УДК 519.6

ПРОБЛЕМЫ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ТЕХНОЛОГИЙ РЕШЕНИЯ БОЛЬШИХ РАЗРЕЖЕННЫХ СЛАУ

В. П. Ильин[1]

Рассматриваются технологические проблемы реализации параллельных алгоритмов решения систем линейных алгебраических уравнений (СЛАУ) с разреженными матрицами высокого порядка, возникающими при сеточных аппроксимациях больших задач математического моделирования. Проводится классификация алгебраических систем, а также сравнительный анализ ресурсоемкости прямых, итерационных и комбинированных методов их решения с учетом различных структур и способов хранения матричных данных. Описываются сложности и основные пути достижения высокой производительности программного обеспечения при использовании многопроцессорных вычислительных систем (МВС) с распределенной и общей памятью, на основе применения MPI, OpenMP и гибридного программирования. Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований и Президиума РАН (коды проектов соответственно 08–01–00526 и 2.5). Статья подготовлена по материалам доклада автора на международной конференции "Параллельные вычислительные технологии" (ПаВТ-2009; http://agora.guru.ru/pavt2009).

Ключевые слова: системы линейных алгебраических уравнений, разреженные симметричные и несимметричные матрицы, прямые, итерационные и комбинированные методы, распараллеливание, компьютерные архитектуры.

1. Введение. При численном исследовании стационарных и нестационарных, линейных и нелинейных процессов в задачах математического моделирования, в том числе оптимизационных и междисциплинарных, многократное решение СЛАУ является наиболее ресурсоемким этапом. Многомерные системы дифференциальных уравнений (СДУ) или эквивалентные вариационные постановки, после их аппроксимации сеточными методами конечных разностей, конечных объемов или конечных элементов (МКР, МКО, МКЭ [1–3]), при современных требованиях к точности расчетов порождают алгебраические системы с разреженными матрицами сверхвысоких порядков (до 10^9 и более). Высокопроизводительные технологии решения задач такого масштаба должны обеспечивать, во-первых, интеграцию ресурсов большого количества процессоров, во-вторых, эффективное распараллеливание вычислений и, в-третьих, оптимизацию программного кода с его адаптацией к конкретному операционному окружению, что означает в целом отображение алгоритмов на архитектуру ЭВМ. Многообразие матричных структур и свойств, активно развивающихся методов решения СЛАУ и непрерывно меняющихся компьютерных платформ делает проблему построения экономичного алгебраического программного обеспечения постоянно актуальной.

Мы рассматриваем три группы алгоритмов, применяемых при использовании МКР, МКО и МКЭ различных порядков для дискретизации разных типов СДУ: диффузии с конвекцией, многофазной фильтрации, систем уравнений Ламе, Навье–Стокса, Максвелла и др.

Первая группа связана с решением задач с полностью или частично разделяющимися переменными и основана на кратных быстрых преобразованиях Фурье (БПФ), экономичных алгоритмах декомпозиции и редукции, а также на комбинированных подходах, ориентированных на достижение рекордной производительности в актуальных характерных приложениях. Такие алгебраические системы и методы имеют самостоятельное значение во многих практических случаях, но могут также зачастую использоваться в качестве экономичных вспомогательных подзадач при решении более сложных проблем. Вторая группа — это семейство универсальных предобусловленных итерационных процессов в подпространствах Крылова, в том числе оригинальные методы полусопряженных и бисопряженных направлений для разного типа матриц (симметричных и несимметричных, положительно определенных и знаконеопределенных, полного и неполного ранга), представляемых в общепринятых сжатых форматах. Третья группа методов посвящена решению крупноблочных СЛАУ, включая алгебраические задачи с седловой точкой, получаемые при дискретизации векторных уравнений электромагнетизма, гидрогазодинамики и др. Здесь подходы основаны

[1] Институт вычислительной математики и математической геофизики СО РАН, просп. Лаврентьева, 6, 630090, Новосибирск; главн. науч. сотр., e-mail: ilin@sscc.ru

на использовании комбинации прямых и итерационных решателей с учетом специальных структурных и спектральных матричных свойств. Блочные алгоритмы лежат также в основе распространенных и хорошо распараллеливаемых методах декомпозиции, для которых типичными являются двухуровневые итерационные процессы.

Описываемые нами реализации алгебраических методов на многопроцессорных и многоядерных вычислительных системах с разделенной, общей и неоднородной многоуровневой памятью основываются на алгебраических принципах декомпозиции областей, минимизации коммуникационных потерь и на использовании высокопроизводительных инструментальных средств библиотеки MKL компании Intel. Распараллеливание алгоритмов осуществляется средствами MPI, OpenMP и гибридного программирования [4]. Проблеме распараллеливания численных решений математического моделирования посвящено большое количество публикаций и цитируемые там работы (см., например, [5, 6]).

2. Алгебраические системы и матричные структуры. В качестве наглядного примера рассмотрим трехмерную смешанную краевую задачу для уравнения конвективной диффузии в параллелепипеде:

$$\begin{gathered}\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}+p\,\frac{\partial u}{\partial x}+q\,\frac{\partial u}{\partial y}+r\,\frac{\partial u}{\partial z}=f(x,y,z),\quad (x,y,z)\in\Omega=[0,a]\times[0,b]\times[0,c],\\ \Gamma=\bigcup_{s=1}^{6}\Gamma_s,\quad u+\kappa_s\,\frac{\partial u}{\partial n}\Big|_{\Gamma_s}=\chi_s,\quad s=1,\ldots,6,\quad p,\,q,\,r=p,\,q,\,r(x,y,z),\end{gathered}\tag{1}$$

где n означает внешнюю нормаль к границе. Остановимся на некоторых возможных особенностях этой проблемы, определяющих вычислительную сложность алгоритмов ее решения. В простейшем случае постоянных коэффициентов p, q, r, κ_s, χ_s задача (1) допускает разделение переменных, что позволяет строить наиболее экономичные методы [1, 2]. При $p=q=r=0$ задача становится самосопряженной (даже если расчетная область Ω не является параллелепипедом, что приводит к симметричным СЛАУ с "хорошими" свойствами). В общем случае уравнение (1) может быть нелинейным (p, q, r и f зависят от u), и тогда после применения линеаризации приходится решать серию линейных задач.

Если использовать типовые аппроксимации исходной задачи на регулярной параллелепипедоидальной сетке $x_{i+1}=x_i+h_i^x$, $i=0,1,\ldots,I$, $y_{j+1}=y_j+h_j^y$, $j=0,1,\ldots,J$, $z_{ki+1}=z_k+h_k^z$, $k=0,1,\ldots,K$, то мы приходим к семиточечной системе сеточных СЛАУ, имеющей порядок IJK:

$$\begin{aligned}(Au)_{i,j,k}&=p^0_{i,j,k}u_{i,j,k}-p^1_{i,j,k}u_{i-1,j,k}-p^2_{i,j,k}u_{i,j-1,k}-p^3_{i,j,k}u_{i+1,j,k}-p^4_{i,j,k}u_{i,j+1,k}-\\&\quad-p^5_{i,j,k}u_{i,j,k-1}-p^6_{i,j,k}u_{i,j,k+1},\\(Au)_{i,j,k}&=\Big[\big(A^x+A^y+A^z\big)u\Big]_{i,j,k}=f_{i,j,k},\quad \big(A^x+A^y\big)A^z=A^z\big(A^x+A^y\big),\\p^1_{1,j,k}&=p^3_{I,j,k}=p^2_{i,1,k}=p^4_{i,J,k}=p^5_{i,j,1}=p^6_{i,j,K}=0,\end{aligned}\tag{2}$$

Формулы для коэффициентов $p^0_{i,j,k},\ldots,p^6_{i,j,k}$ можно найти, например, в [2]. Будем считать здесь и в дальнейшем, если не оговорено противное, что матрица A невырожденна, а это обеспечивает существование единственного решения СЛАУ. Если узлы сетки и соответствующие компоненты сеточных функций пронумеровать одним индексом l, то при их естественной упорядоченности по правилу

$$l=i+(j-1)J+(k-1)JK$$

структура матрицы A имеет простую семидиагональную форму, представленную на рис. 1.

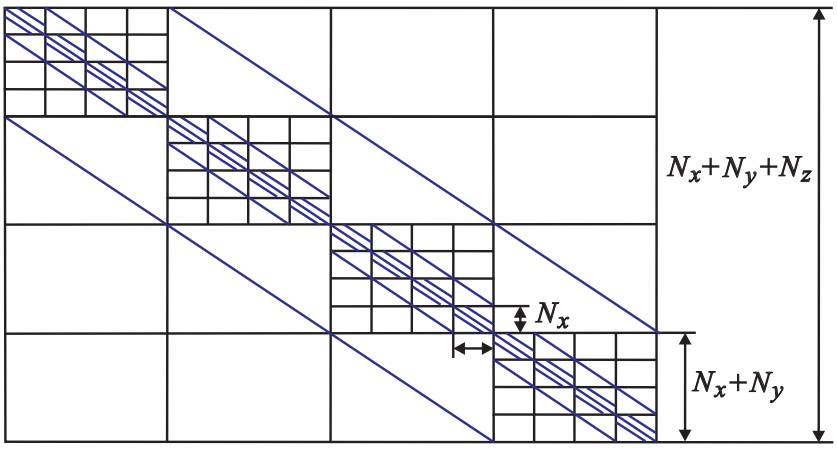

Рис. 1. Структура матрицы A

В более общем случае при использовании неструктурированных сеток матрица системы может быть представлена в виде $(Au)_l=p_{l,l}u_l-\sum\limits_{l'\in\omega_l}p_{l,l'}u_{l'}$, где ω_l означает сеточный шаблон l-го узла, т.е. совокупность узлов с номерами l', "соседних" к l и участвующих в l-м уравнении.

Матрицу СЛАУ всегда можно записать в форме $A = D - L - U$, где $D = \operatorname{diag}\{p_{l,l}\}$ — главная диагональ матрицы A, а L и U — нижняя и верхняя треугольные матрицы, элементы которых составлены из $p_{l,l'}$ при $l' < l$ и $l' > l$ соответственно.

Более сложные задачи возникают при решении СДУ, решениями которых являются векторные функции $u = (u_1, u_2, \dots)$. В таких ситуациях получаемая после дискретизации матрица A представляется в блочной форме, например блочного второго порядка: $A = \begin{bmatrix} A_{11}, & A_{12} \\ A_{21}, & A_{22} \end{bmatrix}$. Характерным случаем здесь является $A_{22} = 0$, когда соответствующая СЛАУ называется алгебраической задачей с седловой точкой.

3. Параллельные алгоритмы численных решений СЛАУ.

3.1. Задачи с разделяющимися переменными. Остановимся сначала на случае разделяющихся краевых задач. В (1) будем предполагать $p = q = r = \kappa_s = 0$, что порождает задачу Дирихле для уравнения Пуассона, при этом матрица A^z имеет базис из собственных векторов: $A^z\varphi^p = \lambda_p\varphi^p$, $p = 1, \dots, K$, что позволяет векторы u и f в (2) разложить в ряды Фурье

$$u = \sum_{p=1}^{K} v^p\varphi^p, \quad v^p = \{v^p_{i,j}\}, \quad f = \sum_{p=1}^{K} f^p\varphi^p, \quad f^p = \left\{ f^p_{i,j} = \sum_{k=1}^{K} f_{i,j,k}\varphi^p_{i,j,k} \right\}, \tag{3}$$

реализация которых особенно экономично выполняется с помощью БПФ. Соответствующие формулы можно найти в [1], в том числе для разных типов граничных условий. В (3) этап вычисления коэффициентов $f^p_{i,j}$ называется прямым преобразованием Фурье, а восстановление по ним исходного вектора f — обратным преобразованием. После прямого БПФ мы приходим к независимым двумерным задачам

$$\overline{A}_p v^p \equiv (A^x + A^y + \lambda_p I)v^p = f^p, \tag{4}$$

в которых пятиточечные матрицы $\overline{A}_p$ могут быть записаны в виде $\overline{A}_p = \overline{A}^x_p + \overline{A}^y_p$, $\overline{A}^x_p = A^x + \lambda_p I$, $\overline{A}^y_p = A^y$.

Решение каждой из этих СЛАУ порядка IJ может быть выполнено с помощью быстрого итерационного неявного метода переменных направлений Писмана–Дугласа–Речфорда

$$\begin{aligned} &(\gamma_n I + \overline{A}^x_p)v^{n-1/2} = (\gamma_n I - \overline{A}^y_p)v^{n-1} + f, \\ &(\gamma_n I - \overline{A}^y_p)w^n = 2(v^{n-1/2} - v^n), \quad v^n = v^{n-1} + w^n, \quad n = 1, 2, \dots, \end{aligned} \tag{5}$$

в котором оптимальные итерационные параметры γ_n вычисляются через границы спектра матриц A^x и A^y (если только они перестановочны, что соответствует полному разделению переменных в исходной задаче (1) [2, 6]). На каждой "полуитерации" серии "одномерных" независимых СЛАУ могут решаться параллельно с помощью экономичных алгоритмов прогонок в направлениях x и y [1, 2].

Если же задача (1) допускает только частичное разделение переменных, что соответствует, например, $r = 0$ и переменным конвективным коэффициентам p и q, то в итерационном методе (5) можно определить оптимальное постоянное значение итерационного параметра $\gamma_n = \gamma_0$, что дает некоторый сходящийся итерационный процесс

$$v^n = Tv^{n-1} + \overline{f}, \quad v^n \xrightarrow[n\to\infty]{} v, \tag{6}$$

где матрица перехода T зачастую может быть симметризована, а $\overline{f}$ — некоторый новый вектор правой части, причем его конкретный вид нам не нужен. Поскольку алгоритм (6) формально определяет некоторую предобусловленную систему

$$\overline{A}v \equiv (I - T)v = \overline{f}, \tag{7}$$

то к ее решению мы можем применить метод сопряженных градиентов

$$\begin{aligned} &r^0 = \overline{f} - \overline{A}v^0 = Tv^0 - v^0 + \overline{f}, \quad p^0 = r^0, \quad n = 1, 2, \dots, \\ &v^n = v^{n-1} + \alpha_n p^{n-1}, \quad r^n = r^{n-1} - \alpha_n\overline{A}p^{n-1}, \quad p^n = r^{n-1} + \beta_n p^{n-1}, \\ &\alpha_n = \frac{(r^{n-1}, r^{n-1})}{(\overline{A}p^{n-1}, p^{n-1})}, \quad \overline{A}p^{n-1} = p^{n-1} - Tp^{n-1}, \quad \beta_n = \frac{(r^n, r^n)}{(r^{n-1}, r^{n-1})}, \end{aligned} \tag{8}$$

в котором используется критерий окончания итераций $(r^n, r^n) \leqslant (\overline{f}, \overline{f})\varepsilon^2$, $\varepsilon \ll 1$. Отметим, что в формулах (8) умножение на матрицу T соответствует проведению одной итерации вида (6). Если же в (7)

матрица $\overline{A}$ несимметрична, то к решению СЛАУ можно применить один из методов полусопряженных направлений ($s = 0$ соответствует алгоритму полусопряженных градиентов, а $s = 1$ — полусопряженных невязок):

$$\begin{aligned} &\overline{r}^{\,0} = \overline{f} - \overline{A}\overline{u}^{\,0}, \quad p^0 = \overline{r}^{\,0}, \quad n = 0, 1, \ldots, \\ &u^{n+1} = u^n + \alpha_n^{(s)} p^n, \quad \overline{r}^{\,n+1} = \overline{r}^{\,n} - \alpha_n^{(s)} \overline{A} p^n, \quad s = 0, 1, \end{aligned} \tag{9}$$

которые минимизируют функционалы $\Phi_s(r^n) = \left(\overline{A}^{-s} r^n, r^n\right)$ в подпространствах Крылова

$$K_{n+1}\left(\overline{r}^{\,0}, \overline{A}\right) = \operatorname{span}\left\{p^0, p^1, \ldots, p^n\right\} = \operatorname{span}\left\{p^0, \overline{A}p^0, \ldots, \overline{A}^{\,n} p^0\right\}$$

при определении итерационных параметров $\alpha_n^{(s)}$ и корректирующих векторов p^n по следующим формулам [3]:

$$\begin{aligned} &\alpha_n^{(s)} = \frac{\left(\overline{A}^{\,s}\overline{r}^{\,n}, \overline{r}^{\,n}\right)}{\left(\overline{A}p^n, \overline{A}^{\,s} p^n\right)}, \quad \beta_n^{(s)} = \frac{\left(\overline{A}^{\,s}\overline{r}^{\,n+1}, \overline{r}^{\,n+1}\right)}{\left(\overline{A}^{\,s}\overline{r}^{\,n}, \overline{r}^{\,n}\right)}, \\ &p^{n+1} = \overline{r}^{\,n+1} + \sum_{k=0}^{n} \beta_{n,k}^{(s)} p^{k+1,l} = p^{n+1,l} + \sum_{k=l}^{n} \beta_{n,k}^{(s)} p^k, \quad l \in [0, 1, \ldots, n], \\ &p^{n+1,0} = \overline{r}^{\,n+1}, \quad p^{n+1,l} = p^{n+1,l-1} + \beta_{n,l-1} p^{l-1}, \quad p^{n+1} = p^{n+1,n}. \end{aligned} \tag{10}$$

Если спектральный радиус матрицы T в (6) близок к единице, т.е. $\rho(T) \leqslant 1 - \delta$, $0 < \delta \ll 1$, и этот итерационный процесс медленно сходится, то его ускорение с помощью крыловских методов дает значительный выигрыш (в [3] представлены оценки числа итераций для симметричного и несимметричного случаев).

Если в системах (4) матрица A^y, например, имеет известный базис из J собственных векторов, то векторы v^p и f^p для каждого p также могут быть разложены по этому базису, в результате чего для новых компонентов Фурье получаем системы "одномерных" уравнений

$$\overline{A}_{p,q} w^{p,q} = f^{p,q}, \quad p = 1, \ldots, K, \quad q = 1, \ldots, J, \tag{11}$$

с трехдиагональными матрицами $\overline{A}_{p,q}$, решение которых на одном процессоре наиболее экономично осуществляется методом прогонки. Однако для распараллеливания предварительно каждую трехдиагональную СЛАУ целесообразно разбить на независимые подсистемы меньших порядков с помощью алгоритма блочной редукции, который фактически реализует прямой метод декомпозиции областей (возможны варианты с явным выделением узлов-разделителей и без них). В некоторых случаях может оказаться эффективным метод четно-нечетной редукции без обратного хода [7]. Наконец, отметим, что безытерационный алгоритм можно получить и с помощью трехкратного преобразования Фурье, если только все матрицы $\overline{A}_{p,q}$ в (11) имеют известный собственный базис. Однако этот подход не представляется целесообразным, поскольку объем вычислений при использовании БПФ для решения трехдиагональной системы нелинейно зависит от ее порядка.

3.2. Универсальные предобусловленные алгоритмы. Рассмотренные выше матричные структуры и алгоритмы носят довольно частный характер, хотя они могут находить применение во многих практических задачах. Однако наиболее распространенные постановки связаны с расчетными областями сложной конфигурации, с переменными коэффициентами, неструктурированными сетками и другими факторами, приводящими к разреженным матрицам с разнообразными "портретами", в которых местоположение ненулевых элементов и их значений можно задать только непосредственным перечислением. Для хранения таких матриц используются типовые сжатые форматы [8, 9], которые значительно экономят оперативную память, но существенно замедляют время выполнения программных реализаций из-за "длинного" доступа к матричным элементам.

Для таких СЛАУ с высокими порядками главным средством решения являются предобусловленные итерационные методы в подпространствах Крылова. К последним относится большое количество алгоритмов сопряженных направлений, бисопряженных и полусопряженных направлений, достаточно полно представленных в книгах [3, 9] и цитируемых в них работах (характерными примерами являются методы (8)–(10)).

Что касается предобусловливания алгебраических систем вида $Au = f$, то оно в общем случае сводится к получению эквивалентной СЛАУ

$$\overline{A}\overline{u} \equiv B^{-1}AC^{-1}Cu = B^{-1}f \equiv \overline{f}, \quad \overline{A} = B^{-1}AC^{-1}, \quad \overline{u} = Cu,$$

где левая и правая предобусловливающие матрицы B и C выбираются из условий экономичной обратимости и минимизации (по возможности) числа обусловленности матрицы $\overline{A}$.

Достаточно простой и экономичной иллюстрацией предобусловливания является метод неполной факторизации в модификации Айзенштата [2, 8]:

$$B = (G - L)G^{-1/2}, \quad C = G^{-1/2}(G - U), \quad G = \frac{1}{\omega} D - \theta S, \quad Se = \left(\frac{1-\omega}{\omega} D + LG^{-1}U\right)e,$$

где G и S — диагональные матрицы, e — вектор из единичных компонент, а ω и θ — итерационные параметры. В этом случае матрица $\overline{A}$ может быть представлена в виде

$$\begin{aligned} &\overline{A} = \left(I - \overline{L}\right)^{-1} + \left(I - \overline{U}\right)^{-1} - \left(I - \overline{L}\right)^{-1}\left(2I - \overline{D}\right)\left(I - \overline{U}\right)^{-1}, \\ &\overline{L} = G^{-1/2}LG^{-1/2}, \quad \overline{U} = G^{-1/2}UG^{-1/2}, \quad \overline{D} = G^{-1/2}DG^{-1/2}, \quad \overline{u} = \left(I - \overline{U}\right)G^{-1/2}u, \end{aligned} \tag{12}$$

и умножение на нее требует практически столько же арифметических операций (I — единичная матрица), что и на исходную матрицу A.

В итерационных процессах данного типа формулы рекуррентного вычисления векторов в крыловских методах распараллеливаются идеально, а вот умножение на предобусловленную матрицу вида (12) требует решения вспомогательных алгебраических систем с треугольными матрицами $\left(I - \overline{L}\right)$ и $\left(I - \overline{U}\right)$, что как раз и оказывается камнем преткновения.

Заметим, что случай тривиального предобусловливания соответствует $B = C - I$, $\overline{A} = A$ и при этом формулы (8) дают классический (явный) метод сопряженных градиентов, который оптимально распараллеливается (в нем легко совместить операции межпроцессорных обменов с выполнением арифметических действий). Однако такой итерационный процесс слишком медленно сходится, и это демонстрирует общее существующее правило: наиболее экономичные алгоритмы характеризуются повышенной логической и информационной сложностью, что снижает эффективность их распараллеливания.

3.3. Блочные итерационные методы. Наиболее естественным путем распараллеливания методов решения краевых задач является декомпозиция расчетной области на некоторое число P подобластей (с налеганием или без него) и построение итерационного процесса, заключающегося в "одновременном" решении независимых задач в подобластях с последующим обменом информацией между ними. С алгебраической точки зрения в простейшем случае мы при этом приходим к крупноблочному методу Якоби.

Например, для $P = 3$ исходная СЛАУ может быть представлена в блочно-трехдиагональном виде в соответствии с рис. 2. Здесь диагональные блоки A_{kk} для $k = 1, 2, 3$ соответствуют задачам в подобластях, а сам алгоритм может быть записан в следующей блочной форме:

$$\begin{aligned} &A = D + C, \quad D = \text{block} - \text{diag}\,(A_{kk}), \\ &Du = f - Cu, \quad C = -L - U, \\ &u^{n+1} = -D^{-1}Cu^n + D^{-1}f = Tu^n + \overline{f}. \end{aligned} \tag{13}$$

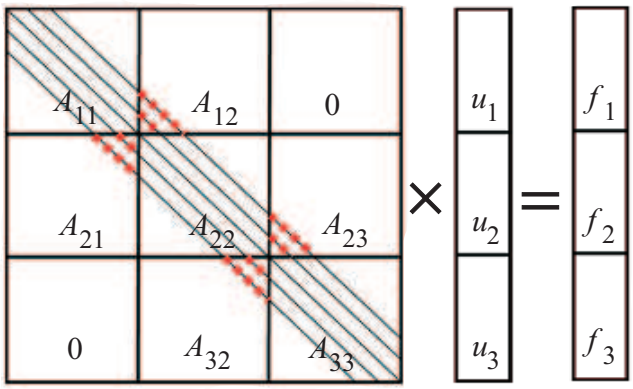

Рис. 2. Пример блочной структуры СЛАУ

При этом на каждой итерации надо обращать матрицу D, что фактически при большом числе узлов сетки, т.е. при высоком порядке подсистем с матрицами A_{kk}, требует применения в каждой подобласти "внутреннего" итерационного процесса, который также может осуществляться каким-то предобусловленным методом в подпространствах Крылова. Таким образом реализуется некий двойной итерационный процесс, возмущенный по отношению к (13), поскольку фактически вместо матрицы вычисляется ее какое-то приближение. Этот фактор, очевидно, замедляет скорость сходимости "внешних" итераций. Повысить эффективность такого крупноблочного метода можно за счет сокращения общего числа внутренних итераций путем выбора специальных динамических критериев их окончания (например, на первых внешних итерациях подсистемы в подобластях решать достаточно грубо).

Другой путь повышения эффективности метода декомпозиции заключается в использовании различных типов "внутренних" краевых условий на границах между подобластями. Это приводит к изменению, в сравнении с (13), матричной формы итерационного процесса: $(D+B)u^{n+1} = f+(B-C)u^{n-1}$. Здесь B —

некоторая матрица с ненулевыми элементами в позициях, соответствующих внутренним околограничным узлам, а ее выбор должен минимизировать спектральный радиус матрицы перехода $(D+B)^{-1}(B-C)$.

Еще один подход к ускорению внешних итераций основан на использовании подобластей с пересечением (налеганием). Однако в этом случае увеличиваются размеры подобластей, что приводит к "удорожанию" внутренних итераций.

В целом методы декомпозиции имеют различные возможности к повышению своей эффективности, но и много открытых вопросов, требующих дальнейших исследований. Отметим, что в [10] и цитируемых там работах изучались перспективные проекционные методы декомпозиции. Что касается отмеченной выше актуальной алгебраической проблемы с седловой точкой $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$, то здесь многочисленные подходы [11] базируются на аппроксимации блочной матричной факторизации

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & -A_{21}A_{11}^{-1}A_{12} \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix}.$$

Не останавливаясь на этой большой и самостоятельной задаче, мы только отметим, что здесь также используются двухуровневые итерационные процессы, основанные на известном алгоритме Узавы, на приближении матричного дополнения Шура $A_{21}A_{11}^{-1}A_{12}$ и на ускорениях в подпространствах Крылова.

4. Некоторые вопросы высокопроизводительных технологий. В данном разделе мы ограничимся только некоторыми замечаниями относительно эффективности распараллеливания и повышения производительности вычислений при решении больших разреженных СЛАУ. Отметим, что общеиспользуемые критерии — ускорение вычислений при решении задачи A на p процессорах и соответствующая эффективность их использования — определяются как $R_p(A) = \dfrac{T_1(A)}{T_p(A)}$, $E_p(A) = \dfrac{R_p(A)}{p}$, где $T_p(A)$ — время реализации задачи на p процессорах. Если $R_p = p$ и $E_p = 1$, то такое ускорение называется линейным и такой случай можно считать идеальным. На практике зачастую приходится мириться с ситуацией $E_p \leqslant 0.5$, хотя случай $E_p < 0.1$ уже обычно рассматривается как "криминальный". Иногда удается достигать и сверхлинейного ускорения $R_p > p$, $E_p > 1$ вследствие неоднородности памяти МВС (например, данные о задаче могут не умещаться в кэше одного процессора, но они могут разместиться в кэшах 20 процессоров, что приведет к резкому сокращению времени расчета).

Время реализации задачи T_p складывается (в первом приближении) из времени выполнения арифметических действий T_p^{a} и времени межпроцессорных обменов T_p^{c}, которые рассматриваются как коммуникационные потери. При синхронной работе процессоров эти времена оцениваются (очень грубо) с помощью выражений

$$T_p^{\mathrm{a}} \approx N_{\mathrm{a}}\tau_{\mathrm{a}}, \quad T_{\mathrm{c}} = N_0\tau_0 + N_{\mathrm{c}}\tau_{\mathrm{c}}, \tag{14}$$

где N_{a} — общее число арифметических операций, выполняемых одним процессором, N_0 — количество информационных обменов, N_{c} — общий объем передаваемых данных (в нашем случае — это число пересылаемых вещественных чисел с двойной стандартной точностью), а τ_{a}, τ_0, τ_{c} — среднее время выполнения одной арифметической операции, время задержки (настройки) при реализации одной передачи и время пересылки одного числа соответственно.

Формулы (14) фактически представляют собой утрированную модель машинного вычислительного процесса. Например, на компьютере Пентиум-4 время деления составляет $\tau_{\mathrm{d}} = 38\tau$, умножения — $\tau_m = 7\tau$, сложения — $\tau_{\mathrm{ad}} = 5\tau$, где τ — длительность машинного такта. На многоядерных процессорах ситуация с оценками длительности отдельных арифметических операций значительно усложняется. Кроме того, необходимо учитывать возможности совмещения вычислений с информационными обменами, а также асинхронной работы различных процессоров. Учесть возможности автоматического формирования конвейерных вычислений, синхронизации работы различных арифметических устройств и особенности доступа к данным из разных уровней кэша "на руках" практически невозможно. Поэтому экспериментальные исследования эффективности распараллеливания алгоритмов и их отображения на архитектуру МВС приходится проводить практически "вслепую" в условиях ненадежности измерений компьютерной производительности при их зависимости от условий коллективной эксплуатации конкретной ЭВМ. Здесь, конечно, необходимо квалифицированное использование профилировщиков и других системных инструментариев для выявления "узких" мест вычислительного процесса.

Важно отметить, что повышения производительности программного кода можно добиться не в связи с распараллеливанием, а за счет использования различных режимов компиляции и вычислительных инструментариев. Например, библиотека MKL (Mathematical Kernel Library) компании Intel содержит реализованные экономично на ассемблере функции исполнения некоторых векторных и матрично-векторных

операций, которые позволяют значительно ускорить время решения СЛАУ.

В заключение подчеркнем существенное различие между понятиями "ускорение времени решения задачи" и "ускорение алгоритма", поскольку, как уже отмечалось, зачастую наиболее продвинутые в теоретическом отношении методы хуже всего распараллеливаются. Поэтому выработка общей методологии эффективного распараллеливания никак не противоречит проведению "штучных" исследований, на уровне инженерного искусства, по оптимизации производительности конкретных алгоритмов на реальной компьютерной платформе.

СПИСОК ЛИТЕРАТУРЫ

1. *Самарский А.А., Николаев Е.С.* Методы решения сеточных уравнений. М.: Наука, 1978.
2. *Ильин В.П.* Методы конечных разностей и конечных объемов для эллиптических уравнений. Новосибирск: Изд-во ИВМ СО РАН, 2000.
3. *Ильин В.П.* Методы и технологии конечных элементов. Новосибирск: Изд-во ИВМиМГ СО РАН, 2007.
4. *Богачев К.Ю.* Основы параллельного программирования. М.: БИНОМ, 2003.
5. *Ильин В.П.* Параллельные алгоритмы для больших прикладных задач: проблемы и технологии // Автометрия. 2007. **43**, № 2. 3–21.
6. *Ильин В.П., Кныш Д.В.* Параллельные алгоритмы решения разделяющихся краевых задач // Санкт-Петербург: Изд-во Политехн. ун-та (СПбПГУ), 2008. 107–118.
7. *Ильин В.П., Кузнецов Ю.И.* Трехдиагональные матрицы и их приложения. М.: Наука, 1985.
8. *Andreeva M.Yu., Il'in V.P., Itskovich E.A.* Two solvers for nonsymmetric SLAE // Bull. NCC. Ser. Num. Anal., 2003. Iss. 12. 1–16.
9. *Saad Y.* Iterative methods for sparse linear systems. NY: PWS Publish., 1996.
10. *Ильин В.П.* О методах сопряженных и полусопряженных направлений с предобуславливающими проекторами // ДАН. 2008. **419**, № 3. 303–306.
11. *Benzi M., Golub G., Leisen J.* Numerical solution of saddle point problems // Acta Numer. 2005. **14**. 1–137.

Поступила в редакцию
10.03.2009