

УДК 519.633:519.684:519.688

ОБЗОР И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПАРАЛЛЕЛЬНЫХ ЧИСЛЕННЫХ АЛГОРИТМОВ РЕШЕНИЯ НЕЛИНЕЙНОГО УРАВНЕНИЯ ШРЁДИНГЕРА

© 2010 г. А. А. Дергачёв¹, О. В. Ефимов², Е. О. Сметанина³

Международный учебно-научный лазерный центр МГУ имени М.В. Ломоносова 119991, ГСП-1, Москва, Ленинские горы, 1, стр. 62.

Рассматриваются параллельные численные методы решения нелинейного уравнения квазиоптики на основе разностных схем и быстрого преобразования Фурье. Приводятся результаты исследования производительности рассматриваемых методов на суперкомпьютере СКИФ «Чебышёв» и кластере IBM BlueGene/P, установленных в Московском государственном университете имени М. В. Ломоносова. Статья подготовлена по результатам обучения по специальной программе «Суперкомпьютерные технологии» и собственных исследований в рамках научной работы.

Ключевые слова: параллельные алгоритмы, нелинейное уравнение Шрёдингера, эффективность распараллеливания, метод прогонки, быстрое преобразование Фурье.

REVIEW AND COMPARATIVE ANALYSIS OF PARALLEL NUMERICAL ALGORITHMS FOR NONLINEAR SCHRÖDINGER EQUATION SOLVING

A. A. Dergachev¹, O. V. Efimov², E. O. Smetanina³

Рассматриваются параллельные численные методы решения нелинейного уравнения квазиоптики на основе разностных схем и быстрого преобразования Фурье. Приводятся результаты исследования производительности рассматриваемых методов на суперкомпьютере СКИФ «Чебышёв» и кластере IBM BlueGene/P, установленных в Московском государственном университете имени М. В. Ломоносова. Статья подготовлена по результатам обучения по специальной программе «Суперкомпьютерные технологии» и собственных исследований в рамках научной работы.

Keywords: parallel algorithms, nonlinear Schrodinger equation, parallelization efficiency, sweep method, fast Fourier transform.

1. Введение и постановка задачи.

Нелинейное уравнение Шрёдингера играет существенную роль во многих областях физики, а также химии, экономики и иных наук. Это уравнение на комплексную функцию $A(\vec{r}, t)$ имеет вид:

$$\alpha i \frac{\partial A}{\partial t} = \beta \frac{\partial^2 A}{\partial x^2} + \gamma |A|^2 A \quad (1)$$

Такое название присвоено уравнению (1) потому, что его линейная часть совпадает с уравнением Шрёдингера. В оптических приложениях чаще используется наименование «нелинейное уравнение квазиоптики».

В нелинейной оптике это уравнение описывает самофокусировку светового пучка в среде с кубической нелинейностью. При описании распространения мощных коротких

¹А. А. Dergachev, dergachev88@yandex.ru

²О. В. Efimov, efimovov@yandex.ru

³Е. О. Smetanina, jannes-2002@yandex.ru

лазерных импульсов в диспергирующих средах нелинейное уравнение для медленно меняющейся комплексной огибающей светового поля дополняют членами, учитывающими дисперсию высших порядков, нелинейность отклика самоиндуцированной плазмы, потери на ионизацию и поглощение лазерного излучения в веществе [1].

В теории лазеров оптического диапазона с помощью нелинейного уравнения квазиоптики изучается усиление излучения, поиск собственных частот и типов колебаний поля в резонаторе. Для исследования динамики усиленного спонтанного излучения в рентгеновском лазере используется квазиоптическое уравнение для поперечной корреляционной функции поля излучения [2].

Распространение лазерных импульсов в оптических волноводах также описывается нелинейным уравнением квазиоптики для комплексной огибающей светового поля, частными решениями которого являются темные (в области нормальной дисперсии групповых скоростей) и светлые (в области аномальной дисперсии групповых скоростей) солитоны [3, 4, 5].

В физике сверхнизких температур нелинейное уравнение Шрёдингера используется для описания поведения неидеального бозе-газа со слабым взаимодействием между частицами [6]. В работе [7] обращается внимание на математическую аналогию между теорией солитонных волн материи и теорией оптических солитонов в волоконных световодах. Также нелинейное уравнение Шрёдингера используется в ядерной физике в рамках квантово-гидродинамической модели [8], и, собственно, в гидродинамике для описания волн на поверхности жидкости [9].

В данной статье будет рассматриваться нелинейное уравнение квазиоптики в применении к задаче самофокусировки лазерного пучка. Эффект самофокусировки возникает под действием изменения показателя преломления, вызванного Керровским эффектом. В этом случае уравнение (1) с начальными условиями для гауссового пучка записывается в виде:

$$\left\{ \begin{array}{l} 2ik \frac{\partial E}{\partial z} = \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2} + \frac{2k^2}{n_0} n_2 |E(x, y, z)|^2 E(x, y, z) \\ E(x, y, 0) = E_0 \exp \left\{ -\frac{x^2 + y^2}{2a_0^2} \right\}, \quad (x, y) \in [-l, l]^2 \end{array} \right. \quad (2)$$

Здесь $E(x, y, z)$ — напряжённость электрического поля, k — волновое число, z — координата вдоль оси распространения светового пучка, x, y — координаты в поперечном сечении, n_0 и n_2 — линейный и нелинейный коэффициенты преломления. В данном уравнении учтены такие физические факторы, как дифракция лазерного пучка и кубическая (по полю) нелинейность.

После обезразмеривания входящих в уравнение величин на их характерные значения $E = \tilde{E} \cdot E_0$, $x = \tilde{x} \cdot a_0$, $y = \tilde{y} \cdot a_0$, $z = \tilde{z} \cdot ka_0^2$ система будет выглядеть следующим образом:

$$\left\{ \begin{array}{l} 2i \frac{\partial \tilde{E}}{\partial \tilde{z}} = \Delta_{\perp} \tilde{E} + R |\tilde{E}|^2 \tilde{E} \\ \tilde{E}(\tilde{x}, \tilde{y}, 0) = \exp \left\{ -\frac{\tilde{x}^2 + \tilde{y}^2}{2} \right\}, \quad (x, y) \in \left[-\frac{l}{a_0}, \frac{l}{a_0} \right]^2 \end{array} \right. \quad (3)$$

Здесь введены обозначения для поперечной части лапласиана Δ_{\perp} и коэффициента

нелинейности R , которые будут использоваться в дальнейшем:

$$\Delta_{\perp} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (4)$$

$$R = \frac{2k^2}{n_0} n_2 E_0^2 a_0^2 \quad (5)$$

2. Причины необходимости использования параллельных методов решения.

Основные проблемы численного моделирования задачи филаментации лазерных импульсов связаны с многомасштабностью задачи. Поперечные масштабы пучка примерно на два порядка превосходят возникающие в нем структуры. В то же время размер расчётной сетки должен на порядок превосходить радиус пучка, чтобы границы сетки не отсекали существенные части пучка, а также чтобы иметь некоторую «буферную область», в которую могла бы расширяться низкоинтенсивная периферийная часть пучка, которая существенно влияет на распространение филамента [1]. В противном случае также неизбежно возникновение краевых эффектов, приводящих к искажению решения. На диаметр филамента должно приходиться достаточное количество точек (не менее 10), иначе резкие перепады интенсивности в окрестности филамента будут содержать слишком высокие пространственные частоты, что приведёт к невыполнению критерия Найквиста, наложению частот и, как следствие, неадекватности получаемого решения. Как показывает практика, этот фактор является важным не только для метода решения, основанного на преобразовании Фурье, но и для остальных методов.

Таким образом, количество точек в поперечном сечении может достигать 10^4 по каждой поперечной координате. Рассматриваемая в статье задача не имеет временной зависимости, однако в реальных задачах филаментации рассматриваются короткие лазерные импульсы. Для них количество временных слоев должно быть порядка $10^2 - 10^3$, а значит общее количество точек достигает величины порядка 10^{11} , а потребность в оперативной памяти — величины около 100 Гб.

3. Параллельные алгоритмы решения задачи.

Нами рассматриваются три метода численного решения уравнения (3). Первый из них основан на использовании явной разностной схемы. Два других предусматривают предварительное расщепление по физическим факторам, при котором интегрирование нелинейного уравнения квазиоптики сводится к последовательному интегрированию на каждом шаге интегрирования двух уравнений, первое из которых описывает только дифракцию, а второе — только нелинейность. Эти уравнения имеют следующий вид:

$$\begin{cases} 2i \frac{\partial E}{\partial z} = \Delta_{\perp} E \\ 2i \frac{\partial E}{\partial z} = R |E|^2 E \end{cases} \quad (6)$$

Если считать интенсивность поля ($I \sim |E|^2$) неизменной на протяжении одного шага нелинейности, что соблюдается при маленьких шагах по z , то интегрирование

уравнения для нелинейности не представляет проблем:

$$E(x, y, z_i + \Delta z) = E(x, y, z_i) \exp \left(-\frac{iR}{2} |E(x, y, z_i)|^2 \Delta z \right) \quad (7)$$

Отметим, что поскольку нелинейность является локальной, то есть набег фазы в точке поперечного сечения зависит только от значения интенсивности поля в этой же точке, то для решения второго уравнения из системы (6) можно успешно применить метод геометрического параллелизма, который будет иметь идеальную масштабируемость при любом количестве используемых для вычислений процессоров.

Для интегрирования уравнения дифракции в случае расщепления по физическим факторам были использованы два метода: метод на основе неявной разностной схемы и метод, использующий быстрое преобразование Фурье.

3.1. Метод на основе явной разностной схемы.

Уравнение (3) можно переписать в виде:

$$\begin{aligned} \Delta E(x, y, z_i) &= E(x, y, z_i + \Delta z) - E(x, y, z_i) = \frac{\partial E(x, y, z_i)}{\partial z} \Delta z = \\ &= \frac{1}{2i} (\Delta_{\perp} E(x, y, z_i) + R |E(x, y, z_i)|^2 E(x, y, z_i)) \Delta z = f(x, y, z_i) \Delta z \end{aligned} \quad (8)$$

Для его решения не достаточно применять метод Эйлера, а нужно использовать более устойчивые методы из класса «предиктор-корректор», например, метод Рунге-Кутты 4-го порядка. К сожалению, применение метода Рунге-Кутты ограничено критерием Куранта-Фридрихса-Леви, по которому шаг по оси z должен быть меньше чем $c \times \max(\Delta x, \Delta y)^2$, что при увеличении количества точек в поперечном сечении приводит к чрезвычайно малому шагу по z . Метод Эйлера, соответственно, требует ещё меньшего шага для получения сколько-нибудь реалистичного решения.

Кроме того, для реализации простейшей схемы расчёта n -го порядка точности с помощью явной схемы необходимо на каждом шаге рассчитать n матриц размером в поперечную сетку. Это требует очень больших затрат памяти, что является минусом этого алгоритма.

Возможным решением проблемы расходования памяти и большого количества пересылок является использование коэффициентов, отличных от классических в формулах Рунге-Кутты, например рассчитанных на основе квадратурных формул Раддо и Лобатто[10], которые дают аналогичный порядок точности при использовании меньшего количества ступеней в методе Рунге-Кутты. При использовании этого метода стоит обратить внимание на правильную организацию пересылок данных при синхронизации промежуточных результатов между процессорами на каждой ступени алгоритма. Подробнее про возможность использования параллельного метода Рунге-Кутты можно ознакомиться в [11, 12]

3.2. Метод на основе преобразования Фурье.

«Метод Фурье» основан на переходе к двумерному фурье-образу матрицы поля:

$$E(k_x, k_y, z) = \iint E(x, y, z) e^{-ik_x x - ik_y y} dx dy \quad (9)$$

Первое из уравнений системы (6) в фурье-представлении будет выглядеть следующим образом:

$$2i \frac{\partial E(k_x, k_y, z)}{\partial z} = (-k_x^2 - k_y^2) E(k_x, k_y, z) \quad (10)$$

Его решение определяется формулой

$$E(k_x, k_y, z + \Delta z) = E(k_x, k_y, z) \exp \left\{ \frac{i}{2} (k_x^2 + k_y^2) |E(k_x, k_y, z)|^2 \right\} \quad (11)$$

Для выполнения быстрого преобразования Фурье использовалась свободно распространяемая библиотека FFTW (версии 2.3) [13]. Подробнее с реализацией и методом создания алгоритма для FFTW можно ознакомиться в статьях [14, 15]. Данная реализация БПФ предполагает ленточное распределение матрицы поля по процессам. Кроме того, FFTW, как любое быстрое преобразование Фурье, эффективнее работает на матрицах, размеры которых являются степенями двойки.

Рассмотрим алгоритм параллельного двумерного преобразования Фурье. Вначале выполняется быстрое фурье-преобразование по строкам. Этот этап происходит локально на каждом процессе, поскольку процесс содержит в своей оперативной памяти всю строку матрицы. Затем происходит транспонирование распределённой матрицы, что связано с обменами данными между всеми процессами (то есть каждый обменивается с каждым). Далее снова выполняется быстрое фурье-преобразование по строкам, которые до транспонирования являлись столбцами.

Существенной особенностью алгоритмов БПФ является расположение полученных коэффициентов в памяти процессоров. Для преобразования Фурье естественным является транспонированное расположение результата в памяти всех процессов. Этому соответствует ключ

FFTW_TRANSPOSED_ORDER функции, реализующей преобразование Фурье. Его альтернативой является ключ FFTW_NORMAL_ORDER, при задании которого после выполнения преобразования Фурье проводится дополнительное транспонирование матрицы спектра. Кроме того существует параметр указанной функции, позволяющий использовать дополнительное временное массива для ускорения преобразования. Наконец, при создании плана фурье-преобразования существует возможность оптимизировать план с целью ускорения работы функции. Это достигается использованием ключей FFTW_ESTIMATE (грубая оценка) и FFTW_MEASURE (при этом производятся замеры времени пересылок, выполняемых в фурье-преобразовании и их оптимизация). Как показали тесты, применение этих ключей обосновано для последовательного преобразования, тогда как для параллельной версии различие скорости расчётов при использовании и без использования этого ключа отличаются в пределах статистической ошибки.

Учёт нелинейности при использовании данного метода осуществляется в рамках описанного в разделе 3. расщепления по физическим факторам.

3.3. Метод на основе неявной разностной схемы.

Этот метод основан на использовании консервативной разностной схемы для неравномерных сеток в поперечном сечении и был предложен в [17]. Она выглядит следу-

ющим образом:

$$\left\{ \begin{array}{l} 2i \frac{h_1}{2} \frac{\hat{E}_{0,j} - E_{0,j}^k}{\Delta z} = \frac{1}{2} \left(\frac{\hat{E}_{1,j} - \hat{E}_{0,j}}{h_1} \right) + \frac{1}{2} \left(\frac{E_{1,j}^k - E_{0,j}^k}{h_0} \right) \\ 2i \frac{h_{i+1} + h_i}{2} \frac{\hat{E}_{i,j} - E_{i,j}^k}{\Delta z} = \frac{1}{2} \left(\frac{\hat{E}_{i+1,j}}{h_{i+1}} - \left(\frac{1}{h_{i+1}} + \frac{1}{h_i} \right) \hat{E}_{i,j} + \frac{\hat{E}_{i-1,j}}{h_i} \right) \\ \quad + \frac{1}{2} \left(\frac{E_{i+1,j}^k}{h_{i+1}} - \left(\frac{1}{h_{i+1}} + \frac{1}{h_i} \right) E_{i,j}^k + \frac{E_{i-1,j}^k}{h_i} \right), \quad i = 1, \dots, N \\ 2i \frac{h_N}{2} \frac{\hat{E}_{N,j} - E_{N,j}^k}{\Delta z} = \frac{1}{2} \left(\frac{\hat{E}_{N,j} - \hat{E}_{N-1,j}}{h_N} \right) + \frac{1}{2} \left(\frac{E_{N,j}^k - E_{N-1,j}^k}{h_{N-1}} \right) \end{array} \right. \quad (12)$$

где $h_i = x_i - x_{i-1}$.

Указанная схема осуществляет расчёт «дифракции по x ». Аналогичная система разностных уравнений рассчитывает «дифракцию по y ». Если шаг сетки равномерный, то есть $h_i = h = \text{const}$, то схема переходит в хорошо известную схему Кранка-Николсона:

$$\left\{ \begin{array}{l} 2i \frac{\hat{E}_{0,j} - E_{0,j}^k}{\Delta z} = \left(\frac{\hat{E}_{1,j} - \hat{E}_{0,j}}{h^2} \right) + \left(\frac{E_{1,j}^k - E_{0,j}^k}{h^2} \right) \\ 2i \frac{\hat{E}_{i,j} - E_{i,j}^k}{\Delta z} = \frac{1}{2} \left(\frac{\hat{E}_{i+1,j} - 2\hat{E}_{i,j} + \hat{E}_{i-1,j}}{h^2} \right) \\ \quad + \frac{1}{2} \left(\frac{E_{i+1,j}^k - 2E_{i,j}^k + E_{i-1,j}^k}{h^2} \right), \quad i = 1, \dots, N \\ 2i \frac{\hat{E}_{N,j} - E_{N,j}^k}{\Delta z} = \left(\frac{\hat{E}_{N,j} - \hat{E}_{N-1,j}}{h^2} \right) + \left(\frac{E_{N,j}^k - E_{N-1,j}^k}{h^2} \right) \end{array} \right. \quad (13)$$

Указанная система является системой относительно значений поля на промежуточном слое $\hat{E}_{i,j}$. Матрицы систем (12) и (13) трехдиагональная, а значит их можно решать методом прогонки [16]. Рассмотрим параллельную реализацию метода прогонки.

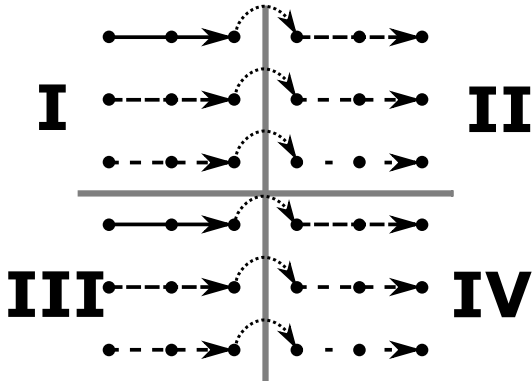


Рис. 1. Прямая прогонка.

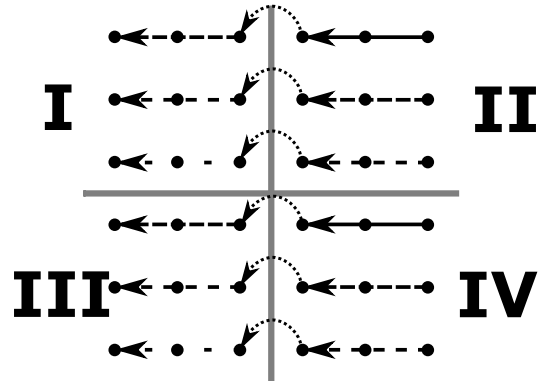


Рис. 2. Обратная прогонка.

На рис. 1, 2 черными точками изображены точки расчетной сетки. Вертикальная и горизонтальная светло-серые прямые показывают разбиение матрицы поперечного

сечения между процессами. Римские цифры обозначают номера процессов. Стрелки соответствуют вычислению прогоночных коэффициентов. Стрелки точками отображают обмены данными между процессами.

Первому расчётному шагу соответствуют сплошная стрелка, а следующим стрелки пунктиром с увеличивающимся расстоянием между штрихами. Для цикла прямой прогонки их выполняют процессы первого столбца матрицы процессов. Далее эти процессы пересылают посчитанные коэффициенты в граничной точке соседнему справа процессу. Таким образом, по каждой строке матрицы процессов запускается конвейерная схема параллельных вычислений.

В цикле обратной прогонки конвейерная схема запускается в обратную сторону (соответствие цвета стрелок и порядка выполнения расчётных операций то же). Пересылке теперь подвергаются рассчитанные значения поля на промежуточном шаге.

Отметим несколько особенностей данного метода. Во-первых, поскольку цикл обратной прогонки запускается после окончания прямой прогонки по всем строкам расчётной сетки, то необходимо хранить в памяти прогоночные коэффициенты для всех строк, то есть необходима дополнительная матрица для прогоночных коэффициентов того же размера, что и матрица поля. Во-вторых, для расчёта коэффициентов, фигурирующих в разностных системах (12) и (13), необходим обмен значениями поля в граничных областях. В-третьих, для высокой эффективности параллельного алгоритма необходимо, чтобы количество строк матрицы поля у отдельного процесса было существенно больше количества процессов в строке матрицы процессов. Действительно, если число строк у отдельного процесса равно N_{loc} , а число процессов в строке матрицы процессов равно q (на рис. 1, 2 $N_{loc} = 3$, $q = 2$), то на цикл прямой прогонки потребуется $N_{loc} + q$ итераций. Таким образом, верхняя оценка на эффективность имеет вид:

$$E_n \leq \frac{N_{loc}q^2}{q^2(N_{loc} + q)} = \frac{1}{1 + q/N_{loc}}. \quad (14)$$

Заметим, что в этом соотношении не учтены потери времени на пересылки и синхронизацию процессов.

Учёт нелинейности при использовании данного метода осуществляется в рамках описанного в разделе 3. расщепления по физическим факторам.

3.4. Сводная информация о рассмотренных методах.

Метод	Количество операций	Объём пересылок	Необходимая память
Рунге-Кутты	$O(N^2)$	$16Np$	$6N^2$
С БПФ при $N = 2^k$	$O(N^2 \log(N))$	$O(N \log(N))p$	N^2 или $2N^2$
С неявной схемой	$O(N^2)$	$4Np$	$2N^2$

Таблица 1. Оценки количества вычислительных операций, сетевых обращений и потребляемой памяти рассматриваемыми методами.

4. Результаты.

Ниже представлены результаты замеров времени работы алгоритма с использованием преобразования Фурье при различных комбинациях параметров, выбранных для анализа их влияния на время работы программы. Замеры времени осуществлялись на кластере СКИФ МГУ «Чебышёв» 10 раз в разное время суток и усреднялись.

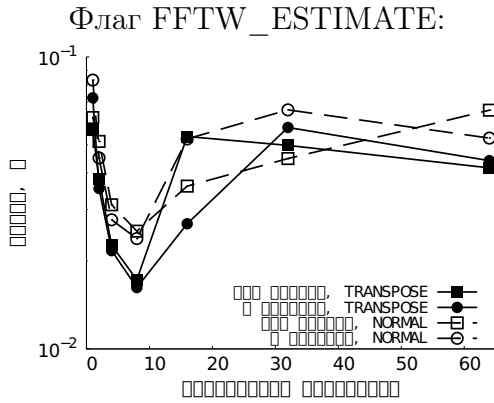


Рис. 3. Время работы Фурье-алгоритма в зависимости от количества процессов. Размер матрицы 512×512 .

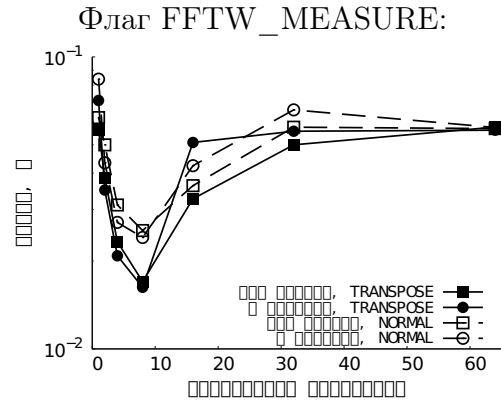


Рис. 4. Время работы Фурье-алгоритма в зависимости от количества процессов. Размер матрицы 512×512 .

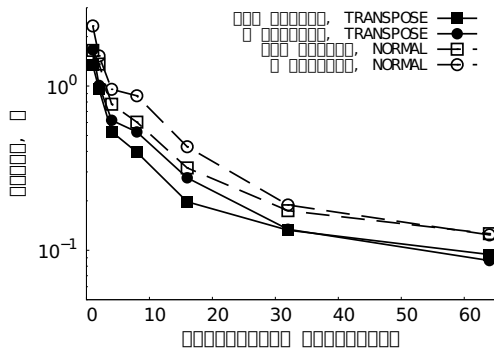


Рис. 5. Время работы Фурье-алгоритма в зависимости от количества процессов. Размер матрицы 2048×2048 .

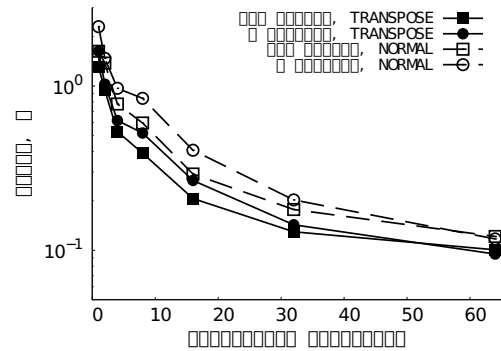


Рис. 6. Время работы Фурье-алгоритма в зависимости от количества процессов. Размер матрицы 2048×2048 .

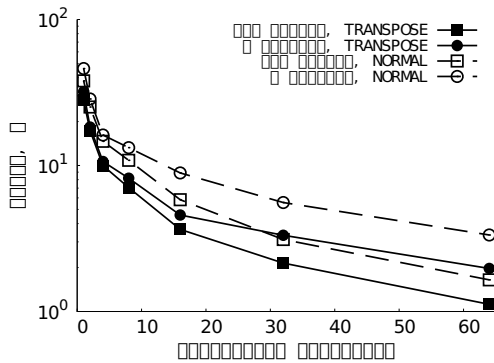


Рис. 7. Время работы Фурье-алгоритма в зависимости от количества процессов. Размер матрицы 8192×8192 .

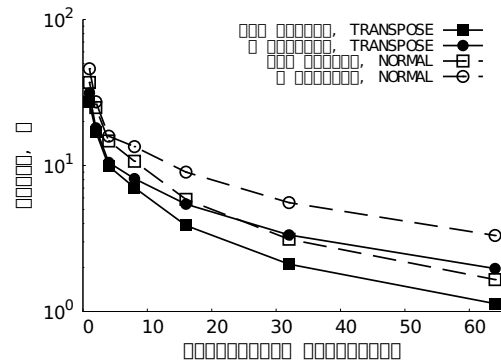


Рис. 8. Время работы Фурье-алгоритма в зависимости от количества процессов. Размер матрицы 8192×8192 .

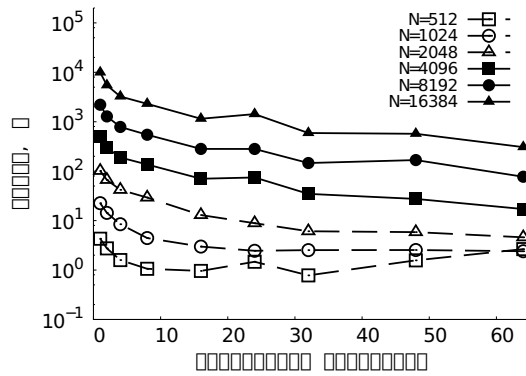


Рис. 9. Время работы Фурье-алгоритма в зависимости от количества процессов. Сохранение данных в файл не производилось.

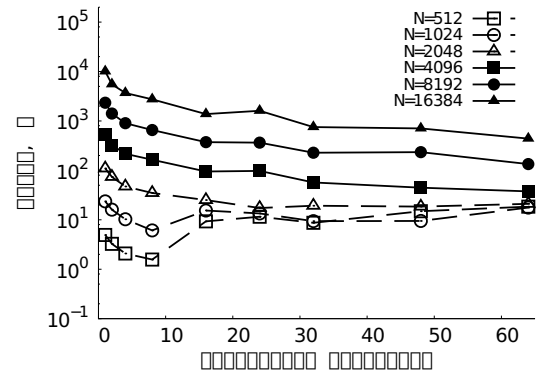


Рис. 10. Время работы Фурье-алгоритма в зависимости от количества процессов. Сохранение данных в файл производилось на каждом десятом шаге.

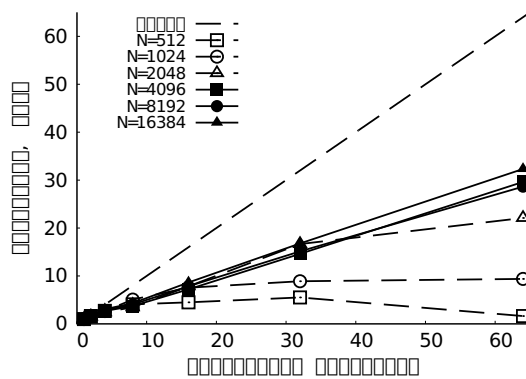


Рис. 11. Ускорение Фурье-алгоритма в зависимости от количества процессов. Сохранение данных в файл не производилось.

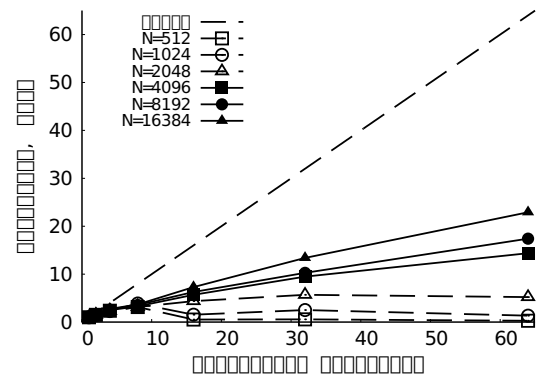


Рис. 12. Ускорение Фурье-алгоритма в зависимости от количества процессов. Сохранение данных в файл производилось на каждом десятом шаге.

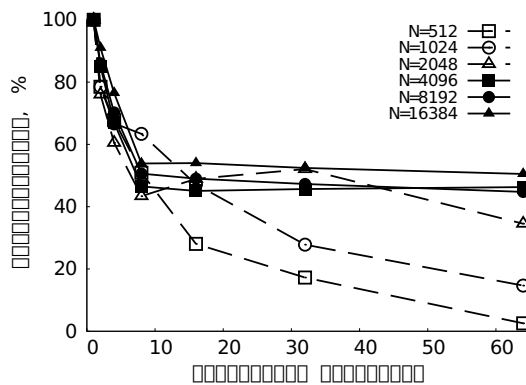


Рис. 13. Эффективность Фурье-алгоритма в зависимости от количества процессов. Сохранение данных в файл не производилось.

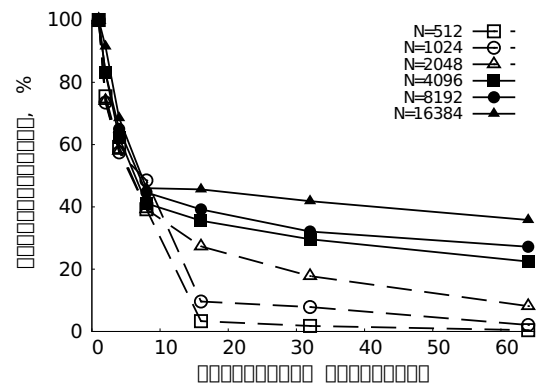


Рис. 14. Эффективность Фурье-алгоритма в зависимости от количества процессов. Сохранение данных в файл производилось на каждом десятом шаге.

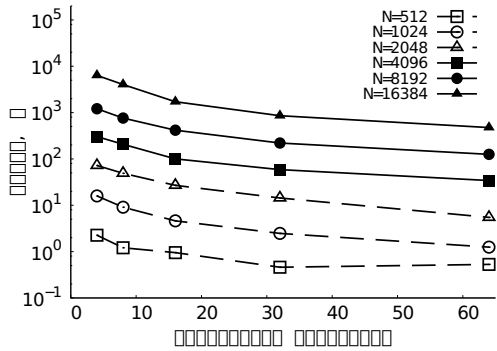


Рис. 15. Время работы алгоритма с неявной схемой в зависимости от количества процессов. Сохранение данных в файл не производилось.

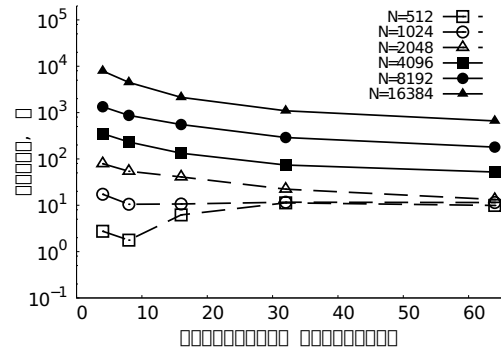


Рис. 16. Время работы алгоритма с неявной схемой в зависимости от количества процессов. Сохранение данных в файл производилось на каждом десятом шаге.

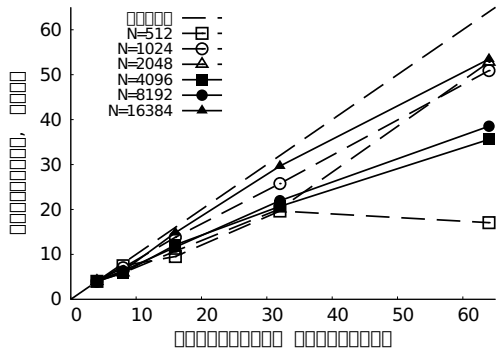


Рис. 17. Ускорение алгоритма с неявной схемой в зависимости от количества процессов. Сохранение данных в файл не производилось.

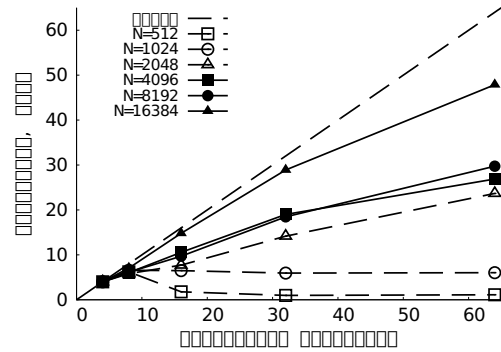


Рис. 18. Ускорение алгоритма с неявной схемой в зависимости от количества процессов. Сохранение данных в файл производилось на каждом десятом шаге.

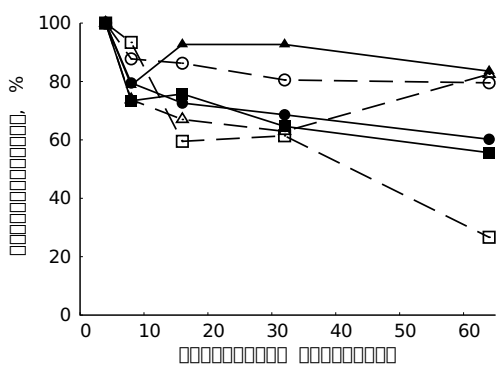


Рис. 19. Эффективность алгоритма с неявной схемой в зависимости от количества процессов. Сохранение данных в файл не производилось.

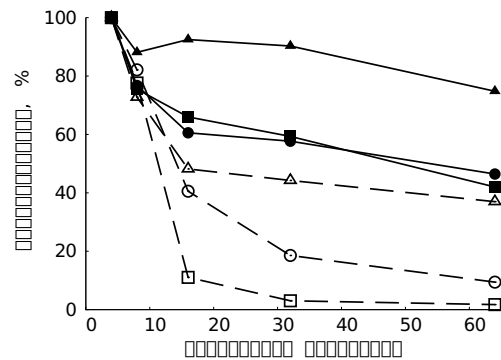


Рис. 20. Эффективность алгоритма с неявной схемой в зависимости от количества процессов. Сохранение данных в файл производилось на каждом десятом шаге.

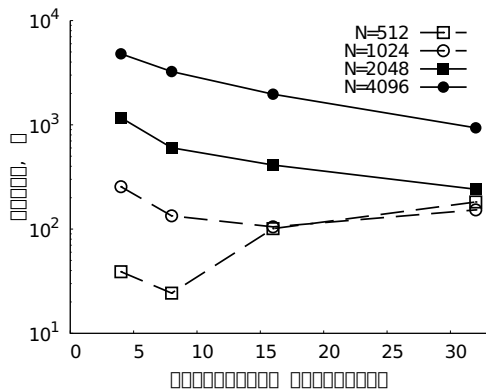


Рис. 21. Время работы алгоритма с неявной схемой на кластере IBM BlueGene/P. Сохранение данных в файл не производилось.

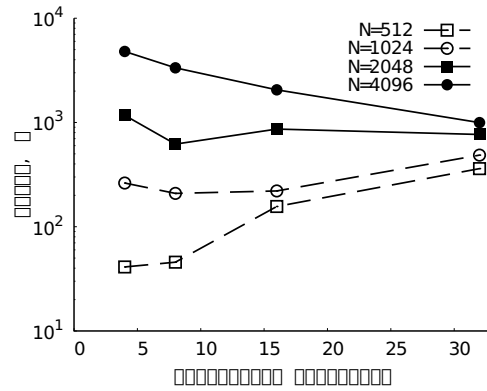


Рис. 22. Время работы алгоритма с неявной схемой на кластере IBM BlueGene/P. Сохранение данных в файл производилось на каждом десятом шаге.

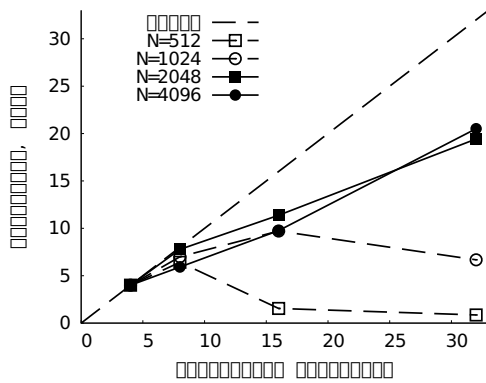


Рис. 23. Ускорение параллельного алгоритма с неявной схемой на кластере IBM BlueGene/P. Сохранение данных в файл не производилось.

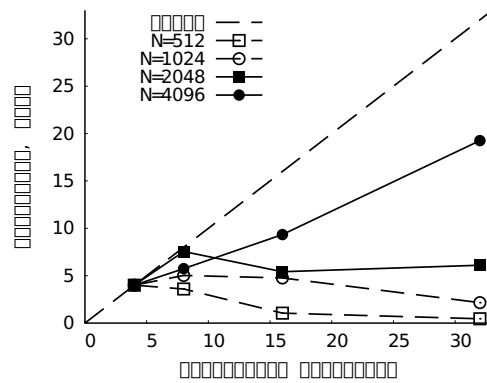


Рис. 24. Ускорение параллельного алгоритма с неявной схемой на кластере IBM BlueGene/P. Сохранение данных в файл производилось на каждом десятом шаге.

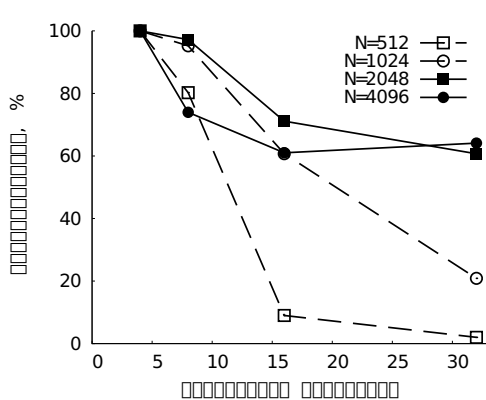


Рис. 25. Эффективность параллельного алгоритма с неявной схемой на кластере IBM BlueGene/P. Сохранение данных в файл не производилось.

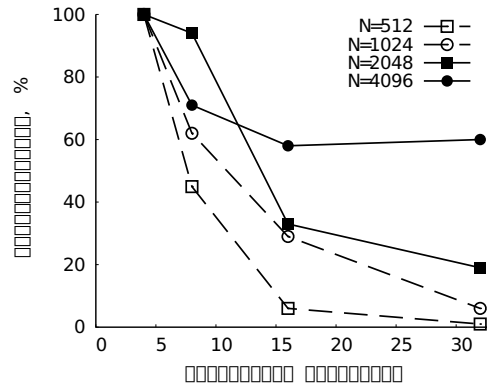


Рис. 26. Эффективность параллельного алгоритма с неявной схемой на кластере IBM BlueGene/P. Сохранение данных в файл производилось на каждом десятом шаге.

Из представленных на рис. 3, 4 графиков видно, что использование более 8 процессоров является неэффективным для матрицы размером 512×512 , так как время работы программы возрастает по сравнению с временем работы программы при тех же параметрах на 8 процессорах. Для размера матрицы 2048×2048 , как видно из рис. 5, 6 увеличение числа процессоров дает ощутимое уменьшение времени работы программы. При использовании матрицы размером 8192×8192 времена работы программы увеличиваются, что позволяет более четко увидеть различие во временах работы программы для различных комбинаций флагов при использовании библиотеки FFTW.

Из графиков на рис. 3–8 видно, что использования флага FFTW_MEASURE не приводит к ускорению работы алгоритма, что связано прежде всего с малым количеством расчетных узлов. Кроме того, использование буфера не только не привело к ускорению алгоритма, но, наоборот, несколько затормозило его. Скорость работы алгоритма с ключом FFTW_TRANSPOSED_ORDER, как и ожидалось, оказалась выше, чем с ключом FFTW_NORMAL_ORDER.

Были проведены замеры времени для лучшего набора опций (отсутствие буфера, опции FFTW_TRANSPOSED_ORDER и FFTW_ESTIMATE). Замеры проводились без сохранения матрицы в файл и с параллельной записью матрицы в файл на каждом десятом шаге. Рассчитанные по полученным данным ускорения и эффективности программ представлены на рис. 11–14. Видно, что для размера матрицы поля, не превосходящего 2048×2048 , использование более 8 процессов нецелесообразно, поскольку не дает прироста скорости. Эффективность реализации резко падает для размеров матрицы поля 512×512 и 1024×1024 . Для больших матриц эффективность остается значительной: около 50% для случая без сохранения данных и около 30% для случая с сохранением.

Для алгоритма, использующего неявную схему, проводились замеры времени работы на суперкомпьютере СКИФ МГУ «Чебышёв» и кластере IBM BlueGene/P ВМиК МГУ.

Результаты замеров времени на СКИФе представлены на рис. 17–20. Большие размеры матрицы и необходимость в дополнительной памяти для метода с использованием неявной схемы не позволили произвести расчет с использованием одного процесса, поэтому нормировка производилась на время работы программы на 4 процессах.

Видно, что при сохранении результатов в ходе работы программы, для матриц размером 512×512 и 1024×1024 при увеличении числа процессов от 8 до 32 ускорение падает.

Эффективность работы программы при сохранении результатов в ходе работы уменьшается, относительно эффективности работы программы не сохраняющей результаты расчетов в файлы. Увеличение числа процессов эффективно при размере матрицы более 1024×1024 , если происходит сохранение результатов.

Результаты замеров времени работы алгоритма с использованием неявной схемы на кластере IBM BlueGene/P приведены на рис. 21–26. Видно стабильное линейное увеличение ускорения работы программы при увеличении числа процессоров уже для сравнительно небольших матриц. Эффективность для больших расчетных сеток превышает 60%, что является хорошим результатом. Таким образом, при работе на кластере IBM BlueGene/P эффективно увеличение числа процессов, на которых запускается программа. Однако, время выполнения шага интегрирования на кластере

IBM Bluegene/P существенно больше, чем на СКИФе, что связано с более слабыми процессорами.

5. Выводы.

В рамках работы было проведено исследование различных алгоритмов решения нелинейного уравнения квазиоптики (нелинейного уравнения Шредингера). Рассматривались методы с использованием разностных схем и метод с применением преобразования Фурье.

Тестовые эксперименты показали, что при использовании схемы предиктор-корректор (Рунге-Кутта 4-го порядка) из-за наличия поперечных координат и участия в уравнении производной по ним необходимо использовать слишком маленький шаг по координате z для получения устойчивого решения. Таким образом, эту схему имеет смысл использовать в расчётах, где величина шага уже лимитирована другими особенностями поставленной задачи. Также следует отметить, что для реализации этого алгоритма необходимо держать в памяти 3 дополнительные матрицы равные по размеру основной, а на каждом шаге необходимо 4 раза обмениваться граничными значениями локальных блоков матрицы.

В случае применения неявной консервативной схемы, устойчивой при любом шаге интегрирования, шаг необходимо выбирать основываясь на эмпирических физических оценках. Кроме того, для реализации этого алгоритма требуется только одна дополнительная матрица, равная основной (она используется для хранения коэффициентов метода прогонки). Алгоритм показал отличную масштабируемость на кластере IBM Bluegene/P, которая не сильно пострадала даже в случае периодического сохранения данных вычислений на диск. Это объясняется небольшой тактовой частотой процессоров при наличии быстрой сети и применении MPI Parallel I/O.

При использовании Фурье-метода можно обойтись без использования дополнительной матрицы, тем самым по сравнению с методом Рунге-Кутта увеличить размер расчётной сетки в 2 раза по каждой координате при использовании того же количества памяти. Из-за более сложной организации пересылок при расчёте параллельного Фурье-преобразования по сравнению с остальными алгоритмами его масштабируемость ниже. Также наблюдается провал в производительности при небольших размерах матриц. Однако скорость вычислений для этого метода больше, по крайней мере при использовании до 64 процессов, что для применения на СКИФ МГУ «Чебышёв» делает его более удачным.

Таким образом, нельзя чётко сказать, какой метод является лучше в общем случае. В случае, если задача не накладывает каких-то особых ограничений, лучше использовать Фурье-метод. Если же имеется возможность использовать для расчёта очень большое число процессоров, то целесообразно применения метода с неявной разностной схемой. Также этот метод будет применим для задач с неравномерной сеткой по поперечному сечению, так как для них нет алгоритма быстрого преобразования Фурье и метод с его использованием теряет свою актуальность.

Коллектив авторов выражает благодарность администрации кластеров СКИФ «Чебышёв» НИВЦ МГУ и IBM Bluegene/P ВМиК МГУ за предоставленное процессорное время для проведения тестирования алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

1. Кандидов В.П., Шлёнов С.А., Косарева О.Г. Филаментация мощного фемтосекундного лазерного излучения. Квантовая электроника, 39, 3, 2009.
2. Ладагин В.К., Стариков Ф.А. Численное решение квазиоптического уравнения для поперечной корреляционной функции поля излучения. Математическое моделирование, 10, 8, 1998.
3. Agrawal G.P. Applications of Nonlinear Fiber Optics (3rd edition). Elsevier Science, 2001.
4. Маханьков В.Г. Солитоны и численный эксперимент. Физика элементарных частиц и атомного ядра, 14, 1, 1983.
5. Витковский В.Э., Федорук М.П. Численное исследование свойств решений нелинейного уравнения Шредингера при распространении лазерных импульсов в световодах. Вычислительные технологии, 13, 6, 2008.
6. Кадомцев Б.Б., Кадомцев М.Б. Конденсаты Бозе-Эйнштейна. УФН, 167, 6, 1997.
7. Беляева Л.Т. и др. Динамика солитонов в модели нелинейного уравнения Шредингера с внешним гармоническим потенциалом. Квантовая Электроника, 35, 9, 2005.
8. Картавенко В.Г. Линейные и нелинейные возбуждения ядерной плотности. Физика элементарных частиц и атомного ядра, 24, 6, 1993.
9. Зейтунян Р.Х. Нелинейные длинные волны на поверхности воды. Успехи Физических Наук, 165, 12, 1995.
10. Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жёсткие и дифференциально-алгебраические задачи. М., Мир, 1999, 685.
11. Howwen P.J., Sommeijer B.P. Parallel ODE solver. // Proceedings of the International Conference on Supercomputing. 2001, 71–81.
12. Jackson K.R., Norsett S.P. The potential for parallelism in Runge-Kutta methods. Part 1: R-K formulas in standard form. SIAM J. Numer. Anal. 32, 2001, 49–82.
13. <http://fftw.org>
14. Matteo Frigo. A fast Fourier transform compiler // Proc. 1999 ACM SIGPLAN Conf. on Programming Language Design and Implementation. 1999. Vol. 34, num. 5, 169–180.
15. Matteo Frigo, Steven G. Johnson. FFTW: An adaptive software architecture for the FFT // Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing. 1998. Vol. 3, 1381–1384.
16. Калиткин Н.Н. Численные методы. М., Наука, 1978.
17. V.P. Kandidov, V.Yu. Fedorov. Properties of self-focusing in elliptic beams. Quantum Electronics, 34, 12, 2004.