УДК 519.61

Неполная столбцово-строчная факторизация матриц в итерационных методах Крылова решения больших систем линейных уравнений

Саух С.Е.
Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины
saukh@svitonline.com

Abstract

Saukh S. Incomplete column-row factorization of matrices in iterative Krylov methods for solving large-scale linear systems. *Incomplete column-row factorization method is proposed. This method does not require permutations of rows and columns in submatrices. Calculable stability of the method is provided by search of pivots in submatrices to minimize divergence on the Frobenius norm between the transformed and regenerate submatrices. Significance of entries of the factor matrices is specified by comparison of norms of rows and columns of transformed and subtracted submatrices. The new method requires less memory than AINV, ILUT and RIF-Ns methods.*

Введение

Для формирования предобусловливателей предложен метод неполной столбцово-строчной (*ICR*–) факторизации несимметричных матриц. Метод не требует перестановок строк и столбцов в субматрицах. Поэтому получаемые факторные матрицы не являются треугольными. Вычислительная устойчивость метода обеспечивается применением оригинальной процедуры поиска ведущих элементов в субматрицах. Поиск выполняется в ограниченном множестве строк с наименьшим количеством ненулевых элементов. Выбор ведущего элемента осуществляется по критерию минимума произведения частичных норм фактор-строки и фактор-столбца, не содержащих самого элемента. Такой выбор ведущего элемента на текущем шаге факторизации позволяет минимизировать расхождение по норме Фробениуса между преобразуемой и преобразованной субматрицами, что обеспечивает устойчивость вычислений для плохо обусловленных матриц. В методе *ICR*–факторизации применена оригинальная оценка малозначимости элементов факторных матриц, основанная на сопоставлении норм строк и столбцов преобразуемых и вычитаемых субматриц. Поскольку каждая вычитаемая субматрица представляется произведением фактор-строки и фактор-столбца, то такое сопоставление норм позволяет ввести обобщенную оценку значимости элементов образуемых факторных матриц и интерпретировать пренебрегаемые элементы как ошибки конечноразрядных вычислений. Преимущества предложеного метода *ICR*–факторизации над методом *ILU*–факторизации и его модификациями иллюстрируются примерами решения тестовых систем уравнений с использованием итерационного метода проекций решений на подпространства Крылова GMRES(m).

Метод CR– факторизации матриц

Основой метода CR–факторизации [1] является последовательность действий, выполняемых над заданной $n\times n$ матрицей $\boldsymbol{A}$ в соответствии с формулами

$$\begin{aligned}
&\boldsymbol{A}_{(-i_1,-j_1)} = \boldsymbol{A} - \boldsymbol{C}_{j_1}\cdot \boldsymbol{R}_{i_1},\\
&\qquad \boldsymbol{A}_{(-i_1,-j_1)}(i_1,:) = 0,\ \boldsymbol{A}_{(-i_1,-j_1)}(:,j_1) = 0,\\
&\qquad i_1 \in \{1,2,\ldots,n\},\\
&\qquad j_1 \in \{1,2,\ldots,n\};\\
&\boldsymbol{A}_{(-i_2,-j_2)} = \boldsymbol{A}_{(-i_1,-j_1)} - \boldsymbol{C}_{j_2}\cdot \boldsymbol{R}_{i_2},\\
&\qquad \boldsymbol{A}_{(-i_2,-j_2)}(i_2,:) = 0,\ \boldsymbol{A}_{(-i_2,-j_2)}(:,j_2) = 0,\\
&\qquad i_2 \in \{1,2,\ldots,n\} \cap i_2 \ne i_1,\\
&\qquad j_2 \in \{1,2,\ldots,n\} \cap j_2 \ne j_1;\\
&\quad\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad (1)\\
&\boldsymbol{A}_{(-i_n,-j_n)} = \boldsymbol{A}_{(-i_{n-1},-j_{n-1})} - \boldsymbol{C}_{j_n}\cdot \boldsymbol{R}_{i_n} = 0,\\
&\qquad i_n \in \{1,2,\ldots,n\} \cap i_n \notin \{i_k \mid k=1,2,\ldots,n-1\},\\
&\qquad j_n \in \{1,2,\ldots,n\} \cap j_n \notin \{j_k \mid k=1,2,\ldots,n-1\}.
\end{aligned}$$

Например, в результате разложения 6×6 матрицы $\boldsymbol{A}$ относительно второй строки $i_1 = 2$ и четвертого столбца $j_1 = 4$, имеем

$$\boldsymbol{A}_{(-2,-4)} = \boldsymbol{A} - \boldsymbol{C}_4 \cdot \boldsymbol{R}_2 =$$

$$= \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16}\\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26}\\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36}\\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46}\\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56}\\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{vmatrix} - \begin{vmatrix} c_{14}\\ c_{24}\\ c_{34}\\ c_{44}\\ c_{54}\\ c_{64} \end{vmatrix} \times$$

$$\times\begin{vmatrix} r_{21} & r_{22} & r_{23} & r_{24} & r_{25} & r_{26}\end{vmatrix} = \begin{vmatrix} \times & \times & \times & 0 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & \times & \times \\ \times & \times & \times & 0 & \times & \times \\ \times & \times & \times & 0 & \times & \times \\ \times & \times & \times & 0 & \times & \times \end{vmatrix},$$

где элементы $\{c_{i4} | i = 1,2,\ldots,6\}$ и $\{r_{2j} | j = 1,2,\ldots,6\}$ столбца $\boldsymbol{C}_4$ и строки $\boldsymbol{R}_2$ легко определяются по соответствующим элементам исходной матрицы $\boldsymbol{A}$, причем, элементы c_{24} и r_{24}, удовлетворяющие равенству $c_{24} \cdot r_{24} = a_{24}$, находятся в предположении, что либо $c_{24} = 1$ и $r_{24} = a_{24}$, либо $r_{24} = 1$ и $c_{24} = a_{24}$.

Поскольку в формируемых матрицах $\boldsymbol{A}_{(-i_k, -j_k)}$ столбцы j_k и строки i_k становятся нулевыми, то итоговая матрица будет $\boldsymbol{A}_{(-i_n, -j_n)} \equiv 0$. Из соотношений (1) следует равенство

$$\boldsymbol{A} = \sum_{k=1}^{n} \boldsymbol{C}_{j_k} \boldsymbol{R}_{i_k} = \boldsymbol{CR} \qquad (2)$$

где матрицы $\boldsymbol{C}$ и $\boldsymbol{R}$ состоят из столбцов $\{\boldsymbol{C}_{j_k} | j_k = 1,2,\ldots,n\}$ и строк $\{\boldsymbol{R}_{i_k} | i_k = 1,2,\ldots,n\}$ соответственно.

Соотношения (1) являются обобщением известных формул факторизации. Установив значения индексов разложения $i_k = k$ и $j_k = k$, приходим к формулам $LU-$факторизации, определяющим нижне- и верхнетреугольную факторные матрицы $\boldsymbol{L} = \boldsymbol{C}$ и $\boldsymbol{U} = \boldsymbol{R}$. Если же в соотношения (1) – (2) ввести групповые индексы $\{i_k\}$ и $\{j_k\}$, а также соответствующие им блочные строки $\left|\{\boldsymbol{R}_{i_k}\}\right|^T$ и столбцы $\left|\{\boldsymbol{C}_{j_k}\}\right|$, то получим формулы обобщенной блочной факторизации. В частности, установив значения групповых индексов $\{i_k\} = \{k, n-k+1\}$ и $\{j_k\} = \{k, n-k+1\}$, получим формулы $QI-$факторизации [2], которые определяют блочные строки $\left|\boldsymbol{R}_k, \boldsymbol{R}_{n-k+1}\right|^T$ и столбцы $\left|\boldsymbol{C}_k, \boldsymbol{C}_{n-k+1}\right|$. Введение в (1) – (2) групповых индексов переменной структуры $\boldsymbol{var}\{i_k\}$ и $\boldsymbol{var}\{j_k\}$, позволяет определить метод факторизации не только с фиксированными, но и с переменными размерами блочных строк $\left|\boldsymbol{var}\{\boldsymbol{R}_{i_k}\}\right|^T$ и столбцов $\left|\boldsymbol{var}\{\boldsymbol{C}_{j_k}\}\right|$.

Основное преимущество метода $CR-$факторизации над существующими методами заключается в адаптивности соотношений (1) – (2) к позиционированию выбираемых ведущих элементов. Заметим, что использование специальных схем хранения разреженных матриц сопряжено с необходимостью выполнения множества неарифметических операций для получения доступа к матричным элементам. Поэтому отсутствие в методе $CR-$факторизации перестановок строк и столбцов приводит к существенному сокращению объема вычислений, в среднем на треть [1].

Однако применение метода $CR-$факторизации, как и других методов разложения разреженных матриц на множители, сопровождается непредсказуемым увеличением количества ненулевых элементов в факторных матрицах $\boldsymbol{C}$ и $\boldsymbol{R}$ относительно их количества в исходной матрице $\boldsymbol{A}$. В условиях жестких ограничений вычислительных ресурсов непредсказуемые требования к объемам памяти в ряде случаев не могут быть удовлетворены. Поэтому прибегают к методам неполной $CR-$факторизации ($ICR-$факторизации) матриц для построения предобусловливателей $\boldsymbol{A} = \tilde{\boldsymbol{A}} + \boldsymbol{Д} = \tilde{\boldsymbol{C}}\tilde{\boldsymbol{R}} + \boldsymbol{Д}$ с ошибкой $\boldsymbol{Д}$. Факторизованные матрицы-предобусловлеватели $\tilde{\boldsymbol{A}}$ используются в итерационных методах проекций решений на подпространства Крылова для ускорения сходимости последовательности решений к точному решению систем алгебраических уравнений вида $\boldsymbol{AX} = \boldsymbol{B}$ [3]. Особенности построения метода $ICR-$факторизации матриц и обсуждение результатов его тестирования рассматриваются ниже.

Особенности методов неполной факторизации матриц

Существующие методы неполной факторизации матриц основаны на различных подходах к отбрасыванию части ненулевых элементов факторных матриц [3-5].

Наиболее легко реализуемым является подход, который состоит в сохранении шаблона $P\langle\boldsymbol{A}\rangle$ размещения ненулевых элементов исходной матрицы $\boldsymbol{A}$ в шаблонах $P\langle\tilde{\boldsymbol{C}}\rangle$ и $P\langle\tilde{\boldsymbol{R}}\rangle$ размещения наиболее значимых элементов матриц $\tilde{\boldsymbol{C}}$ и $\tilde{\boldsymbol{R}}$ [3]. При таком подходе соблюдаются три условия:

$$P\langle\tilde{\boldsymbol{C}}\rangle \subset P\langle\boldsymbol{A}\rangle \text{ и } P\langle\tilde{\boldsymbol{R}}\rangle \subset P\langle\boldsymbol{A}\rangle;$$

$$\forall (i,j) \in P\langle\boldsymbol{A}\rangle: \ \left[\tilde{\boldsymbol{C}}\tilde{\boldsymbol{R}}\right]_{ij} = [\boldsymbol{A}]_{ij};$$

$$P\langle\boldsymbol{A}\rangle \cap P\langle\boldsymbol{Д}\rangle = 0.$$

Введенное для случая $\boldsymbol{L} = \boldsymbol{C}$ и $\boldsymbol{U} = \boldsymbol{R}$ приближенное представление $\boldsymbol{A} \approx \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$ является неполной $LU-$факторизацией матрицы $\boldsymbol{A}$ или $ILU(0)-$факторизацией с нулевым заполнением. Если допускается расширение шаблона $P\langle\tilde{\boldsymbol{L}} + \tilde{\boldsymbol{U}}\rangle$

относительно шаблона $P\langle \boldsymbol{A}\rangle$ в результате учета p дополнительных ненулевых элементов в каждой строке матрицы $\tilde{\boldsymbol{U}}$ и в каждом столбце матрицы $\tilde{\boldsymbol{L}}$, то в этом случае выполняется $ILU(p)-$ факторизация. Достоинством $ILU(p)-$ факторизации, где $p \geq 0$, является предсказуемость требований к объемам памяти, необходимой для размещения факторных матриц. Однако бесконтрольное проникновение ошибок в матрицы $\tilde{\boldsymbol{L}}$ и $\tilde{\boldsymbol{U}}$ снижает аппроксимационные свойства матрицы-предобусловливателя $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$, особенно в случаях плохой обусловленности исходной матрицы $\boldsymbol{A}$. Поэтому $ILU(p)-$ факторизация используется в алгоритмах решения систем уравнений, не имеющих особенностей.

Более сложным в реализации является подход, основанный на различных оценках значимости ненулевых элементов факторных матриц $\boldsymbol{L}$ и $\boldsymbol{U}$, что позволяет ограничить влияние отбрасываемых элементов на проникновение ошибок в матрицу-предобусловливатель вида $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$.

В современных методах неполной $LU-$ факторизации таких, как $ILUT$ [3], $ILUC$ [4], $AINV$ [6], $RIF-Ns$ [5], построение предобусловливателей основано на разложении вида $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{L}}\boldsymbol{D}\tilde{\boldsymbol{U}}$ содержащем диагональную матрицу $\boldsymbol{D}$. Процесс факторизации сопровождается пренебрежением теми ненулевыми элементами l_{jk} матрицы $\boldsymbol{L}$ и u_{kj} матрицы $\boldsymbol{U}$, чьи значения удовлетворяют условиям

$$\left|l_{jk}\right|\left\|e_k^T \boldsymbol{L}^{-1}\right\| \leq \tau, \qquad \left|u_{kj}\right|\left\|\boldsymbol{U}^{-1} e_k\right\| \leq \tau, \quad (3)$$

где $\tau-$ априори задаваемый приемлемый уровень потерь, e_k- вектор с единичным $k-$м елементом. Выполняемая при все меньших значениях τ $ILU(\tau)-$ факторизация матрицы $\boldsymbol{A} = \tilde{\boldsymbol{L}}\boldsymbol{D}\tilde{\boldsymbol{U}} + \boldsymbol{Д}$ позволяет уменьшить ошибку $\boldsymbol{Д}$ до приемлемого уровня, а в предельном случае $\tau = 0$ получить $\boldsymbol{Д} = \boldsymbol{0}$, то есть выполнить факторизацию в полном объеме. Однако при таком способе регулирования ошибки $\boldsymbol{Д}$ наблюдается существенный рост ненулевых элементов в матрицах $\tilde{\boldsymbol{L}}$ и $\tilde{\boldsymbol{U}}$, а шаблон $P\langle \tilde{\boldsymbol{L}} + \tilde{\boldsymbol{U}}\rangle$ становится отличным от шаблона $P\langle \boldsymbol{A}\rangle$. Поэтому $ILU(\tau)-$ факторизация, как правило, дополняется условием, ограничивающим заполненность матриц $\tilde{\boldsymbol{L}}$ и $\tilde{\boldsymbol{U}}$ ненулевыми элементами в количестве $nz(\tilde{\boldsymbol{L}} + \tilde{\boldsymbol{U}})$ таком, что

$$\frac{nz(\tilde{\boldsymbol{L}} + \tilde{\boldsymbol{U}})}{nz(\boldsymbol{A})} \leq \gamma. \quad (4)$$

Априори задаваемое значение параметра γ фактически устанавливает границы заполненности факторных матриц ненулевыми элементами, наибольшими среди не удовлетворяющих условию (3). Таким образом, выполняемая $ILU(\tau,\gamma)-$ факторизация обеспечивает возможность построения предобусловливателя $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{L}}\boldsymbol{D}\tilde{\boldsymbol{U}}$ матрицы $\boldsymbol{A}$ с хорошими аппроксимационными свойствами, что ускоряет сходимость итерационных методов проекций решений на подпространства Крылова [3-5].

Основная трудность реализации методов $ILU(\tau,\gamma)-$ факторизации заключается в оценке матричных норм $\left\|e_k^T \boldsymbol{L}^{-1}\right\|$ и $\left\|\boldsymbol{U}^{-1} e_k\right\|$, входящих в условия (3). Из последовательно формируемых столбцов и строк матриц $\tilde{\boldsymbol{L}}$ и $\tilde{\boldsymbol{U}}$ непосредственно получить такую оценку не представляется возможным. Поэтому $ILU(\tau,\gamma)-$ факторизация матрицы $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{L}}\boldsymbol{D}\tilde{\boldsymbol{U}}$ совмещается с $AINV(\tau,\gamma)-$ факторизацией обратной матрицы $\boldsymbol{A}^{-1} = \boldsymbol{Z}\boldsymbol{D}^{-1}\boldsymbol{W}$, выполняемой в неполном виде $\tilde{\boldsymbol{A}}^{-1} = \tilde{\boldsymbol{Z}}\boldsymbol{D}^{-1}\tilde{\boldsymbol{W}}$, где $\boldsymbol{Z}$ и $\boldsymbol{W}$ верхне- и нижнетреугольная матрицы, а $\tilde{\boldsymbol{Z}}$ и $\tilde{\boldsymbol{W}}$ – аппроксимирующие их матрицы [6]. Алгоритмы совместной $ILU(\tau,\gamma)-$ и $AINV(\tau,\gamma)-$ факторизации требуют дополнительных ресурсов оперативной памяти для размещения матриц $\tilde{\boldsymbol{Z}}$ и $\tilde{\boldsymbol{W}}$ одновременно с матрицами $\tilde{\boldsymbol{L}}$ и $\tilde{\boldsymbol{U}}$. В наиболее рациональных алгоритмах осуществляется формирование и размещение в памяти одной из матриц $\tilde{\boldsymbol{Z}}$ и $\tilde{\boldsymbol{W}}$ [4, 6]. Однако требование дополнительных ресурсов оперативной памяти является существенным недостатком подобных алгоритмов.

Кроме того, алгоритмы неполной факторизации несимметричных матриц часто не могут обеспечить приемлемой аппроксимации факторных матриц в ограниченной оперативной памяти без предварительного упорядочения строк и столбцов исходной матрицы $\boldsymbol{A}$. Поскольку такое упорядочение не учитывает условий вычислительной устойчивости процесса факторизации, то неполная факторизация упорядоченных матриц осуществляется в условиях контроля за близостью ведущих элементов к машинному нулю. Те ведущие элементы, чьи значения не превышают машинный нуль, искусственно корректируются с тем, чтобы обеспечить устойчивость вычислительного процесса в ущерб его точности. Искусственная коррекция ведущих элементов вносит искажения в факторные матрицы, особенно существенные для плохо обусловленных матриц. Получаемые в результате предобусловливатели часто оказываются непригодными для итерационного решения линейных систем алгебраических уравнений с особенностями.

Таким образом, современные методы неполной факторизации матриц обладают двумя существенными недостатками – в них отсутствуют процедуры выбора ведущих элементов для обеспечения вычислительной устойчивости алгоритма и значительно завышены требования к использованию ресурсов оперативной памяти. Таких недостатков лишен метод $ICR-$ факторизации,представленный ниже.

Устойчивость метода $CR-$ факторизации матриц

Анализ соотношений вида (1) показывает итерационный характер формул $CR-$ факторизации матриц с числом итераций n. Поскольку речь идет о матрицах большой размерности, то такая цепь вычислений может приводить к существенному накоплению ошибок. Для обеспечения устойчивости вычислительного процесса следует на каждом шаге факторизации выбирать ведущий элемент так, чтобы минимизировать влияние произведения $\boldsymbol{C}_{j_k}\cdot\boldsymbol{R}_{i_k}$ на субматрицу $\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}$, из которой образуется субматрица $\boldsymbol{A}_{(-i_k,-j_k)}$. Для этого поиск ведущего элемента $a_{(-i_{k-1},-j_{k-1})}(i_k,j_k)$ в субматрице $\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}$ необходимо осуществлять в соответствии с требованием получения таких множителей $\boldsymbol{C}_{j_k}$ и $\boldsymbol{R}_{i_k}$, которые имеют минимально возможную норму

$$\begin{aligned}&\left\|\boldsymbol{A}_{(-i_k,-j_k)}-\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}\right\|=\\&=\left\|\boldsymbol{C}_{j_k}\cdot\boldsymbol{R}_{i_k}\right\|\le\left\|\boldsymbol{C}_{j_k}\right\|\cdot\left\|\boldsymbol{R}_{i_k}\right\|\to\min_{(i_k,j_k)}\end{aligned}. \tag{5}$$

Здесь символом $\|\circ\|$ обозначена октаэдральная норма объекта $\circ$. Выбор именно такой нормы связан с простотой алгоритма ее вычисления и, что самое главное, устойчивостью получаемых результатов к вычислительным ошибкам.

Так как

$$\left\|\boldsymbol{C}_{j_k}\right\|=\sum_i\left|c_{ij_k}\right|=\frac{\sum_i\left|a_{(-i_{k-1},-j_{k-1})}(i,j_k)\right|}{\left|r_{i_k j_k}\right|}, \tag{6}$$

$$\left\|\boldsymbol{R}_{i_k}\right\|=\sum_j\left|r_{i_k j}\right|=\frac{\sum_j\left|a_{(-i_{k-1},-j_{k-1})}(i_k,j)\right|}{\left|c_{i_k j_k}\right|}, \tag{7}$$

$$a_{(-i_{k-1},-j_{k-1})}(i_k,j_k)=c_{i_k j_k}r_{i_k j_k} \tag{8}$$

и, к тому же, j_k – столбец и i_k – строка субматрицы $\boldsymbol{A}_{(-i_k,-j_k)}$ являются нулевыми, то из (5), с учетом (1), имеем уточненное выражение

$$\begin{aligned}&\left[\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(i_k,:)\right\|-\left|a_{(-i_{k-1},-j_{k-1})}(i_k,j_k)\right|\right]\times\\&\times\left[\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(:,j_k)\right\|-\left|a_{(-i_{k-1},-j_{k-1})}(i_k,j_k)\right|\right]\div\\&\div\left|a_{(-i_{k-1},-j_{k-1})}(i_k,j_k)\right|\to\min_{(i_k,j_k)},\end{aligned} \tag{8}$$

где

$$\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(*,j_k)\right\|=\sum_i\left|a_{(-i_{k-1},-j_{k-1})}(i,j_k)\right|,$$

$$\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(i_k,*)\right\|=\sum_j\left|a_{(-i_{k-1},-j_{k-1})}(i_k,j)\right|.$$

Критерий (8) оказывается легко реализуемым, поскольку нахождение входящих в него норм осуществляется рекуррентно и не требует значительных вычислительных затрат. К тому же, для поиска ведущего элемента $a_{(-i_{k-1},-j_{k-1})}(i_k,j_k)$ нет необходимости производить оценки функционала в выражении (8) для всех элементов субматрицы $\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}$. Поиск ведущего элемента достаточно осуществить в $n_k << n$ ненулевых строках субматрицы $\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}$, содержащих наименьшее количество ненулевых элементов. Такой способ поиска ведущего элемента не только обеспечивает устойчивость вычислений, но и уменьшает различие между шаблонами субматриц $\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}$ и $\boldsymbol{A}_{(-i_k,-j_k)}$.

Анализируя выражения (5) и (8), отметим, что предельное (нулевое) значение функционала в (8) достигается при условии, когда ведущий элемент выбирается на пересечении строки и столбца, в одном из которых имеется только один элемент отличный от нуля. В этом случае при переходе от субматрицы $\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}$ к субматрице $\boldsymbol{A}_{(-i_k,-j_k)}$ ошибок не допускается. В остальных случаях возможно возникновение ошибок компьютерных вычислений, однако их величина будет ограничена тем больше, чем меньше будет значение функционала в выражении (5).

Метод $ICR-$ факторизации матриц

Обращаясь к рекуррентным соотношениям вида (1), определим условия, при которых можно пренебречь влиянием строк и столбцов формируемых факторных матриц на субматрицы. Очевидно, в случае несопоставимости строчных и столбцовых октаэдральных норм соответствующих векторов, а именно

$$\left|c_{ij_k}\right|\cdot\left\|\boldsymbol{R}_{i_k}\right\|<<\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(i,*)\right\| \tag{9}$$

и

$$\left|r_{i_k j}\right|\cdot\left\|\boldsymbol{C}_{j_k}\right\|<<\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(*,j)\right\|, \tag{10}$$

элементы c_{ij_k} и $r_{i_k j}$ оказываются несущественными и их можно полагать равными нулю. Для оценивания значимости элементов c_{ij_k} и $r_{i_k j}$ нет необходимости в предварительном вычислении столбца $\boldsymbol{C}_{j_k}$ и строки $\boldsymbol{R}_{i_k}$, поскольку выражения (9) и (10), с учетом (6) и (7), можно представить в тождественном виде

$$\frac{|c_{ij_k}|}{|c_{i_k j_k}|} << \frac{\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(i,*)\right\|}{\left\|\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}(i_k,*)\right\|} = \frac{\sum_j \left|a_{(-i_{k-1},-j_{k-1})}(i,j)\right|}{\sum_j \left|a_{(-i_{k-1},-j_{k-1})}(i_k,j)\right|}, \qquad (11)$$

$$\frac{|r_{i_k j}|}{|r_{i_k j_k}|} << \frac{\left\|\boldsymbol{A}_{(-i_k,-j_k)}(*,j)\right\|}{\left\|\boldsymbol{A}_{(-i_k,-j_k)}(*,j_k)\right\|} = \frac{\sum_i \left|a_{(-i_{k-1},-j_{k-1})}(i,j)\right|}{\sum_i \left|a_{(-i_{k-1},-j_{k-1})}(i,j_k)\right|}. \qquad (12)$$

Полученные соотношения позволяют оценивать значимость элементов столбца $\boldsymbol{C}_{j_k}$ и строки $\boldsymbol{R}_{i_k}$, основываясь только на соотношениях норм между соответствующими строками и столбцами субматрицы $\boldsymbol{A}_{(-i_{k-1},-j_{k-1})}$. Параметризация выражений (11) и (12) позволяет представить их в виде

$$\frac{|c_{ij_k}|}{|c_{i_k j_k}|} < \tau \frac{\sum_j \left|a_{(-i_{k-1},-j_{k-1})}(i,j)\right|}{\sum_j \left|a_{(-i_{k-1},-j_{k-1})}(i_k,j)\right|}, \qquad (13)$$

$$\frac{|r_{i_k j}|}{|r_{i_k j_k}|} < \tau \frac{\sum_i \left|a_{(-i_{k-1},-j_{k-1})}(i,j)\right|}{\sum_i \left|a_{(-i_{k-1},-j_{k-1})}(i,j_k)\right|}, \qquad (14)$$

удобном для практического использования. При этом априори задаваемое значение параметра τ устанавливает границу раздела элементов c_{ij_k} и $r_{i_k j}$ на значимые и незначимые. В процессе факторизации значимые элементы сохраняются в столбцах $\tilde{\boldsymbol{C}}_{j_k}$ и строках $\tilde{\boldsymbol{R}}_{i_k}$.

Учитывая тождественность выражений (13) и (14) выражениям (9) и (10), отбрасывание малозначимых элементов c_{ij_k} и $r_{i_k j}$ в формулах (1) можно интерпретировать как ошибки машинных вычислений, а параметр τ рассматривать в качестве обобщенной характеристики таких ошибок.

Метод факторизации матриц, основанный на формулах (1), с выбором ведущих элементов в соответствии с критерием (8) и пренебрежением элементами факторных матриц, удовлетворяющих условиям (13) и (14), называется методом неполной столбцово-строчной факторизации матриц, или, кратко, методом $ICR-$ факторизации.

В случае $ICR-$факторизации плохо обусловленных матриц чувствительность решений к ошибкам существенно возрастает и, поэтому, параметр τ в (13) и (14) следует выбирать малым настолько, насколько необходимо уменьшить влияние подобных ошибок. Очевидно, для 32-разрядных вычислительных систем выбор значения параметра τ меньше величины 10^{-16} смысла не имеет, поскольку такой выбор эквивалентен установлению значения $\tau = 0$ и, следовательно, выполнению полной $CR-$факторизации. В большинстве случаев значение параметра τ устанавливается в пределах $10^{-1} \div 10^{-4}$, что достаточно для построения эффективных предобусловливателей.

Результаты экспериментальных исследований

Тестирование предложенного метода выполнялось на примерах несимметричных матриц, которые использовались в работе [5]. Их основные характеристики отражены в таблице 1. Здесь указаны наименования тестовых матриц, по которым их можно найти в интернете, размерности матриц *n*, число содержащихся в них ненулевых элементов *nz*.

Все расчеты выполнялись на вычислительном устройстве Desktop Computer. Его параметры, а также характеристики использованного нами программного обеспечения представлены в таблице 2. Для сравнения здесь отражены характеристики вычислительного устройства IBM System x3850 и программного обеспечения, задействованных в работе [5] для факторизации тех же тестовых матриц.

Сопоставление полученных результатов с результатами, представленными в работе [5], выполнялось в условиях ограничения требований к используемым ресурсам памяти для размещения факторных матриц $\tilde{\boldsymbol{C}}$ и $\tilde{\boldsymbol{R}}$, теми объемами, которые использовались в [5] для размещения матриц $\tilde{\boldsymbol{L}}$ и $\tilde{\boldsymbol{U}}$.

Результаты экспериментов (таблица 3) показали близость аппроксимационных свойств предобусловливателей, полученных методами $ICR-$, $left-RIF-Ns-$, $right-RIF-Ns-$, $AINV-$, $ILUC-$факторизации, не взирая на то, что $ICR-$факторизация матриц осуществлялась без привлечения условия (4) для отсечения элементов матриц $\tilde{\boldsymbol{C}}$ и $\tilde{\boldsymbol{R}}$. В большинстве тестов затраты времени на факторизацию оказались сопоставимыми. Преимущества метода $ICR-$факторизации были весьма заметными на матрицах dc3, trans4, trans5, в отношении которых известные методы факторизации использовались без привлечения процедур предварительного упорядочения матричных строк и столбцов. Превосходство заключалось в ускорении сходимости итерационных процедур в случае использования ICR – предобусловливателя.

Таблица 1. Характеристики тестовых матриц

Название матрицы	*n*	*nz*
raefsky5	6 316	167 178
raefsky6	3 402	130 371
ASIC_100ks	99 190	578 890
FEM_3D_thermal1	17 880	430 740
FEM_3D_thermal2	147 900	3 489 300
sme3Da	12 504	874 887
sme3Db	29 067	2 081 063
dc3	116 835	766 396
trans4	116 835	749 800
trans5	116 835	766 396
raefsky1	3 242	293 409
raefsky2	3 242	293 551
raefsky3	21 200	1 488 768
hcircuit	105 676	513 072
ASIC_680ks	682 712	1 693 767
ASIC_320k	321 821	1 931 828
ASIC_320ks	321 671	1 316 085
ASIC_100k	99 340	940 621
epb3	84 617	463 625
poisson3Da	13 514	352 762
sme3Dc	42 930	3 148 656
stomach	213 360	3 021 648

Таблица 2. Характеристики ресурсов используемых для факторизации тестовых матриц

Computer	IBM System x3850	Desktop Computer
Processor	64-bit	32-bit
SMP processors	4	1
Type	Intel Xeon MP	Intel Pentium 4
Clock speed	3.66 GHz	3.0 GHz
Front side bus	667 MHz	800 MHz
L2 cache size	1 MB	1 MB
Memory	DDR2 dual	DDR dual
RAM size	16 GB	1 GB
RAM speed	400 MHz	400 MHz
Operation system	it is not specified	Microsoft Windows XP
Compiler	Fortran–90	C++
Compiler type	Intel Compiler ifort	Microsoft Visual Studio 2008
Optimization	O4	Full

Таблица 3. Количество итераций метода GMRES(30) с разными предобусловливателями

Название матрицы	*ICR*	*lRIF*	*rRIF*	*AINV*	*ILUC*
raefsky5	5	5	5	9	5
raefsky6	4	8	5	9	4
ASIC_100ks	12	10	10	30	8
FEM_3D_thermal1	10	11	9	27	5
FEM_3D_thermal2	9	10	8	23	5
sme3Da	1610	–	2434	–	1903
sme3Db	2193	–	–	–	–
dc3	15	73	73	60	50
trans4	8	25	23	11	21
trans5	10	56	37	17	45
raefsky1	30	33	33	708	118
raefsky2	49	88	67	842	135
raefsky3	87	–	149	–	74
hcircuit	4	29	59	–	11
ASIC_680ks	36	7	7	34	33
ASIC_320k	19	13	16	–	8
ASIC_320ks	2	4	4	16	5
ASIC_100k	21	13	17	221	8
epb3	133	136	104	693	92
poisson3Da	22	26	26	141	16
sme3Dc	2002	–	–	–	–
stomach	6	6	6	74	5

Выводы

Метод ICR – факторизации матриц отличается высокими аппроксимационными свойствами, низкими требованиями к используемым ресурсам памяти и устойчивостью вычислений к ошибкам отсечения малозначимых элементов. Такой метод позволяет создавать эффективные программные средства решения больших разреженных систем уравнений.

Литература

1. Саух С.Е.: Метод *CR*–факторизации матриц большой размерности // Электронное моделирование, N6, 2007. – С. 3 – 20.
2. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. М., Мир, 1991. – 386 с.
3. Saad Y. Iterative Methods for Sparse Linear Systems. University of Minnesota, Minneapolis, MN, 2000. – 448 p.
4. Li N., Saad Y., Chow E. Crout versions of ILU for general sparse matrices // SIAM Journal on Scientific Computing, Vol. 25, 2003. – P. 716 – 728.
5. Rafiei A., Bollhöfer M. Robust incomplete factorization for nonsymmetric matrices // *Technical Report 26-2008, TU Berlin, Institute of Mathematecity, July 2008.*
6. Benzi M., Tuma M. A sparse approximate inverse preconditioner for nonsymmetric linear systems // SIAM Journal on Scientific Computing, Vol. 19, 1998. – P. 968 – 994.

Поступила в редакцию 15.02.2010