cm²/sec	N	$D \times 10^6$, cm²/sectr
0.0521	1.65	0.9863	3.12
0.1028	2.10	1.9739	3.06
0.2036	2.27	2.443	2.92
0.4946	2.76	5.06	2.07

جواب:

N	0	1	2	3	4	5
D	0.93	3.12	3.05	2.75	2.43	2.09

۳۸- وقتی معادله جریان گرما در حالت ایستا (ثابت) بصورت عددی حل می‌شود، دماهای $u(x,y)$ در گره‌های یک شبکهٔ که در میدان مورد نظر است بدست می‌آیند. یک مسأله معینی حل شده

است و مقادیر جدول زیر بدست آمده است: این روش دماهای مربوط به سایر نقاط شبکه را نمی‌دهد، اگر دمای این نقاط مورد درخواست باشند، می‌توانیم از طریق درونیابی آنها را بدست آوریم. با استفاده از داده‌ها، مقادیر دما را در نقاط *(1.2, 0.7)* و *(2.4, 1.6)* و *(0.82, 0.65)* تخمین بزنید.

x \ y	0.0	0.5	1.0	1.5	2.0	2.5
0.0	0.0	5.00	10.00	15.00	20.00	25.00
0.5	5.00	7.51	10.05	12.70	15.67	20.00
1.0	10.00	10.00	10.00	10.00	10.00	10.00
1.5	15.00	12.51	9.95	7.32	4.33	0.0
2.0	20.00	15.00	10.00	5.00	0.00	−5.00

جواب:

$u(0.7, 1.2) = 10.68$
$u(1.6, 2.4) = 0$
$u(0.65, 0.82) = 9.40$

از چند جمله‌ای درجه دوم در دو جهت استفاده شده است.

۳۹ - ستاره S در عمق بزرگ یک تغییرات منظمی در قدر مطلق روشنایی خود دارد. *Lean* و *Campbell* و *Laizi Jacchia* داده‌های مربوط به میانگین روشنایی این ستاره را در کتابشان بنام داستان ستاره‌ها ارائه داده‌اند. *(Blackston, 1944)*، بخشی از این داده‌ها در اینجا داده شده است:

فاز	−110	−80	−40	−10	30	80	110
قدر مطلق	7.98	8.95	10.71	11.70	10.01	8.23	7.86

داده‌ها متناوب هستند و برای فاز *120* − همانگونه هست که برای فاز *120*+ می‌باشد. توابع اسپلاین که در بخش *6.5* مورد بحث واقع گردید، برای رفتارهای متناوب مجاز نیستند، مشتق اول و دوم یک تابع متناوب در نقاط انتهایی یکسان هستند. با در نظر گرفتن این نکته، یک تابع اسپلاین را از طریق درونیابی برای داده‌های فوق بدست آورید. اطلاعات دیگری برای همان ستاره توسط جک چیا و کمپ بل داده شده است:

Phase	−100	−60	−20	20	60	100
Magnitude	8.37	9.40	11.39	10.84	8.53	7.89

قدر مطلق

به نظر شما اطلاعات جدید درونیابی بر اساس تابع اسپلاین چقدر مناسب تر مشاهدات را نمایش می‌دهد؟

جواب:

با اسپلاین درجه سوم با قدر مطلق *7.92* در فاز ۱۲۰ به دست می‌آوریم:

فاز	-100	-60	-20	20	60	100
تخمین قدر مطلق	8.23	9.79	11.56	10.58	8.74	7.93
خطا	0.14	-0.39	-0.17	0.26	-0.21	-0.04

۴۰ - ماتریس‌های ساخته شده معادله (26.5) نقاط روی سطح بی‌زیّر را تولید می‌کنند. نشان دهید که این از 12 نقطه روی حاشیه‌های گروه 16 نقطهٔ شکل (۱۰.۵) می‌گذرد. چطور می‌توان یک سطح بی‌زیّر که از مجموعهٔ چهار نقطه داخلی می‌گذرد بوجود آورد؟

۴۱ - یک آزمایش شیمیایی هفت نقطه داده تولید کرده است.

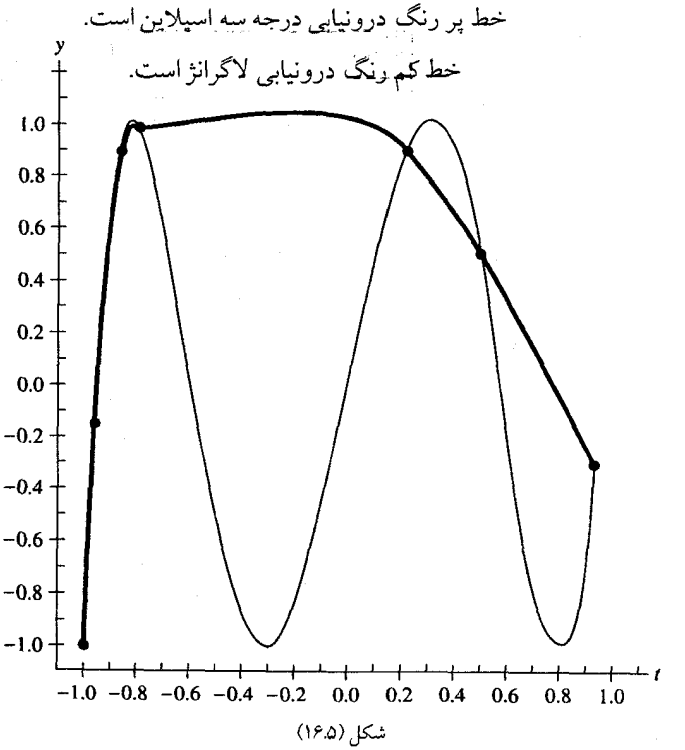
t	-1	-0.96	-0.86	-0.79	0.22	0.5	0.930
y	-1	-0.151	0.894	0.986	0.895	0.5	-0.306

a) - نقاط را رسم کنید و یک خم هموار با درونیابی به آن نظیر کنید.

b) - یک چند جمله‌ای درجه شش درونیابی به این نقاط رسم کنید.

c) - برای رسم این خم با یک برنامه اسپلاین چه تعداد نقطه کافی است.

d) - نتایج بدست آمده را با شکل (۱۶.۵) مقایسه کنید.



شکل (۱۶.۵)

۴۲ - فرض می‌کنیم z تابعی از دو متغیر x و y است یک معادله نرمال که برازنده داده‌های زیر باشد بصورت $z = ax + by + c$ بدست آورید. و از روش حداقل مربعات استفاده کنید.

x	0	1.2	2.1	3.4	4.0	4.2	5.6	5.8	6.9
y	0	0.5	6.0	0.5	5.1	3.2	1.3	7.4	10.2
z	1.2	3.4	-4.6	9.9	2.4	7.2	14.3	3.5	1.3

جواب: $z=2.8530x-1.9145y+1.0399$

۴۳ - یک ترکیب پتانسیل شیمیایی در سه درجهٔ حرارت با دو غلظت مختلف اندازه‌گیری شده و جدول زیر بدست آمده است.

$t°C$ درجه حرارت	-10.	-10.	0.	0.	10.	10.
x غلظت (نسبت به 50%)	-.1	.1	-1	.1	-.1	.1
y پتانسیل شیمیایی	38.	42.	41.	46.	46	49

در صورتیکه بدانیم t و x و y در معادلهٔ $y = a + bt + cx$ صدق می‌نمایند، از روش حداقل مربعات بهترین مقادیر را برای ضرایب a و b و c بدست آورید. از این معادله پتانسیل شیمیایی را در $20°C$ و با غلظت 0.2 تخمین بزنید.

۴۴ - با رسم نقاط داده شده در جدول زیر، به نظر می‌رسد که خم $y = ae^{bx}$ برازنده آن باشد. با روش حداقل مربعات و انتخاب رابطهٔ $Ln\ y = Ln\ a + bx$ مقادیر ثابت a و b را تعیین نمایید.

x	77	100	185	239	285
y	2.4	3.4	7	11.1	19.6

جواب $y = 1.2019\ e^{(9.6027*10^{-3})x}$ یا

$Ln\ y = 0.18395 + 9.6027\times10^{-3}\ x$

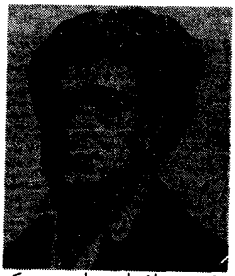
فصل ۶

مشتق‌گیری عددی

موضوعات این فصل

* محاسبه مشتقات
* مشتقات از چند جمله‌ای‌های درونیاب (جدول تفاضل‌ها)
* مشتقات مراتب بالاتر.
* نمودار لوزی برای مشتقات
* فنون برونیابی
* خطای گرد کردن و دقت مشتقات
* برنامهٔ محاسبه مشتقات از جدول داده‌های متساوی الفاصله به نام DER.BAS به زبان BASIC
* تمرینات فصل ششم

گوتفرید ویلهلم لایبنیتز
Leibniz
(۱۷۱۶ - ۱۶۴۶)



لایبنیتز، در سال ۱۶۴۶ در شهر لایپزیگ آلمان به دنیا آمد. وی بزرگترین نابغهٔ جامع قرن هفدهم و رقیب نیوتن در ابداع حسابان می‌باشد. او در جوانی به خلق اولین ایده‌های خصیصه‌های کلی پرداخت که متضمن ریاضیات جامعی بود که بعدها در سال ۱۹۱۰، در قالب اثر عظیم اصول ریاضیات وایتهد و راسل در آمد.

لایبنیتز حسابان خود را زمانی بین سالهای ۱۶۷۳ و ۱۶۷۶ اختراع کرد. در ۲۹ اکتبر ۱۶۷۵ بود که برای اولین بار علامت امروزی انتگرال را، به صورت $S(\int)$ کشیده‌ای که از اولین حرف کلمهٔ لاتین سوما (مجموع Summa) گرفته شده، برای نشان دادن مجموع تقسیم ناپذیرهای کاوالیری به کار برد. چند هفته بعد او دیفرانسیل‌ها و مشتق‌ها و همچنین انتگرال‌هائی نظیر $\int ydx$ را بصورت نوشت، که ما امروزه می‌نویسیم.

بسیاری از قواعد مقدماتی مشتقگیری را که یک دانشجو در اوایل یک درس مقدماتی در حسابان می‌آموزد، لایبنیتز استخراج کرده است. قاعده مشتق nام حاصل ضرب دو تابع هنوز هم قاعده لایبنیتز نامیده می‌شود. در مورد استعداد یگانه او می‌توان گفت: اندیشه ریاضی را دو حوزهٔ وسیع و متضاد است، ریاضیات پیوسته و ریاضیات گسسته. لایبنیتز تنها شخصی در تاریخ ریاضیات است که هر دو جنبهٔ اندیشه را در حدی عالی دارا بود.

۱.۶ محاسبه مشتقات

در حقیقت امروزه، ما برنامه‌های کامپیوتری را برای درونیابی به روش چند جمله‌ای یا به روش اسپلاین مرتبه سوم در اختیار داریم. این مساله را تعمیم می‌دهیم. فرض کنید حرکت راکت را بررسی می‌کنیم. ما مشتق تغییر مکان نسبت به زمان را نیاز داریم که البته سرعت است و نیز محاسباتی که منجر به مصرف سوخت می‌گردند، در اینصورت ضروری است که انتگرال تابعی را بگیریم در حالی که مقادیری از تابع را فقط در زمان‌های گسسته می‌شناسیم.

اگر یک چند جمله‌ای داشته باشیم که موقعیت راکت را بعنوان تابعی از زمان در اختیار ما قرار دهد، مسلماً می‌توانیم مشتق یا انتگرال آن چند جمله‌ای را بدست آوریم.

مسئله این است که ما هرگز بطور واقعی این چند جمله‌ای را نداریم تا از مقادیر و مشتقاتش استفاده کنیم. تخمین خطای مشتقات و انتگرالها، از مقادیر تابع که در نقاط گسسته برحسب زمان شناخته شده‌اند، به مهارت خاص نیاز دارد.

روشهایی برای بدست آوردن مشتقات و انتگرالها از جدول مقادیر تابع وجود دارد که شبیه به روش‌های درونیابی می‌باشند. در این فصل روش‌های محاسبه مشتقات دنبال خواهند شد.

۲.۶ مشتقات از چند جمله‌ای‌های درونیاب

اگر تابعی توسط چند جمله‌ای درونیاب، خوب برآورد شده باشد، توقع داریم که شیب تابع توسط شیب چند جمله‌ای تقریب گردد، اگرچه خواهیم دید که خطای تخمین شیب بزرگتر از خطای تخمین تابع می‌باشد.

ما این بخش را با چند جمله‌ای تفاضل محدود که برای تخمین چند جمله‌ای می‌باشد، آغاز می‌کنیم.

$$\begin{aligned} f(x) &= P_n(x) + \\ &= f[x_0] + f[x_0, x_1](x - x_0) \\ &\quad + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &\quad + \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1) \ldots (x - x_{n-1}) \\ &\quad + \text{خطا} \end{aligned} \tag{1.6}$$

همانطور که در فصل قبل مشاهده کردیم جمله خطای معادله *(6.1)* برابر است با:

$$\text{خطا} = P_n(x) = (x - x_0) \ldots (x - x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}, \qquad x_0 < \xi < x_n. \tag{2.6}$$

مشتق معادله *(1.6)* نتیجه می‌دهد:

$$\begin{aligned} f'(x) = P_n'(x) &= f[x_0, x_1] + f[x_0, x_1, x_2](2x - x_0 - x_1) \\ &\quad + \cdots + f[x_0, x_1, \ldots, x_n] \sum_{k=0}^{n-1} \frac{(x - x_0) \ldots (x - x_{n-1})}{(x - x_k)}. \end{aligned} \tag{3.6}$$

توجه داریم که:

$$[(x - x_0)(x - x_1) \ldots (x - x_{n-1})]' = \sum_{k=0}^{n-1} \frac{(x - x_0) \ldots (x - x_{n-1})}{(x - x_k)}.$$

این معادلات ساده خواهند شداگر مقدار آنها را در $x=x_0$ بدست آوریم. معادله *(3.6)* تبدیل می‌شود به:

$$f'(x_0) = f[x_0, x_1] + f[x_0, x_1, x_2](x_0 - x_1) + \cdots + f[x_0, x_1, \ldots, x_n](x_0 - x_1)(x_0 - x_2) \ldots (x_0 - x_{n-1}). \quad (4.6)$$

مشتق گیری از معادله *(2.6)* خطای معادله *(4.6)* را خواهد داد:

$$\text{خطا} \qquad P'(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \sum_{k=0}^{n} \frac{(x - x_0) \ldots (x - x_n)}{(x - x_k)} + \frac{(x - x_0) \ldots (x - x_n)}{(n+1)!} \frac{d}{dx}[f^{(n+1)}(\xi)] \quad (5.6)$$

اگر چه ξ وابسته به x می‌باشد، لیکن اگر $x=x_0$ را در معادله *(5.6)* قرار دهیم خواهیم داشت:

$$\text{خطا} \qquad P'(x_0) = (x_0 - x_1) \ldots (x_0 - x_n) \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

چون اکثر الگوریتم‌های فصل 6 و 5 برفرض نقاط هم فاصله، استوار می‌باشند، ما از بسط چند جمله‌ای‌های پیشرو نیوتن- گریگوری استفاده می‌کنیم و مشتقات باقیمانده چند جمله‌ای های تفاضل محدود را به عنوان تمرین باقی می‌گذاریم.

با چند جمله‌ای پیشرو نیوتن- گریگوری آغاز می‌کنیم:

$$f(x_s) = P_n(x_s) + \text{error} = f_0 + s\Delta f_0 + \binom{s}{2}\Delta^2 f_0 + \cdots + \binom{s}{n}\Delta^n f_0 + \text{خطا}, \quad (6.6)$$

خطای معادله *(6.6)*، با قاعدهٔ تغییر جمله بعدی که در اختیار داریم، تعیین می‌شود

$$\text{خطا} \qquad P_n(x_s) = \binom{s}{n+1} h^{n+1} f^{(n+1)}(\xi), \qquad x_0 < \xi < x_n. \quad (7.6)$$

در مشتق‌گیری از معادله *(6.6)*، یادآوری می‌کنیم که f_0 و تمام جملات Δ ثابت می‌باشند (در اصل، آنها اعدادی از جدول تفاضل‌ها می‌باشند) داریم:

$$f'(x_s) \doteq P_n'(x_s) = \frac{d}{dx}[P_n(x_s)] = \frac{d}{ds}[P_n(x_s)]\frac{ds}{dx} = \frac{d}{ds}[P_n(x_s)]\frac{1}{h}$$
$$= \frac{1}{h}\left(\Delta f_0 + \frac{1}{2}(s - 1 + s)\Delta^2 f_0 + \frac{1}{6}[(s-1)(s-2) + s(s-2) + s(s-1)]\Delta^3 f_0 + \cdots\right). \quad (8.6)$$

مشتقات فاکتوریل‌های s به سرعت از نظر جبری پیچیده می‌شوند. اگر $s=0$ قرار دهیم به اندازه قابل

توجهی مشتق متناظر به x_0ساده می‌شود. بهرصورت اگر $s=0$ قراردهیم، معادله *(8.6)* تبدیل شودبه:

$$f'(x_0) \doteq \frac{1}{h}\left(\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \frac{1}{4}\Delta^4 f_0 + \cdots \pm \frac{1}{n}\Delta^n f_0\right). \qquad (9.6)$$

در معادله *(9.6)* مقدار مشتق، به مشتق تابع تقریب زده می‌شود و برابر با مشتق یک چند جمله‌ای درجه nام می‌باشد که از نقطه (x_0, f_0) و n نقطه دیگر عبور می‌کند و مقدارش در $x=x_0$ سنجیده می‌شود. خطای معادله *(9.6)*، با مشتق‌گیری جمله خطا در معادله *(7.6)*، بدست می‌آید:

$$\text{خطای } P_n'(x_s) = h^{n+1} f^{(n+1)}(\xi)\left[\frac{d}{ds}\binom{s}{n+1}\right]\frac{1}{h} + \binom{s}{n+1} h^{n+1}\frac{d}{dx}[f^{(n+1)}(\xi)]. \qquad (10.6)$$

مقدار جمله دوم را نمی‌توان بدست آورد، برای اینکه تغییرات ξ نسبت به x مشخص نیست، اما وقتی $s = 0$ قرار دهیم، جمله دوم حذف می‌گردد زیرا که:

$$\binom{s}{n+1} = \frac{1}{(n+1)!}(s)(s-1)\ldots(s-n) = 0$$

در $s = 0$، فقط به اولین جمله معادله *(5.6)* نیاز است. از اینرو:

$$\frac{d}{ds}\binom{s}{n+1} =$$

$$\frac{(s-1)(s-2)\ldots(s-n) + s(s-2)(s-3)\ldots(s-n) + \cdots + s(s-1)\ldots(s-n+1)}{(n+1)!},$$

در $s = 0$ تنها اولین جمله صورت باقی ماند و معادله *(10.6)* تبدیل می‌شود به:

$$\text{خطای } P_n'(x_0) = h^{n+1} f^{(n+1)}(\xi)\left[(-1)^n \frac{n!}{(n+1)!}\right]\left(\frac{1}{h}\right) \qquad (11.6)$$

$$= \frac{(-1)^n}{n+1} h^n f^{(n+1)}(\xi).$$

توجه کنید، اگرچه چند جمله‌ای درونیاب مقدار تابع را بطور دقیق در $s=0$ می‌دهد، اما فرمول مشتق دارای خطای $O(h^n)$ در آن نقطه می‌باشد، مگر اینکه $f^{(n+1)}(\xi)=0$ در مقایسه معادلات *(11.6)* و *(9.6)* می‌بینیم که خطای مشتق همچنین می‌تواند با تغییر $\Delta^n f_0$ در جمله بعد از آخرین جمله بدست آورده شود و به $h^n f^{(n)}(\xi)$ تبدیل می‌شود. (دقیقاً مانند فرمول‌های درونیابی.)

مثال ۱.۶ داده‌های جدول (۱.۶) را مورد استفاده قرار دهید و مشتق y در $x = 1.7$ را تخمین بزنید، و از $h = 0.2$ محاسبات را با یک و دو و سه یا چهار جمله از فرمول انجام دهید.

با یک جمله $$y'(1.7) = \frac{1}{0.2}(1.212) = 6.060.$$

با دو جمله $$y'(1.7) = \frac{1}{0.2}\left(1.212 - \frac{1}{2}(0.268)\right) = 5.390.$$

به سه جمله $$y'(1.7) = \frac{1}{0.2}\left(1.212 - \frac{1}{2}(0.268) + \frac{1}{3}(0.060)\right) = 5.490.$$

با چهار جمله $$y'(1.7) = \frac{1}{0.2}\left(1.212 - \frac{1}{2}(0.268) + \frac{1}{3}(0.060) - \frac{1}{4}(0.012)\right)$$

$$= 5.475.$$

جدول ۱.۶

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1.3	3.669				
		0.813			
1.5	4.482		0.179		
		0.992		0.041	
1.7	5.474		0.220		0.007
		1.212		0.048	
1.9	6.686		0.268		0.012
		1.480		0.060	
2.1	8.166		0.328		0.012
		1.808		0.072	
2.3	9.974		0.400		
		2.208			
2.5	12.182				

داده‌ها در جدول ۱.۶ مربوط به $y=e^x$ می‌باشند که تا سه رقم اعشار گرد شده‌اند. چون مشتق e^x همان e^x می‌باشد، می‌بینیم که خطای مشتق با چهار جمله در مقایسه با 5.475 و 5.474 کاملا کوچک است. این موضوع قابل پیش‌بینی بود، از آنجا که تفاضل چهارم در جدول ۱.۶ زیاد بزرگ نیست، تابع بطور مناسب با یک چند جمله‌ای درجه 4 نشان داده می‌شود.

می‌دانیم $f(x)=e^x$. خطاهای برآورد شده برای محاسبات قبل می‌توانند توسط معادله *(11.6)* محاسبه شوند:

خطا با یک جمله

$$= \frac{(-1)^1}{2}(0.2)f''(\xi), \qquad 1.7 \le \xi \le 1.9,$$

$$= \frac{-0.2}{2}\begin{Bmatrix} e^{1.7} & (\min) \\ e^{1.9} & (\max) \end{Bmatrix} = \begin{Bmatrix} -0.547 & (\min) \\ -0.669 & (\max) \end{Bmatrix}.$$

(خطای واقعی برابر است با: -0.586)

خطا با دو جمله

$$= \frac{(-1)^2}{3}(0.2)^2 f'''(\xi), \qquad 1.7 \le \xi \le 2.1,$$

$$= \frac{0.04}{3}\begin{Bmatrix} e^{1.7} & (\min) \\ e^{2.1} & (\max) \end{Bmatrix} = \begin{Bmatrix} 0.073 & (\min) \\ 0.109 & (\max) \end{Bmatrix}.$$

(خطای واقعی برابر است با: 0.084)

خطا با سه جمله

$$= \frac{(-1)^3}{4}(0.2)^3 f^{iv}(\xi), \qquad 1.7 \le \xi \le 2.3,$$

$$= \frac{-0.008}{4}\begin{Bmatrix} e^{1.7} & (\min) \\ e^{2.3} & (\max) \end{Bmatrix} = \begin{Bmatrix} -0.011 & (\min) \\ -0.020 & (\max) \end{Bmatrix}.$$

(خطای واقعی برابر است با: -0.016)

خطا: با چهار جمله $$= \frac{(-1)^4}{5}(0.2)^4 f^{v}(\xi), \qquad 1.7 \le \xi \le 2.5,$$

$$\approx \frac{0.0016}{5}\begin{Bmatrix} e^{1.7} & (\min) \\ e^{2.5} & (\max) \end{Bmatrix} = \begin{Bmatrix} 0.002 & (\min) \\ 0.004 & (\max) \end{Bmatrix}.$$

(خطای واقعی برابر -0.001)

در حالت چهار جمله، خطاهای واقعی محاسبه شده به علت گرد کردن تطبیق نمی‌کنند. معادله *(11.6)* فقط برای خطای برش به حساب می‌آید. در موارد دیگر، خطای گرد کردن نسبت به خطای برش به حد کافی بزرگ نیست که تخمین‌های معادله *(11.6)* را تحت تأثیر قرار دهد.

فرمول‌های مشتقات ارائه شده توسط معادله *(9.6)* تقریب‌های تفاضل پیشرو نامیده می‌شوند، زیرا آنها تنها تفاضل‌های مقادیر تابع جلوتر از $f(x_0)$ را شامل می‌شوند. برای تقریب تفاضل پیشرو، چند جمله‌ای درونیابی که استفاده می‌شود برای نقاط قرینه نسبت به x_0 مناسب نیست. در فصل قبل دیدیم که درونیابی برای نقاط نزدیک به مرکز محدوده نظیر، خیلی دقیقتر است. این موضوع در مورد محاسبات مشتقات عددی نیز صحیح است. اولین دوجمله معادله *(8.6)* را ملاحظه کنید:

$$f'(x) = \frac{1}{h}\left(\Delta f_0 + \frac{1}{2}(s - 1 + s)\Delta^2 f_0\right) \quad + \text{خطا} \qquad (12.6)$$

چند جمله‌ای درجه دوم درونیاب، با معادله *(12.6)* متناظر با عبور از نقاط x_0 و x_1 و x_2 می‌باشد. اگر $s = 1$ را در معادله *(12.6)* قرار دهیم، می‌توانیم $f'(x)$ را برآورد کنیم:

$$f'(x_1) = \frac{1}{h}\left(\Delta f_0 + \frac{1}{2}\Delta^2 f_0\right) + \quad \text{خطا}$$

تفاضل‌ها را برحسب جملات f می‌نویسیم:

$$\boxed{f'(x_1) = \frac{1}{h}\left(f_1 - f_0 + \frac{1}{2}(f_2 - 2f_1 + f_0\right) = \frac{f_2 - f_0}{2h} + \quad \text{خطا}} \qquad (13.6)$$

معادله *(13.6)* تقریب تفاضل‌های مرکزی نامیده می‌شود. مقدار xای که استفاده می‌شود در مرکز دامنه چند جمله‌ای نظیر است- تفاضل‌های مقادیر تابع در دو طرف $f(x)$ بکار می‌روند.

خطای معادله *(13.6)* بطور مشابه می‌تواند از معادله *(11.6)* تعیین شود.

$$\boxed{\text{خطای} \quad P_2'(x_1) = -\frac{1}{6}h^2 f'''(\xi), \qquad x_0 \le \xi \le x_2.} \qquad (14.6)$$

مخصوصاً به این موضوع توجه کنید که توان h در معادله *(14.6)* برابر دو می‌باشد، بنابراین خطا برابر $O(h^2)$ می‌باشد. این خطا را با اولین توان h در جمله خطا مقایسه کنید، وقتی یک جمله از معادله *(8.6)* استفاده می‌شود، اگرچه در هر دو محاسبه فقط دو مقدار تابعی قرار دارند. همچنین ضریب در جمله خطا به وضوح کوچکتر می‌باشد. فرمول‌های تفاضل مرکزی برای محاسبه مقادیر مشتق مطمئناً مناسبتر هستند.

اگر از معادله *(13.6)* با داده‌های جدول ۱.۶ برای برآورد $f'(1.7)$ *(1.7)* استفاده کنیم، خواهیم داشت:

$$f'(1.7) \doteq \frac{6.686 - 4.482}{2(0.2)} = 5.510.$$

$$\left\{\begin{matrix} -0.030 & (\min) \\ -0.046 & (\max) \end{matrix}\right\}.$$

(خطای واقعی برابر -0.036)

فرمول‌های تفاضل مرکزی، مشابه معادله *(13.6)*، می‌توانند از چند جمله‌ای‌های زوج درجه بالاتر بدست آیند. (چند جمله‌ای‌های فرد، دامنه مناسبی ندارند تا حول هر مقدار x تقارن داشته باشند). برای مثال، فرمول متناظر با چند جمله‌ای درجه چهار و برحسب جملاتی از مقادیر تابع بجای تفاضل‌ها و نسبت به نقطه x_0 بیان شد که برابر است با:

$$f'(x_0) = \frac{1}{h}\frac{f_{-2} - 8f_{-1} + 8f_1 - f_2}{12}, \qquad \text{خطا} = \frac{1}{30}h^4 f^{v}(\xi), \qquad x_{-2} \le \xi \le x_2.$$

در خیلی از کاربردها، ترجیح داده می‌شود که فرمولی ساده‌تر از معادله *(13.6)* مورد استفاده قرار گیرد و خطا توسط کوچک ساختن h کنترل شود. بهر صورت دیدیم که خطای مشتق یک چند جمله‌ای درجه n برابر است با $O(h^n)$، در صورتیکه خطای درونیابی برابر $O(h^{n+1})$ می‌باشد.

۳.۶ مشتقات مراتب بالاتر

می‌توانیم فرمول مشتقات مراتب بالاتر را با مشتق‌گیری بیشتر از معادله *(8.6)* بدست آوریم و مشتق‌گیری از معادله *(10.6)* مقدار خطا را بدست خواهد آمد. برای مثال:

$$\begin{aligned} f''(x_s) \doteq P_n''(x_s) &= \frac{1}{h^2}\frac{d^2}{ds^2}P_n(x_s) \\ &\doteq \frac{1}{h^2}\left(\frac{1}{2}(2)\Delta^2 f_0 + \frac{1}{6}[(s-2) + (s-1) + (s-2) + s \right. \\ &\quad \left. + (s-1) + s]\Delta^3 f_0 + \cdots\right). \qquad (15.6) \end{aligned}$$

$s = 0,$

$$f''(x_0) \doteq \frac{1}{h^2}(\Delta^2 f_0 - \Delta^3 f_0 + \cdots).$$

هیچ طرح ساده‌ای برای ضرایب وجود ندارد و جملات را نیز پیچیده کرده است. فرمول‌ها با مشتقات مراتب بالاتر پیچیده‌تر می‌گردد، در حالیکه ما به دنبال روش ساده‌تری هستیم. قبلاً دیده‌ایم که از روش‌های سمبولیک، فرمول‌های درونیابی، نسبتاً آسان بدست می‌آیند. این موضوع در مورد فرمول‌های مشتق هم واقعیت دارد:

$$E = 1 + \Delta,$$
$$y_s = E^s y_0,$$
$$y'_s = \frac{d}{dx}(E^s y_0) = \frac{1}{h}\frac{d}{ds}(E^s y_0) = \frac{1}{h}(\ln E)E^s y_0. \qquad (16.6)$$

$s = 0.$

$$y'_0 = \frac{1}{h}(\ln E)y_0 = \frac{1}{h}\ln(1 + \Delta)y_0$$
$$= \frac{1}{h}\left(\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 - \frac{1}{4}\Delta^4 y_0 + \cdots\right). \qquad (17.6)$$

در نوشتن معادله *(17.6)*، از بسط مکلورن برای تابع $Ln(1+\Delta)$ استفاده کرده‌ایم.

توجه کنید که معادله *(17.6)* همان معادله *(9.6)* می‌باشد که قبلاً آن را با مفاهیم غیر سمبولیک بدست آوردیم. اگر از سمبل D برای اپراتور مشتق استفاده کنیم.

$$Dy_0 = \frac{1}{h}\ln(1 + \Delta)y_0. \qquad (18.6)$$

با خلاصه کردن هم‌ارزی بین اپراتورها از معادله *(18.6)*:

$$D = \frac{1}{h}\ln(1 + \Delta). \qquad (19.6)$$

ارزش روش سمبولیک در این است که از عملیات جبری بر روی رابطه اپراتورها، نتایج معتبری بدست می‌آید. حال دو طرف معادله *(19.6)* را به توان می‌رسانیم:

$$D^2 = \frac{1}{h^2}\ln^2(1 + \Delta),$$
$$D^3 = \frac{1}{h^3}\ln^3(1 + \Delta), \qquad (20.6)$$
$$D^n = \frac{1}{h^n}\ln^n(1 + \Delta).$$

معادله *(20.6)* نشان می‌دهد که می‌توانیم فرمول مشتقات بالاتر را با ضرب سری معادله *(17.6)* در خودش بدست آوریم. در زیر مشتق مرتبه دوم بدست آورده می‌شود:

$$D^2 y_0 = \frac{1}{h^2}\left(\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \frac{1}{4}\Delta^4 + \cdots\right)^2 y_0$$
$$= \frac{1}{h^2}\left(\Delta^2 - \Delta^3 + \frac{11}{12}\Delta^4 - \frac{5}{6}\Delta^5 + \cdots\right)y_0;$$
$$y'' = \frac{1}{h^2}\left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12}\Delta^4 y_0 - \frac{5}{6}\Delta^5 y_0 + \cdots\right). \qquad (21.6)$$

معادلات *(20.6)* منجر به یک اصل مهم برای تخمین مشتقات از فرمول تفاضل پیشرو می‌گردد. جمله اول تمام فرمول برحسب تفاضل‌ها فقط دارای Δ^n می‌باشد. از این رو به عنوان اولین تقریب داریم:

$$D^n y = \frac{1}{h^n}\Delta^n y + \text{خطا } O(h). \qquad (22.6)$$

مثال ۲.۶ از فرمول *(21.6)* برای تخمین $y''(1.7)$ از جدول ۱.۶ با استفاده از جملات ابتدا تا Δ^3 استفاده کنید. همینطور خطا را برآورد کنید.

$$y''(1.7) = \frac{1}{(0.2)^2}(0.268 - 0.060)$$

$$=5.200$$ (مقایسه کنید با جواب دقیق *5.474*)

$$\text{خطا} \quad = \frac{1}{h^2}\left(\frac{11}{12}h^4 y^{iv}(\xi)\right), \qquad 1.7 \le \xi \le 2.1,$$

با قرار دادن $y=e^x$ و $y^{iv}=e^x$ در قسمت آخر فرمول داریم:

$$\text{ماکزیمم مقدار خطا} \quad = \frac{11}{12}(0.2)^2(e^{2.1}) = 0.298,$$

$$\text{مینیمم مقدار خطا} \quad = \frac{11}{12}(0.2)^2(e^{1.7}) = 0.201.$$

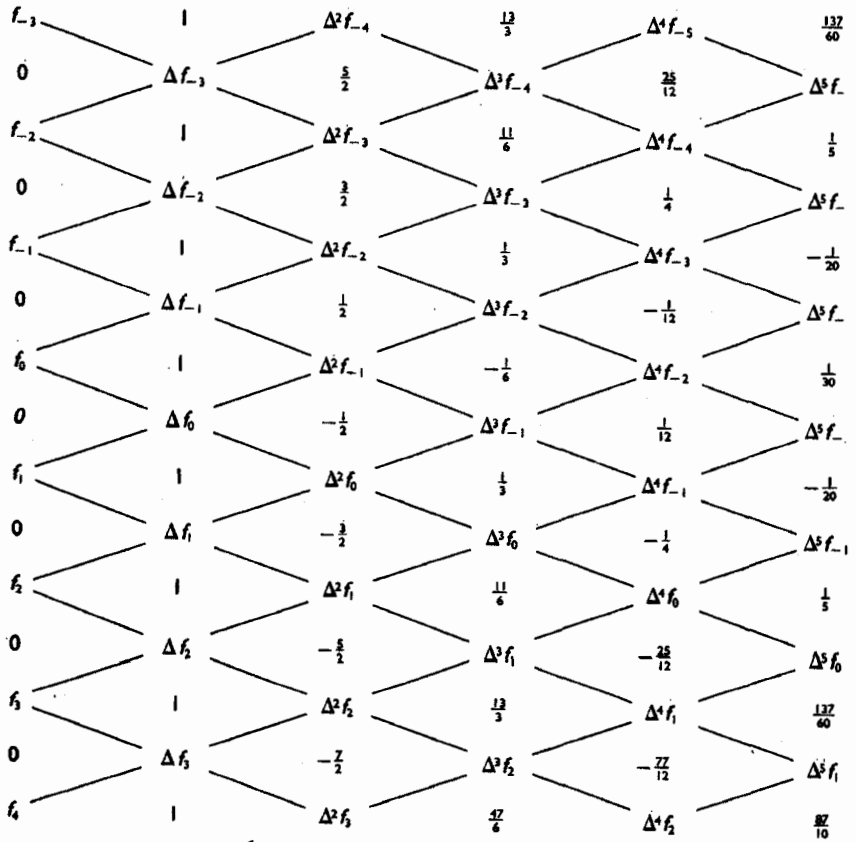
با خطای واقعی یعنی *0.274* مقایسه کنید.

وقتی از تابع شناختی در دست نیست، می‌توانیم از تفاضل چهارم استفاده کنیم تا مقدار $h^4 f^{iv}$ را به همان صورت که در معادله *(22.6)* نشان داده شده‌است، بدست آوریم. اگر تفاضل‌ها، تغییرات بزرگ داشته باشند، با یک نابسامانی مواجه هستیم. در این مورد اگر از *0.010* (مقدار متوسط $\Delta^4 y$) به عنوان برآوردی از $h^4 f^{iv}(\xi)$ استفاده می‌کردیم، خطا حدود *0.23* می‌شد. فرمول تفاضل‌های مرکزی برای مشتقات بالاتر، دقت بیشتری از فرمول تفاضل‌های پیشرو در معادله *(21.6)* دارد. ترجیح می‌دهیم این موضوع را با دیاگرام‌های لوزی *(Lozenge)* بخش بعد توضیح دهیم.

۴.۶ نمودار لوزی برای مشتقات.

از آنجا که فرم‌های معادل متنوعی از چند جمله‌ای‌های درونیاب وجود دارند، ممکن است فرمول‌های مشتق متنوعی را از مشتق گیری چند جمله‌ای‌های مختلف بدست آوریم، همانطور که برای فرمول‌های درونیاب دیدیم مطمئناً همه با هم در ارتباط هستند، این وابستگی را بانموداری مناسب بنام نمودار لوزی در شکل (۱.۶) نشان می‌دهیم.

آزمایش نمودار لوزی در شکل (۱.۶) نشان می‌دهد که این نمودار، یک جدول تفاضل‌ها به شکل سمبولیک با مقادیر عددی تفسیر شده می‌باشد.



شکل ۱.۶ دیاگرام لوزی برای $\mathbf{f'(x)}$ (توجه کنید که تمام فرمولها در $\frac{1}{\mathbf{h}}$ باید ضرب شوند).

این مقادیر به عنوان ضرایبی از تفاضل‌های fدر تشکیل فرمول مشتق بکار می‌روند و با مسیری از میان جدول تفاضل‌ها انتخاب می‌شوند.

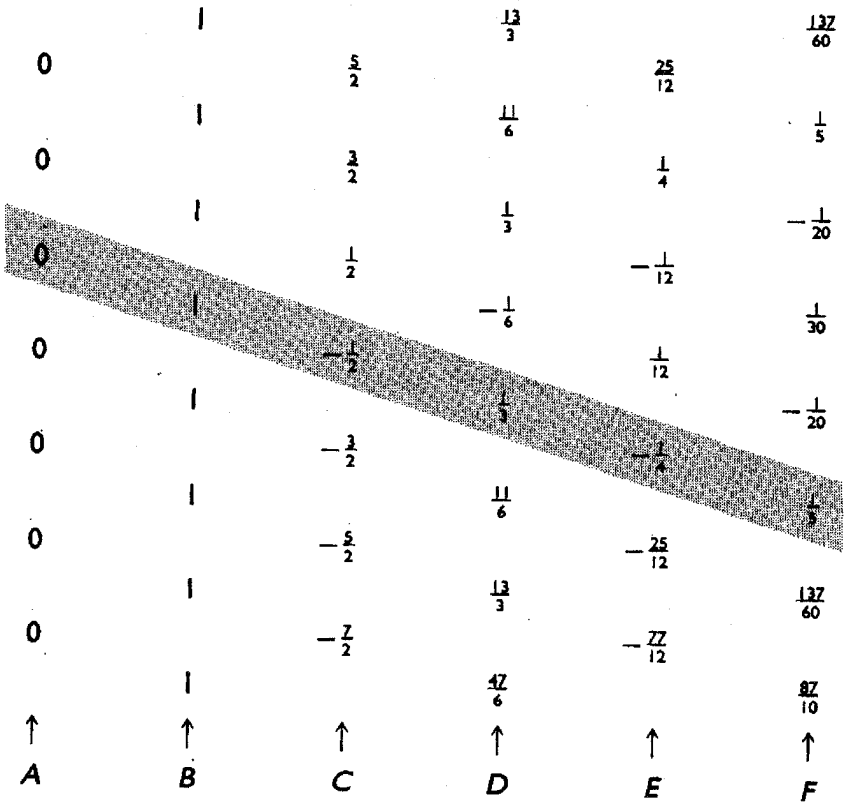
قواعد زیر باید مد نظر گرفته شوند:

۱- از f_0 ستون fها شروع می‌شود. نقطه آغاز، زیرنویسی برای ما تعریف می‌کند و از این رو مقدار x_0 نسبت به زیرنویس‌های f'باید هماهنگ باشد.

۲- پیشروی از چپ به راست و با شکل قطری به‌سمت بالا یا به‌سمت پائین به ستون تفاضل بعدی می‌باشد و یا به عبارت دیگر پیشروی افقی و متناوب است. یک جمله بازای عبور از هر ستون اضافه می‌شود.

۳- جمله‌ای را که اضافه می‌کنیم، عددی در لوزی می‌باشد، ضرب در عامل بالایی، اگر مرحله قبل به سمت پائین بوده است، یا ضرب در عامل پائینی اگر که مرحله قبل قطری به سمت بالا بوده است یا ضربدر متوسط عامل پائینی و بالایی، اگر مرحله قبل افقی بوده است.

ضریب جمله f، همواره صفر است، همانطور که نمودار مفسر نشان می‌دهد.



شکل ۲.۶

ویژگی خاصی برای نمودارهای لوزی وجود دارد که ساختن آنها را آسان می‌کند. نمودار را می‌توان شامل دو قسمت دانست: تفاضل‌های تابع و ضرایب مفسر، که یکی بر روی دیگری مؤثر است. ملاحظه کنید که آرایه ضرایب شکل ۱.۶ به عنوان شکل ۲.۶ در اینجا به نمایش گذاشته شده‌اند. با شروع از ستون‌های A و B، ستونها را دو به دو مقایسه کنید. ببینید که مقادیر A، تفاضل‌های مقادیر B هستند. حال ستون‌های B و C را مقایسه کنید. ستون B، تفاضل‌های مقادیر ستون C می‌باشند. ستون C دارای مقادیری است از ستون D که هر مقدار از ستون D از مقدار بالائی آن کسر شده است. همین موضوع برای ستون‌های C و D برای ستون‌های D و E نیز صادق است. به عبارت دیگر، آرایه‌ای از ضرایب یک جدول تفاضل می‌باشد اما از راست به چپ و از پائین به بالا نوشته شده است. این بدان معنی است که یک خط از ضرایب که آنها را از معادله *(9.6)* بدست می‌آوریم، برای ایجاد آرایه کامل ضرایب در شکل (۱.۶) کافی است. این مجموعه از ضرایب سایه زده شده است.

معادله *(21.6)* کمک می‌کند تا نمودار لوزی، برای مشتق دوم در $x = x_0$ در حالت مشابه

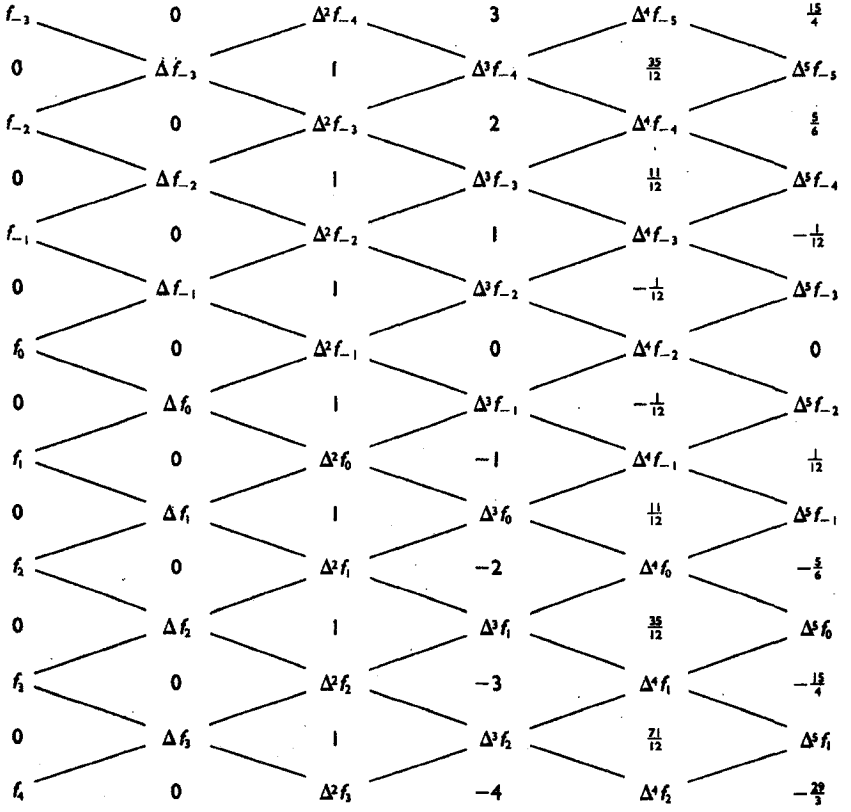
آغاز شود. نتیجه در شکل ۳.۶ مشاهده می‌گردد. از نمودارهای لوزی، برای نوشتن فرمولها، به منظور مشتق‌گیری از یک تابع جدول شده استفاده می‌کنیم. این کار را به وسیله دنبال کردن یک مسیر از میان جدول تفاضل‌ها، انجام می‌دهیم. هر مسیری از جدول می‌تواند انتخاب شود. ضرایب با قواعد بحث شده فوق بدست می‌آیند.

فرض کنید از مسیر افقی که از f_0 شروع شده، استفاده کنیم:

$$\left.\frac{df}{dx}\right|_{x=x_0} = \frac{1}{h}\left[\frac{\Delta f_0 + \Delta f_{-1}}{2} + (0)\Delta^2 f_{-1} + \left(-\frac{1}{6}\right)\frac{\Delta^3 f_{-1} + \Delta^3 f_{-2}}{2} + \cdots\right] \qquad (23.6)$$

$$= \frac{1}{h}\left[\frac{(f_1 - f_0) + (f_0 - f_{-1})}{2} + O(h^3)\right] = \frac{f_1 - f_{-1}}{2h} + O(h^2).$$

شکل (۳.۶) را مورد استفاده قرار می‌دهیم. برای پیدا کردن مشتق دوم، یک مسیر افقی را دنبال می‌کنیم، بدست می‌آوریم:



شکل (۳.۶) دیاگرام لوزی برای $f''(x)$ (تمام فرمولها در $\frac{1}{h^2}$ باید ضرب شوند).

$$\left.\frac{d^2f}{dx^2}\right|_{x=x_0} = \frac{1}{h^2}\left[\Delta^2 f_{-1} + 0 + \left(-\frac{1}{12}\right)\Delta^4 f_{-2} + \cdots\right]$$
$$= \frac{1}{h^2}[(f_1 - 2f_0 + f_{-1}) + O(h^4)] \qquad (24.6)$$
$$= \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2).$$

ملاحظه کنید که دنبال کردن یک مسیر افقی به تقریب تفاضل مرکزی منجر می‌شود. دنبال کردن یک مسیر قطری به سمت پائین، فرمولهایی را بدست می‌دهد که براساس چند جمله‌ای نیوتن- گریگوری پیشرو بنا شده است.

معادلات *(23.6)* و *(24.6)* فرمولهایی با اهمیت خاص هستند. معادله *(23.6)* هم ارز معادله *(13.6)* می‌باشد. ولی توجه کنید که اگرچه هر دو فرمول مذکور یک جمله‌ای می‌باشند، ولی در مقایسه با خطای $O(h)$ برای معادلات *(8.6)* و *(21.6)* خطاهایی از مرتبه $O(h^2)$ دارند. همانطور که در قبل بیان شد، این فرمولها به عنوان تقریب تفاضل مرکزی شناخته می‌شوند و جملهٔ $O(h^2)$ خطای مطلوب است، زیرا نقطه $x = x_0$ مرکز محدوده تغییرات x، جایی است که چند جمله‌ای درونیاب با جدول سازگار است.

فرمول تفاضل مرکزی، برای مراتب بالاتر، ادامه همان مسیر افقی است. در مثال زیر، کارآیی تقریب تفاضل مرکزی برای مشتق دوم، در مقایسه با تقریب تفاضل پیشرو، توضیح داده می‌شود.

مثال ۳.۶ با استفاده از داده‌های جدول (۱.۶)، $f''(1.7)$ را محاسبه کنید.

با استفاده از تفاضل پیشرو، معادله *(21.6)*:

$$f''(1.7) = \frac{0.268}{(0.2)^2} = 6.700,$$

$$\text{خطا} = -1.226$$

با استفاده از تفاضل مرکزی (معادله *(24.6)*).

$$f''(1.7) = \frac{0.220}{(0.2)^2} = 5.500,$$

$$\text{خطا} = -0.026$$

عبارت تفاضل مرکزی، خیلی دقیق‌تر است. برآورد خطای محاسبات بالا از $1.095+$ (کمترین) تا $1.633+$ (بیشترین) برای تفاضل پیشرو و از $0.015+$ (کمترین) تا $0.022+$ (بیشترین) برای تفاضل مرکزی می‌باشد. (توجه کنید که در عبارت دوم، گرد کردن سبب می‌شود که خطای واقعی از این حدود خارج شود). شکل ۳.۶ فرمولی از خطای $O(h^4)$ برای مشتق دوم را نشان می‌دهد. اگر مسیر افقی را ادامه دهیم تا جائی که به ستون چهارم تفاضل برسیم:

$$\left.\frac{d^2f}{dx^2}\right|_{x=x_0} = \frac{1}{h^2}\left[\Delta^2 f_{-1} - \frac{1}{12}\Delta^4 f_{-2}\right] + O(h^4).$$

این رابطه را برحسب مقادیر f بازنویسی می‌کنیم و به یک عبارت هم‌ارز می‌رسیم:

$$\left.\frac{d^2f}{dx^2}\right|_{x=x_0} = \frac{-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2}{12h^2} + O(h^4).$$

با بکار بردن این فرمولها برای داده‌های جدول ۱.۶، $f''(1.7)$ را محاسبه می‌کنیم و داریم:

$$f''(1.7) = \frac{1}{(0.2)^2}\left[0.220 - \frac{1}{12}(0.007)\right] = 5.485$$

$$= \frac{-8.166 + 16(6.686) - 30(5.474) + 16(4.482) - 3.669}{12(0.2)^2} = 5.485.$$

(نتیجه را با مقدار دقیق 5.474 مقایسه کنید. خطای 0.011 - به علت گردکردن خیلی بزرگتر از خطای برش است. (0.00108 - تا 0.0049 -).

تقریب تفاضل مرکزی برای مشتقات بالاتر به خوبی تعمیم پیدا می‌کند. نمودار لوزی می‌تواند برای این موضوع ایجاد شود. اما اگر تنها اولین جمله غیر صفر مورد نیاز باشد، یک فرمول ساده کفایت می‌کند. برای مشتق‌های مرتبه زوج، این فرمول عبارتست از:

$$f^{(n)}(x)|_{x=x_0} = \frac{1}{h^n}\Delta^n f_{-n/2} + O(h^2), \qquad n \text{ زوج}$$

همیشه تفاضل مورد نظر روی خط افقی واصل بین x_0 و f_0 در جدول پیدا می‌شود. اگر جدول تفاضل ساخته نشده باشد، برحسب مقادیر f آنرا بیان می‌کنیم.

برای مشتقات مرتبه فرد، متوسط تفاضل‌های بالا و پائین خط افقی گذرنده از x_0 را بیان می‌کنیم.

$$f^{(n)}(x)|_{x=x_0} = \frac{\Delta^n f_{(-n+1)/2} + \Delta^n f_{(-n-1)/2}}{2h^n} + O(h^2), \qquad (25.6)$$

۵.۶ فنون برونیابی

برای بهبود تخمین مشتق بطور متوالی از جملات نمودار لوزی استفاده می‌کنیم و در این مورد برای مشتق‌گیری از توابع دلخواه، برنامه‌های کامپیوتری خاص وجود دارد. ابتدا این روش را برای یک تابع که نتیجه از یک جدول شناخته می‌شود، توضیح می‌دهیم.

مثال ۴.۶ از جدول ۲.۶ مشتق $f'(x)$ را در $x=2.5$ بدست آورید. فرمول تفاضل مرکزی یعنی معادله (13.6) یا معادله (23.6) نتیجه می‌دهد:

$$f'(x)|_{x=2.5} = \frac{f_1 - f_{-1}}{2h} = \frac{0.25337 - 0.31729}{2(0.1)} = -0.3196.$$

این تخمین دارای خطای $O(h^2)$ می‌باشد.

جدول (۲.۶)

x	$f(x)$	x	$f(x)$
2.0	0.42298	2.6	0.25337
2.1	0.40051	2.7	0.22008
2.2	0.37507	2.8	0.18649
2.3	0.34718	2.9	0.15290
2.4	0.31729	3.0	0.11963
2.5	0.28587		

همچنین مشتق را با استفاده از مقادیر تابع در $x=2.7$ و $x=2.3$ می‌توان محاسبه کرد. برای محاسبه، فاصله 0.2 در نظر گرفته می‌شود و بدست می‌آوریم:

$$f'(x)|_{x=2.5} \doteq \frac{0.22008 - 0.34718}{2(0.2)} = -0.3178.$$

اینجا هم خطا $O(h^2)$ می‌باشد، اما حالا h دو برابر بزرگتر از آنچه که در محاسبات قبلی بوده، می‌باشد.

به منظور ترکیب این دو تخمین و برونیابی برای برآورد دقیق‌تر جواب، وارد جزئیات قضیه می‌شویم و ماهیت[1] خطای معادله *(23.6)* را بررسی می‌کنیم. توسط نمودار لوزی شکل (۱.۶) داریم:

$$f'_0 = \frac{1}{h}\left(\frac{\Delta f_{-1} + \Delta f_0}{2} - \frac{1}{6}\frac{\Delta^3 f_{-2} + \Delta^3 f_{-1}}{2} + \frac{1}{30}\frac{\Delta^5 f_{-3} + \Delta^5 f_{-2}}{2} - \cdots\right). \qquad (26.6)$$

اگر بعد از جمله اول را صرف نظر کنیم، به یک خطا از مرتبه $O(h^2)$ می‌رسیم، اما معادله *(25.6)* نشان می‌دهد که:

$$f'''_0 = \frac{\Delta^3 f_{-2} + \Delta^3 f_{-1}}{2h^3} + O(h^2).$$

بطور هم ارز، ممکن است بنویسیم:

$$-\frac{1}{6}\frac{\Delta^3 f_{-2} + \Delta^3 f_{-1}}{2} = -\frac{1}{6}h^3 f'''_0 + O(h^5) = Ch^3 + O(h^5).$$

در این عبارت، c ثابت می‌باشد. صرفا درک می‌کنیم که مشتق سوم f، در x_0، یک مقدار ثابت دارد. معادله *(26.6)* می‌تواند بصورت زیر نوشته شود:

$$f'_0 = \frac{1}{h}\frac{\Delta f_{-1} + \Delta f_0}{2} + Ch^2 + O(h^4) + \frac{1}{30}h^4 f^{v}(\xi)$$
$$= \frac{1}{h}\frac{\Delta f_{-1} + \Delta f_0}{2} + Ch^2 + O(h^4).$$

1 – *nature*

می‌بینیم که با فرض اینکه خطای معادله (23.6) متناسب با h^2 است، تنها یک خطا از مرتبه h^4 داریم. سپس دو برآورد مشتق را ترکیب می‌کنیم، -0.3196 با $h=0.1$ و -0.3178 با $h=0.2$ به شرح زیر:

اگر از جمله $O(h^4)$ صرفنظر کنیم f'_0 بر اساس دقت کمتر
$$= -0.3178 + Ch^2 + O(h^4),$$
$$\doteq -0.3178 + C(0.2)^2,$$

اگر از جمله $O(h^4)$ صرفنظر کنیم f'_0 بر اساس دقت بیشتر
$$= -0.3196 + Ch^2 + O(h^4),$$
$$\doteq -0.3196 + C(0.1)^2,$$

برای مقادیر معلوم (دقیق) f'_0 را حذف C بین دو عبارت، مساله را حل می‌کنیم.

$$f'_0 = -0.3196 + \frac{1}{3}[-0.3196-(-0.3178)] = -0.3203$$

روش را برای هر دو برآورد، هر یک با خطای $O(h^n)$ و با تغییر h از 2 به 1 بکار می‌بریم.

قاعده این است:

$$\text{مقدار دقیق} \cong \text{دقت بیشتر} + \left(\frac{1}{2^n-1}\right)(\text{دقت بیشتر} - \text{دقت کمتر}) \qquad (27.6)$$

با استفاده از علامت $\cong$ نشان می‌دهیم که رابطه کامل نیست، این عدم دقت نتیجه صرفنظر کردن جمله $O(h^{n+2})$ در هر دو عبارت می‌باشد.

روش برونیابی قابل بسط می‌باشد زیرا می‌توان نشان داد اولین مرتبه برونیابی با خطای $O(h^4)$ خطایی به شکل $Ch^4+O(h^6)$ دارد.

برای داده‌های جدول ۲.۶، تا یک مرتبه بیشتر برونیابی را ادامه می‌دهیم. داده‌ها را به صورت جدول مرتب می‌کنیم.

h	تخمین اولیه	برونیابی مرتبه اول	برونیابی مرتبه دوم
0.1	−0.31960		
		−0.32022	
0.2	−0.31775		−0.32020
		−0.32050	
0.4	−0.30951		

آخرین مقدار ثبت شده بصورت زیر محاسبه شده است.

$$-0.32022 + \frac{1}{2^4-1}(-0.32022 + 0.32050) = -0.32020.$$

یک رقم اضافی برای احتیاط در نظر گرفته شده است تا خطای گردکردن را در محاسبات کاهش دهد.

از آنجاکه نمودار لوزی برای مشتق دوم در شکل ۳.۶، نشان می‌دهد که ضرایب متناوباً صفر می‌شوند، تقریب‌های مشتق دوم دقیقاً به روش مشابه‌می‌تواند برونیابی شود. مرتبه خطا در هر

مرحله به h^4، h^6 و h^8و... افزایش می‌یابد.

۶.۶ گردکردن و دقت مشتقات

درک تأثیر خطای گردکردن روی دقت مشتقاتی که بوسیله فرمول‌های تفاضل محدود محاسبه شده‌اند، مهم است. همانطوری که مشاهده شد، کاهش مقدار h یا افزایش درجه چند جمله‌ای درونیابی شده، دقت فرمول‌های مشتق را افزایش می‌دهد. (این روش عملکرد خطای برش را کاهش می‌دهد.)

بهر حال، وقتی nکاهش یابد، ما نیاز داریم که تفاضل مقادیری از تابع راکه بهم خیلی نزدیک هستند، بدست آوریم. در نتیجه همانطور که دیدیم، یک خطای بزرگ ناشی از گرد کردن ایجاد می‌شود. افزایش خطای گرد کردن، وقتی h کوچکتر می‌شود، باعث خواهد شد تا بهترین دقت در برخی از نقاط میانی ایجاد شود. داده‌های شکل (۴.۶) *(a)* و *(b)*، این موضوع را روشن می‌سازند. آزمون به دو طریق صورت گرفته است، با دقت ساده (که *6* تا *7* رقم با معنی دقت دارد) و دقت مضاعف (که تا *15* رقم با معنی دقت دارد). تابع مورد آزمون $f(x)=e^x$در نقطهٔ $x=0$ می‌باشد. البته مقدار تحقیقی $f'(0)$ مطمئناً یک می‌باشد و تقریب تفاضل مرکزی بکار گرفته شده است.

H	DX	DDX
0.1E 00	0.1001663E 01	0.1000761E 01
0.1E−01	0.1000001E 01	0.9959934E 00
0.1E−02	0.9999569E 00	0.8940693E 00
0.1E−03	0.9959930E 00	−0.8344640E 02
0.1E−04	0.9775158E 00	−0.4768363E 04
0.1E−05	0.9834763E 00	−0.5960462E 05
0.1E−06	0.5960463E 00	−0.1192093E 08
0.1E−07	0.2980230E 01	−0.5960458E 09
0.1E−08	0.2980229E 02	−0.5960459E 11
0.1E−09	0.2980229E 03	−0.5960456E 13

a) دقت ساده

H	DX	DDX
0.1E 00	0.1001667E 01	0.1000834E 01
0.1E−01	0.1000016E 01	0.1000008E 01
0.1E−02	0.1000000E 01	0.1000000E 01
0.1E−03	0.1000000E 01	0.9999999E 00
0.1E−04	0.1000000E 01	0.9999989E 00
0.1E−05	0.1000000E 01	0.1000031E 01
0.1E−06	0.9999999E 01	0.9894834E 00
0.1E−07	0.9999999E 00	−0.4163322E 00
0.1E−08	0.9999999E 00	−0.4163319E 02
0.1E−09	0.1000000E 01	0.2775545E 04

b) دقت مضاعف

شکل ۴.۶ – مشتقات محاسبه شده بوسیله فرمول تفاضل مرکزی تابع $f(x)=e^x$ در نقطه $x=0$.

در آزمون با دقت ساده، مقدار بهینه h تنها 0.01 برای مشتق اول بوده است و بهترین دقت مشتق دوم با کمال تعجب با مقدار بزرگ $h=0.1$ روی داده است. وقتی خطای گردکردن توسط دقت مضاعف کاهش می‌یابد، ضرورتا نتایج دقیق برای مشتق اول در سرتاسر دامنه مقادیر h بدست می‌آیند. برای مشتق دوم $h=0.1*10^{-2}$ یا $h=0.1*10^{-3}$ بهترین دقت را بدست می‌دهند. توجه کنید که رشد خطاها چقدر بزرگ هستند. این مخصوصاً برای مشتق دوم، وقتی h از مقدار بهینه کوچکتر می‌شود، قابل توجه است. حتی انتظار می‌رود، مشتقات بالاتر از مرتبه 2 به خاطر خطای گردکردن دقت مناسبی نداشته باشند. پس خیلی شگفت‌آور نیست که در کامپیوترهای مختلف برای دقت بیشتر، در حدود 2 برابر ارقام با معنی برای دقت مضاعف و یا بیشتر از آن، پیش‌بینی شده است.

برای سهولت، فرمول‌های محاسبه مشتقات را در این قسمت جمع‌آوری کرده‌ایم.

فرمولهایی برای محاسبه مشتقات:

فرمول‌ها برای مشتق اول:

$$f'(x_0) = \frac{f_1 - f_0}{h} + O(h).$$

(تفاضل مرکزی)

$$f'(x_0) = \frac{f_1 - f_{-1}}{2h} + O(h^2).$$

$$f'(x_0) = \frac{-f_2 + 4f_1 - 3f_0}{2h} + O(h^2).$$

(تفاضل مرکزی)

$$f'(x_0) = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + O(h^4).$$

فرمولها برای مشتق دوم:

$$f''(x_0) = \frac{f_2 - 2f_1 + f_0}{h^2} + O(h).$$

(تفاضل مرکزی)

$$f''(x_0) = \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2).$$

$$f''(x_0) = \frac{-f_3 + 4f_2 - 5f_1 + 2f_0}{h^2} + O(h^2).$$

(تفاضل مرکزی)

$$f''(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + O(h^4).$$

فرمولها برای مشتق سوم:

$$f'''(x_0) = \frac{f_3 - 3f_2 + 3f_1 - f_0}{h^3} + O(h).$$

(تفاضل مرکزی)

$$f'''(x_0) = \frac{f_2 - 2f_1 + 2f_{-1} - f_{-2}}{2h^3} + O(h^2).$$

فرمولها برای مشتق چهارم:

$$f^{iv}(x_0) = \frac{f_4 - 4f_3 + 6f_2 - 4f_1 + f_0}{h^4} + O(h).$$

(تفاضل مرکزی)

$$f^{iv}(x_0) = \frac{f_2 - 4f_1 + 6f_0 - 4f_{-1} + f_{-2}}{h^4} + O(h^2).$$

شاید طریقی ساده‌تر و شهودی‌تر، برای بدست آوردن فرمول‌های بالا، از بسط سری تیلر حول نقطه x_0 باشد. فرض کنید نقاط $(x_{-2}$ و $f_{-2})$ و... $(x_2$ و $f_2)$ را وقتی $f_i=f(x_i)$ می‌باشد در اختیار داریم، نقاط از یکدیگر فاصله‌های مساوی دارند. $i=-2$ و... و 2 و $x_i=x_0+i*h$ اگر بسط سری را حول x_0 با مقدار h در نظر بگیریم، بدست می‌آوریم:

$$f(x_1) = f_1 = f_0 + hf_0' + \frac{h^2}{2}f_0'' + \frac{h^3}{6}f_0''' + \frac{h^4}{24}f_0^{iv} + \cdots; \qquad (28.6)$$

$$f(x_{-1}) = f_{-1} = f_0 - hf_0' + \frac{h^2}{2}f_0'' - \frac{h^3}{6}f_0''' + \frac{h^4}{24}f_0^{iv} - \cdots. \qquad (29.6)$$

با کم کردن دو رابطه خواهیم داشت:

$$f_0' = \frac{f_1 - f_{-1}}{2h} + O(h^2);$$

با جمع کردن بطور مشابه خواهیم داشت

$$f_0'' = \frac{f_1 - 2f_0 + f_{-1}}{h^2} + O(h^2).$$

اگر یک سری تیلر را برای $x=\pm 2h$ بسط دهیم خواهیم داشت:

$$f_2 = f_0 + 2hf_0' + 2h^2f_0'' + \frac{4}{3}h^3f_0''' + \frac{2}{3}h^4f_0^{iv} + \cdots; \qquad (30.6)$$

$$f_{-2} = f_0 - 2hf_0' + 2h^2f_0'' - \frac{4}{3}h^3f_0''' + \frac{2}{3}h^4f_0^{iv} - \cdots. \qquad (31.6)$$

اگر فرمول‌های اخیر را با قبلی، ترکیب کنیم، به سهولت می‌بینیم که

$$f_0' = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + O(h^4),$$

$$f_0'' = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + O(h^4).$$

با استفاده از همین روش، می‌توانیم فرمول‌های خاص مورد نیاز را تعمیم دهیم. برای نمونه، فرض کنید نقاط داده شده در فاصله‌های مساوی از یکدیگر قرار ندارند، به شکلی که در جدول زیر آمده است:

x	$f(x)$
0.90	2.4596
1.00	2.7183
1.11	3.0344

بسط حول x_0 عبارت خواهد بود از:

$$f(x_0 + t) = f_0 + tf'_0 + \frac{t^2}{2}f''_0 + \frac{t^3}{6}f'''_0 + \cdots; \qquad (32.6)$$

$$f(x_0 - s) = f_0 - sf'_0 + \frac{s^2}{2}f''_0 - \frac{s^3}{6}f'''_0 + \cdots. \qquad (33.6)$$

توسط این دو معادله، معادله زیر حاصل می‌شود:

$$f'_0 = \frac{\left(\frac{1}{t^2}f(x_0 + t) - \frac{1}{s^2}f(x_0 - s)\right)}{\left(\frac{1}{t} + \frac{1}{s}\right)} - \left(\frac{1}{t} - \frac{1}{s}\right)f_0. \qquad (34.6)$$

با استفاده از مقادیر جدول با فرض $t=0.11$ و $s=0.10$، از معادله *(34.6)* خواهیم داشت:

$$f'_0 = \frac{\left(\frac{1}{0.0121}(3.0344) - \frac{1}{0.01}(2.4596)\right)}{\left(\frac{1}{0.11} + \frac{1}{0.10}\right)} - \left(\frac{1}{0.11} - \frac{1}{0.10}\right)(2.7183)$$

$$= 2.7235.$$

از آنجا که این مقادیر برای $f(x)=e^x$ می‌باشند، جواب دقیق در $x=1.0$ مقدار 2.7183 می‌باشد.

۷.۶ برنامه کامپیوتری برای محاسبه مشتق

برنامه ۱.۶ لیست برنامه محاسبه مشتقات به زبان *BASIC*

برنامهٔ محاسبه مشتقات از جدول داده‌های متساوی الفاصله بنام *DER.BAS* می‌باشد.

```
1   REM       APPLIED NUMERICAL ANALYSIS
20  REM
25  REM THIS PROGRAM GETS DERIVATIVES FROM A EQUISPACED TABLE
30  REM          USING CENTRAL DIFFERENCES
35  REM   N = NO OF PTS, PUT X-VALUES IN 60, Y'S IN 70
40  N = 7
50  DIM X(N), Y(N), DIF(N,5)
60  DATA 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5
70  DATA 3.669, 4.482, 5.474, 6.686, 8.166, 9.974, 12.182
80  REM SET UP THE ARRAYS
90  FOR I = 1 TO N: READ X(I): NEXT I
100 FOR I = 1 TO N: READ Y(I): NEXT I
110 REM CREATE THE TABLE OF DIFFERENCES
120 O = 5: IF N<5 THEN O = N-1
130 FOR I = 1 TO N: DIF(I,1) = Y(I): NEXT I
140 FOR I = 2 TO O: FOR J = 1 TO N-I+1
150 DIF(J,I) = (DIF(J+1,I-1)-DIF(J,I-1))
160 NEXT J: NEXT I
170 REM PRINT THE TABLE
180 FOR I = 1 TO N
190 PRINT USING " ##.###    ";X(I);
200 FOR J = 1 TO N-I+1
210 IF J>5 THEN GOTO 230
220 PRINT USING " ##.###    ";DIF(I,J);
230 NEXT J: PRINT: NEXT I: PRINT
240   REM GET X-VALUE FOR DERIV AND DEGREE OF POLYN
250 INPUT "WHAT X-VALUE FOR DERIV ";XX
260 REM BE SURE XX IS IN TABLE
270 SS = 0: FOR I = 1 TO N: IF XX = X(I) THEN SS = I
275 NEXT I
280  IF SS = 0 THEN PRINT "YOUR VALUE NOT IN TABLE": GOTO 250
290 IF SS < 2 THEN PRINT "X-VALUE TOO SMALL": GOTO 250
300  IF SS = N THEN PRINT "X-VALUE TOO LARGE": GOTO 250
310 H = X(2)-X(1)
320 D1 = (DIF(SS,2)+DIF(SS-1,2))/2/H
330 D2 = DIF(SS-1,3)/H/H
340 PRINT "USING POLY OF DEGREE 2, DERIV IS ";D1
350 PRINT "   THE SECOND DERIVATIVE IS ";D2
360 IF SS < 3 THEN GOTO 250
370 IF SS > N-2 THEN GOTO 250
380 PRINT "USING POLY OF DEGREE 4, DERIV IS ";
```

```
390  PRINT  D1 - (DIF(SS-2,4)+DIF(SS-1,4))/12/H
400  PRINT  "    THE SECOND DERIVATIVE    IS ";
410  PRINT  D2 - DIF(SS-2,5)/12/H/H
420  GOTO  250
```

****

OUTPUT FOR DER.BAS

```
 1.300   3.669   0.813   0.179   0.041   0.007
 1.500   4.482   0.992   0.220   0.048   0.012
 1.700   5.474   1.212   0.268   0.060   0.012
 1.900   6.686   1.480   0.328   0.072
 2.100   8.166   1.808   0.400
 2.300   9.974   2.208
 2.500  12.182

WHAT  X-VALUE  FOR DERIV  ? 1.6
YOUR  VALUE  NOT IN TABLE
WHAT  X-VALUE  FOR DERIV  ? 1.7
USING  POLY  OF DEGREE  2, DERIV  IS  5.509999
   THE SECOND  DERIVATIVE   IS  5.499992
USING  POLY  OF DEGREE  4, DERIV  IS  5.472915
   THE SECOND  DERIVATIVE   IS  5.485406
WHAT  X-VALUE  FOR DERIV  ? 2.5
X-VALUE  TOO LARGE
WHAT  X-VALUE  FOR DERIV  ? 2.3
USING  POLY  OF DEGREE  2, DERIV  IS  10.04
   THE SECOND  DERIVATIVE   IS  10.00001


WHAT  X-VALUE  FOR DERIV  ? 1.9
USING  POLY  OF DEGREE  2, DERIV  IS  6.73
   THE SECOND  DERIVATIVE   IS  6.700012
USING  POLY  OF DEGREE  4, DERIV  IS  6.685
   THE SECOND  DERIVATIVE   IS  6.675016
WHAT  X-VALUE  FOR DERIV  ?
Break in 250
Ok
```

تمرینات فصل ششم

۱ – از جدول تفاضل محدود زیر، مقدار $f'(0.242)$ را از یک چند جمله‌ای درجه دوم بدست آورید که برازنده سه نقطه مناسب جدول $i = 1, 2, 3$ باشد.

i	x_i	f_i	$f_i^{[1]}$	$f_i^{[2]}$	$f_i^{[3]}$
0	0.15	0.1761	2.4355	−5.7505	15.3476
1	0.21	0.3222	1.9754	−3.9088	8.7492
2	0.23	0.3617	1.7409	−2.9464	5.9642
3	0.27	0.4314	1.4757	−2.2307	
4	0.32	0.5051	1.2973		
5	0.35	0.5441			

۲ – خطای تمرین ۱ را از قانون جمله بعدی تخمین بزنید. تابع واقعی برابر $1 + \log_{10}x$ می‌باشد. مقدار خطای تخمینی را با خطای واقعی مقایسه کنید.

۳ – مقادیر تابع $f(x) = x + \frac{sin(x)}{3}$ در هفت نقطه در جدول زیر نوشته شده است. از جدول تفاضل‌های محدود نتایج زیر را معین کنید.

a. $f'(0.72)$ از چند جمله‌ای درجه سوم

b. $f'(1.33)$ از چند جمله‌ای درجه دوم

c. $f'(0.50)$ از چند جمله‌ای درجه چهارم

در هر مورد بهترین شروع x_i را انتخاب کنید.

جواب:

i	x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
0	0.30	0.3985	0.2613	−0.0064	−0.0022	0.0003
1	0.50	0.6598	0.2549	−0.0086	−0.0018	0.0004
2	0.70	0.9147	0.2464	−0.0104	−0.0014	0.0005
3	0.90	1.1611	0.2360	−0.0118	−0.0010	
4	1.10	1.3971	0.2241	−0.0128		
5	1.30	1.6212	0.2113			
6	1.50	1.8325				

a. $p_3(x)=1.2506$ از $f'(0.72)$

b. $p_2(x)=1.0790$ از $f'(1.33)$

c. $p_4(x)=1.2925$ از $f'(0.50)$

۴ - از داده‌های تمرین ۳ فرمول تفاضل مرکزی را بکار برده و مقدار $f'(0.50)$ را محاسبه کنید. ماکزیمم و می‌نیمم تخمین خطا را بدست آورید و با خطای واقعی مقایسه کنید.

جواب: $f'(0.5)=1.2905$

و جواب دقیق 1.29253 می‌باشد.

۵ - اگر داشته باشیم $f(x) = \frac{e^x}{x-2}$. فرمول تفاضل مرکزی را بکار برده و مقدار $f'(0.66)$ را تخمین بزنید. همچنین یک جمله از فرمول تفاضل پیشرو را محاسبه کنید. خطای را با خطای واقعی مقایسه کنید.

۶ - اپراتور مشتق مرتبه اول را در خودش ضرب کنید و فرمول مشتقات مرتبه سوم و چهارم $f(x_i)$ را بدست آورید.

۷ - برای محاسبه و تخمین مشتق دوم $f(x) = \frac{e^x}{x-2}$، یک جمله، دو جمله یا سه جمله از معادله مشتق را در $x = 0.5$ با $h = 0.05$ بکار برید.

خطاها را با قانون جمله بعدی تخمین بزنید و با خطای واقعی مقایسه کنید.

جواب:

تعداد جمله	مقدار	برآورد خطا
1	-3.988	0.508
2	-3.480	-0.073
3	-3.552	0.016

مقدار دقیق -3.54169 می‌باشد.

۸ - روش دیگر بدست آوردن فرمول‌های مشتقات از بسط سری تیلور $f(x)$ در همسایگی نقطه x_0 می‌باشد. به عنوان مثال، با تفاضل سری برای $x=x_0+h$ از سری دیگر بطوری که $x=x_0-h$ باشد برای $f'(x_0)$ فرمول تفاضل مرکزی و جمله خطا و مرتبه خطا را بدست آورید.

۹ - مثال ۸ را برای بدست آوردن فرمول تفاضل پیشرو برای $f'(x_0)$ تکرار کنید، همچنین مجدداً برای بدست آوردن فرمول تفاضل پسرو برحسب x_0 و x_0-h عملیات را بنویسید.

۱۰ - فن برونیابی را بکار برده و $f'(0.90)$ را از جدول تمرین ۳ با دقت $O(h^6)$ بدست آورید.

۱۱ - از جدول تمرین ۳، فرمول تفاضل‌ها پیشرو را بکار برده و دو جمله را بدست آورید. همچنین مقدار خطا را با قانون جمله بعدی محاسبه کنید.

۱۲ - جدول تمرین ۱ را بکار برید و مقدار $f'(0.242)$ را از یک چند جمله‌ای درجه دوم بدست آورید که با مقدار x_i شروع شود.

a. $i = 0$

b. $i = 2$

c. $i = 3$

درجه چند جمله‌ای که دقیق‌ترین مقدار را می‌دهد. چیست؟

۱۳ - نشان دهید که اولین برونیابی برای $f'(x_0)$ و تغییر مقدار h مطابق فرمول زیر است.

$$f'_0 = \frac{1}{H}\left[\frac{\Delta f_{-1} + \Delta f_0}{2} - \frac{1}{6}\,\frac{\Delta^3 f_{-2} + \Delta^3 f_{-1}}{2}\right]$$

که H کوچکتر از مقدار h می‌باشد.

۱۴ - فرمول‌های تمرین ۶ را بکار برده و مشتق سوم و چهارم $f(x) = e^{-x/2}$ را در $x = 0.3$ با انتخاب $h = 0.1$ یا $h = 0.05$ به دست آورید.

۱۵ - از جدول تمرین ۱ مقدار $f'(x)$ را به ازای مقادیر x به دست آورید.

$x=0.21,\ 0.22,\ 0.23,\ 0.24,\ 0.25,\ 0.26,\ 0.27$

جواب:

x	مقدار محاسبه شده	مقدار دقیق
0.21	2.0904	2.0681
0.22	1.9754	1.9741
0.23	1.8973	1.8882
0.24	1.8191	1.8096
0.25	1.7409	1.7372
0.26	1.6627	1.6704
0.27	1.6230	1.6085

۱۶ - برای تابع $f(x)= \sin^2(x/2)$ با انتخاب $h=0.1$ از فرمول تفاضل مرکزی مقدار $f'=(0.32)$ و $f''=(0.32)$ را به دست آورید.

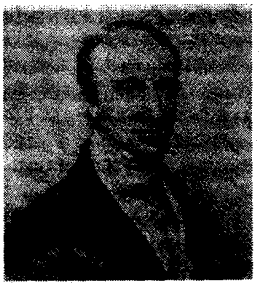
جواب: $f'=(0.32)= 0.15728$

فصل ۷

انتگرال گیری عددی

موضوعات این فصل

* فرمول‌های انتگرال‌گیری نیوتن -کوتز
* روش ذوزنقه‌ای
* انتگرال گیری به روش سیمپسون
* روش رامبرگ
* روش‌های دیگر بدست آوردن فرمول انتگرال
* روش انتگرال گاوس
* انتگرال‌های ناویژه و نامعین
* موارد استفاده توابع اسپلاین
* انتگرال‌های چندگانه (دوگانه، سه گانه و...)
* خطا در انتگرالها و تعمیم آنها. (چندگانه).
* انتگرال‌های چندگانه با حدود متغیر
* برنامه‌های کامپیوتری
* برنامه انتگرال دوگانه به روش رامبرگ به زبان FORTRAN
* برنامه انتگرال گیری به روش سیمپسون به زبان PASCAL
* برنامه انتگرال گیری به روش گاوس به زبان C
* تمرینات فصل هفتم



اوگوستین -لوئی کوشی
Augustin Lois Cauchy
(۱۷۸۹ - ۱۸۵۷)

کوشی برجسته‌ترین آنالیزدان نیمه اول قرن نوزدهم می‌باشد. وی در سال ۱۷۸۹ در پاریس متولد شد. در سال ۱۸۰۵ وارد مدرسه پلی تکنیک شد و تحسین لاگرانژ و لاپلاس را برانگیخت، و در اثر ترغیب آنان تصمیم گرفت از مهندسی راه و ساختمان به نفع علوم مجرد دست بکشد.

کوشی هم در زمینه ریاضیات مجرد و هم در ریاضیات کاربسته بسیار زیاد مطلب نوشته است. مجموعه آثار او، علاوه بر چندین جلد کتاب شامل ۷۸۹ مقاله است و ۲۴ کتاب را شامل می‌شود.

کارهای پر شمار کوشی در ریاضیات پیشرفته شامل تحقیقاتی در همگرایی و واگرایی سری‌های نامتناهی، نظریه توابع حقیقی و مختلط، معادلات دیفرانسیل، دترمینانها، احتمالات، و فیزیک و ریاضی است، در اولین درس نظریهٔ توابع مختلط، به نامساوی کوشی، فرمول انتگرال کوشی، و معادلات دیفرانسیل کوشی - ریمان برمی‌خوریم. قسمت عمدهٔ مباحث حسابان امروزی دانشگاه، نظیر مفاهیم حدّ و پیوستگی، نتیجهٔ کار کوشی است.

کوشی مشتق $y=f(x)$ نسبت به x را به صورت حدّ خارج قسمت تفاضل

$$\frac{\Delta y}{\Delta x}=\frac{f(x+\Delta x)-f(x)}{\Delta x}$$

وقتی $\Delta x\to 0$ تعریف کرد. اگر dx کمیتی متناهی باشد، وی dy تابع $f(x)$ را صرفاً به صورت $f'(x)dx$ تعریف کرد. کوشی در سال ۱۸۴۰ کلمهٔ «مشخصه» را برای معادلهٔ مشخصهٔ ماتریس A با نامیدن معادلهٔ $|A-\lambda I|=0$، وارد نظریهٔ ماتریس‌ها کرد.

کوشی انتگرال معین را به عنوان حد مجموعی از مجموعه‌ای به طور نامتناهی افزایش یابندهٔ از اجزاء کوچکی که به صفر میل می‌کنند، تعریف نمود. بسیار نظیر کاری که امروزهٔ انجام می‌شود. کوشی در سال ۱۸۵۷ در ۶۸ سالگی درگذشت. آخرین سخنان او این بود که «انسان‌ها می‌میرند، اما کرده‌های آنان می‌مانند.»

۱.۷ فرمول‌های انتگرال گیری نیوتن-کوتز

روش معمول در بسط فرمولها برای انتگرال گیری عددی شبیه مشتق‌گیری عددی است. ما یک چند جمله‌ای از میان نقاط تابع عبور می‌دهیم و آنگاه از این چند جمله‌ای تقریب به تابع، انتگرال می‌گیریم. باین طریق راه برای انتگرال‌گیری از یک تابع شناخته شده که تنها بصورت جدولی از مقادیر است باز می‌شود. وقتی که مقادیر در فاصله‌های مساوی از هم قرار دارند، رابطه آشنای چند جمله‌ای نیوتن- گریگوری پیشرو، نقطه شروع مناسبی می‌باشد. بنابراین:

$$\int_a^b f(x)\,dx \doteq \int_a^b P_n(x_s)\,dx. \qquad (1.7)$$

فرمولی که از معادله *(1.7)* بدست آوردیم، دقیق نیست، زیرا چند جمله‌ای با $f(x)$ مساوی نیست. بوسیله انتگرال‌گیری از جمله خطای $P_n(x_s)$، عبارت خطا بدست می‌آید.

$$\text{خطا} \qquad = \int_a^b \binom{s}{n+1} h^{n+1} f^{(n+1)}(\xi)\,dx.$$

راه‌های مختلفی برای بکارگیری معادله *(1.7)* وجود دارد. می‌توان فاصله انتگرال‌گیری *(b و a)* را با یکسری از چند جمله‌ای‌های مناسب در فاصله $(x_n$ و $x_0)$ تطبیق داد. در اینصورت فرمول‌های نیوتن کوتز بدست می‌آیند. این فرمولها، مجموعه‌ای مناسب از قواعد انتگرال‌گیری در ارتباط با تغییر درجه چند جمله‌ای‌های درونیاب می‌باشند. مخصوصاً سه فرمول اول به همراه درجه چند جمله‌ای که می‌تواند یک، دو یا سه باشد با اهمیت هستند و در قسمت‌های زیر به تفصیل آنها را بررسی می‌کنیم.

اگر درجه چند جمله‌ای خیلی بالا باشد، خطاهای ناشی از گرد کردن و بی‌نظمی‌های موضعی می‌توانند مشکل ساز شوند. این موضوع، علت اینکه چرا تنها از درجات پایین فرمول‌های نیوتن- کوتز استفاده می‌شود را بیان می‌کند.

دامنه چند جمله‌ای و فاصله انتگرال گیری لزوماً با هم برابر نیستند. اگر فاصله انتگرال گیری، به خارج از محدوده توسعه یابد، برازش در حین برونیابی با خطاهای بزرگتری مواجه می‌شود. اگر می‌خواهیم تنها از تابعی که مقادیرش معلوم است، انتگرال بگیریم، طبیعتاً از برونیابی اجتناب می‌کنیم.

همانطور که در بخش بعد خواهیم دید، انتگرال گیری از چند جمله‌ایها، خارج از محدوده برازش آنها به برخی از روش‌های مهم برای حل معادلات دیفرانسیل منتهی می‌شود. استفاده از فاصله انتگرال گیری که زیر مجموعه نقاطی است که در آنها، چند جمله‌ای با تابع سازگار است، نیز کاربردهای مخصوصی در حل عددی معادلات دیفرانسیل دارد.

نیاز به استفاده از انتگرال‌گیری عددی یک تابع به شکل جدولی از مقادیر، گسترده‌تر از انتگرال گیری یک تابع معلوم می‌باشد. بیشتر کامپیوترها برای انتگرال گیری از تابعی معین بجای روش‌های

تحلیلی حساب دیفرانسیل و انتگرال، با استفاده از فنون عددی برنامه‌ریزی شده‌اند. اگرچه امکان نوشتن برنامه‌ای که نمادها را بکار گیرد و انتگرال تحلیلی را برای اشکال ساده توابع مقدماتی حساب کند وجود دارد اما این روشها فراگیر نیستند و علاوه براین، در تعداد زیادی از حالات، تابع اولیه برای انتگرال وجود ندارد. انتگرال گیری عددی علی رغم پیچیدگی تابع اولیه یا وجود شکل خاصی برای انتگرال تابع، مورد استفاده قرار می‌گیرد.

حال سه فرمول مهم نیوتن- کوتز را بسط می‌دهیم. در طی انتگرال گیری، نیاز به تغییر متغیر x و s را داریم، زیرا چند جمله‌ایها بصورت جملاتی از s بیان شده‌اند.

با استفاده از فرمول نیوتن - گریگوری و تغییر متغیر $S=\frac{x-x_0}{h}$ مشاهده می‌کنید که $dx=hds$

برای $n = 1$:

$$\begin{aligned}\int_{x_0}^{x_1} f(x)\,dx &\doteq \int_{x_0}^{x_1} (f_0 + s\Delta f_0)\,dx = h\int_{s=0}^{s=1} (f_0 + s\Delta f_0)\,ds \\ &= hf_0 s\Big]_0^1 + h\Delta f_0 \frac{s^2}{2}\Big]_0^1 = h\left(f_0 + \frac{1}{2}\Delta f_0\right) \\ &= \frac{h}{2}[(2f_0 + (f_1 - f_0)] = \frac{h}{2}(f_0 + f_1).\end{aligned} \tag{2.7}$$

فرمول (2.7) بنام ذوزنقه ساده می‌باشد

$$\text{خطا} \quad = \int_{x_0}^{x_1} \frac{s(s-1)}{2}h^2 f''(\xi)\,dx = h^3 f''(\xi_1)\int_0^1 \frac{s^2 - s}{2}ds \tag{3.7}$$

$$= h^3 f''(\xi_1)\left(\frac{s^3}{6} - \frac{s^2}{4}\right)\Big]_0^1 = -\frac{1}{12}h^3 f''(\xi_1), \qquad x_0 \le \xi_1 \le x_1.$$

در بدست آوردن خطای معادله (3.7) لازم است از قضیه مقدار میانگین برای انتگرالها استفاده کنیم. که عبارتست از:

$$\int_a^b f(x)g(x)\,dx = f(\xi)\int_a^b g(x)\,dx, \qquad a \le \xi \le b,$$

مشروط بر اینکه $g(x)$ در فاصله $(b$ و $a)$ تغییر علامت ندهد، این نتیجه در معادله (3.7) مورد استفاده قرار گرفته، زیرا $s\ (s-1)$ در فاصله $(1$ و $0)$ همواره کوچکتر یا مساوی صفر می‌باشد. مقدار ξ_1 ضرورتاً مساوی ξ نیست، اگرچه هر دو در فاصله $(x_1$ و $x_0)$ قرار دارند.

برای $n = 2$ داریم:

$$\begin{aligned}\int_{x_0}^{x_2} f(x)\,dx &\doteq \int_{x_0}^{x_2} \left(f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0\right)dx \\ &= h\int_0^2 \left(f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0\right)ds \\ &= hf_0 s\Big]_0^2 + h\Delta f_0\frac{s^2}{2}\Big]_0^2 + h\Delta^2 f_0\left(\frac{s^3}{6} - \frac{s^2}{4}\right)\Big]_0^2 \\ &= h\left(2f_0 + 2\Delta f_0 + \frac{1}{3}\Delta^2 f_0\right) = \frac{h}{3}(f_0 + 4f_1 + f_2).\end{aligned} \tag{4.7}$$

(بنام روش سیمپسون ساده می‌باشد.)

وقتی از جمله آخر برای بدست آوردن خطا، انتگرال بگیریم، داریم:

$$\int_{x_0}^{x_2} \frac{s(s-1)(s-2)}{6}\Delta^3 f_0\, dx = 0,$$

بنابراین برای بدست آوردن خطا، باید از عبارت بعدی انتگرال بگیریم:

$$\text{خطا} \quad r = \int_{x_0}^{x_2} \frac{s(s-1)(s-2)(s-3)}{24} h^4 f^{iv}(\xi)\, dx \qquad (5.7)$$

$$= -\frac{1}{90} h^5 f^{iv}(\xi_1), \qquad x_0 \le \xi_1 \le x_2.$$

جزئیات انتگرال گیری در معادله (5.7) کاملا روشن است.

بطور مشابه برای $n = 3$ خواهیم داشت:

$$\int_{x_0}^{x_3} f(x)\, dx = \int_{x_0}^{x_3} P_3(x_s)\, dx = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3). \qquad (6.7)$$

$$\text{خطا} \quad = -\frac{3}{80} h^5 f^{iv}(\xi_1), \qquad x_0 \le \xi_1 \le x_3. \qquad (7.7)$$

(روش $\frac{3}{8}$ سیمپسون)

بطور خلاصه فرمول‌های نیوتن- کوتز عبارتند از:

$$\int_{x_0}^{x_1} f(x)\, dx = \frac{h}{2}(f_0 + f_1) - \frac{1}{12} h^3 f''(\xi)$$

$$\int_{x_0}^{x_2} f(x)\, dx = \frac{h}{3}(f_0 + 4f_1 + f_2) - \frac{1}{90} h^5 f^{iv}(\xi)$$

$$\int_{x_0}^{x_3} f(x)\, dx = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) - \frac{3}{80} h^5 f^{iv}(\xi)$$

مطلب مهم دیگر، این‌است که جملات خطا برای هر دو حالت $n=2$ و $n = 3$ ، $O(h^5)$ می‌باشد. باین معنی که خطای انتگرال گیری از چند جمله‌ای درجه 2 شبیه انتگرال گیری از چند جمله‌ای درجه 3 می‌باشد. همچنین توجه کنید که ضریب $(-\frac{1}{90})$ در معادله (5.7) کوچکتر از نظیرش در معادله (7.7) یعنی $(-\frac{3}{80})$ می‌باشد. فرمول براساس چند جمله‌ای درجه 2، به شکل غیر منتظره‌ای دقیق است. این پدیده واقعیت حاکم بر تمام فرمول های نیوتن- کوتز مرتبه زوج می‌باشد، هر کدام مرتبه‌ای از h را در جمله خطای خود دارند که برابر همان خطا در فرمول مرتبه بالاتر بعدی می‌باشد.

۲.۷ روش ذوزنقه‌ای

اولین فرمول نیوتن- کوتز، بر مبنای تقریب $f(x)$ روی $(x_0 و x_1)$ بوسیله خطی مستقیم، روش

ذوزنقه‌ای نامیده می‌شود. این روش را از انتگرال گیری از $P_1(x_s)$ بدست آورده‌ایم، اما روش آشنا و آسان ذوزنقه‌ای را می‌توان از تعریف انتگرال‌های معین به شکل جمع، بدست آورد. جهت محاسبه $\int_a^b f(x)dx$، فاصله از a تا b را به n زیر فاصله تقسیم می‌کنیم. مانند شکل ۱.۷، مساحت زیر منحنی در هر زیر فاصله برابر مساحت ذوزنقه است که از جایگزینی منحنی با خط وتر واصل بین نقاط انتهایی منحنی بدست می‌آید. سپس انتگرال از مجموع مساحت تمام ذوزنقه‌ها بطور تقریبی بدست خواهد آمد. اگر مقدار حد این مجموع را وقتی عرض فاصله به سمت صفر میل می‌کند، بدست آوریم، مقدار دقیق انتگرال معین می‌شود. اما در انتگرال عددی فاصله‌ها معین و محدود می‌باشند. نیاز نیست که زیر فاصله‌ها برابر باشند، اما فرمولها، آسانتر می‌شوند اگر زیر فاصله‌ها برابر باشند، h را برابر ثابت Δx در نظر می‌گیریم. از آنجا که مساحت ذوزنقه برابر با ارتفاع متوسط ضربدر روش می‌باشد، برای هر زیر فاصله داریم:

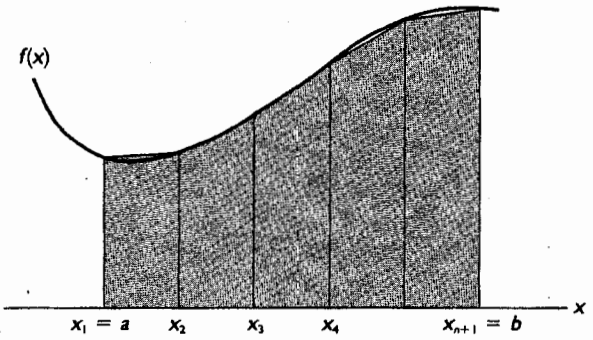
$$\int_{x_i}^{x_{i+1}} f(x)\,dx \doteq \frac{f(x_i) + f(x_{i+1})}{2}(\Delta x) = \frac{h}{2}(f_i + f_{i+1}), \qquad (8.7)$$

و برای $[b$ و $a]$ پس از تقسیم شدن به زیر فاصله‌هائی به طول h داریم:

$$\int_a^b f(x)\,dx \doteq \sum_{i=1}^{n} \frac{h}{2}(f_i + f_{i+1}) = \frac{h}{2}(f_1 + f_2 + f_2 + f_3 + \cdots + f_n + f_{n+1});$$

$$\int_a^b f(x)\,dx = \frac{h}{2}(f_1 + 2f_2 + 2f_3 + \cdots + 2f_n + f_{n+1}). \qquad (9.7)$$

معادله (9.7) هم‌ارز معادله (2.7) می‌باشد. معادله (9.7) قانون ذوزنقه مرکب نامیده می‌شود. این قانون به ما اجازه می‌دهد با اعمال روش مذکور در زیر فاصله‌هائی که در آنها امکان تقریب تابع با قطعات خطی وجود دارد، فرمول را برای ناحیه وسیعی که $f(x)$ در آن غیر خطی می‌باشد بکار بریم. فرمول بطرز جالبی ساده می‌باشد و برای محاسبه انتگرال یک تابع که بصورت تجربی برای مقادیری که با هم فاصله مساوی ندارند، معین شده است، استفاده می‌شود. از شکل ۱.۷ مشهود است که این روش در معرض خطاهای بزرگی قرار دارد، مگر آنکه زیر فاصله‌ها کوچک باشند، چرا که جایگزینی یک منحنی با یک خط مستقیم، چندان دقیق نمی‌باشد.



شکل ۱.۷

مثال ۱.۷ فرض کنید می‌خواهیم از تابع لیست شده در جدول ۱.۷ روی فاصله $x = 3.4$ تا $x = 1.8$ انتگرال بگیریم. از قانون ذوزنقه مرکب نتیجه می‌شود:

$$\int_{1.8}^{3.4} f(x)\,dx \doteq \frac{0.2}{6}[6.050 + 2(7.389) + 2(9.025) + 2(11.023) + 2(13.464) + 2(16.445) + 2(20.086) + 2(24.533) + 29.964] = 23.9944.$$

داده‌ها در جدول ۱.۷ برای تابع $f(x)=e^x$ می‌باشند، بنابراین مقدار تحقیقی انتگرال برابر است با: $e^{3.4}-e^{1.8} = 23.9144$ که در رقم دوم اعشار، خطا داریم.

قبلاً خطای قانون ذوزنقه را در معادله (3.7) نشان دادیم. مجدداً می‌نویسیم:

خطای موضعی برای قانون ذوزنقه [1] $L = -\frac{1}{12}h^3 f''(\xi_1), \quad x_0 \le \xi_1 \le x_1.$

جدول ۱.۷

x	$f(x)$	x	$f(x)$
1.6	4.953	2.8	16.445
1.8	6.050	3.0	20.086
2.0	7.389	3.2	24.533
2.2	9.025	3.4	29.964
2.4	11.023	3.6	36.598
2.6	13.464	3.8	44.701

باید تاکید شود که این خطا، تنها خطای مربوط به یک مرحله می‌باشد و از اینرو خطای موضعی نام گرفته است. طبیعتاً فرمول ذوزنقه را برای یک سری از زیر فاصله‌ها بکار می‌بریم تا انتگرال را روی یک فاصله بزرگ از $x=a$ تا $x=b$ بدست آوریم. ما به خطای کل نیاز داریم که خطای ناحیه‌ای نامیده می‌شود، خطای ناحیه‌ای [2] قانون ذوزنقه، برابر مجموع خطاهای موضعی می‌باشد:

خطای ناحیه‌ای $G = -\frac{1}{12}h^3[f''(\xi_1) + f''(\xi_2) + \cdots + f''(\xi_n)].$ (10.7)

در معادله (10.7) هر یک از مقادیر ξ در یکی از n زیر فاصله متوالی قرار دارد. اگر فرض کنیم $f''(x)$ روی $(a$ و $b)$ پیوسته است، مقادیری از x در $(a$ و $b)$ وجود دارد بطوری که $x=\xi$. و مقدار مجموع در معادله (10.7) برابر $nf''(\xi)$ می‌باشد از آنجا برای $nh = b - a$، خطای ناحیه‌ای برابر است با:

خطای کل (ناحیه‌ای قانون ذوزنقه) $= -\frac{1}{12}h^3 nf''(\xi) = \frac{-(b-a)}{12}h^2 f''(\xi) = O(h^2).$ (11.7)

خطای ناحیه‌ای $O(h^2)$ می‌باشد، در حالیکه خطای موضعی $O(h^3)$ می‌باشد، که واقعیتی معقول

۱- LOCAL ۲- GLOBAL

است، زیرا برای مثال اگر h نصف شود، تعداد زیر فاصله‌ها دو برابر می‌شود، بنابراین دو بار خطاها با هم جمع می‌شوند.

وقتی تابع $f(x)$ معلوم است، معادله *(11.7)* اجازه می‌دهد تا خطای انتگرال گیری عددی را با استفاده از قانون ذوزنقه، حساب کنیم. در استفاده از این معادله، خطا را با محاسبه مقادیر ماکزیمم و مینیمم $f''(x)$ روی فاصله $[agb]$ دسته بندی می‌کنیم. برای مثال بالا، از رابطه خطا، این نتایج را بدست می‌آوریم.

$$\text{خطا} = -\frac{1}{12}h^3 n f''(\xi), \qquad 1.8 \le \xi \le 3.4,$$

$$= -\frac{1}{12}(0.2)^3(8)\begin{Bmatrix} e^{1.8} & (\min) \\ e^{3.4} & (\max) \end{Bmatrix} = \begin{Bmatrix} -0.0323 & (\min) \\ -0.1598 & (\max) \end{Bmatrix}.$$

بهمین ترتیب،

$$\text{خطا} = -\frac{1}{12}(0.2)^2(3.4 - 1.8)\begin{Bmatrix} e^{1.8} & (\min) \\ e^{3.4} & (\max) \end{Bmatrix} = \begin{Bmatrix} -0.0323 \\ -0.1598 \end{Bmatrix}.$$

خطای واقعی برابر -0.080 می‌باشد.

اگر شکل تابعی که مقادیر جدولی آن را در اختیار داریم برای ما شناخته شده نباشد، باید $h^2 f''(\xi)$ را از تفاضل‌های مرتبه دوم تخمین بزنیم.

اگر روش ذوزنقه را با $h = 0.2$ و یکبار دیگر با $h = 0.4$ بکار بریم، خواهیم داشت:

خطا + مقدار محاسبه شده = مقدار واقعی

$$\text{مقدار واقعی} = 23.9944 + C(0.2)^2$$

$$\text{مقدار واقعی} = 24.2328 + C(0.4)^2 \qquad (12.7)$$

از حل این دو معادله مقدار C بدست می‌آید و خواهیم داشت:

$$\text{مقدار واقعی} = 23.9944 + \frac{1}{3}(23.9944 - 24.2328)$$

$$= 23.9149 \quad (\text{در برابر مقدار دقیق } 23.9144) \qquad (13.7)$$

یعنی از دو مقدار غیر دقیق، یک مقدار اصلاح شده را برونیابی کرده‌ایم، آنچه را که مقدار واقعی نامیده‌ایم، دقیق نمی‌باشد، زیرا خطا را به فرض $O(h^2)$ برابر Ch^2 و $O(h^4)$ را برابر Ch^4 در نظر گرفته‌ایم. در روش رامبرگ در این مورد توضیح بیشتر خواهیم داد.

۳.۷ روش سیمپسون[۱]

فرمول‌های نیوتن-کوتز، بر اساس چند جمله‌ای‌های درجه دو و سه بصورت وسیعی بکار می‌روند

۱- Simpson

و در حالت‌های مرکب به عنوان قوانین سیمپسون شناخته می‌شوند. دو حالت وجود دارد، فرمول نیوتن- کوتز، انتگرال یک چند جمله‌ای درجه دوم را روی فواصل $2\Delta x$ با عرض‌های برابر، محاسبه می‌کند. (اینگونه فواصل اغلب پانل نامیده می‌شوند.)

دوباره این معادله را از بخش *(1.7)* تکرار می‌کنیم:

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}(f_0 + 4f_1 + f_2) - \frac{h^5}{90}f^{\text{iv}}(\xi), \qquad x_0 \le \xi \le x_2. \tag{14.7}$$

این روش $\frac{1}{3}$ سیمپسون، دارای خطای موضعی $O(h^5)$ می‌باشد و بسیار متداول است. اگر این روش را برای جفت پانل‌های متوالی برای محاسبه $\int_a^b f(x)dx$ بکار بریم، بدست می‌آید:

$$\int_a^b f(x)\,dx = \frac{h}{3}(f_1 + 4f_2 + 2f_3 + 4f_4 + 2f_5 + \cdots + 2f_{n-1} + 4f_n + f_{n+1}) - \frac{(b-a)}{180}h^4 f^{\text{iv}}(\xi), \qquad x_1 \le \xi \le x_{n+1}. \tag{15.7}$$

در معادله *(15.7)* مقادیر x را به $x_1 = a$ تا $x_{n+1} = b$ اندیس گذاری کرده‌ایم. برای بدست آوردن خطای ناحیه‌ای مجموع حداکثر خطای موضعی را حساب می‌کنیم $\sum f^{IV}(\zeta_i)=nf^{IV}(\zeta k)$، که با فرض شرط پیوستگی مشتق مرتبه چهارم است. در اینجا n، تعداد دفعاتی است که قانون موضعی بکار برده شده است، این فقط نصف تعداد گام‌های Δx می‌باشد، زیرا انتگرال در هر بار روی دو پانل بدست می‌آید، این سبب می‌شود مخرج کسر از *90* به *180* تغییر کند. توجه کنید که قانون $\frac{1}{3}$ سیمپسون دارای خطای ناحیه‌ای $O(h^4)$ می‌باشد. لازم است که در زیر تقسیمات تعداد پانل زوج باشد.

مثال ۲.۷ قانون $\frac{1}{3}$ سیمپسون را برای داده‌های جدول ۱.۷ در محاسبه $\int_{1.8}^{3.4} f(x)\,dx$ بکار برید.

$h = 0.2$ را بکار برده، بدست می‌آوریم:

$$\begin{aligned}\int_{1.8}^{3.4} f(x)\,dx \doteq \frac{0.2}{3}[&6.050 + 4(7.389) + 2(9.025) + 4(11.023)\\ &+ 2(13.464) + 4(16.445) + 2(20.086)\\ &+ 4(24.533) + 29.964]\\ &= 23.9149.\end{aligned}$$

تضمینی وجود ندارد که این مقدار برابر مقدار برونیابی از قانون ذوزنقه برای $h=0.2$ و $h=0.4$ باشد. نشان می‌دهیم از این دو طریق نتیجه برابری بدست می‌آید.

$$\text{خطا} = \frac{-(3.4-1.8)}{180}(0.2)^4\begin{Bmatrix} e^{1.8} & (\min) \\ e^{3.4} & (\max)\end{Bmatrix} = \begin{Bmatrix} -8.6\times10^{-5} \\ -4.3\times10^{-4}\end{Bmatrix}.$$

خطای واقعی برابر است با $-5*10^{-4}$ و خطای برش، قدر مطلق آن است. یک برونیابی می‌تواند برای این مثال ساخته شود. $h=0.4$ را بکار می‌بریم، بدست می‌آوریم:

$$\int_{1.8}^{3.4} f(x)\,dx \doteq \frac{0.2}{3}[6.050 + 4(9.025) + 2(13.464) + 4(20.086) + 29.964]$$
$$= 23.9181.$$

بطور مشابه، با $h = 0.8$ نتیجه 23.9653 بدست می‌آید. نتایج بدست آمده در جدول زیر خلاصه شده است.

	تخمین قانون سیمپسون	مقدار برونیابی
h	خطاها $O(h^4)$	خطاها $O(h^6)$
0.2	23.9149	
		23.9147
0.4	23.9181	
		23.9166
0.8	23.9653	

با استفاده از فرمول نیوتن گریگوری و محاسبه خطا از قانون جمله بعدی جمله خطا برای قانون سیمپسون می‌تواند بصورت زیر نوشته شود:

$c_1h^4 + c_2h^6 + c_3h^8 + \ldots$

سومین فرمول نیوتن- کوتز که استفاده فراوان دارد با انتگرال چند جمله‌ای درونیاب درجه سوم روی دامنه‌ای برابر سه پانل بدست می‌آید. قبلاً فرمول را در معادله *(6.7)* و جمله خطا را در معادله *(7.7)* بدست آورده‌ایم. خطای موضعی برابر است.

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) - \frac{3}{80}h^5 f^{\text{iv}}(\xi), \qquad x_0 \le \xi \le x_3.$$

مقدار ضریب $\frac{3}{8}h$، نام روش سیمپسون، رابه آن می‌دهد. وقتی انتگرال روی فاصله‌ای از x_1 تا $x_{n+1} = b$ بکار می‌رود فاصله باید به مضاربی از 3 پانل تقسیم شود.

$$\int_a^b f(x)\,dx = \frac{3h}{8}(f_1 + 3f_2 + 3f_3 + 2f_4 + 3f_5 + 3f_6 + \cdots + 2f_{n-2} + 3f_{n-1} + 3f_n + f_{n+1}) - \frac{(b-a)}{80}h^4 f^{\text{iv}}(\xi), \qquad x_1 \le \xi \le x_{n+1}. \qquad (16.7)$$

قانون موضعی دارای یک ضریب $\frac{1}{80}$ در جمله خطاست، که کاهش خطای محلی $\frac{3}{80}$ است. زیرا وقتی n پانل وجود دارد فاصله را سه قسمت می‌کنیم و 3 پانل وجود دارد.

همانطورکه قبلاً دیده‌ایم، مرتبه خطای موضعی $O(h^4)$ بیشتر از خطا در قانون $\frac{1}{3}$ سیمپسون نبوده و ضریب بزرگتر می‌باشد. ممکن است مردد شویم که آیا همیشه روش $\frac{3}{8}$ سیمپسون را باید بکار برد؟

همانطورکه قبلاً دیده‌ایم، روش $\frac{1}{3}$ سیمپسون ایجاب می‌کند که فاصله به تعدادی پانل زوج

تقسیم شود. این امر بهنگام استفاده در داده‌های جدول بندی شده غیر ممکن است. در چنین مواردی با استفاده از روش $\frac{1}{3}$ تا سرحد امکان و تکمیل انتگرال گیری از طریق انتخاب سه پانل با روش $\frac{3}{8}$ می‌توان دو روش را با هم ترکیب کرد. یک روش دیگر برای یک تعداد فرد از پانلها این است که انتگرال گیری روی یک پانل با روش ذوزنقه‌ای انجام شود. بهر حال، حاصل آن یک خطای $O(h^2)$ روی آن زیر فاصله بوده و دقت آن تا حدودی کمتر می‌باشد.

مثال ۳.۷ با استفاده از داده‌های جدول ۱.۷، انتگرال را از $x = 1.6$ تا $x = 3.4$ پیدا کنید.

در اینجا 9 پانل وجود دارد، پس نمی‌توانیم روش $\frac{1}{3}$ را بطور مستقیم بکار ببریم. برای حل این مسئله چندین راه داریم: بکار گیری روش ذوزنقه‌ای، بکار گیری روش $\frac{3}{8}$ سیمپسون، یا ترکیب روش $\frac{1}{3}$ سیمپسون با هر یک از روش‌های دیگر. هر یک از انتخاب‌ها را بررسی می‌کنیم.

با روش ذوزنقه‌ای داریم:

$$\int_{1.6}^{3.4} f(x)dx = 25.0947 \qquad \text{خطا} = -0.0836$$

با روش $\frac{3}{8}$ سیمپسون:

$$\int_{1.6}^{3.4} f(x)dx = 25.0119 \qquad \text{خطا} = -0.0008$$

با روش $\frac{1}{3}$ سیمپسون از 1.8 تا 3.4 و روش ذوزنقه‌ای از 1.6 تا 1.8:

$$\int_{1.6}^{3.4} f(x)dx = 25.0152 \qquad \text{خطا} = -0.0041$$

با روش $\frac{1}{3}$ سیمپسون از 2.2 تا 3.4 و روش $\frac{3}{8}$ از 1.6 تا 2.2:

$$\int_{1.6}^{3.4} f(x)dx = 25.0115 \qquad \text{خطا} = -0.0004$$

نتایج همان است که انتظارش را داشتیم. صحیح‌ترین نتیجه وقتی حاصل می‌شود که دو روش $\frac{1}{3}$ و $\frac{3}{8}$ با هم ترکیب می‌شوند. وقتی که روش‌ها مختلف را با هم ترکیب می‌کنیم، هنوز انتخاب دیگری وجود دارد؛ در کدام یک از سه پانل باید از روش $\frac{3}{8}$ استفاده نمائیم؟ با ملاحظه عبارت خطا در روش $\frac{3}{8}$ جواب بدست می‌آید و آن را در جائی که خطا حداقل می‌باشد بکار می‌گیریم. این بدین معنی است که باید سه پانل را در جایی انتخاب کنیم که مشتق چهارم تابع دارای کوچکترین مقدار باشد. در این مثال تابع $f(x) = e^x$ بهترین انتخاب را برای داده‌ها در نظر گرفته‌ایم، هنگامی که $f(x)$ را تنها به عنوان جدول از مقادیر می‌شناسیم، این انتخاب همیشه روشن نخواهد بود، بهر حال اختلافات جزئی خواهند بود و خطاهای ناشی از برش باعث افزایش خطا می‌گردد.

در اینجا برای سهولت، فرمول‌های انتگرال گیری نیوتن - کوتز را گردآوری می‌کنیم. در

قسمت بعد روش‌های دیگر انتگرال گیری را با استفاده از این فرمولها ملاحظه خواهیم کرد.

فرمولهایی برای انتگرال گیری ($\Delta x=h$ فواصل مساوی)

روش ذوزنقه‌ای

$$\int_a^b f(x)\,dx = \frac{h}{2}(f_1 + 2f_2 + 2f_3 + \cdots + 2f_n + f_{n+1}) - \frac{(b-a)}{12}h^2 f''(\xi), \qquad a \le \xi \le b.$$

روش $\frac{1}{3}$ سیمپسون

$$\int_a^b f(x)\,dx = \frac{h}{3}(f_1 + 4f_2 + 2f_3 + 4f_4 + 2f_5 + \cdots + 4f_n + f_{n+1}) - \frac{(b-a)}{180}h^4 f^{iv}(\xi), \qquad a \le \xi \le b.$$

در روش $\frac{1}{3}$ سیمپسون تعداد پانل زوج است.

$$\int_a^b f(x)\,dx = \frac{3h}{8}(f_1 + 3f_2 + 3f_3 + 2f_4 + 3f_5 + 3f_6 + \cdots + 3f_n + f_{n+1}) - \frac{(b-a)}{80}h^4 f^{iv}(\xi), \qquad a \le \xi \le b.$$

در روش $\frac{3}{8}$ سیمپسون

تعداد پانلها تقسیم پذیر بر 3 می‌باشد.

بطور مشابه خواهیم داشت:

$$E = + \frac{h^4}{180}(b-a) \left| f^{(4)}(\xi) \right| \qquad (a \le \xi \le b)$$

از روابط فوق می‌توان خطای حاصل را پیش بینی نموده و با انتخاب مناسب h رابطه مورد نظر را بکار برد.

$$E = + n \frac{h^5}{180} \left| f^{(4)}(\xi) \right|$$

مشاهده می‌شود که با افزایش n مقدار h کاهش می‌یابد. لیکن خطای محاسبات ممکن بدلیل افزایش محاسبات و انباشتگی خطا افزایش یابد.

مثال ۴.۷ با بکار بردن روش مرکب سیمپسون انتگرال زیر را محاسبه کنید.

$$I = \int_0^1 e^x\,dx \qquad n=4$$

$$h = \frac{1-0}{4} = .25$$

ابتدا h را تعیین می‌کنیم.

$$h = \int_0^1 e^x dx = \frac{.25}{3}\left[e^0 + 4e^{.25} + 2e^{.5} + 4e^{.75} + e^1\right]$$

$$I = 1.71832$$

$$E = +\frac{h^4}{180}(b-a)\left|f^{(4)}(\xi)\right|$$

$$E = +\frac{(.25)^4}{180}(1-0)\,e^{\xi} = \frac{1}{4^4 \times 180}\,e^{\xi}$$

$$E \le +\frac{1}{4^4 \times 180}\,e^{1}$$

$$E \le 2 \times 10^{-5} \times e$$

$$E \le 0.55 \times 10^{-4}$$

در این روش لازم است که مشتق چهارم محاسبه شده و ماکزیمم آن در فاصله *[a,b]* بدست آید. چون این عمل با ماشین محاسب ساده نخواهد بود، می‌توان آنرا عیبی برای این روش دانست. در عمل فرمول را بکار برده و مقدار تقریبی انتگرال را برای مقادیر مختلف h که بطور نزولی انتخاب می‌شوند، محاسبه کرده و نتایج را به عنوان روش سازگاری آزمایش می‌کنیم. زمانی که تفاضل دو مقدار متوالی انتگرال I از خطای قابل اغماض کوچکتر است عملیات را خاتمه می‌دهیم.

مثال ۵.۷ انتگرال زیر را با در نظر گرفتن توضیح فوق محاسبه کنید.

$$I = \int_0^1 e^x dx$$

بدین منظور از روش سیمپسون استفاده کرده، و انتگرال را در حالت‌های زیر محاسبه می‌کنیم.

$h = .5$	$I = 1.7189$
$h = .25$	$I = 1.7183$
$h = .125$	$I = 1.7183$

در اینصورت روش سازگار است و $I = 1.7183$

۴.۷ روش انتگرال گیری رامبرگ[1]

مقدار تقریبی انتگرال $I = \int_a^b f(x)\,dx$ با بکاربردن روش ذوزنقه‌ای مورد بررسی قرار می‌دهیم.

$$T_n = h\,(\tfrac{1}{2} f_0 + f_1 + f_2 + \ldots + \tfrac{1}{2} f_n) \qquad (1)$$

$$h = \frac{b-a}{n} \quad , \quad f_k = f(a + kh) \quad , \quad k = 0,1,2,\ldots,n$$

1- Romberg

اکنون مشروط براینکه $f(x)$«بطور کافی خوش رفتار»[1] باشد، درفاصله $[a,b]$ با استفاده از فرمول نیوتن - گریگوری و محاسبه خطا از قانون جمله بعدی می‌توان نشان داد که خطابرابر است با:

$$\text{خطا} = I - T_n = c_1 h^2 + c_2 h^4 + c_3 h^6 + c_4 h^8 + \ldots \quad (2)$$

[۲]

$c_1, c_2, c_3, \ldots$ مستقل از h می‌باشند.

چنانچه تعداد زیر فاصله‌ها را دو برابر کنیم در اینصورت خواهیم داشت:

$$I - T_{2n} = c_1(\frac{h}{2})^2 + c_2(\frac{h}{2})^4 + c_3(\frac{h}{2})^6 \quad (3)$$

از حذف اولین جمله سمت راست در روابط (2) و(3) خواهیم داشت:

$$I - \frac{4T_{2n} - T_n}{3} = d_1 h^4 + d_2 h^6 + d_3 h^8 + \ldots \quad (4)$$

که تقریب آن $O(h^4)$ می‌باشد. ضرایب جدید d_1 و d_2 و d_3... ترکیبی از ضرایب قبلی c_1 و c_2 و c_3 می‌باشند.

$$T_{2n}^{(1)} = \frac{4T_{2n} - T_n}{3} \quad (5)$$

مقدار تقریبی جدید انتگرال برابر است با:

با توجه به T_n و T_{2n} اگر $T_{2n}^{(1)}$ را مستقیماً بدست آوریم می‌بینیم که همان روش سیمپسون حاصل خواهد شد.

$$T_n = h\,(\tfrac{1}{2}f_0 + f_2 + f_4 + \ldots + \tfrac{1}{2}f_n)$$

$$T_{2n} = \frac{h}{2}\,(\tfrac{1}{2}f_0 + f_1 + f_2 + f_3 + \ldots + \tfrac{1}{2}f_{2n})$$

$$T_{2n}^{(1)} = \frac{4T_{2n} - T_n}{3} = \frac{1}{3}\,(\frac{h}{2})(f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 4f_{2n-1} + f_{2n})$$

چنانچه T_{4n} را محاسبه کنیم، در اینصورت خواهیم داشت:

$$T_{4n}^{(1)} = \frac{4T_{4n} - T_{2n}}{3} \quad (6)$$

$$I - T_{2n}^{(1)} = d_1 h^4 + d_2 h^6 + d_3 h^8 + \ldots \quad (7)$$

از رابطه (4) داریم:

$$I - T_{۲n}^{(۱)} = d_۱ \left(\frac{h}{۲}\right)^۴ + d_۲ \left[\frac{h}{۲}\right]^۶ + d_۳ \left[\frac{h}{۲}\right]^۸ + \cdots \quad (8)$$

از رابطه (7) و(8) با حذف اولین جمله سمت راست خواهیم داشت:

$$I - \frac{16\,T_{4n}^{(1)} - T_{2n}^{(1)}}{15} = e_1 h^6 + e_2 h^8 + e_3 h^{10} + \ldots \quad (9)$$

$$T_{4n}^{(2)} = \frac{16\,T_{4n}^{(1)} - T_{2n}^{(1)}}{15} \quad (10)$$

که مقدار تقریبی انتگرال را با خطای $O(h^6)$ نشان می‌دهد.

1- well Behavior

۲ – ۳. Davis and Rabinowitz N. (1967) مراجعه کنید

چنانچه عملیات را ادامه دهیم با محاسبه T_{8n} خواهیم داشت:

$$T_{8n}^{(1)} = \frac{4T_{8n} - T_{4n}}{3} \qquad (11)$$

$$T_{8n}^{(2)} = \frac{16T_{8n}^{(1)} - T_{4n}^{(1)}}{15} \qquad (12)$$

$$I - T_{4n}^{(2)} = e_1 h^6 + e_2 h^8 + e_3 h^{10} + \ldots \qquad (13)$$

$$I - T_{8n}^{(2)} = e_1 \left[\frac{h}{2}\right]^6 + e_2 \left[\frac{h}{2}\right]^8 + e_3 \left[\frac{h}{2}\right]^{10} + \ldots \qquad (14)$$

$$I - \frac{64\,T_{8n}^{(2)} - T_{4n}^{(2)}}{63} = g_1 h^8 + g_2 h^{10} + g_3 h^{12} \ldots \qquad (15)$$

g_k مقداری است ثابت.

$$T_{8n}^{(3)} = \frac{64\,T_{8n}^{(2)} - T_{4n}^{(2)}}{63} \qquad (16)$$

که مقدار تقریبی انتگرال با خطای $O(h^8)$ را نشان می‌دهد.

عناصر قطر جدول زیر که بنام جدول رامبرگ مشهور است، بسرعت بسمت مقدار انتگرال I همگراست.

T_n

T_{2n} $T_{2n}^{(1)}$

T_{4n} $T_{4n}^{(1)}$ $T_{4n}^{(2)}$

T_{8n} $T_{8n}^{(1)}$ $T_{8n}^{(2)}$ $T_{8n}^{(3)}$

معمولاً $n = 1$ یا $h \leq 1$ را در شروع عملیات انتخاب می‌کنیم مگر آنکه فاصله انتگرال گیری زیاد و مشتق مکرراً تغییر علامت دهد. روش رامبرگ برای تابع خوش‌رفتار $f(x)$ یکی از روشهائی است که بسیار قابلیت کاربرد دارد و بسیار کم هزینه می‌باشد.

مثال ۶.۷ انتگرال $I = \int_0^1 e^x\,dx$ را با استفاده از روش رامبرگ محاسبه کنید.

$T1 = h(\frac{1}{2} f_0 + \frac{1}{2} f_1) = 1 \times (\frac{1}{2} e^0 + \frac{1}{2} e^1) = 1.85914 \qquad n=1$

$T_2 = \frac{h}{2}(\frac{1}{2} f_0 + f_1 + \frac{1}{2} f_2) = \frac{1}{2} (\frac{1}{2} e^0 + e^{\frac{1}{2}} + \frac{1}{2} e^1) = 1.75343$

$T_4 = \frac{h}{4} (\frac{1}{2} f_0 + f_1 + f_2 + f_3 + \frac{1}{2} f_4) = \frac{1}{4} (\frac{1}{2} e^0 + e^{1/4} + e^{2/4} + e^{3/4} + \frac{1}{2} e^1) = 1.72722$

$$T_8 = \frac{h}{8}(\tfrac{1}{2} f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + \tfrac{1}{2} f_8) =$$

$$T_8 = 1.721852$$

$$T_2^{(1)} = \frac{4T_2 - T_1}{3} = 1.71886$$

$$T_4^{(1)} = \frac{4T_4 - T_2}{3} = 1.71832$$

$$T_8^{(1)} = \frac{4T_8 - T_4}{3} = 1.71829$$

$$T_4^{(2)} = \frac{16T_4^{(1)} - T_2^{(1)}}{15} = 1.71886$$

$$T_8^{(2)} = \frac{16T_8^{(1)} - T_4^{(1)}}{15} = 1.71829$$

$$T_8^{(3)} = \frac{64T_8^{(2)} - T_4^{(2)}}{63} = 1.71829 \quad , \quad I = 171829$$

$$T_1 = 1.85914$$

$$T_2 = 1.75343 \qquad T_2^{(1)} = 1.71886$$

$$T_4 = 1.72722 \qquad T_4^{(1)} = 1.71823 \qquad T_4^{(2)} = 1.71886$$

$$T_8 = 1.72185 \qquad T_8^{(1)} = 1.71829 \qquad T_8^{(2)} = 1.71829 \qquad T_8^{(3)} = 1.71829$$

۵.۷ روش‌های دیگر برای پیدا کردن فرمول‌های انتگرال‌گیری

جالب است که راه‌های دیگری را برای پیدا کردن فرمول‌های انتگرال گیری غیر از روش انتگرال گیری از چند جمله‌ای‌های درونیاب بررسی نمائیم. یک راه این است که از روش‌های سمبولیک شبیه به روش‌های فصل 5 استفاده نمائیم. با استفاده از اپراتور مرحله‌ای E داریم:

$$f(x_s) = f_s = E^s f_0.$$

با ضرب نمودن در $dx = hds$ و انتگرال گیری از x_0 تا x_1 $(s=0$ تا $s=1)$ بدست می‌آوریم:

$$\int_{x_0}^{x_1} f(x)\,dx = h\int_0^1 E^s f_0\,ds = \left[\frac{hE^s}{\ln E} f_0\right]_0^1 = \frac{h(E-1)}{\ln E} f_0$$

فرض کنید $E = 1+\Delta$ و $Ln(1+\Delta)$ را به عنوان سری توانی بسط می‌دهیم:

با تقسیم Δ بر این مجموعه‌ها بدست می‌آوریم:

$$\ln(1+\Delta) = \Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \frac{1}{4}\Delta^4 + \cdots.$$

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h\Delta}{\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \frac{1}{4}\Delta^4 + \cdots} f_0$$
$$= h(f_0 + \frac{1}{2}\Delta f_0 - \frac{1}{12}\Delta^2 f_0 + \frac{1}{24}\Delta^3 f_0 - \cdots). \qquad (17.7)$$

با این فن، ضرایب بطور قابل ملاحظه‌ای آسانتر از انتگرال گیری جمله به جمله در بخش 7.7 بدست می‌آیند.

معادله (17.7) یک فرمول نیوتن- کوتز نیست، مگر اینکه از دو جمله اول استفاده نمائیم و فاصله انتگرال گیری و دامنه، مناسب چند جمله‌ای باشد. هنگامیکه از n جمله استفاده می‌شود، نشان دهنده یک چند جمله‌ای از درجه n می‌باشد، که از x_0 تا x_n صدق کند، ولی فقط از x_0 تا x_1 انتگرال گرفته شده است.

این نتیجه به ویژه در ارتباط با روش‌های معادله دیفرانسیل قابل استفاده می‌باشد.

می‌توانیم فرمول را با انتگرال‌گیری از چند جمله‌ای درونیاب درجه n ام بر روی دو پانل از x_0 تا x_2 بسط دهیم.

$$\int_{x_0}^{x_2} f(x)\,dx = \left[\frac{hE^s}{\ln E} f_0\right]_0^2 = \frac{h(E^2-1)}{\ln E} f_0 = \frac{h(E+1)(E-1)}{\ln E} f_0.$$

فرض می‌کنیم $E=1+\Delta$ و نیز $E-1=\Delta$ و $E+1=2+\Delta$ با تقسیم Δ بر سری حاصل از $Ln(1+\Delta)$ بدست می‌آوریم:

$$\int_{x_0}^{x_2} f(x)\,dx = h(2+\Delta)(f_0 + \frac{1}{2}\Delta f_0 - \frac{1}{12}\Delta^2 f_0 + \frac{1}{24}\Delta^3 f_0 - \cdots)$$
$$= h(2f_0 + \Delta f_0 - \frac{1}{6}\Delta^2 f_0 + \frac{1}{12}\Delta^3 f_0 - \cdots$$
$$+ \Delta f_0 + \frac{1}{2}\Delta^2 f_0 - \frac{1}{12}\Delta^3 f_0 + \cdots)$$
$$= h(2f_0 + 2\Delta f_0 + \frac{1}{3}\Delta^2 f_0 + 0 - \cdots). \qquad (18.7)$$

این روش می‌تواند به آسانی بسط داده شود. دلیل اینکه چرا ممکن است خواستار فرمولهائی مانند معادله‌های (17.7) و (18.7) باشیم این است که از آنها برای تشکیل نمودار لوزی شبیه به شکل ۱.۷ و جدول ۱.۷ استفاده می‌نمائیم. می‌توانیم روی m پانل بر اساس چند جمله‌ای‌های درجه n انتگرال‌گیری کنیم. تشکیل نمودارها خیلی ساده هستند، اگر بیاد داشته باشیم که ضرایب خودشان، تشکیل یک جدول تفاضل می‌دهند که با تفاضل‌های تابع در هم آمیخته است. از آنجا که آنها دارای کاربرد خاصی می‌باشند در اینجا آنها را نمایش نمی‌دهیم.

اکنون روش جالب دیگری از فرمولها را ارائه می‌دهیم که می‌تواند برای وضعیت‌های گوناگون از جمله بسط فرمول‌های انتگرال گیری بکار رود. این روش ضرایب نامعین نامیده می‌شود. فرمول

را بصورت حاصل جمع $n+1$ جمله با ضرایب نامعلوم بیان می‌کنیم و سپس ضرایب را با در نظر گرفتن درستی فرمول برای همه چند جمله‌ای‌های درجه n یا کمتر محاسبه می‌کنیم. در اینجا آن را با پیداکردن روش $\frac{1}{3}$ سیمپسون بوسیله این فن نشان می‌دهیم. انتگرال را بصورت یک مجموع وزندار از سه مقدار تابع در نقاط هم فاصله توضیح می‌دهیم:

$$\int_{-1}^{1} f(x)\,dx = af(-1) + bf(0) + cf(+1). \qquad (19.7)$$

فاصله متقارن انتگرال گیری باعث سهولت محاسبه می‌شود. ما تصریح می‌کنیم که تابع باید در سه فاصله مساوی، دو مقدار پایانی و نقطه میانی محاسبه گردد. از آنجا که فرمول شامل سه جمله می‌باشد، می‌توانیم درستی آن را برای همه چند جمله‌ای‌های درجه 2 یا کمتر در نظر بگیریم. اگر آن صحیح باشد، مسلما برای سه مورد خاص $f(x)=x^2$، $f(x)=x$ و $f(x)=1$ صحیح می‌باشد. دوباره معادله (19.7) را سه بار می‌نویسیم و هر یک از تعاریف $f(x)$را به نوبه خود بکار می‌گیریم.

$$f(x) = 1: \quad \int_{-1}^{1} dx = 2 = a(1) + b(1) + c(1) = a + b + c;$$

$$f(x) = x: \quad \int_{-1}^{1} x\,dx = 0 = a(-1) + b(0) + c(1) = -a + c;$$

$$f(x) = x^2: \quad \int_{-1}^{1} x^2\,dx = \frac{2}{3} = a(1) + b(0) + c(1) = a + c.$$

با حل همزمان این سه معادله $a = \frac{1}{3}$، $b = \frac{4}{3}$، $c = \frac{1}{3}$را بدست می‌آوریم. در اینجا فاصله بین نقاط برابر یک بود و به وضوح انتگرال متناسب با $\Delta x=h$ می‌باشد. سپس قانون $\frac{1}{3}$ سیمپسون را بدست می‌آوریم:

$$\int_{-h}^{h} f(x)\,dx = h\left[\frac{1}{3}f(-h) + \frac{4}{3}f(0) + \frac{1}{3}f(h)\right].$$

۶.۷ روش انتگرال‌گیری گاوس

فرمول‌های قبلی برای انتگرال‌های عددی مبتنی بر مقادیر قابل پیش‌بینی x در فواصل مساوی بودند، این بدین معنی است که مقادیر x قبلاً تعیین شده بودند. پس با یک فرمول از سه جمله، سه پارامتر وجود داشتند و ضرایب (فاکتورهای وزن‌دار) برای هر یک از مقادیر تابع بکار برده می‌شدند. اگر یک فرمول دارای سه پارامتر است با یک چند جمله‌ای درجه دو یعنی یکی کمتر از تعداد پارامترها مطابقت دارد. گاوس مشاهده کرد که اگر این شرط راکه تابع با مقادیر تعیین شده x محاسبه می‌گردد کنار بگذاریم، یک فرمول سه جمله‌ای شامل 6 پارامتر خواهد بود (سه مقدار xکه اکنون مجهول می‌باشند باضافه سه ضریب وزنی) و با یک چند جمله‌ای درونیابی درجه 5 مطابق می‌باشد. فرمولهائی که براساس این روش می‌باشند انتگرال گاوس نامیده می‌شوند. فرمولها فقط وقتی می‌توانند بکار برده شوند که تابع $f(x)$بطور صریح شناخته شده باشد، بطوریکه بتوان آن را برای هر

مقدار دلخواه x محاسبه نمود.

پارامترها را درحالت ساده‌ای از یک فرمول دو جمله‌ای که شامل چهار پارامتر ناشناخته می‌باشد، تعیین خواهیم نمود:

$$\int_{-1}^{1} f(t) \doteq af(t_1) + bf(t_2).$$

روش همانند روشی است که با تعیین پارامترهای مجهول در قسمت قبل شرح داده شده است. ما از یک فاصله متقارن انتگرال گیری برای سهولت در محاسبه استفاده می‌کنیم و متغیر خود را t نامگذاری می‌کنیم. فرمول برای هر چند جمله‌ای درجه سه معتبر می‌باشد، از این رو اگر $f(t) = t^3$، $f(t) = t^2$، $f(t) = t$ و $f(t) = 1$ باشد، خواهیم داشت:

$$\begin{aligned}
f(t) = t^3: &\quad \int_{-1}^{1} t^3\,dt = 0 = at_1^3 + bt_2^3;\\
f(t) = t^2: &\quad \int_{-1}^{1} t^2\,dt = \frac{2}{3} = at_1^2 + bt_2^2;\\
f(t) = t: &\quad \int_{-1}^{1} t\,dt = 0 = at_1 + bt_2;\\
f(t) = 1: &\quad \int_{-1}^{1} dt = 2 = a + b.
\end{aligned} \tag{20.7}$$

با ضرب معادله سوم در t_1^2 و کم کردن از معادله اول داریم:

$$0 = 0 + b[t_2^3 - t_2 t_1^2] = b(t_2)\,(t_2 - t_1)\,(t_2 + t_1). \tag{21.7}$$

می‌توانیم معادله (21.7) را بوسیله $b = 0$، $t_2 = t_1$ یا $t_1 = -t_2$ بدست آوریم. فقط فرض آخر رضایت‌بخش است و بقیه بی‌اعتبار هستند، پس $t_1 = -t_2$ را انتخاب می‌نمائیم. سپس در می‌یابیم که:

$$a = b = 1,$$

$$t_2 = -t_1 = \sqrt{\frac{1}{3}} = 0.5773,$$

$$\int_{-1}^{1} f(t)\,dt \doteq f(-0.5773) + f(0.5773).$$

قابل توجه است که مجموع این دو مقدار تابع به ما مقدار دقیقی برای انتگرال هر چند جمله‌ای درجه سه در فاصله -1 تا 1 را می‌دهد.

فرض کنید که حدود برای گرفتن انتگرال از a تا b باشد و از -1 تا 1 نباشد. (فرمول را برای این فاصله بدست آوردیم) برای استفاده از پارامترهای جدول بندی شده انتگرال گاوس، باید فاصله انتگرال گیری را از طریق تغییر متغیر به (1و 1-) تبدیل کنیم. ما متغیر داده شده را با متغیر دیگری که بطور خطی به آن وابسته می‌باشد، طبق ذیل عوض می‌کنیم:

اگر در نظر بگیریم $dx = \left(\frac{b-a}{2}\right)dt$, بطوریکه $x = \frac{(b-a)t + b + a}{2}$

پس نتیجه می‌شود:

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{(b-a)t + b + a}{2}\right)dt.$$

مثال ۷.۷ انتگرال $\int_0^{\frac{\pi}{2}} sin(x)\ dx$ $=I$ را محاسبه کنید. (واضح است که $I = 1.0$، بنابراین بسهولت می‌توانیم تخمین خطا را ببینیم).

برای استفاده از فرمول دو جمله‌ای گاوس باید متغیر انتگرال گیری را طوری عوض کنیم که حدود انتگرال گیری از -1 تا 1 باشد.

اگر $x = \frac{(\pi/2)t + \pi/2}{2}$ را بگیریم، بنابراین $dx = (\pi/4)dt$

مشاهده می‌کنیم وقتی که $t = -1$ است $x = 0$ و وقتی که $t = 1$ باشد. $x = \pi/2$ است. پس:

$$I = \frac{\pi}{4}\int_{-1}^{1} \sin\left[\frac{\pi t + \pi}{4}\right] dt$$

فرمول گاوس مقدار انتگرال جدید را بصورت یک جمع وزندار که از دو مقدار برحسب $t=-0.5773$ و $t=0.5773$ می‌باشد محاسبه می‌کند. از این رو:

$$I = \frac{\pi}{4}[(1.0)(\sin(0.10566\pi) + (1.0)(\sin(0.39434\pi)]$$

$$= 0.99847$$

خطا برابر 1.53×10^{-3} می‌باشد.

قدرت روش گاوس ناشی از این حقیقت است که ما به دو مقدار تابع نیازمندیم. اگر از روش ذوزنقه‌ای استفاده کرده بودیم که فقط به دو مقدار تابع احتیاج دارد، برآورد ما $(\pi/4)(0.0+1.0)=0.7854$ می‌شد که جوابی کاملا دور از نتیجه قبلی است. روش $\frac{1}{3}$ سیمپسون که به سه محاسبه تابع نیاز دارد و $I=1.0023$ را نتیجه می‌دهد با خطای 2.3×10^{-3} - که تا حدودی بزرگتر از خطای محاسبه شده در انتگرال گاوس می‌باشد.

انتگرال گاوس را می‌توان به بیش از دو جمله گسترش داد. فرمول برابر است با:

(22.7)

$$\boxed{\int_{-1}^{1} f(t)\,dt \doteq \sum_{i=1}^{n} w_i f(t_i),}$$

این فرمول در مورد توابع $f(t)$ که چند جمله‌ای‌های درجه $2n-1$ یا کمتر هستند صدق می‌کند. بعلاوه، با گسترش روشی که قبلاً برای فرمول دو نقطه‌ای بکار بردیم، برای هر n، یک دستگاه از $2n$ معادله بدست می‌آید:

$$w_i t_1^k + \quad \ldots \quad + w_n t_n^k = \begin{cases} 0, & k = 1, 2, 5, \ldots, 2n-1 \\ \dfrac{2}{k+1} & k = 0, 2, 4, \ldots, 2n-2 \end{cases}$$

روش واضح است. بهر حال، این سری از توابع که با نوشتن $f(t)$ از چند جمله‌ای‌های

متوالی بدست می‌آید به آسانی قابل حل نیست. می‌خواهیم روشی را نشان دهیم که از روش های موجود برای یک دستگاه خطی که در فصل 3 بکار بردیم آسانتر باشد.

نتیجه این می‌شود که t_i ها برای یک n داده شده، ریشه‌های چند جمله‌ای لژاندر درجه n می‌باشند. چند جمله‌ای‌های لژاندر از طریق بازگشتی تعیین می‌شوند.

$$(n+1)L_{n+1}(x) - (2n+1)xL_n(x) + nL_{n-1}(x) = 0,$$
$$\text{with } L_0(x) = 1,\ L_1(x) = x.$$

پس $L_2(x)$ عبارت است از:

$$L_2(x) = \frac{3xL_1(x) - (1)L_0(x)}{2} = \frac{3}{2}x^2 - \frac{1}{2};$$

در اینجا ریشه‌ها عبارتند از: $\pm\sqrt{\frac{1}{3}} = \pm\ 0.5773$ ، یعنی دقیقاً مقادیر t برای فرمول دو جمله‌ای می‌باشند.

با استفاده از رابطه بازگشتی بدست می‌آوریم:

$$L_3(x) = \frac{5x^3 - 3x}{2},$$

و همینطور ادامه می‌دهیم:

$$L_4(x) = \frac{35x^4 - 30x^2 + 3}{8},$$

روش‌های فصل ۳ به ما اجازه می‌دهند تا ریشه‌های این چند جمله‌ای‌ها را پیدا کنیم. پس از معلوم شدن آنها، مجموعه معادلات قابل قیاس با معادلات *(20.7)* به آسانی می‌توانند برای فاکتورهای وزندار حل شوند، چرا که معادلات نسبت به این مجهول‌ها خطی می‌باشند.

جدول ۲.۷ صفرهای چند جمله‌ای‌های لژاندر را تا درجه 5 فهرست می‌کند و برای انتگرال گاوس با چند جمله‌ای مبنا تا درجه 9 مقادیر لازم را می‌دهد. به عنوان مثال، $L_3(x)$ صفرهایی در x $= 0$ و $0.77459667+$ و $0.77459667-$ دارد. قبل از یک مثال در مورد استفاده از انتگرال گاوس، بهتر است خلاصه‌ای از خصوصیات چند جمله‌ای‌های لژاندر آورده شود.

۱- چند جمله‌ای‌های لژاندر در فاصله $[1$ و $-1]$ متعامدند. بدین معنی که

$$\int_{-1}^{1} L_n(x)\,L_m(x)\,dx \begin{cases} = 0 \text{ if } n \neq m; \\ > 0 \text{ if } n = m. \end{cases}$$

این خصوصیتی از چندین تابع مهم دیگر نظیر $\{cos(nx),\ n = 0, 1,\ldots\}$ می‌باشد. در اینجا داریم.

$$\int_{0}^{2\pi} \cos(mx)\cos(nx)\,dx \begin{cases} = 0 \text{ if } n \neq m; \\ > 0 \text{ if } n = m. \end{cases}$$

در این حالت می‌گوئیم که این معادلات در فاصله $[2\pi$ و $0]$ متعامدند.

۲- چند جمله‌های درجه n می‌توانند به عنوان مجموع چند جمله‌ای‌های لژاندر نوشته

شوند:

$$P_n(x) = \sum_{i=0}^{n} c_i L_i(x).$$

۳- n ریشه‌ی $L_n(x)=0$ در فاصله $[1$ و $-1]$ قرار دارند.

با استفاده از این خصوصیات می‌توانیم نشان دهیم که معادله (22.7) برای چند جمله‌ای های درجه $2n-1$ یا کمتر صدق می‌کند.

فاکتورهای وزن دار و مقادیر t برای انتگرال گاوس جدول بندی شده‌اند. (Love 1966) (مقادیری تا 200 جمله فرمولها را داده است). تعدادی از این مقادیر را در جدول ۲.۷ گنجانده‌ایم.

با یک مثال فرمول سه جمله‌ای را نشان می‌دهیم.

مثال ۷. ۸ با استفاده از فرمول سه قسمتی گاوس $I=\int_{0.2}^{1.5} e^{-x^2}dx$ را محاسبه کنید.

$$x = \frac{(1.5 - 0.2)t + 1.5 + 0.2}{2} = 0.65\,t + 0.85$$

پس:

$$I = \frac{1.5 - 0.2}{2} \int_{-1}^{1} e^{-(0.65t+0.85)^2}\,dt$$

$$= 0.65[0.555\ldots e^{-[0.65(-0.774\ldots)+0.85]^2} + 0.888\ldots e^{-[0.65(0.0)+0.85]^2} + 0.555\ldots e^{-[0.65(0.774\ldots)+0.85]^2}]$$

$$= 0.6586$$

(جواب دقیق 0.65882)

تعداد جملات	مقادیر t	فاکتورهای وزن‌دار	معتبر تا درجه زیر
2	−0.57735027	1.0	3
	0.57735027	1.0	
3	−0.77459667	0.55555555	5
	0.0	0.88888889	
	0.77459667	0.55555555	
4	−0.86113631	0.34785485	7
	−0.33998104	0.65214515	
	0.33998104	0.65214515	
	0.86113631	0.34785485	
5	−0.90617975	0.23692689	9
	−0.53846931	0.47862867	
	0.0	0.56888889	
	0.53846931	0.47862867	
	0.90617975	0.23692689	

جدول ۲.۷ مقادیری برای انتگرال گاوس

۷.۷ انتگرال‌های ناویژه و انتگرال‌های نامعین

برای محاسبه انتگرالهای $\int_a^b f(x)d(x)$ پذیرفتیم که $f(x)$ تابعی «خوش رفتار» است یعنی در فاصله $[a$ و $b]$ تابع $f(x)$ کراندار و پیوسته بوده و مشتقات آن نیز تا مرتبه مورد لزوم کراندار و پیوسته‌اند.

زمانی که تابع $f(x)$ در فاصله $[a$ و $b]$ خوش رفتار نباشد، روش‌های قبلی مناسب نخواهند بود.

بعضی اوقات می‌خواهیم انتگرالهایی از این قبیل را محاسبه کنیم.

$$I_1 = \int_0^\infty xe^{-x}\,dx; \qquad (23.7)$$

$$I_2 = \int_0^2 \frac{dx}{\sqrt{x}}; \qquad (24.7)$$

$$I_3 = \int_0^t x\sqrt{x^2+1}\,dx. \qquad (25.7)$$

معادله (23.7) دارای حد بالای نامحدود است و معادله (24.7) برای تابع انتگرال در حد پایین نامحدود است $(x = 0)$ و معادله (25.7) یک انتگرال نامعین است که مقدار آن تابعی از t می‌باشد.

گاهی می‌توانیم چنین انتگرالهایی را به حالت‌های ساده‌تری تبدیل کنیم و این کار می‌تواند با تغییر متغیر انجام شود. برای مثال، می‌توانیم I_1 را به مجموع دو انتگرال تبدیل کنیم.

$$I_1 = \int_0^1 xe^{-x}\,dx + \int_1^\infty xe^{-x}\,dx,$$

و سپس متغیر را در قسمت دوم با جایگزینی $y = 1/x$، تغییر دهیم بطوریکه:

$$\int_1^\infty xe^{-x}\,dx = \int_1^0 \frac{1}{y}e^{-1/y}\left(-\frac{dy}{y^2}\right) = \int_0^1 \frac{e^{-1/y}}{y^3}\,dy.$$

در انتگرال تبدیل شده، مقدار تابع انتگرال در $y=0$ نامعین است $(0/0)$، اما این مسئله مهمی نیست زیرا وقتی $y \to 0$ است حد تابع انتگرال برابر صفر می‌باشد و اگر برای تابع انتگرال $\lim_{y\to 0} f(y) = 0$ یا $f(0)=0$ فرض شود، دیگر انتگرال، ناویژه نیست.

در معادله (23.7) که انتگرال وجود دارد، می‌توانیم روش‌های عددی را مستقیماً بکار بریم. ما فقط $\int_0^A xe^{-x}\,dx$ را با افزایش مقدار A محاسبه نموده و از مقادیر محدود این نتایج به عنوان

مقدار انتگرال با حد بالای نامعین استفاده می‌نمائیم. بدین ترتیب که مقادیر بدست آمده را مقایسه کرده و در صورتیکه به‌سمت عدد معینی میل کند به عنوان جواب می‌پذیریم. وقتیکه همگرایی انتگرال ناشناخته باشد، می‌بایست در انجام این اعمال احتیاط نمود، برای اینکه خطای گردگردن کردن ممکن است یک حد بالاتری به دنباله نتایج بدهد، حتی اگر انتگرال واقعاً واگرا باشد.

مثال ۹.۷ جدول ۳.۷ نتایجی برای انتگرال معادله *(23.7)* می‌دهد. آنها نشان می‌دهند که چطور مقدار انتگرال با افزایش مقدار A به یک مقدار محدود می‌رسد. (از یک برنامه کامپیوتری برای انتگرال گیری رامبرگ استفاده شده است.)

به همین نحو می‌توان مقدار انتگرال در معادله *(24.7)* را از طریق یافتن مقدار I_2 با متغیری بجای حد پائین که به نقطه صفر نزدیک می‌شود پیدا نمود که نقطه‌ای است که در آنجا مقدار انتگرال نامعین است. مسئله این است که نمی‌دانیم چه مقداری برای h انتخاب نمائیم تا دقت مطلوب حاصل شود.

جدول ۳.۷ مقادیری برای انتگرال $A \to \infty \ , \ I = \int_0^A xe^{-x}dx$

A	I
1	0.26424
10	0.99950
100	1.00001
1000	1.00001
10000	1.00001
(∞)	(1.00000)

معادله *(25.7)* یک حالت خاص ساده می‌باشد در طول فصل بعد که معادلات دیفرانسیل بررسی می‌شوند، در مورد وضعیت این معادله بحث خواهیم کرد.

مثال ۱۰.۷ خطای انتگرال $\int_0^1 x^{-1/2}\,dx$ را با بکار بردن روش ذوزنقه‌ای بررسی می‌کنیم. خطا برابر است با:

$$E = (b - a)\frac{h^2}{12}f''(\xi)$$

$$E = \frac{h^2}{12} \times \frac{3}{4}\xi^{-5/2}$$

در این صورت ماکزیمم خطا محدود نخواهد بود و برای این مثال نمی‌توانیم دقت تقریب روش ذوزنقه‌ای را معین نمائیم. مشتقات تابع $f'(0)$ $f''(0)$ $f''(0)$ و... نامعین هستند و تابع خوش رفتار نیست.

مقدار انتگرال ناویژه $\int_0^1 x^{-1/2}\,dx$ را با بکار بردن روش ذوزنقه‌ای بررسی می‌کنیم.

در این روش داریم: $T_n = h(\frac{1}{2}f_0 + f_1 + f_2 + \ldots + \frac{1}{2}f_n)$

اما $f(x)$ در نقطه صفر نامعین است. $f(x) \to \infty$ $x \to 0$

در این صورت داریم:

$$I = \int_0^1 x^{-\frac{1}{2}} dx = \left| \frac{x^{\frac{1}{2}}}{\frac{1}{2}} \right|_0^1 = 2$$

کلیه فرمول‌های نیوتن- کوتز در مورد این مثال با شکست مواجه می‌شوند زیرا که این فرمولها همگی مقدار $f(0)$ را مورد استفاده قرار می‌دهند. برای مسائلی شبیه به این که تابع در نقاط x $= a$ یا $= b$ نامحدود است، به فرمولی احتیاج است که $f(b)$ و $f(a)$ در آن بکار نروند.

مثلاً به این منظور n را عددی زوج انتخاب می‌کنیم و از توابع پله‌ای که مقدار آن برابر $f(x_{2k+1})$ در فاصله $[x_{2k}, x_{2k+2}]$ می‌باشد، استفاده می‌کنیم:

$$I \approx 2hf_1 + 2hf_3 + \ldots + 2hf_{n-1} = 2h\,(f_1 + f_3 + \ldots + f_{n-1})$$

که روش «نقطه میانی» یا فرمول U نامیده می‌شود.

در این روش همگرائی بسیار آهسته است و برای انتگرال $I = \int_0^1 x^{-\frac{1}{2}} dx$ داریم:

h	I
.5	1.414
.25	1.578
.125	1.699
.0625	1.786
⋮	⋮

یا مثلاً باتوجه به‌همگرائی انتگرال فوق با انتخاب ε_k مناسب، انتگرال $I_k = \int_{\varepsilon_k}^1 x^{-\frac{1}{2}} dx$ را می‌توان محاسبه کرد. نظر به‌اینکه جمله باقیمانده $R_k = \int_0^{\varepsilon_k} x^{-\frac{1}{2}}\, dx$ به سمت صفر میل می‌کند، به این منظور اگر $\varepsilon_{k+1} = \frac{\varepsilon_k}{2}$ و $\varepsilon_0 = 0.1$ ، را انتخاب کنیم. زمانی که داریم

$$|I_{k+1}| \leq \varepsilon\, |I_k|$$

عملیات را خاتمه می‌دهیم و I_{k+1} نتیجه مطلوب خواهد بود.

در صورتیکه تغییر متغیر $x = \frac{1}{t}$ را در انتگرال فوق بکار بریم، نوع ناویژگی انتگرال تغییر می‌کند و نتیجه می‌شود که:

$$I = \int_1^{M_k} t^{-\frac{3}{2}} dx$$

$$E_k = \int_{M_k}^{\infty} t^{-\frac{3}{2}} dt$$

چون با افزایش M_k مقدار E_x کاهش یافته و بسمت صفر میل خواهد کرد، پس به عنوان مثال چنانچه $M_0=2$, $M_{k+1}=2M_k$ در این صورت زمانی که $|I_{k+1} - I_k| \leq \varepsilon$ و ε مقدار مثبت به حد کافی کوچک باشد، نتیجه مطلوب حاصل می‌گردد. ($\varepsilon=10^{-4}$ خطای قابل اغماض).

مثال ۱۱.۷ مطلوب است محاسبه انتگرال ناویژه

$$I=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} e^{-x^2/2}dx$$

با توجه به اینکه تابع $f(x) = e^{-x^2}$ زوج است، داریم:

$$I=\frac{2}{\sqrt{2\pi}}\int_{0}^{\infty} e^{-x^2/2}dx$$

$$I=\frac{2}{\sqrt{2\pi}}\int_{0}^{a} e^{-x^2/2}dx+\frac{2}{\sqrt{2\pi}}\int_{a}^{\infty} e^{-x^2/2}dx=I_1+I_2$$

با فرض همگرائی انتگرال فوق، a را چنان انتخاب می‌کنیم که I_2 از حد مطلوب $\varepsilon = 10^{-4}$ کوچکتر باشد، در اینصورت مقدار تقریبی انتگرال برابر I_1 خواهد بود.

$$I_2=\frac{2}{\sqrt{2\pi}}\int_{a}^{\infty} e^{-x^2/2}dx \quad , \quad I\approx I_1$$

از قضیه مقدار میانگین وزن دار برای انتگرال‌ها استفاده می‌کنیم و a را بدست می‌آوریم. فرض کنیم f و g بر $[a\ ,\ b]$ پیوسته باشند. هرگاه g در $[a\ ,\ b]$ اصلاً تغییر علامت ندهد آنگاه به ازای c ای در $[a,b]$ خواهیم داشت.

$$\int_a^b f(x)\,g(x)\,dx = f(c)\int_a^b g(x)\,dx$$

$$I_2=\frac{2}{\sqrt{2\pi}}\int_{a}^{\infty} e^{-x^2/2}dx =\frac{2}{\sqrt{2\pi}}\ \frac{1}{c}\int_{a}^{\infty} e^{-x^2/2}d\frac{x^2}{2}$$

$$I_2=\frac{2}{\sqrt{2\pi}}\ \frac{1}{c}\left[-e^{-x^2/2}\right]_a^{\infty}=\frac{2}{\sqrt{2\pi}}\ \frac{1}{c}\ e^{-\frac{a^2}{2}}$$

$$I_2=\frac{2}{\sqrt{2\pi}}\ \frac{1}{c}\ e^{-12.5} \leq \frac{2}{\sqrt{2\pi}}\ \frac{1}{5}\ e^{-12.5} \langle 10^{-5}$$

مقدار تقریبی انتگرال بازاء $a= 3.69$ برابر است با $I_1 \approx 0.9999$ و مقدار تحقیقی آن $I = 1$ می‌باشد.

۸.۷ کاربردهای توابع اسپلاین درجه سوم

علاوه بر استفاده روشن درونیابی، اسپلاین‌ها (فصل پنجم) را می‌توان برای پیدا کردن مشتق و انتگرال توابع مورد استفاده قرار داد، حتی وقتیکه تابع به عنوان جدولی از مقادیر شناخته شده است. همواری اسپلاین‌ها، بدلیل اینکه لازم است مشتق‌های اول و دوم در دو طرف همسایگی هر قسمت یکسان باشند، می‌تواند در بعضی موارد باعث دقت بیشتری شود.

برای اسپلاین درجه سه تقریب به $f(x)$، می‌توانیم در فاصله $x_i \leq x \leq x_{i+1}$ بنویسیم:

$$f(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i,$$

در جائی که ضرایب مانند بخش 5.5 مشخص شده‌اند از روش مشروح در آن بخش، S_i و S_{i+1} را محاسبه می‌کنیم. از این مقادیر S و مقادیر تابع $f(x)$، ضرایب درجه سه را محاسبه می‌کنیم.

$$a_i = \frac{S_{i+1} - S_i}{6(x_{i+1} - x_i)},$$

$$b_i = \frac{S_i}{2},$$

$$c_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} - \frac{2(x_{i+1} - x_i)S_i + (x_{i+1} - x_i)S_{i+1}}{6},$$

$$d_i = f(x_i).$$

تقریب مشتق‌های اول و دوم سرراست و آسان می‌باشند، اینها را به عنوان مقادیر مشتق‌های چند جمله درجه سه برآورد می‌نمائیم.

$$f'(x) \doteq 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i,$$

$$f''(x) \doteq 6a_i(x - x_i) + 2b_i.$$

در $n+1$ نقطه x_i که تابع در آنجا مشخص بوده و اسپلاین مطابق $f(x)$ است، این فرمولها مخصوصاً ساده هستند.

$$f'(x_i) \doteq c_i,$$

$$f''(x_i) \doteq 2b_i.$$

توجه کنید که اسپلاین درجه سه برای مشتقات تقریبی بالاتر از دو مفید نیست و تابع اسپلاین درجه بالاتر در این مورد لازم است. تخمین انتگرال $f(x)$ بر روی n فاصله بطوری که تابع $f(x)$ به وسیله تابع اسپلاین تخمین زده شده است به آسانی نتیجه می‌شود:

$$\int_{x_1}^{x_{n+1}} f(x)\,dx = \int_{x_1}^{x_{n+1}} P_3(x)\,dx$$

$$= \sum_{i=1}^{n} \left[\frac{a_i}{4}(x - x_i)^4 + \frac{b_i}{3}(x - x_i)^3 + \frac{c_i}{2}(x - x_i)^2 + d_i(x - x_i) \right]_{x_i}^{x_{i+1}}$$

$$= \sum_{i=1}^{n} \left[\frac{a_i}{4}(x_{i+1} - x_i)^4 + \frac{b_i}{3}(x_{i+1} - x_i)^3 + \frac{c_i}{2}(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i) \right].$$

اگر فواصل تماما هم اندازه باشند، ($h=x_{i+1}-x_i$)، این معادله برابر است با:

$$\int_{x_1}^{x_{n+1}} f(x)\,dx = \frac{h^4}{4}\sum_{i=1}^{n} a_i + \frac{h^3}{3}\sum_{i=1}^{n} b_i + \frac{h^2}{2}\sum_{i=1}^{n} c_i + h\sum_{i=1}^{n} d_i.$$

با یک مثال ساده طرز استفاده از تابع اسپلاین را برای محاسبه مشتق‌ها و انتگرالها شرح می‌دهیم.

مثال ۱۲.۷ انتگرال و مشتق‌های تابع $f(x)=\sin(\pi x)$ را روی فاصله $0 \le x \le 1$ محاسبه کنید، x ، 0.25, 0 $= 1.0, 0.75, 0.5$ که در تابع اسپلاین صدق می‌کند (جدول ۴.۷ مشاهده شود). از شرط پایانی S_5 $= 0$ و $S_1 = 0$ استفاده می‌کنیم. با حل ضرایب تابع اسپلاین درجه سه، نتایج در جدول ۴.۷ نشان داده شده است.

اکنون مشتقات را محاسبه می‌کنیم:

در $x = 0.25$:
$f'(x) = 2.2164$ $\quad f''(x) = -7.344$
$= 2.2214);$ $\quad = -6.979);$

در $x = 0.50$:
$f'(x) = 0$ $\quad f''(x) = -10.387$
$= 0);$ $\quad = -9.870);$

در $x = 0.75$:
$f'(x) = -2.2164$ $\quad f''(x) = -7.344$
$= -2.2214);$ $\quad = -6.979).$

خطاهای مشتق‌های اول که از اسپلاین درجه سه محاسبه شده کوچکتر از آنهایی است که از چند جمله‌ای درجه چهار درونیابی بدست می‌آیند، خطاهای مشتق‌های دوم از تابع اسپلاین تا حدودی بزرگتر از خطای چند جمله‌ای می‌باشند.

جدول ۴.۷

i	x	$f(x)$
1	0	0
2	0.25	0.7071
3	0.50	1.0000
4	0.75	0.7071
5	1.0	0

جدول ۵.۷

i	x	S_i	a_i	b_i	c_i	d_i
1	0	0	−4.8960	0	3.1340	0
2	0.25	−7.3440	−2.0288	−3.6720	2.2164	0.7071
3	0.50	−10.3872	2.0288	−5.1936	0	1.0
4	0.75	−7.3440	4.8960	−3.6720	−2.2164	0.7071

به سادگی انتگرال را محاسبه می‌کنیم:

$$\int_0^1 f(x)\,dx = \frac{(0.25)^4}{4}(0) + \frac{(0.25)^3}{3}(-12.5376) + \frac{(0.25)^2}{2}(3.1340) + 0.25(2.4142)$$
$$= 0.6362 \quad \text{دقیق} \quad = 0.6366; \text{خطا} \quad = +0.0004).$$

برای این مثال، دقت بطور قابل ملاحظه بیشتر از دقت حاصل از روش $\frac{1}{3}$ سیمپسون می‌باشد که مقدار آن 0.6381 بوده و خطای آن برابر -0.0015 می‌باشد.

۹.۷ انتگرال‌های چندگانه

ابتدا موردی را در نظر می‌گیریم که در آن حدود انتگرال گیری ثابت باشند، در ریاضیات یاد گرفتیم که یک انتگرال دو گانه ممکن است به عنوان یک انتگرال مکرر محاسبه شود، به عبارت دیگر، می‌توانیم بنویسیم:

$$\iint_A f(x, y)\,dA = \int_a^b \left(\int_c^d f(x, y)\,dy \right) dx = \int_c^d \left(\int_a^b f(x, y)\,dx \right) dy. \qquad (26.7)$$

در معادله (26.7) ناحیه مستطیل A با خطوط زیر محدود شده است.

$$x = a \ , \ x = b \ , \ y = c \ , \ y = d$$

در محاسبه انتگرال‌های مکرر، x را به هنگام انتگرال گیری در ارتباط با y ثابت نگه می‌داریم (برعکس حالت دوم). تطبیق فرمول‌های انتگرال گیری عددی که قبلاً در این فصل برای انتگرال گیری نسبت به یک متغیر مستقل بسط یافته‌اند، کاملا آسان می‌باشد. بیاد آورید که هر یک از فرمول‌های انتگرال گیری تنها یک ترکیب خطی از مقادیر تابع بوده که بازاء مقادیر مختلف متغیر مستقل محاسبه گشته‌اند، به عبارت دیگر، یک فرمول انتگرال تنها یک جمع وزندار مقادیر از تابعی معین می‌باشد.

انتگرال درونی به عنوان جمع وزندار مقادیر تابع می‌باشد که یکی از دو متغیر ثابت نگه‌داشته شده است، سپس می‌توانیم جمع وزندار این مجموع‌ها را بهم بیفزائیم. در صورتی که تابع

فقط درگره‌های (نقاط تقاطع) یک شبکه مستطیل در سراسر ناحیه مشخص شود، مقید می‌شویم از این مقادیر استفاده نمائیم. فرمول‌های نیوتن- کوتز یک مجموعه مناسب برای استفاده می‌باشند. هیچ دلیلی نیست که به چه علت همان فرمول باید در هر جهت مورد استفاده قرار گیرد، اگر چه اغلب مناسب انجام این کار می‌باشد.

مثال ۱۳.۷ این روش را با محاسبه انتگرال تابع جدول شده ۶.۷ بر روی یک ناحیه مستطیل شکل که به وسیله $x = 1.5$, $y = 3.0$, $y = 0.2$, $y = 0.6$ محدود گشته، شرح می‌دهیم.

از روش ذوزنقه‌ای در جهت x و از روش سیمپسون در جهت y استفاده می‌کنیم. (از آنجا که تعداد پانلها در جهت x زوج نمی‌باشد، از روش $\frac{1}{3}$ سیمپسون به آسانی استفاده نمی‌شود.) اینکه کدام انتگرال را اول محاسبه نمائیم اهمیتی ندارد. فرض کنیم با y ثابت شروع کنیم:

$$y = 0.2: \quad \int_{1.5}^{3.0} f(x, y)\,dx = \int_{1.5}^{3.0} f(x, 0.2)\,dx = \frac{h}{2}(f_1 + 2f_2 + 2f_3 + f_4)$$

$$= \frac{0.5}{2}[0.990 + 2(1.568) + 2(2.520) + 4.090]$$

$$= 3.3140;$$

$$y = 0.3: \quad \int_{1.5}^{3.0} f(x, 0.3)\,dx = \frac{0.5}{2}[1.524 + 2(2.384) + 2(3.800) + 6.136]$$

$$= 5.0070.$$

بطور مشابه، در:

$$y = 0.4, \quad I = 6.6522;$$

$$y = 0.5, \quad I = 8.2368;$$

$$y = 0.6, \quad I = 9.7435.$$

اکنون این مقادیر را در جهت y بر حسب روش سیمپسون جمع می‌کنیم:

$$f(x, y)\,dx = \frac{0.1}{3}[3.3140 + 4(5.0070) + 2(6.6522) + 4(8.2368) + 9.7435]$$

$$= 2.6446. \quad \blacksquare$$

جدول ۶.۷ جدول بندی یک تابع دو متغیره، $u = f(x,y)$

x \ y	0.1	0.2	0.3	0.4	0.5	0.6
0.5	0.165	0.428	0.687	0.942	1.190	1.431
1.0	0.271	0.640	1.003	1.359	1.703	2.035
1.5	0.447	0.990	1.524	2.045	2.549	3.031
2.0	0.738	1.568	2.384	3.177	3.943	4.672
2.5	1.216	2.520	3.800	5.044	6.241	7.379
3.0	2.005	4.090	6.136	8.122	10.030	11.841
3.5	3.306	6.679	9.986	13.196	16.277	19.198

مثال قبلی نشان می‌دهد که انتگرال دوگانه از طریق عددی به معنی تبدیل به جمع بندی دوگانه مقادیر تابع وزندار می‌باشد. محاسباتی که انجام داده‌ایم به این صورت نوشته می‌شود:

$$\begin{aligned}\int\int f(x, y)\, dx\, dy &= \sum_{j=1}^{m} v_j \sum_{i=1}^{n} w_i f_{ij} \\ &= \frac{\Delta y}{3}\frac{\Delta x}{2}[(f_{1,1} + 2f_{2,1} + 2f_{3,1} + f_{4,1}) \\ &\quad + 4(f_{1,2} + 2f_{2,2} + 2f_{3,2} + f_{4,2}) \\ &\quad + \cdots + (f_{1,5} + 2f_{2,5} + 2f_{3,5} + f_{4,5})].\end{aligned}$$

مناسب است که این رابطه رابه صورت اپراتور تصویری بنویسیم که در آن فاکتورهای وزندار به صورت آرایه‌ای نشان داده می‌شوند.

$$\int\int f(x, y)\, dx\, dy = \frac{\Delta y}{3}\frac{\Delta x}{2}\begin{Bmatrix}1 & 4 & 2 & 4 & 1\\ 2 & 8 & 4 & 8 & 2\\ 2 & 8 & 4 & 8 & 2\\ 1 & 4 & 2 & 4 & 1\end{Bmatrix} f_{i,j}. \qquad (27.7)$$

ما در فرمول بالا اعداد را به این ترتیب تعبیر می‌کنیم: از مقدارهای *1,4,2,4,1* به عنوان فاکتورهای وزندار برای مقادیر تابع در سطر بالای قسمت جدول ۶.۷ که بر روی آن انتگرال می‌گیریم، (مقادیری که در آن *1.5*=x و y از *0.2* تا *0.6* تغییر می‌کند.) استفاده می‌کنیم. همینطور ستون دوم آرایه در معادله *(27.7)* نشان دهنده فاکتورهای وزندار است که برای یک ستون از مقادیر تابع انتگرال می‌گیریم، (مقادیری از تابع که $y = 0.4$ و x از *1.5* تا *3* تغییر می‌کند.) مشاهده می‌شود که مقادیر در اپراتور تصویری معادله *(27.7)* از ضرایب نیوتن- کوتز برای انتگرال گیری یک متغیری پیروی می‌کنند.

دیگر ترکیبات فرمول نیوتن- کوتز نتایج مشابهی می‌دهند. احتمالا آسانترین روش برای محاسبه دستی استفاده از اپراتور انتگرال گیری تصویری می‌باشد.

انتگرال گیری تصویری به آسانی با هر نوع ترکیب مطلوب از فرمول‌های انتگرال گیری تطبیق پیدا می‌کند. به جز مشکل نمایش بیش از دو بعد، این فن اپراتور همچنین برای انتگرال های سه گانه و چهار گانه بکار می‌رود. نمایش دیگر برای چنین اپراتور تصویری وجود دارد که جهت انتقال به یک برنامه کامپیوتری آسانتر می‌باشد. فرمول انتگرال گیری عددی را برای یک متغیر ملاحظه نمائید:

$$\int_{-1}^{1} f(x)\, dx \doteq \sum_{i=1}^{n} a_i f(x_i). \qquad (28.7)$$

ما در بخش *8.7* دیده‌ایم که چنین فرمول‌هائی می‌توانند دقیق بوجود آیند، در صورتی که $f(x)$ یک چند جمله‌ای با یک درجه معین باشد. فرض کنید که معادله *(28.7)* برای چند جمله‌ای‌های تا درجه s باشد.

حالا فرمول انتگرال چند گانه را ملاحظه می‌کنیم:

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(x, y, z)\,dx\,dy\,dz \stackrel{?}{=} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_i a_j a_k f(x_i, y_j, z_k). \qquad (29.7)$$

می‌خواهیم نشان دهیم که معادله *(29.7)* دقیقاً برای تمام چند جمله‌ایها برحسب x, y, z تا درجه s صادق است. چنین چند جمله‌ای، یک ترکیب خطی از جملاتی به شکل $x^{\alpha}y^{\beta}z^{\gamma}$ می‌باشد که‌در آنجا α, β, γ اعداد صحیح غیر منفی هستند که جمع آنها برابر یا کوچکتر از s می‌باشد. اگر بتوانیم ثابت کنیم که معادله *(29.7)* جمله عمومی این حالت است، در این صورت برای چند جمله‌ای مورد نظر صادق است.

برای انجام آن فرض می‌کنیم که:

$$f(x,y,z) = x^{\alpha}\, y^{\beta}\, z^{\gamma}$$

پس چون حدود ثابت بوده و انتگرال قابل فاکتور گیری است،

$$\begin{aligned} I &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} x^{\alpha}y^{\beta}z^{\gamma}\,dx\,dy\,dz \\ &= \left(\int_{-1}^{1} x^{\alpha}\,dx\right)\left(\int_{-1}^{1} y^{\beta}\,dy\right)\left(\int_{-1}^{1} z^{\gamma}\,dz\right). \end{aligned}$$

با جایگزین نمودن هر قسمت بر حسب معادله *(28.7)* بدست می‌آوریم:

$$I = \left(\sum_{i=1}^{n} a_i x_i^{\alpha}\right)\left(\sum_{j=1}^{n} a_j y_j^{\beta}\right)\left(\sum_{k=1}^{n} a_k z_k^{\gamma}\right) = \sum_{i=1}^{n} a_i x_i^{\alpha}\sum_{j=1}^{n} a_j y_j^{\beta}\sum_{k=1}^{n} a_k z_k^{\gamma}. \qquad (30.7)$$

حالا به یک روش مقدماتی برای بدست آوردن ضرب جمع‌ها نیاز داریم. آن را برای یک حالت ساده نشان می‌دهیم:

$$\begin{aligned} \left(\sum_{i=1}^{3} u_i\right)\left(\sum_{j=1}^{2} v_j\right) &= \sum_{i=1}^{3}\left(\sum_{j=1}^{2} u_i v_j\right) \\ &= \sum_{i=1}^{3}\sum_{j=1}^{2} u_i v_j. \end{aligned}$$

آخرین معادله کاملا بصورت نمادی است. اولی را با بسط دو طرف ثابت می‌کنیم.

$$\begin{aligned} \left(\sum_{i=1}^{3} u_i\right)\left(\sum_{j=1}^{2} v_j\right) &= \sum_{i=1}^{3} u_i \sum_{j=1}^{2} v_j \\ &= (u_1 + u_2 + u_3)(v_1 + v_2) \\ &= u_1v_1 + u_1v_2 + u_2v_1 + u_2v_2 + u_3v_1 + u_3v_2; \\ \sum_{i=1}^{3}\sum_{j=1}^{2} u_i v_j &= (u_1v_1 + u_1v_2) + (u_2v_1 + u_2v_2) + (u_3v_1 + u_3v_2). \end{aligned}$$

با برداشتن پرانتزها می‌بینیم که دو طرف یکسان هستند. با استفاده از این اصل می‌توانیم معادله *(30.7)* را به شکل زیر بنویسیم:

$$I = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} a_i a_j a_k x_i^{\alpha} y_j^{\beta} z_k^{\gamma}, \qquad (31.7)$$

که نشان می‌دهد تساوی مورد بحث معادله *(29.7)* معتبر است و می‌توانیم برنامه‌ای برای انتگرال سه گانه به وسیله سه حلقه *for* تو در تو بنویسیم. ضرایب a_i از فرمول انتگرال گیری عددی انتخاب می‌شود. در بعضی موارد تغییر متغیر برای بررسی معادله *(28.7)* ضروری است.

هرگاه یک انتگرال چند گانه را بطور عددی محاسبه می‌کنیم.بطوری که تابع انتگرال شناخته شده باشد، انتخاب شکل معادله *(29.7)* مناسب است. فرمول‌های انتگرال گاوس نسبت به فرمول‌های نیوتن- کوتز از کارآئی بالاتری برخوردار است.

مثال ۱۴.۷ محاسبه کنید: $I = \int_0^1 \int_{-1}^0 \int_{-1}^1 yze^x \, dx \, dy \, dz$

انتگرال گاوس را با استفاده از یک فرمول سه جمله‌ای برای x و فرمول‌های دو جمله‌ای برای y و z بکار می‌بریم. ابتدا تغییر متغیر می‌دهیم تا حدود را برای y و z با *(1, −1)* تطبیق دهیم:

$$y = \frac{1}{2}(u + 1), \qquad dy = \frac{1}{2} du;$$

$$z = \frac{1}{2}(v - 1), \qquad dz = \frac{1}{2} dv.$$

انتگرال می‌شود:

$$I = \frac{1}{16} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (u + 1)(v - 1)e^x \, dx \, du \, dv.$$

فرمول‌های دو و سه نقطه‌ای گاوس از بخش *12.7* عبارتند از:

$$\int_{-1}^1 f(x) \, dx = (1)f(-0.5774) + (1)f(0.5774),$$

$$\int_{-1}^1 f(x) \, dx = \left(\frac{5}{9}\right)f(-0.7746) + \left(\frac{8}{9}\right)f(0) + \left(\frac{5}{9}\right)f(0.7746).$$

پس انتگرال بدین صورت می‌شود:

$$I = \frac{1}{16} \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{3} a_i a_j b_k (u_i + 1)(v_j - 1)e^{x_k},$$

$$a_1 = 1, \qquad a_2 = 1,$$

$$b_1 = \frac{5}{9}, \qquad b_2 = \frac{8}{9}, \qquad b_3 = \frac{5}{9},$$

و مقادیر x, v, u مطابق بالاست.

تعدادی از جملات محاسبه جمع، عبارتند از:

$$I = \frac{1}{16}\Big[(1)(1)\Big(\frac{5}{9}\Big)(-0.5774 + 1)(-0.5774 - 1)e^{-0.7446}$$
$$+ (1)(1)\Big(\frac{8}{9}\Big)(-0.5774 + 1)(-0.5774 - 1)e^{0}$$
$$+ (1)(1)\Big(\frac{5}{9}\Big)(-0.5774 + 1)(-0.5774 - 1)e^{0.7746}$$
$$+ (1)(1)\Big(\frac{5}{9}\Big)(0.5774 + 1)(-0.5774 - 1)e^{-0.7746}\Big]$$

با محاسبه بدست می‌آید: $I = -0.58758$ و مقدار خلیلی مساویست با:

$$-\frac{1}{4}(e - e^{-1}) = -0.58760.$$

۱۰.۷ خطا در انتگرال‌های چندگانه

عبارت خطا در مورد انتگرال گیری چند گانه قابل قیاس با حالت یک بعدی است. با استفاده از روش ذوزنقه‌ای در هر دو جهت، با فواصل یکسان، و انتخاب n فاصله در جهت x و m فاصله در جهت y نشان می‌دهیم که این موضوع در مورد انتگرال گیری دوگانه نیز صادق می‌باشد.

از روش ذوزنقه‌ای داریم:

$$\text{خطای} \quad \int_a^b f(x)\,dx = -\frac{b-a}{12}h^2 f''(\xi) = O(h^2),$$

$$h = \Delta x = \frac{b-a}{n}.$$

در بسط انتگرال گیری رامبرگ دریافتیم که عبارت خطا را می‌توان بدین صورت نوشت:

$$\text{خطا} \quad = O(h^2) = Ah^2 + O(h^4) \doteq Ah^2 + Bh^4,$$

که در اینجا A ثابت بوده و مقدار B بستگی به مشتق چهارم تابع دارد. با افزودن این عبارت خطا به روش ذوزنقه‌ای بدست می‌آوریم (مشابه با معادله *(28.7)*:

$$\int_a^b f(x)\,dx\Big|_{y=y_j} = \frac{h}{2}(f_{0,j} + 2f_{1,j} + 2f_{2,j} + \cdots + f_{n,j}] + A_j h^2 + B_j h^4.$$

با جمع نمودن در جهت y و منحصرا با حفظ عبارات خطا، بدست می‌آوریم.

$$\int_c^d \int_a^b f(x,y)\,dx\,dy = \frac{k}{2}\frac{h}{2}\sum_{i=0}^{n}\sum_{j=0}^{m} a_i a_j f_{i,j} + \frac{k}{2}(A_0 + 2A_1 + 2A_2 + \cdots + A_m)h^2$$
$$+ \frac{k}{2}(B_0 + 2B_1 + 2B_2 + \cdots + B_m)h^4 + \bar{A}k^2 + \bar{B}k^4,$$

$$k = \Delta y = \frac{d-c}{m}. \qquad (32.7)$$

در معادله *(32.7)*، $\bar{A}$ و $\bar{B}$ ضرایب خطا در جهت y هستند. ضرایب A و B برای عبارات خطا در جهت x ممکن است برای هر کدام از $(m+1)$ مقدار y متفاوت باشند، اما هر یک از جمع‌ها

در پرانتزهای معادله *(32.7)*، $2n$ برابر مقدار میانگین A یا B هستند، بنابراین جملات خطا از این قرارند:

$$\text{خطا} = \frac{k}{2}(nA_{av})h^2 + \frac{k}{2}(nB_{av})h^4 + \bar{A}k^2 + \bar{B}k^4. \qquad (33.7)$$

چون Δy و Δx هر دو ثابت هستند، ممکن است انتخاب کنیم $\Delta y = k = \alpha\Delta x = \alpha h$ بطوری که $\alpha = \dfrac{\Delta y}{\Delta x}$ و با $nh = b - a$ معادله *(33.7)* را مجدداً می‌نویسیم:

$$\text{خطا} = \left(\frac{b-a}{2}A_{av}\alpha\right)h^2 + \left(\frac{b-a}{2}B_{av}\alpha\right)h^4 + A\alpha^2h^2 + B\alpha^4h^4$$
$$= K_1h^2 + K_2h^4.$$

در اینجا K_2 به مشتقات جزئی مرتبه چهارم وابسته خواهد بود. این موضوع، پیش‌بینی ما را که عبارت خطا در انتگرال گیری چندگانه از طریق عددی به همان شکل انتگرال یگانه است، تأیید می‌کند. در اینصورت انتگرال گیری رامبرگ ممکن است برای انتگرال گیری چندگانه بکار برده شود، که به موجب آن تخمین $O(h^4)$ را در دو محاسبه مختلف ذوزنقه‌ای در یک فاصله برونیابی می‌کنیم. از دو محاسبه مختلف $O(h^4)$ ممکن است به خطای $O(h^6)$ برونیابی کنیم.

۱۱.۷ انتگرال‌گیری چند گانه با حدود متغیر

اگر حدود انتگرال گیری ثابت نباشد بطوری که در صفحه y و x ناحیه‌ای که بر روی آن انتگرال $f(x,y)$ حساب می‌شود به صورت مستطیل نباشد، بایستی ناحیه فوق را تغییر دهیم. مثال ساده‌ای را برای تشریح این روش در نظر می‌گیریم.

مثال ۱۵.۷ مطلوبست محاسبه انتگرال

$$\int\int f(x,y)\,dy\,dx$$

در ناحیه محدود به وسیله خطوط $x = 0$ و $x = 1$ و $y = 0$ و منحنی $y = x^2+1$ در نظر گرفته می‌شود. ناحیه در شکل ۳.۷ رسم شده است. اگر خطوط عمودی به فاصله $\Delta x = 0.2$ در نظر گرفته شود، این ناحیه در شکل ۳.۷ بصورت خطوط تیره نشان داده شده است. واضح است که می‌توانیم انتگرال درونی را بازاء مقادیر ثابت x در امتداد هر یک از خطوط عمودی (شامل $x=0$ و $x=1$) تخمین بزنیم. اگر از روش ذوزنقه‌ای با پنج پانل برای هر یک از اینها استفاده کنیم، سریهائی از جمع‌ها را بدست می‌آوریم:

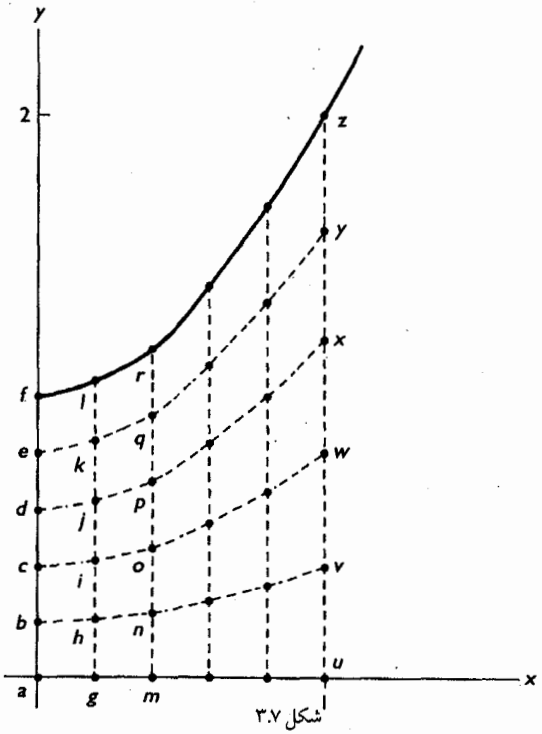
$$S_1 = \frac{h_1}{2}(f_a + 2f_b + 2f_c + 2f_d + 2f_e + f_f),$$

$$S_2 = \frac{h_2}{2}(f_g + 2f_h + 2f_i + 2f_j + 2f_k + f_l),$$

$$S_3 = \frac{h_3}{2}(f_m + 2f_n + \cdots),$$

$$\vdots$$

$$S_6 = \frac{h_6}{2}(f_u + 2f_v + 2f_w + 2f_x + 2f_y + f_z).$$



شکل ۳.۷

اندیس‌ها در اینجا نشان دهنده مقادیر تابع در نقاطی است که در شکل ۳.۷ نمایش داده شده‌اند. مقادیر h_i همانند مقادیر معادلات بالا نیست، اما در هر یک از فواصل عمودی تقسیم بر 5 شده است. ترکیب این جمعها برای تخمین انتگرال دوگانه عبارت خواهد بود از:

$$\text{انتگرال} = \frac{0.2}{2}(S_1 + 2S_2 + 2S_3 + 2S_4 + 2S_5 + S_6).$$

برای توضیح بیشتر فرض می‌کنیم که $f(x,y) = xy$، پس:

$$S_1 = \frac{1.0/5}{2}(0 + 0 + 0 + 0 + 0 + 0) = 0,$$

$$S_2 = \frac{1.04/5}{2}(0 + 0.0832 + 0.1664 + 0.2496 + 0.3328 + 0.208) = 0.1082,$$

$$S_3 = \frac{1.16/5}{2}(0 + 0.1856 + 0.3712 + 0.5568 + 0.7428 + 0.464) = 0.2692,$$

$$S_4 = \frac{1.36/5}{2}(0 + 0.3264 + 0.6528 + 0.9792 + 1.3056 + 0.816) = 0.5549,$$

$$S_5 = \frac{1.64/5}{2}(0 + 0.5248 + 1.0496 + 1.5744 + 2.0992 + 1.312) = 1.0758,$$

$$S_6 = \frac{2.0/5}{2}(0 + 0.8 + 1.6 + 2.4 + 3.2 + 2.0) = 2.0;$$

$$\text{انتگرال} = \frac{0.2}{2}(0 + 0.2164 + 0.5384 + 1.1098 + 2.1516 + 2.0)$$
$$= 0.6016$$

توسعهٔ روش به نواحی پیچیده‌تر و تطبیق آن برای استفاده از روش سیمپسون باید روشن شده باشد. اگر توابعی که ناحیه را تعریف می‌کنند تک مقدار نباشند، باید ناحیه را به چند ناحیه جزئی تقسیم کرد تا مشکل رفع شود. (همانطور که در روش‌های تحلیلی انتگرال گیری انجام می‌دهیم).

محاسبات قبلی خیلی دقیق نبودند زیرا که در روش ذوزنقه‌ای خطا نسبتاً بزرگ است. روش گاوس مناسبتر است، حتی اگر تعداد نقاط کمتری در ناحیه بکار بریم.

اگر روش سه نقطه در جهت x و روش چهار نقطه در جهت y را بکار بریم، مطابق بخش 6.7 باید حدود انتگرال را تغییر دهیم:

$$\int_0^1 \int_0^{x^2+1} x\,y\,dy\,dx$$

به

$$\frac{1}{4}\int_{-1}^{1}\int_{-1}^{1} \frac{s+1}{2}\left[\frac{(x^2(s)+1)^2 t + (x^2(s)+1)^2}{2}\right] dt\,ds$$

جایگذاری زیر را انجام می‌دهیم:

$$x = \frac{s+1}{2} \qquad y = \frac{(x^2(s)+1)^2 t + (x^2(s)+1)^2}{2}.$$

انتگرال تقریبی برابر مجموع زیر است

$$\sum_{i=1}^{3}\sum_{j=1}^{4} w_i W_j f(s_i, t_j),$$

مقادیر w_i و W_j و s_i و t_j از جدول ۲.۷ می‌باشند از آن جدول $w_1 = 0.5555555$ و $w_3 = w_1$ و $w_2 = 0.8888889$ و $s_1 = -0.77459667$ و $s_3 = -s_1$ و $s_2 = 0.0$. مقادیر W_j و t_j بهمین طریق بدست می‌آیند:

برای هر مقدار ثابت i، 3 و 2 و $i = 1$ فرض می‌کنیم S_i متناظر مقدار انتگرال گاوس برای یک ثابت s_i باشد، بطوری که

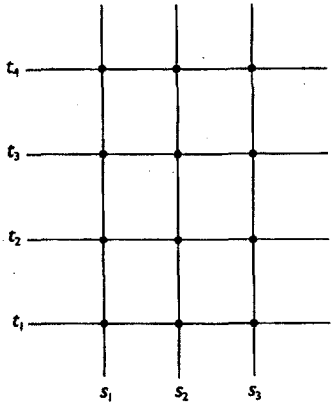
$$S_i = \sum_{j=1}^{4} w_j f(s_i, t_j)$$

مقادیر زیر بدست می‌آیند:

$$S_1 = (0.00279158 + 0.02487506 + 0.05050174 + 0.03741447) = 0.11558285,$$
$$S_2 = (0.01886891 + 0.16813600 + 0.34135240 + 0.25289269) = 0.78125000,$$
$$S_3 = (0.06845742 + 0.61000649 + 1.23844492 + 0.91750833) = 2.83441716.$$

این مقادیر مطابق زیر جمع می‌شوند

$$\frac{w_1 S_1 + w_2 S_2 + w_3 S_3}{4} = 0.58333334$$



که برابر جواب دقیق تا هفت رقم می‌باشد در این حالت فقط *12* مقدار از تابع حساب شده است. در ناحیه زیر S_i در امتداد عمودی محاسبه می‌شود.

۱۲.۷ برنامه‌های کامپیوتری

۱.۷ لیست برنامه انتگرال دوگانه به‌روش رامبرگ به زبان *FORTRAN*

این برنامه بنام *DBLINT.F* می‌باشد.

```
      SUBROUTINE  DBLINT(FCT,XA,XB,YA,YB,TOL,RESULT)
C
C -----------------------------------------------------------------
C APPLIED  NUMERICAL  ANALYSIS
C -----------------------------------------------------------------
C
C -----------------------------------------------------------------
C
C   SUBROUTINE  DBLINT :
C              THIS ROUTINE COMPUTES THE INTEGRAL OF A
C      FUNCTION OF TWO VARIABLES.  THE ROMBERG METHOD IS USED. INITIALLY,
C   FOUR SUBDIVISIONS ARE USED. THESE ARE HALVED UNTIL THE TOLERANCE
C   IS MET, WITH A MAXIMUM NUMBER OF SUBDIVIDINGS OF FIVE.
C -----------------------------------------------------------------
C   PARAMETERS  ARE :
C
C   FCT  - FUNCTION SUBPROGRAM TO COMPUTE F(X,Y). DECLARED EXTERNAL
C          IN CALLING PROGRAM.
C   XA,XB  - LOWER AND UPPER LIMITS FOR X.
C   YA,YB  - LOWER AND UPPER LIMITS FOR Y.
C    TOL   - TOLERANCE TO TERMINATE INTEGRATION. WHEN NOT MET, A
C              MESSAGE IS PRINTED AND LAST VALUE RETURNED.
C   RESULT - RETURNS VALUE OF INTEGRAL TO CALLER.
C    ARRAY  - DOUBLY SUBSCRIPTED ARRAY TO HOLD INTERMEDIATE VALUES
C          FOR COMPARISON AND EXTRAPOLATION.
C   A FUNCTION SUBPROGRAM NAMED SUMROW, IS CALLED TO COMPUTE SUMS
C ACROSS ONE ROW OF THE REGION.
C
C -----------------------------------------------------------------
      REAL  FCT,XA,XB,YA,YB,TOL,RESULT
      INTEGER  I,N,J,K
      REAL  ARRAY(6,6),DELX,DELY,SUMROW,Y
      EXTERNAL  FCT
C
C -----------------------------------------------------------------
C
```

```
      C  INITIALIZE  DEL VALUES  AND  SUM TOP AND BOTTOM
ROWS
           DELX  = (XB  - XA)  / 4.0
           DELY  = (YB  - YA)  / 4.0
           N = 4
           ARRAY(1,1)    = SUMROW(FCT,XA,XB,YA,DELX,N)        +
          +              SUMROW(FCT,XA,XB,YB,DELX,N)
      C
      C  ---------------------------------------------------------------
      C
      C  GET  THE  SUMS  FOR  INTERMEDIATE    ROWS
      C
           Y = YA
           DO  10 I = 2,N
             Y = Y + DELY
                                ARRAY(1,1)      =     ARRAY(1,1)      +
2.0*SUMROW(FCT,XA,XB,Y,DELX,N)
         10 CONTINUE
           ARRAY(1,1)    = ARRAY(1,1)    * DELX  * DELY  / 4.0
      C
      C  ---------------------------------------------------------------
      C
      C    NOW   HALVE   THE   VALUES   OF   DELX   AND   DELY,
RECOMPUTE   THE INTEGRAL,   AND
      C  EXTRAPOLATE,    THEN  TEST  TO  SEE  IF  TOLERANCE    IS
MET.  REPEAT  UP TO
      C  FIVE  TIMES.
           DO  40 J = 1,5
             DELX  = DELX  / 2.0
             DELY  = DELY  / 2.0
             N = 2*N
      C  ---------------------------------------------------------------
      C  DO  TOP  AND  BOTTOM   ROWS  FIRST
      C
           ARRAY(J+1,1)    = SUMROW(FCT,XA,XB,YA,DELX,N)        +
          +               SUMROW(FCT,XA,XB,YB,DELX,N)
      C
      C  THEN  THE  INTERMEDIATE    ROWS
      C
           Y = YA
           DO  20 I = 2,N
             Y = Y + DELY
                               ARRAY(J+1,1)      =     ARRAY(J+1,1)      +
2.0*SUMROW(FCT,XA,XB,Y,DELX,N)
         20   CONTINUE
           ARRAY(J+1,1)    = ARRAY(J+1,1)    * DELX  * DELY  / 4.0
      C  ---------------------------------------------------------------
      C  NOW  WE  EXTRAPOLATE
           DO  30 K = 1,J
           ARRAY(J+1,K+1)    = ARRAY(J+1,K)    + 1.0 / (4.0**K - 1.0) *
          +                  ( ARRAY(J+1,K)    - ARRAY(J,K)    )
         30    CONTINUE
```

```
          IF ( ABS(ARRAY(J+1,J+1)     - ARRAY(J+1,J))    - TOL ) 50,50,40
        40   CONTINUE
      C
      C  ---------------------------------------------------------------------
      C
      C  WE HAVE  A NORMAL  TERMINATION   OF LOOP 40 ONLY
WHEN  THE TOLERANCE   IS
      C  NOT MET,  SO PRINT  MESSAGE  AND  RETURN.
      C
           PRINT  201, TOL
           RESULT  = ARRAY(6,6)
           RETURN
        50 RESULT  = ARRAY(J+1,J+1)
           RETURN
      C
        201 FORMAT(/'  TOLERANCE   OF ',E14.7,' NOT  MET  AFTER
FIVE ',
          +        'EXTRAPOLATIONS.'/)
           END
      C  ---------------------------------------------------------------------
           REAL  FUNCTION  SUMROW(FCT,XA,XB,Y,DELX,N)
      C  ---------------------------------------------------------------------
      C    FUNCTION  SUMROW  :
      C                THIS FUNCTION  COMPUTES  THE WEIGHTED
SUM  FOR
      C    TRAPEZOIDAL  RULE INTEGRATION  ACROSS  ONE ROW
OF A REGION,  FROM
      C      XA  TO  XB  WITH  INTERVALS   OF DELX,  WHERE  THE
VALUE  OF Y IS Y.
      C
      C  ---------------------------------------------------------------------
      C    PARAMETERS   ARE  :
      C    FCT    - EXTERNAL   FUNCTION  THAT  COMPUTES  F(X,Y)
      C    XA,XB   - LIMITS  FOR  X VALUES
      C    Y      - VALUE   OF  Y
      C    DELX    - STEP  SIZE  FOR  X
      C    N      - NUMBER   OF INTERVALS
      C
      C  ---------------------------------------------------------------------
      C  GET  FIRST  AND  LAST  VALUES   TO  START.
           SUMROW   = FCT(XA,Y)   + FCT(XB,Y)
      C  NOW  ADD  IN  THE  INTERMEDIATE    VALUES.
           X = XA
           DO  10 I = 2,N
             X = X + DELX
             SUMROW   = SUMROW   + 2.0 * FCT(X,Y)
        10   CONTINUE
           RETURN
           END
```

۲.۷ لیست برنامه انتگرال گیری به روش سیمپسون به زبان *PASCAL*

این برنامه بنام *SIMPSON1.PAS* می‌باشد.

```
Program simpson1(INPUT,OUTPUT);
(*

 APPLIED  NUMERICAL   ANALYSIS

-----------------------------------------------------------------
    This program is a modIfication  of the one given in the book,
since the interval is subdivided  until the new value from
Simpson's  Rule  dIffers  within a small tolerance  of the previous
value.

    The  initial h is (b-a)/2.0. If S(n)  is the previous  value
for the integral for h(n),  the  S(n+1)  is S(n)/2  +  4*(the  new
values  of F(x)  at the new x's with h(n+1)  =  h(n)/2.

    The  new values  are actually the leaves of a BINARY   tree  with
the ROOT  node being f((b-a)/2). It turns  out that we have a
complete  binary tree from level 2 down to the bottom.

    In this example  we compute  the integral of f(x)  =  1/(x*x)  on
the interval [0.2, 1.0]. The exact answer is 4.
*)
Const
  a = 0.2;  b = 1.0;  TOL  = 0.00001;
Type
  vector  = Array[1..1023] OF Real;
Var
  h,                    (* step_size  *)
  center_val,           (* f((b-a)/2)  *)
  first_val,            (* f(a)        *)
  last_val,             (* f(b)        *)
  sum_of_leaves,        (* these  are the new values added to
                        array *)
  old_sum_of_leaves,
  old_sum,              (* S(n)        *)
  new_sum,              (* S(n+1)      *)
  integral_value        (* improved  value from new_sum  and
                        old_sum  *)
                : Real;
  leftTREE_point,
  rightTREE_point,
  leftTREE_value,
  rightTREE_value    : vector;

  i,n           : Integer;
  done          : BOOLEAN;
```

```
Function F(x : Real): Real;
Begin
  F := 1.0/(x*x)

End;

Procedure setup_initial_value;
Begin
  h := (b-a)/2.0;
  first_val := F(a); last_val := F(b);  center_val := F(a+h);
  old_sum  := h*(first_val + last_val + 4.0*center_val)/3.0
End;

Procedure get_next_value;
Begin
  h := h/2.0;
  leftTREE_point[1]  := a+h; rightTREE_point[1]  := b-h;
  leftTREE_value[1]  := F(leftTREE_point[1]);
  rightTREE_value[1]  := F(rightTREE_point[1]);
  sum_of_leaves  := leftTREE_value[1]  + rightTREE_value[1];
  new_sum  := old_sum/2.0 + 2.0*h*(2.0*sum_of_leaves  -
center_val)/3.0
End;

Procedure update_old_values;
Begin
  old_sum  := new_sum;
  old_sum_of_leaves  := sum_of_leaves;
  h := h/2.0;  i := n + 1;
  sum_of_leaves  := 0.0
End;

Procedure add_new_leaves;
Begin
  leftTREE_point[i]  := leftTREE_point[i  DIV  2] - h;
  rightTREE_point[i]  := rightTREE_point[i  DIV  2] - h;
  leftTREE_point[i+1]  := leftTREE_point[i  DIV  2] + h;
  rightTREE_point[i+1]  := rightTREE_point[i  DIV  2] + h;

  leftTREE_value[i]  := F(leftTREE_point[i]);
  rightTREE_value[i]  := F(rightTREE_point[i]);
  leftTREE_value[i+1]  := F(leftTREE_point[i+1]);
  rightTREE_value[i+1]  := F(rightTREE_point[i+1])
End;

Procedure update_sum_of_leaves;
Begin
  sum_of_leaves  := sum_of_leaves  + leftTREE_value[i]  +
    leftTREE_value[i+1]  + rightTREE_value[i]  +
rightTREE_value[i+1]
End;
```

```
Procedure get_value_of_integral;
Begin
  integral_value := new_sum + (new_sum - old_sum)/15.0
End;


Begin
  setup_initial_value;
  get_next_value;

  n := 1; done := TRUE;

  While (ABS(new_sum-old_sum) > TOL) AND done DO Begin
    update_old_values;
    While i < 2*n+1 DO Begin
      add_new_leaves;
      update_sum_of_leaves;
      i := i+2
                End; (* While i < *)

    n := i-1;
    new_sum := old_sum/2.0 + 2.0*h*(2.0*sum_of_leaves
                          - old_sum_of_leaves)/3.0;
    done := (n<1023)
                                    End; (* While
*)

  get_value_of_integral;
  WriteLn; WriteLn;
  WriteLn(' THE INTEGRAL IS ', integral_value:8:5,
        ' THE EXACT ANSWER IS: 4.0');

  WriteLn(' THE NUMBER OF POINTS USED IS: ', 2*n+3:4);
WriteLn

End.
```

۳.۷ لیست برنامه انتگرال گیری به روش گاوس به زبان C

این برنامه بنام *PGAUSSIAN.C* می‌باشد.

```
/*
   ********************************
   **                            **
   **       PGAUSSIAN.C          **
   **                            **
   ********************************


   APPLIED  NUMERICAL  ANALYSIS

   ------------------------------------------------------------------------

     This program implements the Gaussian Quadrature.
     It can handle only 2, 3, 4, 5 terms depending
     on the parameter, number_of_terms.

     This example program solves the Integral given in Example 4.15.
*/

#include  <stdio.h>
#include  <math.h>

     /* values of t and values of w */

#define    t2_a  -0.5773502692          /* 2 TERMS  */
#define    w2_a   1.0

#define    t3_a  -0.7745966692          /* 3 TERMS  */
#define    w3_a   0.5555555556
#define    t3_b   0.0
#define    w3_b   0.8888888889

#define    t4_a  -0.8611363116          /* 4 TERMS  */
#define    w4_a   0.3478548451
#define    t4_b  -0.3399810436
#define    w4_b   0.6521451549

#define    t5_a  -0.9061798459          /* 5 TERMS  */
#define    w5_a   0.2369268850
#define    t5_b  -0.5384693101
#define    w5_b   0.4786286705
#define    t5_c   0.0
#define    w5_c   0.5688888889

   float  x[5];  /* original independent  variable which lies between  a and
b */

   float  t,     /* transformed  interval which lies between  -1 and +1  */
```

```
        a, b,       /* limits of original variable */
        bMINUSa,    /* (b-a)/2   */
 bPLUSa,     /* (b+a)/2   */
        sum;

  int    number_of_point;

  static int number_of_terms[4]  = { 2, 3, 4, 5 };
/*
-------------------------------------------------------------------
   Find x values as functions  of the t values
*/
void compute_x_values()
 {
   switch (number_of_terms[number_of_point])
    {
      case 2:
           x[0] =  bMINUSa*t2_a  + bPLUSa;
           x[1] = -bMINUSa*t2_a  + bPLUSa;
           break;

      case 3:
           x[0] =  bMINUSa*t3_a  + bPLUSa;
           x[1] =                  bPLUSa;
           x[2] = -bMINUSa*t3_a  + bPLUSa;
           break;

      case 4:
           x[0] =  bMINUSa*t4_a  + bPLUSa;
           x[1] =  bMINUSa*t4_b  + bPLUSa;
           x[2] = -bMINUSa*t4_b  + bPLUSa;
           x[3] = -bMINUSa*t4_a  + bPLUSa;
           break;

  case 5:
           x[0] =  bMINUSa*t5_a  + bPLUSa;
           x[1] =  bMINUSa*t5_b  + bPLUSa;
           x[2] =                  bPLUSa;
           x[3] = -bMINUSa*t5_b  + bPLUSa;
           x[4] = -bMINUSa*t5_a  + bPLUSa;
           break;
      }        /* end of switch */
 }       /* end of compute_x_values  */


/*
-------------------------------------------------------------------
   Define the function to be integrated.
*/

float f(x)
```

```
  float x;

{
   return(1.0/exp(x*x));
}    /* end of f */



/*
---------------------------------------------------------------------
   This procedure implements the method of
   GAUSSIAN  QUADRATURE.
*/

float GAUSSIAN_integral(a,   b)
  float a, b;
  {
    int   i;
    float sum;

    switch (number_of_terms[number_of_point])
{
   case 2 : sum = w2_a*f(x[0]) + w2_a*f(x[1]);
  break;
   case 3 : sum = w3_a*f(x[0]) + w3_b*f(x[1])
    + w3_a*f(x[2]);
  break;
      case 4 : sum = w4_a*f(x[0]) + w4_b*f(x[1])
                        + w4_b*f(x[2]) + w4_a*f(x[3]);
                 break;
      case 5 : sum = w5_a*f(x[0]) + w5_b*f(x[1]) + w5_c*f(x[2])
                        + w5_b*f(x[3]) + w5_a*f(x[4]);
                 break;
    }     /* end of switch */
    return(bMINUSa   * sum);
}      /* end of GAUSSIAN_integral    */
main()        /* MAIN  */
{
    a = 0.2;  b = 1.5;  number_of_point  = 3;
    bMINUSa  = (b-a)/2.0;   bPLUSa  = (b+a)/2.0;
    compute_x_values();

     printf("\n\n THE VALUE  OF THE DEFINITE  INTEGRAL  IS:
%.7f \n\n",
   GAUSSIAN_integral(a,   b));
    printf(" THE NUMBER  OF POINTS  CHOSEN  WAS:  %d \n\n",
   number_of_point);
}    /* end of the gaussqdt program */
```

تمرینات فصل هفتم

۱- قانون ذوزنقه‌ای را بکار برده و با انتخاب ... $h = 2, 1, 0.5, 0.25, 0.125,$ دنباله مقادیر تقریبی را برای سازگاری آزمایش نموده و مقدار تقریبی انتگرال زیر را با خطایی کمتراز $.005$ محاسبه کنید.

$$\int_1^3 Ln(x)dx$$

جواب:

$h=2$ $\quad I_1 = 1.0986$

$h=1$ $\quad I_2 = 1.2424$

$h=0.5$ $\quad I_4 = 1.2826$

$h=0.25$ $\quad I_8 = 1.2923$

$h=0.125$ $\quad I_{16} = 1.2953$

۲- در قوانین ذوزنقه‌ای و سیمپسون کران بالای خطا را برحسب h برای محاسبه $\int_1^3 Ln(x)dx$ مشخص نمائید. برای آنکه خطا در محاسبه تقریبی انتگرال فوق کمتر از $.005$ باشد، مقادیر h را معین نمائید. سپس مقدار تقریبی انتگرال را با روش سیمپسون محاسبه کنید.

جواب: خطا در روش ذوزنقه

$$|E| = \frac{b-a}{12}h^2 \left| f''(\xi) \right| \le 0.005$$

$h < 0.173$

$h = 0.125$ یا $n=16$

خطا در روش سیمپسون

$$|E| = \frac{b-a}{180}h^4 \left| f^{(4)}(\xi) \right| \le 0.005$$

$$h^4 \le \frac{75}{1000}$$

$$h \le 0.525$$

$h=0.5$ یا $n=4$

$$I = \frac{h}{3}[f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+f(x_4)] = 1.2953$$

۳- با استفاده از روش انتگرال گیری رامبرگ مقدار انتگرال $\int_1^3 Ln(x)dx$ را با خطائی کمتر از $.005$ بدست آورید.

جواب:

$T_1=1.0986$

$$T_2=1.2424 \quad T_2^{(1)}=\frac{4T_2-T_1}{3}=1.2903$$

$$T_4=1.2826 \quad T_4^{(1)}=\frac{4T_4-T_2}{3}=1.2961 \quad T_4^{(2)}=\frac{4^2T_4^{(1)}-T_2^{(1)}}{15}=1.2958$$

$$T_8=1.2923 \quad T_8^{(1)}=\frac{4T_8-T_4}{3}=1.2956 \quad T_8^{(2)}=\frac{4^2T_8^{(1)}-T_4^{(1)}}{15}=1.2956$$

$$T_8^{(3)}=\frac{4^2T_8^{(2)}-T_4^{(2)}}{4^3-1}=1.2956$$

۴- فرمول انتگرال گیری گاوس چهار نقطه را بدست آورید. انتگرال $\int_1^3 Ln(x)dx$ را با روش فوق حل کرده و مقدار آنرا معین نمائید.

جواب:

$$I=\int_{-1}^{3} \operatorname{Ln} x\, dx=\int_{-1}^{+1} \operatorname{Ln}(2+u)\, du=\int_{-1}^{+1} f(u)\, du$$

$$\bar{I}=w_0f(u_0)+w_1f(u_1)+w_2f(u_2)+w_3f(u_3)$$

$$w_0=w_3=0.3478548$$

$$w_1=w_2=0.6521452$$

$$-u_0=u_3=0.8611363$$

$$-u_1=u_2=0.3399810$$

$\bar{I}=1.2958459$ مقدار تقریبی

$I=1.2958364$ مقدار تحقیقی

۵- مقادیر یک تابع در جدول زیر داده شده است:

x	$f(x)$	x	$f(x)$
1.0	1.543	1.5	2.352
1.1	1.668	1.6	2.577
1.2	1.811	1.7	2.828
1.3	1.971	1.8	3.107
1.4	2.151		

مقدار انتگرال $I=\int_{1.0}^{1.8} f(x)\, dx$ را با روش ذوزنقه با مقادیر $h=0.1$ a) $h=0.2$ b)

c) $h=0.4$ محاسبه کنید. اگر مقادیر جدول از تابع $f(x)=\cosh(x)$ به‌دست آمده باشد، نتایج را با مقادیر دقیق محاسبه کنید.

جواب:

	h	مقدار تقریبی	خطا
a	0.1	1.7683	0.0013
b	0.2	1.7728	0.0058
c	0.3	1.7904	0.0234

مقدار دقیق $I=1.7669731$

۶- انتگرال زیر را به روش سه جمله گاوس محاسبه کنید.

$$I=\int_0^1 \frac{\sin x}{x}dx$$

جواب - $.94608$

۷- نشان دهید که:

$$I=\int_{-\infty}^{+\infty}\frac{1}{\sqrt{2\pi}}\,e^{-\frac{x^2}{2}}dx=1$$

۸- با توجه به اینکه $e^{10}=22026.5$, $e^{-10}=4\times10^{-5}$ انتگرال زیر را به دو انتگرال تفکیک و نتیجه را بدست آورید.

$$I=\int_0^{\infty}\frac{dx}{e^x+e^{-x}}$$

جواب: $I=.785398$

۹- قضیه اعداد اول بیان می‌کند که اعداد اول در فاصله $a<x<b$ تقریباً برابرند با:

$$I=\int_a^b\frac{dx}{\mathrm{Ln}\,x}$$

این انتگرال را بکار برده و مقدار تقریبی اعداد اول در فاصله $99<x<111$ یا به عبارت دیگر مقدار انتگرال را بدست آورید و با مقدار تحقیقی مقایسه کنید.

۱۰- انتگرال زیر را تا چهار رقم با معنی صحیح محاسبه کنید.

$$I=\int_0^1 x^x dx$$

جواب: $.7834$

۱۱- انتگرال زیر را تا چهار رقم با معنی صحیح محاسبه کنید.

$$I=\int_0^{\frac{\pi}{2}}\frac{\cos x}{1+x}\,dx$$

جواب: $.6736$

۱۲- نشان دهید که:

$$I=\int_{\frac{\pi}{2}}^{\frac{\pi}{2}}\sqrt{1-0.25\sin^2(x)}\;dx=1.46746$$

۱۳- مطلوبست محاسبه انتگرال

$$I=\int_0^{\frac{\pi}{2}}e^{\sin(x)}dx$$

جواب: 3.1044

۱۴- y در فاصله $0 \le x \le \pi$ ، y به عنوان تابعی از x با رابطه زیر تعریف شده است.

$$y = -\int_0^x \mathrm{Ln}\left(2\sin\frac{t}{2}\right)dt$$

مقدار ماکزیمم y را تا پنج رقم محاسبه کنید.

جواب - $y = 1.01494$ ، $x = \frac{\pi}{3}$

۱۵- نشان دهید که:

$$I = \int_0^1 \log x \cos x \, dx = .94608$$

۱۶- مطلوبست محاسبه انتگرال $I = \int_0^\infty \left(\frac{e^{-x^2}}{1+x^2}\right) dx$ تا چهار رقم با معنی صحیح.

جواب: *.6716*

۱۷- انتگرال زیر را تا پنج رقم محاسبه کنید:

$$I = \int_0^\infty e^{-x} \mathrm{Ln}\, x \, dx$$

جواب: *.57722* - مقدار واقعی c - که c مقدار ثابت اویلر است.

۱۸- انتگرال‌های زیر را محاسبه کنید:

$$I_1 = \int_0^\infty \frac{x\,dx}{e^x - 1} dx \quad , \quad I_2 = \int_0^\infty \frac{x\,dx}{e^x + 1} dx$$

جواب: $I_1 = 1.64493$ مقدار واقعی $\frac{\pi^2}{6}$، $I_2 = .822467$ مقدار واقعی $\frac{\pi^2}{12}$.

۱۹- نشان دهید که:

$$I = \int_0^1 \frac{(\sin x)^{\frac{3}{2}}}{x^2} dx = 1.9049$$

۲۰- چون روش $\frac{1}{3}$ سیمپسون وقتی $f(x)$ چند جمله‌ای درجه سوم می‌باشد، جواب دقیق را می‌دهد، مقدار انتگرال سه‌گانه زیر باید دقیق باشد. مقدار انتگرال زیرا به روش عددی و تحلیلی بدست آورید.

$$\int_0^1 \int_0^1 \int_0^1 x^3 y z^2 \, dx \, dy \, dz$$

جواب: *0.0416667 = 1/24* روش سیمپسون را بکار می‌بریم:

در $z = 0.5$:

$y = 0.5$, $I_x = (0 + 4/64 + 1/8) / 6 = 1/32$,

$y = 1.0$, $I_x = (0 + 4/32 + 1/4) / 6 = 1/16$,

$I_{z=0.5} = (0 + 4/32 + 1/16)/ 6 = 1/32$ برای $z=0.5$.

در $z = 1.0$:

$y = 0.5$, $I_x = (0 + 4/16 + 1/2) / 6 = 1/8$,

$y = 1.0,\ I_x = (0 + 4/8 + 1) / 6 = 1/4,$

$I_{z=1.0} = (0 + 4/8 + 1/4)/ 6 = 1/8$ برای $z=1.0$.

۲۱ - انتگرال $\iint \sin(x).\sin(y)\ dx\ dy$ روی قسمتی از دایره به شعاع یک که در ربع اول واقع است تعریف شده است. انتگرال را با ثابت گرفتن y و انتخاب $h=0.25$ اول نسبت x و سپس نسبت به y بدست آورید.

جواب: با روش ذوزنقه $I=0.1046$

با روش سیمپسون $I=0.1118$

۲۲ - با تغییر مقدار Δx و Δy روش ذوزنقه‌ای را در دو جهت بکار برده و نشان دهید که کاهش خطا تقریباً متناسب با h^2 می‌باشد.

$$\int_0^1 \int_0^1 (x^2 + y^2)\, dx\, dy$$

جواب:

h	0.5	0.25	0.125	0.0625
I	0.75	0.6875	0.671875	0.667968
خطا	0.0833	0.02083	0.005208	0.001301

۲۳ - چه مقدار جمله در فرمول گاوس لازم است تا انتگرال $x^3+sin(x)*e^{x-3}$ روی فاصله $[-1.5,\ 2.7]$ تا شش رقم با معنی صحیح بدست آید.

۲۴ - روش ضرایب نامعین را بکار برده و فرمول‌های تفاضل مرکزی را بدست آورید.

۲۵ - روش ضرایب نامعین را بکار برده و روش سیمپسون را بدست آورید.

۲۷ - روش ضرای نامعین را بکار برده و روش ذوزنقه‌ای را بدست آورید.

۲۸ - فرمول نیوتن-کوتز را برای تعداد تقسیمات $n=4$ و $n=5$ بدست آورید

۲۹ - انتگرال دوگانه زیر را محاسبه کنید.

$$I=\int_{0.1}^{0.7} \int_{-0.2}^{0.6} e^x \sin y\ dy\ dx$$

جواب: a) به روش ذوزنقه $\Delta x= \Delta y=0.1$ $I=0.140586$

b) به روش سیمپسون $I=0.140587$

c) به روش گاوس $I=0.140585$

۳۰- مطلوب است محاسبه انتگرال دوگانه زیر

$$I=\int\int e^{-x^2y^2}\,dx\,dy$$

روی ناحیه محدود به سهمی $y=x^2$ و $y=2x^2-1$. مقادیر Δx و Δy را به طور مناسب انتخاب کنید.

جواب: با انتخاب $\Delta x=0.02$ و در جهت y چهار فاصله انتخاب کنید.

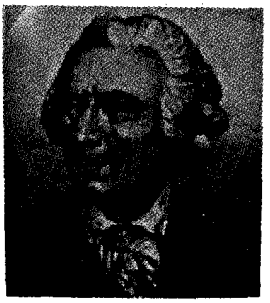
$$I=0.64307$$

فصل ۸

حل عددی معادلات دیفرانسیل معمولی

موضوعات این فصل

* مسأله سیستم جرم - فنر
* حل معادلات دیفرانسیل با استفاده از سری تیلر
* روش اویلر
* روش هیون
* حل دستگاه معادلات دیفرانسیل
* خطای موضعی و ناحیه‌ای
* روش رانج - کوتای مرتبه دوم و چهارم
* روش‌های چندگامه
* روش میلن - سیمپسون یا روش پیش بینی - تصحیح
* روش آدامز - مولتن
* الگوریتم آدامز
* روش رانج - کوتا - فیلبرگ
* روش کول - نیومروف
* حل دستگاه معادلات دیفرانسیل
* برنامه کامپیوتری بنام ADAMS.BAS برای حل دستگاه معادلات دیفرانسیل مرتبه اول از روش آدامز به زبان BASiC می‌باشد.
* برنامه کامپیوتری بنام RKF.PRS برای حل دستگاه معادلات دیفرانسیل مرتبه اول به زبان PACAL می‌باشد.
* برنامه کامپیوتری بنام RKSYST.C برای حل دستگاه معادلات دیفرانسیل به زبان C می‌باشد.
* تمرینات فصل هشتم



لئونهارت اویلر
EULER
(۱۷۰۷ - ۱۷۸۳)

اویلر در سال ۱۷۰۷ در شهر بال سویس به دنیا آمد. وی پر تألیف‌ترین نویسنده در تاریخ در موضوع ریاضیات می‌باشد، نام وی به هر شاخه‌ای از این علم پیوسته است. زمینه اصلی نوشتار او در پهنهٔ ریاضیات کاربردی، به ویژه نظریهٔ حرکت ماه، مسألهٔ سه جسم مکانیک سماوی، ربایش بیضوی، ئیدرولیک، کشتی سازی، و نظریهٔ موسیقی است.

اویلر نویسندهٔ زبر دست کتابهای درسی بود. در بین این کتابها مدخل در آنالیز بینهایت کوچکها به سال ۱۷۴۸ ، اثر بسیار با ارزش حساب دیفرانسیل به سال ۱۷۵۵ و کتاب سه جلدی تأسیس حساب انتگرال مربوط به سال‌های ۱۷۴۷ - ۱۷۶۷ می‌باشد.

اویلر فکر فا کتور انتگرال را در حل معادلات دیفرانسیل به کار گرفت. دانشجویان ریاضی و مهندسی اغلب در سال اول تحصیل خود با معادلات دیفرانسیل اویلر و نیز قضیهٔ اویلر دربارهٔ توابع همگن را فرا می‌گیرند. در ستایش او گفته‌اند: «اویلر را می‌توان، بدون هیچ اغراقی تجسم آنالیز دانست.»

حل عددی معادلات دیفرانسیل معمولی

ابتدا حل عددی معادلات دیفرانسیل با مقدار اولیه را بررسی می‌کنیم. مانند

$$y' = f(x,y) \qquad y(x_0) = y_0$$

(a) $y' = x+y \qquad y(0)=1$

(b) $y' = e^{\sin(x^2+2y)} \qquad y(0) = 1$

البته بیشتر در مواردی که حل به طریق تحلیلی بسادگی امکان پذیرنباشد، از روش عددی جواب معادلات دیفرانسیل را بدست می‌آوریم.

چنانچه معادله دیفرانسیل مرتبه اول را مورد بررسی قرار دهیم با توجه به اینکه معادلات دیفرانسیل از مرتبه بالاتر یا دستگاه معادلات دیفرانسیل قابل تحویل به دستگاه معادلات دیفرانسیل مرتبه اول هستند، بسادگی می‌توان جواب این نوع از معادلات را نیز بدست آورد.

به عنوان مثال دستگاه زیر را در نظر می‌گیریم:

$$\begin{cases} y'' = 2y' + 3z + x^2 \\ z' = 3y' + y^2 + xz^3 \end{cases}$$

با انتخاب:

$$y = y_1 \quad , \quad y' = y_2 \quad , \quad z = y_3$$

نتیجه می‌شود:

$$y_1' = y_2$$

$$y_2' = 2y_2 + 3y_3 + x^2$$

$$y_3' = 3y_2 + y_1^2 + x y_3^3$$

این دستگاه معادلات دیفرانسیل مرتبه اول را می‌توان از روش کلی حل کرد.

هر روشی را که برای حل معادله دیفرانسیل مرتبه اول بتوان بکار برد، به آسانی می‌توان آنرا برای دستگاه معادلات تعمیم داد و در اینصورت انواع مختلف معادلات را می‌توان حل کرد.

دو مساله زیر را در نظر می‌گیریم.

(a) $y'' + y = x$, $y(0)=1$, $y'(0)=2$

(b) $y'' + y = x$, $y(0)=1$, $y(1)=2$

جواب عمومی این معادله برابر است با:

$$y(x) = A\,Cosx + B\,Sinx + x$$

a) $A=1$, $B=1$

b) $A = 1$, $B = \dfrac{1 - Cos(1)}{Sin(1)}$

مساله با مقادیر اولیه

$$y(x) = Cos(x)+Sin(x)+x$$

مساله با مقادیر مرزی

$$y(x)=Cos(x)+\frac{1-Cos(1)}{Sin(1)}Sin(x)+x$$

شرایط داده شده تاثیری در حل تحلیلی مساله ندارند. اما این شرایط برای نوع روش عددی که بکار می‌بریم، بسیار با اهمیت می‌باشند.

در مسایل با مقادیر اولیه اطلاع کافی برای مشخص کردن تابع در یک نقطه، در اختیار داریم، و قادریم که جواب را در همسایگی آن نقطه تعمیم دهیم. ولی در مسایل با مقادیر کرانه‌ای چنین امکانی وجود ندارد.

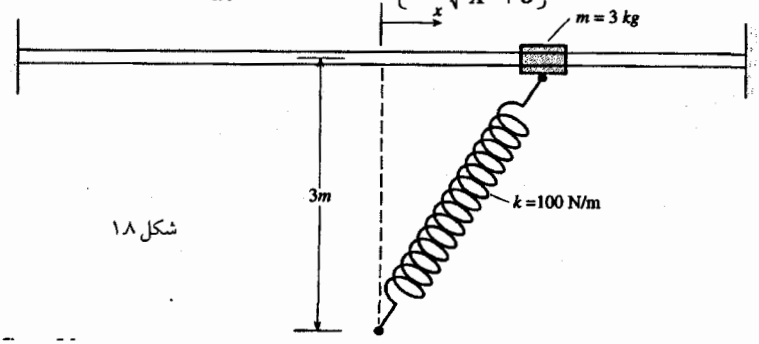
در قسمت بعد طرز تشکیل یک معادله دیفرانسیل مرتبه دوم را در یک سیستم فیزیکی شرح می‌دهیم.

۱۸ مسأله ارتعاش جرم - فنر

جرمی در امتداد یک میله بدون اصطکاک حرکت می‌کند و بوسیله یک فنر به یک مرکز در زیر میله کشیده می‌شود، مطابق شکل ۱۸ که این سیستم را نشان می‌دهد، می‌خواهیم معادله حرکت جرم را تعیین کنیم.

فرض می‌کنیم که فنر بدون کشیده شدن دارای طولی برابر $\sqrt{10} \approx 3.162$ متر باشد. ثابت فنر، k، برابر $100\ N/m$ (نیوتن در متر) و جرم، m، برابر $3Kg$ کیلوگرم می‌باشد. بآسانی دیده می‌شود که حالت تعادل وزنه در $x = 1^m$ (یا $x = 1.0\ m$ - متر) می‌باشد. x فاصله جرم از مرکز میله می‌باشد و y طول فنر است وقتی که جرم در نقطه x می‌باشد. مقدار y_0، طول فنر بدون کشیدگی می‌باشد. ($y_0 = \sqrt{10}$ متر) معادله دیفرانسیلی که حرکت جرم را توضیح می‌دهد، معادله دیفرانسیل مرتبه دوم می‌باشد.

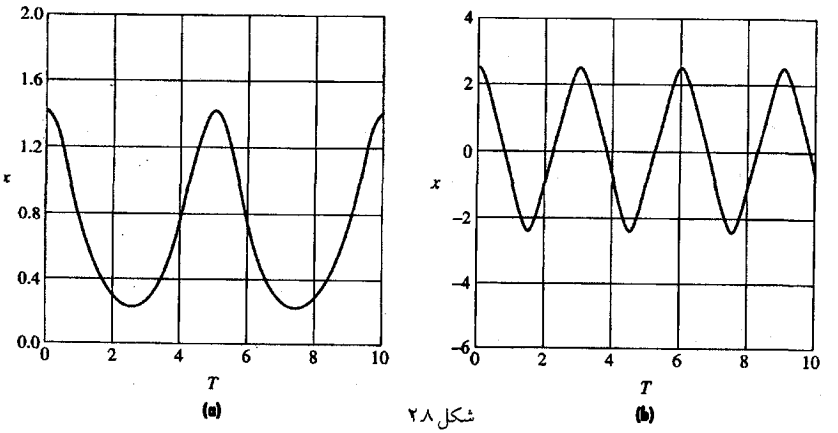
$$\frac{d^2x}{dt^2} = -\frac{k}{m}\ x\ \left[1-\frac{y_0}{\sqrt{x^2+9}}\right]$$



شکل ۱۸

جواب بستگی به این دارد که در ابتداء وزنه به چه فاصله‌ای از مرکز میله در سمت چپ یا راست آن قرار دارد (x_0). سرعت اولیه جرم برابر x'_0 می‌باشد. می‌توانیم این مسأله را بصورت تحلیلی حل کنیم، اما تأکید ما روی حل عددی مسأله می‌باشد، که در این فصل بررسی خواهد شد. ترسیم حرکت جرم در شکل ۲.۸ نشان داده شده است. یک جواب در قسمت (a) با $x_0 = 1.4$ شروع می‌شود و جواب دیگر در قسمت (b) با $x_0 = 2.5$ شروع می‌شود (سرعت اولیه $x'_0 = 0$ می‌باشد) و در شکل نشان داده شده است.

مشاهده می‌شود در قسمت (a) که فنر فقط کمی کشیده می‌شود. جرم از نقطه مرکز میله عبور نمی‌کند. ترسیم با برنامه‌ای بنام $ACSL$ انجام گرفته است.



شکل ۲.۸

۲.۸ حل معادلات دیفرانسیل با استفاده از سری تیلر

معادله دیفرانسیل مرتبه اول $y' = f(x,y)$, $y(x_0) = y_0$ را بررسی می‌کنیم. می‌پذیریم که این معادله دارای جواب یکتا در فاصله $[x_0, b_0]$ می‌باشد. $y(x)$ را در مجاورت نقطه x_i در این فاصله بسط می‌دهیم:

$$y(x)=y(x_i)+(x-x_i)\,y'(x_i)+\frac{(x-x_i)^2}{2}y''(x_i)+\ldots+\frac{(x-x_i)^n}{n!}y^{(n)}(x_i) \qquad (1)$$
$$+\frac{(x-x_i)^{n+1}}{(n+1)!}y^{(n+1)}(\xi) \qquad x\in[x_0,b],\ \xi\in(x_i,x)$$

بخصوص اگر $x = x_{i+1}$ در اینصورت $x_{i+1} - x_i = h$ و خواهیم داشت:

$$y(x_{i+1})= y(x_i)+hy'(x_i)+\frac{h^2}{2}y''(x_i)+\ldots+\frac{h^n}{n!}y^{(n)}(x_i)+\frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_i)$$
$$x_i < \xi_i < x_{i+1} \qquad (2)$$

اگر از باقیمانده $\frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_i)$ صرفنظر کنیم، جواب تقریبی رادر نقطه x_{i+1} خواهیم داشت:

$$y(x_{i+1}) \approx y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \ldots + \frac{h^n}{n!} y^{(n)}(x_i) \qquad (3)$$

برای اینکه معادله (3) را بکار بریم لازم است که $y^{(n)}(x_i), \ldots, y'(x_i), y(x_i)$ را محاسبه کنیم.

اگر x_i و $y(x_i)$ معین باشند، مشتقات مطابق زیر حساب می‌شوند.

$$y'(x_i) = f(x_i, y(x_i))$$

$$y''(x) = \frac{d}{dx} f(x,y(x)) = \frac{\delta f}{\delta x} + \frac{\delta f}{\delta y}\frac{dy}{dx} = f_x(x,y) + f_y(x,y)\, y'(x)$$

$$y''(x) = f_x(x,y) + f_y(x,y)f(x,y)$$

$$y^{(n)}(x) = \frac{d^{n-1}}{dx^{n-1}} f(x,y(x)) = \frac{d}{dx} y^{(n-1)}(x)$$

مقدار $y(x_0)$ داده شده است، بنابراین با طرز عمل فوق $y(x_1)$ می‌تواند با خطای $O(h^n)$ محاسبه شود. برای محاسبه $y(x_2)$ لازم است که $y(x_1)$ را دقیقاً حساب کرده، تا بتوان $y'(x_1)$، $y''(x_1)$ را محاسبه کرد. با بکار بردن مقدار تقریبی $y(x_1)$ این مشتقات با خطای مربوطه به دست می‌آیند. نتیجه بعنوان آلگوریتم تیلر در معادله دیفرانسیل زیر داده شده است:

$$y' = f(x,y)$$

$$y_0 = y(x_0)$$

$$y_i^{(k)} = \frac{d^{k-1}}{dx^{k-1}} f\left(x, y(x)\right)\Big|_{(x_0, y_0)}$$

$$k = 0,1,2, \ldots, n$$

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!} y''_i + \ldots + \frac{h^n}{n!} y_i^{(n)}$$

$$x_{i+1} = x_i + h$$

$$i = 0,1, \ldots, m-1$$

اعداد $y_m, \ldots, y_2, y_1$ مقادیر تقریبی $y(x_m), \ldots, y(x_2), y(x_1)$ می‌باشند.

مثال ۱۸ آلگوریتم تیلر از مرتبه 3 را در مورد معادله دیفرانسیل زیر بکار برده و مقادیر تقریبی $y(.2)$، $y(.1)$ را بدست آورید.

$$y' = x + y \quad , \quad y(0) = 1 \quad , \quad h = .1$$

جواب – $y''(x)$ ، $y'''(x)$ بسهولت بدست می‌آیند.

$$y'' = \frac{d}{dx}(x + y(x)) = 1 + y' = 1 + x + y$$

$$y''' = \frac{d}{dx} y'' = \frac{d}{dx}(1 + y') = 1 + x + y$$

در صورتی که الگوریتم تیلر را با $x_0 = 0$ ، $y_0 = 1$ و مشتقات بدست آمده، بکار بریم.

$$y_{i+1} = y_i + h(x_i + y_i) + \frac{h^2}{2}(1 + x_i + y_i) + \frac{h^3}{6}(1 + x_i + y_i)$$

$$y_1 = 1.0 + .1(0+1.0) + \frac{(.1)^2}{2}(1.0 + 0 + 1.0) + \frac{(.1)^3}{6}(1.0 + 0 + 1.0)$$

$y_1 \approx 1.11033 \qquad x_1 = .1$

$$y_2 = y_1 + .1(.1 + y_1) + \frac{(.1)^2}{2}(1.0 + .1 + y_1) + \frac{(.1)^3}{6}(1.0 + .1 + y_1)$$

$y_2 \approx 1.24278 \quad , \quad x_2 = .2$

برای برآورد خطا می‌توان مشتق مرتبه چهارم را بسادگی بدست آورد.

$y^{(4)} = 1 + x + y$

و خطا برابر $\frac{h^4}{4!}(1+x_i+y_i)$ و یا برابر 0.00000417 می‌باشد.

۳۸ روش اویلر[1]

حل عددی معادله مرتبه اول $y' = f(x,y) \quad y(x_0) = y_0$ را بررسی می‌کنیم. فرض کنیم فاصله‌ای را که لازم است جواب معادله بر روی آن بررسی شود، به زیر فاصله‌های مساوی تقسیم کرده باشیم، می‌کوشیم جواب مساله را بعنوان یک سری از نقاط ... , y_r , ... , y_1, y_2 پیدا نمائیم.

y_r جواب تقریبی عددی $y(x_r)$ را در $x = x_r$ نشان می‌دهد.

یعنی: $y_r \approx y(x_r)$

اگر بتوانیم y_{r+1} را با دانستن y_r پیدا نمائیم. در اینصورت، می‌توانیم y_1 را از y_2 y_0 را از y_1 و ... y_{r+1} را از y_r بدست آوریم.

(توجه شود که منظور از جواب معادله دیفرانسیل محاسبه مقدار تابع $y(x)$ در نقاط مورد نظر است. $y(x_1)$, $y(x_2)$ ... $y(x_r)$)

می‌دانیم که: $y(x_{r+1}) = y(x_r + h)$

$$y(x_{r+1}) = y(x_r) + h\,y'(x_r) + \frac{h^2}{2!}y''(x_r) + \ldots$$

$$y(x_{r+1}) = y(x_r) + h\,y'(x_r) + O(h^2)$$

$$y(x_{r+1}) \approx y(x_r) + hf(x_r, y(x_r))$$

$r = 0,1,2,\ldots$

و می‌توانیم ... , y_3 , y_2 , y_1 را با استفاده از این رابطه محاسبه نمائیم.

مثال ۲۸ معادله دیفرانسیل را با توجه به شرط اولیه به روش اویلر حل کنید.

$$\frac{dy}{dx} = -2x - y \qquad y(0) = -1$$

محاسبه به سهولت انجام می‌گیرد. با انتخاب $h = 0.1$ جواب در جدول ۱۸ نوشته شده است.

1- Euler's Method

جدول ۱۸

x_n	y_n	y'_n	hy'_n
0.0	−1.00000	1.00000	0.10000
0.1	−0.90000	0.70000	0.07000
0.2	−0.83000	0.43000	0.04300
0.3	−0.78700	0.18700	0.01870
0.4	−0.76830	−0.03170	

جواب تحلیلی *0.81096* − و خطا *0.04266* − می‌باشد.

۴۸ روش هیون[1]

این روش در مواقعی بکار می‌رود که دقت بیشتری برای جواب تقریبی ضروری است. برای حل معادله دیفرانسیل $y' = f(x,y)$, $y(x_0) = y_0$ می‌کوشیم با انتگرال گیری از طرفین این تساوی، جواب را بدست آوریم. اگر $x_1 = x_0 + h$ داریم:

$$\int_{x_0}^{x_1} y'(x)\,dx = \int_{x_0}^{x_1} f\left[x, y(x)\right] dx \qquad (1)$$

با ساده کردن طرف چپ، خواهیم داشت:

$$y(x_1) = y(x_0) + \int_{x_0}^{x_1} f\left[x, y(x)\right]\, dx \qquad (2)$$

در طرف راست، با روش ذوزنقه‌ای می‌توان مقدار تقریبی انتگرال را جایگزین کرد.

$$y(x_1) = y(x_0) + \frac{h}{2}\left[f\left(x_0, y(x_0)\right) + f\left(x_1, y(x_1)\right)\right] + R \qquad (3)$$

این مقدار y_1 را بعنوان جواب تقریبی در نظر می‌گیریم. اگر بجای $y(x_1)$ در طرف راست معادله (3) مقدار تقریبی آنرا از روش اویلر یا جواب «پیش بینی شده» قرار دهیم و از کلیه باقیمانده‌ها صرفنظر نمائیم، نتیجه می‌شود که:

$$y_1 = y_0 + \frac{h}{2}\left[f(x_0, y_0) + f\left(x_1, y_0 + h f(x_0, y_0)\right)\right] \qquad (4)$$

و در حالت کلی

$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f\left(x_{i+1}, y_i + h f(x_i, y_i)\right)\right] \qquad (5)$$

$x_{i+1} = x_i + h$ و $i = 0,1,2,\ldots$

این جواب تقریبی جدید را جواب «تصحیح شده» می‌نامیم.

۵۸ حل دستگاه معادلات دیفرانسیل

می‌توانیم با بسط آلگوریتم اویلر دستگاه معادلات دیفرانسیل مرتبه اول را حل نموده و جواب‌های تقریبی آنرا که با $y_n(x_i) = y_{ni}$ نشان می‌دهیم، بدست آوریم.

$y'_1 = f_1(x, y_1, y_2, \ldots, y_n)$ $\qquad$ $y_1(x_0) = y_{10}$

۱− Heun's Method

$$y'_2 = f_2(x,y_1,y_2,...,y_n) \qquad y_2(x_0) = y_{20}$$

. . . . . .

$$y'_n = f_n(x,y_1,y_2,...,y_n) \qquad y_n(x_0) = y_{n0}$$

برای پیدا کردن مقادیر تقریبی جواب‌های $y_1(x)$,$y_2(x)$,...,$y_n(x)$ در نقاط x_1 ,x_2 , ... , x_m آلگوریتم اویلر زیر را بکار می‌بریم:

$$\begin{cases} y_{1\,i+1} = y_{1i} + h\, f_1\,(x_i, y_{1i}, y_{2i}, ..., y_{ni}) \\ y_{2i+1} = y_{2i} + h\, f_2\,(x_i, y_{1i}, y_{2i}, ..., y_{ni}) \\ \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \\ y_{ni+1} = y_{ni} + h\, f_n\,(x_i, y_{1i}, y_{2i}, ..., y_{ni}) \\ x_{i+1} = x_i + h \qquad\qquad i = 1, 2, 3, ..., m \end{cases}$$

مثال ۳.۸ روش اویلر را برای معادله دیفرانسیل $y' = 2y$ بکار برده و با توجه به اینکه:

$$x_0 = 1 \quad , \quad y_0 = 1 \quad , \quad h = .05$$

جواب معادله را در نقاط x_1, x_2, x_3 بدست آورید.

$$y_1 = y_0 + h\, f(x_0,y_0) = 1 + (.05)(2) = 1.1 \qquad , x_1 = .05$$

$$y_2 = y_1 + h\, f(x_1,y_1) = 1.1 + (.05)(2.2) = 1.21 \quad , x_2 = .1$$

$$y_3 = y_2 + h\, f(x_2,y_2) = 1.21 + (.05)(2.42) = 1.33 \quad , x_3 = .15$$

$$x_{i+1} = x_i + h \qquad \text{و} \quad i = 0,1,2,3,...,m-1$$

جواب تحلیلی معادله دیفرانسیل فوق $y=e^{2x}$ و جواب‌های آن تا چهار رقم اعشار عبارتند از:

$y(.05)=1.1052$

$y(.10)=1.2214$

$y(.15)=1.3499$

مثال ۴.۸ روش اویلر را برای بدست آوردن جواب معادله دیفرانسیل $y'=2x$ روی فاصله $[0,1]$ با فرض $y(0)=0$ و $h=.1$ بدست آورید.

جواب دقیق معادله $y = x^2$ می‌باشد. از روش اویلر داریم:

$$y_{i+1} = y_i + 2h\, x_i$$

در جدول ۲.۸ مقادیر x_i, y_i, $y(x_i)$ برای $h=.1$ نوشته شده است.

i	x_i	y_i	$y(x_i)$
0	0.0	0.0000	0.0000
1	.1	.000	.0100
2	.2	.0200	.0400
3	.3	.0600	.0900
4	.4	.1200	.1600
5	.5	.2000	.2500
6	.6	.3000	.3600
7	.7	.4200	.4900
8	.8	.5600	.6400
9	.9	.7400	.8100
10	1.0	.9200	1.00.00

مثال ۵۸ با استفاده از الگوریتم هیون جواب معادله دیفرانسیل زیر را در نقطه $x_1 = .05$ و $x_2 = .1$ بدست آورید.

$y' = 2y \quad , \quad x_0 = 0 \quad , \quad y_0 = 1 \quad , \quad h = .05$ جواب –

$$x_1 = .05 \ , \ y_1 = y_0 + \frac{h}{2}\left[f(x_0,y_0) + f(x_1,y_0 + h\, f(x_0,y_0))\right]$$

$$y_1 = 1 + \frac{.05}{2}\left[2 + f(.05\ ,\ 1.10)\right] = 1.105$$

$$x_2 = .10 \ , \ y_2 = y_1 + \frac{h}{2}\left[f(x_1,y_1) + f(x_2,y_1 + h\, f(x_1,y_1))\right]$$

$$y_2 = 1.105 + \frac{.05}{2}\left[2.21 + f(.10\ ,\ 1.2155)\right] = 1.221025$$

مثال ۶۸ دستگاه معادلات دیفرانسیل مرتبه‌اول زیر را با روش اویلر حل نمایید، در صورتیکه: $h = 0.1$

$$x_0 = 0 \quad , \quad y_1(0) = 0 \quad , \quad y_2(0) = 1 \quad ,$$

$$\begin{cases} y'_1 = y_1 + y_2 \\ y'_2 = -\ y_1 + y_2 \end{cases}$$

با استفاده از یک برنامه کامپیوتری، جواب تقریبی برای $x_0, x_1, \ldots, x_9$ در جدول زیر آمده است و ستون آخر، نتیجه جواب تحلیلی را نشان می‌دهد.

$$\begin{cases} y_1(x) = e^x \, Sin(x) \\ y_2(x) = e^x \, Cos(x) \end{cases}$$

توجه شود که این روش دارای دقت زیادی نمی‌باشد.

x_i	y_{1i}	y_{2i}	$y_1(x_i)$	$y_2(x_i)$
.00	.00000	1.00000	.00000	1.00000
.10	.10000	1.11000	.11033	1.09965
.20	.22000	1.20000	.24266	1.19706
.30	.36200	1.29800	.39891	1.28957
.40	.52800	1.39160	.58094	1.37406
.50	.71996	1.47796	.79044	1.44689
.60	.93975	1.55376	1.02885	1.50386
.70	1.18910	1.61516	1.29730	1.54020
.80	1.46953	1.65777	1.59651	1.55055
.90	1.78226	1.67659	1.92667	1.52891

مثال ۷.۸ آلگوریتم اویلر را بکار برده جواب دستگاه معادلات مرتبه اول زیر را در $x_2 = 1.04$ بدست آورید. $(h=0.02)$

$$y'_1 = x + y_1 + y_2 + y_3 = f_1(x,y_1,y_2,y_2) \qquad y_{1,0}(1) = 3$$
$$y'_2 = x + 2y_1 + 4y_2 + 6y_3 = f_2(x,y_1,y_2,y_3) \qquad y_{2,0}(1) = 4$$
$$y'_3 = x^2 + 3y_1 + 10y_3 = f_3(x,y_1,y_2,y_3) \qquad y_{3,0}(1) = 5$$

$$y_{1,1} = 3 + .02(1 + 3 + 4 + 5) = 3.26$$
$$y_{2,1} = 4 + .02(1 + 6 + 16 + 30) = 5.60$$
$$y_{3,1} = 5 + .02(1 + 9 + 50) = 6.25$$

برای $x_1 = 1.02$

$$y_{1,2} = 3.26 + .02(1.02 + 3.26 + 5.06 + 6.20) = 3.5708$$
$$y_{2,2} = 5.06 + .02(1.02 + 6.52 + 20.24 + 37.20) = 6.3596$$
$$y_{3,2} = 6.20 + .02(1.0404 + 9.78 + 62.00) = 7.656408$$

برای $x_2 = 1.04$

مثال ۸.۸ مطلوبست حل معادله دیفرانسیل زیر به روش عددی اویلر و روش تحلیلی

$y' = xy$ $\qquad y(2)=1 \qquad (x_0 = 2, y_0 = 1)$

با انتخاب $h=.1$ داریم:

$$y_{r+1} = y_r + h f(x_r,y_r)$$
$$y_{r+1} = y_r + hx_r y_r$$

$$y_1 = y_0 + (.1)\, x_0\, y_0 = 1 + .2 = 1.2$$
$$y_2 = y_1 + hx_1\, y_1 = 1.2 + (.1)(2.1)(1.2) = 1.452$$
$$y_3 = y_2 + hx_2\, y_2 = 1.452 + (.1)(2.2)(1.452) = 1.77144$$
$$y_4 = \ldots$$

با حل تحلیلی معادله دیفرانسیل فوق خواهیم داشت:

$$y(x) = e^{(x^2/2 - 2)}$$

$y(2.1) = 1.227525$	$y_1 = 1.2$	خطا $E \approx .027$
$y(2.2) = 1.521962$	$y_2 = 1.452$	$E \approx .07$
$y(2.3) = 1.905987$	$y_3 = 1.77144$	$E \approx .13$

مشاهده می‌شود که خطا مرتباً افزایش می‌یابد، زیرا که خطا در محاسبه y_{r+1} نسبت به y_r که خود با خطا محاسبه شده است، باعث انباشتگی خطا می‌شود.

۶۸ خطای موضعی و ناحیه‌ای

خطای موضعی را با جایگذاری جواب واقعی در فرمول تعریف می‌کنیم:

$$L = y(x_{r+1}) - y(x_r) - h\, f(x, y(x_r)) \qquad (1)$$

که خطای حاصل در y_{r+1} بر اثر برش سری می‌باشد و $y_r \approx y(x_r)$ فرض شده است در صورتیکه خطای ناحیه‌ای که بصورت زیر تعریف می‌شود، شامل کلیه خطاهائی است که در محاسبه نقاط قبلی بوجود آمده است.

$$G = y(x_r) - y_r \qquad (2)$$

با توجه به رابطه اویلر $y_{r+1} = y_r + h\, f(x_r, y_r)$

چنانچه $y_r = y(x_r)$ یعنی با توجه به (2) خطای ناحیه‌ای در x_r صفر باشد، خواهیم داشت: $L =$

$$y(x_{r+1}) - y(x_r) - hf(x, y(x_r)) = y(x_{r+1}) - y_r - hf(x, y_r)$$

$$L = y(x_{r+1}) - y_{r+1}$$

اگر در محاسبه y_{r+1} که با استفاده از y_r بدست می‌آید، y_r بدون خطا باشد، خطای y_{r+1} را خطای موضعی *(LOCAL)* می‌نامند، در غیر اینصورت y_r خطای ناحیه‌ای *(GLOBAL)* نامیده می‌شود.

$$y_r = y(x_r)$$

رابطه (1) را می‌توان چنین نوشت:

$$L = y(x_{r+1}) - y(x_r) - hf(x_r, y(x_r))$$

$$L = (y(x_r) + hy'(x_r) + \frac{h^2}{2!} y''(x_r) + \ldots) - y(x_r) - hy'(x_r)$$

$$L = \frac{h^2}{2!} y''(x_r) + \frac{h^3}{3!} y'''(x_r) + \ldots$$

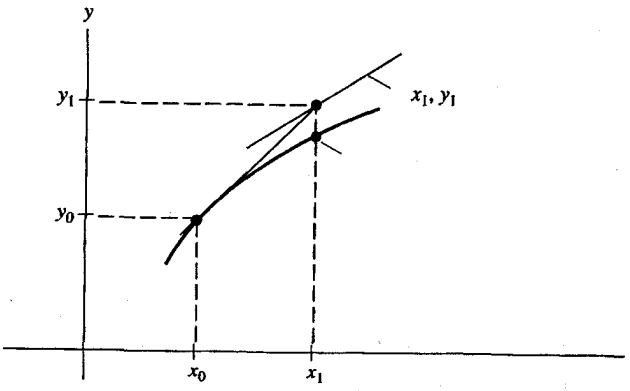
$$L = O(h^2)$$

یعنی خطائی موضعی از مرتبه h^2 می‌باشد.

تفسیر هندسی این خطاها در زیر مشاهده می‌شود.

a- جواب $y' = f(x,y)$ که از $(x_1, y(x_1))$ می‌گذرد، تحقیقی است

b- جواب $y' = f(x,y)$ که از (x_1, y_1) می‌گذرد، تقریبی است.



برای آنکه خطای ناحیه‌ای هر چه بیشتر کوچک باشد، می‌کوشیم روشهائی را پیدا نمائیم که خطای موضعی برای آنها کوچک باشد.

۷.۸ روش‌های رانج - کوتا[1]

روش‌های متعددی برای حل عددی معادلات دیفرانسیل با مقدار اولیه به فرم زیر وجود دارد.

$$y' = f(x,y) \quad , \quad y(x_0) = y_0$$

با بسط سری تیلور و انتخاب n جمله اول بسط، می‌توان دقت جواب معادله دیفرانسیل را افزایش داد. در روش اویلر دو جمله اول بسط و تا $y'(x_0)$ مورد استفاده قرار می‌گیرد، لیکن ملاحظه می‌شود که مشتقات مرتبه بالاتر یعنی $y''(x_0)$ و $y'''(x_0)$ و برای دقت بیشتر جواب می‌توانند مورد استفاده قرار گیرند.

دو ریاضیدان آلمانی کوتا *(Kutta)* و رانج *(Runge)* روشهائی را ارائه دادند که دارای دقت و کارآئی بیشتر نسبت به روش‌های قبلی می‌باشند، ضمن آنکه بجای محاسبه مستقیم مشتقات مرتبه بالاتر تنها با استفاده از تابع $f(x_n, y_n)$ بازاء مقادیر مختلف نتیجه حاصل می‌شود.

1- Runge- Kutta Methods

روش رانج -کوتای مرتبه دوم

$$y' = f(x\ ,\ y) \qquad y(x_0) = y_0$$

$$y_{n+1} = y_n + ak_1 + bk_2 \qquad (1)$$

$$k_1 = h\ f\ (x_n,y_n)$$

$$k_2 = h\ f\ (x_n + \alpha\ h\ ,\ y_n + \beta\ k_1)$$

رانج -کوتا بجای بسط سری تیلور تا مشتق مرتبه دوم، این فرمول *(1)* را پیشنهاد کرده‌اند.

انتظار داریم که این فرمول نسبت به فرمول اویلر خطای موضعی کوچکتری داشته باشد، در واقع α, b, a را چنان انتخاب می‌نمائیم که خطای موضعی هر چه ممکن است کوچک گردد:

خطای موضعی $= L = y\ (x_{n+1}) - y_{n+1}$

و برای $y(x_{n+1})$ داریم:

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}\ y''(x_n) + \ldots \qquad (2)$$

y_{n+1} را از رابطه *(1)* می‌توان بدست آورد:

$$y_{n+1} = y_n + ahf\ (x_n,y_n) + bhf((x_n + \alpha h)\ ,\ y_n + \beta hf(x_n,y_n)) \qquad (3)$$

با توجه به اینکه می‌دانیم:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}\ f''(x) + \ldots$$

بسط تیلور تابع دو متغیره زیر را مورد استفاده قرار می‌دهیم:

$$f(x+h\ ,\ y+k)=f(x,y)+\quad h\ \frac{\delta\ f(x,y)}{\delta\ x} + k\ \frac{\delta\ f(x,y)}{\delta\ y} + \frac{h^2}{2}\ \frac{\delta^2\ f(x,y)}{\delta\ x^2}$$

$$+ \frac{2hk\ \delta^2 f}{2\ \delta x\ \delta y} + \frac{k^2\ \delta^2 f}{2\ \delta\ y^2} + \ldots$$

پس با جایگذاری در رابطه *(3)* خواهیم داشت:

$$y_{n+1} = y_n + ahf_n + bh\ [f_n + \alpha h\ \frac{\delta f_n}{\delta x} + \beta hf_n\ \frac{\delta f_n}{\delta y} + O(h^2)]$$

$$y_{n+1} = y_n + (a+b)\ hf_n + \alpha bh^2\ \frac{\delta f_n}{\delta x} + \beta bh^2\ f_n\ \frac{\delta f_n}{\delta y} + O(h^3) \qquad (4)$$

چنانچه رابطه *(2)* را نیز بر حسب مشتقات نسبی بنویسیم، با توجه به رابطه زیر داریم:

$$y' = f(x,y)$$

$$y'' = \frac{d}{dx}\ f(x,y) = \frac{\delta f}{\delta x} + \frac{\delta f}{\delta y}\ \frac{dy}{dx} = \frac{\delta f}{\delta x} + f\frac{\delta f}{\delta y}$$

$$y'^n = \frac{d}{dx}\ y'' = \ldots$$

$$y(x_{n+1}) = y(x_n) + hf(x_n,y(x_n)) + \frac{h^2}{2!}\left[\frac{\delta f(x_n,y(x_n))}{\delta x} + f(x_n,y(x_n))\frac{\delta f(x_n,y(x_n))}{\delta y}\right] + O(h^3) \qquad (5)$$

از تفاضل رابطه (5) و (4) خواهیم داشت:

$$L = y(x_{n+1}) - y_{n+1}$$

با انتخاب a, b, α, β سعی می‌کنیم طرف دوم این رابطه هر چه ممکن است کوچک شود. در صورت تساوی چهار جمله اول داریم:

$$y_n = y(x_n)$$

$$(a+b)h = h \qquad a+b=1 \qquad (6)$$

$$\alpha b h^2 \frac{\delta f_n}{\delta x} + \beta b h^2 f_n \frac{\delta f_n}{\delta y} = \frac{h^2}{2}\frac{\delta f}{\delta x} + \frac{h^2}{2} f_n \frac{\delta f_n}{\delta y}$$

$$\alpha b = ½ \quad , \quad \beta b = ½ \qquad (7)$$

از روابط (6) و (7) که سه معادله با چهار مجهول را نشان می‌دهند (یکی از مجهولات را به دلخواه انتخاب می‌کنیم) خواهیم داشت:

$a = 0$	$b = 1$	$\alpha = \beta = \frac{1}{2}$
$a = ½$	$b = ½$	$\alpha = \beta = 1$
$a = ¼$	$b = ¾$	$\alpha = \beta = \frac{2}{3}$

هر یک از این انتخابها یک فرمول رانج - کوتا می‌باشند و در کلیه آنها خطای موضعی برابر $O(h^3)$ می‌باشد. (روش رانج - کوتای مرتبه دوم)

مناسب‌ترین انتخاب برای روش رانج - کوتا چنین است:

$$y_{n+1} = y_n + ½(k_1 + k_2)$$
$$k_1 = h f(x_n,y_n)$$
$$k_2 = h f(x_n + h , y_n + k_1)$$

واضح است که این روش بسیار دقیق‌تر از روش اویلر می‌باشد.

و خطا برابر $O(h^3)$ می‌باشد.

مثال ۹۸ معادله دیفرانسیل زیر را با استفاده از روش رانج - کوتای مرتبه دو بررسی نمایید.

$$y' = xy \qquad y(2)=1$$

با انتخاب $h=.1$ داریم:

$$y' = f(x,y) = xy \quad , \quad x_0 = 2 \quad , \quad y_0 = 1$$
$$y_1 = y_0 + ½(k_1 + k_2)$$
$$k_1 = hf(x,y) = .1 \times 2 \times 1 = .2$$
$$k_2 = hf(x_0 + h , y_0 + k_1) = (.1)\ f(2.1,1.2) = (.1)(2.1)(1.2) = .252$$
$$y_1 = 1 + ½(.2 + .252) = 1.226$$

قبلاً این مثال را با روش اویلر حل کردیم و جواب آن برابر بود با:

$$y_1 = 1.2$$

و از روش تحلیلی جواب برابر است با:

$y(2.1)=1.22753$

و همانطور که توضیح داده شد، این روش بسیار دقیق‌تر از روش اویلر است.

روش رانج کوتای مرتبه چهارم

می‌توانیم روشی را که در قسمت قبل بکار بردیم، مجدداً مورد استفاده قرار دهیم؛ y_{n+1} را بر حسب y_n چنان بدست می‌آوریم که خطای موضعی هر چه ممکن است کوچک شود:

روش رانج - کوتا از مرتبه چهار معروف‌ترین و پر استفاده‌ترین این روشهاست.

$$y_{n+1} = y_n + ak_1 + bk_2 + ck_3 + dk_4$$
$$k_1 = hf\,(x_n,y_n)$$
$$k_2 = hf\,(x_n + \alpha h\ ,\ y_n + \beta k_1)$$
$$k_3 = hf\,(x_n + \gamma h\ ,\ y_n + \varphi k_1 + \theta k_2)$$
$$k_4 = hf\,(x_n + \delta h\ ,\ y_n + \rho k_1 + \mu K_2 + \eta k_3\ )$$

بسط سری تیلور $y(x_n+h)$ را تا مشتق چهارم بدست می‌آوریم و همچنین بسط توابع دو متغیره تیلور را برای k_2 و k_3 و k_4 بدست آورده و نتیجه را در y_{n+1} قرار می‌دهیم با مقایسه این دو بسط به 9 معادله و 13 مجهول می‌رسیم و خطای موضعی برابر است با: $L = O(h^5)$

بعد از انتخاب دلخواه 4 مجهول می‌توان 9 مجهول دیگر را محاسبه نمود: $\alpha=\frac{1}{2}\ \mu$, $\rho = 0$, $\varphi = 0$, رابطه مشهور زیر که یکی از جوابها می‌باشد، بدست می‌آید.

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$k_1 = h\,f\,(x_n,y_n)$$
$$k_2 = h\,f\,(x_n + \tfrac{1}{2}\,h\ ,y_n + \tfrac{1}{2}\,k_1)$$
$$k_3 = h\,f\,(x_n + \tfrac{1}{2}\,h\ ,\ y_n + \tfrac{1}{2}\,k_2)$$
$$k_4 = h\,f\,(x_n + h\ ,y_n + k_3)$$

مثال ۱۰.۸ معادله دیفرانسیل زیر را به روش رانج - کوتای مرتبه چهار حل کنید:

$$\frac{dy}{dx} = -2x-y \quad , \quad y\,(0) = -1 \quad , \quad h = 0.1$$

از تابع $f(x\ ,\ y) = -2x-y$ استفاده کرده و نتیجه تا $x = 0.5$ محاسبه شده و در جدول ۳۸ ثبت شده است.

جدول ۳۸

x_n	y_n	k_1	k_2	k_3	k_4	Average
0.0	−1.0000	0.1000	0.0850	0.0858	0.0714	0.0855
0.1	−0.9145	0.0715	0.0579	0.0586	0.0456	0.0583
0.2	−0.8562	0.0456	0.0333	0.0340	0.0222	0.0337
0.3	−0.8225	0.0222	0.0111	0.0117	0.0011	0.0115
0.4	−0.8110	0.0011	−0.0090	−0.0085	−0.0181	−0.0086
0.5	−0.81959					

۸.۸ روش‌های چندگامه[1]

روش چند گامه مقدار جواب y_{n+k} یا $y(x_{n+k})$ را بعنوان تابعی از مقادیر قبلی ، y_{n+k-1} ، ... ، y_{n+k-2} ، y_n تعریف می‌کند. در این حالت دارای K گام هستیم، تا از جواب y_n به جواب y_{n+k} برسیم، اگر بخصوص $k=1$، به روش‌های تک گام بر می‌گردیم که در فصل قبل بحث شده است.

روش حل معادله دیفرانسیل را صریح گوئیم اگر مقدار جواب y_{n+1} را بتوانیم مستقیماً پیدا کنیم، و غیر صریح گوئیم اگر فرمول در سمت راست نیز شامل مقدار متغیر جواب باشد و مقدار y_{n+1} را نتوان برحسب y_n ، ... ، y_1 ، y_0 بیان کرد.

۹.۸ روش میلن[2] - سیمپسون یا روش پیش بینی - تصحیح

نظر کلی این روش برای حل معادله $y'=f(x,y)$ چنین است که ابتدا روش درونیابی را برای بدست آوردن مقدار تقریبی y_{n+1} بکار می‌بریم. (پیش بینی)[3] و سپس این مقدار را با فرمول دیگری بهبود می‌بخشیم (تصحیح[4]). برای درونیابی از روش کرس[5] و برای روش تصحیح از فرمول کوتز می‌توان استفاده کرد.

فرض می‌کنیم جواب معادله دیفرانسیل را در نقاط:

$$(x_{n-3}, y_{n-3}) \ , \ (x_{n-2}, y_{n-2}) \ , \ (x_{n-1}, y_{n-1}) \ , \ (x_n \ , \ y_n)$$

بدست آورده‌ایم و اکنون می‌خواهیم جواب را در x_{n+1} با دقت بیشتری تعیین کنیم. انتگرال زیر را در نظر می‌گیریم:

$$\frac{dy}{dx} = f(x,y),$$

$$\int_{x_{n-3}}^{x_{n+1}} \left(\frac{dy}{dx}\right) dx = \int_{x_{n-3}}^{x_{n+1}} f(x,y)\,dx = \int_{x_{n-3}}^{x_{n+1}} P_2(x)\,dx;$$

$$y_{n+1} - y_{n-3} = \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}) + \frac{28}{90}h^5 y^{v}(\xi_1), \qquad x_{n-3} < \xi_1 < x_{n+1}. \tag{1}$$

انتگرال تابع $f(x \ , \ y)$ را با جایگزینی با یک چند جمله‌ای درونیاب مرتبه دوم که به سه نقطه در x_{n-2} و x_{n-1} و x_n نظیر می‌شود، بدست آورده‌ایم.(پیش بینی).

سپس انتگرال زیر را برای تصحیح بکار می‌بریم:

$$\int_{x_{n-1}}^{x_{n+1}} \left(\frac{dy}{dx}\right) dx = \int_{x_{n-1}}^{x_{n+1}} f(x,y)\,dx = \int_{x_{n-1}}^{x_{n+1}} P_2(x)\,dx,$$

$$y_{n+1,c} - y_{n-1} = \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}) - \frac{h^5}{90} y^{v}(\xi_2), \qquad x_{n-1} < \xi_2 < x_{n+1}. \tag{2}$$

1- Multi-Step Method
2- Milne
3- Predictor
4- Corrector
5- Coarse

این انتگرال را به روش سیمپسون حل کرده‌ایم.

با حذف جمله خطا در (1) مقدار تقریبی y_{n+1} (پیش بینی) را بدست می‌آوریم و سپس مقدار بهبود یافته y_{n+1} را از فرمول تصحیح (2) تعیین می‌کنیم.

(با خطای موضعی $(-\frac{1}{90} h^5 y^{(v)}(\xi_2))$

فرمول «پیش بینی» یک فرمول «صریح» می‌باشد و y_{n+1} را با داشتن y_{n-3} ، y_{n-2} ، y_{n-1} ، y_n ، بدست می‌آوریم. در صورتیکه فرمول «تصحیح» یک فرمول «غیر صریح» می‌باشد (y_{n+1} در سمت راست نیز ظاهر می‌شود).

فرمول «پیش بینی» میلن که می‌توان آنرا چنین نوشت:

$$y_{n+1} - y_{n-3} = \frac{4}{3} h\,(2f(x_{n-2}, y_{n-2}) - f(x_{n-1},y_{n-1}) + 2f(x_n,y_n)) \qquad (3)$$

و فرمول «تصحیح» سیمپسون

$$y_{n+1} - y_{n-1} = \frac{h}{3}(f(x_{n-1},y_{n-1}) + 4f(x_n,y_n) + f(x_{n+1},y_{n+1})) \qquad (4)$$

را توأماً بکار می‌بریم، باین ترتیب که ابتدا y_3, y_2, y_1 را با داشتن y_0 بدست می‌آوریم و به این منظور می‌توانیم از روش‌های رانج - کوتا نیز استفاده کنیم. سپس از فرمول «پیش‌بینی» (3)، y_4 را محاسبه کنیم و از فرمول (4) می‌توانیم مقدار تصحیح شده y_4 را محاسبه نمائیم. در مورد اخیر می‌توان به تصحیح y_4 با استفاده از روش تکراری ادامه داد و باین منظور معادله غیر خطی (4) را به فرم زیر می‌نویسیم:

$$y_{n+1}^{(r+1)} - y_{n-1} = \frac{h}{3}\left[f(x_{n+1}, y_{n+1}^{(r)}) + 4f(x_n, y_n) + f(x_{n-1}, y_{n-1})\right] \qquad (5)$$

و $y_{n+1}^{(1)}$ از $y_{n+1}^{(0)}$ و $y_{n+1}^{(2)}$ از $y_{n+1}^{(1)}$ و ... با انتخاب $r = 0, 1, 2, \ldots$ بدست می‌آیند.

برای روش پیش بینی - تصحیح از علائم زیر استفاده می‌کنیم:

P فرمول پیش بینی را بکار می‌بریم

E مقدار $f(x,y)$ را محاسبه می‌کنیم

C فرمول تصحیح را بکار می‌بریم

عملیات را تا رسیدن به نتیجه مطلوب ادامه می‌دهیم $P(EC)^k$

مثال ۱۱۸ معادله دیفرانسیل $y' = xy$ با مقدار اولیه $y(2)=1$ مفروض است، مطلوبست محاسبه $y(2.1), y(2.2), \ldots$

فرمول میلن - سیمپسون را بکار می‌بریم و $h=.1$ را در نظر می‌گیریم، با استفاده از روش رانج - کوتای مرتبه چهار خواهیم داشت:

$y(x_1) \cong y_1 = 1.22753$, $f_1 = 2.57781$

$y(x_2) \cong y_2 = 1.52196$, $f_2 = 3.34831$

$y(x_3) \cong y_3 = 1.90599$, $f_3 = 4.38378$

از (3) داریم:

$$P: y_4^{(0)} = y_0 + \frac{4}{3}h[2f_3 - f_2 + 2f_1]$$

$$y_4^{(0)} \cong 2.40998$$

$$E: f(x_4, y_4^{(0)}) = (2.4)(2.40998) = 5.78395$$

از (5) داریم:

$$C: y_4^{(1)} \cong y_2 + \frac{h}{3}[f(x_4, y_4^{(0)}) + 4f(x_3,y_3) + f(x_2,y_2)]$$

$$y_4^{(1)} \cong 1.52196 + \frac{.1}{3}[5.78395 + 4(4.38378) + 3.34831]$$

$$PEC: y_4^{(1)} = 2.4108727$$

$$E: f(x_4, y_4^{(1)}) = (2.4)(2.41087) = 5.78609$$

$$C: y_4^{(2)} = y_2 + \frac{h}{3}[f(x_4, y_4^{(1)}) + 4f(x_3,y_3) + f(x_2,y_2)]$$

$$y_4^{(2)} \cong 1.52196 + \frac{.1}{3}[5.78609 + 4(4.38378) + 3.34831]$$

$$P(EC)^2: y_4^{(2)} = 2.41094$$

$$y(x_4) \cong y_4^{(2)} \cong 2.41094$$

برای سایر جوابها عملیات را بطور مشابه ادامه می‌دهیم:

$$P: y_5^{(0)} = y_1 + \frac{4}{3}h(2f(x_4,y_4) - f(x_3,y_3) + 2f(x_2,y_2))$$

$$E: f(x_5, y_5^{(0)})$$

$$C: y_5^{(1)} = y_3 + \frac{h}{3}(f(x_5, y_5^{(0)}) + 4f(x_4,y_4) + f(x_3,y_3))$$

۱۰۸ روش پیش بینی - تصحیح آدامز (مولتن)[1]

برای حل معادله دیفرانسیل

$$y' = f(x,y) \qquad y(x_0) = y_0 \qquad (1)$$

با فرض $x_i = x_0 + ih$ خواهیم داشت:

$$\int_{x_i}^{x_i+1} y'(x)dx = \int_{x_i}^{x_i+1} f(x,y(x))dx \qquad (2)$$

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_i+1} f(x,y(x))dx \qquad (3)$$

1- Adams, (Moulton)

چنانچه انتگرال در نقاط x_{i-p}, ... , x_i, x_{i+1} برای pای مثبت مشخص باشد می‌توان مقدار تقریبی انتگرال را در (3) بدست آورد.

$$\int_{x_i}^{x_i+1} y(x)dx \approx A_{i+1}\, y(x_{i+1}) + A_i\, y(x_i) + \ldots + A_{i-p}\, y(x_{i-p})$$

$$y_{i+1} = y_i + A_{i+1}\, f(x_{i+1}, y_{i+1}) + A_i\, f(x_i, y_i) + \ldots + A_{i-p}\, f(x_{i-p}, y_{i-p}) \qquad (4)$$

عموماً معادله (4) نسبت به y_{i+1} غیر خطی است و معادله «تصحیح» نامیده می‌شود و با روش تکراری نتیجه مطلوب بدست می‌آید.

برای تعیین و «پیش بینی» y_{i+1} مقدار تقریبی انتگرال را با توجه به اینکه انتگرال در نقاط زیر x_{i-q}, ... , x_{i-1}, x_i برای q مثبت مشخص باشد، بدست می‌آوریم.

$$\int_{x_i}^{x_i+1} y(x)\, dx = B_i\, y(x_i) + B_{i-1}\, y(x_{i-1}) + \ldots + B_{i-q}\, y(x_{i-q}) \qquad (5)$$

$$y_{i+1} = y_i + B_i\, f(x_i, y_i) + B_{i-1}\, f(x_{i-1}, y_{i-1}) + \ldots + B_{i-q}\, f(x_{i-q}, y_{i-q})$$

رابطه (5) برای پیش بینی y_{i+1} می‌تواند مورد استفاده قرار گیرد.

با توجه به اینکه انتخاب p , q بطور دلخواه می‌باشد روابط (4) و(5) به صور مختلف بدست می‌آیند.

۱۱۸ الگوریتم آدامز - مولتن

معادله دیفرانسیل زیر داده شده است:

$$y' = f(x,y) \qquad y(x_0) = y_0$$

اگر y_1 , y_2, y_3را به روش رانج - کوتای مرتبه چهار بدست آوریم، می‌خواهیم مقدار انتگرال را بطور تقریبی در y_4 تعیین کنیم.

ابتداء یک چند جمله‌ای درجه سوم از چهار نقطه x_{i-3} , x_{i-2}, x_{i-1} , x_iگذرانده و انتگرال را از x_i تا x_{i+1} حساب می‌کنیم:

$$y_{i+1} - y_i = \int_{x_i}^{x_i+1} f(x)dx = \frac{h}{24}(-9f_{i-3} + 37f_{i-2} - 59f_{i-1} + 55f_i) + \frac{251h^5}{720} f^{(5)}(\xi)$$

سپس چند جمله‌ای درجه سوم دیگری از چهار نقطه x_{i-2} , x_{i-1} , x_i و x_{i+1} گذرانده و انتگرال را حساب می‌کنیم تا مقدار تصحیح بدست آید.

$$y_{i+1} - y_i = \int_{x_i}^{x_i+1} f(x)dx = \frac{h}{24}(f_{i-2} - 5f_{i-1} + 19f_i + 9f_{i+1}) - \frac{19h^5}{720} f^{(5)}(\xi) \qquad (7)$$

در نتیجه مقدار «پیش بینی» در x_{n+1} محاسبه می‌شود.

$$y_{n+1} = y_n + \frac{h}{24}[55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}] + O(h^5). \qquad (5.14)_{(8)}$$

و رابطه «تصحیح» کننده برابر است با:

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \cdot O(h^5) \qquad (9)$$

مثال ۱۲.۸ روش آدامز را بکار برده جواب معادله دیفرانسیل، $y' = x+y$، $y(0)=1$ را در $y(.4)$ ،$y(.5)$ بطور تقریبی تعیین کنید.

مقادیر $y(.1), y(.2), y(.3)$ را از روش رانج - کوتای مرتبه چهار بدست می‌آوریم و خواهیم داشت:

$y(0) = 1.0$

$y(.1) = 1.110342$

$y(.2) = 1.242806$

$y(.3) = 1.399718$

و برای آنکه روابط (8)، (9) را بکار بریم، برای $i = 3$ داریم:

$f(x_0,y_0) = 1.000000$

$f(x_1,y_1) = 1.210342$

$f(x_2,y_2) = 1.442806$

$f(x_3,y_3) = 1.699718$

$$y_4^{(0)} = y_3 + \frac{h}{24}(-9f_0 + 37f_1 - 59f_2 + 55f_3) \quad , \quad f_i = f(x_i,y_i)$$

$$y_4^{(0)} = 1.399718 + \frac{.1}{24}[-9 + 37(1.210342) - 59(1.442806) + 55(1.699718)] = 1.583641$$

مقدار پیش بینی شده y_4 می‌باشد

و از رابطه (9) آنرا تصحیح می‌کنیم.

$$y_4^{(1)} = y_3 + \frac{h}{24}(f_1 - 5f_2 + 19f_3 + 9\ f_4^{(0)})$$

$$f_4^{(0)} = f(x_4, y_4^{(0)}) = x_4 + y_4^{(0)} 4 + 1.583641 = 1.983641$$

پس خواهیم داشت:

$$y_4^{(1)} = 1.399718 + \frac{.1}{24}[1.210342 - 5(1.442806) + 19(1.699718) + 9(1.983641)] = 1.583650$$

بـرای دقت بـیشتر $y_4^{(2)}$ را مـی‌توان بـدست آورد. لیکـن بـا انـتخاب $y_4 = y_4^{(1)}$ مسـاله را ادامـه می‌دهیم:

$$f_4 = x_4 + y_4 = 1.983650$$

از فرمول پیش بینی (8) داریم:

$$y_5^{(0)} = y_4 + \frac{h}{24}(-9f_1 + 37f_2 - 59f_3 + 55f_4)$$

$$y_5^{(0)} = 1.583650 + \frac{.1}{24}[-9(1.210342) + 37(1.442806) - 59(1.699718) + 55(1.983650)] = 1.797434$$

از فرمول تصحیح (9) داریم:

$$y_5^{(0)} = y_4 + \frac{h}{24}(f_2 - 5f_3 + 19f_4 + 9y_4^{(0)})$$

$$f_4^{(0)} = x_5 + y_4^{(0)} = .5 + 1.797434 = 2.297434$$

$$y_5^{(1)} = 1.583650 + \frac{.1}{24}[1.442806 - (5)(1.699718) + (19)(1.983650) + 9(2.297434) + 1.7974444]$$

$$y(.4) = 1.583650 \quad , \quad y(.5) = 1.797444$$

این مثال نکات بسیار مهمی را در مورد روش آدامز نشان می‌دهد.

با فرض آنکه مقادیر قبلی تابع f ذخیره گردیده است در محاسبه $y_4^{(0)}$ تنها محاسبه یک مقدار $f(x,y)$ در رابطه «پیش بینی» لازم است. و در رابطه «تصحیح» نیز تنهایک بار تعیین مـقدار $f(x,y)$ در هـر تکرار لازم است. این در مقایسه با روش رانج - کوتای مرتبه چهار که برای هر جواب چهار بار باید تابع $f(x\ ,\ y)$ تعیین و محاسبه گردد، بسیار روش مطلوبتری می‌باشد. معمولاً در ایـن روش و در رابطه تکراری، در صورت انتخاب مناسب h، بیش از دو تکرار لزومی ندارد.

در اینصورت روش آدامز از نظر محاسباتی بسیار کاراتر از روش رانج - کوتای مرتبه چهار می‌باشد.

۱۲۸ روش رانج - کوتا - فیلبرگ[۱]

در روش رانج - کوتا برای آنکه دقت جواب مسأله را بررسی کنیم. بعد از محاسبه y_{n+1} مقدار h را نصف کرده و مجدداً جواب را در x_{n+1} محاسبه و y_{n+1} را بدست می‌آوریم. اگر دو مقدار بدست آمده نزدیک باشند جواب از دقت کافی برخوردار است. و برای انجام عملیات نیاز به $11 = 4+7$ بار محاسبه $f(x,\ y)$ می‌باشد.

در روش فیلبرگ از روش رانج - کوتای مرتبه چهار و مرتبه پنج تـواماً اسـتفاده مـی‌کنیم. و جواب را در $(y_{n+1}\ ,\ \dot{y}_{n+1})$ با استفاده از $(x_n\ ,\ y_n)$ بدست می‌آوریم در اینصورت اگر این دو جواب بدست آمده $y_{n+1}\ ,\ x_{n+1}$ بهم نزدیک باشند، جواب از دقت کافی برخوردار است. در این روش فقط به شش دفعه محاسبه $f(x,\ y)$ نیاز داریم که یکی از روش‌های متداول و مشهور برای حـل مـعادله

۱ - Runge - kutta - Fehlberg

دیفرانسیل مرتبه اول می‌باشد.

$$k_1 = h \cdot f(x_n, y_n),$$

$$k_2 = h \cdot f\left(x_n + \frac{h}{4}, y_n + \frac{k_1}{4}\right),$$

$$k_3 = h \cdot f\left(x_n + \frac{3h}{8}, y_n + \frac{3k_1}{32} + \frac{9k_2}{32}\right),$$

$$k_4 = h \cdot f\left(x_n + \frac{12h}{13}, y_n + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}\right),$$

$$k_5 = h \cdot f\left(x_n + h, y_n + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104}\right),$$

$$k_6 = h \cdot f\left(x_n + \frac{h}{2}, y_n - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40}\right);$$

$$\hat{y}_{n+1} = y_n + \left(\frac{25k_1}{216} + \frac{1408k_3}{2565} + \frac{2197k_4}{4104} - \frac{k_5}{5}\right), \text{ with global error } O(h^4),$$

$$y_{n+1} = y_n + \left(\frac{16k_1}{135} + \frac{6656k_3}{12825} + \frac{28561k_4}{56430} - \frac{9k_5}{50} + \frac{2k_6}{55}\right),$$

with global error $O(h^5)$;

$$, E \doteq \frac{k_1}{360} - \frac{128k_3}{4275} - \frac{2197k_4}{75240} + \frac{k_5}{50} + \frac{2k_6}{55}.$$

جدول ۴۸

مثال ۱۳۸ معادله دیفرانسیل مرتبه اول

$$\frac{dy}{dx} = -2x - y$$
$$y(0) = -1 \quad , \quad h = 0.1$$

را با استفاده از روش رانج - کوتا - فیلبرگ حل کنید.

$$k_1 = 0.1,$$
$$k_2 = 0.0925000,$$
$$k_3 = 0.0889609,$$
$$k_4 = 0.0735157,$$
$$k_5 = 0.0713736,$$
$$k_6 = 0.0853872,$$

$\hat{y}_1 = -0.914512212,$ $y_1 = -0.914512251,$ خطا $E = -0.000000040.$

جواب دقیق برابر $y(0.1) = -0.914512254$ می‌باشد و جواب بالا تا 8 رقم با معنی صحیح بدست آمده است.

۱۳۸ حل دستگاه معادلات دیفرانسیل

می‌توانیم از بسط روش رانج - کوتای مرتبه چهارم برای حل دستگاه معادلات زیر استفاده نمائیم:

$$\begin{cases} y'_1 = f_1(x,y_2,y_2) & y_1(x_0) = y_{10} \\ y'_2 = f_2(x,y_1,y_2) & y_2(x_0) = y_{20} \end{cases} \qquad x \in [x_0 , b] \qquad (1)$$

با در نظر گرفتن $x_i = x_0 + ih$ و y_{1i} ، y_{2i} مقدار تقریبی $y_1(x_i)$ $y_2(x_i)$ خواهیم داشت:

$$y_{1i+1} = y_{1i} + h\,\phi_1\,(x_i, y_{1i}, y_{2i}) \qquad (2)$$

$$y_{2i+1} = y_{2i} + h\,\phi_2\,(x_i, y_{1i}, y_{2i}) \qquad (3)$$

بطوریکه:

$$\begin{cases} \phi_1\,(x_i,y_{1i},y_{2,i},h) = \frac{1}{6}\,(k_{11} + 2k_{12} + 2k_{13} + k_{14}) \\ \phi_2\,(x_i,y_{1i},y_{2i},h) = \frac{1}{6}\,(k_{21} + 2k_{22} + 2k_{23} + k_{24}) \end{cases}$$

$$\begin{cases} k_{11} = f_1(x_i,y_{1i},y_{2i}) \\ k_{21} = f_2(x_i,y_{1i},y_{2i}) \end{cases}$$

$$\begin{cases} k_{12} = f_1\,(x_i + \frac{h}{2} , y_{1i} + \frac{h}{2}\,k_{11} , y_{2i} + \frac{h}{2}\,k_{21}) \\ k_{22} = f_2\,(x_i + \frac{h}{2} , y_{1i} + \frac{h}{2}\,k_{11} , y_{2i} + \frac{h}{2}\,k_{21}) \end{cases}$$

$$\begin{cases} k_{13} = f_1\,(x_i + \frac{h}{2} , y_{1i} + \frac{h}{2}\,k_{12} , y_{2i} + \frac{h}{2}\,k_{22}) \\ k_{23} = f_2\,(x_i + \frac{h}{2} , y_{1i} + \frac{h}{2}\,k_{12} , y_{2i} + \frac{h}{2}\,k_{22}) \end{cases}$$

$$\begin{cases} k_{14} = f_1\,(x_i + h , y_{1i} + hk_{13} , y_{2i} + hk_{23}) \\ k_{24} = f_2\,(x_i + h , y_{1i} + hk_{13} , y_{2i} + hk_{23}) \end{cases}$$

مثال ۱۴۸ جواب تقریبی دستگاه معادلات زیر را در $x=.1$ بدست آورید.

$$\begin{cases} y'_1 = y_1 + y_2 & y_1(0) = 0 \\ y'_2 = -y_1 + y_2 & y_2(0) = 1 \end{cases}$$

با انتخاب $h=.1$ و کاربرد روش رانج - کوتای مرتبه چهار خواهیم داشت:

شرایط اولیه $i = 0 \;,\; x_0 = 0 \;,\; y_{10} = 0 \;,\; y_{20} = 1$

$$\begin{cases} k_{11} = f_1(x_0,y_{10},y_{20}) = 1 \end{cases}$$

$$\begin{cases} k_{21} = f_2(x_0, y_{10}, y_{20}) = 1 \end{cases}$$

$$\begin{cases} k_{12} = f_1\left(x_0 + \frac{h}{2}, y_{10} + \frac{h}{2}k_{11}, y_{20} + \frac{h}{2}k_{21}\right) = 1.1 \\ k_{22} = f_2\left(x_0 + \frac{h}{2}, y_{10} + \frac{h}{2}k_{11}, y_{20} + \frac{h}{2}k_{21}\right) = 1.0 \end{cases}$$

$$\begin{cases} k_{13} = f_1\left(x_0 + \frac{h}{2}, y_{10} + \frac{h}{2}k_{12}, y_{20} + \frac{h}{2}k_{22}\right) = 1.105 \\ k_{23} = f_2\left(x_0 + \frac{h}{2}, y_{10} + \frac{h}{2}k_{12}, y_{20} + \frac{h}{2}k_{22}\right) = .995 \end{cases}$$

$$\begin{cases} k_{14} = f_1(x_0 + h, y_{10} + hk_{13}, y_{20} + hk_{23}) = 1.2100 \\ k_{24} = f_2(x_0 + h, y_{10} + hk_{13}, y_{20} + hk_{23}) = .9890 \end{cases}$$

$$\begin{cases} \phi_1(x_0, y_{10}, y_{20}, h) = 1.103333 \\ \phi_2(x_0, y_{10}, y_{20}, h) = .996500 \end{cases}$$

$$\begin{cases} y_{11} = .110333 \\ y_{21} = 1.099650 \end{cases}$$

۱۴۸ روش کول - نیومروف[1]

در معادلاتی به شکل $y''=f(x,y)$ که شامل مشتق اول نیستند، برای حل به روش مخصوصی نیازمندیم. در این مورد چندین ریاضیدان بطور مستقل تحقیق کرده و به نتیجه مطلوب رسیده‌اند. برای حل از فرمول اپراتور زیر استفاده می‌نمائیم.

$$\frac{\delta^2}{U^2} = 1 + \frac{\delta^2}{12} - \frac{\delta^4}{240} + \ldots \qquad \begin{cases} U = \delta - \frac{\delta^3}{24} + \frac{3\delta^5}{640} - \frac{5\delta^7}{7164} + \cdots \\ U^2 = \delta^2 - \frac{\delta^4}{12} + \frac{\delta^6}{90} - \frac{\delta^8}{560} + \frac{\delta^{10}}{3150} - \cdots \end{cases}$$

از اینرو داریم:

$$\delta^2 y_n = \left(1 + \frac{\delta^2}{12} - \frac{\delta^4}{240} + \ldots\right) h^2 y''_n = h^2\left(1 + \frac{\delta^2}{12} - \frac{\delta^4}{240} + \ldots\right) f_n$$

یا: $$y_{n+1} - 2y_n + y_{n-1} = \frac{h^2}{12}(f_{n+1} + 10f_n - f_{n-1}) - \frac{h^6}{240} y_n^{(6)} + O(h^8)$$

1- Cool - Nevmorof Method

با توجه به $\delta^4 \approx U^4 = h^4 D^4$, $= f_n$. y_n''

با صرفنظر کردن از جملات مرتبه h^6 و بالاتر، فرمول زیر را خواهیم داشت:

$$y_{n+1} - 2y_n + y_{n-1} = \frac{h^2}{12}(f_{n+1} + 10f_n + f_{n-1})$$

با خطای موضعی $O(h^6)$ در حالت کلی، فرمول غیر صریح است، اما در حالت خاص وقتی که $f(x,y)=y.g(x)$ داریم:

می‌توانیم فرمول صریح زیر را بدست آوریم:

$$y_{n+1} = \frac{\beta_n y_n - \alpha_{n-1} y_{n-1}}{\alpha_{n+1}} \quad , \quad \left[\alpha = 1 - \left[\frac{h^2}{12}\right] g \quad , \quad \beta = 2 + \frac{5h^2}{6} g\right]$$

۱۵۸ برنامه‌های کامپیوتری

برنامه ۱۸ لیست برنامه کامپیوتری آدامز - مولتن به زبان *BASIC*

این برنامه به نام *ADAMS.BAS* برای حل معادله دیفرانسیل مرتبه اول از روش آدامز می‌باشد. از این برنامه برای حل معادله دیفرانسیل $y' = -2x - y$ ، $y(0) = -1$ استفاده شده است.

```
' ************************************************
'          PROGRAM   ADAMS.BAS
' ************************************************
'        APPLIED  NUMERICAL   ANALYSIS
'
' This program gets the solution to a first-order
'   differential equation by the Adam's method.
' It can use either three or four past points to compute
'   the next point. Starting values (that have been generated
'   by some single-step method) are read from a file named
'   ADAMS.DTA.  The first item in that file is the number of
'   past points that are to be used. The items that follow
'   are the starting data pairs, x, f(x). Each item is
'   delimited with a comma (or end of line).
' This solves dy/dx = -2x - y,  y(0) = -1
' Variables used are:
'      x()     array of x-values
'      y()     array of y-values
'      f()     array of f(x)-values,  f(x) = dy/dx
'      i,j     loop control indices
'       N      number of past point to be used
'  response$   user request to repeat computations
'     delx     the spacing between x-values
' Subprogram declarations
DECLARE  SUB sub3 (x!(), y!(), f!(), delx)  'Computes from 3 points
DECLARE  SUB sub4 (x!(), y!(), f!(), delx)  'Computes from 4 points
DECLARE  FUNCTION fcn! (x!, y!)             'Computes f(x)
' Open file and read in the data
OPEN "ADAMS.DTA"  FOR INPUT AS #1
INPUT #1, N
' See if N = 3 or 4
IF N < 3 OR N > 4 THEN
  PRINT "Invalid value for N in file"
  STOP
END IF

' Get the past x- and y-values
FOR i = 1 TO N
  INPUT #1, x(i), y(i)
NEXT i
```

```
' Find delx
delx = x(N) - x(N - 1)

' Compute dy/dx = f(x)-values for past data
FOR i = 1 TO N
  f(i) = fcn(x(i), y(i))
NEXT i

' Print a heading and the initial data

CLS
PRINT "  x  "; "     y    "; "    dy/dx"
PRINT
FOR i = 1 TO N
  PRINT USING "##.###    "; x(i);
  PRINT USING "###.#######        "; y(i); f(i)
NEXT i
PRINT
' While "response$ = Y" do a set of five iterations

response$ = "Y"

DO WHILE response$ = "Y"

  FOR i = 1 TO 5
    IF N = 3 THEN
      CALL sub3(x(), y(), f(), delx)
    ELSE
      CALL sub4(x(), y(), f(), delx)
    END IF

' Compute f-value at new point
    f(N + 1) = fcn(x(N + 1), y(N + 1))

' Print the new values
    PRINT USING "##.###    "; x(N + 1);
    PRINT USING "###.#######        "; y(i); f(i)

' Reset the arrays for next computation
    FOR j = 1 TO N
      x(j) = x(j + 1)
      y(j) = y(j + 1)
      f(i) = f(i + 1)
    NEXT j
  NEXT i

' See if user wants more computations
   INPUT "Enter Y if you want another five computations  ", response$
   IF response$ = "y" THEN response$ = "Y"

LOOP  ' End of WHILE loop
```

```
FUNCTION  fcn (x, y)
' This computes  a new value for dy/dx

  fcn = -2 * x - y
END  FUNCTION

SUB  sub3 (x(), y(), f(), delx)
'  This computes  a new value for y from  3 past values
  y(4) = y(3) + delx / 12 * (23 * f(3) - 16 * f(2) + 5 * f(1))
  x(4) = x(3) + delx
END  SUB

SUB  sub4 (x(), y(), f(), delx)
' This computes  a new y-value from  4 past values
  y(5) = y(4) + delx / 24 * (55 * f(4) - 59 * f(3) + 37 * f(2) - 9 *
f(1))
  x(5) = x(4) + delx
END  SUB
**************************************************************
     OUTPUT  FOR  ADAMS.BAS

   x         y          dy/dx

 0.000  -1.000000    1.000000
 0.200  -0.856192    0.456192
 0.400  -0.810960    0.010960
 0.600  -1.000000    1.000000
 0.800  -0.810960    0.456192
 1.000  -0.924510    0.010960
 1.200  -1.277456   -1.122544
 1.400   0.000000    0.000000
Enter  Y  if you  want  another  five  computations  y
 1.600  -0.885216    0.456192
 1.800  -1.669697    0.010960
 2.000  -2.491281   -1.114784
 2.200  -2.851074   -1.548927
 2.400   0.000000    0.000000
Enter  Y  if you  want  another  five  computations
```

برنامه ۲.۸ لیست برنامه روش رانج - کوتا- فیلبرگ به زبان *PASCAL*

لیست ریز برنامه کامپیوتری بنام *RKF.PAS* برای حل معادله دیفرانسیل مرتبه اول می‌باشد.

```
(*
                    RUNGE-KUTTA-FEHLBERG      method

                  ordinaray differential  equations

                     APPLIED  NUMERICAL   ANALYSIS
-------------------------------------------------------------------------
*)
PROCEDURE   rkf(t0,  h : REAL;
          VAR  x0, xend : vector);
VAR
  workarray      (* a twodimensional  array to store the k values *)
          : matrix;
  i,row,col,
  count       : INTEGER;
  tEND,  error     (* tEND  is t[i+1]  =  t[i]  +  hSTART   *)
          : REAL;
(*
-------------------------------------------------------------------
   This function  returns  the  Larger  of two real numbers.
*)
FUNCTION   TheLargerOf(a,b   : REAL): REAL;
BEGIN
  IF a > b THEN  TheLargerOf  := a
  ELSE  TheLargerOf  := b
END;                    (* TheLargerOf  *)
(*
-------------------------------------------------------------------
   This procedure  evaluates  the  new state  vector  and  updates
   t0 for the next time step.
*)
(*
    This procedure  computes  the  estimated  error from  the  new
    ki's (workarray)  in getting  the  next state  vector.
*)
PROCEDURE   computeERROR;
VAR
  temp  : REAL;
BEGIN
  error  := 0.0;
  FOR  i := 1 TO   numberOFequations   DO  BEGIN
    temp  := ABS(workarray[1,i]/360.0  - 128.0*workarray[3,i]/4275.0
             -2197.0*workarray[4,i]/75240.0
             +workarray[5,i]/50.0  +  2.0*workarray[6,i]/55.0);
    error  := TheLargerOf(error,temp)     END
END;                                    (* ComputeError   *)
```

```
(*
   This procedure updates the state vector from time[i] to time[i+1].
*)
PROCEDURE   ComputeNewX;
VAR
  i : INTEGER;
BEGIN
  FOR i := 1 TO numberOFequations   DO
    xend[i] := x0[i] + 25.0*workarray[1,i]/216.0
               +1408.0*workarray[3,i]/2565.0
               +2197.0*workarray[4,i]/4104.0
               -workarray[5,i]/5.0;
END;                                  (* ComputeNewX   *)

BEGIN      (* rkf *)
  tEND  := t0 + hSTART;
  count := 0;
  WHILE  (t0 < tEND)  AND  (count < MaxCount)  DO BEGIN
    count := count + 1;
(*
-------------------------------------------------------------------
   Get first estimate of the delta X's
*)
  getDERIVS(x0,f,t0);
  FOR i := 1 TO numberOFequations   DO        BEGIN
    workarray[1,i] := h*f[i];
    xend[i] := x0[i] + workarray[1,i]/4.0  END;
(*
-------------------------------------------------------------------
   Get second estimate. The xend vector holds the x values.
*)
  getDERIVS(xend,f,t0+h/4.0);
  FOR i := 1 TO numberOFequations   DO        BEGIN
    workarray[2,i] := h*f[i];
    xend[i] := x0[i] + (workarray[1,i]*3.0 +
               workarray[2,i]*9.0)/32.0    END;
(*
-------------------------------------------------------------------
   Repeat for third estimate
*)
   getDERIVS(xend,f,    t0+3.0*h/8.0);
   FOR i := 1 TO numberOFequations   DO        BEGIN
     workarray[3,i] := h*f[i];
     xend[i] := x0[i] + (workarray[1,i]*1932.0 -
                workarray[2,i]*7200 +
                workarray[3,i]*7296)/2197.0 END;
(*
-------------------------------------------------------------------
   Fourth estimates, i.e. k4's
*)
  getDERIVS(xend,f,    t0+12.0*h/13.0);
  FOR i := 1 TO numberOFequations   DO        BEGIN
```

```
      workarray[4,i] := h*f[i];
      xend[i] := x0[i] + 439.0*workarray[1,i]/216.0
                 - 8.0*workarray[2,i]
                 + 3680.0*workarray[3,i]/513.0
                 -845.0*workarray[4,i]/4104.0 END;
(*
-------------------------------------------------------------------
   Fifth estimates
*)
   getDERIVS(xend,f,t0+h);
   FOR i := 1 TO numberOFequations   DO          BEGIN
      workarray[5,i] := h*f[i];
      xend[i] := x0[i] - 8.0*workarray[1,i]/27.0
                 + 2.0*workarray[2,i]
                 -3544.0*workarray[3,i]/2565.0
                 +1859.0*workarray[4,i]/4104.0
                 -11.0*workarray[5,i]/40.0    END;
(*
-------------------------------------------------------------------
   Sixth estimates
*)
   getDERIVS(xend,f,t0+h/2.0);
   FOR i := 1 TO numberOFequations   do  workarray[6,i] := h*f[i];
(*
-------------------------------------------------------------------
   Compute  error estimate
*)
    error := 0.0;
    computeERROR;
   IF (error < tol) THEN                    BEGIN
      t0 := t0 + h;
      ComputeNewX;
      FOR i := 1 TO numberOFequations   DO
        x0[i] := xend[i]          END;

   IF (error>tol)  THEN    BEGIN
      WriteLn(outfile);
      WriteLn(outfile,'':20,   'FOR A STEPSIZE  OF ', h:6:4,
                    ' THE ERROR  ESTIMATE  IS : ',error:8:6);
      WriteLn(outfile);
      h := h/2.0          END;
   IF (error < h*tol/10.0) THEN   h := 2.0*h;

   IF (t0 + h > tEND)  THEN  h := tEND - t0
                                          END; (* while *)

END;
                                    (* rkf procedure  *)
```

برنامه ۳۸روش رانج - کوتا- فیلبرگ

لیست زیر برنامه کامپیوتری برای حل یک دستگاه معادلات دیفرانسیل مرتبه اول از روش رانج - کوتا - فیلبرگ بنام *RKSYST.C* به زبان *C* آورده شده است.

```
/*
                ********************************
                **                            **
                **          RKFSYST.C         **
                **                            **
                ********************************

                        Chapter 5

            APPLIED NUMERICAL ANALYSIS (Sixth edition)
                        1999
-----------------------------------------------------------------

     This program solves a system of N first order differential
     equations by the RUNGE-KUTTA-FEHLBERG method. The equations
     are of the form:
         DX1/DT = F1(T,X)
         DX2/DT = F2(T,X), ETC.

     The particular problem solved here is the second order
     differential equation:

         Y'' = 2 - 4Y*Y/SQR(SIN(x)) ,   1 <= x <= 2.
         where Y(1) = sin(1)*sin(1)
               Y'(1) = 2*sin(1)*cos(1)

         The Analytical solution to this equation is:
               Y(x) = sin(x)*sin(x).

     The computed and the analytical solution is printed out at
     each time step.

*/
#include <stdio.h>
#include <math.h>

#define  maxN       11   /* max number of first order equations */
#define  hSTART     0.1
#define  tol        0.00005
#define  MaxCount   10
#define  tINIT      1.0       /* initial time */
#define  tFINAL     2.0       /* final time at last interval */

float x0[maxN],       /* state vector at beginning of interval */
      xend[maxN],     /* state vector at end of interval */
      f[maxN],        /* array that holds the values of the derivatives  */
      h,        /* adjustable step size */
      t0,      /* initial time value at beginning of interval */
      finalTime;     /* value equal to: tFINAL - hSTART/2.0 */

float workarray[7][maxN],     /* a two-dimensional array to store the k values */
      tEND, error;            /* tEND is t[i+1] = t[i] + hSTART */

int   i_rkf, row, col, count_rkf;
int   numberOFequations,      /* actual number of differential equations */
      i, count;

/*
-----------------------------------------------------------------
   This Procedure sets up all the initial values.
*/
```

```
initialize()
 {
   numberOFequations = 2;

   x0[1] = sin(1.0)*sin(1.0);
   x0[2] = 2.0*sin(1.0)*cos(1.0);

   t0 = tINIT;
   h = hSTART;
 }          /* end of initialize */

getDERIVS(x0, f, t0)
  float x0[], f[], t0;

/*
-------------------------------------------------------
This procedure evaluates the first-order equations for
the given time and state values.

   HERE:  dx1/dt = x2;  dx2/dt := 2 - 4*x1*x1/(sin(t))**2
*/
 {
   f[1] = x0[2];
   f[2] = 2.0 - 4.0*(x0[1]*x0[1])/(sin(t0)*sin(t0));
 }            /* end of getDERIVS */

/*
   ---------------------------------------------------------------
    This function returns the larger of two real numbers.
*/

float TheLargerOf(a, b)
float a, b;
 {
   if (a > b)
     return(a);
   else
     return(b);
 }                  /* end of TheLargerOf */

/*
-----------------------------------------------------------------------
     This procedure computes the estimated error from the new
     ki's (workarray) in getting the next state vector.
*/

computeERROR()
 {
   float temp;

   error = 0.0;
   for ( i = 1; i <= numberOFequations; ++i )
    {temp = fabs(workarray[1][i]/360.0
                      - 128.0*workarray[3][i]/4275.0
                      - 2197.0*workarray[4][i]/75240.0
                      + workarray[5][i]/50.0 +
                      + 2.0*workarray[6][i]/55.0);
     error = TheLargerOf(error,temp);
    }                 /* end of the for i loop */
 }                        /* end of computeERROR */

/*
-----------------------------------------------------------------------
 This procedure updates the state vector from time[i] to time[i+1].
*/
ComputeNewX()
 {
```

```
   for ( i = 1; i <= numberOFequations; ++i )
      xend[i] = x0[i] + 25.0*workarray[1][i]/216.0
                +1408.0*workarray[3][i]/2565.0
                +2197.0*workarray[4][i]/4104.0
                -workarray[5][i]/5.0;
}                              /* end of ComputeNewX */

/*
---------------------------------------------------------------------
     This Procedure solves a system of differential equations using
     the 4th/5th order method of Runge-Kutta-Fehlberg from t[i] to
     t[i+1]. Use is made of the error estimate to adjust the stepsize,
     h-value, for the next subinterval.
*/
       rkf(t0, h, x0, xend)
       float t0, h, x0[], xend[];
{
  tEND = t0 + hSTART;
  count_rkf = 0;
  while ((t0 < tEND) && (count_rkf < MaxCount))
   {
     count_rkf = count_rkf + 1;
/*
---------------------------------------------------------------------
    Get first estimate of the delta x's
*/
   getDERIVS(x0,f,t0);
   for ( i_rkf = 1; i_rkf <= numberOFequations; ++i_rkf )
    {
      workarray[1][i_rkf] = h*f[i_rkf];
      xend[i_rkf] = x0[i_rkf] + workarray[1][i_rkf]/4.0;
    }
/*
---------------------------------------------------------------------
    Get second estimate. The xend vector holds the x values.
*/
   getDERIVS(xend,f,t0+h/4.0);
   for ( i_rkf = 1; i_rkf <= numberOFequations; ++i_rkf )
    {
      workarray[2][i_rkf] = h*f[i_rkf];
      xend[i_rkf] = x0[i_rkf] + (workarray[1][i_rkf]*3.0
                    + workarray[2][i_rkf]*9.0)/32.0;
    }
/*
---------------------------------------------------------------------
*/  Repeat for third estimate
    getDERIVS(xend,f, t0+3.0*h/8.0);
    for ( i_rkf = 1; i_rkf <= numberOFequations; ++i_rkf )
     {
       workarray[3][i_rkf] = h*f[i_rkf];
       xend[i_rkf] = x0[i_rkf] + (workarray[1][i_rkf]*1932.0
                     - workarray[2][i_rkf]*7200
                     + workarray[3][i_rkf]*7296)/2197.0;
     }
/*
---------------------------------------------------------------------
    Fourth estimates, i.e. k4's
*/
   getDERIVS(xend,f, t0+12.0*h/13.0);
   for ( i_rkf = 1; i_rkf <= numberOFequations; ++i_rkf )
    {
      workarray[4][i_rkf] = h*f[i_rkf];
      xend[i_rkf] = x0[i_rkf] + 439.0*workarray[1][i_rkf]/216.0
                    - 8.0*workarray[2][i_rkf]
                    + 3680.0*workarray[3][i_rkf]/513.0
                    -845.0*workarray[4][i_rkf]/4104.0;
    }
/*
```

```
    Fifth estimates
*/
   getDERIVS(xend,f,t0+h);
   for ( i_rkf = 1; i_rkf <= numberOFequations; ++i_rkf )
    {
      workarray[5][i_rkf] = h*f[i_rkf];
      xend[i_rkf] = x0[i_rkf]  - 8.0*workarray[1][i_rkf]/27.0
                  + 2.0*workarray[2][i_rkf]
                  -3544.0*workarray[3][i_rkf]/2565.0
                  +1859.0*workarray[4][i_rkf]/4104.0
                  -11.0*workarray[5][i_rkf]/40.0;
    }
/*
---------------------------------------------------------------------
    Sixth estimates
*/
   getDERIVS(xend,f,t0+h/2.0);
   for ( i_rkf = 1; i_rkf <= numberOFequations; ++i_rkf )
     workarray[6][i_rkf] = h*f[i_rkf];
/*
---------------------------------------------------------------------
    Compute error estimate
*/
    error = 0.0;
    computeERROR();
   if (error < tol)
    {
      t0 = t0 + h;
      ComputeNewX();
      for ( i_rkf = 1; i_rkf <= numberOFequations; ++i_rkf )
         x0[i_rkf] = xend[i_rkf];
    }
   if (error > tol)
    {
      printf("\n\n");
      printf("\t\t FOR A STEPSIZE OF%.4f\t",h);
        printf(" THE ERROR ESTIMATE IS  %.6f", h,error);
      printf("\n");
      h = h/2.0;
    }
   if (error < h*tol/10.0)
      h = 2.0*h;
      if (t0 + h > tEND)
         h = tEND - t0;
      }       /* end of while loop */
    }         /* end of rkf procedure */

   main()             /* begin of main */
   {
      initialize();
      count = 0;
      printf("\n\t AT TIME T =  %.2f\n", t0);
      for ( i = 1; i <= numberOFequations; ++i )
        printf("\t\t   X[ %d ] =  %.6f\n", i, x0[i]);
      printf("\n");

      finalTime = tFINAL - hSTART/2.0;
      while ((t0 < finalTime) && (count < MaxCount))
       {
         rkf(t0,h, x0,xend);
         t0 = t0 + hSTART;
         printf("\n\t AT TIME T =  %.2f\n ", t0);
         for ( i = 1; i <= numberOFequations; ++i )
            printf("\t\t   X[ %d ] =  %.6f\n", i, xend[i]);
         printf("\t\t\t   EXACT IS :  %.6f   FOR X[1]\n\n ",
   (sin(t0)*sin(t0)));
       }              /* end of while loop */
   }                  /* end of the rkfsyst.c program */

   ***********************************************************************
```

تمرینات فصل هشتم

۱- جواب تقریبی معادله دیفرانسیل $y' = y + x^2$ را با روش اویلر در نقاط $x = .2$ و $x = .4$ پیدا کنید، در صورتیکه در $x = 0$ ، $y = 1$ یکبار $h = .2$ و بار دیگر $h = .4$ انتخاب شود.

۲- روش رانج - کوتای مرتبه 2 را بکار برده جواب معادله دیفرانسیل $y' = y + x^2$ را در نقطه $x = .2$ بدست آورید.

۳- روش رانج - کوتای مرتبه 4 را بکار برده و جواب معادله دیفرانسیل فوق را در نقطه $x = .2$ پیدا نمائید.

۴- فرمول پیش بینی - تصحیح زیر را برای حالت $PECE$ بکار برده، جواب معادله $y' = y + x^2$ را در $x = .2, x = .4$ محاسبه کنید.

۵- جواب دقیق معادله دیفرانسیل $y' = y + x^2$ را بدست آورید و دقت محاسبات را در مسایل فوق تعیین نمائید. $(x = .2 , x = .4)$

۶- با یک برنامه کامپیوتری برای آلگوریتم اویلر، جواب معادله $y' = x^2 + y^2$ را با فرض $x_0 = 1 , y_0 = 0 , h=0.1$ در 20 نقطه متوالی محاسبه کنید.

جواب: $y(2) = 6.15633$ و $h=0.1$ که به روش اویلر خطا زیاد می‌باشد.

جواب دقیق $y(2)=6.88125$

۷- معادله دیفرانسیل $y'' = y^2 + x^2 , y(1) = 1 , y'(1) = 2$ را بصورت یک دستگاه معادلات دیفرانسیل مرتبه اول بنویسید و جواب دستگاه را با فرض $h = .02$ در $x = 1.04$ بدست آورید.

۸- برای معادله دیفرانسیل $\frac{dy}{dx}=x^3+y^2$, $y(0)=0$ با انتخاب $h=0.2$ جواب معادله را در نقاط $x=0.6$ و $x = 0.4$ و $x=0.2$ به روش رانج- کوتای به دست آورید. سپس با روش میلن- سیمپسون جواب معادله را در $x = 1.0$ و $x=0.8$ محاسبه کنید.

جواب:

x	0	0.2	0.4	0.6	0.8	1.0
y	0	0.00040	0.00060	0.03247	0.10343	0.25757

۹- جواب معادله دیفرانسیل $\frac{dy}{dx}=xy^2$ ، $y(1)=1$ را در $x=1.2$ و $x = 1.1$ با روش اویلر و روش رانج-کوتای مرتبه دوم به دست آورید و با جواب دقیق مسأله $y=\frac{2}{3-x^2}$ نتیجه را مقایسه کنید.

جواب:

$$x=1.1 \qquad x=1.2$$

و

$$y=1.17 \qquad y=1.282$$

۱۰- معادله دیفرانسیل مرتبه دوم زیر مفروض است:

$$\frac{d^2y}{dx^2} + \frac{1}{x}\frac{dy}{dx} + (1 - \frac{1}{4x^2})\, y = 0$$

با توجه به اینکه می‌دانیم $x\in [\frac{\pi}{2}, \pi]$, $y(\frac{\pi}{2}) = 1$, $y'(\frac{\pi}{2}) = -\frac{1}{\pi}$

با انتخاب $h = .1$ جواب‌های معادله را در نقاط واقع بر فاصله $[\frac{\pi}{2}, \pi]$ بدست آورید. برای حل این مساله می‌توان برنامه‌ای بزبان C نوشت. جواب‌های تقریبی را با جواب دقیق معادله $y(x) = (\frac{\pi}{2x})^{1/2} Sin(x)$ در نقاط مورد نظر مقایسه نمائید.

۱۱- با توجه به مثال بخش آلگوریتم آدامز، مقادیر $y(.6)$, $y(.7)$ را در معادله دیفرانسیل زیر $y' = x + y$, $y(0) = 1$ بدست آورید و نتیجه را با جواب دقیق معادله $y = 2e^x - x - 1$ مقایسه نمائید.

۱۲- جواب معادله دیفرانسیل $y' = xy^{1/3}$ را با شرط اولیه $y(1) = 1$ در نقاط $x = 1(.5)5$ به روش تیلور بدست آورید.

جواب:

$x_1 = 1.5$, $y_1 = 1.68618$, $x_2 = 2.0$, $y_2 = 2.82846,...$

۱۳- جواب معادله دیفرانسیل $y' = xy^{1/3}$ را با شرط اولیه $y(1) = 1$ و با انتخاب $h = .1$ در نقاط $x = 1(.1)2$ به روش رانج - کوتای مرتبه 2 بدست آورید.

جواب: $x_1 = 1.1$, $y_1 = 1.10682$,..., $x_{10} = 2.0$, $y_{10} = 2.82846$

۱۴- دستگاه معادلات دیفرانسیل زیر را با شرایط اولیه $y(0) = 0$, $z(0) = 1$ به روش رانج - کوتای مرتبه 2 حل نموده و جواب دستگاه را برای $y(½)$, $z(½)$ بدست آورید.

$$\begin{cases} xy' + z' + ½y = 0 \\ xz' - y' + ½z = 0 \end{cases}$$

جواب: $y(½) = .21729$, $z(½) = .92044$

۱۵- معادله دیفرانسیل $y' = -\frac{2x + y}{2y - x}$ را باشرط اولیه $y(1) = 0$ در نظر می‌گیریم. جواب معادله دیفرانسیل را با یکی از روش‌های عددی در نقاط $x = 1(.2)2$ محاسبه و با جواب‌های تحقیقی که از جواب تحلیلی زیر

$Ln\ (x^2+y^2) = arctg\ \frac{y}{x}$ بدست می‌آید مقایسه کنید.

۱۶- معادله دیفرانسیل $y' = -\frac{2xy + e^y}{x^2 + xe^y}$ را با شرط اولیهٔ $y(1) = 0$ در نظر می‌گیریم. جواب معادله را با یکی از روش‌های عددی در نقاط $x=1(.2)2$ محاسبه و با جواب های تحقیقی که از جواب تحلیلی $x^2y + xe^y = 1$ بدست می‌آید مقایسه کنید.

۱۷- جواب معادله دیفرانسیل $(1 - x^2)y'' - 2xy' + 20y = 0$ را با بکار بردن روش رانج - کوتای مرتبه 2 در نقاط $x = 0(.25)1$ بدست آورید و با جواب تحقیقی که از جواب تحلیلی زیر $y = \frac{1}{8}(35x^4 - 30x^2 + 3)$ بدست می‌آید، مقایسه کنید. شرایط اولیه عبارتند از: $y(0) = .375\ ,\quad y'(0) = 0$

۱۸- جواب معادله دیفرانسیل $y' = -xy^2$ با شرط اولیه $y(0) = 2$ با استفاده از روش رانج - کوتای مرتبه 2 در نقاط $x = 0(.2).6$ محاسبه شده است.

$y(0)=2,\ y(.2) = y_1 \approx 1.92308\ ,\ y(.4)=y_2 \approx 1.72414,$

$y(.6) = y_3 \approx 1.47059$

جواب معادله دیفرانسیل را در نقطه $x = .8$ با استفاده از روش *Milne - Simpson* بدست آورید.

جواب:

روش پیش بینی $P : y_{n+1} = y_{n-3} + \frac{4h}{3}(2f_{n-2} - f_{n-1} + 2f_n)$

روش تصحیح $C : y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n-1} + 4f_n + f_{n+1})$

$f_0 = f(x_0,y_0)=0\ \ f_1=f(x_1,y_1)= -.73964\ \ f_2=f(x_2,y_2)= -1.18906$

$f_3 = f(x_3,y_3)= -1.29758$

$P:\quad y_4 = y_0 + \frac{4}{3}h(2f_1 - f_2 + 2f_3) = 1.23056$

$E:\quad f_4 = f(x_4,y_4) = -x_4 y^2_4 = -1.21142$

$C:\quad y_4^{(k+1)} = y_2 + \frac{h}{3}(f(x_2,y_2)) + 4f(x_3,y_3) + f(x_4, y_4^{(k)})$

$y_4^{(1)} = 1.72414 + \frac{.2}{3}(-1.18906 + 4(-1.29758) + (1.18698) = 1.21971$

$E:\quad f_4 = f(x_4,y_4) = y'_4 = -1.19015$

$C:\ y_4^{(2)} = y_2 + \frac{.2}{3}(-1.18906 + 4(-1.29758) + (-1.19015) = 1.21950$

$y_4^{(3)} = 1.21953,\quad y_4^{(4)} = 1.21953$

۱۹- جواب معادله دیفرانسیل *Vanderpol* رابا شرایط اولیه $y(0)=1,\ y'(0)=0$ به روش رانج - کوتای مرتبه 4 بدست آورید.

$y'' - (.1)(1 - y^2)y' + y = 0$

جواب: معادله را به دستگاه معادلات دیفرانسیل مرتبه اول تبدیل می‌کنیم با فرض $y_1 = y$ ، $h = .2$ داریم:

$$\begin{cases} y_1' = y_2 = f_1(t, y_1, y_2) \\ y_2' = (.1)(1 - y_1^2)y_2 - y_1 = f_2(t, y_1, y_2) \end{cases}$$

فرمول‌های رانج - کوتا را برای این دستگاه می‌نویسیم:

$$\begin{cases} y_{1,r+1} = y_{1,r} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ y_{2,r+1} = y_{2,r} + \frac{1}{6}(\ell_1 + 2\ell_2 + 2\ell_3 + \ell_4) \end{cases}$$

$$\begin{cases} k_1 = hf_1(t_r, y_{1,r}, y_{2,r}) = hy_{2,r} \\ k_2 = hf_1\left[t_r, y_{1,r} + \frac{k_1}{2}, y_{2,r} + \frac{\ell_1}{2}\right] = h\left[y_{2,r} + \frac{\ell_1}{2}\right] \\ k_3 = hf_1\left[t_r, y_{1,r} + \frac{k_2}{2}, y_{2,r} + \frac{\ell_2}{2}\right] = h\left[y_{2,r} + \frac{\ell_2}{2}\right] \\ k_4 = hf_1\left[t_r, y_{1,r} + k_3, y_{2,r} + \ell_3\right] = h\left[y_{2,r} + \ell_3\right] \end{cases}$$

$$\begin{cases} \ell_1 = hf_2(t_r, y_{1,r}, y_{2,r}) = (.1)(1 - y_{1,r}^2)y_{2,r} - y_{1,r} \\ \ell_2 = hf_2(t_r, y_{1,r} + \frac{k_1}{2}, y_{2,r} + \frac{\ell_1}{2}) \\ \ell_3 = hf_2(t_r, y_{1,r} + \frac{k_2}{2}, y_{2,r} + \frac{\ell_2}{2}) \\ \ell_4 = hf_2(t_r, y_{1,r} + \frac{k_3}{2}, y_{2,r} + \frac{\ell_3}{2}) \end{cases}$$

$$\begin{cases} k_1 = (.2)(0) = 0 \\ k_2 = (.2)(-.1) = -.02 \\ k_3 = (.2)(-.1) = -.02 \\ k_4 = (.2)(-.198) = -.04 \end{cases}$$

$$\begin{cases} \ell_1=(.2)\left\{(.1)[1-1](0)-1\right\}=-.2 \\ \ell_2=(.2)\left\{(.1)[1-1](-.1)-1\right\}=-.2 \\ \ell_3=(.2)\left\{(.1)[.02](-.1)-.99\right\}=-.198 \\ \ell_4=(.2)\{(.1)[.04](-.198)-.98\}=-.196 \end{cases}$$

$$y_{11}=.98$$

$$y_{21}=-.199$$

بهمین ترتیب بر حسب افزایش t سایر مقادیر y_{1r} , y_{2r} حاصل می‌شوند.

۲۰ - جواب معادله دیفرانسیل $y'=2x-y$ با شرط اولیه $y(0)=-1$ و $h=0.1$ با استفاده از روش رانج - کوتای مرتبه چهار برای $x_1=0.1$ و $x_2=0.2$ و $x_3=0.3$ بدست آورید و سپس روش میلن سیمپسون را بکار برده و جواب را در $x_4=0.4$ و $x_5=0.5$ بدست آورید.

جواب: جواب تحلیلی $y(x)=-3e^{-x}-2x+2$ می‌باشد.

جواب روش عددی را در $x=0.4$ و $x=0.5$ با جواب تحلیلی در این نقاط مقایسه کنید.

۲۱ - جواب معادله دیفرانسیل $y'=2x-y$ ؛ با شرط اولیه $y(0)=-1$ و $h=0.1$ با استفاده از روش رانج - کوتای مرتبه چهار برای $x_1=0.1$ و $x_2=0.2$ و $x_3=0.3$ بدست آورید و سپس روش آدامز - مولتن را بکار برده و جواب را در $x_4=0.4$ و $x_5=0.5$ بدست آورید.

جواب: با جواب مسأله ۲۰ مقایسه کنید.

۲۲ - جواب معادله دیفرانسیل زیر را

$$\frac{dy}{dx}=2x-y$$
$$y(0)=-1$$

با روش رانج - کوتا - فیلبرگ در $x=0.2(0.2)2$ بدست آورید.

جواب: جواب تحلیلی برابر است با: $y(x)=e^{x}+2x-2$

x	y محاسبه شده	y از مقدار تحلیلی
0.2	-0.781269	-0.781269
0.4	-0.529680	-0.529680
0.6	-0.251188	0.251188
0.8	0.049329	0.049329
1.0	0.367879	0.367879

۲۳ - جواب معادله دیفرانسیل زیر را

$$\frac{dy}{dx}=\frac{1}{x+y}$$
$$y(0)=2$$

با روش رانج - کوتا - فیلبرگ در $x=2$ بدست آورید.

جواب:

x	0.2	0.4	0.6	0.8	1
y	2.09327	0.217549	2.24927	2.20794	2.19060

۲۴ – جواب معادله دیفرانسیل زیر را به روش اویلر و رانج-کوتای مرتبه دو در $x=0.1$ به دست آورید.

$$\frac{dy}{dx}=x+y+xy$$
$$y(0)=1$$

جواب: $y(0.1)=1.11418$

۲۵ – جواب معادله دیفرانسیل $\frac{dy}{dx}=\sin(x)+y$ با شرط اولیه $y(0)=2$ را با روش اویلر و رانج-کوتای مرتبه دو در $x=0.1(0.1)0.5$ به دست آورید.

جواب:

x	0.1	0.2	0.3	0.4	0.5
y	2.2150	2.4630	2.7473	3.0715	3.4394

۲۶ – جواب معادله دیفرانسیل $\frac{dy}{dx}=f(x,y)=2x(y-1)$ در چند نقطه زیر داده شده است.
جواب معادله دیفرانسیل را در $x=0.5$ به روش میلن سیمپسون و آدامز به دست آورید.

x	y	f
0	0	0
0.1	−0.01005	−0.20201
0.2	−0.04081	−0.41632
0.3	−0.09417	−0.65650
0.4	−0.17351	−0.93881

جواب معادله دیفرانسیل را در $x=0.5$ به روش میلن سیمپسون و آدامز به دست آورید.

جواب: $y=-0.28403$ دقیق.

۲۷ – برای معادله دیفرانسیل $\frac{dy}{dx}=y-x^2$ و $y(0)=1$
جواب معادله در چند نقطه داده شده است.

$$y(0.2)=1.2186,\ y(0,4)=1.4682,\ y(0.6)=1.7379$$

روش میلن سیمپسون را به کار برده و جواب معادله را در $x=1.2$ به دست آورید.

جواب: $y(1.2)=2.5199$

۲۸ – جواب معادله دیفرانسیل زیر را به روش رانج – کوتا به‌دست آورید.

$$y'=y\ \sin(\pi x) \qquad x=0.2(0.2)\ 0.6$$

$y(0)=1$

سپس روش میلن سیمپسون را به کار برده و جواب معادله را در $x=0.8$ و $x=1$ به دست آورید.

جواب:

رانج - کوتا	x	0.2	0.4	0.6
	y	1.06267	1.24601	1.51691
میلن سیمپسون	x	0.8	1.0	
	y	1.77973	1.89205	

۲۹ - معادله دیفرانسیل مرتبه سوم زیر را به یک دستگاه معادلات دیفرانسیل مرتبه اول تبدیل کنید.

$y'''+xy''-xy'-2y=x$

$y(0) = y''(0)=0,\ y'(0)=1$

و جواب معادله را در $x=0.2,\ 0.4,\ 0.6$ به روش رانج - کوتا به دست آورید.

جواب:

t	y	y'	y''
0.2	0.2003	1.0053	0.0079
0.4	0.4042	1.0418	0.3095
0.6	0.6210	1.1380	0.6733

۳۰ - معادله دیفرانسیل زیر را در نظر می‌گیریم

$$\frac{d^2x}{dt^2}+64x = 16\cos(8t)$$

$$(x(0) = x'(0) = 0$$

جواب معادله را در $t=0.1(0.1)0.8$ به روش رانج - کوتای مرتبه چهار به دست آورید و با جواب تحلیلی $x=t\ \sin(8t)$ مقایسه کنید.

جواب:

t	0	0.1	0.2	0.3
x	0	0.0715	0.1983	0.2004
x'	0	1.2710	0.9512	-1.0812
x تحقیقی	0	0.718	0.1999	0.2016

برنامه هائی به زبان MATLAB

برنامه ۱ - brackPlot

زیر فاصله هائی را تعیین می کند که تابع $y = f(x)$ در آن زیر فاصله ها تغییر علامت می دهد.
در این صورت طبق قضیه بولزانو در آن زیر فاصله حداقل دارای یک ریشه است .

```
function Xb = brackPlot(fun,xmin,xmax,nx)
% brackPlot  Find subintervals on x that contain sign changes of f(x)
%
% Synopsis:  Xb = brackPlot(fun,xmin,xmax)
%            Xb = brackPlot(fun,xmin,xmax,nx)
%
% Input:     fun = (string) name of mfile function that evaluates f(x)
%            xmin,xmax = endpoints of interval to subdivide into brackets.
%            nx = (optional) number of samples along x axis used to test for
%                 brackets. The interval xmin <= x <= xmax is divided into
%                 nx-1 subintervals. Default:  nx = 20.
%
% Output:    Xb = two column matrix of bracket limits.  Xb(k,1) is the left
%                 (lower x value) bracket and Xb(k,2) is the right bracket for
%                 the k^th potential root.  If no brackets are found, Xb = [].

if nargin<4, nx=20; end

% --- Create data for a plot of f(x) on interval xmin <= x <= xmax
xp = linspace(xmin,xmax);  yp = feval(fun,xp);
% --- Save data used to draw boxes that indicate brackets
ytop = max(yp);  ybot = min(yp);          % y coordinates of the box
ybox = 0.05*[ybot ytop ytop ybot ybot];  % around a bracket
c = [0.7 0.7 0.7];                        % RGB color used to fill the box

% --- Begin search for brackets
x = linspace(xmin,xmax,nx);    % Vector of potential bracket limits
f = feval(fun,x);              % Vector of f(x) values at potential brackets
```

```
nb = 0; Xb = [];            % Xb is null unless brackets are found
for k = 1:length(f)-1
 if sign(f(k))~=sign(f(k+1)) % True if sign of f(x) changes in the interval
  nb = nb + 1;
  Xb(nb,:) = [x(k) x(k+1)]; % Save left and right ends of the bracket
  hold on; fill([x(k) x(k) x(k+1) x(k+1) x(k)],ybox,c); % Add filled box
 end
end
if isempty(Xb)  % Free advice
 warning('No brackets found. Check [xmin,xmax] or increase nx');
 return;        % return without drawing a plot
end
% --- Add plot here so that curve is on top of boxes used to indicate brackets
plot(xp,yp,[xmin xmax],[0 0]);
grid on; xlabel('x');
if isa(fun,'inline')
 ylabel(sprintf('Roots of f(x) = %s',formula(fun)));  % label is formul in f(x)
else
 ylabel(sprintf('Roots of f(x) defined in %s',fun)); % label is name of m-file
end
hold off
```

برنامه ۲ - fixpt

روش تکرار تابعی $x_{k+1} = g(x_k)$ و $p(k) = x_k$

```
function [k,p,err,P] = fixpt(g,p0,tol,max1)

% Input - g is the iteration function
%       - p0 is the initial guess for the fixed-point
%       - tol is the tolerance
%       - max1 is the maximum number of iterations
% Output - k is the number of iterations that were carried out
%          - p is the approximation to the fixed-point
%          - err is the error in the approximation
%          - P'contains the sequence {pn}

% NUMERICAL METHODS: Matlab Programs
% (c) 2004
P(1)= p0;
for k=2:max1
        P(k)=feval(g,P(k-1));
        err=abs(P(k)-P(k-1));
        relerr=err/(abs(P(k))+eps);
        p=P(k);
        if (err<tol) | (relerr<tol),break;end
end

if k == max1
        disp('maximum number of iterations exceeded')
end

P=P';
```

برنامه ۳ – bisect

برنامه روش **نصف کردن** فاصله [a,b] با خطای قابل اغماض delta

```
function [c,err,yc]=bisect(f,a,b,delta)

%Input - f is the function input as a string 'f'
%       - a and b are the left and right endpoints
%       - delta is the tolerance
%Output - c is the zero
%        - yc= f(c)
%        - err is the error estimate for c
% NUMERICAL METHODS: Matlab Programs
% (c) 2004
ya=feval(f,a);
yb=feval(f,b);
if ya*yb > 0,break,end
max1=1+round((log(b-a)-log(delta))/log(2));
for k=1:max1
        c=(a+b)/2;
        yc=feval(f,c);
        if yc==0
                a=c;
                b=c;
        elseif yb*yc>0
                b=c;
                yb=yc;
        else
                a=c;
                ya=yc;
        end
        if b-a < delta, break,end
end
c=(a+b)/2;
err=abs(b-a);
yc=feval(f,c);
```

برنامه ٤ –newton

روش نیوتن برای حل معادله $f(x) = 0$ با نقطه p_0 و خطای قابل اغماض epsilon و delta و حداکثر تکرار max1 می باشد.

```
function [p0,err,k,y]=newton(f,df,p0,delta,epsilon,max1)

%Input - f is the object function input as a string 'f'
%     - df is the derivative of f input as a string 'df'
%     - p0 is the initial approximation to a zero of f
%          - delta is the tolerance for p0
%          - epsilon is the tolerance for the function values y
%          - max1 is the maximum number of iterations
%Output - p0 is the Newton-Raphson approximation to the zero
%          - err is the error estimate for p0
%          - k is the number of iterations
%          - y is the function value f(p0)

% NUMERICAL METHODS: Matlab Programs
% (c) 2004
for k=1:max1
        p1=p0-feval(f,p0)/feval(df,p0);
        err=abs(p1-p0);
        relerr=2*err/(abs(p1)+delta);
        p0=p1;
        y=feval(f,p0);
        if (err<delta)|(relerr<delta)|(abs(y)<epsilon),break,end
end
```

برنامه ۵ – muller

روش مولر برای محاسبه ریشه معادله f(x) = 0 می باشد. به طوری که p0 و p1 و p2 نقاط تقریبی اولیه می باشند epsilon و delta خطاهای قابل اغماض می باشند.

```
function [p,y,err]=muller(f,p0,p1,p2,delta epsilon,max1)

%Input - f is the object function input as a string 'f'
%       - p0, p1, and p2 are the initial approximations
%    - delta is the tolerance for p0, p1, and p2
%       - epsilon the the tolerance for the function values y
%    - max1 is the maximum number of iterations
%Output- p is the Muller approximation to the zero of f
%     - y is the function value y = f(p)
%     - err is the error in the approximation of p.

% NUMERICAL METHODS: Matlab Programs
% (c) 2004

%Initalize the matrices P and Y
P=[p0 p1 p2];
Y=feval(f,P);

%Calculate a and b in formula

for k=1:max1
  h0=P(1)-P(3);h1=P(2)-P(3);e0=Y(1)-Y(3);e1=Y(2)-Y(3);c=Y(3);
  denom=h1*h0^2-h0*h1^2;
  a=(e0*h1-e1*h0)/denom;
  b=(e1*h0^2-e0*h1^2)/denom;

  %Suppress any complex roots
  if b^2-4*a*c > 0
    disc=sqrt(b^2-4*a*c);
  else
    disc=0;
  end
```

```
  %Find the smallest root
  if b < 0
    disc=-disc;
  end

  z=-2*c/(b+disc);
  p=P(3)+z;

  %Sort the entries of P to find the two closest to p
  if abs(p-P(2))<abs(p-P(1))
    Q=[P(2) P(1) P(3)];
    P=Q;
    Y=feval(f,P);
  end
  if abs(p-P(3))<abs(p-P(2))
    R=[P(1) P(3) P(2)];
    P=R;
    Y=feval(f,P);
  end

  %Replace the entry of P that was farthest from p with p
  P(3)=p;
  Y(3) = feval(f,P(3));
  y=Y(3);

  %Determine stopping criteria
  err=abs(z);
  relerr=err/(abs(p)+delta);
  if (err<delta)|(relerr<delta)|(abs(y)<epsilon)
    break
  end
end
```

برنامه ٦ – jacobi

روش ژاکوبی برای حل دستگاه n معادله n مجهول به روش تکراری به کار می رود.

A ماتریس ضرایب و B ماتریس N×1 مقادیر سمت راست و P ماتریس مقادیر اولیه می باشد.

delta خطای قابل اغماض می باشد. max1 حداکثر تکرار است.

```
function X=jacobi(A,B,P,delta, max1)

% Input  - A is an N x N nonsingular matrix
%        - B is an N x 1 matrix
%        - P is an N x 1 matrix; the initial guess
%                 - delta is the tolerance for P
%                 - max1 is the maximum number of iterations
% Output - X is an N x 1 matrix: the jacobi approximation to
%                  the solution of AX = B

% NUMERICAL METHODS: Matlab Programs
% (c) 2004

N = length(B);

for k=1:max1
  for j=1:N
    X(j)=(B(j)-A(j,[1:j-1,j+1:N])*P([1:j-1,j+1:N]))/A(j,j);
  end
  err=abs(norm(X'-P));
  relerr=err/(norm(X)+eps);
  P=X';
    if (err<delta)|(relerr<delta)
    break
  end
end

X=X';
```

برنامه۷ -gseid

روش گاوس- سیدل برای حل دستگاه n معادله n مجهول به روش تکراری به کار می رود. A ماتریس ضرایب و B ماتریس N×1 مقادیر سمت راست و P ماتریس مقادیر اولیه می باشد. delta خطای قابل اغماض می باشد. max1 حداکثر تکرار است.

```
function X=gseid(A,B,P,delta, max1)
% Input  - A is an N x N nonsingular matrix
%        - B is an N x 1 matrix
%        - P is an N x 1 matrix; the initial guess
%         - delta is the tolerance for P
%        - max1 is the maximum number of iterations
% Output - X is an N x 1 matrix: the gauss-seidel approximation to
%          the solution of AX = B
% NUMERICAL METHODS: Matlab Programs
% (c) 2004
N = length(B);
for k=1:max1
  for j=1:N
    if j==1
      X(1)=(B(1)-A(1,2:N)*P(2:N))/A(1,1);
    elseif j==N
          X(N)=(B(N)-A(N,1:N-1)*(X(1:N-1))')/A(N,N);
    else
          %X contains the kth approximations and P the (k-1)st
      X(j)=(B(j)-A(j,1:j-1)*X(1:j-1)'-A(j,j+1:N)*P(j+1:N))/A(j,j);
    end
  end
  err=abs(norm(X'-P));
  relerr=err/(norm(X)+eps);
  P=X';
    if (err<delta)|(relerr<delta)
    break
end
X=X';
```

برنامه ۸- cholesky

روش چولسکی برای حل دستگاه $Ax = b$ در صورتی که A متقارن باشد مورد استفاده قرار می گیرد. A را تجزیه می کنیم $A=C'*C$ و سپس دستگاه معادلات را حل می کنیم.

```
function C = Cholesky(A)
% Cholesky  Cholesky factorization of a symmetric, positive definite matrix
%
% Synopsis:  C = Cholesky(A)
%
% Input:     A = symmetric positive definite matrix
%
% Output:    C = upper triangular matrix such that A = C'*C

[m,n] = size(A);
if m~=n,  error('A must be square');  end
C = zeros(n,n);

for i=1:n
 for j=i:n
  if j==1
   s = A(i,i);  % i=1, j=1 is special case
  else
   s = A(i,j) - C(1:i-1,i)'*C(1:i-1,j);
  end
  if j>i
   C(i,j) = s/C(i,i);
  else
   if s<=0, error('C is not positive definite to working precision');  end
   C(i,i) = sqrt(s);
  end
 end
end
```

برنامه۹ – newtonsys

روش نیوتن را برای حل دستگاه معادلات غیر خطی به کار می بریم. jfun نام برنامه ای است که ماتریس j (ماتریس ژاکوبیان تابع f) و بردار f را برمی گرداند. در این صورت دستگاه n معادله n مجهولی حل می گردد.

```
function x = newtonSys(Jfun,x0,xtol,ftol,maxit,verbose,varargin)
% newtonSys  Newton's method for systems of nonlinear equations.
%
% Synopsis:  x = newtonSys(Jfun,x0)
%            x = newtonSys(Jfun,x0,xtol)
%            x = newtonSys(Jfun,x0,xtol,ftol)
%            x = newtonSys(Jfun,x0,xtol,ftol,verbose)
%            x = newtonSys(Jfun,x0,xtol,ftol,verbose,arg1,arg2,...)
%
% Input:  Jfun = (string) name of mfile that returns matrix J and vector f
%         x0   = initial guess at solution vector, x
%         xtol = (optional) tolerance on norm(dx).  Default: xtol=5e-5
%         ftol = (optional) tolerance on norm(f).   Default: ftol=5e-5
%         verbose = (optional) flag.  Default: verbose=0, no printing.
%         arg1,arg2,... = (optional) optional arguments that are passed
%                 through to the mfile defined by the 'Jfun' argument
%
% Note:  Use [] to request default value of an optional input.  For example,
%        x = newtonSys('JFun',x0,[],[],[],arg1,arg2) passes arg1 and arg2 to
%        'JFun', while using the default values for xtol, ftol, and verbose
%
% Output:  x = solution vector;  x is returned after k iterations if tolerances
%          are met, or after maxit iterations if tolerances are not met.

if nargin < 3 | isempty(xtol),   xtol = 5e-5;  end
if nargin < 4 | isempty(ftol),   ftol = 5e-5;  end
if nargin < 5 | isempty(maxit),  maxit = 15;   end
if nargin < 5 | isempty(verbose), verbose = 0;  end
xeps = max(xtol,5*eps);  feps = max(ftol,5*eps);   % Smallest tols are
5*eps
if verbose, fprintf('\nNewton iterations\n  k    norm(f)     norm(dx)\n'); end
```

```
x = x0;  k = 0;        % Initial guess and current number of iterations
while k <= maxit
  k = k + 1;
  [J,f] = feval(Jfun,x,varargin{:});  %  Returns Jacobian matrix and f vector
  dx = J\f;
  x = x - dx;
  if verbose, fprintf('%3d %12.3e %12.3e\n',k,norm(f),norm(dx)); end
  if ( norm(f) < feps ) | ( norm(dx) < xeps ), return; end
end
warning(sprintf('Solution not found within tolerance after %d
iterations\n',k));
```

برنامه ۱۰-۱power

روش توانی برای به دست آوردن بزرگترین مقدار ویژه از نظر قدر مطلق lambda و بردار ویژه متناظر با آن V مورد استفاده قرار می گیرد.

```
function [lambda,V]= power1(A,X,epsilon,max1)
%Input - A is an nxn matrix
%     - X is the nx1 starting vector
%     - epsilon is the tolerance
%     - max1 is the maximum number of iterations
%Output - lambda is the dominant eigenvalue
%      - V is the dominant eigenvector
% NUMERICAL METHODS: Matlab Programs
% (c) 2004
%Initialize parameters
lambda=0;
cnt=0;
err=1;
state=1;
while ((cnt<=max1)&(state==1))
  Y=A*X;
   %Normalize Y
  [m j]=max(abs(Y));
  c1=m;
  dc=abs(lambda-c1);
  Y=(1/c1)*Y;
    %Update X and lambda and check for convergence
  dv=norm(X-Y);
  err=max(dc,dv);
  X=Y;
  lambda=c1;
  state=0;
  if (err>epsilon)
    state=1;
  end
  cnt=cnt+1;
end
V=X;
```

برنامه ۱۱ - lagrange

روش لاگرانژ را برای درونیابی یک چند جمله ای درجهٔ n نظیر به n+1 نقطه داده شده (x,y) به کار می بریم. L ماتریس تولید شده می باشد و C ماتریسی است که شامل ضرایب چند جمله ای لاگرانژ می باشد.

```
function [C,L]=lagran(X,Y)

%Input  - X is a vector that contains a list of abscissas
%       - Y is a vector that contains a list of ordinates
%Output - C is a matrix that contains the coefficents of
%         the Lagrange interpolatory polynomial
%       - L is a matrix that contains the Lagrange
%         coefficient polynomials

% NUMERICAL METHODS: Matlab Programs
% (c) 2004
w=length(X);
n=w-1;
L=zeros(w,w);

%Form the Lagrange coefficient polynomials
for k=1:n+1
  V=1;
  for j=1:n+1
    if k~=j
      V=conv(V,poly(X(j)))/(X(k)-X(j));
    end
  end
  L(k,:)=V;
end

%Determine the coefficients of the Lagrange interpolator
%polynomial

C=Y*L;
```

برنامه ۱۲ – divDiffTable

این برنامه جدول ضرایب تفاضلهای محدود را تولید می کند. نقاط (x,y) داده شده اند. D ماتریس ضرایب تفاضل های محدود را می دهد و عناصر روی قطر D ضرایب چند جمله ای نیوتن می باشد.

```
function D = divDiffTable(x,y)
% divdiffTable  Construct a table of divided-difference coefficients
%
% Synopsis:  D = divDiffTable(x,y)
%
% Input:     x,y = vectors containing the tabulated y = f(x) data
%
% Output:    D = matrix containing divided-difference coefficients in
%            its lower triangle.  Diagonal entries are coefficients
%            of the Newton polynomial
n = length(y);
if length(x)~=n,  error('x and y are not compatible');  end

D = zeros(n,n);
D(:,1) = y(:);   % First column is zeroth order difference, f[x_i] = y_i
for j=2:n
 for i=j:n
  D(i,j) = (D(i,j-1)-D(i-1,j-1))/(x(i)-x(i-j+1));
 end
end
```

برنامهٔ ۱۳ – fitnorm

در این برنامه روش **حداقل مربعات** خطاها مورد استفاده قرار می گیرد. نقاط (x,y) داده شده اند. basefun نام برنامه ای است که ماتریس A را برگشت می دهد. ستونهای A مقادیر توابع پایه می باشند که بازاء x محاسبه شده اند.

```
function [c,R2,rout] = fitnorm(x,y,basefun)
% fitnorm  Least-squares fit via solution to the normal equations
%        Given ordered pairs of data, (x_i,y_i), i=1,...,m, fitnorm
%        returns the vector of coefficients, c_1,...,c_n, such that
%           F(x) = c_1*f_1(x) + c_2*f_2(x) + ... + c_n*f_n(x)
%        minimizes the L2 norm of y_i - F(x_i).
%
% Synopsis:  c     = fitnorm(x,y,basefun)
%        [c,R2]  = fitnorm(x,y,basefun)
%        [c,R2,r] = fitnorm(x,y,basefun)
%
% Input:   x,y    = vectors of data to be fit
%        basefun = (string) name of user-supplied m-file that computes
%                  matrix A.  The columns of A are the values of the
%                  basis functions evaluated at the x data points.
%
% Output:  c = vector of coefficients obtained from the fit
%        R2 = (optional) adjusted coefficient of determination, 0 <= R2 <= 1
%           R2 close to 1 indicates a strong relationship between y and x
%        r  = (optional) residuals of the fit

if length(y)~= length(x);  error('x and y are not compatible');  end

A = feval(basefun,x(:)); %  Coefficient matrix of overdetermined system
c = (A'*A)\(A'*y(:));    %  Solve normal equations, y(:) is always a column
if nargout>1
  r = y - A*c;           %  Residuals at data points used to obtain the fit
  [m,n] = size(A);
  R2 = 1 - (m-1)/(m-n-1)*(norm(r)/norm(y-mean(y)))^2;
  if nargout>2,  rout = r;  end
end
```

برنامه ۱٤ – romberge

این برنامه برای محاسبه انتگرال به **روش رامبرگ** به کار می رود.

a و b حد پایین و بالا انتگرال می باشند. f تابع مورد نظر می باشد. n حداکثر تعداد سطرهای جدول بوده و h کوچکترین تقسیم فاصله می باشد. quad نتیجه انتگرال است.

```
function [R,quad,err,h]=romberge(f,a,b,n,tol)
%Input  - f is the integrand input as a string 'f'
%      - a and b are upper and lower limits of integration
%      - n is the maximum number of rows in the table
%      - tol is the tolerance
%Output - R is the Romberg table
%      - quad is the quadrature value
%      - err is the error estimate
%      - h is the smallest step size used
% NUMERICAL METHODS: Matlab Programs
% (c) 2004
M=1;
h=b-a;
err=1;
J=0;
R=zeros(4,4);
R(1,1)=h*(feval(f,a)+feval(f,b))/2;
while((err>tol)&(J<n))|(J<4)
  J=J+1;
  h=h/2;
  s=0;
  for p=1:M
    x=a+h*(2*p-1);
    s=s+feval(f,x);
  end
  R(J+1,1)=R(J,1)/2+h*s;
  M=2*M;
  for K=1:J
    R(J+1,K+1)=R(J+1,K)+(R(J+1,K)-R(J,K))/(4^K-1);
  end
  err=abs(R(J,J)-R(J+1,K+1));
end
quad=R(J+1,J+1);
```

برنامهٔ ۱۵ – gauss

در این برنامه **روش انتگرال گیری گاوس** را به کار می بریم.

f تابع مورد نظر a و b حد پایین و بالای انتگرال می باشند .

A بردار مقادیر نقاط تقسیم در جدول می باشد.

W بردار وزنها نظیر به نقاط تقسیم در جدول می باشد.

```
function quad=gauss(f,a,b,A,W)

%Input  - f is the integrand input as a string 'f'
%       - a and b upper and lower limits of integration
%       - A is the 1 x N vector of abscissas from the Table
%       - W is the 1 x N vector of weights from the Table
%Output - quad is the quadrature value

% NUMERICAL METHODS: Matlab Programs
% (c) 2004
N=length(A);
T=zeros(1,N);
T=((a+b)/2)+((b-a)/2)*A;
quad=((b-a)/2)*sum(W.*feval(f,T));
```

برنامه۱۶ - odeRK4

روش رانج - کوتای مرتبه ٤ را برای حل معادله دیفرانسیل مرتبه اول $y' = f(x,y)$ به کار می بریم.

diffeq نام برنامه ای است که مقدار تابع f(x,y) را معین می کند.

y جواب عددی مسأله در $x_j = t(j) = (j-1)* h$ می باشد.

```
function [t,y] = odeRK4(diffeq,tn,h,y0)
% odeRK4  Fourth order Runge-Kutta method for a single, first order ODE
%
% Synopsis:  [t,y] = odeRK4(fun,tn,h,y0)
%
% Input:     diffeq = (string) name of the m-file that evaluates the right
%                     hand side of the ODE written in standard form
%            tn  = stopping value of the independent variable
%            h   = stepsize for advancing the independent variable
%            y0  = initial condition for the dependent variable
%
% Output:    t = vector of independent variable values:  t(j) = (j-1)*h
%            y = vector of numerical solution values at the t(j)

t = (0:h:tn)';                % Column vector of elements with spacing h
n = length(t);                % Number of elements in the t vector
y = y0*ones(n,1);             % Preallocate y for speed
h2 = h/2;  h3 = h/3;  h6 = h/6;  % Avoid repeated evaluation of constants

% Begin RK4 integration; j=1 for initial condition
for j=2:n
  k1 = feval(diffeq, t(j-1),    y(j-1)       );
  k2 = feval(diffeq, t(j-1)+h2, y(j-1)+h2*k1 );
  k3 = feval(diffeq, t(j-1)+h2, y(j-1)+h2*k2 );
  k4 = feval(diffeq, t(j-1)+h,  y(j-1)+h*k3  );
  y(j) = y(j-1) + h6*(k1+k4) + h3*(k2+k3);
end
```

برنامهٔ ۱۷ – milne

روش میلن- سیمپسون: برای حل معادله دیفرانسیل $y' = f(x,y)$ می باشد که روشی بر مبنای پیش بینی- تصحیح می باشد.

```
function M=milne(f,T,Y)

%Input  - f is the function entered as a string 'f'
%       - T is the vector of abscissas
%       - Y is the vector of ordinates
%Remark. The first four coordinates of T and Y must
%       have starting values obtained with RK4
%Output - M=[T' Y'] where T is the vector of abscissas and
%       Y is the vector of ordinates

% NUMERICAL METHODS: Matlab Programs
% (c) 2004
n=length(T);
if n<5, break, end;

F=zeros(1,4);
F=feval(f,T(1:4),Y(1:4));
h=T(2)-T(1);
pold=0;
yold=0;

for k=4:n-1
  %Predictor
  pnew=Y(k-3)+(4*h/3)*(F(2:4)*[2 -1 2]');
  %Modifier
  pmod=pnew+28*(yold-pold)/29;
  T(k+1)=T(1)+h*k;
  F=[F(2) F(3) F(4) feval(f,T(k+1),pmod)];
  %Corrector
  Y(k+1)=Y(k-1)+(h/3)*(F(2:4)*[1 4 1]');
  pold=pnew;
  yold=Y(k+1);
  F(4)=feval(f,T(k+1),Y(k+1));
end
M=[T' Y'];
```

برنامهٔ ۱۸- odeRK4sysv -

در این برنامه **روش رانج - کوتای مرتبه ٤** را برای حل دستگاه معادلهٔ دیفرانسیل مرتبه اول به کار می بریم.

```
function [t,y] = odeRK4sysv(diffeq,tn,h,y0,varargin)
% odeRK4sysv  Fourth order Runge-Kutta method for systems of first order
ODEs
%          Vectorized version with pass-through parameters.
%
% Synopsis:  [t,y] = odeRK4sysv(diffeq,tn,h,y0)
%          [t,y] = odeRK4sysv(diffeq,tn,h,y0,arg1,arg2,...)
%
% Input:     diffeq = (string) name of the m-file that evaluates the right
%                hand side of the ODE system written in standard form.
%          tn     = stoping value of the independent variable
%          h      = stepsize for advancing the independent variable
%          y0     = vector of the dependent variable values at x = x0
%          arg1,arg2 = list of additional arguments passed through
%                odeRK4sysv to the ``diffeq'' routine.
%
% Output:    t = vector of independent variable values:  t(j) = x0 + j*h
%          y = matrix of dependent variables values, one column for each
%            state variable.  Each row is from a different time step.

t = (0:h:tn)';          % Column vector of elements with spacing h
nt = length(t);         % number of steps (+1 for the initial conditions)
neq = length(y0);       % number of equations simultaneously advanced
y = zeros(nt,neq);      % Preallocate y for speed
y(1,:) = y0(:)';        % Assign IC. y0(:) is column, y0(:)' is row vector
flag = [];              % Dummy argument for compatibility with rhs functions
                        % developed for built-in routines

h2 = h/2;  h3 = h/3;  h6 = h/6;  % Avoid repeated evaluation of constants
k1 = zeros(neq,1);  k2 = k1;     % Preallocate memory for the Runge-Kutta
k3 = k1;  k4 = k1;  ytemp = k1;  % coefficients and a temporary vector
```

```
% Outer loop for all steps:  j = time step index;  k = equation number index
% Note use of transpose on definition of yold, and in formula for y(j,:)
for j=2:nt
  told = t(j-1);  yold = y(j-1,:)';                % Temp variables
  k1 = feval(diffeq,told,yold,flag,varargin{:});   % Slopes at the start
  ytemp = yold + h2*k1;
  k2 = feval(diffeq,told+h2,ytemp,flag,varargin{:}); % 1st slope at midpoint
  ytemp = yold + h2*k2;
  k3 = feval(diffeq,told+h2,ytemp,flag,varargin{:}); % 2nd slope at
midpoint
  ytemp = yold + h*k3;
  k4 = feval(diffeq,told+h,ytemp,flag,varargin{:}); % Slope at endpoint
  y(j,:) = ( yold + h6*(k1+k4) + h3*(k2+k3) )';     % Advance all equations
end
```

برنامهٔ ۱۹ – hermint

در این برنامهٔ **روش هرمیت** را برای درون یابی چند جمله ای درجهٔ 3 نقاط داده شده (x,y) به کارمی بریم . f و fp مقادیر f(x) و مشتق آن $f'(x)$ می باشند و به ازای xhat مقدار yhat محاسبه می شود.

```
function yhat = hermint(x,f,fp,xhat)
% hermint  Piecewise-cubic Hermite interpolation
%
% Synopsis: yhat = hermint(x,f,fp,xhat)
% Input:   x    = vector of independent variable values
%          f, fp = vectors of f(x) and f'(x)
%          xhat = (scalar or vector) x values where interpolant is evaluated
%
% Output:  yhat = scalar or vector value of cubic hermite interpolant at
%                 x = xhat.  size(yhat) = size(xhat)
n = length(x);
if    length(f)~=n,    error('x and f are not compatible');
elseif length(fp)~=n,  error('x and fp are not compatible');  end
% --- Construct coefficients of the piecewise interpolants
x = x(:);  xhat = xhat(:);   % Convert to column vectors
f = f(:);  fp = fp(:);
dx = diff(x);              % Vector of x(i+1) - x(i) values
divdif = diff(f)./dx;      % Vector of divided differences, f[x(i),x(i+1)]
a = f(1:n-1);
b = fp(1:n-1);
c = ( 3*divdif - 2*fp(1:n-1) - fp(2:n) ) ./dx;
d = ( fp(1:n-1) - 2*divdif + fp(2:n) )  ./dx.^2;
% --- Locate each xhat value in the x vector
i = zeros(size(xhat));  % i is index into x such that x(i) <= xhat <= x(i+1)
for m=1:length(xhat)    % For vector xhat: x( i(m) ) <= xhat(m) <= x(
i(m)+1 )
  i(m) = binSearch(x,xhat(m));
end
% --- Nested, vectorized evaluation of the piecewise polynomials
xx = xhat - x(i);
yhat = a(i) + xx.*(b(i) + xx.*(c(i) + xx.*d(i)) );
```

برنامهٔ ۲۰ – splint

در این برنامه روش چند جمله ای درونیاب درجه 3 اسپلاین را به کار می بریم و در هر سه حالت شرایط انتهائی با "مقدار ثابت" , "طبیعی" و "غیر از گره" محاسبات را انجام می دهیم. و بازاء هر xhat مقدار yhat را محاسبه می کنیم.

```
function [yhat,aa,bb,cc,dd] = splint(x,y,xhat,opt1,opt2)
% splint  Cubic-spline interpolation with various end conditions
%
% Synopsis:  yhat = splint(x,y,xhat)
%          yhat = splint(x,y,xhat,endType)
%          yhat = splint(x,y,xhat,fp1,fpn)
%          [yhat,a,b,c,d] = splint(x,y,xhat)
%          [yhat,a,b,c,d] = splint(x,y,xhat,endType)
%          [yhat,a,b,c,d] = splint(x,y,xhat,fp1,fpn)
%
% Input:  x,y  = vectors of discrete x and y = f(x) values
%       xhat = (scalar or vector) x value(s) where interpolant is evaluated
%       endType = (string, optional) either 'natural' or 'notaKnot';  used
%          to select either type of end conditions.  End conditions must be
%          same on both ends. Default: endType='notaKnot'.  For fixed slope
%          end conditions, values of f'(x) are specified, not endType
%       fp1 = (optional) slope at x(1), i,e., fp1 = f'(x(1))
%       fpn = (optional) slope at x(n), i,e., fpn = f'(x(n));
%
% Output:  yhat = (vector or scalar) value(s) of the cubic spline interpolant
%             evaluated at xhat.  size(yhat) = size(xhat)
%        a,b,c,d = (optional) coefficients of cubic spline interpolants

% --- Process optional input arguments
if nargin<3
 error('minimum of three input arguments needed');
elseif nargin==3
 endType = 'notaknot';
elseif nargin==4   % four input arguments => natural or not-a-knot end
conditions
 if ~ischar(opt1)
  error('Third argument must be a string indicating type of end conditions');
 end
```

```
  if strncmp('not',lower(opt1),3)
    endType = 'notaknot';
  elseif strncmp('nat',lower(opt1),3)
    endType = 'natural';
  else
    error(sprintf('endType = %s not allowed',endType));
  end
elseif nargin==5  % five input arguments => fixed slope end conditions
  if ischar(opt1) | ischar(opt2)
    error('Fourth, and fifth arguments must be numbers');
  end
  yp1 = opt1;   ypn = opt2;  endType = 'fixSlope';
else
  error(sprintf('%d input arguments not allowed',nargin));
end

% --- Set up system of equations for b(i)
x = x(:);  y = y(:);  xhat = xhat(:);  % convert to column vectors
n = length(x);
dx = diff(x);              % Vector of x(i+1) - x(i) values
divdif = diff(y)./dx;      % divided difference, f[x(i),x(i+1)]

alpha = [0; dx(1:n-2); 0];              % sub diagonal
bbeta = [1; 2*(dx(1:n-2)+dx(2:n-1)); 1];  % main diagonal
gamma = [0; dx(2:n-1); 0];              % super diagonal
A     = tridiags(n,bbeta,alpha,gamma);    % Sparse, tridiagonal matrix
delta = [ 0; 3*(divdif(2:n-1).*dx(1:n-2) + divdif(1:n-2).*dx(2:n-1)); 0 ];

% --- Modify system of equations as appropriate for the end conditions
if strncmp('not',lower(endType),3)        % not a knot
  A(1,1) = dx(2);   A(1,2) = dx(1) + dx(2);          % Equation for b(1)
  delta(1) = ( dx(2)*(2*dx(2)+3*dx(1))*divdif(1)...
           + dx(1)*dx(1)*divdif(2) ) /(dx(1)+dx(2));
  A(n,n-1) = dx(n-2) + dx(n-1);   A(n,n) = dx(n-2);   % Equation for b(n)
  delta(n) = ( dx(n-2)*(2*dx(n-2)+3*dx(n-1))*divdif(n-1) ...
           + dx(n-1)*dx(n-1)*divdif(n-2) ) / (dx(n-2)+dx(n-1));
elseif strncmp('nat',lower(endType),3)    % natural end conditions
  A(1,2) = 0.5;  delta(1) = 1.5*divdif(1);          % y''(x(1)) = 0
  A(n,n-1) = 1;  A(n,n) = 2;  delta(n) = 3*divdif(n-1);  % y''(x(n)) = 0
elseif strncmp('fix',lower(endType),3)    % prescribed slope end conditions
  delta(1) = yp1;  delta(n) = ypn;
```

```
else
 error(sprintf('Logic error:  endType = %s',endType));
end

% --- Solve the system for b
mmdflag = spparms('autommd');    % Store minimum degree ordering flag
spparms('autommd',0);            % Set that flag to zero
b = A\delta;                     % Solve the system
spparms('autommd',mmdflag);      % Reset the minimum degree ordering
flag

% --- Compute coefficients of spline interpolants
a = y(1:n-1);
c = (3*divdif - 2*b(1:n-1) - b(2:n))./dx;
d = (b(1:n-1) - 2*divdif + b(2:n))./dx.^2;
b(n) = [];                       % discard b(n)

% --- Locate each xhat value in the x vector
i = zeros(size(xhat));  % i is index into x such that x(i) <= xhat <= x(i+1)
for m=1:length(xhat)    % For vector xhat: x( i(m) ) <= xhat(m) <= x(
i(m)+1 )
  i(m) = binSearch(x,xhat(m));
end

% --- Nested, vectorized evaluation of the piecewise polynomials
xx = xhat - x(i);
yhat = a(i) + xx.*(b(i) + xx.*(c(i) + xx.*d(i)) );

if nargout>1,  aa = a;  bb = b;  cc = c;  dd = d;  end  % optional outputs
```

برنامه ۲۱ - compSplinePlot

این برنامه سه حالت شرایط انتهایی اسپلاین درجه 3 را محاسبه می کند و خم آنها را جهت مقایسه نشان میدهد.

```
function compSplinePlot(n)
% compSplinePlot  Compare end conditions for cubic-spline interpolants
%            Approximations to y = x*exp(-x) are constructed and plotted
%
% Synopsis:  compSplinePlot
%         compSplinePlot(n)
% Input:    n = (optional) number of knots in the range 0 <= x <= 5
%            Default:  n=6
% Output:   Plot of spline approximations to y = x*exp(-x) with not-a-knot,
%         natural, zero-slope, and exact-slope end conditions.  Normalized
%         errors for each interpolant are also computed and printed
if nargin<1, n=6;  end
x = linspace(0,5,n)';              % Generate discrete data set
y = x.*exp(-x);
xi = linspace(min(x),max(x))';     % Evaluate spline at these xi
ye = xi.*exp(-xi);                 % Exact f(x) at the xi
yi = splint(x,y,xi,'natural');     % Spline with natural end conditions
errNat = norm(yi-ye)
subplot(2,2,1);  plot(x,y,'bo',xi,ye,'b-',xi,yi,'r--');  axis([0 6 0 0.5]);
legend('knots','spline','x*exp(-x)');  title('Natural end conditions');
yi = splint(x,y,xi,0,0);           % Spline with zero-slope end conditions
errz = norm(yi-ye)
subplot(2,2,2);  plot(x,y,'bo',xi,ye,'b-',xi,yi,'r--');  axis([0 6 0 0.5]);
legend('knots','spline','x*exp(-x)');  title('Zero-slope end conditions');
yi = splint(x,y,xi);               % Spline with not-a-knot end conditions
errNot = norm(yi-ye)
subplot(2,2,3);  plot(x,y,'bo',xi,ye,'b-',xi,yi,'r--');  axis([0 6 0 0.5]);
legend('knots','spline','x*exp(-x)');  title('Not-a-knot end conditions');
yp1 = (1-x(1))*exp(-x(1));         % Exact slope at x(1)
ypn = (1-x(n))*exp(-x(n));         % and at x(n)
yi = splint(x,y,xi,yp1,ypn);       % Spline with exact-slope end conditions
errExs = norm(yi-ye)
subplot(2,2,4);  plot(x,y,'bo',xi,ye,'b-',xi,yi,'r--');  axis([0 6 0 0.5]);
legend('knots','spline','x*exp(-x)');  title('Exact-slope end conditions');
```

منابع

Acton, F. S. (1970). *Numerical Methods That Work.* New York: Harper and Row.

Allaire, Paul E. (1985). *Basics of the Finite Element Method.* Dubuque, Iowa: Brown.

Allen, D. N. (1954). *Relation Methods.* New York: McGraw-Hill.

Anderson, E., Z. Gai, C. Bishof, J. Demmel, J. Dongarra, et al. (1992). *LAPACK Users' Guide.* Philadelphia: SIAM.

Andrews, G., and R. Olsson (1993). *The SR Programming Language.* Redwood City, Calif.: Benjamin Cummings.

Andrews, Larry C. (1985). *Elementary Partial Differential Equations with Boundary Value Problems.* Philadelphia: Saunders College Publishing.

Arney, David C. (1987). *The Student Edition of DERIVE, Manual.* Reading, Mass.: Addison-Wesley—Benjamin/Cummings.

Atkinson, K. E. (1978). *An Introduction to Numerical Analysis.* New York: Wiley.

Bartels, Richard, J. Beatty, and B. Barsky (1987). *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Los Altos, Calif.: Morgan Kaufmann.

Birkhoff, Garrett, Richard Varga, and David Young (1962). Alternating direction implicit methods. *Advances in Computers* 3:187–273.

Bracewell, Ronald N. (1986). *The Hartley Transform.* New York: Oxford University Press.

Brigham, E. Oron (1974). *The Fast Fourier Transform.* Englewood Cliffs, N.J.: Prentice-Hall.

Burnett, David S. (1987). *Finite Element Analysis: From Concepts to Applications.* Reading, Mass.: Addison-Wesley.

Campbell, Leon, and Laizi Jacchia (1941). *The Story of Variable Stars.* Philadelphia: Blakiston.

Carnahan, Brice (1964). *Radiation Induced Cracking of Pentanes and Dimethylbutanes.* Ph.D. dissertation, University of Michigan.

Carnahan, Brice, et al. (1969). *Applied Numerical Methods.* New York: Wiley.

Carslaw, H. S., and J. C. Jaeger (1959). *Conduction of Heat in Solids.* 2nd ed. London: Oxford University Press.

Chandy, K. M., and S. Taylor (1992). *An Introduction to Parallel Programming.* Boston: Jones and Bartlett.

Char, B., K. Geddes, G. Gonnet, B. Leong, M. Monagan, and S. Watt (1992). *First Leaves: A Tutorial Introduction to Maple V.* New York: Springer-Verlag.

Condon, Edward, and Hugh Odishaw, eds. (1967). *Handbook of Physics.* New York: McGraw-Hill.

Conte, S. D., and C. de Boor (1980). *Elementary Numerical Analysis.* 3rd ed. New York: McGraw-Hill.

Cooley, J. W., and J. W. Tukey (1965). An algorithm for the machine calculations of complex Fourier series. *Mathematics of Computation* 19:297–301.

Corliss, G., and Y. F. Chang (1982). Solving ordinary differential equations using Taylor series. *ACM Transactions on Mathematical Software* 8:114–144.

Crow, Frank (1987). Origins of a teapot. *IEEE Computer Graphics and Applications* 7(1):8–19.

Davis, Alan J. (1980). *The Finite Element Method.* Oxford: Clarendon Press.

Davis, Phillip J., and Philip Rabinowitz (1967). *Numerical Integration.* Waltham, Mass.: Blaisdell.

de Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.

De Santis, R., F. Gironi, and L. Marelli (1976). Vector-liquid equilibrium from a hard-sphere equation of state. *Industrial and Engineering Chemistry Fundamentals* 15(3):183–189.

Dongarra, J., I. Duff, D. Sorensen, and H. van der Vorst (1991). *Solving Linear Systems on Vector and Shared Memory Computers*. Philadelphia: SIAM.

Dongarra, J. J., J. R. Bunch, C. B. Moler, and G. W. Stewart (1979). *LINPACK User's Guide*. Philadelphia: SIAM.

Douglas, J. (1962). Alternating direction methods for three space variables. *Numerical Mathematics* 4:41–63.

Duffy, A. R., J. E. Sorenson, and R. E. Mesloh (1967). Heat transfer characteristics of belowground LNG storage. *Chemical Engineering Progress* 63(6):55–61.

Etter, D. M. (1993). *Quattro Pro—A Software Tool for Engineers and Scientists*. Redwood City, Calif.: Benjamin/Cummings.

Fike, C. T. (1968). *Computer Evaluation of Mathematical Functions*. Englewood Cliffs, N.J.: Prentice-Hall.

Forsythe, G. E., M. A. Malcolm, and C. B. Moler (1977). *Computer Methods for Mathematical Computation*. Englewood Cliffs, N.J.: Prentice-Hall.

Forsythe, G. E., and C. B. Moler (1967). *Computer Solution of Linear Algebraic Systems*. Englewood Cliffs, N.J.: Prentice-Hall.

Fox, L. (1965). *An Introduction to Numerical Linear Analysis*. New York: Oxford University Press.

Gear, C. W. (1967). The numerical integration of ordinary differential equations. *Mathematics of Computation* 21:146–156.

Gear, C. W. (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, N.J.: Prentice-Hall.

Hageman, L. A., and D. M. Young (1981). *Applied Iterative Methods*. New York: Academic Press.

Hamming, R. W. (1973). *Numerical Methods for Scientists and Engineers*. 2nd ed. New York: McGraw-Hill.

Hamming, R. W. (1971). *Introduction to Applied Numerical Analysis*. New York: McGraw-Hill.

Harrington, Steven (1987). *Computer Graphics: A Programming Approach*. New York: McGraw-Hill.

Henrici, P. H. (1964). *Elements of Numerical Analysis*. New York: Wiley.

Hornbeck, R. W. (1975). *Numerical Methods*. New York: Quantum.

Householder, A. S. (1970). *The Numerical Treatment of a Single Nonlinear Equation*. New York: McGraw-Hill.

IEEE Standard for Binary Floating-Point Arithmetic (1985). Institute of Electrical and Electronics Engineers, Inc., New York.

JaJa, J. (1992). *An Introduction to Parallel Algorithms*. Reading, Mass.: Addison-Wesley.

Jones, B. (1982). A note on the T transformation. *Nonlinear Analysis, Theory, Methods and Applications* 6:303–305.

Kahaner, D., C. Moler, S. Nash (1989). *Numerical Methods and Software*. Englewood Cliffs, N.J.: Prentice-Hall.

Lee, Peter, and Geoffrey Duffy (1976). Relationships between velocity profiles and drag reduction in turbulent fiber suspension flow. *Journal of the American Institute of Chemical Engineering* 22(4):750–753.

Love, Carl H. (1966). *Abscissas and Weights for Gaussian Quadrature*. National Bureau of Standards, Monograph 98.

Muller, D. E. (1956). A method of solving algebraic equations using an automatic computer. *Math Tables and Other Aids to Computation* 10:208–215.

O'Neill, Mark A. (1988). Faster Than Fast Fourier. *BYTE* 13(4):293–300.

Orvis, William J. (1987). *1-2-3 for Scientists and Engineers*. San Francisco: Sybex.

Peaceman, D. W., and H. H. Rachford (1955). The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics* 3:28–41.

Penrod, E. B., and K. V. Prasanna (1962). Design of a flat-plate collector for a solar earth heat pump. *Solar Energy* 6(1)9–22.

Pinsky, Mark A. (1991). *Partial Differential Equations and Boundary Value Problems with Applications*. 2nd ed. New York: McGraw-Hill.

Pizer, Stephen M. (1975). *Numerical Computing and Mathematical Analysis*. Chicago: Science Research Associates.

Pokorny, C., and C. Gerald (1989). *Computer Graphics: The Principles Behind the Art and Science*. Irvine, Calif.: Franklin, Beedle, and Associates.

Prenter, P. M. (1975). *Splines and Variational Methods*. New York: Wiley.

Press, W., B. Flannery, S. Teudolsky, and W. Vetterling (1992). *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. 2nd ed. New York: Cambridge University Press.

Press, W., B. Flannery, S. Teudolsky, and W. Vetterling (1992). *Numerical Recipes in C: The Art of Scientific Computing*. 2nd ed. New York: Cambridge Univeristy Press.

Press, W., B. Flannery, S. Teudolsky, and W. Vetterling (1989). *Numerical Recipes in Pascal: The Art of Scientific Computing*. Rev. 1st ed. New York: Cambridge University Press.

Ralston, Anthony (1965). *A First Course in Numerical Analysis*. New York: McGraw-Hill.

Ramirez, Robert W. (1985). *The FFT, Fundamentals and Concepts*. Englewood Cliffs, N.J.: Prentice-Hall.

Rice, John R. (1983). *Numerical Methods, Software, and Analysis*. New York: McGraw-Hill.

Richtmyer, R. D. (1957). *Difference Methods for Initial Value Problems*. New York: Wiley Interscience.

Sedgwick, R. (1992). *Algorithms in C++*. Reading, Mass.: Addison-Wesley. [Other versions available: *in C, in Pascal.*]

Shampine, L., and R. Allen (1973). *Numerical Computing*. Philadelphia: Saunders.

Smith, G. D. (1978). *Numerical Solution of Partial Differential Equations*. 2nd ed. London: Oxford University Press.

Stallings, William (1990). *Computer Organization and Architecture*. New York: Macmillan.

Stewart, G. W. (1973). *Introduction to Matrix Computations*. New York: Academic Press.

Stoer, J., and R. Bulirsch (1980). *Introduction to Numerical Analysis*. New York: Springer-Verlag.

Stubbs, D. and N. Webre (1987). *Data Structures with Abstract Data Types*. Monterey, Calif.: Brooks/Cole.

Traub, J. F. (1964). *Iterative Methods for the Solution of Equations*. Englewood Cliffs, N.J.: Prentice-Hall.

Varga, Richard (1959). p-Cyclic matrices: A generalization of the Young–Frankel successive overrelaxation scheme. *Pacific Journal of Mathematics* 9:617–628.

Vichnevetsky, R. (1981). *Computer Methods for Partial Differential Equations. Vol. 1: Elliptic Equations and the Finite Element Method*. Englewood Cliffs, N.J.: Prentice-Hall.

Waser, S., and M. J. Flynn (1982). *Introduction to Arithmetic for Digital Systems Designers*. New York: Holt, Rinehart and Winston.

Wilkinson, J. H. (1963). *Rounding Errors in Algebraic Processes*. Englewood Cliffs, N.J.: Prentice-Hall.

Wilkinson, J. H. (1965). *The Algebraic Eigenvalue Problem*. London: Oxford University Press.

Wolfram, Stephen (1988). *Mathematica: A System for Doing Mathematics by Computer*. Reading, Mass.: Addison-Wesley.

Gerald C. F. , P. O. Wheatley (1999) Applied Numerical Analysis. New york: Addison_Wesley