

УДК 004.3

О.А. Дмитриева
Донецкий национальный технический университет
dmitriv@r5.dgtu.donetsk.ua

Упрощение итераций при параллельной реализации неявных методов решения систем обыкновенных дифференциальных уравнений

В работе для ускорения процесса счета наряду с параллельной реализацией предлагается использование преобразований подобия, приводящее исходный функциональный определитель к нижнему треугольному или диагональному виду. Получаемые на основе такого подхода расчетные схемы решения систем обыкновенных дифференциальных уравнений обладают меньшей вычислительной сложностью и являются весьма эффективными при решении жестких дифференциальных уравнений или их систем.

Ключевые слова: параллельная реализация, преобразования подобия, функциональный определитель, расчетные схемы, системы обыкновенных дифференциальных уравнений, вычислительная сложность, жесткие дифференциальные уравнения

Введение

При реализации численного решения задачи Коши с помощью неявных методов для каждой расчетной точки или блока точек возникает необходимость в решении системы нелинейных уравнений, размерность которой определяется количеством стадий неявного метода и количеством уравнений в системе [1-2]. Классический подход к решению данной проблемы заключается в использовании итерационного метода Ньютона с полным якобианом [3]. При решении жестких систем этот подход является непривлекательным из-за высокой трудоемкости реализации, связанной, прежде всего, с необходимостью многократного переопределения величины шага интегрирования на участках быстрого изменения производной [4-5]. Именно проблемы эффективной реализации решения таких систем являются основной целью данной статьи. Поэтому в работе для ускорения процесса счета наряду с параллельной реализацией предлагается использование преобразований подобия, приводящее исходный функциональный определитель к нижнему треугольному или диагональному виду. Получаемые на основе такого подхода расчетные схемы обладают меньшей вычислительной сложностью и являются весьма эффективными при решении жестких уравнений [6-8].

Уменьшения влияния ошибок при численной реализации

Рассмотрим задачу Коши для обыкновенного дифференциального уравнения

$$x' = f(t, x(t)), \quad x(t_0) = x_0, \quad (1)$$

Один шаг численного решения задачи (1), заданной в виде

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

реализуется с помощью следующих расчетных формул

$$x_{n+1} = x_n + h * \sum_{i=1}^s b_i k_i, \quad (2)$$

где

$$k_i = f(t_n + c_i h, g_i);$$

$$g_i = x_n + h * \sum_{j=1}^s a_{ij} k_j, \quad i = 1, 2, \dots, s.$$

Для системы (1) размерностью m при выборе s -стадийного метода каждая итерация требует s вычислений правой части и включает в себя решение нелинейной системы размерности $m \times s$. Высокая требуемая точность решения, а, следовательно, большое число стадий s , и(или) большая размерность системы m приводят к высокому порядку нелинейной системы размерности $m \times s$. В качестве исходных методов численного решения задачи Коши будем рассматривать неявные методы, которые характеризуются высоким порядком точности и наличием L-устойчивости.

Для уменьшения влияния ошибок округления введем меньшие величины [1]

$$z_i = g_i - x_n, \quad (3)$$

что позволит получить соотношение вида

$$z_i = h \sum_{j=1}^s a_{ij} f(t_n + c_j h, x_n + z_j), \quad (4)$$

$i = 1, 2, \dots, s.$

Если решение системы (4) z_1, z_2, \dots, z_s известно, уравнение (2) определяет явную схему нахождения x_{n+1} , причем реализация этой явной

схемы требует s дополнительных вычислений функции. Так как исходя из построения, матрица коэффициентов неявного метода $[A = (a_{ij})]$ является невырожденной, (4) можно записать в следующем виде

$$\begin{pmatrix} z_1 \\ \dots \\ z_s \end{pmatrix} = A \begin{pmatrix} h\tilde{f}(t_n + c_1 h, x_n + z_1) \\ \dots \\ \tilde{f}(t_n + c_s h, x_n + z_s) \end{pmatrix} \quad (5)$$

Если ввести новые обозначения для коэффициентов,

$$[d]_1, d_2, \dots, d_s = [b]_1, b_2, \dots, b_s)A^{-1},$$

то исходное уравнение будет эквивалентно

$$x_{n+1} = x_n + \sum_{j=1}^s d_j z_j \quad (6)$$

Если уравнение (4) является нелинейным, для его решения необходимо использовать какой-либо итерационный метод простой итерации, релаксации или метод Ньютона. Преимущества последнего было отмечено в работе [2]. Применительно к системе (4) метод Ньютона на каждой итерации потребует решения линейной системы с матрицей

$$(I - hA, J),$$

где I - единичная матрица,

J имеет вид

$$\begin{pmatrix} \frac{\partial \tilde{f}(t_n + c_1 h, x_n + z_1)}{\partial x} \\ \dots \\ \frac{\partial \tilde{f}(t_n + c_s h, x_n + z_s)}{\partial x} \end{pmatrix}.$$

Если заменить все матрицы Якоби

$$\frac{\partial \tilde{f}}{\partial y}(t_n + c_j h, x_n + z_j)$$

приближенной матрицей

$$J \approx \frac{\partial \tilde{f}}{\partial y}(t_n, x_n). \quad (7)$$

тогда для (4) можно будет построить упрощенные итерации по схеме

$$(I - hA, J)\Delta Z^k = -Z^k + h(A, J)F(Z^k), \quad (8)$$

$$Z^{k+1} = Z^k + \Delta Z^k,$$

где $[F(Z)^k] = (z_1^k, z_2^k, \dots, z_s^k)^T$ - k -ое приближение решения,

$$\Delta Z^k = (\Delta z_1^k, \Delta z_2^k, \dots, \Delta z_s^k)^T \text{ - приращения}$$

$$[F(Z)^k] = (f(t_n + c_1 h, x_n + z_1^k), \dots,$$

$$f(t_n + c_s h, x_n + z_s^k))^T$$

Упрощение итераций приведением матрицы к особенному виду

Каждая итерация в (8) требует s вычислений правой части и решения линейной системы размерности $m \times s$. Матрица $(I - hA, J)$ одна и та же для всех итераций. Но, даже если ее LU-разложение для упрощенного приведенного варианта будет находиться только один раз, при большой размерности системы это будет связано со значительными вычислительными затратами.

Чтобы быть более точным, затраты на LU-разложение для системы размерности $m \times s$ определяются соотношением [3]

$$\frac{2s^2 m^3}{3} + O(s^2 m^2).$$

Как правило, последнее слагаемое в этом выражении игнорируется, но трудоемкость LU-разложения все-таки остается высокой. Для сокращения этих затрат предлагается несколько подходов. Самым известным из них является использование преобразований подобия, которыми якобиан вида $(I - hA, J)$ преобразуется к блочно-диагональному [4] $(I - hD, J)$ или треугольному [5] $(I - hT, J)$ виду. Примером такого преобразования может служить следующий

Предварительно умножим соотношение (8) на

$$(hA)^{-1}, I$$

и приведем A^{-1} к диагональному или блочно-диагональному виду

$$Q^{-1}A^{-1}Q = \Delta \quad (9)$$

В преобразованных переменных

$$W^k = (Q^{-1}, J)Z^k$$

итерации будут эквивалентны следующим

$$\begin{aligned} (h^{-1}\Delta, I - I, J)\Delta W^k &= \\ &= -h^{-1}(\Delta, J)W^k + [(QJ)^{-1}, I]F((Q, J)W^k). \end{aligned} \quad (10)$$

или

$$W^{k+1} = W^k + \Delta W^k, \quad (11)$$

Для классических неявных методов, которые используются для решения жестких уравнений и их систем, диагональные вхождения D сложны и требуют дальнейших модификаций, вовлекающих в расчеты сложную арифметику. Но окончательно построенный таким образом итерационный метод очень эффективен, обладает естественным параллелизмом, что позволяет его реализации в параллельных вычислительных системах. Фактически, подходом преобразования подобия, затраты на LU-разложение могут быть уменьшены

до величины $\frac{8m^3}{3}$.

Альтернативный подход непосредственно заменяет функциональный якобиан неявного метода $(I - hA, I)$ диагональным блоком или треугольной блочной матрицей $(I - hB, I)$, где B является диагональной или треугольной матрицей с вещественными диагональными элементами b_{ii} . Этот подход был проанализирован в [9] и [10]. Основные затраты такого подхода состоят в оценке трудоемкости LU-разложения матрицы $(I - hB_{LU}, I)$, которые составляют $O(m^3)$. Кроме того, при использовании параллельной вычислительной системы с s процессорами эти затраты уменьшаются на величину s . Т.е. для многостадийных методов выигрыш в параллельной реализации является очень значительным.

Уменьшение вычислительной сложности треугольным блочным методом может быть достигнуто также, если учитывать вид исходного уравнения или системы уравнений. Так, если исходная система (1) может быть раздроблена на слабо соединенные подсистемы, можно выгодно применить функциональный якобиан параллельного неявного метода к этим свойствам сцепления.

Для больших систем с полной матрицей Якоби J выигрыш возможен, если привести матрицу J к форме Хессенберга

$$S^{-1}JS = H = \begin{pmatrix} * & & * & * \\ * & & & * \\ & \backslash & & \\ & & * & * \\ & & & * \end{pmatrix}$$

На выполнение такого преобразования потребуется $\frac{2m^3}{3}$ операций, умножение треугольной матрицы на вектор потребует $\frac{m^2}{2}$ операций. Таким образом, если имеется представление Хессенберга, решение (10) может быть получено за $O(m^2)$ операций. Такой подход дает очень высокую эффективность, когда матрица Якоби обновляется не на каждом шаге.

Сходимость итерационного процесса с диагональными и треугольными матрицами

Для анализа сходимости определим ошибку, достигаемую на каждой итерации как разность между получаемым результатом и точным решением

$$e^{(k)} = x^{(k)} - X \quad (12)$$

Запишем в зависимости от версии преобразования матрицы к треугольному или диагональному виду в соответствующих формах:

1) для треугольной блочной матрицы

$$T = L + D$$

$$(I - T, h) (e^{(k+1)} - e^{(k)}) = -e^{(k)} + h(A, D) (F(x + e^{(k)}) - F(x))$$

для диагональной матрицы

$$(I - D, h) (e^{(k+1)} - e^{(k)}) = -e^{(k)} + h(A - L, I) (F(x + e^{(k)}) - F(x)) + h(L, I) (F(x + e^{(k+1)}) - F(x))$$

Компоненты $F(x + e^{(k)}) - F(x)$ могут быть уточнены как

$$F(x + e^{(k)}) - F(x) = (I, D)e^{(k)} + (J, D)\Delta_j e^{(k)} + O((e^{(k)})^2).$$

Игнорируя остаточные члены второго порядка, (при анализе сходимости первого порядка), оценки ошибки для треугольного или диагонального представлений могут быть записаны в следующем виде

$$e^{(k+1)} = M(I + P\Delta)e^{(k)}, \quad (13)$$

$$e^{(k+1)} = (I - N\Delta)^{-1} M(I + G\Delta)e^{(k)}, \quad (14)$$

где

$$M = (I - T, h)^{-1} (A - T, h),$$

$$N = (I - D, h)^{-1} (L, h),$$

$$P = (A - T)^{-1} A, L$$

$$G = (A - D)^{-1} (A - L), L.$$

Если мы игнорируем Δ_j (проводится линейный анализ сходимости), то приближения ошибки обеих версий будут характеризоваться только матрицей M

$$e^{(k+1)} = M e^{(k)}, \quad (15)$$

Однако, если значением Δ_j нельзя пренебречь, то приближения ошибок будут оцениваться соотношениями (13-14).

Численный эксперимент

Реализация предлагаемых алгоритмов ориентирована на использование многопроцессорных вычислительных системах SIMD (single instruction stream - multiple data stream) структуры с линейкой процессорных элементов. Набор процессоров известен до начала вычислений и не меняется в процессе счета, при этом каждый процессорный элемент может выполнить любую арифметическую операцию за один такт, временные затраты, связанные с

обращением к запоминающему устройству, отсутствуют.

В качестве примера рассмотрим двухстадийный неявный метод с матрицей

$$A = \begin{pmatrix} 5 & 1 \\ 12 & 12 \\ 3 & 1 \\ 4 & 4 \end{pmatrix},$$

для которой нижняя треугольная матрица будет иметь вид

$$T = \begin{pmatrix} 0.4167 & 0 \\ 0.7500 & 0.4 \end{pmatrix},$$

а диагональная

$$D = \begin{pmatrix} 0.2584 & 0 \\ 0 & 0.6449 \end{pmatrix}.$$

Характеристики параллелизма, ускорение и эффективность, исследовались для систем с изменяющимися трудоемкостями вычисления правых частей ft , принимающими значения $ft = \{10, 50, 100, 500, 5000\}$. Реализация двухстадийного неявного метода с нижней треугольной матрицей на SIMD структуре с числом процессорных элементов, совпадающих с размерностью системы, дает следующие показатели ускорения (рис. 1) и эффективности (рис. 2)

S

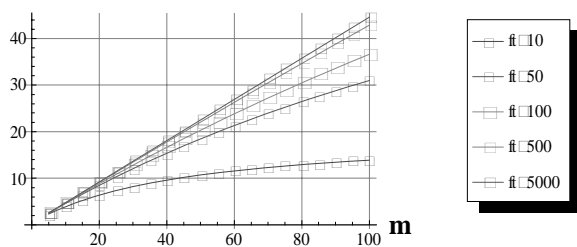


Рисунок 1 – Характеристики ускорения 2-х стадийного метода с нижней треугольной матрицей

E

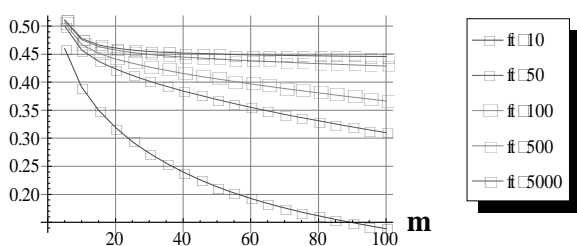


Рисунок 2 – Характеристики эффективности 2-х стадийного метода с нижней треугольной матрицей

Характеристики ускорения и эффективности при реализации двухстадийного неявного метода с диагональной матрицей на SIMD структуре приведены на рис. 3-4.

S

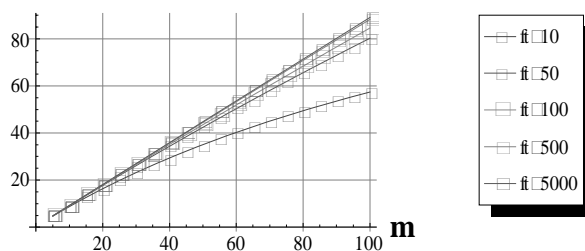


Рисунок 3 – Характеристики ускорения 2-х стадийного метода с диагональной матрицей

E

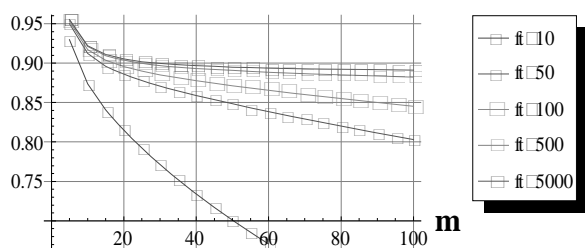


Рисунок 4 – Характеристики эффективности 2-х стадийного метода с диагональной матрицей

Четырехстадийный метод с исходной матрицей

$$A = \begin{pmatrix} 0.1130 & -0.4031 & 0.2580 & -0.0099 \\ 0.2344 & 0.2069 & -0.0479 & 0.0160 \\ 0.2167 & 0.4061 & 0.1890 & -0.2418 \\ 0.2205 & 0.3882 & 0.3288 & 0.0625 \end{pmatrix}$$

будет иметь соответственно треугольную и диагональную матрицы следующего вида

$$T = \begin{pmatrix} 0.1130 & 0 & 0 & 0 \\ 0.2344 & 0.2905 & 0 & 0 \\ 0.2167 & 0.4834 & 0.3083 & 0 \\ 0.2205 & 0.4668 & 0.4414 & 0.1176 \end{pmatrix}$$

$$D = \begin{pmatrix} 0.3205 & 0 & 0 & 0 \\ 0 & 0.0892 & 0 & 0 \\ 0 & 0 & 0.1817 & 0 \\ 0 & 0 & 0 & 0.2334 \end{pmatrix}.$$

S

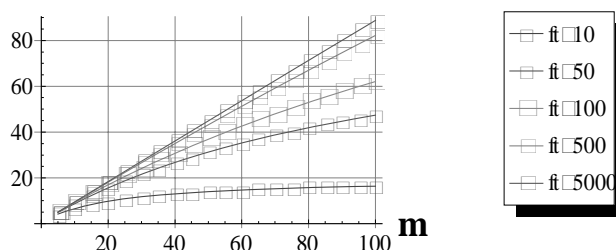


Рисунок 5 – Характеристики ускорения 4-х стадийного метода с нижней треугольной матрицей

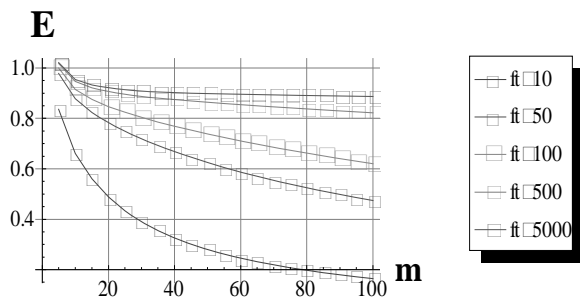


Рисунок 6 – Характеристики эффективности 4-х стадийного метода с нижней треугольной матрицей

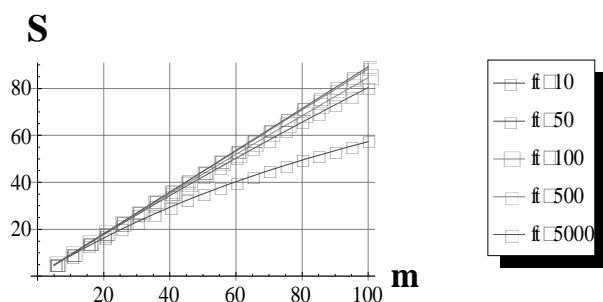


Рисунок 7 – Характеристики ускорения 4-х стадийного метода с диагональной матрицей

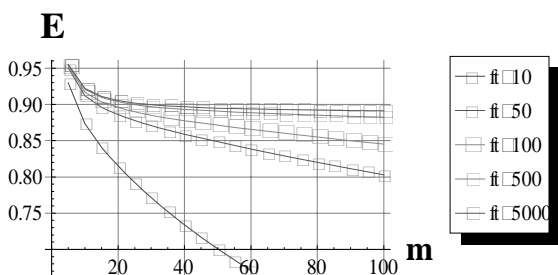


Рисунок 8 – Характеристики эффективности 4-х стадийного метода с диагональной матрицей

Сравнение полученных показателей по группам позволяет утверждать, что показатели неявных методов с диагональными матрицами значительно превосходят соответствующие характеристики для методов с нижними треугольными матрицами, при этом показатели тем лучше, чем выше трудоемкости вычисления правых частей, что объясняется сокращением отношения общего времени счета к времени реализации обменов между процессорными элементами.

Реализация с помощью рассмотренных подходов тестовых задач рассмотрим на примере жесткой задачи Дэвисона (Problem of Davison) [11], в которой рассматривается система из 80 дифференциальных уравнений

$$x'(t) = Ax(t) + \frac{4}{\pi} E_d \sum_{k=0}^4 \frac{\sin((2k+1)\pi t)}{2k+1},$$

где

$$x'(t) = (x_1'(t), x_2'(t), \dots, x_m'(t))^T,$$

$$x(t) = (x_1(t), x_2(t), \dots, x_m(t))^T,$$

$$A = \{a_{ij}\}, \quad a_{ii} = -(1.5)^{m-1},$$

$$a_{i,i-1} = a_{i,i+1} = 0.1, \quad m = 80,$$

$$x(t_0 = 0) = 0, \quad t_{\text{end}} = 5.$$

Результаты реализации системы Дэвисона показаны на рис.9 – 11.

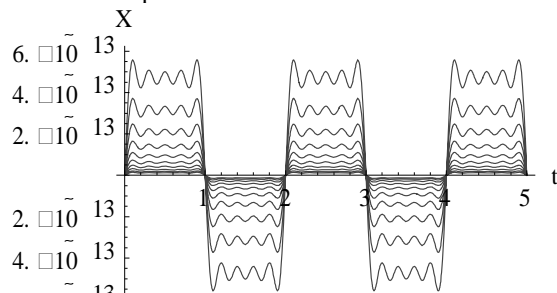


Рисунок 9 – Решение системы Дэвисона для уравнений с 1-го по 10-е

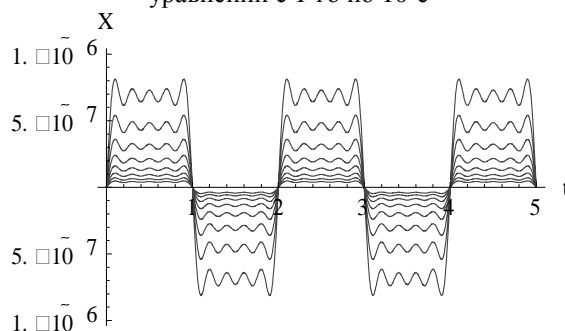


Рисунок 10 – Решение системы Дэвисона для уравнений с 35-го по 45-е

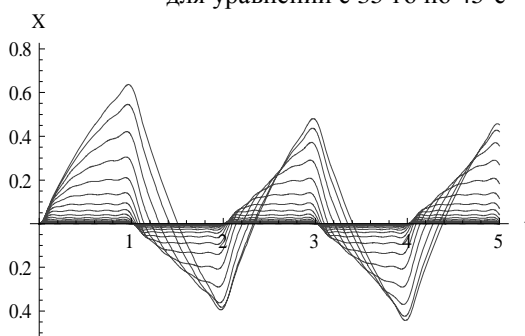


Рисунок 11 – Решение системы Дэвисона для уравнений с 70-го по 80-е

В качестве второй тестовой задачи выберем проблему Шефера (Problem of Schäfer) [12], которая состоит из 8 жестких уравнений (рис.12).

$$x1' = -1.71x1 + 0.43x2 + 8.32x3 + 0.0007,$$

$$x2' = 1.71x1 - 8.75x2,$$

$$x3' = -10.03x3 + 0.43x4 + 0.035x5,$$

$$x4' = 8.32x2 + 1.71x3 - 1.12x4,$$

$$x5' = -1.745x5 + 0.43x7 + 0.43x6,$$

$$\begin{aligned}x_6^f &= -280x_6 * x_8 + 0.69x_4 - \\ &- 1.71x_5 - 0.43x_6 + 0.69x_7, \\ x_7^f &= 280x_6 * x_8 - 1.81x_7, \\ x_8^f &= -280x_6 * x_8 + 1.81x_7.\end{aligned}$$

С начальными условиями

$$\begin{aligned}x_1[t_0] &= 0.0316516757046 & x_2[t_0] &= 0.00648154953106 \\ x_3[t_0] &= 0.00458345106475 & x_4[t_0] &= 0.0897432327352 \\ x_5[t_0] &= 0.162451453753 & x_6[t_0] &= 0.685043896144 \\ x_7[t_0] &= 0.00564670034192 & x_8[t_0] &= 0.0000532996580805\end{aligned}$$

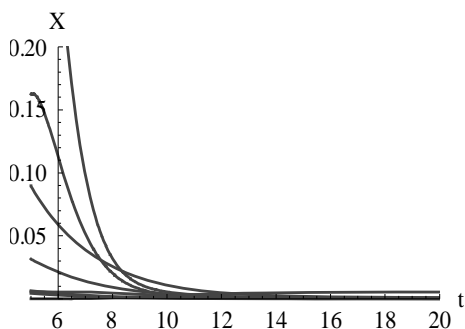


Рисунок 12 – Решение жесткой системы Шефера

Выводы

В работе для ускорения процесса счета наряду с параллельной реализацией предлагается использование преобразований подобия, приводящее исходный функциональный определитель к нижнему треугольному или диагональному виду. Получаемые на основе такого подхода расчетные схемы решения систем обыкновенных дифференциальных уравнений обладают меньшей вычислительной сложностью и являются весьма эффективными при решении жестких уравнений. Выполнен анализ сходимости итерационных процессов, построенных на многостадийных неявных методах решения систем обыкновенных дифференциальных уравнений с диагональной или треугольной матрицами. Проведены численные эксперименты, позволившие получить характеристики параллелизма: коэффициенты ускорения и эффективности для систем с изменяющимися трудоемкостями вычисления правых частей. В качестве тестовых задач были выбраны системы жестких уравнений Дэвисона (Problem of Davison) и Шефера (Problem of Schäfer).

Литература

1. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Жесткие задачи / Э. Хайрер, Г. Ваннер. – М.: Мир, 1999. - 685с.
2. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи / Э. Хайрер, С. Нерсетт, Г. Ваннер. - М.: Мир, 1990. - 512с.
3. P. J. van der Houwen, B. P. Sommeijer. Iteration of Runge-Kutta Methods with Block Triangular Jacobians //Journal of Applied Mathematics and Mechanics Volume 76, Issue 7, pages 367–375, 1996
4. P. J. van der Houwen, J. J. B. de Swart. Triangularly implicit iteration methods for ODE-IVP solvers. CWI Report NM-R9510, submitted for publication 1995. P. 21.
5. Prothero, A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, Math. Comp. 24 (1974) P. 145-162.
6. Дмитриева О.А. Параллельное моделирование жестких динамических систем диагонально неявными методами с адаптацией шага / О.А. Дмитриева, Я.А. Куприй // Наукові праці Донецького національного технічного університету. Серія "Інформатика, кібернетика та обчислювальна техніка". – 2010. – Випуск 12(165). – С. 111-116.
7. Дмитриева О.А. Вариация шага при решении жестких уравнений блочными методами / О.А. Дмитриева // Сборник трудов конференции МОДЕЛИРОВАНИЕ – 2010. 14-16 мая 2010, г. Киев. – К.: Институт проблем моделирования в энергетике, 2010. – Т.2. – С. 5-12
8. Фельдман Л.П. Эффективные методы распараллеливания численного решения задачи Коши для обыкновенных дифференциальных уравнений / Л.П. Фельдман, О.А. Дмитриева // Математическое моделирование. – 2001. – Т. 13. – № 7. – С.66-72.
9. Дмитриева О.А. Управление шагом в блочных диагонально-неявных методах решения обыкновенных дифференциальных уравнений / О.А. Дмитриева // Наукові праці Донецького національного технічного університету. Серія "Інформатика, кібернетика та обчислювальна техніка". – 2010. – Вип. 11(164). – С. 14-18.
10. Feldman L.P., Dmitrieva O.A., Gerber S.. Abbildung der blockartigen Algorithmen auf Parallelrechnerarchitekture. In: Tavangarian,D., Grützner,R. (Hrsg.): Tagungs-band 15. ASIM-Symposium Simulationstechnik in Rostock, September 2002, SCS-Europe BVBA, Ghent/Belgium 2002, S.359-364.
11. Enright, W. H.: Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations. ACM Trans. Math. Software 4 (1978), 127-136.
12. Hairer, E.; Wanner, G.: Solving ordinary differential equations. II: Stiff and differential-algebraic problems. Springer Series in Comput. Math., Vol. 14, Springer-Verlag, Berlin 1991.

Надійшла до редакції 10.02.2011