Применение параллельных алгоритмов для решения системы линейных алгебраических уравнений с ленточной матрицей итерационными методами на кластерной системе

Демешко И.П., Акимова Е.Н., Коновалов А.В.

Представлены результаты применения параллельных итерационных алгоритмов решения системы линейных алгебраических уравнений с ленточной матрицей, получаемой в двумерных упруго-пластических задачах. Реализация алгоритмов осуществлена на многопроцессорной вычислительной системе МВС-1000/32 на языке Си с помощью библиотеки параллельного программирования MPI.

1. Введение

Для решения упруго-пластических задач с большими пластическими деформациями методом конечных элементов нагрузка в виде перемещения инструмента прикладывается малыми шагами Δh. В процессе решения необходимо выполнить около тысячи шагов по нагрузке, причем на каждом шаге нужно около десяти раз решить систему линейных алгебраических уравнений (СЛАУ). Таким образом, решение СЛАУ занимает до половины всего времени решения задачи. Применение параллельных вычислений для решения СЛАУ может позволить значительно сократить время решения задачи.

2. Описание задачи

В качестве примера рассматривается решение методом конечных элементов двумерной задачи сжатия цилиндра плоскими плитами. Материал цилиндра упруго-пластический, изотропный и изотропно-упрочняемый. На контакте с плитами приняли закон трения Кулона. Шаг нагрузки Δh выбирали так, чтобы отношение $\Delta h / h$ (h - высота цилиндра) не превышало предела упругости по деформации, в нашем случае 0,002, что обеспечивало устойчивость вычислительной процедуры. Величину относительного сжатия цилиндра приняли равной 0,5. На каждом шаге нагрузки решение основывается на принципе виртуальной мощности в скоростной форме [1]:

$$\int_V (\sigma + \Delta t \dot{\sigma}) \cdot\cdot \nabla h dV + \int_\Sigma (P + \Delta t \dot{P}) \cdot h d\Sigma = 0 \qquad (1)$$

со следующими определяющими соотношениями, полученными в работе [2]:

$$\dot{\sigma} = \lambda \dot{\Theta} I + 2(\lambda \Theta + \mu) D - \nabla \mathrm{v} \cdot \sigma - \sigma \cdot \nabla \mathrm{v}^T - JbS \ ,$$

$$S = \sigma - \sigma_0 I \ ,$$

$$\sigma_0 = K\Theta \ ,$$

$$K = \lambda + \tfrac{2}{3}\mu \ ,$$

$$D = 0.5(\nabla \mathrm{v} + \nabla \mathrm{v}^T) \ ,$$

$F(S, k) = 0.5 S \cdot\cdot S - k^2 = 0$, - условие текучести Мизеса (2)

$$b = \left[\left[\mu \left(1 - \frac{2}{3}\Theta \right) S - S \cdot S \right] \cdot\cdot D - \frac{\mu}{3} \dot{\Theta}\, S \cdot\cdot I \right] \Bigg/ \left[k^2 \left(1 + \frac{1}{\mu} \frac{dk}{d\chi} \right) \right] .$$

Здесь σ - тензор напряжений Коши; P - плотность поверхностных сил; Δt - промежуток времени для шага приращения нагрузки; h - вариация кинематически допустимых полей скоростей; ∇ - набла-оператор; V, Σ - объем и поверхность тела соответственно; dV, $d\Sigma$ - элементы объема и площади поверхности цилиндра соответственно; λ, μ - коэффициенты

Ламе; I - единичный тензор; точкой и двумя точками обозначено соответственно скалярное и двойное скалярное произведение тензоров; точкой сверху обозначена полная производная по времени; ∇V - градиент скорости перемещений; D - тензор скоростей деформаций; σ_0 - среднее нормальное напряжение, K - объемный модуль упругости; Θ - относительное изменение индивидуального объема бесконечно малой частицы среды; k - напряжение текучести; $J = 0$ при $F < 0$ или при $F = 0$, $S \cdot\cdot \dot{S} \le 0$; $J = 1$ при $F = 0$, $S \cdot\cdot \dot{S} > 0$; χ - параметр упрочнения, в силу малости упругих деформаций $\dot{\chi} = \sqrt{2D \cdot\cdot D}$.

Равенство (1) с помощью конечно-элементной аппроксимации сводится к системе линейных алгебраических уравнений (СЛАУ):

$$Ax = b, \tag{3}$$

где A, x, b - соответственно матрица, вектор решения и вектор правой части системы. Матрица A имеет ленточный вид.

Решение задачи сжатия цилиндра на шаге нагрузки состоит из трех основных этапов:

1) подготовка матрицы A,
2) решение СЛАУ,
3) вычисление напряженно-деформированного состояния в конце шага нагрузки.

Этап 1 выполняется один раз, а этапы 2 и 3 – десять раз, поскольку выполнение условия текучести Мизеса (2) достигается итерационно по шагам нагрузки. Анализ показал, что на шаге нагрузки время вычисления этих этапов с учетом итераций составляет соответственно 15%, 45% и 40%.

Для решения СЛАУ (3) использовали итерационные методы [3] со следующими рекуррентными соотношениями:

1. Метод простой итерации (МПИ):

$$z^{k+1} = z^k - \frac{1}{\lambda_{\max}}\left[(A + \alpha_k E)z^k - b\right],\ z_0 = 0,$$

где $\lambda_{\max}$ - максимальное собственное значение, E - единичная матрица, α_k - параметр регуляризации.

2. Метод минимальных невязок (ММН):

$$z^{k+1} = z^k - \frac{\left(A\left(Az^k - b\right),\ Az^k - b\right)}{\left\| A\left(Az^k - b\right)\right\|^2}\left(Az^k - b\right),\quad z_0 = 0.$$

3. Метод наискорейшего спуска (МНС):

$$z^{k+1} = z^k - \frac{\left\| A^T Az^k - A^T b \right\|^2}{\left\| A\left(A^T Az^k - A^T b\right)\right\|^2} A^T\left(Az^k - b\right), z_0 = 0.$$

4. Метод сопряженных градиентов (МСГ) [4]:

$$z^1 = z^0 - \frac{\left\| r^0 \right\|^2 \cdot \left(B^k z^0 - b\right)}{\left(B^k r^0, r^0\right)},\ r^0 = B^k z^0 - b,$$

где $z^{k+1} = z^k - \gamma_k\left(B^k z^k - b\right) + \beta_k\left(z^k - z^{k-1}\right)$,

$$\gamma_k = \frac{\left\| r^k \right\|^2 \left(B^k p^k, p^k\right) - \left(r^k, p^k\right)\left(B^k r^k, r^k\right)}{\left(B^k r^k, r^k\right)\left(B^k p^k, p^k\right) - \left(B^k r^k, p^k\right)},$$

$$r^k = B^k z^k - b,$$

$$\beta_k = \frac{\left\| r^k \right\|^2 \left(B^k r^k, p^k\right) - \left(r^k, p^k\right)\left(B^k r^k, r^k\right)}{\left(B^k r^k, r^k\right)\left(B^k p^k, p^k\right) - \left(B^k r^k, p^k\right)},$$

$$p^k = z^k - z^{k-1},\ k = 0, 1, 2\ldots$$

В приведенных формулах k - номер итерации.

Условием остановки итерационного процесса является: $\|Ax-b\|/\|B\| \le \varepsilon$, где ε - заданная точность решения. Выбор ε определялся из соотношения: $(\|x_\Gamma\|-\|x_H\|)/\|x_\Gamma\|$, где $\|x_\Gamma\|$- норма решения, полученного методом Гаусса, $\|x_H\|$- норма решения, полученного итерационным методом. Из вычислительных экспериментов выбрали $\varepsilon \le 0.1$.

В качестве начального приближения решения СЛАУ(3) принимали вектор ее правой части. На каждой следующей итерации за начальное приближение брали решение системы, полученное на предыдущей итерации.

3. Описание алгоритма

Алгоритм распараллеливания решения СЛАУ(3) итерационными методами основан на преобразовании ленточной матрицы в вертикальную полосу и разбиении ее горизонтальными полосами на m блоков по числу процессоров. Схема алгоритма представлена на рис. 1.

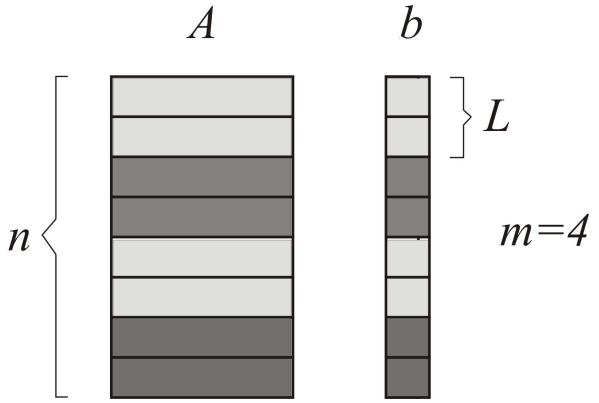

Рис. 1. Схема параллельного итерационного алгоритма решения СЛАУ

Вектор решения и вектор правой части СЛАУ разбивается на m частей так, чтобы $n = m \cdot L$, где n – размерность системы уравнений, L - число строк в одном блоке. Каждый процессор вычисляет свою часть вектора решения и передает вычисленные значения всем остальным процессорам.

4. Результаты

Распараллеливание и численную реализацию задачи выполнили на многопроцессорной системе МВС-1000/32, установленной в ИММ УрО РАН. Использовали язык Си и библиотеку параллельного программирования MPI[5]. МВС-1000/32 – многопроцессорный вычислительный комплекс кластерного типа, имеющий 32 процессора.

Для проведения вычислительного эксперимента на многопроцессорной вычислительной системе (МВС) запускали программу решения СЛАУ разработанными параллельными алгоритмами. При запуске программа считывала всю заранее подготовленную информацию о СЛАУ. Эксперименты были проведены на матрицах, соответствующих размерности сетки до 60x60.

На рис.2 приведены результаты решения СЛАУ(3), с помощью рассмотренных итерационных методов. Результаты представлены для матрицы с шириной ленты 171 и количеством переменных 3362, полученной для разбиения области сеткой 40x40. Видно, что методы минимальных невязок, простой итерации, наискорейшего спуска показали близкие результаты при одинаковых условиях решения задачи. Метод сопряженных градиентов требует большее число итераций по сравнению с остальными рассмотренными методами и большее время для решения СЛАУ.

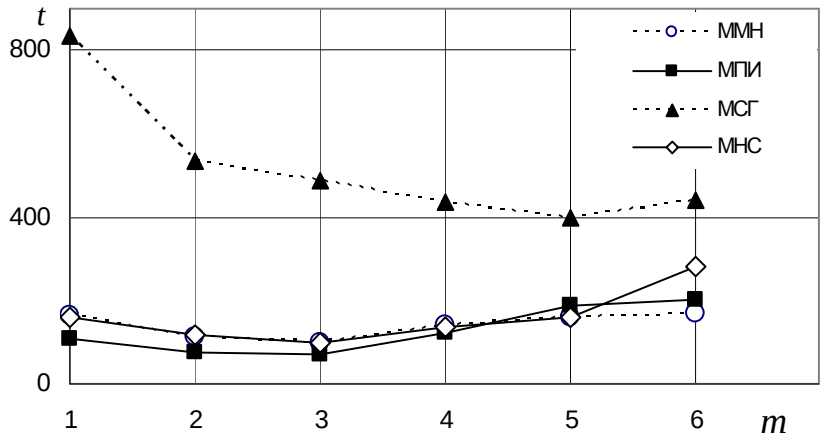

Рис. 2. Зависимость времени вычисления t СЛАУ параллельными итерационными алгоритмами от числа процессоров m

На рис.3 представлена зависимость ускорения a параллельных алгоритмов решения СЛАУ(3) итерационными методами от числа процессоров m для сеток разной размерности. Результаты приведены для случая решения СЛАУ методом простой итерации, так как графики ускорения параллельных алгоритмов ММН, МПИ и МНС расположены близко друг к другу и показывают наилучшие результаты при решении рассматриваемой задачи.

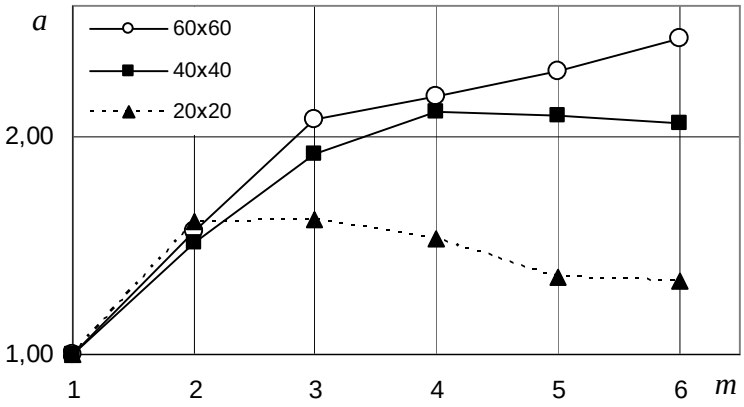

Рис. 3. Зависимость ускорения a параллельного алгоритма решения СЛАУ методом простой итерации от числа процессоров m для сеток разной размерности.

Для всех рассмотренных итерационных методов с увеличением размерности матрицы A эффективность распараллеливания возрастает. Это объясняется тем, что при увеличении размерности сетки вычислительная трудоемкость для части матрицы, находящейся на одном процессоре, увеличивается. При этом объем передаваемых данных между процессорами увеличивается незначительно. В результате отношение времени вычислений к времени передачи данных растет с увеличением размерности матрицы. Следовательно, с увеличением размерности матрицы можно добиться большего ускорения на большем числе процессоров при распараллеливании итерационных алгоритмов.

5. Выводы

При решении двумерных упруго-пластических задачах с большими пластическими деформациями использование разработанных параллельных алгоритмов решения СЛАУ(3)

итерационными методами позволяет сократить время решения задачи при использовании МВС кластерного типа. При этом с увеличением размерности задачи эффективность применения параллельных алгоритмов растет.

Работа выполнена в рамках программы Президиума РАН "Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация".

Литература

1. Поздеев А. А., Трусов П.В., Няшин Ю.И. Большие упруго-пластические деформации. М: Наука, 1986, 232с.
2. Коновалов А. В. Определяющие соотношения для упругопластической среды при больших пластических деформациях // Известия РАН. Механика твердого тела. 1997. № 5. С. 139-149.
3. Васин В.В., Ерёмин И.И. Операторы и итерационные процессы Фейеровского типа. Теория и приложения. – Екатеринбург, 2005. 210с.
4. Фадеев В.К., Фадеева В.Н. Вычислительные методы линейной алгебры. – Москва: Гос. издат. физико-математической литературы, 1963, 734с.
5. MPI: A Message-Passing Interface Standard. Message Passing Interface Forum. – Version 1.1. 1995. – http://www-unix.mcs.anl.gov/mpi