

Нестеренко А.Н., Химич А.Н., Яковлев М.Ф.

## **НЕКОТОРЫЕ ВОПРОСЫ РЕШЕНИЯ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С РАСПРЕДЕЛЕННОЙ ПАМЯТЬЮ**

### Введение

В работе рассматриваются вопросы организации вычислений при решении систем нелинейных уравнений высокого порядка на многопроцессорных вычислительных системах с распределенной памятью MIMD-архитектуры (multiple instruction, multiple data). Реализация методов решения задач данного класса на компьютерах MIMD-архитектуры предполагает следующее: массив обрабатываемых данных (функции и элементы матрицы Якоби) размещается в распределенной памяти компьютера (distributed memory); т. е. в локальной памяти каждого процессора; в каждом процессоре одновременно и независимо реализуется собственная программа вычислений; декомпозиция данных по процессорам выполняется, исходя из сбалансированности всех процессоров, коммуникационные функции и синхронизация работы процессоров осуществляется средствами MPI (message-passing interface).

Ниже для такого типа многопроцессорной системы показано, как можно автоматически распределять обработку информации по процессорам, определены критерии окончания итерационных процессов решения систем нелинейных уравнений, обеспечивающие получение решения с заданной точностью, оценена погрешность решения систем нелинейных уравнений в условиях приближенно заданных исходных данных и приведены времена решения системы нелинейных уравнений двухтысячного порядка на одном из компьютеров MIMD-архитектуры.

### 1. Общие подходы к решению систем нелинейных уравнений

Решение систем нелинейных уравнений наряду с решением систем линейных алгебраических уравнений, вычислением собственных значений и векторов, интегрированием задач с начальными условиями для систем обыкновенных дифференциальных уравнений является одной из часто встречающихся в вычислительной практике задач. Основные алгоритмы решения этих задач при всем различии в их названии базируются в той или иной степени на методе Ньютона. Различие между всеми так называемыми квазиньютоновскими методами в основном состоит только в способе вычисления приближенного значения матрицы Якоби: это могут быть как разностные формулы, так и

рекуррентные формулы вычисления приближенной матрицы Якоби или обратной к ней на последовательности итераций. Тем не менее все квазиньютоновские методы обеспечивают сверхлинейную скорость сходимости итерационных процессов вблизи решения при условии, что начальное приближение лежит в области притяжения решения. Цель данной работы – показать, как можно реализовать основные алгоритмы решения систем нелинейных уравнений на компьютерах МІМД-архитектуры, чтобы эффективно решать эти задачи.

Пусть задана система  $n$  нелинейных уравнений

$$f(x) = 0, \quad (1)$$

где  $x = (x_1, x_2, \dots, x_n)^T$ ,  $f(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$  - вектор искомого решения и вектор-функция соответственно. Если  $H = \left\{ \frac{\partial f_i}{\partial x_j} \right\}_{i,j=1}^n$  - матрица Якоби системы (1) (или не-

которое приближение к ней), то итерационный процесс нахождения решения, реализующий метод Ньютона при заданном начальном приближении  $x^{(0)}$ , можно записать в виде

$$H^{(k)} w^{(k)} = -f(x^{(k)}), \quad (2)$$

где  $w^{(k)} = x^{(k+1)} - x^{(k)}$  - поправка,  $H^{(k)} = H(x^{(k)})$ ,  $k=0, 1, \dots$  - номер итерации,

$$x^{(k+1)} = x^{(k)} + w^{(k)}. \quad (3)$$

Для решения таких задач, кроме начального приближения, задается дополнительно область, в которой ищется решение  $D = \{a_i \leq x_i \leq b_i, \quad i = 1, 2, \dots, n\}$ , и требуемая точность получения приближения к решению системы. При этом  $x^{(0)} \in D$ .

Как видно из формул (2), (3), на каждой итерации необходимо решать систему линейных алгебраических уравнений вида (2), вычисляя при этом значение вектор-функции и матрицу Якоби. О критериях окончания итерационного процесса, которые обеспечивают заданную точность получения приближения к решению системы, речь пойдет ниже.

## 2. Способ задания уравнений для МІМД-компьютеров

Для решения систем нелинейных уравнений на МІМД-компьютерах необходимо задавать систему уравнений специальным образом, который бы обеспечивал максимально возможное распараллеливание операций на каждой итерации при вычислении вектор-функции, приближения к матрице Якоби и решению системы уравнений (2).

При этом следует иметь в виду, что только в очень редких случаях формулы вычисления компонент вектор-функции существенно отличаются одна от другой. Как правило, в этих случаях системы уравнений не могут быть большого порядка и поэтому решать их на МММД-компьютерах нецелесообразно. В большинстве случаев запись компонент вектор-функции подчиняется некоторому закону построения и такие системы могут иметь достаточно большой порядок. Решение систем больших порядков целесообразно распараллеливать для уменьшения времени получения результатов.

Опишем один из возможных способов задания вектор-функции на языке Си.

Например, для решения системы

$$\sum_{j=1}^n x_j - 0.5(3n+1) + 2x_i^2 - 2 \left( 1 + 2 \frac{i}{n} + \left( \frac{i}{n} \right)^2 \right) = 0, i=1, 2, \dots, n$$

эта функция имеет вид

```
void f (int n, int l, int m, double *x, double *y)
{ double ss, sss, fv;
  int i, j;
  for( i=l; i<m; i++)
  {   ss=0.e0;
      sss=i+1;
      for( j=0; j<n; j++)   ss+=x[j];
      fv=ss-0.5e0*(3.e0*n+1.e0)+2.e0*x[i]*x[i]-
          2.e0*(1.e0+2.e0*sss/n +(sss/n)*(sss/n) );
      y[i-1] = fv;
  }
}
```

Здесь обозначено:

$n$  – порядок системы нелинейных алгебраических уравнений,

$l$  – начальный номер уравнения в отдельном процессоре, которое используется для вычисления вектор-функции и соответствующих строк матрицы, аппроксимирующей матрицу Якоби,

$m-1$  – конечный номер уравнения в отдельном процессоре, которое используется для вычисления вектор-функции и соответствующих строк матрицы, аппроксимирующей матрицу Якоби,

$x$  – вектор переменных,

$y$  – значение вектор-функции в точке  $x$ .

Величины  $l$  и  $m$  для каждого процессора вычисляются перед обращением к программе вычисления вектор-функции.

Если система нелинейных уравнений имеет нерегулярную структуру, то придется каждое уравнение пометить собственной меткой типа «case  $k$ :» ( $k$  – целое число) и обеспечивать обращение к функции по описанному выше правилу, используя в программе оператор типа «switch ( $i$ )». Возможны и некоторые другие способы задания вектор-функции, но обращение к их вычислению должно быть организовано так, как показано в приведенном примере.

Такой способ задания вектор-функции имеет следующие достоинства:

- 1) как правило, системы нелинейных уравнений большого порядка имеют регулярную структуру, и тогда фактически необходимо написать заголовок цикла и уравнение, зависящее от параметра цикла;
- 2) при скрытом параллелизме [1] вычисление вектор-функции автоматически распределяется по выбранному числу процессоров; способ распределения опишем ниже;
- 3) приближение к матрице Якоби также вычисляется по блокам, и поэтому для решения системы линейных алгебраических уравнений вида (2) применяется один из вариантов блочного метода Гаусса [2], [3].

Автоматическое распределение вычисления компонент вектор-функции системы нелинейных уравнений на  $p$  блоков ( $p$  - количество используемых процессоров) выполняется следующим образом: вычисляется отношение  $\left\lceil \frac{n}{p} \right\rceil = q$ , где  $\lceil a \rceil$  целая часть  $a$ , затем вычисляется величина  $s = p(q+1) - n$ ; тогда последние  $s$  процессоров будут обрабатывать блоки по  $q$  уравнений, а первые  $p-s$  процессоров - блоки по  $q+1$  уравнению.

### 3. Точность решения системы нелинейных уравнений

Для получения решения системы нелинейных уравнений с заданной точностью  $\varepsilon$  докажем две теоремы.

Теорема 1. Если  $x$  – точное решение системы (1), то при выполнении в итерационном процессе (2) условия

$$\|f(x^{(k)})\| \leq \frac{\varepsilon}{\|H^{-1(k)}\|} \quad (4)$$

выполняется неравенство  $\|x^{(k)} - x\| \leq \varepsilon$ , где  $\varepsilon$  - задаваемая пользователем точность получения решения, а  $H^{(k)-1}$  - матрица, обратная по отношению к матрице Якоби.

Доказательство. Разложим функцию  $f(x)$  в окрестности точки  $x^{(k)}$  в ряд Тейлора с точностью до малых величин второго порядка:

$$f(x) = f(x^{(k)}) + H^{(k)}(x - x^{(k)}) + (\dots).$$

Так как  $f(x) = 0$ , то  $x^{(k)} - x = H^{-1(k)} f(x^{(k)}) - H^{-1(k)}(\dots)$ . Тогда справедливо неравенство  $\|x^{(k)} - x\| \leq \|H^{-1(k)}\| \|f(x^{(k)})\| + \|H^{-1(k)}(\dots)\|$ . Это неравенство будет автоматически выполнено, если потребовать выполнение неравенства  $\|H^{-1(k)}\| \|f(x^{(k)})\| \geq \|x^{(k)} - x\|$ . Если теперь потребовать, чтобы левая часть последнего неравенства не превышала заданного значения  $\varepsilon$ , то и норма разности между приближенным и точным значениями решения не будет превосходить  $\varepsilon$ . Отсюда следует условие (4) окончания итерационного процесса, обеспечивающее заданную пользователем точность решения.

Теорема 2. Если в итерационном процессе (2) выполнено условие

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{\varepsilon}{\|H^{-1}(x^{(k)})\| \|H(x^{(k)})\|}, \quad (5)$$

то решение системы нелинейных уравнений получено с точностью  $\varepsilon$ , то есть выполнено неравенство  $\|x^{(k)} - x\| \leq \varepsilon$ .

Доказательство. При использовании метода Ньютона решается задача (2)

$$H(x^{(k)})(x^{(k+1)} - x^{(k)}) = -f(x^{(k)}),$$

откуда можно получить неравенство  $\|f(x^{(k)})\| \leq \|H(x^{(k)})\| \|x^{(k+1)} - x^{(k)}\|$ . Если теперь потре-

бовать выполнение условия  $\|H(x^{(k)})\| \|x^{(k+1)} - x^{(k)}\| \leq \frac{\varepsilon}{\|H^{-1}(x^{(k)})\|}$ , то автоматически, со-

гласно теоремы 1, будет выполнено неравенство  $\|x^{(k)} - x\| \leq \varepsilon$ . Отсюда получается еще одно условие окончания итерационного процесса

$$\|x^{(k+1)} - x^{(k)}\| \leq \frac{\varepsilon}{\|H^{-1}(x^{(k)})\| \|H(x^{(k)})\|}, \quad (6)$$

которое обеспечит заданную точность решения. Тем самым доказательство теоремы завершено.

Условие (6) окончания итерационного процесса является более жестким, нежели условие (4). Потому для окончания итерационного процесса целесообразно пользоваться именно условием (4).

#### 4. Решение систем нелинейных уравнений в условиях приближенных исходных данных

При моделировании реальных процессов на ЭВМ с помощью систем нелинейных уравнений часто приходится иметь дело с приближенными исходными данными. Приближенность данных может быть обусловлена следующими причинами:

- 1) погрешностями коэффициентов системы, так как они, являясь результатом различных измерений, не могут быть точными;
- 2) погрешностями задания функций; эти погрешности вызваны тем, что задаваемые нелинейные уравнения являются часто некоторым приближением к реальным нелинейным уравнениям; кроме того, часто реальные нелинейные уравнения приближаются более простыми (которые аппроксимируют реальные нелинейные уравнения) нелинейными уравнениями с целью экономии арифметических операций при каждом вычислении функций;
- 3) применением численного метода решения и округлением чисел при вычислениях;
- 4) получением системы уравнений с использованием дискретизации задач различного типа по пространственным переменным.

Поэтому при оценке точности решения необходимо учитывать приближенность исходных данных.

Теорема 3. Если вместо точной системы (1) решается приближенная система

$$\bar{f}(\bar{x}) = 0, \quad (7)$$

для которой выполняется неравенство

$$\|f(v) - \bar{f}(v)\| \leq \delta, \quad (8)$$

где  $v$  – любой вектор, то при выполнении в итерационном процессе вида (2) неравенства

$\|\bar{f}(x^{(k)})\| \leq \frac{\varepsilon}{\|\bar{H}^{-1(k)}\|}$  точность полученного решения оценивается формулой

$$\|\bar{x}^{(k)} - x\| \leq \varepsilon + \|\bar{H}^{-1}\| \delta. \quad (9)$$

Доказательство. Запишем разложение  $\bar{f}(x) = \bar{f}(\bar{x}) + (x - \bar{x})\bar{H}'(\bar{x})$ . Так как  $\bar{f}(\bar{x}) = 0$  в силу (7), то  $\|x - \bar{x}\| \leq \|\bar{H}^{-1}(\bar{x})\| \|\bar{f}(x)\|$ .

Теперь оценим норму отклонения полученного приближенного решения  $x^{(k)}$  от точного решения системы (1)

$$\begin{aligned} \|x^{(k)} - x\| &\leq \|x^{(k)} - \bar{x}\| + \|\bar{x} - x\| \leq \varepsilon + \|\bar{H}^{-1}(\bar{x})\| \|\bar{f}(x)\| \leq \\ &\leq \varepsilon + \|\bar{H}^{-1}(\bar{x})\| (\|\bar{f}(x) - f(x)\| + \|f(x)\|) \leq \varepsilon + \|\bar{H}^{-1}(\bar{x})\| \delta. \end{aligned}$$

Здесь использовано тождество  $f(x) = 0$ . Тем самым теорема доказана.

## 5. Практическое применение результатов

Хотя приведенное выше условие окончания итераций (4), обеспечивающее заданную точность приближения, и оценка погрешности (9) решения в условиях приближенных исходных данных основывались на методе Ньютона, тем не менее эти результаты справедливы для любых итерационных методов решения систем нелинейных уравнений и задач минимизации, в том числе, например, для квазиньютоновских процессов, описанных в [4]-[7]. Однако при практической реализации итерационных процессов необходимо поступать следующим образом: сначала в итерационном процессе проверяется условие типа  $\|f(x^{(k)})\| \leq \varepsilon$  и только по достижению этого неравенства производится переход к проверке условия (4).

В таблице приведены результаты решения нелинейной системы уравнений двухтысячного порядка на MIMD-компьютере различными методами.

Метод Ньютона реализован так, как описано выше. Метод Бурдакова характерен тем, что он имеет глобальную сходимость. В методе Dennis-More обратная матрица Якоби вычисляется по рекуррентной формуле. Из таблицы видно, что для решения системы двухтысячного порядка практически любым из приведенных методов необходимо использовать всего семь процессоров.

Количество процессоров	Время решения задачи (в сек.) Метод Бурдакова	Время решения задачи (в сек.) Метод Dennis-More	Время решения задачи (в сек.) Метод Ньютона
1	237,72	141,60	235,55
2	221,26	103,89	218,54
3	156,91	75,31	156,78
4	122,94	64,72	145,14
5	101,92	53,41	121,69
6	89,07	50,42	103,41
7	79,61	42,37	82,78
8	80,46	38,31	83,29
9	78,19	36,51	81,40
10	72,38	34,82	79,03
11	71,98	33,93	76,36
12	74,71	32,43	73,31

## 6. Список литературы

1. Молчанов И.Н., Галба Е.Ф., Попов А.В., Химич А.Н., Яковлев М.Ф. Скрытый параллелизм- составная часть программных средств для параллельных компьютеров. // УкрПРОГР'98. Київ. 1998. С. 267-270.
2. Численные методы для многопроцессорного вычислительного комплекса ЕС/ Михалевич В.С., Бик Н.А., Брусникин Б.Н.,..., Химич А.Н. и др./ Под редакцией И.Н. Молчанова.- М.: Издание ВВИА им. проф. Н.Е. Жуковского, 1986.- 401 с.
3. <http://www.netlib.org/scalapack/>
4. О.П. Бурдаков Некоторые глобально сходящиеся модификации метода Ньютона для решения систем нелинейных уравнений. //ДАН СССР. 1980. т. 254, № 3, с. 521-523.
5. J.E. Dennis, Jr. and Jorge More. Quasi-Newton Methods, Motivation and Theory. // SIAM Review.- v. 19, № 1.-January 1977.- P. 46-89.
6. Broyden C.G. A class of methods for solving nonlinear simultaneous equations// Math. Comput.- 1965.- v. 19. №92.- P. 577-593.
7. Powell M.J.D. A new algorithm for unconstrained optimization. //Nonlinear Programming./Eds. Rosen J.B. et al.- New York: Acad. Press, 1970.-P. 31-65.