

О. ВААРМАНН

О НЕКОТОРЫХ ПАРАЛЛЕЛЬНЫХ ИТЕРАЦИОННЫХ МЕТОДАХ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

(Представил А. Хумал)

В последнее время в связи с необходимостью повышения эффективности решения математических задач на ЭВМ усилился интерес к методам распараллеливания вычислений, которые заключаются в преобразовании алгоритмов и данных к виду, удобному для обработки на распределительных вычислительных системах. Распараллеливание вычислений может быть в принципе осуществлено на различных уровнях [1]: на уровне построения физических моделей, объектов или процессов исследуемого явления, метода решения, алгоритма, программы, обмена информации в машине, ввода и вывода информации. В центре внимания настоящей работы — распараллеливание на уровне метода (или алгоритма) решения задачи, которое проводится главным образом путем модифицирования известных последовательных методов. Наибольший успех в разработке методов, организующих параллельные вычисления, достигнут в линейной алгебре (см., напр., [2]). Работ же по этой тематике применительно к решению нелинейных уравнений пока мало. Рассматриваемые ниже методы решения нелинейных уравнений являются методами линеаризации. Поэтому для решения линейных проблем можно, естественно, использовать методы и способы распараллеливания вычислений, разработанные для линейной алгебры. К тому же приемы распараллеливания мы рассматриваем с учетом характера вычислительной схемы, специально предназначенной для решения нелинейных задач.

1. Пусть требуется решить систему нелинейных уравнений

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0, \\ &\dots \\ f_n(x_1, \dots, x_n) &= 0 \end{aligned} \tag{1}$$

или уравнение

$$F(x) = 0, \tag{2}$$

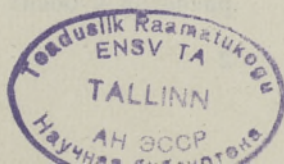
где $x = (x_1, \dots, x_n)^T$ и $F(x) = (f_1(x), \dots, f_n(x))^T$ — требуемое число раз дифференцируемая функция в рассматриваемой области S .

Методы (вернее схемы вычислений), применяемые ниже для решения уравнения (2), строятся с учетом того, что перемножение матрицы на матрицу требует $O(n^3)$ операций, а произведение матрицы на вектор — $O(n^2)$ операций. Этот факт был использован многими авторами (см. напр., [3-6]).

Известно, что если U — некоторая несингулярная $n \times n$ -матрица и X_0 — некоторое приближение к U^{-1} , тогда итерационный процесс

$$X_{k+1} = X_k [I + T_k + \dots + T_k^{q-1}], \quad q \geq 2, \quad k = 0, 1, \dots, \tag{3}$$

где I — единичная матрица и $T_k = I - UX_k$, сходится при условии $\|I - UX_0\| < 1$ к U^{-1} , причем $\|T_{k+1}\| \leq \|T_k\|^q$ [7].



Очевидно, что при $q \rightarrow \infty$ формула (3) принимает вид (ср. с [8])

$$X_{h+1} = X_h (UX_h)^{-1}. \quad (4)$$

Заметим, что основной операцией в формуле (3) является перемножение матриц, которое можно получить в виде скалярных произведений векторов, причем округляя не каждое слагаемое, а лишь результат. Если операции при этом выполнять с двойной точностью, то ошибка вычисления будет совпадать с ошибкой единичного округления [9].

Если $X_h \approx U^{-1}$, то $UX_h \approx I$ и, следовательно, матрица UX_h является диагонально доминирующей. Поэтому операция обращения матрицы UX_h устойчива. Если перемножение матриц в формуле (4) также вычислять в виде скалярных произведений, а последние выполнять с двойной точностью, то элементы матрицы X_{h+1} , как правило, будут найдены с высокой точностью. Следовательно, в некоторых случаях целесообразнее находить U^{-1} не прямым методом, а по формуле (4), используя преобразование UX_h . При этом очевидно, что абсолютная точность вычислений преобразует (4) в тождество. Многочисленные числовые эксперименты показывают, что в случае очень плохой обусловленности матрицы U формула (4) более эффективна, чем формула (3), а в случае ее умеренно плохой обусловленности более предпочтительна формула (3) [8]. Итак, если матрица очень плохо обусловлена и ее размерность невелика, то для обращения ее разумнее использовать формулу (4). Эта идея применительно к решению уравнения (1) приводит к методу Ньютона типа

$$x_{h+1} = x_h - A_h F(x_h), \quad (5)$$

$$A_h = A_{h-1} (F'(x_h) A_{h-1})^{-1}, \quad k=0, 1, \dots, \quad (6)$$

где $A_{-1} = I$ и $k=0, 1, \dots$. Однако громоздкость вычислений (только для вычисления A_k требуется произвести $O(3n^3)$ умножений) осложняет его применение. Одним из способов уменьшить объем вычислений в среднем на одну итерацию является следующий: использовать формулу (6) не на каждом шаге, а через шаг или через несколько шагов, по аналогии с [3-6], а время расчетов сокращать за счет распараллеливания вычислений.

Ниже будет рассматриваться, в частности, случай, когда у нас имеются два процессора. Легко видеть, что после вычисления матрицы $B_k = F'(x_k) A_{k-1}$ дальнейшие вычисления можно разбить на две параллельные ветви.

В первой путем решения уравнения $B_k y_k = F(x_k)$ или обращения матрицы B_k вычисляется вектор $y_k = B_k^{-1} F(x_k)$, а затем умножается матрица A_{k-1} на вектор y_k , что дает $A_k F(x_k) = A_{k-1} B_k^{-1} F(x_k)$ и, тем самым, $x_{k+1} = x_k - A_k F(x_k)$. После этого можно перейти к вычислению $F(x_{k+1})$ и $F'(x_{k+1})$. Отметим при этом, что наличие диагонально доминирующей матрицы B_k позволяет привлекать метод типа переменной метрики [10], а также итерационные методы (метод Якоби и др.), обеспечивающие высокую точность вычисления вектора y_k .

Во второй ветви вычисляется A_{k+1} в явном виде по формуле (6). Если вычисление $F(x)$ и $F'(x)$ не займет много времени, то время расчетов в первой ветви не превысит времени расчетов во второй, и применение параллельного метода позволит выиграть время, приблизительно равное тому, которое потребовалось бы затратить на вычисление $F(x_{k+1})$, $F'(x_{k+1})$ и умножение A_k на вектор $F(x_k)$ при реализации последовательного варианта этого метода.

З а м е ч а н и е 1. Здесь не учитывается возможность уравнивать время расчетов в обеих ветвях за счет обмена информацией между ветвями

и одновременного вычисления $F(x)$ и $F'(x)$, что также повышает эффективность параллельного метода [11].

Замечание 2. Так как в данном случае $A_{k-1}(F'(x_k)A_{k-1})^{-1} = (A_{k-1}F'(x_k))^{-1}A_{k-1}$, то, подставив в формулу $A_k = (A_{k-1}F'(x_k))^{-1}A_{k-1}$ вместо A_{k-1} матрицу $[F'(x_k)]^T$, получим из (5)–(6) аналог метода Гаусса—Ньютона.

Замечание 3. Если $F'(x)$ симметричная (положительно определенная), очень плохо обусловленная матрица, а в нашем распоряжении нет устойчивого алгоритма для ее обращения, то в качестве A_0 можно принять $[F'(x_0) + aI]^{-1}$, где a — некоторое малое положительное число. Прибавление члена aI обеспечивает полный ранг матрицы $F'(x_0) + aI$.

В случае плохой обусловленности матриц могут быть эффективными и такие параллельные варианты, как

$$x_{k+1} = x_k - A_k \sum_{i=0}^{p-1} (I - F'(x_k)A_k)^i F(x_k),$$

$$A_{k+1} = A_k \sum_{i=0}^{q-1} (I - F'(x_k)A_k)^i, \quad k=0, 1, \dots,$$

имеющие при $p, q \geq 2$ квадратичную скорость сходимости, хотя в принципе вместо последней формулы пригодна любая, обеспечивающая по крайней мере точность $\|I - F'(x_{k+1})A_{k+1}\| = O(b_k)$, $b_k = db_{k-1}$ ($d > 0, db_0 < 1$) (ср. с [12]), в частности, можем принять $A_{k+1} = [F'(x_k)]^{-1}$ (ср. с [3]).

2. При решении нелинейных уравнений итерационными методами общее время расчетов включает время на вычисление значений функции и ее производных, а также время на алгебраические процедуры (обращение матриц, вычисление собственных значений и т. д.). Для задач, где вычисление значений функции или ее производных связано с большими затратами времени, могут быть полезными методы, в которых используются производные (или их разностные аналоги) не выше первого порядка, а также методы с трудоемкими алгебраическими процедурами. Последниеобладают, как правило, высокими порядками скорости сходимости и могут потребовать меньше итераций, а следовательно, и меньше операций на расчеты значений функций или ее производных, чем методы низкого порядка [13].

В [14] для решения уравнения (2) описан метод

$$x_{k+1} = v_k - 2A_k F(v_k) + q^{-1}A_k [F(v_k + qA_k F(v_k)) - F(v_k)], \quad (7)$$

$$v_k = x_k - A_k F(x_k), \quad (8)$$

где A_k — некоторая аппроксимация для $[F'(x_k)]^{-1}$ и q — произвольный, не равный нулю параметр. Метод (7)–(8) имеет четвертый порядок скорости сходимости, если

$$\|I - F'(x_k)A_k\| = O(\|F(x_k)\|) = O(\|F(x_0)\|^{4k}). \quad (9)$$

Вообще говоря, существует такая закономерность, что чем выше порядок сходимости метода, тем строже требования к точности решения линейных проблем.

Рассмотрим некоторые варианты метода (7)–(8), способные обеспечить выполнимость условия (9).

Решение уравнения (2) методами типа (7)–(8) равносильно решению на каждом итерационном шаге линейных уравнений

$$[F'(x_k) + V_k]t_k = -F(x_k), \quad (10)$$

$$[\bar{F}'(x_k) + V_k] \zeta_k = \bar{F}(v_k), \quad (11)$$

$$[F'(x_k) + V_k] \omega_k = Q^{-1} [F(v_k + Q \zeta_k) - F(v_k)] - 2F(v_k), \quad (12)$$

где $t_k = v_k - x_k$ и $V_k = A_k^{-1} - F'(x_k)$.

Решение уравнений (10)–(12) методом Гаусса и обращение матрицы методом Гаусса требуют приблизительно одинакового числа арифметических операций, поэтому для решения (2) можно рекомендовать следующий последовательный вариант метода (7)–(8): некоторым прямым методом (методом Гаусса) вычислять $[F'(x_k)]^{-1}$, а полученную матрицу уточнять итерационным способом согласно формуле (3), т. е.

$$x_{k+1} = v_k - 2A_k^q F(v_k) + Q^{-1} A_k^q [F(v_k + Q A_k^q F(v_k)) - F(v_k)], \quad (13)$$

$$v_k = x_k - A_k^q F(x_k), \quad (14)$$

где

$$A_k^q u = A_k \sum_{i=0}^{q-1} (I - F'(x_k) A_k)^i u, \quad q \geq 2, k=0, 1, \dots, \quad (15)$$

$$A_k = [F'(x_k)]^{-1}. \quad (15a)$$

Подстановка в формулы (15) и (15a) $q \geq 4$ и $A_k = [F'(x_{k-1})]^{-1}$, $k=1, 2, \dots$, причем $A_1 = A_0 = [F'(x_0)]^{-1}$, приводит к параллельному варианту метода (7)–(8), т. е. к возможности одновременного вычисления (начиная с $k=1$) x_{k+1} и v_k по формулам (13)–(14) и $A_{k+1} = [F'(x_k)]^{-1}$ ($k=1, 2, \dots$) некоторым методом обращения матрицы.

Знание точной или приближенной инверсии якобиана полезно во многих случаях. Например, она позволяет вычислять число обусловленности или оценку для него, облегчает оценивать погрешность приближенного решения уравнения и т. д.

Рассмотрим еще некоторые варианты метода (7)–(8), которые могут быть полезными в случае плохой обусловленности $F'(x)$.

Несмотря на то, что формулы (3) и (4) не экономичны в смысле количества нужных арифметических операций, они иногда могут оправдывать себя в методах высокого порядка скорости сходимости, обеспечивая высокую точность вычислений.

Приведем пример распараллеливания вычислений методом (7)–(8) с использованием формулы (4):

1. После вычисления B_k^{-1} вычисляется по формулам (4), (13) и (14)

$$t_k = A_k F(x_k) = A_{k-1} B_k^{-1} F(x_k), \quad v_k = x_k - A_k F(x_k), \quad \zeta_k = A_k F(v_k),$$

$$F(v_k + Q F(v_k)), \quad \omega_k = Q^{-1} A_k [F(v_k + Q \zeta_k) - F(v_k)],$$

$$x_{k+1} = v_k - 2\zeta_k + \omega_k.$$

2. Находится по формуле (4) матрица $A_k = A_{k-1} B_k^{-1}$. Известно [12, 14], что формула

$$A_{k+1} = A_{k+1}^q = A_k \sum_{i=0}^{q-1} (I - F'(x_{k+1}) A_k)^i \quad (16)$$

обеспечивает четвертый порядок скорости сходимости метода (7)–(8) при $q \geq 4$. Легко видеть, что последовательно-параллельный вариант метода (7)–(8) с двухступенчатой аппроксимацией инверсии якобиана

$$x_{k+1} = v_k - 2A_k^2 F(v_k) + Q^{-1} A_k^2 [F(v_k + Q A_k^2 F(v_k)) - F(v_k)],$$

$$v_k = x_k - A_k^2 F(x_k),$$

$$A_k \equiv A_k^2 = 2\bar{A}_k - \bar{A}_k F'(x_k) \bar{A}_k, \quad A_k^2 u = 2\bar{A}_k u - \bar{A}_k F'(x_k) \bar{A}_k u, \quad v \in R^n,$$

$$\bar{A}_{k+1} = 2A_k - A_k F'(x_{k+1}) A_k$$

эквивалентен последовательному варианту метода (7) — (8) с использованием формулы (16) при $q = 4$. Так как во время вычисления \bar{A}_{k+1} распараллеливать расчеты методом (7) — (8) (как и (5) — (6)) не удастся, разумнее назвать его псевдопараллельным или частично-параллельным методом.

Отметим, что многоступенчатые варианты метода типа Чебышева изучены в [15].

Замечание 4. Сочетание формул (3) и (4) дает для нахождения левой или правой матрицы к $(m \times n)$ -матрице U соответствующие рекуррентные формулы:

$$Y_{k+1} = D_{k+1}^{-1} X_{k+1}, \quad Z_{k+1} = X_{k+1} B_{k+1}^{-1}, \quad (17)$$

где

$$X_{k+1} = X_k [I_{m \times m} + T_k + \dots + T_k^{q-1}] = [I_{n \times n} + E_k + \dots + E_k^{q-1}] X_k,$$

$$T_k = I_{m \times m} - U X_k, \quad E_k = I_{n \times n} - X_k U, \quad X_0 = U + U X_0 U U^+,$$

$$U X_{k+1} = B_{k+1} = (b_{ij}^{(k+1)})_{i,j=1, \dots, m}, \quad X_{k+1} U = D_{k+1} = (d_{ij}^{(k+1)})_{i,j=1, \dots, n}.$$

Однако реализация формул (17) весьма громоздка в вычислительном отношении, поэтому любопытно выяснить эффективность применения в методах высокого порядка скорости сходимости таких упрощенных формул, как

$$Y_{k+1} = \begin{pmatrix} d_{11}^{(k+1)} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & d_{nn}^{(k+1)} \end{pmatrix}^{-1} X_{k+1}, \quad Z_{k+1} = X_{k+1} \begin{pmatrix} b_{11}^{(k+1)} & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & b_{mm}^{(k+1)} \end{pmatrix}^{-1}.$$

ЛИТЕРАТУРА

1. Глушков В. М., Молчанов И. Н. Кибернетика (Киев), № 4, 82—88 (1981).
2. Фаддеева В. Н., Фаддеев Д. К. Препринты ЛОМИ Р-6-77, Р-6-81. Л., 1977, 1981.
3. Коган Т. И. В кн.: Научные труды Ташкентского гос. ун-та, вып. 320. Ташкент, «Фан», 1968, с. 46—53.
4. Ваарманн О. Изв. АН ЭССР, Физ. Матем., 20, № 4, 386—394 (1971).
5. Hald, O. H. Numer. Math., 23, № 5, 411—426 (1975).
6. Волокитин С. С. В кн.: Методы оптимизации и их приложения. Иркутск, СО АН СССР, 1979, с. 117—129.
7. Petryshyn, W. V. Proc. Amer. Math. Soc., 16, № 5, 893—902 (1965).
8. Sittion, G. A. IBM J. Res. Develop., 15, № 5, 413—417 (1971).
9. Воеводин В. В., Ким Г. Д. В кн.: Вычислительные методы и программирование, вып. XXVI. М., МГУ, 1977, с. 36—42.
10. Broeder, G. J. Comput. and Appl. Math., 6, № 4, 267—273 (1980).
11. Роозе А. Изв. АН ЭССР, Физ. Матем., 31, № 1, 32—37 (1982).
12. Ваарманн О. Изв. АН ЭССР, Физ. Матем., 27, № 3, 251—258 (1978).
13. Бард Я. Нелинейное оценивание параметров. М., «Статистика», 1979.
14. Ваарманн О., Полль В. Изв. АН ЭССР, Физ. Матем., 26, № 2, 123—127 (1977).
15. Райzman А. И. В кн.: II симпозиум по методам решения нелинейных уравнений и задач оптимизации. Хаапсалу, 4—7 июня 1981 г. Таллин, «Валгус», 1981, с. 101—103.

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
1 марта 1982

MONEDEST MITTELINEAARSETE VÖRRANDITE PARALLEELSETEST ITERATIIVSETEST LAHENDUSMEETODITEST

Matemaatiliste ülesannete lahendamise kiirendamiseks püütakse tänapäeval luua üha enam selliseid meetodeid ja algoritme, mis sobivad rakendada paralleelselt töötavatel arvutitel. Sel eesmärgil muudetakse vanu järjestikmeetodeid paralleelseiks või töötatakse välja uusi, paralleeltoimiga arvutitele vastavaid meetodeid. Seni on paremaid tulemusi andnud paralleelsete arvutusalgoritmide väljatöötamine lineaaralgebra ülesannete tarvis, seevastu mittelineaarsetele võrrandisüsteemidele sobivate osas pole nime-tamisyäärset edu olnud.

Käesolevas töös on esitatud mitu mittelineaarsete võrrandisüsteemide lahendamiseks kasutatavat 2. ja 4. järku paralleelset iteratsioonimeetodit, millest mõned on tuntud järjestikuste iteratsioonimeetodite modifikatsioonid. Kõrvaltulemusena on saadud ka mõned uued Newtoni meetodi variandid ja ühte 4. järku iteratsioonimeetodite klassi kuuluvate meetodite järjestikused variandid.

ON CERTAIN PARALLEL ITERATIVE METHODS FOR SOLVING NONLINEAR EQUATIONS

Increasing availability of multi-processor computer systems has prompted designing algorithms for parallel computation. The effectiveness of multi-processor computing systems in systems simulation will heavily depend on the efficiency of parallel algorithms in exploiting the multi-processing capability.

A considerable success has been reached in the development of parallel numerical methods for solving linear algebra problems [2], but so far relatively few results have been published on algorithms which have been constructed for parallel computation in the numerical solution of nonlinear equations.

In this paper several rapidly convergent parallel iterative methods are studied for solving $F(x)=0$, where $F(x)$ is a vector-function, mapping R^n into itself. As by-products, two new serial variants of certain iterative methods are obtained (see (5)—(6) and (13)—(15a), resp). A major drawback of the serial method (5)—(6) is that for inverting the Jacobian matrix a time-consuming procedure (6) is used, but this increase in complexity is accompanied by a decrease in susceptibility to rounding errors. In [8] it is shown that the formula (6) works very well even for inverting extremely ill-posed matrices. It is plain that an algorithm capable of solving complicated problems will usually not be efficient for solving easier problems.

Specifically, we consider the case where there are two processors, and we assume that communication between them is not too expensive as opposed to the computation itself. For solving $F(x)=0$ formulas have been derived where computations can be performed in parallel on two processors. Since the methods under discussion involve a linearization of the system of equations about a point then, naturally, for solving corresponding linear problems at every single iteration step, parallel methods developed in linear algebra, can be employed. But here we consider such parallel methods which take advantage of special structure of iteration formulas for solving nonlinear equations.

Sometimes merely an implementation of methods having a large order of convergence permits to reduce run times in seconds on the computer. Here, to this end, additionally, parallelism is used. Finally, iterative methods of the 4th order suited for parallel computation of a solution of systems of nonlinear equations with ill-posed Jacobian matrices are proposed. In particular, a class of methods of the 4th order with multistep successive approximation of inverse is considered.