

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ИТЕРАЦИОННОГО АЛГОРИТМА РЕШЕНИЯ НЕСИММЕТРИЧНЫХ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ С ЧАСТИЧНЫМ СОХРАНЕНИЕМ СПЕКТРАЛЬНОЙ/СИНГУЛЯРНОЙ ИНФОРМАЦИИ ПРИ ЯВНЫХ РЕСТАРТАХ

С.А. Харченко

Введение. Моделирование задач вычислительной аэро- и гидродинамики является актуальной проблемой многих отраслей промышленности. Параллельная версия программного комплекса “FlowVision” [1-2], разрабатываемого в ООО “ТЕСИС”, используется для моделирования широкого круга промышленных задач, включая задачи с подвижными телами и свободными поверхностями для областей со сложной 3D геометрией. В программном комплексе “FlowVision” используются неявные схемы аппроксимации, что приводит к необходимости решать соответствующие системы линейных уравнений. Для адекватного воспроизведения тонких физических эффектов в геометрически сложных трехмерных областях требуются подробные расчетные сетки, содержащие от сотен тысяч до сотен миллионов расчетных ячеек. Моделирование подобных задач требует огромных вычислительных ресурсов и может быть проведено только на самой современной мощной параллельной вычислительной технике.

При моделировании задач гидродинамики с использованием неявных схем аппроксимации системы линейных уравнений приходится решать многократно с достаточно высокой точностью, причем этот этап вычислений является одним из самых ресурсоемких. По этой причине повышение эффективности параллельных алгоритмов решения систем линейных уравнений является одним из способов повышения эффективности использования вычислительных ресурсов при моделировании.

Современные эффективные параллельные итерационные алгоритмы решения несимметричных систем линейных уравнений включают в себя построение предобуславливания и соответствующие итерации, как правило крыловского типа. В симметричном случае традиционно используется итерационный алгоритм CG [3], в несимметричном — это такие итерационные алгоритмы как алгоритм Ланцоша [4], QMR [5] или BiCGSTAB[6]. Все упомянутые итерационные алгоритмы являются алгоритмами Крыловского типа с короткими рекуррентными соотношениями. В несимметричном случае возможно также использование алгоритма GMRES [7] с полными ортогонализациями векторов и его аналогов. Существенным преимуществом алгоритма GMRES без рестартов является его высокая устойчивость по отношению к ошибкам округления в несимметричном случае, однако высокие затраты памяти и вычислений на ортогонализации векторов делают его неконкурентоспособным. Для уменьшения затрат памяти и вычислений при ортогонализациях векторов было предложено несколько модификаций этого алгоритма с неявными рестартами, таких как IRGMRES из [8]. В данной работе рассматривается параллельная реализация алгоритма SOFGMRES [9] с явными рестартами и сохранением части информации при рестартах.

Работы выполнялись в рамках работ по государственному контракту 02.514.11.4125 с Министерством образования и науки РФ.

Алгоритм SOFGMRES(m). Рассмотрим систему линейных уравнений вида

$$Ax = b, \quad (1)$$

где $A \in \mathbb{R}^{N \times N}$ - заданная невырожденная, возможно знакоопределенная несимметричная матрица, $b \in \mathbb{R}^N$ - заданный вектор правой части, $x \in \mathbb{R}^N$ - неизвестный вектор. В дальнейшем будем предполагать, что предобуславливатель и ненулевое начальное приближение, если таковые имеются, уже учтены соответствующим образом в A и b .

Основой алгоритма SOFGMRES(m) являются матричные соотношения [9] вида:

$$\begin{cases} AY_k = W_k R_k, \\ Y_k^T Y_k = I_k, \\ W_k^T W_k = I_k, \end{cases} \quad (2)$$

где $Y_k \in \mathbb{R}^{N \times k}$, $W_k \in \mathbb{R}^{N \times k}$, $R_k \in \mathbb{R}^{k \times k}$, матрица R_k - верхняя треугольная. Соотношения (2) будем называть *матричными соотношениями в QR форме*. Обозначим через Y_k^\perp дополнение к Y_k до ортонормированного базиса в \mathbb{R}^N , W_k^\perp дополнение к W_k до ортонормированного базиса в $Span(A)$.

Пусть для некоторого k , $0 \leq k < N$, имеют место матричные соотношения (2). Пусть $y_{k+i} \in \mathbb{R}^N$, $i=1, \dots, m$, $k+m < N$, есть некоторый дополнительный набор направлений такой, что матрица $[Y_k, y_{k+1}, \dots, y_{k+m}]$ является матрицей с ортонормированными столбцами, т.е.,

$$[Y_k, y_{k+1}, \dots, y_{k+m}]^T [Y_k, y_{k+1}, \dots, y_{k+m}] = I_{k+m}.$$

В соответствии с построениями работы [9], рассмотрим QR разложение матрицы $[AY_k, b, Ay_{k+1}, \dots, Ay_{k+m}]$, где b - правая часть из (1). Имеют место соотношения:

$$\begin{cases} [AY_k, b, Ay_{k+1}, \dots, Ay_{k+m}] = P_{k+m+1} S_{k+m+1}, \\ P_{k+m+1}^T P_{k+m+1} = I_{k+m+1}, \end{cases} \quad (3)$$

где $P_{k+m+1} \in \mathbb{R}^{M \times (k+m+1)}$ - матрица с ортонормированными столбцами, а матрица $S_{k+m+1} \in \mathbb{R}^{(k+m+1) \times (k+m+1)}$ - верхняя треугольная. Выделяя из соотношений (3) $(k+1)$ -ый столбец, отвечающий правой части b , для $i=0, \dots, m$ матричные соотношения (3) можно переписать в следующем виде:

$$\begin{cases} AY_{k+i} = P_{k+i+1} H_{k+i}, \\ Y_{k+i}^T Y_{k+i} = I_{k+i}, \\ P_{k+i+1}^T P_{k+i+1} = I_{k+i+1}, \\ b = P_{k+i+1} \gamma_{k+i+1}, \end{cases} \quad (4)$$

где $Y_{k+i} = [Y_k, y_{k+1}, \dots, y_{k+i}]$, матрица P_{k+i+1} содержит первые $(k+i+1)$ столбцов матрицы P_{k+m+1} , первые $(k+1)$ компоненты вектора $\gamma_{k+i+1} \in \mathbb{R}^{k+i+1}$ есть первые $(k+1)$ элемента $(k+1)$ -го столбца матрицы S_{k+m+1} , а остальные компоненты нулевые, верхняя хессенбергова матрица H_{k+i} составлена из первых $(k+i+1)$ столбцов матрицы S_{k+m+1} исключая $(k+1)$ -ый столбец.

Матрица H_{k+i} в соотношениях (4) верхняя хессенбергова, поэтому существует набор вращений Гивенса $G_j \in \mathbb{R}^{(k+i+1) \times (k+i+1)}$, $j=1, \dots, k+i$,

$$G_j = \begin{bmatrix} I_{j-1} & & & 0 \\ & c_j & s_j & \\ & -s_j & c_j & \\ 0 & & & I_{k+i-j} \end{bmatrix}, c_j^2 + s_j^2 = 1,$$

такой, что

$$H_{k+i} = Q_{k+i} \begin{bmatrix} R_{k+i} \\ 0 \end{bmatrix}, \quad (5)$$

где

$$Q_{k+i} = G_1 * \dots * G_{k+i} -$$

унитарная матрица, которая представляет собой произведение вращений Гивенса, R_{k+i} - верхняя треугольная матрица. Равенство (8) является QR-разложением матрицы H_{k+i} . Заметим, что по построению первые k вращений Гивенса – единичные матрицы.

Обозначим

$$W_{k+i} = P_{k+i+1} Q_{k+i} \begin{bmatrix} I_{k+i} \\ 0 \end{bmatrix}, \quad (6)$$

тогда первое равенство из (4) можно представить в QR форме $AY_{k+i} = W_{k+i} R_{k+i}$. Соотношения (4) будем называть *рекуррентными матричными соотношениями алгоритма SOFGMRES(m)*.

Алгоритм SOFGMRES(m):

Step 0. Инициализация:

$$x_0 = 0;$$

Пусть имеют место соотношения (4) при $i=0$;

For $i=1, \dots, m$ Do

Step 1. Решение локальной задачи минимизации:

$$\|H_{k+i-1}z_{i-1} - \begin{bmatrix} \gamma_{k+1} \\ 0 \end{bmatrix}\| \rightarrow \min_{z_{i-1}} ;$$

Step 2. Вычисление вектора редуцированной невязки:

$$t_{i-1} = \begin{bmatrix} \gamma_{k+1} \\ 0 \end{bmatrix} - H_{k+i-1}z_{i-1} ;$$

Step 3. Вычисление вектора текущей невязки:

$$r_{i-1} = P_{k+i}t_{i-1} ;$$

Step 4. Вычисление нового приближения к решению:

$$\text{If } \|r_{i-1}\| \leq \varepsilon \text{ then} \\ x_{i-1} = Y_{k+i-1}z_{i-1} ; \\ \text{Stop}$$

End If

Step 5. Ортогонализация: для текущего вектора невязки r_{i-1} достраиваем QR разложение вида

$$r_{i-1} = Y_{k+i} [g_{1,k+i}, \dots, g_{k+i,k+i}]^T, \quad (7)$$

откуда находим новый нормированный вектор направления y_{k+i} , ортогональный направлениям y_1, \dots, y_{k+i-1} ;

Step 6. Умножение предобусловленной матрицы A на вектор направления y_{k+i} :

$$\hat{p}_{k+i} = A y_{k+i} ;$$

Step 7. Ортогонализация: для текущего вектора \hat{p}_{k+i} достраиваем QR разложение (6);

Обозначим

$$Y_{k+i} = [Y_k, y_{k+1}, \dots, y_{k+i}], \quad P_{k+i+1} = [P_{k+i}, p_{k+i+1}], \\ H_{k+i} = \begin{bmatrix} & & h_{1,k+i+1} \\ H_{k+i-1} & & \vdots \\ & & h_{k+i,k+i+1} \\ 0 & & h_{k+i+1,k+i+1} \end{bmatrix} ;$$

End For

Из описания алгоритма SOFGMRES(m) следует, что на каждой итерации этого алгоритма необходимо произвести одно умножение вектора невязки на предобусловленную матрицу и две ортогонализации наборов векторов. Также очевидно, что по сравнению с алгоритмом GMRES требуется хранить два набора векторов вместо одного.

Теория сходимости алгоритма SOFGMRES(m). Приведем без доказательств основные утверждения, описывающие теорию сходимости алгоритма SOFGMRES(m), подробные доказательства представлены в работе [9].

Лемма 1. Для сингулярных чисел матриц A и R_{k+i} для всех $0 \leq i \leq m$ имеют место неравенства теоремы Коши о разделении.

Лемма 2. Для всех $0 < i \leq m$ имеет место соотношение $r_{i-1} \perp \text{Span}(W_{k+i-1})$.

Теорема 1. Пусть первые k вектора направлений в матричных соотношениях (6) получены проведением k итераций алгоритма Арнольди с матрицей A и начальным вектором $b/\|b\|$. Пусть на всех итерациях внутреннего цикла алгоритма SOFGMRES(m) имеют место неравенства $g_{k+i,k+i} \neq 0$ для всех $0 < i \leq m$. Тогда полученные после m итераций внутреннего цикла алгоритма SOFGMRES матричные соотношения можно рассматривать как матричные соотношения алгоритма Арнольди.

Лемма 3. Имеют место неравенства:

$$\|r_i\| \leq \|r_{i-1}\| \sqrt{1 - \theta_{i-1}^2 \psi_{i-1}^2},$$

где

$$\theta_{i-1} = \frac{|(Ar_{i-1}, r_{i-1})|}{\|r_{i-1}\|^2}, \quad \psi_{i-1} = \|\{R_{k+i}\}_{*,k+i}\|^{-1}, \quad \theta_{i-1} \psi_{i-1} \leq 1.$$

Определение 1. Пусть имеют место матричные соотношения в QR форме (2). Обозначим

$$\Theta_A(Y_k) = \min_{\substack{w \perp \text{Span}(W_k) \\ \|w\| \neq 0}} \frac{|(Aw, w)|}{\|w\|^2}, \quad \Psi_A(Y_k)^{-1} = \max_{\substack{y \perp \text{Span}(Y_k) \\ \|y\| \neq 0}} \frac{\|Ay\|}{\|y\|}, \quad \eta_A(Y_k) = \Theta_A(Y_k) * \Psi_A(Y_k).$$

Величину $\eta_A(Y_k)$ будем называть обобщенным обратным числом обусловленности матрицы A на подпространстве $\text{Span}(Y_k)$.

Лемма 4. Монотонность характеристик подпространств: если $\text{Span}(Y) \subset \text{Span}(Z)$, то

$$\Theta_A(Y) \leq \Theta_A(Z), \quad \Psi_A(Y) \leq \Psi_A(Z), \quad \eta_A(Y) \leq \eta_A(Z).$$

Лемма 5. Имеют место неравенства:

$$0 \leq \eta_A(Y_k) \leq 1.$$

Теорема 2. Имеют место неравенства:

$$\|r_i\| \leq \|r_{i-1}\| \sqrt{1 - \eta_A^2([Y_k, y_{k+1}, \dots, y_{k+i-1}])} \leq \|r_{i-1}\| \sqrt{1 - \eta_A^2(Y_k)}.$$

Сохранение части спектральной/сингулярной информации при рестартах. Из Теоремы 1 следует, что если начальное подпространство отсутствует ($k=0$) и на итерациях не было обрыва в вычислениях (равенство нулю коэффициента $g_{k+i, k+i}$ из (7)), то алгоритм SOFGMRES(m) математически эквивалентен алгоритму GMRES. При этом в алгоритме SOFGMRES(m) требуется вдвое больше памяти для хранения векторов и вдвое большие затраты на ортогонализацию векторов. С другой стороны, в алгоритме SOFGMRES(m) в выборе начального подпространства имеется произвол. С точки зрения оценки Теоремы 2, для быстрой сходимости алгоритма SOFGMRES(m) лучшим является такое начальное подпространство $\text{Span}(Y_k)$, для которого обратное число обусловленности $\eta_A(Y_k)$ матрицы A на подпространстве $\text{Span}(Y_k)$ максимально близко к единице.

Существуют различные способы нахождения подходящего начального подпространства $\text{Span}(Y_k)$.

Базовым способом нахождения такого подпространства является следующий: для небольшого m проводим цикл итераций алгоритма SOFGMRES(m), строим новое приближение к решению, из полученного нового набора направлений оставляем некоторые линейные комбинации векторов построенного набора, остальные направления отбрасываем, получаем новое начальное подпространство для последующих циклов итераций, и т. д. Ясно, что фильтрация части направлений ухудшает сходимость по сравнению с полным сохранением подпространств (по существу по сравнению с алгоритмом GMRES без рестартов). С другой стороны, существует естественный способ фильтрации новых направлений, при котором число сохраняемых направлений минимально и потери скорости сходимости из-за фильтрации векторов также минимальны.

Предлагаемый способ фильтрации основывается на оценке Теоремы 2 [9]. Построим два поднабора \tilde{Y}_{l_1} и \tilde{Y}_{l_2} линейных комбинаций текущих векторов y_{k+1}, \dots, y_{k+m} , отдельно обеспечивающих выполнение приближенных соотношений:

$$\Theta_A(Y_{k+m}) \approx \Theta_A([Y_k, \tilde{Y}_{l_1}]), \quad \Psi_A(Y_{k+m}) \approx \Psi_A([Y_k, \tilde{Y}_{l_2}]). \quad (8)$$

Обозначим

$$R_{k+m} = \begin{bmatrix} R_k & \hat{R}_m^{(12)} \\ 0 & \hat{R}_m^{(22)} \end{bmatrix}.$$

Для обеспечения второго приближенного равенства из (8) вычисляем правые сингулярные пары $\{\sigma_j, v_j\}$, $j=1, \dots, m$ матрицы $\hat{R}_m^{(22)}$ и полагаем параметр l_2 равным числу сингулярных чисел матрицы $\hat{R}_m^{(22)}$, превышающим заданный порог σ , $\sigma > 1$. Набор векторов блока \tilde{Y}_{l_2} строим как линейную комбинацию векторов y_{k+1}, \dots, y_{k+m} с коэффициентами, равными компонентам соответствующего сингулярных векторов v_j .

Для обеспечения первого приближенного равенства из (8) вычислим матрицу

$$T_m = [w_{k+1}, \dots, w_{k+m}]^T [y_{k+1}, \dots, y_{k+m}] (\hat{R}_m^{(22)})^T.$$

Для симметризованной матрицы $T_m^+ = \frac{T_m + T_m^T}{2}$ вычислим все собственные пары этой матрицы

$\{\lambda_j, x_j\}$, $j=1, \dots, m$. Полагаем параметр l_1 равным числу собственных чисел матрицы T_m^+ , меньших заданного порога λ , $0 < \lambda < 1$. Базис подпространства $\text{Span}(\tilde{Y}_{l_1})$ строим как

ортономированную линейную комбинацию векторов y_{k+1}, \dots, y_{k+m} с коэффициентами $(\hat{R}_m^{(22)})^{-1} x_j$ для соответствующих собственных векторов x_j .

Для обеспечения одновременного выполнения приближенных соотношений (8) в силу монотонности характеристик подпространств (Лемма 4) теперь достаточно вычислить единый ортономированный базис \hat{Y}_l такой, что $(\text{Span}(\tilde{Y}_{l_1}) \cup \text{Span}(\tilde{Y}_{l_2})) \subset \text{Span}(\hat{Y}_l)$.

Для минимизации числа сохраненных направлений при большом числе рестартов в алгоритме SOFGMRES иногда целесообразно проводить так называемые двойные и даже тройные фильтрации, при которых в фильтрации участвуют не только текущие направления, но и группы направлений, сохраненных на предыдущих циклах алгоритма.

Комбинированная MPI+threads параллельная реализация ортогонализации векторов на основе преобразований Хаусхолдера. С точки зрения параллельной реализации основными вычислительными операциями алгоритма SOFGMRES являются умножение матрицы системы уравнений на вектор, умножение предобуславливателя на вектор (эти две операции в алгоритме SOFGMRES для простоты изложения сведены к одной), а также ортогонализация векторов и операция вычисления линейной комбинации векторов направлений. Параллельные алгоритмы умножения матрицы системы уравнений и предобуславливания на вектор подробно рассмотрены в других работах [10-12], поэтому в данной работе остановимся только на параллельной реализации алгоритмов, связанных с ортогонализациями векторов.

Хорошо известно [13], что с точки зрения минимизации ошибок округления ортогонализацию векторов целесообразно производить неявно с использованием преобразований Хаусхолдера, как это делается в случае вычисления QR разложения прямоугольной матрицы.

Пусть для начала для некоторой прямоугольной матрицы $C \in \mathbb{R}^{N \times k}$, $k \ll N$, построено QR разложение этой матрицы через преобразования Хаусхолдера в виде произведения

$$C = \left(\prod_{i=1}^k G_i \right) \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad (9)$$

где матрица $R \in \mathbb{R}^{k \times k}$ - квадратная верхняя треугольная, а матрицы $G_i \in \mathbb{R}^{N \times N}$ - ортогональные матрицы преобразования Хаусхолдера, имеющие вид $G_i = I_N + \tau_i u_i u_i^T$, где τ_i - некоторое число, $u_i \in \mathbb{R}^N$ - некоторый вектор, первые $(i-1)$ компонент которого нулевые. Расширение QR разложения для расширенной на один столбец матрицы $[C, c] \in \mathbb{R}^{N \times (k+1)}$, как это хорошо известно, можно вычислить применением в соответствующем порядке транспонированных преобразований Хаусхолдера к столбцу c и в последующем с добавлением из результата нового столбца в матрицу R и построением нового преобразования Хаусхолдера.

В контексте ортогонализаций векторов из QR разложения (9) можно явно выделить ортономированные векторы базиса подпространства $\text{Span}(C)$ в виде произведения $p_j = \left(\prod_{i=1}^k G_i \right) \begin{bmatrix} e_j \\ 0 \end{bmatrix}$.

В случае параллельных вычислений разные части матрицы C обычно расположены на разных процессорах. В этом случае QR разложение имеет смысл вычислять в распределенном виде способом, отличным от (9).

Пусть теперь прямоугольная матрица $C \in \mathbb{R}^{N \times k}$, $k \ll N$, имеет вид

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \quad (10)$$

где числа строк в матрицах C_1 и C_2 равны соответственно N_1 и N_2 , $N = N_1 + N_2$. Построим ее QR разложение на основе преобразований Хаусхолдера с явным использованием блочной структуры матрицы C . Для этого строим на основе преобразований Хаусхолдера QR разложения блочных строк матрицы C :

$$C_1 = Q_1 R_1, \quad C_2 = Q_2 R_2. \quad (11)$$

Дополнительно к (11) на основе преобразований Хаусхолдера вычислим QR разложение:

$$\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} \Theta_1 \\ \Theta_2 \end{bmatrix} R. \quad (12)$$

Объединяя соотношения (10)-(12) можно получить соотношение $C = Q R$, где

$$Q = \begin{bmatrix} Q_1 * \Theta_1 \\ Q_2 * \Theta_2 \end{bmatrix}. \quad (13)$$

матрица с ортонормированными столбцами, а матрица R - верхняя треугольная, тем самым получено QR разложение матрицы C . Во всех соотношениях (11) - (13) при написании факторов QR разложения имеется в виду неявное мультипликативное представление Q фактора в виде произведения преобразований Хаусхолдера типа представления (9).

Рассмотрим теперь организацию вычислений в случае добавления в матрицу C одного столбца $c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$. На основе преобразований Хаусхолдера строим QR разложения блочных строк расширенной матрицы $[C, c] \in \mathbb{R}^{N \times (k+1)}$:

$$[C_1, c_1] = [Q_1, q_1] \begin{bmatrix} R_1 & r_1 \\ 0 & \tilde{r}_1 \end{bmatrix}, [C_2, c_2] = [Q_2, q_2] \begin{bmatrix} R_2 & r_2 \\ 0 & \tilde{r}_2 \end{bmatrix}. \quad (14)$$

На основе преобразований Хаусхолдера достраиваем QR разложение (12):

$$\begin{bmatrix} R_1 & r_1 \\ R_2 & r_2 \\ 0 & \tilde{r}_1 \\ 0 & \tilde{r}_2 \end{bmatrix} = \begin{bmatrix} \Theta_1 & \theta_1 \\ \Theta_2 & \theta_2 \\ 0 & \tilde{\theta}_1 \\ 0 & \tilde{\theta}_2 \end{bmatrix} \begin{bmatrix} R & r \\ 0 & \tilde{r} \end{bmatrix}. \quad (15)$$

Из соотношений (14)-(15) следует равенство $[C, c] = \tilde{Q} \begin{bmatrix} R & r \\ 0 & \tilde{r} \end{bmatrix}$, где

$$\tilde{Q} = \begin{bmatrix} Q_1 * \Theta_1 & Q_1 * \theta_1 + q_1 * \tilde{\theta}_1 \\ Q_2 * \Theta_2 & Q_2 * \theta_2 + q_2 * \tilde{\theta}_2 \end{bmatrix} = \begin{bmatrix} Q_1 & q_1 & 0 & 0 \\ 0 & 0 & Q_2 & q_2 \end{bmatrix} \begin{bmatrix} \Theta_1 & \theta_1 \\ 0 & \tilde{\theta}_1 \\ \Theta_2 & \theta_2 \\ 0 & \tilde{\theta}_2 \end{bmatrix}, \quad (16)$$

причем матрица \tilde{Q} - матрица с ортонормированными столбцами. Заметим также, что в QR разложении (15) в матрице с левой стороны равенства число строк вдвое превышает число столбцов. Из формулы (16) следует, что для умножения на Q фактор QR разложения требуется сначала входной вектор умножить на Q фактор разложения (15), переставить компоненты результата, а затем локально раздельно умножить на Q факторы первой и второй блочных строк.

По аналогии с описанной выше техникой можно организовать распределенное вычисление QR разложения прямоугольной матрицы и в случае, когда эта матрица разбивается на большее чем 2 число блочных строк, например на t блочных строк, $t > 2$. В этом случае в соответствующем совместном QR разложении типа (15) число строк будет в t раз больше чем число столбцов.

Пусть теперь имеется параллельный компьютер с неоднородным доступом к памяти. А именно, пусть имеется набор узлов с распределенной памятью, а в каждом узле имеется несколько многоядерных процессоров работающих по общей памяти. На компьютере с такой неоднородной архитектурой естественно организовать комбинированные параллельные вычисления, в которых обмены и синхронизации по распределенной памяти осуществляются на основе MPI, а по общей - на основе какого-либо из механизмов работы с потоками, например, на основе технологии Intel TBB [14]. Для такого компьютера имеется MPI+threads реализация алгоритма решения системы линейных уравнений, например [12]. При этом предполагается, что число MPI процессов равно числу процессоров, а каждый MPI процесс порождает число потоков, равное числу ядер процессора [12]. Дополнительные вычисления по ортогонализации векторов на основе преобразований Хаусхолдера для алгоритма SOFGMRES(m) можно организовать следующим образом.

Распределение по MPI процессам матрицы системы уравнений порождает распределение по MPI процессам строк матрицы, для которой необходимо вычислить QR разложение на основе преобразований Хаусхолдера. Все строки этой матрицы для текущего MPI процесса разбиваются на число частей, равное числу потоков у этого MPI процесса. QR разложение своей части строк матрицы вычисляется потоками независимо. Затем для построения единого QR разложения достраивается объединяющий QR для текущего MPI процесса. Аналогично для распределенной памяти, сначала все MPI процессы достраивают свои части QR разложений с учетом биения по блочным строкам, а затем R части QR разложений собираются например на 0-м процессе и достраивается заключительный объединяющий QR. При умножении на Q фактор информация движется в обратном направлении: сначала производится умножение на Q фактор объединяющих QR разложений, затем после обмена если необходимо локальными переставленными частями вектора результата производится умножение на локальный Q фактор этого MPI процесса/потока вычислений.

Результаты численных экспериментов. Численные эксперименты по решению системы линейных уравнений проводились с симметричной положительно определенной матрицей размера $N=1\,135\,596$, число ненулевых элементов в верхнем треугольнике матрицы $Nz=75\,651\,236$. Тестовая система уравнений возникает при моделировании перемещений напряженно-деформированного состояния для задачи со сложной 3D геометрией. Выбор в качестве тестовой матрицы симметричной положительно определенной матрицы обусловлен необходимостью провести прямое сравнение алгоритма SOFGMRES(m) с традиционными итерационными алгоритмами.

Заполнение предобуславливателя для тестовой задачи составляло 185 % от заполнения матрицы. В алгоритме SOFGMRES(m) использовались следующие параметры: размер базового цикла $m=10$, нижний порог фильтрации $\lambda=0.001$, верхний порог фильтрации $\sigma=2.0$, количество циклов итераций до двойной фильтрации 10, количество двойных фильтраций до тройной фильтрации 3. При фильтрации направлений после любого цикла сохранялось как минимум одно направление соответствующее наименьшему собственному значению матрицы T_m^+ . Критерием остановки итераций во всех экспериментах было относительное уменьшение нормы невязки для нулевого начального приближения в $\varepsilon=10^{-9}$ раз. Расчеты проводились на Intel Core I7 920 (Nehalem) процессоре, 4 ядра, частота 2.67 GHz.

Таблица 1. Сравнение итерационных методов в режиме 1MPI x 4 threads.

Метод	Число итераций	Время, сек.
CG	480	104
Алгоритм Ланцоша	531	252
SOFGMRES(m)	525	260

В алгоритме SOFGMRES(m) число сохраненных векторов после окончания итераций $k=13$, максимальное число векторов для хранения данных в процессе итераций равно 58.

Таблица 2. Масштабируемость алгоритма SOFGMRES(m) в различных режимах вычислений.

Режим, Mpi x threads	1x1	2x1	1x2	4x1	2x2	1x4
Время, сек.	642	354	431	231	259	260

Предобуславливатель и число итераций в Таблице 2 не зависит от режима вычислений и равно 525 благодаря специальной технологии организации параллельных вычислений, описанной в работе [12].

Выводы. В работе предложена параллельная реализация итерационного алгоритма SOFGMRES(m) для компьютеров с неоднородным доступом к памяти. Численный эксперимент показал, что даже при малом размере базового цикла ($m=10$) и при огромном числе рестартов в алгоритме (52 рестарта) с точки зрения сходимости алгоритм SOFGMRES(m) ведет себя так, как будто рестартов итераций нет совсем. Число итераций и время вычислений для тестовой задачи почти совпадает с таковым у алгоритма Ланцоша (на одну итерацию в алгоритме Ланцоша приходится 2 умножения на предобусловленную матрицу), и всего в 2.5 раза выше, чем у классического алгоритма CG, применимого только в симметричном положительно определенном случае. Эксперименты также показали, что с точки зрения масштабируемости вычислений на тестовом компьютере более эффективно распараллеливание по MPI, чем по нитям вычислений. Это возможно связано с архитектурными особенностями процессора Intel Core I7 920 (Nehalem) и с особенностями использования памяти процессора в различных режимах вычислений.

ЛИТЕРАТУРА:

1. Aksenov A., Dyadkin A., Pokhilko V., "Overcoming of Barrier between CAD and CFD by Modified Finite Volume Method", Proc 1998 ASME Pressure Vessels and Piping Division Conference, San Diego, ASME PVP-Vol 377-1, 1998.
2. Aksenov A.A., Kharchenko S.A., Konshin V.N., Pokhilko V.I., "FlowVision software: numerical simulation of industrial CFD applications on parallel computer systems", Parallel CFD 2003 conference, Book of abstracts, p. 280-284, 2003.
3. Тыртышников Е.Е. Краткий курс численного анализа. Москва: ВИНТИ. 1994.
4. Paige C.C., Saunders M.A. "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares", ACM Transactions on Mathematical Software, Vol. 8, No. 1, March 1982, 43-71.
5. Freund R.W., Nachtigal N.M., "QMR: a quasi-minimal residual method for non-Hermitian linear systems", Numer. Math., 60 (1991), 315-339.
6. H.A.Van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems", SIAM J. Sci. Stat. Comput., v.13, 2, 631-644, 1992.
7. Saad Y., Schultz M.H., "GMRES: A generalized minimum residual algorithm for solving non-symmetric linear systems", SIAM J. Sci. Stat. Comput., 7 (1986), 856-869.

8. Morgan R.B., "Implicitly Restarted GMRES and Arnoldi Methods for Nonsymmetric Systems of Equations", Technical Report, 1997.
9. Харченко С.А., Еремин А.Ю., "Новые алгоритмы типа GMRES(k) с рестартами и анализ их свойств сходимости на основе QR формы матричных соотношений", Зап. Научн. Семина. ПОМИ, т. 268, 2000, 190-241.
10. Дядькин А.А., Харченко С.А. "Алгоритмы декомпозиции области и нумерации ячеек с учетом локальных адаптаций расчетной сетки при параллельном решении систем уравнений в пакете FlowVision" // Труды Всероссийской научной конференции "Научный сервис в сети Internet: многоядерный компьютерный мир", 201-206, 2007.
11. Харченко С.А. "Влияние распараллеливания вычислений с поверхностными межпроцессорными границами на масштабируемость параллельного итерационного алгоритма решения систем линейных уравнений на примере уравнений вычислительной гидродинамики " // Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск, Изд. ЮУрГУ, 2008, с.494-499.
12. Сушко Г.Б., Харченко С.А. " Экспериментальное исследование на СКИФ МГУ "Чебышев" комбинированной MPI+threads реализации алгоритма решения систем линейных уравнений, возникающих во FlowVision при моделировании задач вычислительной гидродинамики" // Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2009), Нижний Новгород, 30 марта – 3 апреля 2009 г. Челябинск, Изд. ЮУрГУ, 2009, с.316-324.
13. Walker H.F., "Implementation of the GMRES method using householder transformations", SIAM J. on Sci. and Stat. Comp., V. 9, 1, (1988), 152-163.
14. Intel Threading Building Blocks Tutorial – 1.6 / Intel Corp. 2007.