برخی از روشهای حل معادلات خطی و برنامه‌ای برای حل معادلات تنک

محمد رضایی پژند

دانشیار دانشکده مهندسی دانشگاه فردوسی (مشهد)

چکیده

درباره مشخصات برنامه رایانه ای نوشته شده بحث می‌شود . برنامه مزبور توانایی انبارسازی ماتریس تنک با ساختمان غیر مشخص را دارا بوده و تنها درایه های ناصفر را در حافظه نگه داری و فقط آنها را وارد محاسبات خواهد نمود . افزون بر این ، پاره ای از روشهای حل معادلات خطی ارائه می‌گردد و در باره ویژگیهای آنها اظهار نظر خواهد شد .

پیش گفتار

بیشتر فنون عددی رایج از قبیل روش اجزای محدود (Finite Element Method) که در رشته های علوم و مهندسی رایج است به یک دستگاه معادلات خطی منجر می‌گردد [1] . از ویژگیهای مهم این دستگاه معادلات این است که ماتریس ضرایب آنها تنک (Sparse Matrix) ، معین مثبت (Positive Definite) و متقارن می باشد [2] و [3] . باید توجه نمود که ماتریس تنک دارای تعداد کمی درایه ناصفر بوده و ممکن است بیش از ۹۵٪ درایه های آن صفر باشند [4] . ازآنجایی که هزینه حل معادلات به دست آمده از تحلیل قسمت عمده‌ای از هزینه کل را تشکیل می‌دهد ، پرداختن به این نکته از اهمیت بسزایی برخوردار است . بدین سبب پژوهشگران این رشته تلاشهای زیادی برای ارائه روشهای مناسب از خود نشان داده‌اند [5] .

یادآوری می‌گردد که بطور معمول کلیه گرههای شبکه اجزای محدود و یا درجات آزادی آنها با هم ارتباط ندارند . این ویژگی سبب می‌شود تا درجات آزادی مزبور بر یکدیگر اثر نگذارند . بدین سبب

این مقاله در تاریخ ۷۱/۲/۲۶ دریافت شد و در تاریخ ۷۱/۵/۳۱ برای نشر در نشریه تصویب گردید .

درایه های مربوط به چنین درجات آزادی در ماتریس مشخصه صفر می‌گردند و تنها آن درایه هایی ناصفراند که بر یکدیگر اثر بگذارند . از آنجایی که بطور معمول درجات آزادی بسیار محدودی از شبکه اجزای محدود باهم ارتباط دارند ، بنابراین درایه های کمی از ماتریس مشخصه مربوطه ناصفر خواهند بود و در نتیجه چنین ماتریسی تنک می‌گردد .

یک دسته از فنون حل دستگاه معادلات خطی بر پایه روش حذفی گاس (Gaussian Elimination) استوار می‌باشند که به آنها روشهای مستقیم (Direct Methods) نیز گفته می‌شود . دسته دیگر به نام روشهای تکراری (Iterative Methods) معروف‌اند [6]. البته ترکیب این دو روش نیز ممکن می‌باشد [7] و [8] . شایان توجه است که روشهای مستقیم با انجام تعداد معینی عملیات محاسباتی به ریشه های دستگاه معادلات خطی می‌رسند. از سوی دیگر ، روشهای تکراری به طور معمول نیاز به حافظه کمی دارند و با تکرار چرخه مشخصی به تعداد مورد نیاز به حل دستگاه می‌پردازند [9] .

در سالهای اخیر روشهای متنوعی برای حل دستگاه معادلات خطی ارائه شده است . با وجود این که هر روش ویژگی خود را دارد ، با این حال ، سرعت در حل و نیاز به حافظه کم رایانه (Computer) از هدفهای اصلی فنون مزبور می‌باشد . افزون بر آن ، سادگی درک گامهای محاسباتی ، کاربرد آسان این گامها در برنامه سازی و دسترس بودن برنامه ها [10] از دیگر ویژگیهای مطلوب بوده اند. سرانجام ، سودجویی از ساختمان ماتریس ضرایب تنک و یا تغییر این ساختمان به سوی یک شکل خاص - که به مرتب سازی ماتریس تنک (Ordering Sparse Matrix) معروف است - از اهداف دیگر تحلیل گران این رشته می‌باشد [11] .

برخی از ساختمانهای ماتریس تنک مانند شکل نواری (Band Matrix) از اهمیت بالایی برخوردار است . این گونه ماتریسها به حافظه کمتری نسبت به ماتریس پرنیاز داشته و عملیات مورد نیاز حل چنین معادلاتی کم می‌باشد . شکلهای نیمرخی (Profile) و یا نوارگی متغیر از این جهت مطلوبتراند. با هدف استفاده بهینه از این گونه ساختمانهای ماتریس ضرایب، روشهای مناسبی تاکنون ارائه گردیده است [12] و [13] . ازسوی دیگر ، برای به کارگیری بهتر هسته اصلی (Main Core) رایانه و انبارسازی برخی از عوامل در خارج از هسته ، فنون حذفی دیگری همانند روش جبهه‌ای (Frontal Method) در

دسترس است [14]. به همین صورت تلاشهای زیادی در زمینه ارائه روشهای تکراری در سالهای اخیر صورت گرفته و می‌گیرد [15].

در این نوشته حذفی گاس و روشهای وابسته به آن به اختصار تشریح می‌گردد. تا حدامکان، نکات ضعف و قدرت هر روش و مشخصات برنامه سازی آنها درج خواهد شد. یک نکته ظریف این فنون پرشدن درایه های خالی (Fill-in) است که بطور خلاصه بدان پرداخته می‌گردد. چگونگی انبارسازی درایه های ماتریس ضرایب از دیگر نکات مورد بحث خواهد بود. از روشهای تکرای نیز سخن به میان می‌آید. در پایان یک روش تکراری و انبارسازی متناسب با آن اختیار شده و برنامه محاوره‌ای برای حل دستگاه معادلات خطی ارائه می‌شود.

حذفی گاس و انواع آن

درنیمه اول قرن نوزدهم گاس روشی برای حل دستگاه معادلات خطی ارائه نمود که نه تنها در سالهای پیشین، بلکه پس از رایج شدن رایانه ها در نیمه دوم قرن بیستم ارزش خود را دارابوده است [5]. روش حذفی گاس بسیار ساده و پر قدرت می‌باشد. سادگی آن بدین گونه است که براحتی قابل درک بوده، برنامه سازی آن آسان بوده و از همه مهمتر آن که استفاده از گامهای محاسباتی آن راحت خواهد بود. انواع مختلفی از روش حذفی گاس و برنامه های آن در دسترس تحلیل گران می‌باشد. روش حذفی گاس معمولی برای معادلاتی که دارای ماتریس ضرایب پر از درایه ناصفر و یا ماتریس ضرایب کوچک تنک -که به عنوان نمونه پنجاه سطر و یا کمتر دارند- به کار می‌رود. براین پایه، روش مزبور از ویژگی تنگی ماتریس ضرایب بهره نمی‌گیرد و نباید آن را برای دستگاه معادلات خطی تنک به کار برد.

بطور کلی در روش گاس در گام اول ماتریس ضرایب تجزیه می‌شود و به دنبال آن مجهولات میانی و نهایی به دست می‌آیند. گام نخست زمان برترین گام بوده و بیشترین نیاز به حافظه رایانه را دارد. برای نشان دادن چگونگی حذف ضرایب معادلات، یک دستگاه معادلات متقارن مطابق زیر در نظر گرفته و یک ضریب کلی a_{ik} از سطر i و ستون k حذف می‌گردد. نتیجه کار یک ضریب کلی و یک طرف دوم معادله می‌باشد که در زیر به نظر خوانندگان خواهد رسید:

$$[A]\{x\} = \{b\}$$

$$\begin{cases} \cdots \\ \cdots \quad a_{kk}x_k + \cdots + a_{kj}x_j + \cdots = b_k \\ \cdots \\ \cdots \quad a_{ik}x_k + \cdots + a_{ij}x_j + \cdots = b_i \\ \cdots \end{cases}$$

$$\begin{cases} \cdots \\ \cdots \quad a_{kk}x_k + \cdots + a_{kj}x_j + \cdots = b_k \\ \cdots \\ \cdots + 0 + \cdots + (a_{ij} - a_{ik}a_{kj} / a_{kk})\, x_j + \cdots = b_i-(a_{ik}b_k/a_{kk}) \\ \cdots \end{cases}$$

$$\begin{cases} a_{ij})_{new} = a_{ij})_{old} - a_{ik}a_{kj} / a_{kk})_{old} \\ b_i)_{new} = b_i)_{old} - a_{ik}b_k / a_{kk})_{old} \end{cases}$$

همانند عملیات حذفی که ارائه شد برای هریک از درایه های دیگر ممکن است . از سوی دیگر ، می‌توان بجای حذف ضرایب ناخواسته با تجزیه ماتریس متقارن [A] پرداخت و آن را به حاصل ضرب یک ماتریس پایین مثلثی با درایه قطری واحد ([L]) و یک ماتریس قطری ([D]) تبدیل نمود . با این کار معادلات خطی به سه دستگاه ساده تبدیل شده و دو دسته مجهول میانی({y} و {z}) در آنها وارد می‌شوند . در نتیجه ، حل سه دستگاه معادلات به دست آمده با توجه به ساختمان ماتریس ضرایب آنها بسادگی و مطابق زیر قابل انجام می‌باشند :

$$[A]=[L][D][L]^T$$

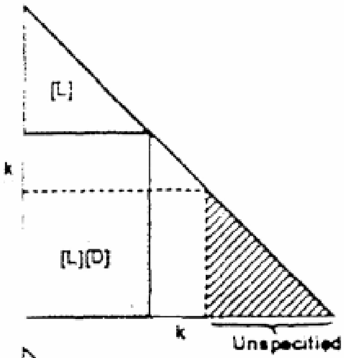
$$[L]\{y\}=\{b\}$$

$$[D]\{z\}=\{y\}$$

$$[L]^T\ \{x\}=\{z\}$$

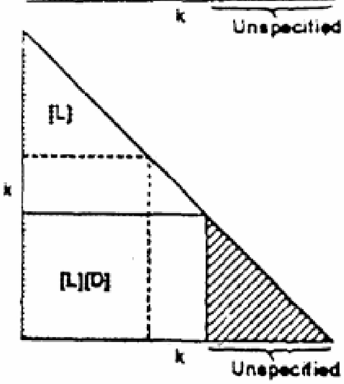
روش حذفی گاس چندین نوع دارد . یک دسته آنها به نام روش کروت (Crout) و یا گامهای محاسباتی ضرب خارجی (Outer Product Algorithm) موسوم است . یک ماتریس متقارن را می‌توان به دو گونه با روش کروت تجزیه کرد . شکل سطری و شکل ستونی این دو گونه را تشکیل می‌دهند . تجزیه به شکل سطری بدین ترتیب است که سطر کلی k از ماتریس بالا مثلثی [L] در گام k ام

محاسبه می‌گردد . یک ماتریس کمکی [M] - که در شروع تجزیه برابر با[A] اختیار می‌شود - در این روش مورد استفاده قرار می‌گیرد . در گام k ام ، سطر k از ماتریس [L] و ستون k ام ماتریس [L][D] محاسبه شده و جانشین سطر k ام و ستون k ام ماتریس [M] می‌شوند . با این حساب در طول تجزیه ، ماتریس [M] بخشهایی از [L] ، [L][D] و [A] را داراست [3]. معادلات و اشکال زیر مطلب ارائه شده را روشنتر می‌سازند :



$$m_{ji} = \begin{cases} 0 & j < i \\ 1 & j = i < k \\ l_{ji} & i < j < k \\ (ld)_{ji} & i < k\ ,\ j \geq k \\ \text{unspecified} & k \leq i \leq j \end{cases}$$

[M] Before Step



$$m_{ji} = \begin{cases} 0 & j < i \\ 1 & j = i \leq k \\ l_{ji} & i < j \leq k \\ (ld)_{ji} & i \leq k\ ,\ j > k \\ \text{unspecified} & k < i \leq j \end{cases}$$

[M] After Step

اینک شکل ستونی روش کروت به اختصار به نظر خوانندگان می‌رسد . در شکل مزبور ، ستون k ام ماتریس [L] در گام k ام تجزیه پیدا می‌گردد . به سخن دیگر ، در گام k ام ، ستون k ام [L] محاسبه شده و در ستون k ام ماتریس [M] در زیر قطر اصلی جا داده می‌شود . افزون بر آن درایه k ام ماتریس [D] محاسبه و انبار می‌گردد.

دسته دوم روش حذفی گاوس به نام چولسکی (Cholesky) و یا شکل ضرب داخلی (Inner Product Form) موسوم است [16]. در این روشها تجزیه به صورتی انجام می‌پذیرد که در گام k ام ، درایه k ام ماتریس قطری [D] و نیز سطر k ام ماتریس پایین مثلثی [L] محاسبه می‌شوند .

حل معادلات نواری

در رشته های علوم و مهندسی دستگاههای معادلاتی وجود دارند که درایه های ناصفرماتریس ضرایب آنها نزدیک به قطر اصلی می‌باشند و کمی دورتر از این قطر کلیه درایه های صفر جای گرفته اند . به عنوان نمونه ، در تحلیل ماتریسی سازه ها و یا روش اجزای محدود و هنگامی که سازه مستوی و یک شکل عادی دارد ، می‌توان گره‌ها را درجهت بعد کوچکتر سازه و به ترتیب شماره گذاری کرد . معادلات حاکم نتیجه شده از این شماره گذاری دارای ماتریس ضرایب به شکل نواری باریک خواهد بود . باید دانست که تعداد درایه ناصفر ماتریس مزبور مستقل از ترتیب شماره گذاری بوده ، لیکن چگونگی پخش آنها بستگی مستقیمی به این ترتیب دارد [1] و [17] . شایان توجه می‌باشد که تاکنون برای یک شکل کلی سازه قانون مشخص و محضی برای به دست آوردن کمترین نوار ممکن ارائه نشده است . با این حال روشهای تجربی (Heuristic Algorithm) گوناگونی که پاسخ تقریبی را پیدا می‌کنند در دسترس می‌باشند [18] .

حل معادلات نواری با روش حذفی گاس انجام می‌پذیرد . در حقیقت ، روش حل از درایه های درون نوار و یا نیم نوار در ماتریسهای متقارن استفاده کرده و درایه های خارج از این دامنه را در محاسبات وارد نمی‌کند . به سخن ساده تر ، عملیات حذفی فقط در محدوده نوار و یا نیم نوار انجام گرفته و تنها این درایه های محدود در حافظه رایانه انبار می‌گردند . از آنجایی که بطور معمول درایه های مزبور در قیاس با کل درایه‌های ماتریس ضرایب بسیار محدود هستند، بنابراین حل معادلات نواری بسیار ارزانتر از روش حذفی گاس معمولی می‌باشد. گامهای محاسباتی حل معادلات نواری همانند روش حذفی گاس ساده و قابل درک بوده و به آسانی می‌توان برای آنها برنامه سازی نمود . برنامه های آماده این روش به صورت حل در هسته اصلی (In-Core Solution) [13] و نیز به صورت حل در خارج از هسته (Out-of-Core Solution) در دسترس همگان می‌باشد [19]. در ضمن درمعادلات متقارن ویانامتقارن نواری که درایه های قطری غالب (Dominating Diagonal) در آنها موجود نیست ، می‌توان سطر و ستونهای ماتریس را تعویض نموده و تغییر لولا داد تا از نظر عددی پایدار گردد [20] . البته این کار پهنای نوار را تغییر داده و بطور معمول آن را افزایش می‌دهد .

یک هدف مشترک کلیه تحلیل گران انبار نمودن داده‌ها به صورت اقتصادی است . بر این اساس یا باید از حافظه اصلی که سریع نیز هست استفاده گردد و یا این که حافظه های آهسته جانبی مانند دیسک به کار گرفته شوند . روشهای گوناگونی برای انبار نمودن داده‌ها موجود است [21]. در ادامه به دو روش ساده انبار سازی ماتریس نواری اشاره می‌شود . هر دوروش درایه‌های نواری را به یک ماتریس مستطیلی انتقال می‌دهند . چگونگی انبار سازی درایه های نیم نوار ماتریس متقارن در این نوشته تشریح خواهد شد و با کمی تغییر می‌توان آنها را برای ماتریس نامتقارن به کار بست . روش نخست انبار نمودن بدین صورت عمل می‌کند که درایه های قطری به ستون اول ماتریس مستطیلی انتقال داده شده و درایه های موازی قطر در ستونهای بعدی این‌ماتریس جامی‌گیرند.در دومین روش ، درایه های قطری و درایه های موازی آن به صورت سطری به ماتریس مستطیلی انتقال داده می‌شود . اینک اگر فرض گردد که ماتریس نواری متقارن [A] با نیم نوار b در ماتریس مستطیلی [T] جا گرفته باشد ، می‌توان روابط مورد نیاز و چگونگی انبارسازی یک نمونه را مطابق زیر ارائه ساخت :

روش ۱ = $k = j - i + 1 , \quad t_{ik} = a_{ij}$

روش ۲ = $k = i + b - j , \quad t_{kj} = a_{ij}$

برای : $i = 1, 2, 3, \ldots, N$ و $j = i, (i+1), \ldots, (b+i-1) \leq N$

$$[A] = \begin{vmatrix} a_{11} & a_{21} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{32} & a_{42} & 0 \\ 0 & a_{32} & a_{33} & 0 & a_{53} \\ 0 & a_{42} & 0 & a_{44} & a_{54} \\ 0 & 0 & a_{53} & a_{54} & a_{55} \end{vmatrix} \qquad [T] = \begin{vmatrix} a_{11} & a_{21} & 0 \\ a_{22} & a_{32} & a_{42} \\ a_{33} & 0 & a_{53} \\ a_{44} & a_{54} & 0 \\ a_{55} & 0 & 0 \end{vmatrix}$$

$$[T] = \begin{vmatrix} 0 & 0 & 0 & a_{42} & a_{53} \\ 0 & a_{21} & a_{32} & 0 & a_{54} \\ a_{11} & a_{22} & a_{33} & a_{44} & a_{55} \end{vmatrix}$$

حل معادلات نیمرخی

یک گام جلوتر به سوی استفاده بهینه از انبار سازی و کاربرد درایه های ناصفر ماتریس ضرایب به کارگیری روشهای نیمرخی است . در حقیقت ، ماتریسهای نواری به وجود آمده از تحلیل مسایل عملی همواره دارای تغییراتی ازنظر تعداد درایه‌های ناصفر در سطرها می‌باشند. کاربرد روشهای حل معادلات نیمرخی به تحلیل گر این توانایی رامی‌دهد که از این تغییرات استفاده مطلوب کرده و کلیه سطرهای ماتریس نواری را به یک صورت انبار نساخته و تا حد امکان با درایه های صفر آخر هر سطر کار نکند . اولین روش حل نیمرخی در سال ۱۹۶۶ توسط جنینگز(Jennings) ارائه شده است [22] . او روش چولسکی را به کار برده و تنها درایه های داخل نیمرخ ماتریسهای ضرایب و تجزیه های آن را مورد توجه قرار داده است. در ادامه تلاشهای کارشناسان این زمینه روشهای آسمانخراش (Skyline) [23] . ستون فعال (Active Column) [24] و نوار متغییر (Variable Band) [25] کشف و ارائه شده اند.

به غیراز روش انبارسازی نیمرخی - که قدری پیچیده تر از ماتریس نواری است - سایر کارهای مورد نیاز حل همانند روش حذفی گاس می‌باشد. در واقع روش حذفی گاس با تغییراتی مورد استفاده حل دستگاه معادلات نیمرخی قرارمی‌گیرد. باید توجه داشت که روش نیمرخی نیاز به عملیات و حافظه کمتری نسبت به روش حل نواری دارد. بهترین بازده روش هنگامی است که نیمرخ فشرده بوده و پر از درایه‌های ناصفر باشد. با این حال ، گامهای محاسباتی و کاربرد داده ها و نیز انبار سازی و استفاده مجدد از آنها پیچیده تر از روش نواری می‌باشد .

اینک به یک فن انبار سازی ماتریس ضرایب و یا ماتریس اتصال (Connectivity Matrix) که در تحلیل ماتریسی سازه ها معمول است پرداخته خواهد شد [17] . در این روش درایه های ستونهای ماتریس مورد نظر انبار شده و درایه های صفر بالای هر ستون به کار نخواهند رفت . از ماتریس یک بعدی آسمانخراش استفاده خواهد شد ({T}) و ستون به ستون درایه های ماتریس مورد نظر ([A]) در آن جا داده می‌شود . در ضمن برای دانستن آغاز و پایان هر ستون باید از یک بردار نشانه ({IA}) سود برد. به سخن دیگر بردار مزبور وظیفه نگهداری نشانی درایه قطری را داشته و تعداد درایه ستون J ام با استفاده از حاصل عبارت IA(J+1)-IA(J) محاسبه می‌گردد . روابط مورد نیاز روش و یک نمونه

انبارسازی در ادامه به نظر خوانندگان خواهد رسید .

$j=1,2,3, \dots , N$

$i=j,(j-1),(j-2), \dots , [j+IA(j)-IA(j+1)+1]$

$k=IA(j),[IA(j)+1], \dots , [IA(j+1)-1]$

$t_k=a_{ij}$

$$[A]=\begin{vmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 & 0 \\ & a_{22} & 0 & a_{24} & 0 & a_{26} \\ & & a_{33} & a_{34} & 0 & a_{36} \\ & & & a_{44} & a_{45} & a_{46} \\ & & & & a_{55} & a_{56} \\ & & & & & a_{66} \end{vmatrix} = \begin{vmatrix} t_1 & t_3 & & t_8 & & \\ & t_2 & & t_7 & & t_{15} \\ & & t_4 & t_6 & & t_{14} \\ & & & t_5 & t_{10} & t_{13} \\ & & & & t_9 & t_{12} \\ & & & & & t_{11} \end{vmatrix}$$

$$\{IA\} = \{1,2,4,5,9,11,16\}$$

$$\{T\} = \{t_1,t_2,t_3,\dots,t_{15}\}$$

شایان توجه می‌باشد که تعداد متنوعی از روشهای حل معادلات نیمرخی دردست است. [26] و [27] . برخی از آنها دستگاه معادلات را در هسته اصلی رایانه حل نموده [23] و تعداد دیگری این کار را در خارج از هسته اصلی به انجام می‌رسانند [28] و [29] . باید دانست که دسترسی به انواع روشهای حل معادلات نیمرخی سبب گسترش کاربرد این فنون گشته و با وجود پیچیده‌تر بودن آنها نسبت به روشهای حل نواری ، تحلیل گران با چگونگی کارکرد و نیز کاربرد روشهای حل دستگاه معادلات خطی نیمرخی آشنا می‌باشند . گذشته از روشهای اشاره شده تاکنون - که برای معادلات متقارن کاربرد دارند - حل‌کننده نامتقارن نیمرخی نیز وجود دارد [30] .

روش جبهه‌ای

هرگاه روش حذفی گاس به گونه ای مورد استفاده قرار گیرد که تشکیل و حذف درایه های ماتریس ضرایب یکی پس از دیگری انجام پذیرد ، روش جبهه ای به کار گرفته شده است . هر ضریب

متغیر که پیش از حذف بخشی از آن و یا به طور کامل تشکیل شده باشد به نام متغیر فعال (Active Variable) موسوم می‌باشد . مجموع متغیرهای فعال جبهه ، پهنای جبهه (Frontwidth) و یا جبهه موج (Wavefront) خوانده می‌شوند .باید دانست روش حل جبهه ای معادلات نخستین بـار تـوسط پژوهشگران تحلیل سازه برای حل مسایل اجزای محدود به کار رفته است [31] . هنوز هـم بـه دلیـل سازگاری طبیعی روش جبهه ای با روش اجزای محدود [32] بیشتر در این زمینه کاربرد دارد ، هر چند که روش کلی بوده و می‌توان آن را برای هر نوع دستگاه معادلات به کار بست . این روش حتی برای حل دستگاه معادلات نامتقارن نیز به کار رفته است [33] و [34] و [35] .

در روش جبهه ای تجزیه ماتریس ضرایب به طور تدریجی انجام می‌پذیرد . برای تشریح روش مزبور فرض گردد که روش برای یک شبکه اجزای محدود به کار رود . در یک گام کلی هر درایه ماتریس ضرایب با سوارکردن یک جزء جدید تغییر می‌کند . در واقع ، اثر افزودن جزء جدید به سایر معادلات تشکیل شده تا این گام باید درنظر گرفته شود . می‌توان رابطه ریاضی نکات درج شده را به قرار زیر ارائه ساخت :

$$a_{ij})_{new} = a_{ij})_{old} + a_{ij})_{element}$$

باعمل سوار نمودن در هر گام تعدادی متغیر فعال به وجود می‌آید . این متغیرها که وابسته به درجات آزادی شبکه اجزای محدود می‌باشند ، یک جبهه‌ای درآن شبکه به وجود می‌آورند که همانند جبهه موجی با عمل سوارسازی در حرکت است . براین پایه ، جبهه مزبور بطور دایم تغییر خواهد نمود . در حقیقت ، جبهه نشانه اجزای سوار شده تا آن گام می‌باشد که هریک از این اجزاء یک یا چند درجه آزادی حذف نشده داشته باشند . اینک به عمل حذفی گاس اشاره می‌گردد که توانایی حذف درجـات آزادی به طور کامل سوار شده را دارا است . به عنوان نمونه فرض شود در گام sام درایه a_{ij} بطور کامل تشکیل شده باشد و بتوان آن را حذف کرد . در این صورت مطابق روش حذفی گاس هریک از درایه های کلی ماتریس ضرایب باید با توجه.به رابطه زیر تغییر نماید و اثر حذف شدن درایه مورد نظر را در بر گیرد.

$$a_{ij}^{(s+1)} = a_{ij}^{(S)} - a_{ik}^{(S)} \quad a_{kj}^{(S)} \quad / a_{kk}^{(S)}$$

بنابرآنچه که ارائه شد فقط پس از تشکیل کامل هر درایه عمل حذف انجام می‌شود و چنانچه این کار زودتر صورت گیرد اشتباه غیر قابل جبرانی وارد محاسبات می‌گردد . به سخن دیگر ، هر مجهول معادلات زمانی قابل حذف است که کلیه درایه های سطر و ستون متناظر آن بطور کامل تشکیل شده و سوار سازی اجزای باقی مانده هیچ گونه اثری در این درایه ها نداشته باشد . در هر گام فقط درایه های حذف نشده را در هسته اصلی نگه می‌دارند و سایر درایه‌ها را به خارج از هسته منتقل می‌سازند و برای محاسبات بعدی انبار می‌کنند . بدین جهت نیاز به حافظه اصلی در این روش کم بوده و می‌توان مسایل بزرگی را با آن حل کرد . در ضمن آن گونه که درجات آزادی در پهنای نوار ماتریس ضرایب معادلات موثر بود، در روش جبهه‌ای پهنای جبهه به شماره گذاری اجزاء وابسته می‌باشد. باید افزود که هر شماره گذاری اجزاء شبکه منجر به یک دسته پهنای جبهه شده و آن پهنایی که کمترین درجات آزادی را در طول حل معادلات به دست دهد بهترین شماره گذاری ممکن را دارد. زیرا هم حافظه مورد نیاز و هم تعداد عملیات لازم برای حذف با مربع پهنای جبهه متناسب است [36]. با این حساب ، تغییر شماره گذاری اجزاء و یافتن شماره گذاری بهینه در حل معادلات به روش جبهه ای بسیار با اهمیت است. برای این کار روش محضی تا کنون به دست نیامده ولی روشهای تجربی متفاوتی در دسترس می‌باشد [37] .

روش جبهه ای یک نوع تعمیم یافته روش نیمرخی است . اگر روش جبهه ای به صورت حل در هسته اصلی رایانه مورد استفاده قرار گیرد ، همانند روش نیمرخی عمل خواهد نمود . در حالت کلی روش جبهه ای را برای حل خارج از هسته اصلی به کار می‌برند و با این روش حافظه اصلی مورد نیاز را کاهش می‌دهند [38] . باید افزود با آن که روشهای نیمرخی و جبهه ای دارای تشابهاتی هستند ، با این حال انبار نمودن داده ها و سازمان دادن عملیات و برنامه سازی روش جبهه ای نیاز به مهارت زیادتری دارد و بسادگی روش نیمرخی نیست . یکی از نقاط قدرت روش جبهه ای مؤثر بودن آن در تحلیل شبکه های با اجزای مرتبه بالا است . در این اجزاء ، گره های میان پهلویی و داخلی موجود بوده که روش جبهه‌ای توانایی سوار نمودن و حذف سریع آنها را دارا می‌باشد .

دیگر روشهای حذفی

در سه دهه اخیر و بویژه در دهه ۱۹۸۰-۱۹۷۰ روشهای متعددی از حل دستگاه معادلات خطی بوسیله تحلیل گران ارائه شده است. برخی از این روشها در بخشهای پیشین مورد بررسی قرار گرفتند و در این بخش نیز تنها به دو روش دیگر اشاره می‌شود. نخستین روش که تعمیم یافته روش جبهه ای بوده و بوسیله اسپیلپنینگ (Speelpening) ارائه شده به نام روش جزءگسترش یافته (Generalized Element Method) معروف است. روش مزبور بطور معمول مسایل بزرگ اجزای محدود را به صورت خارج از هسته اصلی رایانه حل می‌نماید [39].

روش جزء گسترش یافته با داشتن ماتریسهای سختی در حافظه جانبی آغاز می‌شود. با سوار نمودن اجزای کنارهم، متغیرهای مشترک بین آنها را حذف می‌کند. در نتیجه این کار یک جزء گسترش یافته خلق می‌شود و سپس این جزء جدید جای جزء پیشین انبار می‌گردد. در ضمن نشانه های مورد نیاز عمل حذف در یک بایگان ذخیره می‌شوند. همانند این عملیات در گامهای بعدی اجرا می‌گردد و در آنها اجزا و یا اجزای گسترش یافته فشرده شده و جزء گسترش یافته جدیدی به دست می‌آید. در پایان کار یک جزء گسترش یافته به وجود خواهد آمد و کلیه متغیرهای آن حذف می‌شود.

باید در نظر داشت که روش جزء گسترش یافته همانند روش جبهه ای کار می‌کند، با این تفاوت که می‌تواند چندین ماتریس و یا جبهه فعال داشته باشد. البته در روش مزبور سوار نمودن اجزاء به اختیار بوده و مانند روش جبهه ای تنها با شماره‌گذاری اجزاء کار نمی‌کند. در ضمن گامهای محاسباتی، داده پردازی و برنامه سازی آن مشکلتر از روش جبهه ای است.

دومین روش حذفی مورد بحث بر ویژگی تنک بودن ماتریس ضرایب دستگاه معادلات استوار بوده و به روش ماتریس تنک موسوم است. باید افزود که روشهای زیادی به این نام معروف می‌باشند و در این جا تنها به یکی از آنها اشاره می‌گردد. همه این روشها کلی بوده و می‌توان از آنها در حل معادلات ماتریسی تنک استفاده کرد. ویژگی اصلی این روشها در این است که فقط درایه های ناصفر را انبار نموده و تنها آنها را در عملیات وارد می‌سازند. بدین سبب شایسته تر از روشهای نواری و نیمرخی عمل می‌نمایند. البته از آن روشها پیچیده تر بوده و درک گامهای محاسباتی، برنامه سازی و کاربرد آنها به مهارت بیشتری نیاز دارد [40]. یک تفاوت عمده این روش با فنون ارائه شده پیشین در این است که

بجای کار با ماتریس ضرایب معادلات ، با گراف وابسته (Associated Graph) به این ماتریس کار می‌نماید و رئوس (Vertex) این گراف را حذف می‌کند [41].

روش ماتریس تنک در سه گام معادلات را حل می‌نماید . تجزیه ماتریس در دو گام انجام می‌شود و مجهولات در گام سوم پیدا می‌گردند . دو گام انجام شده در تجزیه ماتریس به ترتیب تجزیه نشانه‌ای (Symbolic Factorization) و تجزیه عددی نام دارند [42]. در تجزیه نشانه ای فقط ساختمان ماتریس پایین مثلثی پیدا شده و مشخص می‌شود که چه درایه هایی ناصفراند . در گام دوم با توجه به محل درایه های ناصفر مقدار عددی آنها نیز حساب می‌شود . سرانجام گام سوم مطابق روش حذفی گاس معمولی عمل کرده و مقدار عددی مجهولات پیداخواهد شد .

در پایان این بخش به یک روش انبارسازی درایه های ماتریس تنک پرداخته می‌شود [43]. در این روش فقط درایه های ناصفر ماتریس ضرایب در یک بردار یک بعدی ({A}) نگه‌داری می‌گردند . یک بردار نشانه ستون ({JA}) - که با مشخص نمودن ستون درایه ناصفر هر سطر محل آن را به دست می‌دهد - نیز به کار می‌رود. تعداد درایه های بردارهای {A} و {JA} باهم مساوی و هر دو برابر با تعداد درایه ناصفر ماتریس ضرایب هستند. افزون بر اینها ، به یک بردار یک بعدی دیگر ({IA}) نیاز می‌باشد تا آغاز و پایان درایه های ناصفر هر سطر را مشخص سازد. اگر سطرها به ترتیب و پشت سرهم انبار شوند با استفاده از این بردار و محاسبه حاصل IA(I+1)-IA(I) می‌توان درایه ی ناصفر سطر i ام را نیز یافت. باید دانست که تعداد درایه بردار {IA} یک عدد بیشتر از تعداد سطر ماتریس ضرایب است. اکنون روابط این انبار سازی و یک نمونه ماتریس انبار شده به نظر خوانندگان می‌رسد.

$$IA(i) = k \quad , \quad JA(k) = J \quad , \quad A(k) = a_{ij}$$

$$\begin{matrix} a_{11} & 0 & a_{13} & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & a_{34} & 0 \\ a_{41} & 0 & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{matrix}$$

	1	2	3	4	5	6	7	8	9	10	11	12
IA	1	3	4	7	11	13						
JA	1	3	2	1	3	4	1	3	4	5	4	5
A	a_{11}	a_{13}	a_{22}	a_{31}	a_{33}	a_{34}	a_{41}	a_{43}	a_{44}	a_{45}	a_{54}	a_{55}

پرشدن درایه‌های خالی

هنگامی که ماتریس ضرایب به ماتریسهای مثلثی تجزیه می‌گردد ، بسیار اتفاق می‌افتد که در محل درایه صفر ماتریس ضرایب ، یک درایه ناصفر در ماتریس مثلثی به وجود آید . باید توجه داشت که بطور معمول ماتریس مثلثی از همان حافظه ماتریس ضرایب استفاده می‌کند . در مورد روش گاس معمولی چون کلیه درایه های ماتریس ضرایب انبار شده ، بنابراین جای کافی برای درایه های پر شده موجود است. حال یک ماتریس ضرایب نواری را در نظر بگیرید ، اگر آن وارون گردد ، ماتریس وارونه دیگر شکل نواری نداشته و درایه های ناصفر درهمه جا پخش می‌شوند [4] . در این حالت اگر فقط درایه های درون نوار در حافظه نگه داری شده باشند و همین محلها برای ماتریس وارون بکار رود اشکال بزرگی پیش می‌آید و نتیجه عملیات غلط خواهد بود . با توجه به این نمونه اهمیت پر شدن درایه های خالی روشن می‌شود. بدین ترتیب توصیه شده است که چنانچه نیاز دیگری نباشد دستگاه معادلات به یکی از روشهای دیگر حل گردد و از وارون ماتریس ضرایب استفاده نشود . در ضمن افزون بر میزان حافظه مورد نیاز ، حل دستگاه معادلات با روش وارون سازی وقت گیرتر از حل معادلات به روشهای ارائه شده دیگر نیز است .

اینک پرشدن درایه های خالی در روشهای نواری و نیمرخی مورد توجه قرار می‌گیرند . یک ویژگی خوب روش حذفی گاس در ماتریس نواری این است که پدیده پرشدن درایه های خالی فقط در درون نوار ماتریس اتفاق می‌افتد و نه در بیرون آن . بنابراین با انبار سازی کلیه درایه های درون نوار و عملیات تنها بر روی آنها مسأله ای ازاین جهت به وجود نمی‌آید . البته هرچه نوار پرتراز درایه های ناصفر بوده و یا به سخن دیگر نوار باریکتر باشد کار حل با سرعت بیشتر پیشرفت نموده و حافظه کمتری مورد نیاز است . باید افزود که چنانچه با عملیات سطری و ستونی جای برخی از سطرها و ستونهای ماتریس نواری عوض شود ، وضعیت نوار و پرشدن درایه های خالی تغییر خواهد نمود .

پرشدن درایه های خالی در روش نیمرخی نیز همانند روش نواری است . در این روش فقط درایه‌های پر شده در درون نیمرخ قرار گرفته و در بیرون آن به وجود نمی‌آیند . بدین دلیل کلیه درایه های صفر و ناصفر درون نیمرخ انبار شده و از جاهای خالی این نیمرخ برای پرشدن درایه ها استفاده می‌گردد [44]. با این حساب ، هرچه نیمرخ فشرده‌تر باشد روش حل نیمرخی مناسبتر خواهد بود .

باتوجه به نکات ارائه شده مشخص می‌شود که هر چند انبارسازی وعملیات تنها برروی درایه های ناصفر بسیار اقتصادی به نظر می‌رسد ولی چنانچه پدیده پرشدن درایه های خالی در روشهای حذفی چاره سازی نگردد مانع از رسیدن به پاسخ درست معادلات خواهد شد . در حقیقت اگر روشهای حل معادلات تنک سعی خود را به کمینه سازی مقدار پرشدن درایه های خالی نکنند، این روشها ممکن است با تعداد زیادی درایه‌های خالی که پر شده‌اند اقتصادی تر از روشهای دیگر حل دستگاه معادلات نباشند . با کمال خوشبختی یک دسته روشهای حل معادلات خطی در دست است که مسأله پرشدن درایه های خالی را ندارند . نام این دسته " روشهای تکراری " بوده و در بخش بعدی تشریح می‌شوند .

روشهای تکراری

هر تحلیلی که با استفاده از روش تفاوتهای محدود (Finite Difference) و یا روش اجزای محدود انجام پذیرد به دستگاه معادلاتی منجر می‌شود که دارای ماتریس ضرایب تنک بوده و درایه های ناصفر کمی دارد . معادلاتی که از مسایل عملی نتیجه می‌گردند دارای مجهولات فراوانی اند . افزون برآن ، جهت دادن تحلیل به سوی این هدف که ماتریس ضرایب دارای ساختمان مشخصی - مانند نوار باریک و یا نیمرخ فشرده - باشد در مسایل بزرگ کمتر عملی و یا حداقل مشکل است . در چنین مسایلی می‌توان از روشهای تکراری استفاده نمود .

در روشهای تکراری با تخمین اولیه جوابها و تکرار عملیات به جوابهای بهتری می‌رسند. در این راستا می‌توان حل تکراری معادلات را به گونه ای انجام داد که فقط درایه های ناصفر ماتریس ضرایب انبار شده و در عملیات شرکت نمایند . بر این پایه نیاز به حافظه کمی بوده و زمان حل متناسب با تعداد درایه ناصفر و نه کل درایه می‌باشد . باید دانست که در روشهای حذفی تعدادعملیات و زمان مورد نیاز حل قابل پیش‌بینی بوده ، در صورتی که در روشهای تکراری این موارد به تخمین اولیه جوابها و نوع معادلات دارد و نمی‌توان از پیش آنها را معین ساخت .

دو روش بسیار معروف تکراری حل معادلات به نامهای ژاکوبی (Jacobi) و گاس - سایدل (Gauss-seidel) موسوم هستند . در روش ژاکوبی از معادله اول مجهول اول ، از معادله دوم مجهول دوم و به همین صورت از معادله iام مجهول iام و تا آخر ... را برحسب سایر مجهولات می‌نویسند .

سپس جوابهای تخمینی را در سمت راست قرار داده و مجهولات حساب می‌شوند . در ادامه جوابهای به دست آمده را دوباره در معادلات قرار داده و جوابهای بهتری را حساب می‌کنند . این کار آنقدر ادامه می‌یابد تا جوابهای معادلات با تقریب مناسبی پیدا گردند . اگر کار به گونه ای انجام شود که بدون درنگ پس از محاسبه هر جواب از آن در جایگزینی در معادله بعدی استفاده گردد روش سریعتری به نام گاس - سایدل به دست خواهد آمد . چنانچه معادلات به صورتی مرتب شوند که درایه های قطری ناصفر داشته باشند و میزان خطای مجاز دو تکرار متوالی E باشد ، روابط زیر جوابها و شرط همگرایی گام (i+1) ام را با روش گاس - سایدل به دست می‌دهد :

$$x_j^{i+1}=(b_j - \sum_{l=1}^{j-1} a_{jl}x_l^{i+1} - \sum_{l=j+1}^{n} a_{jl}x_l^{i}) / a_{jj}$$

$$E > | x_j^{i+1} - x_j^{i} | / | x_j^{i+1} |$$

باید توجه داشت که روشهای تکراری برای معادلات مشخصی کاربرد دارند و نمی‌توان برای هرنوع دستگاه معادلات خطی از آنها استفاده نمود . در حقیقت ، روشهای مزبور فقط برای معادلاتی که درایه های قطری غالب ناصفر دارند همگرا می‌باشند . به عنوان نمونه ، کلیه معادلاتی که دارای ماتریس ضرایب معین مثبت هستند با این روشها قابل حل می‌باشند . شایان توجه است که بیشتر تحلیلهای سازه انجام شده به روشهای ماتریسی و اجزای محدود منجر به چنین ماتریسهایی خواهد شد . نکته دیگر آن که می‌توان با کاربرد ضریب تسریع همگرایی(B) زودتر به نتایج مطلوب رسید . این ضریب به نوع معادلات بستگی دارد و بین ۱/۳ تا ۱/۹ پیشنهاد شده است [45] . اکنون رابطه تسریع در همگرایی به صورت زیر نوشته خواهد شد :

$$x_j^{i+1} = x_j^{i} + B(x_j^{i+1} - x_j^{i})$$

نمونه های متعددی از کاربرد روشهای تکراری در دست می‌باشند [46]. برخی از پژوهشگران

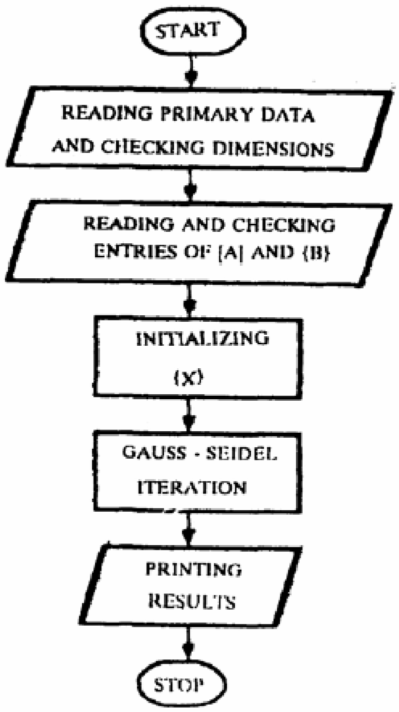
روشهای حذفی و تکراری را ترکیب نموده و از آن در تحلیل عددی مسایل سود جسته اند [8] . باید دانست که روش گاس - سایدل برای تحلیلهای مکرر و نیز تحلیل بهینه سازه ها مفید است . در این مسایل تنها تغییر جزیی در سازه در هرگام تحلیل به وجود می‌آید و بنابراین می‌توان جوابهای گام پیشین را برای تکرار به کار گرفت و با سرعت به پاسخ جدید دست یافت . یکی از کارهای انجام شده اخیر در این زمینه توسط بهروش و کاوه ارائه شده است [47]. آنها تأکید نموده اند که حل معادلات سازه های بزرگ با روش حذفی نیاز به حافظه زیاد دارد که فراهم سازی آن در همه شرایط امکان پذیر نیست . افزون برآن ، اگر رایانه با ظرفیت کافی نیز در دسترس باشد ، به سبب بزرگ بودن ماتریس ضرایب ، پاسخهای به دست آمده با روش حذفی در اثر خطاهای ناشی از گرد شدن دقیق نخواهد بود . براین اساس پژوهشگران مزبور از روش تعمیم یافته گاس - سایدل استفاده کرده اند . در روش ارائه شده بهروش و کاوه ، نخست سازه به زیر سازه هایی (Substructures) پاره سازی گردیده و سپس این بخشها با روش حذفی تحلیل شده اند . از پاسخهای به دست آمده برای مقادیر اولیه روش تکراری استفاده گردیده است . باید توجه نمود که روش مزبور حالت کلی کانی و کراس می‌باشد .

نظر به نکات ارائه شده در این نوشته محدود ، روشن است که روشهای تکراری و به خصوص حل تکراری گاس - سایدل دارای توانایی خوبی می‌باشد . بویژه اگر از آن در حل معادلات با ماتریس ضرایب تنک که دارای درایه های قطری غالب هستند استفاده شده و ضریب تسریع همگرایی نیز به کار برده شود . براین پایه ، برنامه محاوره ای تهیه شده که در ادامه مشخصات آن به نظرخوانندگان خواهد رسید. برنامه مزبور از روش انبارسازی درایه های ماتریس تنک (با ساختمان نامشخص) استفاده کرده و فقط درایه های ناصفر را در حافظه نگه داری نموده و تنها آنها را وارد عملیات می‌سازد . افزون برآن ، میزان خطای مجاز ریشه های معادلات قابل وارسی بوده و می‌توان پاسخهای مورد نظر را با دقت مطلوب به دست آورد . در ضمن ، برنامه ارائه شده به صورت دسته ای از زیر روالهای (Subroutines) مجزا نوشته شده تا علاقه‌مندان قسمتهای مورد نظر را بسادگی و بدلخواه تغییر داده و ازآن برای حل مسایل مناسب - به عنوان نمونه در تحلیل به روش اجزای محدود مسایل بزرگ و یا تحلیلهای مکرر و یا تحلیل بهینه سازه های بزرگ - سود جویند

در باره برنامه

دو زبان برنامه سازی بیش از دیگران در تحلیل سازه ها به کار رفته است . نخست زبان فرترن (FORTRAN) بوده که اولین زبان پیشرفته محاسباتی دنیا نیز می‌باشد . زبان مزبور بسیار سریع اجرا می‌شود و نسخ مختلف آن بسیار همانند یکدیگر است . دومین زبان برنامه سازی معمول ، زبان بیسیک (BASIC) می‌باشد . سادگی این زبان سطح پایین باعث فراگیرشدن آن شده است . با وجود این ، کندی اجرا و نیز تفاوتهای زیاد نسخ متعدد این زبان از نقاط ضعف آن می‌باشد [48]. باید توجه داشت که مقالات فراوانی در نیمه دوم قرن بیستم میلادی انتشار یافته که درآنها برنامه سازی با زبان فرترن انجام پذیرفته است . براین پایه ، برنامه مورد بحث به زبان فرترن خواهد بود .

در نسخه پیشین این نوشته صورت برنامه ارائه شده بود . بنابه پیشنهاد داوران بخش مزبور حذف گردید . علاقه مندان می‌توانند برای دسترسی به صورت برنامه با نویسنده تماس بگیرند . در ادامه نمودار جریان برنامه و ساده یک مثال به نظر خوانندگان می‌رسد .



```
*****EXAMPLE NO.1*****
NUMBER OF EQUATIONS, N=   3
NUMBER OF NON-ZERO ENTRIES IN [A], NNZ = 9
MAXIMUM NUMBER OF ITERATIONS, MIT =  300
ACCEPTANCE ERROR, ERROR =   .0000100
CONVERGENCE SPEEDING FACTOR, CSF  = 1.80000
MAXIMUM NUMBER OF DIMENSION IN PROGRAM, MDIM  =  10000
NUMBER OF DIMENSION REQUIRED FOR THIS PROBLEM, NDIMR = 32
IROW =   1    JCOL =  1  AIJ =    10.00000
IROW =   1    JCOL =  2  AIJ =     2.00000
IROW =   1    JCOL =  3  AIJ =     1.00000
IROW =   2    JCOL =  1  AIJ =     1.00000
IROW =   2    JCOL =  2  AIJ =    20.00000
IROW =   2    JCOL =  3  AIJ =    -1.00000
IROW =   3    JCOL =  1  AIJ =     1.00000
IROW =   3    JCOL =  2  AIJ =    -1.00000
IROW =   3    JCOL =  3  AIJ =    30.00000
B( 1 )  =     .1700000E +02
B( 2 )  =     .3800000E +02
B( 3 )  =     .8900000E +02
RESULTS ARE:
X( 1 )  =     .9999966E +00
X( 2 )  =     .2000002E +01
X( 3 )  =     .3000002E +01

NUMBER OF ITERATIONS USED, NIT  =  53
```

نتیجه

اهمیت و روشهای حل دستگاه معادلات خطی در حد امکان این نوشته تشریح شد. باید دانست حجم مطالعات انجام شده در این زمینه آنقدر گسترده است که نمی توان جزئیات کلیه روشها را یک جا ارائه ساخت. باوجود این، برخی از روشهای حذفی و تکراری مورد بحث قرار گرفته و ویژگیهای فنون

مختلف به نظر خوانندگان رسید. باتوجه به توانایی خوب روش تکراری گاس - سایدل در حل دستگاه معادلات با ماتریس ضرایب تنک، از آن در تهیه یک برنامه محاوره ای استفاده شد. برنامه مزبور می‌تواند دستگاه معادلات خطی که ماتریس ضرایب معین و مثبت دارند و درایه های قطری آنها غالب است - راحل نماید. از ضریب تسریع در همگرایی نیز سود برده شده است.

برنامه ارائه شده دارای ویژگیهای مناسبی است. یک دستور ابعادی ثابت دارد که می‌توان در صورت نیاز بسادگی آن را تغییر داد. سایر دستورهای ابعاد متغیر می‌باشند. پیکره برنامه از چندین زیر روال تشکیل شده است که می‌توان با تغییر مختصری آنها را برای حل مسایل عملی مانند تحلیل به روش اجزای محدود، تحلیلهای مکرر و یا تحلیل بهینه سازه های بزرگ، آماده ساخت. روش حل نیاز به حافظه بسیار کمی دارد و تنها با درایه های ناصفر کار می‌کند. هیچ گونه درایه خالی در این روش حل پر نخواهد شد. سرانجام آن که برنامه مزبور بسادگی قابل استفاده است و می‌توان دقت محاسبات را وارسی نموده وبه همگرایی شتاب داد.

مراجع

[1] Thomas J.R. Hughes, The Finite Element Method, Linear Static and Dynamic Finite Element Analysis, Prentice Hall International Editions, Englewood Cliffs, New Jersey (1987).

[2] Alan George, " Sparse Matrix Aspects of the Finite Element Method ", pp.3-22, in *Computing Methods in Applied Science and Engineering,* (R. Glowinski and J.L. Lions, Eds.), Springer-Verlag (1976).

[3] D.J. Rose, G.G. Whitten, A.H. Sherman and R.E. Tarjan , " Algorithms and Software for In-core Factorization of Sparse Symmetric Positive Definite Matrices ", *Computers & Structures* , Vol. 11, pp. 597-1608 (1980).

[4] R.K. Livesley, Mathematical Methods for Engineers, John Wiley & Sons, New York (1989).

[5] Christian Meyer, "Solution of Linear Equations State of-the-Art", *Journal of the Structural Division, ASCE* Vol. 99, No. ST7, pp. 215-246 (1984).

[6] Joan Walsh, "Direct and Indirect Methods", pp.41-55, in *Large Sparse Sets of Linear Equations*, (J. K. Reid, Ed.), Academic Press (1971).

[7] A. Behravesh nad A. Kaveh, "Iterative Solution of Large Structures", *Computers & Structures* ,Vol.35, No, 3, pp. 279-282 (1990).

[8] Edward L. Wilson, " Solution of Sparse Stiffness Matrices for Structural Systems ", pp. 1-24, in *Sparse Matrix Proceedings 1978* , (I.S. Duff and G.W., Stewart, Eds,), SIAM (1979).

[9] A. Samuelsson, N-E. Wiberg and L. Bernspang, "A Study of the Efficiency of Iterative Methods for Linear Problems in Structural Mechanics", *International Journal for Numerical Methods in Engineering* , Vol. 22, pp. 209-218 (1986).

[10] Iain S. Duff, "A Survey of Sparse Matrix Software", pp. 165-199, in *Sources and Development of Mathematical Software* ,(Wayne R. Cowell, Ed.), Printice-Hall inc. (1984).

[11] M. Rezaiee-Pajand and G.T. Will, "Efficiently Ordering Sparse Matrix Equations", Department of Civil Engineering, University of Toronto, Tornoto, Ontario M5S1A4, Canada, (Summer 1990).

[12] Iain S. Duff, "A Survey of Sparse Matrix Research", pp. 500-535, in *Proceeding of the IEEE*, Vol. 65, No, 4, (April 1977).

[13] Christian Meyer, "Special Problems Related to Linear Equation Solvers", *Journal of the Structural Division, ASCE* , Vol. 101, No ST4, pp. 869-890, (April 1975).

[14] Iain S.Duff, "Recent Developments in the Solution of Large Sparse Linear Equations", pp. 407-426, in *Computing Methods in Applied Science and Engineering* , (R.Glowinski and J.L. Lions, Eds), North-Holland Publishing Company, INRIA (1980).

[15] David J. Evans, "Iterative Methods for Sparse Matrices ", pp.45-111, in *Sparsity and its Applications*, (D.J. Evans.Ed.), Cambridge University Press (1985).

[16] Kincho H.Law and Steven J. Fenves, "Sparse Matrices, Graph Theory and Reanalysis", pp. 234-249, in *Proceedings of the First International Conference on*

Computing in Civil Engineering, ASCE (1981).

[17] John F. Fleming, "Computer Analysis of Structural Systems", McGraw-Hill Book Company, New York (1989).

[18] A. Kaveh. "An Algebraic Graph Theoretical Method for Bandwidth Optimization" *International Journal of Engineering*, Iran University of Science and Technology, Vol. 1, No. la, pp. 41-49 (1990).

[19] Edvard L. Wilson, Klaus-Jurgen Bathe and William P. Doherty, "Direct Solution of Large System of Linear Equations", *Computers & Structures*, Vol.4, pp. 363-372 (1974).

[20] Alan George, "A Survey of Sparse Matrix Methods in the Direct Solution of Finite Element Equations", pp. 15-20, in *Proceedings of the 1973 Summer Computer Simulation Conference*, Montreal, P.Q., Canada.

[21] Udo W. Pooch and Al Nieder, "A Survey of Indexing Techniques for Sparse Matrices", *Computing Surveys*, Vol. 5, No. 2, pp. 109-133 (1973).

[22] A. Jennigs. "A Compact Storage Scheme for the Solution of Symmetric Linear Simultaneous Equations", *Computer Journal,9,* pp.281-285 (1966).

[23] Carlos A. Felippa, "Solution of Linear Equations With Skyline-Stored Symmetric Matrix", *Computers* & Structures, Vol. 5,pp. 13-29 (1975).

[24] R. L. Taylor, E. L. Wilson and S.J. Sackett, "Direct Solution of Equations by Frontal and Variable Band, Active Column Methods", pp. 521-552, in *Nonlinear Finite Element Analysis in Structural Mechanics*, (W. Wudnerlich, E. Stein and K-J. Bathe, Eds.), Springer Verlag (1981).

[25] A. Jennigs, "Solution of Variable Bandwidth positive Definite Simultaneous Equations" , *Computer journal* , Vol. 14, pp. 466 (1971).

[26] Digambar p. Mondkar and Graham H. Powell, "Towards Optimal In-core Equation Solving" , *Computers & Structures,* Vol. 4, pp. 531-548 (1974).

[27] Digambar p. Mondkar and Graham H. Powell, "Large Capacity Equation Solver for Structural Analysis" , *Computers & Structures* , Vol. 4, pp. 699-728 (1974).

[28] A. Jennigs and A.D. Tuff, "A Direct Method for the Solution of Large Sparse Symmetric Simultaneos Equations" , in *Large Sparse Sets of Linear Equations* (J.K. Reid, Ed.) , Academic press (1971).

[29] E.L. Wilson and H.H. Dovey, "Solution or Reduction of Equilibrium Equations for Large Complex Structural Systems" , *Advances in Engineering Sofrware* , Vol. 1, No. 1, pp. 19-25 (1978).

[30] Eliezer Mendelsshon, "Solution of Linear Equations With a Symmetrically Skyline-Stored Nonsymmetric Matrix", *Computers & Structures* , Vol. 18, No.2, pp. 215-246 (1984).

[31] Bruce M. Irons, "A Frontal Solution program for Finite Element Analysis", *International Journal for Numetical Methods in Engineering* , Vol. 2, pp. 5-32, (1970).

[32] Syed F. Abbas , "Some Novel Applications of the Frontal Concepts",*International Journal for Numerical Methods in Enggineering,* Vol. 15, pp. 519-536,(1980).

[33] P. Hood, "Frontal Solution Program for Unsymmetric Matrices" , *International*

Journal for Numerical Methods in Engineering, Vol. 10, pp. 379-399(1976).

[34] R.J. Collins , " A Modified Prefrontal Routine" , *International Journal for Numerical Methods in Enggineering*, Vol. 11, No.4, pp. 765-766 (1977).

[35] P. Hood , " Note on Frontal Solution Program for Unsymmetric Matrices" , *International Journal for Numerical Methods in Engineeriing*, Vol. 11, pp. 1055 (1977).

[36] J.K. Reid," Frontal Methods for Solving Finite Element Systems of Linear Equations", pp. 265-281, in *Sparse Matrices and Their Uses*, (I.S. Duff, Ed.), Academic Press (1981).

[37] S. W. Sloan and W. S. Ng, "A Direct Comparison of Three Algorithms for Reducing Profile and Wavefront " , *Computers & Structures* , Vol . 33, No. 2 , pp. 411-419 (1989).

[38] Yoji Shimazaki , " Frontal - Skyline Method for Unsymmetric Matrices", *Proc . of JSCE Structural Eng./Earthquake Eng.* , Vol. 2, No. 1, pp. 101-108 (April 1985).

[39] B. Speelpenning , " The Generalized Element Method " , Department of Computer Science , Yniversity of I11inois , Urbana , Champaign , I11inois , Report UIUCDC-R-78-946 (1978).

[40] I.S. Duff , " A Sparse Future" , pp. 1-29 , in *Sparse Matrices and Their Uses*, (I.S. Duff, Ed .), Academic Press (1981).

[41] Kincho H. Law , " Sparse Matrix Factor Modification in Structural Reanalysis ". *International Journal for Numerical Methods in Engineering*, Vol . 21, pp. 37- 63

(1985).

[42] F.G. Gustavson, " Some Basic Techniques for Solving Sparse Systems of Linear Equations" , pp. 41-52, in *Sparse Matrices and Their Appliations,* Plenum Press (1972).

[43] S. C. Eisenstat , M.H. Schultz and A. H. Sherman , " Considerations in the Design of Software for Sparse Gaussian Elimination ", pp. 263-273, in *Sparse Matrix Computations,* (James R. Bunch and Donald J. Rose, Eds.), Academic Press Inc. (1976).

[44] Alan George and Wai-Hung Liu , "A Note on Fill for Sparse Matrices", *SIAM J. Numer . Anal.* , Vol. 12, No. 3, pp. 452-455 (1975).

[45] Klaus - Jurgen Bathe , Finnite Element Procedures in Engineering Analysis, Prentice- Hall Inc ., Englewood Cliffs , New Jersey (1982).

[46] David M. Young, " Iterative Solution of Linear Systems Arising from Finite Element Techniques" , pp. 439-464 , in *The Mathematic of Finite Element and Application* (MAFELAP 1975) , (J.R. Whiteman, Ed.), Academic Press (1976).

[47] Alaodin Behravesh and Ali Kaveh, " Direct-Iterative Method for Analysis of Larger Structures , *International Journal of Engineering* , Iran University of Science and Technology, Vol. 1, No. 1b, pp. 1-10 (1990) (In Persian).

[48] James A.D. Balfour , Computer Analysis of Structural Frameworks , Nichols Publishing Company, New York , (1986).