بسم الله الرّحمن الرّحيم

وزارت علوم، تحقیقات و فناوری
دانشگاه تربیت معلم آذربایجان
دانشکده علوم پایه

پایان نامه
جهت اخذ درجه کارشناسی ارشد
رشته ریاضی کاربردی

الگوریتم سیمپلکس پایه-ناقص برای مسائل برنامه‌ریزی خطی

استاد راهنما
دکتر علی خانی

استاد مشاور
دکتر بهروز خیرفام

پژوهشگر
سمیه فیضی

دی‌ماه ۱۳۸۹
تبریز - ایران

خدایا...

خدایا! منم همان بنده‌ی ناسپاست که در هر لحظه از زندگی‌اش تو را دید و انگار که ندید، صدایت را شنید و انگار که نشنید.
خدایا! منم همانی که نافرمانی‌ات هیچ غمی را در دلش نیافرید و از فرمان برداری‌ات برخودش بالید؛ چقدر لحظه هایی که فراموشت کرد و خیال کرد هر چه دارد از خودش است.
خدایا! من همانی هستم که نعمت و رحمت تو را به حساب شایستگی خودش گذاشت و در هنگام بلا و آزمایش، بی صبرانه به درگاه تو شکایت کرد و به یاد لحظه‌های فراموش شده‌ات افتاد.
خدایا! آری منم؛ ولی خود می‌دانی من هر چقدر بد بوده و هستم با تو صادق‌ام و این‌بار هم صادقانه و با دنیایی امید به مهر و محبت تو، برایت می‌گویم:

خدای من! کمکم کن تا با این داشته‌هایم بتوانم خدمتی آنچنان که تو خواهی به خلق تو داشته باشم و این بارگرانی که بر دوشم سنگینی می کند را در این راه ناتمام به سلامت به مقصد برسانم.

خدایا

چقدر دوست داشتنی و پرستیدنی هستی

تقدیم به

شهید گمنام آرمیده در دانشگاه

و شهدای راه عزت و علم

دکتر مسعود علی محمدی و دکتر مجید شهریاری

که سرخی خونشان نشانه‌ی راه است

و راه ناتمامشان بار گرانی بر دوش

سپاس گزاری...

خدایا! سپاست می گویم به خودت که مرا هستی دادی که تو را بشناسم؛
خدایا! سپاس تو را به قدرت فهمم که بخشیدی تا بفهمم که تو چقدر شایسته‌ی پرستیدن هستی؛ سپاس تو را به تمام نعمت‌هایی که ارزانی‌ام کردی تا من را به تو برسانند. سپاس تو را به پدر دلسوز و مادر مهربانم که وجودشان بارقه‌ای از محبت توست و بودنشان مایه‌ی تسکین و آرامشم. پروردگارم! تو را سپاس به تمام کسانی که کوشیدند تا مرا با الفبای زندگی و خوب زندگی کردن آشنا کنند؛ سپاس تو را به تمام معلمین و اساتیدم که مرا به فهمیدن و تفکر تربیت کردند و در این سال‌های متمادی تحصیل، از کودکی تا جوانی اکنونم، پله پله همراهم شدند تا بدانم که "علم" چقدر چیز خوبی است.
و ای رب من! تو را سپاس به تمام نعمت‌هایی که برایم دادی و از توان شمارش خارج است و سپاس تو را به تمام نعمت‌هایی که برایم ندادی تا در حکمتت تردید نکرده باشم.

اینک به پاس لطف الهی که پایان‌نامه‌ی حاضر، آماده شده است برخود واجب می‌دانم از حمایتها و مساعدت‌های استاد راهنمایم جناب آقای دکتر علی خانی سپاس‌گزاری نمایم و همچنین از جناب آقای دکتر بهروز خیرفام که مشاوره این پایان‌نامه را به عهده گرفتند، کمال تشکر را دارم و از آقای دکتر جعفر پورمحمود نیز سپاسگزارم که قبول زحمت فرموده و داوری این تحقیق را برعهده گرفتند.

سمیه فیضی
دی‌ماه ۱۳۸۹

چکیده

پایه‌ی استاندارد، که نقش اساسی در اصطلاح‌شناسی سیمپلکس بازی می‌کند، توسعه یافته تا شامل حالت ناقص (با تعداد ستون‌هایی کم‌تر از سطرها) شود. نتایج محاسباتی با عملیات متراکم برای این حالت مناسب بوده‌اند. در این پایان‌نامه، ما یک الگوریتم سیمپلکس اولیه را با استفاده از عامل‌های LU پایه‌ای ناقص، معرفی و پیشنهاد می‌کنیم. الگوریتم پیشنهادی، مسائل برنامه‌ریزی خطی جهان واقعی را، که اغلب تبهگن هستند، به‌صورت رضایت‌بخش‌تری در مقایسه با الگوریتم سیمپلکس، حل خواهد کرد.

الگوریتم پیشنهادی روی یک مجموعه از ۵۰ مسئله‌ی $Netlib$و هم یک مجموعه از ۱۵ مسئله‌ی خیلی بزرگ جهان واقعی، شامل ۸ مسئله‌ی $Kennington$ و ۵ مسئله‌ی $BPMPD$، اجرا و آزمایش شده است. آن به‌طور مشخص، ۵.۳ $MINOS$ را پشت سرگذاشته است. همچنین نتایج آشکار می‌کنند که هیچ رابطه‌ی حتمی بین ناکارآمدی الگوریتم و تبهگنی وجود ندارد.

کلمات کلیدی: برنامه ریزی مقیاس-بزرگ؛ الگوریتم سیمپلکس؛ تبهگنی اولیه؛ پایه-ناقص؛ فاکتورگیریLU.

فهرست مطالب

فهرست مطالب **ج**

چکیده . . . . . ح

۱ مباحثی از جبر خطی عددی **۱**

۱.۱ حل دستگاه خطی Ax=b با استفاده از فاکتورگیری ۲

۱.۱.۱ مقدمه ۲

۲.۱.۱ تجزیه LU ۴

۲.۱ تجزیه QR ۱۰

۳.۱ فرم هزنبرگ ۱۵

۲ یک روش پایه-ناقص توسعه یافته‌ی روش سیمپلکس برای برنامه‌ریزی خطی **۱۷**

۱.۲ مقدمه ۱۸

۱.۱.۲ عمومیت دادن پایه ۲۰

۲.۱.۲ دستورالعمل اولیه ۲۳

۳ سیمپلکس پایه‌ناقص اولیه برای مسائل برنامه‌ریزی خطی **۳۲**

۱.۳ مقدمه ۳۳

۲.۳ مفاهیم و تعاریف اصلی ۳۵

۳.۳ شرایط بهینگی ۳۸

۴.۳ جهت جستجو ۴۰

۱.۴.۳ استخراج ۴۱
۲.۴.۳ طرح جستجوی خط ۴۲
۵.۳ تشکیل الگوریتم ۴۴
۱.۵.۳ تکرار توسعه یافته ۴۴
۲.۵.۳ تکرار کامل ۴۵
۳.۵.۳ قرارداد ۴۷
۶.۳ الگوریتم ۴۷
۷.۳ امتحان کدها ۵۰
۱.۷.۳ نتایج برای مجموعه‌ی ۱ . . . . . ۵۱
۲.۷.۳ نتایج برای مجموعه‌ی ۲ . . . . . ۵۲
۳.۷.۳ نتایج تبهگنی ۵۵

کتاب‌نامه ۶۰

واژه‌نامه فارسی به انگلیسی ۶۲

واژه‌نامه انگلیسی به فارسی ۶۵

پیشگفتار

برنامه‌ریزی خطی با مسئله مینیمم‌سازی یا ماکزیمم‌سازی یک تابع خطی، با محدودیت‌های خطی در شکل مساوی و یا نامساوی سروکار دارد. یکی از معروفترین روش های حل مسئله‌ی برنامه‌ریزی خطی روش سیمپلکس است که در سال ۱۹۴۷ توسط جرج.بی.دانتزیک معرفی شد. تاکنون برنامه‌ریزی خطی بطور وسیع در ارتش، صنعت، حکومت، شهرسازی و سایر حوزه‌ها بکار گرفته شده است. ویژگی برنامه‌ریزی خطی از عوامل متعددی از جمله توانمندی مدل‌سازی مسائل بزرگ و پیچیده و نیز بکارگیری کامپیوترهای نسل جدید در حل مسائل بزرگ با استفاده از الگوریتم‌های کارا ناشی شده است.

از زمان ابداع روش سیمپلکس تاکنون افراد زیادی در پیشرفت برنامه‌ریزی خطی، در قالب تئوری ریاضی آن، معرفی روش‌ها و کدهای کارا، کشف الگوریتم‌های نوین و کاربردهای جدید و کاربرد برنامه‌ریزی خطی به عنوان ابزاری برای حل مسائل پیچیده، مثلاً برنامه‌های گسسته، برنامه‌های غیرخطی، مسائل ترکیباتی، مسائل برنامه‌ریزی احتمالی و مسائل کنترل بهینه مشارکت داشته‌اند.

وقتی تبهگنی در الگوریتم سیمپلکس رخ می‌دهد متغیرهای پایه با مقدار صفر می‌توانند به طول گام صفر در الگوریتم و در نتیجه ایجاد دور منتهی شوند. روش‌های محدودی برای اجتناب از دور وجود دارند اما این روش‌ها در عمل قابل رقابت با الگوریتم سیمپلکس نیستند و کارایی چندانی در جهان واقع ندارند. برای حل این مشکل و فرار از دور در هر زمان چاره‌های مختلفی بکار رفته است مخصوصا لیچنر، دانتزیک و دیویس یک " الگوریتم بهبود اکید برنامه‌ریزی خطی فاز ۱" که بدون هیچ توقفی پیش می‌رود، پیشنهاد کردند. تا اینکه در سال ۱۹۹۸ مفهوم پایه ناقص توسط پن [۱] معرفی شد که از تجزیه‌ی QR استفاده شده بود؛ سپس این الگوریتم با تجزیه‌ی LU اصلاح شد. پیاده‌سازی نسخه‌های جدولی چنین الگوریتم‌هایی در آزمایش‌های محاسباتی به‌صورت خیلی

[۱] Pan

مناسب اجرا شدند. بعلاوه استفاده از پایه‌های ناقص فقط برای الگوریتم‌های اولیه سودمند نیست بلکه برای موارد دوگان نیز سودمند است.

این پایان‌نامه الگوریتم جدیدی برای حل مسائل برنامه‌ریزی خطی را مورد بحث قرار می‌دهد که در سال ۲۰۰۸ توسط پَن معرفی شد؛ اساس این الگوریتم و وجه تمایز آن با الگوریتم سیمپلکس استاندارد، نوع ماتریس پایه بکار رفته در این روش و نحوه به‌روز رسانی این پایه است. در این پایان‌نامه تعریف جامعی از ماتریس پایه ارائه می‌شود که در آن لزوماً ماتریس پایه ماتریس مربعی نیست و می تواند مستطیلی باشد که چنین پایه‌ای پایه‌ی ناقص نامیده می‌شود. الگوریتم پیشنهادی در هر تکرار از تجزیه‌ی LU مستطیلی برای ماتریس پایه استفاده می‌کند و در نتیجه مسائل برنامه‌ریزی خطی را با اجرائیات و عملیات کم تراکم حل می‌کند. نتایج عددی که از پیاده کردن این الگوریتم روی یک مجموعه از ۵۰ مسئله‌ی $Netlib$ بدست آمده نشان می‌دهد که هیچ رابطه‌ی حتمی بین ناکارآمدی یک الگوریتم و تبهگنی وجود ندارد. در واقع مزیت دیگر این الگوریتم نسبت به الگوریتم سیمپلکس عدم ایجاد دور در طول الگوریتم و لذا حل مسائل تبهگن با استفاده از این الگوریتم است.

پایان‌نامه به این‌صورت سازمان یافته که در فصل بعد مباحث آنالیز عددی، که در طول مقاله‌ی اصلی مورد نیازند، بررسی می‌شوند. در فصل ۲ یک الگوریتم برای مسائل برنامه‌ریزی خطی آورده شده است که در آن از تجزیه‌ی QR استفاده شده‌است. در فصل ۳ نیز الگوریتم اصلی و تعاریف مربوط به آن و نیز نتایج عددی الگوریتم پیشنهادی مورد بررسی قرار گرفته است.

فصل ۱

مباحثی از جبر خطی عددی

فصل ۱

مباحثی از جبر خطی عددی

۱.۱ حل دستگاه خطی Ax=b با استفاده از فاکتورگیری

۱.۱.۱ مقدمه

دسته معادلات زیر را در نظر بگیرید

$$
\begin{aligned}
&a_{۱,۱}x_۱ + a_{۱,۲}x_۲ + \cdots + a_{۱,n}x_n = b_۱ \\
&a_{۲,۱}x_۱ + a_{۲,۲}x_۲ + \cdots + a_{۲,n}x_n = b_۲ \\
&\vdots \\
&a_{n,۱}x_۱ + a_{n,۲}x_۲ + \cdots + a_{n,n}x_n = b_n \quad (۱.۱)
\end{aligned}
$$

که در آن ضرایب a_{ij} و b_i $(i = ۱, ۲, ..., n)$ اعداد حقیقی معلوم و متغیرهای x_i مجهول‌اند. چنین دسته معادلات را دستگاه خطی [۱] گویند. این فصل برخی روش‌های عددی را برای تعیین

[۱] linear system

x_iها مورد بحث قرار می‌دهد.
تعداد عملیات الگوریتم‌های حل معادلات (۱.۱) می‌تواند با مرتبه n دستگاه به‌سرعت افزایش یابد و بنابراین در کاربرد آنها برای n بزرگ، انتخاب دقیق روش‌های عددی را می‌طلبد. دستگاه خطی (۱.۱) را به طور مختصر به‌صورت Ax=b می‌نویسیم که در آن $x, b \in R^n$ و همچنین $A \in R^{n\times n}$.
اگر A نامنفرد باشد، آنگاه دستگاه (۱.۱) برای هر بردار طرف راست $b \in R^n$ دارای جواب منحصر بفرد $x = A^{-1}b$ است. این عبارت صوری، شایستگی کاربردی مهمی را در خود دارد: محاسبه‌ی A^{-1} ، معکوس A، معمولا بیشتر از خود دستگاه برای یافتن جواب x، به عملیات محاسباتی نیاز دارد. شاید بتوان ویژگی این فصل را، به‌طور وسیع‌تر، مطالعه روش‌های تعیین x بدون محاسبه صریح A^{-1} دانست. روش‌های تعیین x به دو دسته مستقیم و غیرمستقیم تقسیم می‌شوند.

تعریف ۱.۱.۱. روش‌هایی که بعد از دنباله‌ای متناهی و از قبل مشخص شده‌ی عملیات محاسباتی، یک n تایی $\overline{x}$ از درایه‌های x را تولید می‌کنند، روش‌های مستقیم [۲] نامیده می‌شوند.
در مقابل، روش‌های غیرمستقیم [۳] یا تکراری [۴] قرار دارند که دنباله $x^{(m)}$ از بردار جواب‌های تقریبی را نتیجه می‌دهد. امید می‌رود که این دنباله، به معنی $\| x - x^{(m)} \| \to$ ۰ هرگاه $m \to \infty$، به بردار جواب صحیح همگرا باشد.

یک روش کلی به نام تجزیه به عامل‌ها یا فاکتورگیری [۵]، غالبا توسعه‌ی الگوریتم‌های عددی کارآمد را هدایت می‌کند. هدف از فاکتورگیری این است که تجزیه‌ی $A = BC$ ساخته شود و سپس حل $BCx = b$ در دو مرحله صورت می‌گیرد. ابتدا $z := Cx$ را به‌عنوان مجهول میانی تعریف کرده و سپس $Bz = b$ را برای تعیین z حل می‌کنیم. سپس مجهول اصلی x را با حل $Cx = z$ به دست می‌آوریم. این روند وقتی ارزش پیگیری دارد که بتوان B و C را طوری به‌دست آورد که دستگاه‌های مربوط به آن‌ها به‌سادگی حل شوند.

[۲] direct
[۳] indirect
[۴] iterative
[۵] factorization

قاعده کرامر [۶] روش مستقیمی برای حل دستگاه (۱.۱) است که به‌عنوان الگوریتم عددی، فوق‌العاده ضعیف است. محاسبه‌ی $n+1$ دترمینان از طریق بسط استاندارد به‌وسیله کهادها [۷]، به یک عدد نجومی $O((n+1)n!)$ از عملیات نیاز دارد. در حالیکه روش مستقیم با بیشترین تعداد عملیات، که روش حذف گاوس دارد، فقط $O(n^3)$ عملیات لازم دارد و البته با معرفی روش تجزیه به عامل‌ها در بخش‌های بعد، نشان خواهیم داد که تعداد عملیات موردنیاز روش تجزیه به عامل‌ها، که برگرفته شده از روش گاوس است، کمتر از این مقدار یعنی $O(n^3)$ است.

۲.۱.۱ تجزیه LU

همان طور که قبلا ذکر شد یکی از روش‌های مستقیم حل دستگاه $Ax = B$ روش تجزیه به عامل‌های ماتریس A است.
در بعضی از کاربردها حل چند دستگاه خطی با ماتریس ضرایب یکسان، اما بردار طرف راست متفاوت مانند

$$Ax_1 = b_1\ , \quad Ax_2 = b_2\ , \ \cdots$$

مورد نیاز است. در این گونه مسائل روش تجزیه به عامل‌ها مفید است.
در بخش‌های بعد سه روش تجزیه LU یعنی روش‌های دولتیل و کروت و روش چولسکی معرفی می‌شوند؛ در واقع روش چولسکی حالت خاصی از روش تجزیه‌ی LU است که در آن ماتریس A متقارن و معین مثبت است.
ماتریس پائین مثلثی $L \in R^{n\times n}$ را به‌صورت زیر تعریف می‌کنیم

$$L := \begin{bmatrix} 1 & & & \\ l_{2,1} & 1 & & \\ \vdots & & & \\ l_{n,1} & \cdots & l_{n,n-1} & 1 \end{bmatrix}$$

[۶] cramer's rule
[۷] minors

می‌توان ثابت کرد که $A = LU$. این تجزیه‌ی A را تجزیه به عامل‌های LU گوئیم. وقتی A نامنفرد باشد دارای یک تجزیه به عامل‌های منحصربفرد LU است که در آن درایه‌های قطری L صرفا عدد ۱ می‌باشد. تجزیه‌ی LU راه قویی را برای حل دستگاه‌های خطی عرضه می‌کند و روش حذف گاوس یک الگوریتم مناسبی را برای محاسبه‌ی L و U فراهم می‌کند. در این بخش به بررسی چند الگوریتم می‌پردازیم که تجزیه به عامل‌های مثلثی را تولید می‌کنند، در حالی‌که از بعضی از پیچیدگی‌های روش حذف گاوس به دورند. بخصوص روش‌های دولتیل و کروت[۸] را ارائه می‌کنیم که نیاز به ذخیره و بازیابی نتایج میانی را برای محاسبه‌ی L و U مرتفع می‌سازند.

روش های دولتیل و کروت

با فرض اینکه یک تجزیه‌ی LU از $A \in R^{n\times n}$ وجود دارد آنگاه با $n^۲$ رابطه‌ی

$$a_{i,j} = \sum_{p=۱}^{min\{i,j\}} l_{i,p}u_{p,j} \qquad (۲.۱)$$

می‌توانیم عامل‌های $l_{i,j}$ و $u_{i,j}$ را به‌دست آوریم. چون L و U پائین و بالا مثلثی‌اند روی هم فقط $n^۲ + n$ درایه‌ی غیرصفر دارند. بعلاوه، با این قرار که برای $i = ۱, ۲, \cdots, n$ ، $l_{i,i} = ۱$ لذا فقط $n^۲$ کمیت $l_{i,j}$ و $u_{i,j}$ باید محاسبه شود، بنابراین معادلات (۲.۱) برای این کار کافی است. بخصوص این‌که داریم

$$u_{i,j} = a_{i,j} - \sum_{j=۱}^{i-۱} l_{i,p}u_{p,j} \qquad j = i, i+۱, \cdots, n \qquad (۳.۱)$$

و برای $i = j+۱, j+۲, \cdots, n$

$$l_{i,j} = \frac{۱}{u_{j,j}}(\sum_{p=۱}^{j-۱} l_{i,p}u_{p,j}) \qquad (۴.۱)$$

[۸] Doolittle and Crout

در حالیکه معادلات (۳.۱) و(۴.۱) جالب به نظر نمی‌رسند، یک روش حل هوشمندانه، آن‌ها را به الگوریتم مفیدی به‌نام دولتیل تبدیل می‌کند.

الگوریتم دولتیل

ماتریس نامنفرد $A \in R^{n\times n}$ داده شده است؛ ماتریس‌های L و U متعلق به $R^{n\times n}$ در تجزیه LU ماتریس A به‌صورت زیر محاسبه می‌شوند:

$۱ - FOR \qquad k = ۱, ۲, \cdots n :$

$۲ - \qquad FOR \;\; j = k, k+۱, \cdots, n :$

$۳ - \qquad u_{k,j} \leftarrow a_{k,j} - \sum_{p=۱}^{k-۱} l_{k,p}u_{p,j}.$

$۴ - \qquad Next \; j.$

$۵ - \quad FOR \;\; i = k+۱, k+۲, \cdots, n :$

$۶ - \qquad l_{i,k} \leftarrow (a_{i,k} - \sum_{p=۱}^{k-۱} l_{i,p}u_{p,k})/u_{k,k}.$

$۷ - \quad Next \; i.$

$۸ - Next \;\; k.$

$۹ - End.$

(وقتی که $k = ۱$ مقدار ۰ را به مجموع‌های این الگوریتم نسبت می‌دهیم.)

به‌ازای تکرار kام، حلقه‌ی بیرونی جهت محاسبه‌ی این مرحله‌ی تکرار به مقادیر متغیرهای $l_{i,j}$ یا $u_{i,j}$ که باید در زیرحلقه‌های با تکرار شمار $k+۱$ و $k+۲$ و... محاسبه شوند، نیاز نیست. بعلاوه، اگر مجهولات را در این مرحله به همین ترتیبی که فهرست شده است به دست آوریم آنگاه تمام کمیت‌های طرف راست معادلات (۳.۱) و (۴.۱) در زمان نیاز به آن‌ها معلوم‌اند.

یک جایگزین برای روش دولتیل، روش کروت است که از یک ترکیب دیگر برای تعیین درایه‌های L و U استفاده می‌کند.

الگوریتم کروت

ماتریس $A \in R^{n\times n}$ داده شده‌است، ماتریس‌های $L \in R^{n\times n}$ و U در تجزیه‌ی LU ماتریس A، به‌صورت زیر محاسبه می‌شود :

۱ – $FOR\ j = ۱, ۲, ..., n :$

۲ – $\quad FOR\ i = ۱, ۲, .., j :$

۳ – $\quad u_{i,j} \leftarrow a_{i,j} - \sum_{p=۱}^{i-۱} l_{i,p}u_{p,j}.$

۴ – $\quad Next\ i.$

۵ – $\quad FOR\ i = j + ۱, j + ۲, ..., n :$

۶ – $\quad l_{i,j} \leftarrow (a_{i,j} - \sum_{p=۱}^{i-۱} l_{i,p}u_{p,j})/u_{j,j}.$

۷ – $\quad Next\ i.$

۸ – $Next\ j.$

۹ – $End.$

قابل ذکر است که تفاوت روش‌های دولتیل و کروت تنها در ترکیب به دست آوردن مولفه‌های ماتریس‌های L و U است.

تجزیه‌ی چولسکی

هرگاه $A \in R^{n\times n}$ متقارن و معین مثبت باشد، از روش کارآمد تجزیه‌ی مثلثی بنام تجزیه‌ی چولسکی می‌توان استفاده کرد. این روش A را به‌صورت حاصل‌ضربی به شکل CC^T تجزیه می‌کند که در آن $C \in R^{n\times n}$ پائین مثلثی و معروف به مثلث چولسکی[۹] برای A است. محل‌های ذخیره‌ی مورد نیاز این تجزیه، مانند تجزیه‌ی LU از یک ماتریس نامنفرد دلخواه در $R^{n\times n}$، بیش از تعداد مورد نیاز ذخیره‌ی ماتریس اصلی نیست. تجزیه‌ی A به CC^T به $(n^۲ + n)/۲$ محل ذخیره نیاز دارد که با تعداد موردنیاز ذخیره‌ی درایه‌های متمایز ماتریس اصلی A که متقارن می‌باشد،

[۹] cholesky triangle

یکسان است.

قضیه ۱.۱.۱. *اگر $A \in R^{n\times n}$ متقارن و معین‌مثبت باشد، آنگاه ماتریس پائین‌مثلثی $C \in R^{n\times n}$ وجود دارد که $A = CC^T$. بعلاوه تمام درایه‌های قطری C مثبت هستند.*

به منظور تهیه‌ی یک الگوریتم کاربردی برای محاسبه‌ی مثلث چولسکی C می‌توان مانند روش‌های دولتیل و کروت با تعیین درایه‌های $c_{i,j}$ از حل معادلات $a_{i,j} = \sum_{k=۱}^{n} c_{i,k}c_{j,k}$ ادامه داد.

الگوریتم چولسکی

فرض کنید $A \in R^{n\times n}$ معین مثبت و متقارن باشد. طی مراحل زیر درایه‌های $c_{i,j}$ مثلث چولسکی برای A محاسبه می‌شود:

$۱ - c_{۱,۱} \leftarrow \sqrt{a_{۱,۱}}$

$۲ - FOR\ i = ۲, ۳, ..., n :$

$۳ - \qquad c_{i,۱} \leftarrow a_{i,۱}/c_{۱,۱}$

$۴ - Next\ i.$

$۵ - FOR\ j = ۲, ۳, ..., n-۱ :$

$۶ - \qquad c_{j,j} \leftarrow (a_{j,j} - \sum_{k=۱}^{j-۱} c_{j,k}^{۲})^{۱/۲}.$

$۷ - \qquad FOR\ i = j+۱, j+۲, ..., n :$

$۸ - \qquad c_{i,j} \leftarrow (a_{i,j} - \sum_{k=۱}^{j-۱} c_{i,k}c_{j,k})/c_{j,j}.$

$۹ - \qquad Next\ i.$

$۱۰ - Next\ j.$

$۱۱ - c_{n,n} \leftarrow (a_{n,n} - \sum_{k=۱}^{n-۱} c_{n,k}^{۲})^{۱/۲}.$

$۱۲ - End.$

قضیه ۱.۱.۱ تضمین می‌کند که هیچ‌کدام از درایه‌های $c_{i,j}$ که به‌عنوان مخرج کسر در مرحله‌ی ۸ استفاده می‌شود، صفر نیست. بعلاوه، رابطه‌ی $a_{i,i} = \sum_{k=1}^{n} c_{i,k}^{2}$ نتیجه می‌دهد که $|c_{i,k}| \leqslant \sqrt{a_{i,i}}$.
بنابراین درایه‌های a بزرگی درایه‌های C را کنترل می‌کند.

مزایای روش LU

روش LU دارای مزیت‌هایی است که چند مزیت آن به شرح زیرند:

۱- برای ذخیره‌ی هر دو ماتریس L و U، به فضایی بیش از فضای مورد نیازِ ذخیره‌ی A احتیاج نیست؛ زیرا ذخیره‌ی درایه‌های به‌طور مشخص۰ یا ۱ضروری نیست.

۲- تجزیه به عامل‌های LU راه آسانی را برای محاسبه‌ی دترمینان A فراهم می‌سازد. زیرا $A = LU$ آنگاه $det(A) = det(L)det(U)$. اما دترمینان هر ماتریس بالا یا پائین مثلثی صرفا با حاصل‌ضرب درایه‌های قطری آن به‌دست می‌آید، بنابراین $det(L) = 1$.در نتیجه $det(A) = det(U) = u_{1,1}u_{2,2}...u_{n,n}$. در نتیجه، در مقابل روش معمولی محاسبه‌ی دترمینان یعنی با بسط دادن کهادها که $O(n!)$ عملیات لازم دارد، با استفاده از تبدیل سطری می توان $dat(A)$ را با $O(n^3)$ عملیات محاسبه کرد.

۳- در سیستم $Ax = b$، اگر b تغییر کند باز هم متغیر x را می‌توان به‌دست آورد. زیرا وقتی A به حاصل‌ضرب دو ماتریس L و U تجزیه شد، در آن‌صورت با یک جای‌گذاری ساده، مقدار متغیر x را می‌توان به‌دست آورد. به‌عبارت دیگر در این روش اساسی‌ترین قسمت کار، تجزیه A به حاصل‌ضرب دو ماتریس L و L بوده و این تجزیه مستقل از ماتریس b می‌باشد.

۴- به‌راحتی روش LU را می‌توان با دقت دوگانه و سه‌گانه محاسبه نمود که این خود دقت

زیادی را در نتایج به‌وجود می‌آورد.

۲.۱ تجزیه QR

تجزیه LU معرفی شده در بخش قبل، ماتریس A را به‌صورت $A = LU$ تجزیه می‌کند که در آن L از ساختاری خاص برخوردار است و U بالا مثلثی است. بویژه اینکه L پائین مثلثی با درایه‌های قطری واحد است. تفکری مشابه تجزیه‌ی QR را پشتیبانی می‌کند؛ ماتریس A را به‌صورت $A = QR$ تجزیه می‌کنیم که اکنون در اینجا R بالامثلثی و مجددا Q از یک ساختار خاصی برخوردار است.

تعریف ۱.۲.۱. یک ماتریس $Q \in R^{n\times n}$ متعامد است هرگاه $Q^TQ = I$.

یک ویژگی معادل در ماتریس متعامد $Q \in R^{n\times n}$ این است که ستون‌های آن تشکیل یک مجموعه‌ی متعامد یکه از بردارهای $q_1, q_2, ..., q_n$ متعلق به R^n داده باشد که q_iها ستون‌های Q هستند.
ماتریس‌های متعامد چندین خاصیت کلیدی دارند؛ برای هر بردار $x \in R^n$ داریم

$$\| Qx \|_2^2 = (Qx)^TQx = x^TQ^TQx = x^TIx = \| x \|_2^2$$

بنابراین به‌عنوان تبدیلات خطی، ماتریس‌های متعامد طول اقلیدسی را حفظ می‌کنند. از آن نتیجه می‌شود که تمام مقادیر ویژه‌ی یک ماتریس متعامد Q از مقدار مطلق واحد برخوردارند و بنابراین $|detQ| = 1$. می‌توان نشان داد که حاصلضرب دو ماتریس متعامد، متعامد است. در حقیقت مجموعه‌ی ماتریس‌های متعامد در $R^{n\times n}$ یک گروه تشکیل می‌دهند.
دو دسته از ماتریس‌های متعامد، بخصوص مفیدند. اولین دسته، ماتریس‌های چرخش هستند که در فضای R^2 به‌صورت زیرند:

$$Q = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}$$