

УДК 681.3

ИТЕРАЦИОННЫЙ МЕТОД РЕШЕНИЯ ГРАНИЧНОЙ ЗАДАЧИ ДЛЯ СИСТЕМЫ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПЕРВОГО ПОРЯДКА С ПАРАМЕТРОМ ДЛЯ КЛАСТЕРНЫХ СИСТЕМ

Т. Д. Давиташвили¹, Г. В. Меладзе¹, В. Г. Саакян², П. А. Церетели¹

В данной работе рассматриваются проблемы построения, исследования и реализации синхронных параллельных итерационных методов решения нелинейных уравнений. Рассматривается также проблема нахождения автомодельного решения математической модели неустановившегося движения газа, возникшего под действием плоского поршня при наличии объемных источников (стоков) массы, импульса и энергии. Изучаются проблемы построения и реализации итерационного метода решения полученной граничной задачи с учетом особенностей параллельных вычислений. Работа выполнена при поддержке Международного научно-технического центра (МНТЦ) в рамках проекта “Разработка высокопроизводительного вычислительного кластера и баз данных в Армении” (А 823).

Ключевые слова: итерационные методы, краевая задача, многопроцессорные вычислительные системы, параллельное программирование, кластерные системы.

1. Введение. Для численного моделирования сложных прикладных задач в настоящее время перспективным направлением является использование вычислительных систем с параллельной обработкой данных. Однако новые исследования в области параллельных вычислений пока выглядят довольно скромно по сравнению с результатами в области последовательных вычислений.

Быстрое развитие многопроцессорной вычислительной техники в последнее время характеризуется тенденцией увеличения доли кластерных систем. Вычислительный кластер — это совокупность компьютеров, объединенных с помощью сети для решения одной задачи. В качестве узлов используются доступные на рынке компьютеры (они могут быть как однопроцессорными, так и многопроцессорными, например SMP-серверы). При этом каждый узел работает под управлением своей копии операционной системы.

На широкое распространение кластерных систем указывает и тот факт, что в 24-й редакции списка top500 (ноябрь, 2004 г.), который содержит названия 500 наиболее мощных суперкомпьютеров в мире, число кластерных установок составляет почти 60 %. Это говорит еще и о том, что кластеры по своим техническим и вычислительным характеристикам ничем не уступают современным мощнейшим суперкомпьютерам. Такое широкое распространение кластерных вычислительных систем обусловлено, в первую очередь, сравнительной простотой их построения, гибкостью и масштабируемостью, а также высоким показателем соотношения “производительность/цена” — стоимость кластерных систем значительно ниже стоимости традиционных суперкомпьютеров такой же производительности. Причина последнего состоит в том, что для их построения используются массовые процессоры, стандартные сетевые технологии и свободно распространяемое программное обеспечение. С другой стороны, быстрое развитие кластерных технологий сделало параллельные вычисления доступным для многих.

Основная цель использования параллельных вычислительных систем, в том числе и кластеров, — уменьшить время выполнения программы за счет максимального использования ресурсов вычислительной системы. В этой связи немаловажным фактором при создании параллельных программ является выбор технологии параллельного программирования, которая должна обеспечивать всестороннее использование потенциала вычислительной системы и тем самым создание эффективных параллельных программ. На сегодняшний день существует более 100 таких технологий (например, ATLAS, Linda, Cray MPP Fortran, HPF, MPI, PVM, OpenMP и др.; подробный обзор см. в книге [2]). Одной из наиболее распространенных технологий является MPI, которая позволяет взаимодействовать параллельным процессам с помощью передачи сообщений друг другу. Для создания параллельных программ, как правило, используется подход SPMD, в рамках которого для всех процессов используется один и тот же код.

¹ Тбилисский государственный университет им. Ив. Джавахишвили, факультет прикладной математики и компьютерных наук, пр. Чавчавадзе, 1, 0128, г. Тбилиси, Грузия; e-mail: meladze.cyber@viamnet.edu.ge, paata@vvv.srcc.msu.su

² Институт проблем информатики и автоматизации Национальной академии наук Армении, ул. П. Севака, 1, 375014, г. Ереван, Армения; e-mail: svlad@sci.am

В 2004 г. в рамках проекта ARMCLUSTER при поддержке Международного научно-технического центра (МНТЦ) в Институте проблем информатики и автоматизации Национальной академии наук Армении (г. Ереван) был построен 128-процессорный кластер с пиковой производительностью 783.36 Gflops. Этот кластер состоит из 64 узлов Intel Dual Xeon 3.06 GHz, каждый узел содержит 1 GB оперативной памяти. В качестве основной коммуникационной сети используется сеть Myrinet. Рассматриваемая в статье задача была решена для данной кластерной установки.

В последнее время много внимания уделяется построению и изучению параллельных алгоритмов решения нелинейных уравнений и систем. В [3] предложены синхронные параллельные итерационные методы, в которых использованы обратные интерполяционные формулы Лагранжа. Доказывается сходимость этих методов, если начальные приближения выбраны достаточно близко к искомому единственному нулю функции, а также рассматривается вопрос об оптимальном выборе числа процессоров. В [4] приведен обзор итерационных методов решения нелинейных уравнений. В [5] рассматриваются методы решения одного нелинейного уравнения, которые основаны на разложении функции в ряд Тейлора и аппроксимации многочленами. В [6, 7] обсуждаются некоторые параллельные методы решения нелинейных уравнений и содержатся ссылки на другие работы. Интересные результаты, связанные с исследованием скорости сходимости итерационных методов (в том числе и параллельных), получены в [8].

На основе сравнения этих методов с оптимальными последовательными методами можно заключить, что ускорение с ростом числа процессоров увеличивается лишь логарифмически.

2. Постановка задачи. Рассмотрим задачу нахождения такого значения λ^* параметра λ и такой вектор-функции $X^*(t, \lambda) = (x^1(t, \lambda), \dots, x^N(t, \lambda))$, которые удовлетворяют системе уравнений

$$\frac{dX}{dt} = F(t, X(t, \lambda), \lambda), \quad t \in (0, 1), \quad (1)$$

и условиям

$$X(0, \lambda) = \Psi(\lambda), \quad x^1(1, \lambda) = 0, \quad (2)$$

где $\lambda \in (\Lambda_0, \Lambda_1)$, а $F(t, X, \lambda) = \{f^1(t, X(t, \lambda), \lambda), \dots, f^N(t, X(t, \lambda), \lambda)\}$ и $\Psi(\lambda) = \{\psi^1(\lambda), \dots, \psi^N(\lambda)\}$ — вектор-функции, такие, что $f^i : [0, 1] \times R^N \times (\Lambda_0, \Lambda_1) \rightarrow R^1$ и $\psi^i : (\Lambda_0, \Lambda_1) \rightarrow R^1$, $i = \overline{1, N}$. Условия, которым удовлетворяют вектор-функции F и Ψ , будут сформулированы ниже.

Наряду с задачей (1), (2) рассмотрим эквивалентную ей задачу Коши

$$\frac{dX}{dt} = F(t, X(t, \lambda), \lambda), \quad X(0, \lambda) = \Psi(\lambda) \quad (3)$$

с уравнением относительно λ :

$$x^1(1, \lambda) = \psi^1(\lambda) + \int_0^1 f^1(\tau, X(\tau, \lambda), \lambda) d\tau = 0. \quad (4)$$

Для решения поставленной задачи используется синхронный итерационный метод решения нелинейных уравнений для параллельных вычислительных систем.

3. Описание итерационного метода. Сначала сделаем одно замечание.

Пусть $\{\varepsilon_i\}_{i=0}^\infty$ — сходящая к нулю последовательность, в которой $\varepsilon_i \neq 0$ кроме конечного числа индексов i . Будем говорить, что последовательность $\{\varepsilon_i\}_{i=0}^\infty$ сходится со скоростью сходимости λ , если имеет место оценка

$$|\varepsilon_i| \leq \gamma \theta^{\lambda^i} \quad \text{при } i > i_0, \quad (5)$$

где $\lambda > 1$ — вещественное число, а $\gamma > 0$ и $0 < \theta < 1$ — постоянные.

Пусть $f(x)$ — функция одной вещественной переменной. Будем предполагать, что выполнено условие (A1): можно найти такой интервал $I = (a, b)$, в котором уравнение $f(x) = 0$ имеет единственный корень $\bar{x}$, и $f(x) \in C^k(I)$. Потребуем также, чтобы $f'(x) \neq 0$, когда $x \in I$. Поскольку $f(\bar{x}) = 0$, то найдется такая окрестность I^* точки 0, где существует функция $g = f^{-1}$ и $g \in C^k(I^*)$.

Рассмотрим итерационную последовательность N -мерных векторов

$$X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_N^{(i)}), \quad i = 0, 1, 2, \dots, \quad (6)$$

каждая компонента которых суть приближенное значение корня $\bar{x}$. Далее мы покажем, что при определенных условиях каждая компонента вектора $X^{(i)}$ сходится к $\bar{x}$, когда $i \rightarrow \infty$.

Пусть заданы $\mathbf{X}^{(0)} = (x_1^{(0)}, \dots, x_{N-1}^{(0)}, x_N^{(0)})$ — N различных начальных приближений к $\bar{x}$ из I , а также значения функции и ее производных в этих точках:

$$\begin{aligned} f(x_1^{(0)}) = y_1^{(0)}, \dots, f(x_N^{(0)}) = y_N^{(0)}, \\ f'(x_1^{(0)}), \dots, f'(x_N^{(0)}), \\ \dots \dots \dots \\ f^{(\alpha_1-1)}(x_1^{(0)}), \dots, f^{(\alpha_N-1)}(x_N^{(0)}), \quad (\alpha_j \leq k+1, \quad j = \overline{1, N}). \end{aligned} \quad (7)$$

Если известны значения производных функции $y = f(x)$, то нахождение значений производных обратной функции не представляет больших трудностей (см. [9]). Так, если $x = g(y)$, то $x' = g' = \frac{1}{y'}$, $x'' = g'' = \frac{-y''}{(y')^3}$. В общем случае

$$\frac{d^k x}{dy^k} = g^{(k)}(y) = \frac{\chi_k}{(y')^{2k-1}},$$

где χ_k — многочлен относительно $y', y'', \dots, y^{(k)}$: $\chi_1 = 1$, $\chi_2 = -y''$ и $\chi_{n+1} = \frac{d}{dx}(\chi_n)y' - (2n-1)\chi_n y''$.

Очевидно, что $\bar{x} \equiv g(0)$. Для нахождения приближенных значений $\bar{x}$ функцию $g(y)$ заменим интерполяционным полиномом Эрмита на каком-нибудь подмножестве множества $\{y_j^{(i)}\}_{j=1}^N$ и вычислим значения этого полинома в точке 0.

Пусть M ($2 \leq M \leq N-1$) — число узлов, используемых в интерполяционной формуле Эрмита. Для j -компоненты вектора $\mathbf{X}^{(i)}$ построим интерполяционный полином Эрмита $H_j^{(i)}(y)$. Узлы для интерполяционных полиномов будем выбирать следующим образом: рассмотрим множество индексов $A = \{1, 2, \dots, N\}$ и выберем N различных подмножеств A_j этого множества, в которых количество элементов равно M . Пусть также $j, j+1 \in A_j$. Для удобства реализации потребуем, чтобы все полиномы Эрмита имели одинаковый порядок $m-1$, где $2 \leq m \leq k$.

В качестве узлов для полинома $H_j^{(i)}(y)$ возьмем точки $\{y_s^{(i)}\}$, $s \in A_j$. Очевидно, в узлах интерполяции при $\sum_{s \in A_j} \alpha_{js} = m$ и $\alpha_{js} \leq \alpha_s$, $j = \overline{1, N}$, должны выполняться следующие условия:

$$\begin{aligned} H_j^{(i)}(y_s^{(i)}) &= x_s^{(i)}, \\ [H_j^{(i)}(y_s^{(i)})]' &= [x_s^{(i)}]', \\ \dots \dots \dots \\ [H_j^{(i)}(y_s^{(i)})]^{(\alpha_{js}-1)} &= [x_s^{(i)}]^{(\alpha_{js}-1)}, \quad s \in A_j. \end{aligned} \quad (8)$$

Рассмотрим итерационный алгоритм вычисления j -компоненты вектора $\mathbf{X}^{(i+1)}$:

$$x_j^{(i+1)} = H_j^{(i)}(0), \quad j = \overline{1, N}. \quad (9)$$

В результате получим последовательность векторов $\{\mathbf{X}^{(i)}\}_{i=0}^\infty$. Отметим, что этот алгоритм рассчитан для многопроцессорной системы типа SIMD и при $\alpha_1 = \alpha_2 = \dots = \alpha_N = 1$ совпадает с итерационным алгоритмом из [3], а его численная реализация требует N параллельных процессоров.

В доказательстве сходимости итерационного алгоритма (6), (9) используется его матричная интерпретация и оценивается скорость его сходимости.

Пусть для построения полинома $H_j^{(i)}(y)$ используются значения функции и производных: $\{y_s^{(i)}\}$, $\{g(y_s^{(i)})\}, \dots, \{g^{(\alpha_{js}-1)}(y_s^{(i)})\}$, $s \in A_j$. Рассмотрим вектор Λ_j , где $t_{js} = 0$, если $s \notin A_j$, и $t_{js} = \alpha_{js}$, если $s \in A_j$:

$$\Lambda_j = (t_{j1}, t_{j2}, \dots, t_{jN}). \quad (10)$$

Таким образом, каждому полиному $H_j^{(i)}(y)$ соответствует вектор Λ_j , $j = \overline{1, N}$, а итерационному методу в целом — матрица Λ со строками Λ_j .

Пусть $\mathbf{X}^{(0)} = (x_1^{(0)}, \dots, x_{N-1}^{(0)}, x_N^{(0)})$ — начальные приближения итерационного процесса (6), (9). Будем предполагать, что выполнено условие (A2): функция $f^{(m)}(x)$ ограничена на I , а $g^{(m)}(y) \neq 0$ и $g'(y) \neq 0$ на промежутке I^* .

Введем следующие обозначения:

$$q = \max_{1 \leq j \leq N} |\bar{x} - x_j^{(0)}| = \|\varepsilon^{(0)}\|, \quad \varepsilon^{(0)} = (\varepsilon_1^{(0)}, \dots, \varepsilon_N^{(0)}), \quad \varepsilon_s^{(i)} = |\bar{x} - x_s^{(i)}|, \quad p = \frac{\|g^{(m)}(y)\|}{m!} \|f'(x)\|^m. \quad (11)$$

Теорема 1. Пусть для функции $f(x)$ выполнены условия (A1) и (A2) ($k = m$), а начальные данные выбраны из окрестности $I^0 \subseteq I$, для точек которой имеет место неравенство $pq^{m-1} < 1$. Тогда итерационный процесс (6), (9) сходится со скоростью сходимости, равной спектральному радиусу m матрицы Λ , а погрешность, начиная с некоторой итерации i_0 , оценивается следующим образом: $\varepsilon_j^{(i)} \leq \left((p^{1/(m-1)}q)^\beta\right)^{m^i}$, где $\beta = \text{const}$, $j = \overline{1, N}$, $i \geq i_0$.

Доказательство теоремы приведено в [10].

Как следует из теоремы 1, сходимость итерационного процесса (6), (9) зависит от выбора начальных приближений. Обычно его область сходимости I^0 является частью области I , в которой уравнение $f(x) = 0$ имеет единственный корень, $f(x) \in C^m(I)$ и существует обратная функция. Говоря другими словами, в условиях этой теоремы имеет место локальная сходимость. Однако, используя результаты из [11], можно построить такую итерационную процедуру, которая при некоторых условиях позволяет за конечное число шагов отыскать требуемые начальные данные из I^0 . После вычисления этих начальных данных реализуется итерационный процесс (6), (9). Областью его сходимости является весь промежуток I . Соответствующая теорема сходимости формулируется в [10].

В качестве примера использования предложенного итерационного метода рассмотрим следующую нелинейную краевую задачу [12]:

$$y'' = y^2 - 1, \quad x \in (0, 1), \quad y(0) = 0, \quad y(1) = 1.$$

Это уравнение перепишем в виде $x = \int_0^y \left(2\left(c + \frac{y^3}{3} - y\right)\right)^{-1/2} dy$, где c — постоянная и $y'(0) = \sqrt{2c}$. Отсюда, учитывая граничные условия, получим нелинейное уравнение относительно c :

$$\phi(c) = 1 - \int_0^1 \left(2\left(c + \frac{y^3}{3} - y\right)\right)^{-1/2} dy = 0.$$

Очевидно, что областью определения функции $\phi(c)$ является интервал $(2/3, +\infty)$. Для решения уравнения $\phi(c) = 0$ воспользуемся тремя итерационными методами:

- 1) метод Ньютона;
- 2) метод (6), (9), где $N = 6$, $M = 2$, $m = 4$, $\alpha_{js} = 2$, $s \in A_j$, $j = \overline{1, 6}$, $A_j = \{j, j+1\}$, $j = \overline{1, 5}$, $A_6 = \{6, 1\}$; в этом случае итерационная формула будет иметь вид

$$x_j^{(i+1)} = \frac{1}{(y_{j+1} - y_j)^2} \sum_{k=0}^1 y_{j+k}^2 \left(x_{j+1-k} + \frac{2x_{j+1-k} y_{j+1-k}}{(-1)^k (y_{j+1} - y_j)} - \frac{y_{j+1-k}}{f'(x_{j+1-k})} \right), \quad (12)$$

где индекс итераций “ (i) ” у всех членов в правой части равенства (12) опущен, а суммы в нижних индексах взяты по модулю 6 (число процессоров);

- 3) модифицированный метод с итерационным нахождением начальных приближений.

Во всех трех случаях для вычисления интеграла была использована квадратурная формула Симпсона. Точность расчетов: $\varepsilon = 10^{-6}$. Для неизвестного c получено приближенное значение $c^* = 0.97106$.

Анализ результатов показывает, что метод Ньютона сходится на интервале $[0.67, 1.42]$, длина которого равна 0.75, а среднее число итераций равно 6. Метод (6), (9) сходится на интервале $[0.67, 1.66]$, длина которого равна 0.99, а число итерации равно 3. Областью сходимости модифицированного метода с итерационным нахождением начальных приближений является интервал $[0.67, K]$, где K — любое конечное число из интервала $(2/3, +\infty)$.

4. Алгоритм решения задачи (3), (4). Для решения этой задачи рассмотрим итерационный метод, который использует методику, предложенную в разделе 3. Для наглядности мы будем применять интерполяционные формулы, в которых использованы только два узла. Поэтому для реализации предложенного алгоритма на многопроцессорных вычислительных системах достаточно трех процессоров. Отметим, что случай других интерполяционных формул и, тем самым, использование большего числа процессоров можно рассмотреть аналогично на основе результатов из раздела 3.

Будем считать, что задача Коши (3) имеет решение $\bar{X}(t, \lambda) = (\bar{x}^1(t, \lambda), \dots, \bar{x}^N(t, \lambda))$ для каждого λ и $\bar{x}^i(t, \lambda) \in C^1[0, 1]$ при $\lambda \in (\Lambda_0, \Lambda_1)$. Допустим, что выполняется следующее условие:

$$\frac{d}{dx} \bar{x}^1(1, \lambda) \neq 0, \quad (13)$$

где $\bar{x}^1(1, \lambda) : (\Lambda_0, \Lambda_1) \rightarrow (\bar{\Lambda}_0, \bar{\Lambda}_1)$. Тогда существует обратная функция $\varphi \equiv [\bar{x}^1(1, \lambda)]^{-1}$. Очевидно, что функция φ будет иметь такую же гладкость, что и функция $\bar{x}^1(1, \lambda)$.

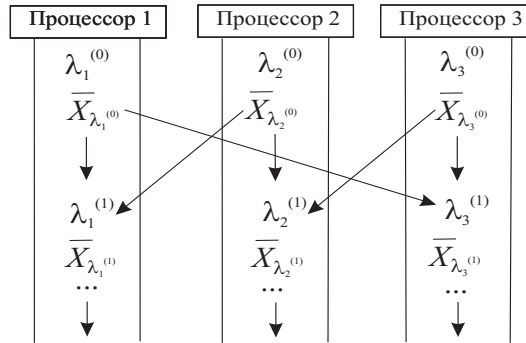
Выбирая начальные приближения параметра λ : $\lambda_1^{(0)}, \lambda_2^{(0)}, \lambda_3^{(0)}$, рассмотрим последовательность следующих задач:

$$\frac{dX(t, \lambda_j^{(i)})}{dt} = F(t, X(t, \lambda_j^{(i)}), \lambda_j^{(i)}), \quad X(0, \lambda_j^{(i)}) = \Psi(\lambda_j^{(i)}), \quad (14)$$

$$\lambda_j^{(i+1)} = H_j^{(i)}(0), \quad (15)$$

где $i = 0, 1, 2, \dots$ — номер итерации, $j = 1, 2, 3$ — номер процессора и $H_j^{(i)}(z)$ — обратный интерполяционный полином Эрмита, построенный для узлов $z_j^{(i)}, z_{j+1}^{(i)}$, таких, что $z_j^{(i)} = \bar{x}^1(1, \lambda_j^{(i)})$, $z_{j+1}^{(i)} = \bar{x}^1(1, \lambda_{j+1}^{(i)})$, $j = \overline{1, 3}$, $z_4^{(i)} = z_1^{(i)}$.

Графически описанный процесс можно изобразить в следующем виде:



Значение производных первого порядка по параметру λ функции $\bar{x}^1(1, \lambda)$ можно вычислить из следующей задачи Коши:

$$\frac{dU(t, \lambda_j^{(i)})}{dt} = \sum_{k=1}^N \frac{\partial F(t, X(t, \lambda_j^{(i)}), \lambda_j^{(i)})}{\partial x^k(t, \lambda_j^{(i)})} u^k(t, \lambda_j^{(i)}) + \frac{\partial F(t, X(t, \lambda_j^{(i)}), \lambda_j^{(i)})}{\partial \lambda}, \quad (16)$$

$$U(0, \lambda_j^{(i)}) = \frac{d}{d\lambda} \Psi(\lambda_j^{(i)}), \quad (17)$$

где через $U(t, \lambda) = (u^1(t, \lambda), \dots, u^N(t, \lambda))$ обозначена $\frac{d\bar{X}(t, \lambda)}{d\lambda}$.

Пусть задача Коши (16), (17) для каждого $\lambda \in (\Lambda_0, \Lambda_1)$ имеет решение $U(t, \lambda)$, $t \in [0, 1]$. Для построения обратного полинома Эрмита в (15) используем значения обратной функции $\varphi(z)$ и ее производных в заданных точках. Тогда соответствующая итерационная формула будет иметь вид

$$\lambda_j^{(i+1)} = \frac{1}{(z_{j+1} - z_j)^2} \sum_{k=0}^1 z_{j+k}^2 \left(\lambda_{j+1-k} + \frac{2\lambda_{j+1-k} z_{j+1-k}}{(-1)^k (z_{j+1} - z_j)} - z_{j+1-k} \varphi'(z_{j+1-k}) \right). \quad (15^*)$$

Для сходимости итерационного процесса (14)–(15*), (16), (17) имеет место следующая

Теорема 2. Пусть $F(t, X, \lambda) \in C^4(\Lambda_0, \Lambda_1)$ для любых фиксированных t и X , $\Psi(\lambda) \in C^4(\Lambda_0, \Lambda_1)$ и существуют такие постоянные L_1 и L_2 , что $\|F(t, X, \lambda_1) - F(t, Y, \lambda_2)\| \leq L_1 \|X - Y\| + L_2 |\lambda_1 - \lambda_2|$.

Пусть, кроме того, выполняется условие (13). Тогда если $p\eta^3 < 1$, где $\eta = \max_{1 \leq j \leq 3} |\lambda_j^{(0)} - \lambda^*|$ и $p =$

$\frac{\|\varphi^{(4)}(\xi)\| \|u^1(1, \lambda)\|^4}{24}$, то $\lambda_j^{(i)} \rightarrow \lambda^*$ при $i \rightarrow \infty$ и $j = \overline{1, 3}$ со скоростью сходимости, равной 4. Кроме того,

$$|\lambda_j^{(i)} - \lambda^*| \leq p^{-1} (\eta^3 p)^{4^i}, \quad \|\bar{X}(t, \lambda_j^{(i)}) - \bar{X}(t, \lambda^*)\| = O[(p\eta^3)^{4^i}], \quad j = 1, 2, 3.$$

Доказательство теоремы 2 приведено в [10].

5. Некоторые автомодельные задачи газовой динамики

Задача 1. Применим описанный метод для решения задачи движения поршня с ускорением и замедлением в среде с объемными стоками массы. Рассмотрим одномерное неустановившееся движение газа, возникшее под действием плоского поршня при наличии объемных источников (стоков) массы, импульса и энергии в газе. В начальный момент времени газ покоится и имеет распределенную по пространству плотность. Соответствующая математическая модель записывается в следующем виде [13]:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + v \frac{\partial \rho}{\partial r} + \rho \frac{\partial v}{\partial r} &= -AT^{n_1} \rho^{n_1+n_2}, \\ \rho \left(\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial r} \right) + RT \frac{\partial \rho}{\partial r} + R\rho \frac{\partial T}{\partial r} &= 0, \\ \frac{R}{\gamma-1} \left(\frac{\partial T}{\partial t} + v \frac{\partial T}{\partial r} \right) + RT \frac{\partial v}{\partial r} &= AT^{n_1+1} \rho^{n_1+n_2-1}, \end{aligned} \quad (18)$$

$$v(r, 0) = 0, \quad T(r, 0) = 0, \quad \rho(r, 0) = \rho_0 r^l, \quad (19)$$

$$v(r^*(t), t) = v_0 t^{n_0}, \quad r^*(t) = v_0 \frac{t^{n_0+1}}{n_0+1}, \quad (20)$$

где ρ — плотность, v — скорость, T — температура, $AT^{n_1} \rho^{n_1+n_2}$ — удельная мощность источников ($A > 0$) или стоков ($A < 0$) массы, R — газовая постоянная, γ — отношение удельных теплоемкостей, A , ρ_0 , v_0 — размерные постоянные, $r^*(t)$ — координата поршня, n_0 , n_1 , n_2 , l — безразмерные постоянные. Будем предполагать, что газ является совершенным, т.е. подчиняется законам $p = R\rho T$, $\varepsilon = \frac{RT}{\gamma-1}$, где p — давление и ε — удельная внутренняя энергия. Для определенности будем считать, что поршень вдвигается в газ ($v_0 > 0$).

Используем предложенный в разделе 4 итерационный метод для построения автомодельных решений системы уравнений (18)–(20). Анализ размерностей показывает, что решение этой системы будет автомодельным, если выполняются следующие условия:

$$2n_0n_1 + (n_0+1)l(n_1+n_2-1) + 1 = 0, \quad n_0+1 > 0, \quad n_0 > -\frac{l+1}{l+3}.$$

В таком случае существует единственная безразмерная комбинация неизвестных переменных r и t вида $s = \frac{r}{v_0 t^{n_0+1}}$, а искомые функции представимы в виде

$$\rho = \delta(s) \rho_0 v_0^l t^{(n_0+1)l}, \quad v = \alpha(s) v_0 t^{n_0}, \quad T = \varphi(s) R^{-1} v_0^2 t^{2n_0}, \quad (21)$$

где $\delta(s)$, $\alpha(s)$ и $\varphi(s)$ — безразмерные функции, соответственно, плотности, скорости и температуры, удовлетворяющие следующей системе обыкновенных дифференциальных уравнений:

$$\begin{aligned} [(n_0+1)l\delta - (n_0+1)s - \alpha]\dot{\delta} + \delta\dot{\alpha} &= -\sigma_0 \delta^{n_1+n_2} f^{n_1}, \\ \delta[n_0\alpha - ((n_0+1)s - \alpha)\dot{\alpha}] + \delta\dot{\varphi} + \varphi\dot{\delta} &= 0, \\ \frac{1}{\gamma-1} [2n_0\varphi - ((n_0+1)s - \alpha)\dot{\varphi}] + \varphi\dot{\alpha} &= -\sigma_0 \delta^{n_1+n_2-1} \varphi^{n_1}, \end{aligned} \quad (22)$$

где безразмерная постоянная σ_0 имеет вид $\sigma_0 = A\rho_0^{n_1+n_2-1} v_0^{l(n_1+n_2-1)+2n_1}$, а точкой обозначена производная по s .

Условие (20), заданное на поршне при $r = r^*(t)$, в безразмерной форме запишется в виде

$$\alpha(s_0) = 1, \quad \text{где} \quad s_0 = \frac{r^*}{v_0 t^{n_0+1}} = \frac{1}{n_0+1}. \quad (23)$$

Начальные условия (19) при переходе к безразмерным переменным формально превращаются в условия, заданные при $s = \infty$. Однако вследствие того, что газокинематические возмущения в среде распространяются с конечной скоростью, эти условия должны быть заданы в некоторой конечной точке $s = s_1$:

$$\alpha(s_1) = 0, \quad \varphi(s_1) = 0, \quad \delta(s_1) = s_1^l. \quad (24)$$

Если $s_1 = \frac{r_1}{v_0 t^{n_0+1}}$, где $r_1 > r^*$ — координата, характеризующая положение фронта ударной волны, движущейся впереди поршня, то при $r = r_1$ должны выполняться условия, связывающие параметры газа перед фронтом ударной волны и за ним. Так как “исчезновение” массы из пространства, заполненного газом, носит объемный, а не поверхностный характер, то законы сохранения на фронте ударной волны имеют вид обычных условий Гюгонио:

$$\rho_l(\dot{r}_1 - v_l) = \rho_r \dot{r}_1, \quad \rho_l(\dot{r}_1 - v_l) + p_l = \rho_r \dot{r}_1^2, \quad \frac{1}{2}(\dot{r}_1 - v_l)^2 + \frac{\gamma}{\gamma-1} \frac{p_l}{\rho_l} = \frac{1}{2} \dot{r}_1^2, \quad (25)$$

где индексом “ l ” обозначены параметры газа за фронтом, индексом “ r ” — перед фронтом ударной волны, $\dot{r}_1 = \frac{dr_1}{dt}$ — абсолютная скорость фронта ударной волны и $\rho_r = \rho_0 r_1^l$. Учитывая формулы (21) и (25) и то, что $s_1 = \frac{r_1}{v_0 t^{n_0+1}}$, получаем

$$\delta_l = (\gamma+1) \frac{s_1^l}{\gamma-1}, \quad \alpha_l = 2(n_0+1) \frac{s_1}{\gamma+1}, \quad \varphi_l = 2(\gamma-1) \left[\frac{(n_0+1)s_1}{\gamma+1} \right]^2, \quad s_1 > s_0. \quad (26)$$

Таким образом, когда фронт ударной волны находится в точке s_1 , за фронтом ударной волны выполняются условия (24), а перед фронтом — условия (26). Требуется найти решения $\alpha(s)$, $\delta(s)$, $\varphi(s)$ системы уравнений (22) между поршнем и фронтом ударной волны, удовлетворяющие условиям (23) и (26). Для дальнейшего удобно выполнить замену переменных: $\tau = (s - s_1)(s_0 - s_1)^{-1}$, $u = (n_0 + 1)s - \alpha$. Тогда задачу (22), (23), (26) после элементарных преобразований можно записать в следующем виде:

$$\dot{u} = -(s_0 - s_1)(u^2 - \gamma\varphi)^{-1} \left\{ [2n_0 + (n_0 + 1)(l + \gamma)]\varphi + n_0(n_0 + 1)(s_1 + \tau(s_0 - s_1))u - (2n_0 + 1)u^2 + \gamma\sigma_0\varphi^{n_1+1}\delta^{n_1+n_2-1} \right\}, \quad (27)$$

$$\dot{\delta} = \delta(s_0 - s_1)u^{-1}[(n_0 + 1)(l + 1) + \sigma_0\varphi^{n_1}\delta^{n_1+n_2-1} - \dot{u}(s_0 - s_1)^{-1}],$$

$$\dot{\varphi} = \varphi(s_0 - s_1)u^{-1}[2n_0 + (n_0 + 1)(\gamma - 1) + (\gamma - 1)\sigma_0\varphi^{n_1}\delta^{n_1+n_2-1} - (\gamma - 1)\dot{u}(s_0 - s_1)^{-1}],$$

$$u(0) = (\gamma - 1)(\gamma + 1)^{-1}(n_0 + 1)s_1, \quad \delta(0) = (\gamma + 1)(\gamma - 1)^{-1}s_1^l, \quad \varphi(0) = 2(\gamma - 1)(\gamma + 1)^{-2}(n_0 + 1)^2s_1^2, \quad (28)$$

$$u(1) = 0. \quad (29)$$

Таким образом, мы получили зависящую от параметра $\lambda = s_1$ систему обыкновенных дифференциальных уравнений.

Задача 2. Рассмотрим одномерное течение идеального газа, возникшего под действием плоского поршня. При этом имеются объемные источники массы, а диссипативные процессы и источники тепла отсутствуют. В начальный момент газ покоится и его плотность равномерно распределена в пространстве.

Соответствующая математическая модель запишется в следующем виде:

$$\frac{\partial \rho}{\partial t} + v \frac{\partial \rho}{\partial r} + \rho \frac{\partial v}{\partial r} = A\rho^n, \quad (30)$$

$$\rho \frac{\partial v}{\partial t} + \rho v \frac{\partial v}{\partial r} + B \frac{\partial \rho}{\partial r} = 0,$$

$$v(r, 0) = 0, \quad \rho(r, 0) = \rho_0 r^l, \quad (31)$$

$$v(r^*, t) = v_0 t^{n_0}, \quad (32)$$

где t — время, r — переменная Эйлера, ρ — плотность, v — скорость, $A\rho^n$ — удельная мощность источников ($A > 0$) или стоков ($A < 0$) массы, r^* — координата поршня, A , B , ρ_0 , v_0 — размерные постоянные и n , l , n_0 — безразмерные постоянные.

Для определенности будем считать, что поршень вдвигается в газ, т.е. $v_0 > 0$. Если $l \neq 0$, $n \neq 1$, $n_0 = 0$, $l(1-n)-1 = 0$, то решение задачи (30)–(32) будет автомодельным. В этом случае существует единственная безразмерная комбинация независимых переменных r и t вида $s = \frac{r}{v_0 t}$, а искомые функции представимы

в виде $\rho(r, t) = \delta(s)\rho_0 v_0^l t^l$, $v(r, t) = \alpha(s)v_0$, где безразмерные функции $\delta(s)$, $\alpha(s)$ соответствуют функциям плотности и скорости и удовлетворяют следующей системе обыкновенных дифференциальных уравнений:

$$(l + \alpha')\delta + (\alpha - s)\delta' = A_0\delta^n, \quad (33)$$

$$(\alpha - s)\delta\alpha' + B_0\delta' = 0,$$

где штрихом обозначена производная по s , а безразмерные переменные A_0, B_0 имеют вид $A_0 = A\rho_0^{n-1}v_0^{-1}$, $B_0 = Bv_0^{-2}$.

Пусть s_1 — искомый параметр. Условия (2.51) и (2.52) из [13] в безразмерной форме запишутся так:

$$\delta(s_1) = \frac{s_1^{l+1}}{2B_0} (s_1 + \sqrt{s_1^2 - 4B_0}), \quad \alpha(s_1) = \frac{-s_1 + \sqrt{s_1^2 - 4B_0}}{2}, \quad (34)$$

$$\alpha(s_0) = 1. \quad (35)$$

Задача состоит в нахождении $\alpha(s)$ и $\delta(s)$ — решений системы уравнений (33), которые удовлетворяют начальным (34) и граничным (35) условиям.

Для дальнейших рассуждений выполним замену переменных: $\tau = \frac{s - s_1}{s_0 - s_1}$, $x_1 = \alpha - s_1 - \tau(s_0 - s_1)$, $x_2 = \delta$, $\lambda = s_1$. Тогда задачу (33)–(35) после элементарных преобразований можно записать в виде

$$\begin{aligned} \dot{x}_1 &= \left(\frac{A_0 x_2^{n-1} - l}{B_0 - x_1^2} B_0 - 1 \right) (1 - \lambda), \\ \dot{x}_2 &= - \frac{A_0 x_2^{n-1} - l}{B_0 - x_1^2} x_1 x_2 (1 - \lambda), \end{aligned} \quad (36)$$

$$x_1(0) = \frac{-\lambda + \sqrt{\lambda^2 - 4B_0}}{2}, \quad x_2(0) = \frac{\lambda^{l+1}}{2B_0} (\lambda + \sqrt{\lambda^2 - 4B_0}), \quad (37)$$

$$x_1(1) = 0. \quad (38)$$

Итак, мы получили зависящую от параметра систему обыкновенных дифференциальных уравнений. Для нахождения параметра воспользуемся условием (38).

6. Реализация алгоритма для кластерных вычислительных систем. Как было показано выше, предложенный в разделе 3 метод является параллельным. Вычисления каждой итерации для всех пар приближенных значений λ можно проводить независимо друг от друга. Обмен данными будет происходить после каждой итерации — вычисленное новое приближение нужно будет послать соседнему процессору по кольцевой схеме.

Программа, реализующая алгоритм, изложенный в разделе 3 для решения задач (27)–(29), (36)–(38), написана с использованием библиотеки MPI. Схематично программу можно описать следующим образом.

1. Ввод точек a и b отрезка, на котором задачи (27)–(29), (36)–(38) имеют решение и итерационный процесс сходится.

2. Кроме точек a и b определяются точки $\lambda_1, \dots, \lambda_{p-2}$ в интервале (a, b) , где p — число используемых процессоров. Получается p пар точек $(a, \lambda_1), (\lambda_1, \lambda_2), \dots, (\lambda_{p-2}, b), (a, b)$. Каждая пара будет находиться на отдельном процессоре. Обозначим их через λ_1 и λ_2 .

3. Решение задач Коши (27), (28) и (36), (37) для λ_1 и λ_2 выполняется на каждом процессоре одновременно. Поскольку λ_2 на i -м процессоре и λ_1 на $(i+1)$ -м процессоре (по кольцевой схеме) имеют одинаковые значения, задачи (27), (28) и (36), (37) для λ_2 можно решить на отдельных процессорах и полученные значения переслать соседнему процессору, где они будут рассматриваться в качестве решения системы для λ_1 . Целесообразность выбора первого или второго варианта обусловлена временем вычислений, необходимых для решения задачи Коши и скоростью передачи данных на выбранной кластерной системе.

4. Решение задачи Коши (16), (17) для вычисления производных функции φ нужно для реализации метода, описанного в разделе 3, с использованием производных первого порядка.

5. Вычисление λ согласно итерационной формуле (15*), приведенной в разделе 3, на всех процессорах одновременно.

6. Если хотя бы на одном процессоре заданная точность достигнута, процесс заканчивается.

7. Если заданная точность не достигнута ни на одном из процессоров, от j -го процессора вычисленное значение λ пересылается на $(j-1)$ -й процессор по кольцевой схеме. Это значение λ пересылается в λ_2 .

Если применяется алгоритм без использования производных, программа немного упростится — из нее будет изъята реализация этапа 4.

Очевидно, что для реализации указанного алгоритма число используемых процессоров должно быть больше 2. Оценивать эффективность распараллеливания и масштабируемость программы в классическом

понимании будет некорректно, т.к. при изменении числа процессоров меняются разбиение интервала и начальные приближения. А это уже другая задача, которая может потребовать и другое число итераций. Если судить об эффективности, сравнивая времена выполнения задачи на p -процессорной системе с разделением на $p - 1$ интервалов и на одном процессоре с таким же разделением отрезка (a, b) , то можно сказать, что эффективность распараллеливания будет тем больше, чем больше вычислений требуются для решения задач Коши, поскольку задачи Коши решаются в параллельном режиме, а пересылки в любом случае происходят один раз (все процессоры пересылают одно число своему соседу) для каждой итерации. Могут добавиться пересылки, необходимые для реализации второго варианта из этапа 3 данного алгоритма.

7. Результаты численных экспериментов. Для решения систем (27)–(29) и (36)–(38) обыкновенных дифференциальных уравнений первого порядка с параметром применялся метод (14), (15) в режиме, когда задача Коши (14) решается приближенно, а итерационная последовательность в (15) определяется по формуле (15*) (случай а). Для сравнения этот способ применялся в режиме, когда вместо (15) использовался итерационный метод Ньютона (случай б).

В обоих случаях для приближенного решения задачи Коши применялся метод Эйлера, а для приближенного вычисления интеграла — квадратурная формула Симпсона. Точность расчетов составляла $\varepsilon = 10^{-6}$.

Таблица 1

№ варианта	γ	l	n_0	n_1	n_2
1	5/3	1.0	−0.1	2.0	−5/3
2	5/3	1.5	1.0	1.0	−1.0
3	5/3	0.0	1.0	−0.5	0.0

Расчеты для задачи 1 выполнены для значений γ , l , n_0 , n_1 и n_2 , приведенных в табл. 1. Полученные результаты помещены в табл. 2.

Таблица 2

№ варианта	σ_0	случай а)		метод Ньютона (случай б)		прибл. значение λ
		интервал сходимости	число итераций	интервал сходимости	число итераций	
1	0	$1.1 - 2 \cdot 10^8$	15	1.3 – 1.33	4	1.340538
1	10	1.3103 – 1.32	2	1.24 – 1.255	47	1.310369
2	0	0.502 – 0.552	4	0.51 – 0.75	5	0.556423
2	10	0.5 – 0.582	2	0.51 – 0.66	5	0.540998
3	0	$0.52 - 3 \cdot 10^{11}$	12	расходится		0.572396

Расчеты для задачи 2 выполнены для значений параметров A_0 , B_0 и n , приведенных в табл. 3. Полученные результаты помещены в табл. 4.

Таблица 3

№ варианта	A_0	B_0	n
1	0	1.5	2.1
2	0	0.5	2.5
3	10	0.25	0.5

Таблица 4

№ варианта	случай а)		метод Ньютона (случай б)		прибл. значение λ
	интервал сходимости	число итераций	интервал сходимости	число итераций	
1	4.0 – 7.9	4	4.1 – 5.8	14	4.958113
2	1.5 – 3.0	4	1.6 – 2.3	7	1.934315
3	1.4 – 3.0	4	1.41 – 1.57	10 – 19	1.475985

Заметим, что увеличение количества узлов и, тем самым, увеличение количества процессоров заметно улучшает качество метода (увеличивается интервал и скорость сходимости).

СПИСОК ЛИТЕРАТУРЫ

1. Андреев А., Воеводин Вл., Жуматий С. Кластеры и суперкомпьютеры — близнецы или братья? // Открытые системы. 2000. № 5, 6. 9–14.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
3. George F. C. Parallel rootfinding algorithms // Ph.D. Thesis. Department of Mathematics, Michigan State University, USA. 1974.

4. *Miranker W.L.* A survey of parallelism in numerical analysis // SIAM Review. 1971. **13**, N 4. 524–547.
5. *Rice J.R.* Matrix representation of nonlinear equations // Application to Parallel Computation. 1971. **25**, N 116. 639–647.
6. *Gal Sh., Miranker W.L.* Optimal sequential and parallel search for finding a root // J. of Combinatorial Theory (A). 1977. **23**. 1–14.
7. *Miranker W.L.* Parallel methods for solving equations // Math. and Computers in Simulation . 1978. **20**. 93–101.
8. *Schmidt J.W.* On the R-order of coupled sequences // J. Computing. 1981. **26**. 333–342.
9. *Островский А.М.* Решение уравнений и систем уравнений. М.: ИЛ, 1963.
10. *Criado F., Davitashvili T.D., Meladze H.V., N.M.Skhirtladze N.M.* On one numerical method for solving some self-similarity problems of gas dynamics on a multiprocessor // Intern. J. Computer Math. 2000. **74**. 63–85.
11. *Кульчицкий О.Ю., Шимелевич Л.И.* О нахождении начального приближения для метода Ньютона // ЖВМ и МФ. 1974. **14**. 1016–1018.
12. *Форсайт Дж., Малькольм М., Моулер К.* Машинные методы математических вычислений. М.: Мир, 1980.
13. *Волосевич П.П., Леванов Е.И., Схиртладзе Н.М., Лацабидзе Г.С.* Движение поршня с ускорением и замедлением в среде с объемными стоками массы. Препринт Института прикладной математики РАН. № 92. М., 1976.

Поступила в редакцию
01.06.2005
