ПРОБЛЕМЫ РАСПАРАЛЛЕЛИВАНИЯ МНОГОШАГОВЫХ АЛГОРИТМОВ ТИПА АДАМСА

Дмитриева О.А.
Донецкий национальный технический университет
E-mail: dmitriv@r5.dgtu.donetsk.ua

Dmitrieva O.A. Parallelism problems multistep methods of type Adams algorithms. *The work is dedicated to the problems of parallelism multistep algorithms, made for the numerical solution of usual systems of differential equations and researching possibilities of their reflection on the modern calculating multiprocessor systems. Such way of desiding this problem can be effectively used in the modeling of complicated dynamic systems with concentrated parameters and will allow considerably decrease the time of decision, so that the time of modeling too. The offered algorithms are made for using in multiprocessors calculating systems SIMD type with topology structures 1D- and 2D- torus. Here with supposes, that every processors element can do any arithmetic operation, the time expenditures, connected with the calling to memorizing device, are absent. For the estimations of mentioned above algorithms the more widespread criterions are used: the acceleration coefficient and the effectiveness.*

Введение

В числе важнейших работ по развитию вычислительных технологий приоритетным является распараллеливание вычислений [1], создание новых методов и алгоритмов, ориентированных на эффективное использование в многопроцессорных системах [2], а также модернизация существующих с реализацией возможностей широкого параллелизма, поскольку уже в ближайшем будущем перспективы развития суперЭВМ будут определяться успехами не столько электроники, сколько вычислительной математики [3]. В течение последних двух–трех десятилетий пиковая производительность параллельных вычислительных систем возросла на несколько порядков. Радикально изменилась технологическая база и архитектура. Но главным препятствием к внедрению практически всех параллельных архитектур является отсутствие параллельных алгоритмов [4]. Речь, в первую очередь, идет о больших системах дифференциальных уравнений, как обыкновенных, так и в частных производных, являющихся основой больших современных научных и инженерных задач [5]. Многие численные методы решения приходится пересматривать, а от некоторых полностью отказываться [5,6]. В то же время, целый ряд вопросов, которые были несущественны или

вообще не возникали при проведении последовательных вычислений, приобрели исключительную важность для эффективного и правильного использования вычислительных систем. Опыт эксплуатации параллельных систем показал [7], что для эффективного их применения нужно радикально менять структуру численных методов. Эти обстоятельства послужили основой для адаптации известных методов решения систем обыкновенных дифференциальных уравнений на параллельных системах с SIMD архитектурой.

Вывод расчетных формул для многошаговых методов

Пусть математическую модель динамической системы можно представить в виде системы ОДУ с постоянными коэффициентами и начальными условиями

$$\frac{d\bar{x}}{dt} = A\bar{x} + \bar{f}(t), \quad \bar{x}(t_0) = \bar{x}^0 = (x_1^0, x_2^0, ..., x_m^0)^t, \tag{1}$$

где $\bar{x}$ - вектор неизвестных сигналов,

$\bar{f}(t)$ - вектор воздействий, $t \in [0,T]$,

$A = \{a_{i,j}\}, i, j = \overline{1,m}$ - матрица коэффициентов системы.

Численное решение задачи Коши для системы обыкновенных дифференциальных уравнений с постоянными коэффициентами (1) можно получить последовательно по шагам с помощью следующих формул Адамса-Башфорта и Адамса-Моултона [8]:

$$\bar{x}_{k+1}^{(p)} = \bar{x}_k + \frac{h}{24}\left[55(A\bar{x}_k + \bar{f}_k) - 59(A\bar{x}_{k-1} + \bar{f}_{k-1}) + 37(A\bar{x}_{k-2} + \bar{f}_{k-2}) - 9(A\bar{x}_{k-3} + \bar{f}_{k-3})\right], \tag{2}$$

$$\bar{x}_{k+1}^{(k)} = \bar{x}_k + \frac{h}{24}\left[9(A\bar{x}_{k+1}^{(p)} + \bar{f}_{k+1}) + 19(A\bar{x}_k + \bar{f}_k) - 5(A\bar{x}_{k-1} + \bar{f}_{k-1}) - (A\bar{x}_{k-2} + \bar{f}_{k-2})\right], \tag{3}$$

где h - выбранный шаг интегрирования

Приведенный метод в целом является явным. Сначала по формуле Адамса-Башфорта (2) вычисляется значение $\bar{x}_{k+1}^{(p)}$, являющееся прогнозом для $\bar{x}_{k+1}$. Затем $\bar{x}_{k+1}^{(p)}$ используется для расчета скорректированного значения $\bar{x}_{k+1}^{(k)}$, вычисляемого по формуле Адамса-Моултона (3).

Представляемый метод имеет четвертый порядок точности, хотя можно получить эти методы сколь угодно высокого порядка, используя все большее число предыдущих точек, а, следовательно, интерполяционный

полином более высокой степени. Схематически работу методов при этом можно проиллюстрировать следующим образом (рис. 1 – 2). Для расчета прогнозируемых значений в точке t_{n+1} с точностью $O(\tau^k)$ необходимо использовать значения функции в точках $t_{n-k+1}, t_{n-k+2}, \ldots, t_n$.

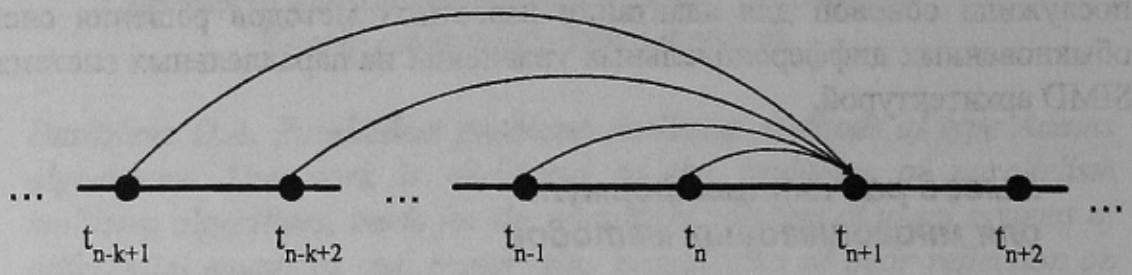


Рисунок 1 - Схема нахождения прогнозируемого значения

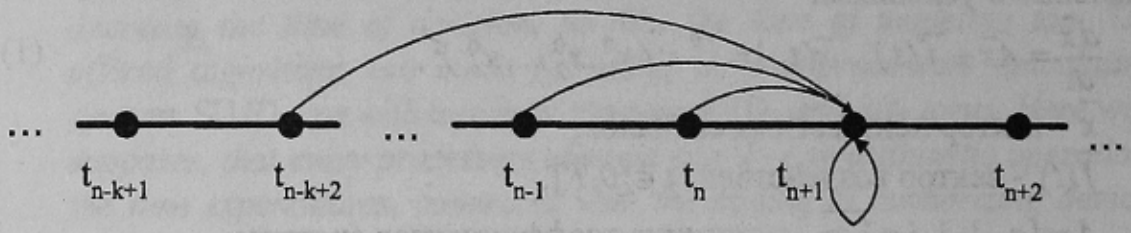


Рисунок 2 - Схема корректировки искомого значения

Коррекция значения в точке t_{n+1} осуществляется за счет использования уже полученного прогнозируемого значения и значений функции в точках $t_{n-k+2}, \ldots, t_n$. С ростом точности формулы становятся все более громоздкими, поскольку в расчетную схему включается все большее число точек, но принцип остается тем же. Если построить интерполяционный многочлен по точкам, решение в которых уже известно, а затем проинтегрировать его, то можно получить прогнозируемое значение в новой точке

$$\bar{x}^{(p)}_{n+1} = \bar{x}_n + \int_{t_{n-p}}^{t_n} L(t)\,dt . \qquad (4)$$

Для корректировки полученного значения снова необходимо проинтегрировать интерполяционный многочлен, который строится на множестве точек, определенном на рис. 2.

$$\bar{x}^{(k)}_{n+1} = \bar{x}_n + \int_{t_{n-p+1}}^{t_{n+1}} L(t)\,dt . \qquad (5)$$

Вычислительные схемы для расчета с привлечением большого числа точек могут быть получены с помощью средств пакета Mathematica®

Приведем в качестве примера формулы прогноза-коррекции восьмого порядка точности

Формула прогноза

$$\frac{1}{120960}\,(h\,(-36799\,F_{-7+n} + 295767\,F_{-6+n} - 1041723\,F_{-5+n} + 2102243\,F_{-4+n} - 2664477\,F_{-3+n} + 2183877\,F_{-2+n} - 1152169\,F_{-1+n} + 434241\,F_n)) + u_n$$

Формула коррекции

$$\frac{1}{120960}\,(h\,(1375\,F_{-6+n} - 11351\,F_{-5+n} + 41499\,F_{-4+n} - 88547\,F_{-3+n} + 123133\,F_{-2+n} - 121797\,F_{-1+n} + 139849\,F_n + 36799\,F_{1+n})) + u_n$$

Рис. 3 Результаты преобразований в системе Mathematica®

Видно, что приведенные формулы являются очень сложными, а получение их коэффициентов вручную является трудноразрешимой задачей. Предложенный подход и использование средств пакета Mathematica® позволяют генерировать многошаговые формулы решения систем обыкновенных дифференциальных уравнений, обеспечивающие любой порядок точности. Это позволяет вычислять коэффициенты формул заданной точности уже на этапе разработки метода.

Отображение на параллельные структуры с решеткой процессорных элементов

Пусть модель, на которую ориентируется решение, имеет следующие особенности: используется вычислительная система SIMD структуры с квадратной сеткой, содержащей m×m процессорных элементов. Для простоты изложения рассматривается случай, когда количество процессорных элементов в каждой строке совпадает с размерностью задачи. Для эффективной работы методов Адамса в каждый процессорный элемент, имеющий индексы i,j, пересылается элемент исходной матрицы матрицы A, приведенной предварительно к виду (6). Таким преобразованием можно избавиться от сдвигов, необходимых на каждом шаге для подготовки умножения матрицы на вектор [9].

$$\breve{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{22} & a_{23} & \dots & a_{21} \\ & \dots & & \\ a_{mm} & a_{m1} & \dots & a_{mm-1} \end{pmatrix}. \qquad (6)$$

Из вектора неизвестных $\bar{x}$ формируется новая матрица, элементы которой также пересылаются в соответствующие процессорные элементы. Таким образом, в каждом процессорном элементе с индексами i,j оказывается соответствующее значение ai,j и элемент новой матрицы. Первоначально выполняется умножение соответствующих значений во всех процессорных элементах. Затем осуществляется одиночный циклический сдвиг полученных значений и сложение содержимого процессорных элементов. Количество позиций, на которое производится очередной сдвиг, определяется элементами следующего ряда: $2^0,2^1,2^2,...,2^{k-1}$, где k = ближайшее целое сверху $[\log_2 m]$. Учитывая, что в процессорных элементах уже находятся рассчитанные ранее значения для предыдущих точек, можно произвести необходимые операции умножения и сложения. Легко видеть, что на такой сетке в каждом ПЭ i-ой строки получается новое значение $x_i^{(p)}$ на n+1 шаге. Вычисление всех значений $x_{n+1}^{(p)}$ определяется временем, затрачиваемым на одно умножение $t_{умн}$, суммой времен, необходимых для осуществления сдвигов и сложения сдваиванием $\sum_{s=0}^{k-1}(t_{сл}+2^s * t_{сдв})$, а также временами $4^{t_{умн}}$ - умножение хранящихся значений на новые коэффициенты, $3^{t_{сл}}$ - суммирование полученных результатов, $t_{умн}+t_{сл}$ время для умножения на h/24 и суммирования со значением, полученным на предыдущем шаге. Всего получаем.

$$T_{m*m}(\bar{x}_{n+1}^{(p)}) = t_{умн} + (m-1)t_{сдв} + \log_2 m t_{сл} + 4t_{умн} + 3t_{сл} + t_{умн} + t_{сл}.$$

Но необходимо учесть, что это время характеризует только нахождение значений вектора неизвестных прогнозируемых значений. При этом, получающиеся в сетке процессорных элементов результаты, расположены не в том порядке, как при изначальной засылке. Для того, чтобы выполнить расчет скорректированных значений, нужно потратить какое-то время на переупорядочивание элементов. На это потребуется m-1 циклических сдвигов по столбцам (в первом -0, во втором 1, ..., в m столбце необходимо выполнить - 2^{k-1} сдвигов). Итого дополнительно потребуется времени $(m-1)*t_{сдв}$. После этого для получения скорректированных результатов нужно повторить ту же последовательность операций, но с другими значениями. Общее время, затрачиваемое на нахождение вектора неизвестных $\bar{x}_{n+1}$ на решетке процессоров $m \times m$, оценивается как

$$T_{m*m}(\bar{x}_{n+1}) = 2*\left[6t_{умн} + (m-1)t_{сдв} + \log_2 m t_{сл} + 4t_{сл} + (m-1)t_{сдв}\right]. \qquad (7)$$

Для оценок качества рассмотренного алгоритма используем критерии ускорения и эффективности [2]. При расчете значений вектора $\bar{x}_{n+1}$ на однопроцессорной ЭВМ потребуется время, равное

$$T(\bar{x}_{n+1}) = m*2*[mt_{умн} + (m-1)t_{сл} + 4t_{умн} + 3t_{сл} + t_{умн} + t_{сл}]. \quad (8)$$

Для получения оценок показателей параллелизма алгоритма воспользуемся соотношениями, приведенными в [10]. Тогда ускорение

$$S_{m*m} \approx \frac{4m^2 t_{оп} + 18mt_{оп}}{2*[0.2(m-1)t_{оп} + (\log_2 m + 4)t_{оп}]} \approx 10*m. \quad (9)$$

Эффективность методов Адамса при решении на SIMD структуре 2D-тор

$$E_{m*m} \approx \frac{10*m}{m^2} \approx \frac{10}{m}. \quad (10)$$

Полученные результаты значительно отличаются от характеристик потенциального параллелизма методов Адамса на такой топологической структуре (ускорение $\approx m^2$, эффективность ≈ 1), поскольку при расчете этих характеристик не учитывались времена, затрачиваемые на сдвиги, хотя, как оказалось, эти времена существенно влияют на показатели параллелизма.

Отображение на параллельные структуры с линейкой процессорных элементов

Одним из часто встречающихся способов коммутации параллельных вычислительных систем является 1D – тор, или линейка процессоров. При реализации решения задачи (2-3) на модели SIMD-структуры, построенной из последовательно соединенных процессорных элементов (последний связан с первым) используются условия, оговоренные ранее. Из всех рассмотренных способов первоначальной засылки значений в процессорные элементы оптимальным оказался способ, когда в i-ый процессорный элемент записываются значения i-ой строки матрицы A: $a_{i1}, a_{i2}, ..., a_{im}$, и значения векторов неизвестных, полученных на предыдущих шагах, первоначально: $\bar{x}_0, \bar{x}_1, \bar{x}_2, \bar{x}_3$, а на $n+1$ шаге $\bar{x}_n, \bar{x}_{n-1}, \bar{x}_{n-2}, \bar{x}_{n-3}$.

При таком подходе одновременно все процессоры могут начать проводить вычисления прогнозируемых значений по (2). Через время

$$m*t_{умн} + (m-1)t_{сл} + 4t_{умн} + 3t_{сл} + t_{умн} + t_{сл} = 2m*t_{оп} + 8t_{оп}$$

в каждом i-том процессорном элементе получаем прогнозируемое значение $\bar{x}^{(p)}_{n+1(i)}$. Эти действия необходимо выполнить только один раз,

перед началом основных вычислений и хранить в каждом i-ом процессорном элементе получившиеся значения:

$$\begin{aligned} & a_{i1}x_{n(1)} + a_{i2}x_{n(2)} + \ldots + a_{im}x_{n(m)} \\ & \ldots \\ & a_{i1}x_{n-3(1)} + a_{i2}x_{n-3(2)} + \ldots + a_{im}x_{n-3(m)}. \end{aligned} \qquad (11)$$

Проиллюстрируем процесс вычислений с помощью следующих упрощенных схем:

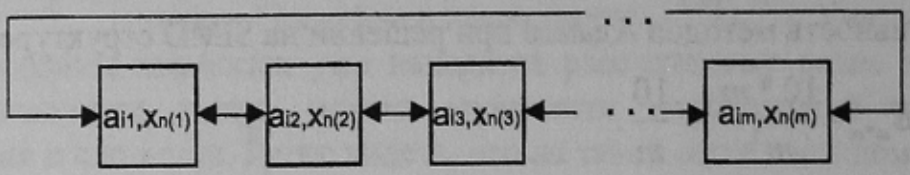


Рисунок 4. Начальная загрузка данных в процессорные элементы

За первый такт во всех процессорных элементах произойдет умножение. Сдвинем циклически содержимое каждого процессорного элемента на один шаг влево и сложим вновь поступившие данные с уже имеющимися (рис.1.4).

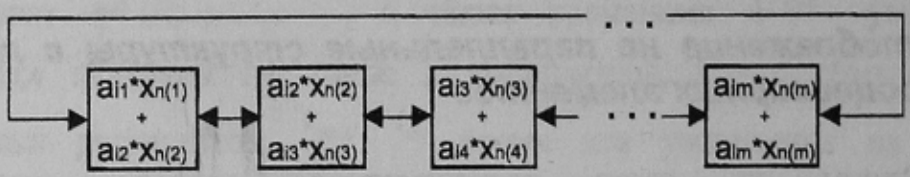


Рисунок 5. Данные процессорных элементов после умножения, сдвига и сложения

Повторим циклический сдвиг, но уже на два шага и сложим полученные результаты

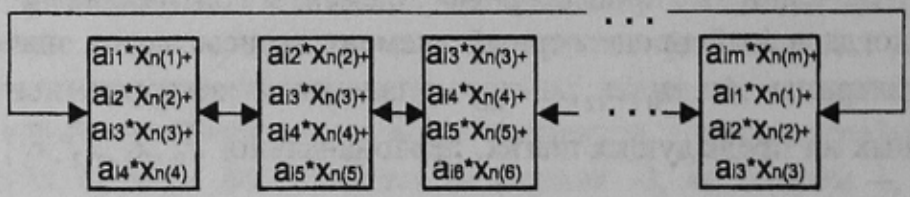


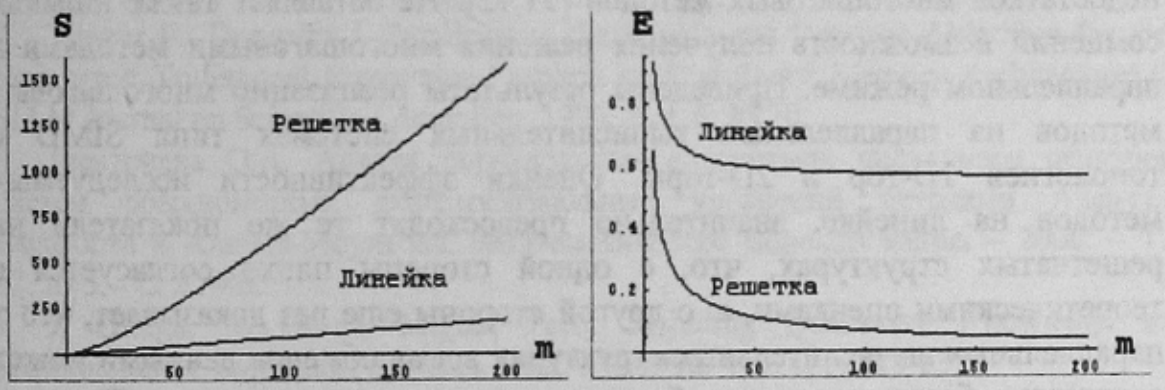
Рисунок 6. Данные процессорных элементов после повторного циклического сдвига

Нетрудно видеть, что после $log_2 n$ сдвигов и $log_2 n$ сложений в каждом процессорном элементе получится результат скалярного произведения двух векторов. Теперь необходимо передать полученные прогнозируемые значения по типу «все-всем», так как в каждом

процессорном элементе нам необходимо иметь значения векторов $\bar{x}^{(p)}_{n+1}$. Время, затрачиваемое на обмены, определится как $(m-1)t_{сдв}$, после чего в каждом ПЭ будут содержаться вектора прогнозируемых значений, и потребуется еще время на переупорядочивание элементов и приведение их к нужному виду. При этом значения, хранящиеся в первом ПЭ, останутся без изменений, во втором на их переупорядочивание потребуется время $(m-1)t_{сдв}$, в третьем - $(m-2)t_{сдв}$, в последнем - $t_{сдв}$. Учитывая, что все сдвиги могут выполняться одновременно, нужное расположение значений получается за время $(m-1)t_{сдв}$, тогда общее время, затрачиваемое на распространение «все-всем», осуществится за время $2(m-1)t_{сдв}$.

Вторая группа сдвигов, потребовавшаяся на переупорядочивание значений, может быть исключена, если провести первоначальную перестановку элементов в строках, т.е. записать в каждый i-ый процессорный элемент i-ую строку матрицы A. Тогда после получения в каждом ПЭ прогнозируемого значения, передача «все-всем» может быть осуществлена за время $(m-1)t_{сдв}$. Перед расчетом скорректированного значения $\bar{x}^{(k)}_{n+1}$ из ПЭ удаляются значения $\bar{x}_{n-3}$ как уже не использующиеся. Для расчета скорректированного значения потребуется время

$$m * t_{умн} + (m-1)t_{сл} + 4t_{умн} + 3t_{сл} + t_{умн} + t_{сл} = 2m * t_{оп} + 8t_{оп},$$



а) ускорение б) эффективность

Рисунок 7 - Сравнительные характеристики параллельных методов Адамса при реализации на SIMD структурах

рассылка «все-всем» $(m-1)t_{сдв}$, тогда полный цикл расчета значений для одной точки закончится через

$$T_m(\bar{x}^{(k)}_{n+1}) = 4m * t_{оп} + 0.2(m-1) * t_{оп} + 8t_{оп}. \qquad (12)$$

Ускорение определяется как

$$S_m = \frac{N*[4m^2*t_{on} + 18m*t_{on}]}{N*[4m*t_{on} + 0.2(m-1)*t_{on} + 8t_{on}]} \approx m, \qquad (13)$$

а эффективность соответственно

$$E_m = \frac{m}{m} \approx 1.$$

И, хотя ускорение для линейки получилось меньше, чем на решетке процессоров, но, учитывая, что эффективность работы этого алгоритма $E_m = \frac{m}{m} \approx 1$, можно говорить, что решение задачи Коши для систем обыкновенных дифференциальных уравнений методами Адамса предпочтительнее проводить на параллельных компьютерах с топологией 1D - тор.

Заключение

Рассмотренные закономерности распараллеливания решения линейных систем обыкновенных дифференциальных уравнений позволили распространить их на любые системы ОДУ. Кроме того, разработанный с помощью подсистемы *Mathematica*® подход, позволил достаточно просто получать расчетные коэффициенты, поскольку именно трудоемкость процедуры получения коэффициентов считалась одним из основных недостатков многошаговых методов [11,12]. Не оставляет также никаких сомнений возможность получения решения многошаговыми методами в параллельном режиме. Приведены результаты реализации многошаговых методов на параллельных вычислительных системах типа SIMD с топологией 1D-тор и 2D-торы. Оценки эффективности исследуемых методов на линейке, значительно превосходят те же показатели на решетчатых структурах, что, с одной стороны плохо согласуется с теоретическими оценками, а, с другой стороны еще раз доказывает, что в параллельных вычислительных структурах время обменов данными может составлять большую часть общего времени решения задачи. Поэтому подходить к выбору структуры вычислительной системы для решения каждой конкретной задачи необходимо не только с позиций наращивания количества процессорных элементов, но и оптимизации алгоритмов для сокращения количества обменов между ними. Модифицированы параллельные алгоритмы умножения матриц, позволяющие в 2 раза сократить число обменов между процессорными элементами, выполняемых на каждом шаге. Сокращение числа обменов достигается за счет изменения порядка начальной загрузки значений коэффициентов матриц в решающее поле микропроцессоров параллельных вычислительных систем.

Разработаны алгоритмы, позволяющие получать с помощью подсистемы *Mathematica®* коэффициенты формул прогноза и коррекции для многошаговых методов решения СОДУ при любой конфигурации процессорного поля.

Литература

1. Жуков І. А. Теорія та принципи організації паралельних обчислювальних структур і систем для розв'язання задач великої розмірності: Автореф. дис… д-ра техн. наук: 05.13.13/ ІПМЕ НАН України. – К., 1997. – 32 с.
2. Воеводин В.В. Математические основы параллельных вычислений. - М.: Изд-во МГУ, 1991. - 345 с.
3. Забродин А.В., Елизаров Г.С., Каратанов В.В. и др. Возможности и ближайшие перспективы создания высокопроизводительных вычислительных систем.//Тезисы докладов конференции "Высокопроизводительные вычисления и их приложения" 30 октября - 2 ноября 2000 года, Черноголовка. – М.:МГУ, 2000.
4. Voevodin V.V. Information structure of sequental programs// Rus. J. Num. An. and Math. Modelling. 1995. Vol. 10, № 3. P. 279-286.
5. Ортега Дж.,Пул У. Введение в численные методы решения дифференциальных уравнений/ Пер. с англ.; Под ред. Абрамова А.А.- М.: Наука. Гл. ред. физ.-мат. лит., 1986.-288 с.
6. G.A. Alverson, W.G. Griswold, C.Lin, D. Notkin and L. Snydler. Abstractions for Portable, Scalable Parallel Programming // IEEE Trans. on Parallel and Distributed Systems, January 1998. Vol. 9, №1, P. 71-86.
7. Tisseur F., Dongarra J. Parallelizing the Divide and Conquer Algorithm for the Symmetric Tridiagonal Eigenvalue Problem on Distributed Memory Architectures.// SIAM Journal on Scientific Computing. 1999. Vol. 6, № 20.
8. Дмитриева О.А. Анализ параллельных алгоритмов численного решения систем обыкновенных дифференциальных уравнений методами Адамса-Башфорта и Адамса-Моултона. //Математическое моделирование. – 2000. – Т. 12, № 5. - С. 81-86.
9. J. Ortega, R. Gasca, Miguel Toro: Including Qualitative Knowledge in Semiqualitative Dynamical Systems. IEA/AIE, 1998. Vol. 1, P. 329-335.
10. Фельдман Л.П., Дмитриева О.А. Эффективные методы распараллеливания численного решения задачи Коши для обыкновенных дифференциальных уравнений. //Математическое моделирование, том+ 13, № 7, 2001. – С.66-72.
11. Z. Bai, J. Demmel, J. Dongarra, A. Petitet, H. Robinson, and K. Stanley. The Spectral Decomposition of Nonsymmetric Matrices on DMCs , IAM Journal on Scientific Computing, Vol. 18, № 5, P. 1446-1461.
12. Дмитриева О.А. Параллельные блочные многошаговые алгоритмы численного решения систем обыкновенных дифференциальных уравнений большой размерности. //Научные труды Донецкого государственного технического университета. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 15: - Донецк:, 2000, с. 53-58.

Дата надходження до редколегії: 16.10.2003 р.