МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Физический факультет
Кафедра высшей математики

С. В. Смирнов

ОСНОВЫ ВЫЧИСЛИТЕЛЬНОЙ ФИЗИКИ

Часть II

Учебное пособие

Новосибирск
2017

УДК 519.6
ББК 22.19я73
С50

Рецензент
д-р физ.-мат. наук, чл.-корр. РАН *М. П. Федорук*

Смирнов, С. В.
С50 Основы вычислительной физики : учеб. пособие: в 2 ч. / С. В. Смирнов. – Новосибирск : ИПЦ НГУ, 2017. – Ч. 2. – 104 с.

ISBN 978-5-4437-0676-4
ISBN 978-5-4437-0677-1 (часть 2)

Настоящее учебное пособие соответствует материалу лекций 7–11 по дисциплине «Основы вычислительной физики», читаемых студентам 4-го курса физического факультета НГУ, и содержит рассмотрение ряда базовых вопросов методов вычислений, используемых в физике. Пособие знакомит читателей с численными методами решения задач линейной алгебры, дискретным преобразованием Фурье и численными схемами для интегрирования уравнения теплопроводности и нелинейного уравнения Шрёдингера. Отбор материала и уровень строгости изложения адаптированы для студентов-физиков.

Для студентов 4-го курса физического факультета НГУ, студентов старших курсов и аспирантов физических и технических специальностей вузов.

УДК 519.6
ББК 22.19я73

Пособие подготовлено при частичной финансовой поддержке Минобрнауки РФ (соглашение № 14.B25.31.0003, гос. задание № 3.5572.2017/БЧ).

ISBN 978-5-4437-0676-4
ISBN 978-5-4437-0677-1 (часть 2)

Оглавление

Предисловие

Настоящее учебное пособие соответствует материалу лекций 7–11 по дисциплине «Основы вычислительной физики», читаемых студентам 4-го курса физического факультета НГУ, и содержит рассмотрение ряда базовых вопросов методов вычислений, используемых в физике. Первая глава данного пособия знакомит читателей с численными методами решения задач линейной алгебры, включая решение систем линейных уравнений, вычисление определителя и обратной матрицы, поиск собственных значений и собственных векторов. Во второй главе рассматриваются вопросы, связанные с дискретным преобразованием Фурье: эффекты подмены частот и частокола, алгоритмы быстрого преобразования Фурье и их эффективная программная реализации на языке C — библиотека FFTW. Третья глава посвящена численному решению уравнения теплопроводности, на примере которого рассматриваются вопросы устойчивости явных и неявных численных схем. Четвёртая, последняя, глава знакомит читателей с нелинейным уравнением Шрёдингера, его физической интерпретацией, точными аналитическими решениями, полученными в некоторых частных случаях, и методами численного интегрирования, включая метод Фурье расщепления по физическим процессам и разностные методы в представлении взаимодействия. Отбор материала и уровень строгости изложения адаптированы для студентов-физиков.

Автор выражает глубокую признательность Александру Ивановичу Черных и Максиму Александровичу Никулину за ценный вклад в отбор материала курса лекций, целый ряд исключительно полезных советов, идей и критических замечаний, позволивших существенно улучшить текст пособия.

1. Задачи линейной алгебры

Что общего между моделированием аэродинамических потоков, расчётом опор моста, поиском мод волоконного фотонного кристалла и определением уровней энергии квантовой частицы в потенциальной яме? Эффективные алгоритмы решения всех перечисленных и многих других физических задач основаны на сведении их к задачам линейной алгебры: решению систем линейных уравнений, вычислению определителей матриц, нахождению обратных матриц и поиску собственных чисел и собственных векторов матрицы. Поскольку ЭВМ может оперировать лишь с конечным набором дискретных значений, матрицы и векторы возникают в численном моделировании естественным обра-

зом как конечномерные аналоги операторов в Гильбертовых пространствах. Так, например, для решения уравнений в частных производных непрерывные функции заменяются их дискретными сеточными аналогами, а действующие на них дифференциальные операторы — разностными выражениями [1, с. 78], связывающими значения функции в нескольких соседних узлах сетки. Таким образом, дискретизация задач с операторами в частных производных обычно приводит к системе линейных уравнений[1], количество которых пропорционально числу узлов сетки и, следовательно, может быть достаточно большим числом. В данной главе мы познакомимся с некоторыми наиболее простыми методами решения основных задач линейной алгебры.

1.1. Метод исключения Гаусса

Пусть необходимо решить систему линейных алгебраических уравнений:

$$\sum_{j=1}^{n} A_{ij} x_j = b_i, \tag{1}$$

где A — матрица $n \times n$, $\mathbf{x} = (x_1, \ldots, x_n)^T$ — искомый вектор, $\mathbf{b} = (b_1, \ldots, b_n)^T$ — известный вектор правых частей уравнений. Из курса алгебры известно, что в случае невырожденной матрицы A система (1) имеет единственное решение, которое может быть записано по формуле Крамера:

$$x_i = \frac{\Delta_i}{\Delta}, \tag{2}$$

где $\Delta = \det A$ — определитель матрицы системы, Δ_i — определитель матрицы A, в которой i-й столбец был заменен на вектор $\mathbf{b}$. Если для нахождения x_i по формуле (2) мы будем вычислять определители с помощью разложения по строке (столбцу), то для нахождения всех компонент вектора $\mathbf{x}$ нам потребуется совершить $\mathcal{O}(2^n(n+1)!)$ арифметических операций. Предельное расчётное быстродействие самого мощного на сегодняшний день суперкомпьютера[2] позволит вычислить за два года работы определитель матрицы размером всего лишь $n = 19$. Быстрый (экспоненциальный) рост числа операций при увеличении размера матрицы n делает метод Крамера непригодным для вычислений определителей матриц даже небольшого размера. Как мы вскоре увидим,

[1] Нелинейные системы могут быть решены с использованием итераций, при этом на каждом шаге необходимо решать систему линейных уравнений.

[2] $1{,}25 \times 10^{17}$ операций в секунду согласно рейтингу www.top500.org.

эффективная стратегия состоит в использовании метода решения системы линейных уравнений для вычисления определителя, а не наоборот, как предполагает формула Крамера (2).

Для начала рассмотрим наиболее простую модификацию метода (как будет показано в следующем параграфе, она может давать совершенно неудовлетворительные результаты даже для хорошо обусловленных систем). В данном рассмотрении мы будем использовать для коэффициентов системы, изменённых в процессе решения, те же обозначения A_{ij}, b_i, что и для их первоначальных (входных) значений. Жертвуя при этом математической строгостью обозначений, взамен мы получаем более тесную связь формул решения с компьютерной программой, в которой содержимое массивов (матриц) изменяется в процессе решения, в то время как имена массивов остаются прежними.

Решение системы уравнений методом исключения Гаусса происходит в два этапа, которые называют *прямым* и *обратным ходом* метода. *Прямой ход* заключается в приведении матрицы системы к верхнетреугольному виду (так, чтобы все элементы ниже главной диагонали были равны нулю: $A_{ij} = 0 \quad \forall i > j$). Зануление производится последовательно по столбцам: $k = 1, 2, \ldots, n-1$, а в пределах каждого столбца — по строкам: $i = k+1, k+2, \ldots, n$. Для зануления коэффициента A_{ik} нужно вычислить отношение $c = A_{ik}/A_{kk}$, после чего вычесть из i-го уравнения системы k-е, умноженное на c:

$$\begin{aligned} c &\leftarrow A_{ik}/A_{kk}; \\ A_{ik} &\leftarrow 0; \\ A_{ij} &\leftarrow a_{ij} - c \cdot A_{kj}, \quad j = k+1, \ldots, n; \\ b_i &\leftarrow b_i - c \cdot b_k. \end{aligned} \tag{3}$$

Здесь и далее символ $\leftarrow$ обозначает оператор присваивания, т. е. запись $x \leftarrow y$ подразумевает, что значение переменной y записывается в переменную (ячейку памяти) x. Повторение процесса (3) в цикле для строк $i = k+1, \ldots, n$ приведёт к занулению всех элементов k-го столбца матрицы ниже главной диагонали. Выполняя описанные выше операции последовательно для столбцов $k = 1, 2, \ldots, n-1$, мы преобразуем матрицу системы к верхнетреугольному виду:

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1,n-1} & A_{1n} \\ 0 & A_{22} & \dots & A_{2,n-1} & A_{2n} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & A_{n-1,n-1} & A_{n-1,n} \\ 0 & 0 & \dots & 0 & A_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix}. \quad (4)$$

После этого решение системы $\mathbf{x}$ может быть легко найдено с помощью *обратного хода* метода исключения Гаусса: вначале вычисляется последняя компонента $x_n = b_n/A_{nn}$, затем найденное значение подставляется в предпоследнее уравнение: $x_{n-1} = (b_{n-1} - A_{n-1,n}x_n)/A_{n-1,n-1}$ и т. д. Общий вид формулы обратного хода:

$$x_i = \frac{1}{A_{ii}} \left(b_i - \sum_{j=i+1}^{n} A_{ij} x_j \right), \quad i = n-1,\, n-2,\, \dots, 1. \quad (5)$$

Подсчитаем число операций, необходимых для решения системы методом исключения Гаусса. Вычитание k-го уравнения из i-го в соответствии с (3) требует $(n-k+1)$ операций умножения, столько же вычитаний и одно деление — для простоты будем считать их вместе как $2(n-k+1)$ операцию. Общее число операций прямого хода при этом есть

$$n_1 = \sum_{k=1}^{n-1} \sum_{i=k+1}^{n} 2(n-k+1) = 2 \sum_{k=1}^{n-1} (n-k)(n-k+1) \approx \frac{2}{3} n^3. \quad (6)$$

(Данная асимптотика может быть получена переходом к суммированию по $k' = n-k$ и последующей заменой дискретной суммой на интеграл.) Число операций обратного хода (5) есть

$$n_2 = \sum_{i=1}^{n} (1 + 2(n-i)) = \mathcal{O}(n^2). \quad (7)$$

Поскольку $n_1 = \mathcal{O}(n^3)$, числом операций обратного хода можно пренебречь, так что полное число операций $n_\Sigma = n_1 + n_2 \approx n_1 \approx \frac{2}{3} n^3$.

1.2. Метод исключения Гаусса с выбором главного элемента

Напишем программу в соответствии с полученными выше формулами (3), (5) и попробуем решить следующую систему уравнений:

$$\begin{pmatrix} 5 & 0{,}2 & 1 \\ 1 & 0{,}04 & 1 \\ -5 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -10 \\ -10 \\ 20 \end{pmatrix}. \tag{8}$$

Полученные в результате корни $x_1 = x_2 = 0$, $x_3 = -10$ при подстановке в систему (8) не дадут равенства правой и левой частей. Разберёмся, почему так происходит.

На первом шаге прямого хода метода исключения Гаусса первое уравнение (8) должно быть умножено на $c = A_{21}/A_{11} = 1/5 = 0{,}2$ и вычтено из второго уравнения. Выполняя эти вычисления в уме, мы получим во второй строке матрицы (0; 0; 0,8). Обратим внимание на зануление диагонального элемента $A_{22} = 0$: на следующем шаге метода исключения (3) это должно привести к делению на 0 с аварийным остановом программы или появлению в расчётах нечисловых значений `inf` и `nan`.

Однако, поскольку входящие в матрицу (8) десятичные числа 0,2 и 0,04 непредставимы в виде конечных двоичных дробей, при вычислении на ЭВМ возникает погрешность округления, в результате чего вместо $A_{22} = 0$ мы получим хотя и очень малое, но отличное от нуля значение $A_{22} = -6{,}94 \cdot 10^{-18}$:

$$\Lambda^{(1)} = \left(\begin{array}{ccc|c} 5 & 0{,}2 & 1 & -10 \\ 0 & -6{,}94 \cdot 10^{-18} & 0{,}8 & -8 \\ 0 & 1{,}2 & 2 & 10 \end{array}\right). \tag{9}$$

(Здесь $\Lambda^{(1)}$ — расширенная матрица 3×4 системы уравнений после зануления столбца $k = 1$.) Поскольку $A_{22} \neq 0$, в дальнейших вычислениях не возникнет деления на ноль — мы не получим ни аварийного останова программы, ни значений `inf` и/или `nan`, которые могли бы сигнализировать о возникновении нештатной ситуации в расчётах.

Для зануления элемента A_{32} в столбце $k = 2$ в соответствии с (3) необходимо вычесть из третьей строки (9) вторую, умноженную на $c = 1{,}2/(-6{,}94 \cdot 10^{-18}) = -1{,}73 \cdot 10^{+17}$. Чтобы понять, чему будет равен результат машинной операции

$$A_{32} \leftarrow 2 - 0{,}8 \cdot (-1{,}73 \cdot 10^{+17}) = 2 + 1{,}38 \cdot 10^{+17}, \tag{10}$$

вспомним, что компьютер оперирует с числами с плавающей точкой (запятой), имея в распоряжении 52 двоичных разряда [1, п. 2.2]. Единица в младшем разряде (ULP, или машинное эпсилон) равно $\varepsilon = 0{,}00\ldots01_2 = 2^{-52} \approx 2{,}22 \cdot 10^{-16}$ и характеризует *относительную* точность вычислений. Поскольку операнды в (10) отличаются более чем в ε раз, можно ожидать, что $A_{32} \leftarrow 2 + 1{,}38 \cdot 10^{+17} = 1{,}38 \cdot 10^{+17}$. Подчеркнём, что результат машинного сложения в (10) *в точности* (*побитово* в машинном представлении) равен второму слагаемому!

Произошедшее является примером *катастрофической потери точности*: несмотря на то, что результат операции (10) имеет все 52 верных двоичных разряда, при вычитании строки 2 матрицы (9) из строки 3 ($A_{3j} \leftarrow A_{3j} - c\,A_{2j}$) информация о третьей строке была полностью утрачена, оказавшись за пределами точности вычислений:

$$\Lambda^{(2)} = \left(\begin{array}{ccc|c} 5 & 0{,}2 & 1 & -10 \\ 0 & -6{,}94 \cdot 10^{-18} & 0{,}8 & -8 \\ 0 & 0 & 1{,}38 \cdot 10^{+17} & -1{,}38 \cdot 10^{+18} \end{array}\right). \qquad (11)$$

Как следствие, строки 2 и 3 матрицы (11) оказались почти пропорциональны друг другу. Используя формулы обратного хода (5), несложно получить из (11) ответ $\mathbf{x} = (0;\, 0;\, -10)$, который, очевидно, не удовлетворяет третьему уравнению исходной системы (8).

Заметим, что, хотя коэффициенты в (9) и даже в (8) уже содержат погрешность округления, это оказывает лишь незначительное влияние на ответ: прямым вычислением можно убедиться, что относительная погрешность вычисления корней (8) и (9) имеет тот же порядок величины, что и погрешность округления ε. Причиной катастрофической потери точности является вычитание двух строк матрицы с домножением на большой коэффициент c (3), что, в свою очередь, связано с возникновением в процессе решения близкого к нулю диагонального элемента матрицы системы.

Чтобы избежать рассмотренной выше потери точности, достаточно лишь немного модифицировать метод исключения Гаусса, дополнив его *выбором главного элемента* на каждом шаге k прямого хода. А именно, перед применением формулы (3) при фиксированном k и всех $i = k+1, \ldots, n$ необходимо вначале найти строку m с максимальным по модулю элементом A_{mk} и поменять её местами с k-й строкой расширенной матрицы системы. В программной реализации метода более простой и эффективной альтернативой перестановке строк с копированием данных может служить массив перестановок индексов, через который будет вестись обращение к массиву матричных элементов: например,

вместо `A[i][j]-=c*A[k][j]` формула (3) может быть запрограммирована как `A[p[i]][j]-=c*A[p[k]][j]`, где `A` — расширенная матрица системы, `int p[n]` — массив перестановок индексов. Массив p[n] вначале должен быть инициализирован единичной перестановкой `p[i]=i`, а перед каждым шагом прямого хода необходимо менять местами значения в ячейках `p[k]` $\leftrightarrow$ `p[m]`. Видно, что выбор главного элемента лишь немного усложняет код программы, но существенно повышает её надёжность, позволяя предотвратить потерю точности.

В заключение данного параграфа ещё раз обратим внимание на то, как важно помнить о специфике машинных вычислений и критически относиться к результатам, выдаваемым даже написанной «без ошибок» компьютерной программой.

1.3. Погрешность и невязка

Говоря о любом численном методе, всегда следует иметь в виду два вопроса: его эффективность (количество операций для получения ответа) и точность, которую он обеспечивает. Первый аспект уже был рассмотрен нами выше, теперь поговорим о погрешности численного решения систем линейных уравнений.

Обычно используют две меры погрешности. Первая, наиболее естественная и представляющая наибольший интерес, — *вектор ошибки*, определяемый как разность двух решений: точного ($\mathbf{x}$) и полученного в численных расчётах ($\mathbf{x}_*$):

$$\delta = \mathbf{x} - \mathbf{x}_*. \tag{12}$$

Вторая — *невязка*, полученная при подстановке численного решения в исходную систему уравнений:

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}_* = A(\mathbf{x} - \mathbf{x}_*) = A \cdot \delta. \tag{13}$$

Вектор невязки (13) может быть легко вычислен непосредственно после получения решения системы. Важным достоинством метода исключения Гаусса с выбором главного элемента является гарантированно малая величина невязки: $|\mathbf{r}| \approx |A| \times |\mathbf{x}| \times \varepsilon$, т. е. компоненты вектора невязки (13) по порядку величины будут равны произведению матричных элементов A, компонент вектора решений $\mathbf{x}$ и машинного ε.

Однако малость вектора невязки $\mathbf{r}$ (13) ещё не означает малости погрешности δ (12) численного решения. Действительно, как следует из (13), $\delta = A^{-1}\mathbf{r}$, т. е. в случае, если матрица A является «почти вырожденной», численное решение может иметь большую погрешность $|\delta|$ даже при относительно малой величине невязки $|\mathbf{r}|$.

Какую матрицу следует считать «почти вырожденной»? Первое, что приходит в голову — это малость определителя $\det A$. Однако обратим внимание, что если выполнить масштабное преобразование (1), разделив каждый элемент матрицы A на одну и ту же константу $\alpha \gg 1:\quad A_{ij} \leftarrow \alpha^{-1} A_{ij}$, определитель матрицы A уменьшится в большое число α^n раз, при этом вряд ли можно ожидать сколько-нибудь заметного изменения корней $\mathbf{x}_*$ и относительной погрешности их вычисления[3]. В самом деле, ЭВМ оперирует числами с плавающей точкой, сохраняя отдельно порядок и мантиссу каждого числа. Умножение всех чисел в расчётах на одну и ту же константу $\alpha^{-1} \ll 1$ приведёт лишь к уменьшению порядка всех чисел, что никак не отразится на арифметических операциях над матричными элементами A_{ij}. Таким образом, понятие «почти вырожденной» матрицы не сводится к малости определителя $\det A$.

Чтобы лучше понять суть дела, вспомним, что при изучении численных методов интегрирования обыкновенных дифференциальных уравнений [1, с. 89] мы уже встречались с примерами *плохо обусловленных* задач, небольшое изменение условий которых приводит к значительному изменению ответа. Наличие малой ошибки $(\varepsilon/2)$ во входных параметрах и погрешностей округления при выполнении промежуточных вычислений приведут к погрешности численного решения δ. При одинаковой относительной погрешности входных условий $\varepsilon/2$ погрешность ответа δ может существенно отличаться для *хорошо* и *плохо обусловленных* задач.

Аналогичное справедливо и в отношении численного решения систем линейных уравнений. Абсолютную погрешность (12) решения можно оценить как

$$|\delta| \approx |\mathbf{x}_*| \cdot \operatorname{cond}(A) \cdot \varepsilon, \tag{14}$$

где $\operatorname{cond}(A)$ — *число обусловленности* матрицы A:

$$\operatorname{cond}(A) = \left(\max_{\mathbf{x}} \frac{|A\mathbf{x}|}{|\mathbf{x}|} \right) \Big/ \left(\min_{\mathbf{x}} \frac{|A\mathbf{x}|}{|\mathbf{x}|} \right). \tag{15}$$

Другими словами, число обусловленности (15) есть отношение максимального и минимального числа раз, в которое матрица A изменяет норму произвольного вектора $\mathbf{x}$. Как следует из определения, $\operatorname{cond}(A) \geq 1$ для любой невырожденной матрицы A; в случае, если матрица A вырожденная, знаменатель (15) обращается в ноль.

[3] Здесь предполагается, что произведения $\alpha^{-1} A_{ij}$ не настолько малы, чтобы происходила потеря порядка.

Для эрмитовых матриц выражение (15) может быть записано в виде $\mathrm{cond}(A) = \max|\lambda_j| / \min|\lambda_j|$, где λ_j — собственные числа.

В полном соответствии с рассмотренным выше примером, из формул (14) и (15) видно, что погрешность численного решения системы оказывается нечувствительной к масштабным преобразованиям $A_{ij} \leftarrow \alpha^{-1} A_{ij}$. Также из (15) следует, что повышенная погрешность численного решения системы связана с наличием в матрице одновременно больших и малых масштабов. Сложение чисел различных порядков величины приводит к отбрасыванию части значащих разрядов меньшего по модулю слагаемого (вплоть до катастрофической потери точности, рассмотренной в п. 1.2). Поскольку количество разрядов, отбрасываемых при сложении $a + b$, приблизительно равно модулю разности порядков a и b, т. е. $|\log|a/b||$, можно ожидать, что логарифм числа обусловленности (15) будет приближённо равен числу теряемых значащих цифр при решении системы (1).

1.4. Определитель и обратная матрица

Рассмотренный выше метод исключения Гаусса не только позволяет решать системы линейных уравнений, но и является основой для эффективных способов вычисления определителя и обратной матрицы. Обратим внимание, что преобразование (3) квадратной матрицы A_{ij} оставляет неизменным её определитель. Полученная в результате прямого хода метода исключения матрица имеет верхнетреугольный вид (4), и её определитель равен произведению диагональных элементов. (В этом несложно убедиться, вычисляя определитель матрицы (4) разложением по первому столбцу.) Число операций, необходимых для вычисления определителя рассмотренным способом, приблизительно равно $\frac{2}{3}n^3 + n \approx \frac{2}{3}n^3$. При вычислении определителей матриц большого размера и в случае, если матричные коэффициенты сильно отличаются по модулю от 1, следует иметь в виду возможность переполнения и потери порядка.

Рассмотрим задачу о нахождении обратной матрицы A^{-1}, полагая A, как и раньше, невырожденной квадратной матрицей $n \times n$. Аналогично формуле Крамера (2) для решения линейных систем и нахождению определителя матрицы через разложение по строке (столбцу), хорошо известное из алгебры выражение для обратной матрицы

$$A^{-1} = \frac{\mathrm{adj}(A)}{\det A}$$

являет собой ещё один пример неэффективного способа вычисления

(здесь $\mathrm{adj}(A)$ — присоединённая матрица, составленная из алгебраических дополнений). Значительно более быстрый и притом относительно несложный в реализации способ состоит в следующем. По определению обратной матрицы

$$A_{ik}(A^{-1})_{kj} = \delta_{ij}, \tag{16}$$

где δ_{ij} — символ Кронекера; выражение (16) при фиксированном j можно рассматривать как систему линейных алгебраических уравнений на n неизвестных — матричные элементы j-го столбца матрицы A^{-1}. Несмотря на то, что формально для нахождения всех матричных коэффициентов A^{-1} требуется решить n линейных систем (16) — по одной для каждого значения $j = 1, \ldots, n$, — необходимое для этого число арифметических операций при правильной организации вычислений возрастёт лишь приблизительно в 2,5 раза, оставаясь равным $\mathcal{O}(n^3)$. Действительно, заметим, что преобразование (3) матрицы A во время прямого хода метода исключения Гаусса никак не зависит от вектора $\mathbf{b}$ правых частей системы. Поскольку системы уравнений (16) при $j = 1, \ldots, n$ имеют общую квадратную матрицу A, отличаясь лишь вектором правых частей уравнений $\mathbf{b_1}, \ldots, \mathbf{b_n}$, можно обобщить выражение (3) для эффективного решения n систем (16):

$$\begin{aligned} c &\leftarrow A_{ik}/A_{kk}; \\ A_{ik} &\leftarrow 0; \\ A_{ij} &\leftarrow a_{ij} - c \cdot A_{kj}, \quad j = k+1, \ldots, n; \\ B_{ij} &\leftarrow b_{ij} - c \cdot B_{kj}, \quad j = k+1, \ldots, n, \end{aligned} \tag{17}$$

где B_{ij} — квадратная матрица, составленная из векторов в правой части систем (16) при $j = 1, \ldots, n$; её начальное значение $B_{ij} = \delta_{ij}$.

1.5. Метод прогонки для трёхдиагональных матриц

В предыдущих пунктах мы рассматривали общий случай системы линейных уравнений с n неизвестными. Вместе с тем в физике есть множество частных случаев, в которых решение систем уравнений специального вида может быть выполнено значительно эффективнее. С одним из примеров таких систем мы уже знакомились при рассмотрении задачи полиномиальной интерполяции [1, с. 72]. Для определения коэффициентов полинома $P(x) = a_0 + a_1 x^1 + \ldots + a_n x^n$ с помощью решения системы алгебраических уравнений требуется $\mathcal{O}(n^3)$ арифметических операций, тогда как при использовании интерполяционных полиномов Лагранжа или Ньютона аналогичный «подготовительный»

этап вычисления коэффициентов может быть выполнен всего за $\mathcal{O}(n^2)$ операций.

Другим чрезвычайно важным примером являются системы уравнений с *разреженными* матрицами, т. е. матрицами, большинство элементов которых равны нулю. В этом пункте мы познакомимся с хрестоматийным методом прогонки для трёхдиагональных матриц — по сути, с частным случаем рассмотренного выше метода исключения Гаусса. Трёхдиагональные матрицы уже встречались нам при обсуждении интерполяционных кубических сплайнов [1, с. 83]. Кроме того, ленточные матрицы регулярно возникают при решении дифференциальных уравнений в частных производных с использованием конечно-разностных и конечно-элементных методов, знакомство с которыми нам ещё предстоит. Размер таких матриц обычно равен числу узлов сетки (а в некоторых случаях может и превосходить его в несколько раз) и, следовательно, может быть достаточно большим — вплоть до десятков и сотен тысяч. Использование специальной (разреженной) структуры таких матриц позволяет повысить эффективность расчётов на много порядков величины.

Запишем систему уравнений с трёхдиагональной матрицей в виде

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i, \quad i = 1, \ldots, n, \quad a_1 = c_n = 0. \tag{18}$$

Прямой ход метода исключения Гаусса (3) для системы (18) сводится к исключению поддиагональных элементов a_i, $\quad i = 2, \ldots, n$:

$$\xi \leftarrow \frac{a_i}{b_{i-1}}, \quad a_i \leftarrow 0, \quad b_i \leftarrow b_i - \xi c_{i-1}, \quad d_i \leftarrow d_i - \xi d_{i-1}. \tag{19}$$

Сумма в формуле для обратного хода (5) в случае трёхдиагональной матрицы будет содержать всего один член:

$$x_n = \frac{d_n}{b_n}; \quad x_i = \frac{1}{b_i}\big(d_i - c_i x_{i+1}\big), \quad i = n-1, \ldots, 1. \tag{20}$$

Решение системы методом прогонки (19), (20) требует всего $4n$ ячеек памяти для хранения матричных коэффициентов (18) и $8n$ арифметических операций, что значительно экономичнее метода исключения Гаусса (3), (5) для матриц общего вида.

Можно показать [2, гл. 3], что достаточным условием устойчивости метода прогонки может служить *преобладание диагональных элементов*: $|b_i| \geq |a_i| + |c_i|$ со строгим неравенством хотя бы для одного i. Подчеркнём, что данное условие является достаточным, но не необходимым, и в физических расчётах прогонка в большинстве случаев оказывается достаточно устойчивой даже при нарушении преобладания диагональных элементов.

1.6. Модификация метода прогонки для периодических граничных условий

Рассмотренный в предыдущем пункте метод прогонки будет применён нами в п. 3.3 для решения уравнения теплопроводности $u_t - u_{xx} = f(x,t)$ с использованием неявных схем. Заменяя частную производную u_{xx} на конечную разность $\frac{1}{h^2}(u_{j+1} - 2u_j + u_{j-1})$, мы получим систему уравнений, каждое из которых связывает значения температуры в трёх соседних узлах. Использование нулевых условий на границе $u(0,t) = u(L,t) = 0$ обеспечит зануление двух матричных коэффициентов в первом и последнем уравнении, так что мы получим систему с трёхдиагональной матрицей (18). Что делать, если нам необходимо решить уравнение теплопроводности на цилиндре, используя периодические граничные условия $u(0,t) = u(2\pi,t)$, $u_x(0,t) = u_x(2\pi,t)$? Очевидно, что при этом каждое уравнение, в том числе самое первое и последнее, содержит по три члена, ни один из которых не равен нулю, так что матрица системы уже не является трёхдиагональной (ни даже ленточной). Рассмотрим кратко один из экономичных способов решения таких систем, предложенный в работе [A1]. Запишем исходную систему линейных уравнений в виде

$$A\mathbf{x} = \mathbf{d}, \tag{21}$$

где вектор $\mathbf{x}$ — искомое решение, $\mathbf{d}$ — известный вектор правых частей, A — матрица системы (21) порядка n:

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 & a_1 \\ a_2 & b_2 & c_2 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & 0 & 0 & \dots & 0 & a_n & b_n \end{pmatrix}. \tag{22}$$

Обратим внимание, что первые $n-1$ строк и столбцов матрицы A образуют трёхдиагональную подматрицу — обозначим её A'. Перепишем исходную систему (21) через A', что позволит нам решать возникающие уравнения методом прогонки:

$$\begin{pmatrix} A' & \mathbf{u}' \\ \mathbf{v}'^{\mathrm{T}} & b_n \end{pmatrix} \begin{pmatrix} \mathbf{x}' \\ x_n \end{pmatrix} = \begin{pmatrix} \mathbf{d}' \\ d_n \end{pmatrix}, \tag{23}$$

где ^{T} обозначает транспонирование, штрихованными буквами обозначены векторы из $(n-1)$-мерного подпространства. В частности, $\mathbf{u}', \mathbf{v}'$ —

векторы, состоящие из первых $(n-1)$ элементов n-го столбца и n-й строки A (22):

$$\begin{array}{rcl} \mathbf{u}' & = & (a_1 \quad 0 \quad \ldots \quad 0 \quad c_{n-1})^{\mathrm{T}}, \\ \mathbf{v}' & = & (c_n \quad 0 \quad \ldots \quad 0 \quad a_n)^{\mathrm{T}}, \\ \mathbf{d}' & = & (d_1 \quad d_2 \quad \ldots \quad d_{n-1})^{\mathrm{T}}. \end{array} \tag{24}$$

Выражая $\mathbf{x}'$ из первого (векторного) уравнения системы (23):

$$\mathbf{x}' = (A')^{-1}(\mathbf{d}' - x_n\mathbf{u}') \equiv \mathbf{p}' + x_n\mathbf{q}'. \tag{25}$$

Здесь $(n-1)$-мерные векторы $\mathbf{p}'$ и $\mathbf{q}'$ удовлетворяют системам уравнений с трёхдиагональной матрицей A':

$$A'\mathbf{p}' = \mathbf{d}', \qquad A'\mathbf{q}' = -\mathbf{u}'. \tag{26}$$

Выразим x_n из второго (скалярного) уравнения системы (23):

$$x_n = \frac{1}{b_n}\big(d_n - (\mathbf{v}', \mathbf{x}')\big). \tag{27}$$

Подставляя $\mathbf{x}'$ из (25) в (27) и раскрывая скалярное произведение, получаем окончательно для x_n:

$$x_n = \frac{d_n - (\mathbf{v}', \mathbf{p}')}{b_n + (\mathbf{v}', \mathbf{q}')} = \frac{d_n - c_n p'_1 - a_n p'_{n-1}}{b_n + c_n q'_1 + a_n q'_{n-1}}. \tag{28}$$

Таким образом, для решения задачи (21) необходимо выполнить следующие шаги:

1) найти $\mathbf{p}', \mathbf{q}'$ (26) методом прогонки (19), (20);

2) вычислить x_n по формуле (28);

3) вычислить $\mathbf{x}' = \mathbf{p}' + x_n\mathbf{q}'$ (25).

Вычисления по указанному алгоритму потребуют порядка $14n$ операций, что почти вдвое больше, чем для «обычного» метода прогонки (19), (20) для трёхдиагональных матриц. Действительно, на шаге 1 дважды используется метод прогонки, однако ввиду того, что обе системы (26) имеют одну и ту же матрицу A', их одновременное решение возможно за $12n$ операций. Ещё $2n$ операций требуется на шаге 3.

1.7. Спектральная задача

Расчёт резонаторов и волноводов в оптике, определение резонансных частот для классических механических систем, поиск уровней энергии квантовых частиц и ряд других численных задач сводятся к поиску собственных значений и собственных векторов матриц:

$$A\mathbf{u} = \lambda\mathbf{u}, \tag{29}$$

где λ и $\mathbf{u}$ — собственное значение и собственный вектор матрицы A порядка n. Из курса алгебры известно, что необходимым и достаточным условием существования нетривиального решения $\mathbf{u}$ (29) служит равенство нулю определителя

$$0 = \det|A - \lambda E| \equiv P_n(\lambda), \tag{30}$$

где E — единичная матрица. Выражение в правой части (30) есть многочлен степени n от λ, называемый *характеристическим полиномом*.

Как решать характеристическое уравнение (30)? Даже вычисление коэффициентов характеристического полинома с использованием хорошо известных формул для определителя требует $\mathcal{O}(n!)$ арифметических операций, что делает данный метод совершенно непригодным при $n \geq 15$ (и крайне медленным при меньших n). Вместе с тем, как уже отмечалось ранее, в физических расчётах возникают матрицы порядка $n > 10^3$ или даже 10^4. Ниже мы познакомимся с идеей нескольких наиболее простых методов решения спектральной задачи и свойственными им ограничениями.

1.7.1. Метод интерполяции

Простейшее эффективное решение задачи (29) известно как *метод интерполяции*. Выберем произвольным образом $n+1$ точку $x_0, \ldots, x_n$ и вычислим в них характеристический полином: $y_j \equiv P(x_j), \quad j = 0, \ldots, n$. Затем построим интерполяционный полином $\tilde{P}_n$ степени n, проходящий через заданные точки $(x_0, y_0), \ldots, (x_n, y_n)$ [1, гл. 5]. Поскольку через $n+1$ точку можно провести один и только один полином степени n, имеем $P_n(x) \equiv \tilde{P}_n(x) \quad \forall x$. Следовательно, описанный выше алгоритм позволяет вычислять коэффициенты полинома $\tilde{P}_n$, отличающегося от P_n только погрешностью вычислений. Зная коэффициенты полинома P_n, можно достаточно эффективно вычислить корни этого полинома — например, используя метод парабол [2, с. 146].

Оценим количество арифметических операций, необходимых для использования метода интерполяции. Поскольку вычисление $P_n(x_j)$

есть нахождение определителя матрицы $A - x_j E$ с известными коэффициентами, оно требует $\approx \frac{2}{3}n^3$ операций (см. п. 1.4 на с. 13). Вычисление значений полинома в $n+1$ точке потребует $\approx \frac{2}{3}n^4$ операций. Построение интерполяционного полинома требует всего $\mathcal{O}(n^2)$ операций; считая, что метод парабол сходится за 10 итераций [2, с. 165], вычислительной сложностью поиска корней полинома также можно пренебречь при $n > 20$.

Поскольку метод интерполяции основан на вычислении конечных разностей [1, п. 5.2], его применимость ограничена матрицами сравнительно небольшого порядка ($n \leq 10 \ldots 20$). Кроме того, к числу недостатков данного метода можно отнести относительно высокую сложность его программирования, поскольку метод включает в себя несколько подзадач: вычисление определителей, построение интерполяционного полинома, поиск корней.

1.7.2. Степенной метод

Зачастую интерес представляют не все решения (29), но лишь одно или несколько наибольших (или, наоборот, наименьших) собственных значений. Например, в телекоммуникациях и нелинейной оптике наибольший интерес представляет основная или несколько первых мод оптических волокон, поскольку высоковозбуждённые моды имеют, как правило, высокие потери. Аналогично, в задачах квантовой механики зачастую бывает необходимо найти лишь основное и первые возбуждённые квантовые состояния. Кроме того, при дискретизации спектральной задачи $\hat{H}\psi(x) = E\psi(x)$ и переходе к её конечномерному (матричному) аналогу (29) погрешность ответа быстро возрастает с числом нулей волновой функции $\psi(x)$. Вследствие этого лишь относительно небольшая часть численных решений — основное и несколько первых возбуждённых состояний — имеют отношение к исходной задаче и несут физический смысл. В таких случаях вместо полного решения проблемы собственных значений оказывается выгоднее использовать тот или иной итерационный процесс, сходящийся к одному собственному значению и одному собственному вектору. Один из наиболее простых примеров таких итерационных процессов, позволяющих найти решение *частичной проблемы собственных значений*, известен как *степенной метод* или *счёт на установление*.

Перенумеруем собственные числа симметричной матрицы A в порядке убывания модуля:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|. \tag{31}$$

В качестве начального приближения для итерационного процесса выберем произвольным образом вектор

$$\mathbf{u}^{(0)} = a_1\mathbf{u_1} + a_2\mathbf{u_2} + \dots, \tag{32}$$

где $\mathbf{u_j}$ — j-й собственный вектор матрицы: $A\mathbf{u_j} = \lambda_j\mathbf{u_j}$. На каждой итерации будем умножать вектор на матрицу:

$$\mathbf{u}^{(k)} = A\mathbf{u}^{(k-1)}. \tag{33}$$

Покажем, что итерационный процесс (33) сходится к собственному вектору $\mathbf{u_1}$, соответствующему наибольшему по модулю собственному значению λ_1. Используя начальные условия (32), на k-й итерации процесса (33) будем иметь:

$$\mathbf{u}^{(k)} = \sum_{j=1}^{n} a_j\lambda_j^k\mathbf{u_j} = \lambda_1^k\left(a_1\mathbf{u_1} + \sum_{j=2}^{n} a_j\left(\frac{\lambda_j}{\lambda_1}\right)^k\mathbf{u_j}\right). \tag{34}$$

Здесь мы вынесли λ_1^k, с тем чтобы коэффициент при $\mathbf{u_1}$ в разложении $\mathbf{u}^{(k)}$ был равен a_1 — так же, как и в начальных условиях (32). При этом видно, что коэффициенты при $\mathbf{u_j}$ $(j = 2, \dots, n)$ содержат малые величины вида $(\lambda_j/\lambda_1)^k$, которые стремятся к нулю при $k \to \infty$ в силу неравенств (31). Следовательно, в пределе $k \to \infty$ направление вектора $\mathbf{u}^{(k)}$ будет совпадать с направлением искомого вектора $\mathbf{u_1}$. Обрывая процесс (33) на k-м шаге, получаем в результате искомый вектор $\mathbf{u_1}$ с погрешностью $\mathcal{O}((\lambda_2/\lambda_1)^k)$. Из (33) видно, что для вычисления λ_1 достаточно найти отношение произвольной компоненты вектора $\mathbf{u}$ на двух последовательных итерациях:

$$\lambda_{1;\,j}^{(k)} = u_j^{(k)}/u_j^{(k-1)} + \mathcal{O}((\lambda_2/\lambda_1)^k). \tag{35}$$

Итерационный процесс (33) можно останавливать, когда несколько значений λ_1, вычисленных по формуле (35) с использованием разных компонент j вектора $\mathbf{u}^{(k)}$, совпадают в требуемом числе знаков.

Используя в (35) норму векторов вместо фиксированной (j-й) компоненты $\mathbf{u^{(k)}}$, можно вдвое ускорить сходимость метода:

$$\lambda_1^{(k)} = \sqrt{\frac{(\mathbf{u^{(k)}}, \mathbf{u^{(k)}})}{(\mathbf{u^{(k-1)}}, \mathbf{u^{(k-1)}})}} + \mathcal{O}((\lambda_2/\lambda_1)^{2k}). \tag{36}$$

Заметим, что если начальное приближение (32) было выбрано так, что $a_1 \approx 0$, то, несмотря на малость множителей $(\lambda_j/\lambda_1)^k$, основной

вклад в сумму $\sum a_j \lambda_j^k \mathbf{u_j}$ (34) при недостаточно больших значениях k может давать не первое, а второе или последующие слагаемые[4]. С одной стороны, данное обстоятельство может приводить к ошибочному результату; для предотвращения ошибки следует выбирать вектор $\mathbf{u}^{(0)}$ несимметричным, либо его симметрия должна соответствовать симметрии искомого ответа[5]. С другой стороны, указанное обстоятельство может быть использовано для нахождения второго и последующих собственных значений и соответствующих им собственных векторов. Для этого нужно выбрать начальное приближение $\mathbf{u}^{(0)}$ ортогональным найденным ранее векторам $\mathbf{u_1}, \mathbf{u_2}, \ldots$, вычитая из произвольно выбранного начального приближения его проекцию на первые m найденных ранее решений:

$$\mathbf{u}^{(0)} \quad \leftarrow \quad \mathbf{u}^{(0)} - \sum_{j=1}^{m} \mathbf{u_j} \frac{(\mathbf{u_j}, \mathbf{u}^{(0)})}{\|\mathbf{u_j}\|^2}. \tag{37}$$

Поскольку векторы $\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_m}$ были найдены в результате численного решения и известны приближённо, процесс ортогонализации (37) имеет смысл проводить не только перед началом итерационного процесса (33), но также и после его завершения, и в нескольких промежуточных точках.

1.7.3. Метод обратных итераций со сдвигом

Рассмотренный выше степенной метод позволяет искать наибольшее по модулю собственное значение матрицы. Для решения обратной задачи — поиска *наименьшего* по модулю собственного значения — очевидно, могут быть использованы итерации с заменой матрицы A на A^{-1}. Действительно, если $\lambda_1, \ldots, \lambda_n$ — собственные числа матрицы A, то матрица A^{-1} будет иметь собственные числа, равные $\lambda_1^{-1}, \ldots, \lambda_n^{-1}$. При выполнении условия (31), степенной метод сойдётся к λ_n^{-1}, что позволит найти наименьшее по модулю собственное значение λ_n матрицы A.

Выполним теперь ещё одно преобразование матрицы A, что позволит нам весьма эффективно (быстро) найти любое собственное значение λ, для которого известно начальное приближение $\tilde{\lambda} \approx \lambda$. А именно, *сдвинем* матрицу на величину $\tilde{\lambda}$ и применим степенной метод для $(A - \tilde{\lambda}E)^{-1}$:

---

[4] В пределе $k \to \infty$ направление вектора $\mathbf{u^{(k)}}$ может совпадать с направлением $\mathbf{u_1}$ даже при $a_1 = 0$ из-за наличия ошибок округления.

[5] Например, при поиске основного уровня энергии квантовой частицы в симметричной потенциальной яме $U(x) = U(-x)$ начальное приближение $\psi^{(0)}(x)$ должно быть либо чётной функцией x, либо функцией без определённой чётности.

$$\mathbf{u}^{(k+1)} = \big(A - \tilde{\lambda}E\big)^{-1}\mathbf{u}^{(k)}. \tag{38}$$

Идея перехода к обратной *сдвинутой* матрице в (38) состоит в том, что собственное число $(\lambda - \tilde{\lambda})^{-1}$ матрицы $(A - \tilde{\lambda}E)^{-1}$ будет больши́м по величине, что обеспечит быструю сходимость степенного метода.

В соответствии с (35), (36), сходимость метода (38) будет тем быстрее, чем меньше модуль разности $|\lambda - \tilde{\lambda}|$, т. е. чем более точным было начальное приближение $\tilde{\lambda}$. Следовательно, скорость сходимости можно повысить (сделать квадратичной), если в (38) вместо фиксированного сдвига $\tilde{\lambda}$ использовать приближение $\lambda^{(k)}$, найденное на предыдущей итерации. Заметим, однако, что использование переменного сдвига $\tilde{\lambda}$ в (38) предполагает вычисление обратной матрицы на каждом шаге итерационного процесса. В этой связи выгоднее переписать (38), домножив обе части равенства на матрицу $(A - \tilde{\lambda}E)$, и вычислять $\mathbf{u}^{(k+1)}$, решая систему линейных уравнений:

$$\big(A - \lambda^{(k)}E\big)\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)}, \quad \lambda^{(k+1)} = \lambda^{(k)} + \frac{\|\mathbf{u}^{(k)}\|}{\|\mathbf{u}^{(k+1)}\|}. \tag{39}$$

Для начала итерационного процесса (39) необходимо выбрать произвольный вектор $\mathbf{u}^{(0)}$ и начальное приближение к искомому собственному значению $\lambda^{(0)}$. Применяя (39) при $k = 0$, находим вектор $\mathbf{u}^{(1)}$ из решения системы линейных уравнений и вычисляем $\lambda^{(1)}$. Далее повторяем процесс (39) в цикле ($k = 1, 2, \ldots$), завершая его по достижении требуемого уровня точности.

1.7.4. Метод Ньютона

Посмотрим на спектральную задачу $A\mathbf{u} - \lambda\mathbf{u} = 0$ как на систему из n уравнений на $n+1$ неизвестную величину $u_1, \ldots, u_n, \lambda$. Система является нелинейной ввиду наличия члена $\lambda\mathbf{u}$. Линеаризуем её подстановками $\mathbf{u} = \mathbf{u}^{(k)} + \delta\mathbf{u}^{(k+1)}$, $\lambda = \lambda^{(k)} + \varepsilon^{(k+1)}$. Здесь $\mathbf{u}^{(k)}$ и $\lambda^{(k)}$ — известные на k-й итерации приближения к искомым собственному вектору и собственному значению, $\delta\mathbf{u}^{(k+1)}$ и $\varepsilon^{(k+1)}$ — малые поправки к ним, которые будут найдены на $(k+1)$-й итерации. Пренебрегая квадратичными членами $\varepsilon^{(k+1)}\delta\mathbf{u}^{(k+1)}$, получаем линейную систему на $(k+1)$-й итерации:

$$\big(A - \lambda^{(k)}E\big)\delta\mathbf{u}^{(k+1)} - \varepsilon^{(k+1)}\mathbf{u}^{(k)} = -\big(A - \lambda^{(k)}E\big)\mathbf{u}^{(k)}.$$

Поскольку собственный вектор определён с точностью до множителя, можно положить $\delta u_n^{(k+1)} = 0$, в результате чего получим систему из n уравнений на n неизвестных величин $\delta u_1^{(k+1)}, \ldots, \delta u_{n-1}^{(k+1)}, \varepsilon^{(k+1)}$:

$$B^{(k)} \cdot \begin{pmatrix} \delta u_1^{(k+1)} \\ \vdots \\ \delta u_{n-1}^{(k+1)} \\ \varepsilon^{(k+1)} \end{pmatrix} = -\big(A - \lambda^{(k)} E\big)\mathbf{u}^{(k)}, \qquad \text{где } B^{(k)} \equiv$$

$$\equiv \begin{pmatrix} A_{11} - \lambda^{(k)} & A_{12} & \dots & A_{1,n-1} & -u_1^{(k)} \\ \vdots & \vdots & & \vdots & \vdots \\ A_{n-1,1} & A_{n-1,2} & \dots & A_{n-1,n-1} - \lambda^{(k)} & -u_{n-1}^{(k)} \\ A_{n,1} & A_{n,2} & \dots & A_{n,n-1} & -u_n^{(k)} \end{pmatrix}. \tag{40}$$

Решение исходной спектральной задачи получается в результате итерационного процесса $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \delta\mathbf{u}^{(k+1)}$, $\lambda^{(k+1)} = \lambda^{(k)} + \varepsilon^{(k+1)}$.

1.7.5. Метод вращений Якоби

Однократное применение большинства рассмотренных выше методов позволяет найти лишь одно собственное значение и соответствующий ему собственный вектор матрицы. Для полноты картины рассмотрим *метод вращений Якоби* для одновременного нахождения *всех* собственных значений и векторов симметричных вещественных матриц[6]. Идея метода Якоби лежит в основе большинства используемых в настоящее время наиболее эффективных алгоритмов: матрица A приводится к диагональному виду Λ путём последовательности преобразований подобия.

Действительно, из курса алгебры известно, что если A — симметричная матрица, то существует ортогональная матрица V ($V^T = V^{-1}$), такая что $V^{-1}AV = \Lambda$, где Λ — диагольная матрица, состоящая из собственных значений матрицы A. В методе Якоби задача нахождения преобразования подобия V решается методом последовательных приближений: строится последовательность матриц $A^{(0)} = A$, $A^{(1)}$, $A^{(2)}, \dots$, так что $A^{(k)} \to \Lambda$. Для построения следующего приближения $A^{(k+1)}$ по заданной матрице $A^{(k)}$ необходимо вначале найти максимальный по модулю элемент, лежащий выше главной диагонали, и

[6] Помимо того, что симметричные (эрмитовы) матрицы играют важную роль во многих областях физики, эрмитовость матрицы является достаточным условием устойчивости собственных векторов по матричным элементам, в то время как спектральная задача для несимметричных матриц может быть плохо обусловленной. Красивый пример высокой чувствительности несимметричной матрицы Уилкинсона к малому возмущению матричных элементов приведён в монографии [2, с. 161]. Более подробный анализ вопроса можно найти в книге [А3, с. 93].

запомнить его индексы i_k, j_k:

$$i_k,\, j_k: \quad \left|a^{(k)}_{i_k j_k}\right| = \max_{i<j} \left|a^{(k)}_{ij}\right|. \tag{41}$$

Основная идея метода заключается в том, чтобы с помощью поворота в плоскости (i_k, j_k)

$$A^{(k+1)} = V^T_{(i_k j_k)} A^{(k)} V_{(i_k j_k)} \tag{42}$$

занулить максимальный по модулю недиагональный элемент (i_k, j_k) на следующем, $(k+1)$-м, шаге итерационного процесса, приблизив тем самым матрицу $A^{(k+1)}$ к искомой диагональной матрице Λ. На следующей итерации данный элемент вновь будет отличен от нуля, однако несложно увидеть, что после каждого поворота (42) сумма квадратов внедиагональных элементов матрицы A будет уменьшаться на величину $2a^2_{i_k j_k}$, а сумма квадратов диагональных элементов, напротив, возрастёт на ту же величину: сумма квадратов всех матричных элементов $a_{ij} a_{ij}$ есть скаляр и не изменяется при поворотах системы координат. Отсюда очевидно, что итерационный процесс, построенный на зануления максимальных по модулю недиагональных элементов посредством плоских вращений, будет сходиться к искомой диагональной матрице $\mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$. Результирующая матрица поворота, равная произведению матриц плоских вращений на каждом шаге, будет содержать координаты собственных векторов. Рассмотрим детали алгоритма и формулы для эффективного преобразования матрицы A.

Поворот в плоскости (i_k, j_k) описывается ортогональной матрицей $V_{(i_k j_k)}$, для элементов v_{ij} которой можем записать:

$$\begin{aligned} v_{ii} &= 1 \quad \text{при} \quad i \neq i_k,\, i \neq j_k; \\ v_{ij} &= 0 \quad \text{при} \quad i \neq i_k,\, i \neq j_k,\, j \neq i_k,\, j \neq j_k; \\ v_{i_k i_k} &= v_{j_k j_k} = c; \\ -v_{i_k j_k} &= v_{j_k i_k} = s, \end{aligned} \tag{43}$$

где для краткости использованы обозначения

$$c \equiv \cos\varphi_k, \quad s \equiv \sin\varphi_k. \tag{44}$$

Подставляя (43) в (42), получим для элемента (i_k, j_k) матрицы $A^{(k+1)}$:

$$a^{(k+1)}_{i_k j_k} = (c^2 - s^2) a^{(k)}_{i_k j_k} + c \cdot s \cdot (a^{(k)}_{j_k j_k} - a^{(k)}_{i_k i_k}).$$

Выбрав φ_k так, чтобы

$$\mathrm{tg}(2\varphi_k) = \frac{2a^{(k)}_{i_k j_k}}{a^{(k)}_{i_k i_k} - a^{(k)}_{j_k j_k}}, \quad |\varphi_k| \leq \frac{\pi}{4}, \tag{45}$$

получим $a_{i_k j_k}^{(k+1)} = 0$, что приблизит матрицу $A^{(k+1)}$ к диагональной. Для выполнения (45) можно вычислить значения $c \equiv \cos\varphi_k$ и $s \equiv \sin\varphi_k$ по следующим формулам:

$$c = \sqrt{\frac{1}{2}\left(1 + \frac{|q|}{d}\right)}, \quad s = \operatorname{sign}(q a_{i_k j_k}) \sqrt{\frac{1}{2}\left(1 - \frac{|q|}{d}\right)}, \tag{46}$$

где

$$q = a_{i_k i_k} - a_{j_k j_k}, \quad d = \sqrt{q^2 + 4a_{i_k j_k}^2}. \tag{47}$$

Вычислив значения c и s по формулам (46) и (47) и подставив их в (43) и далее в (42), мы можем найти следующее приближение $A^{(k+1)}$, сделав шаг итерационного процесса. Заметим, однако, что непосредственное вычисление произведения матриц в формуле (42) крайне неэффективно ввиду большого количества нулей и единиц в $V_{(i_k j_k)}$ (43). В самом деле, при повороте в плоскости (i_k, j_k) из n^2 элементов матрицы $A^{(k)}$ изменятся лишь $4n - 4$ чисел в двух строках и двух столбцах с номерами i_k и j_k. В этой связи для выполнения плоского поворота (42) матрицы $A^{(k)}$ следует отказаться от использования общего вида матричного умножения в пользу специальных формул (48–50), которые могут быть легко получены из (42) исключением суммирования нулевых элементов и умножений на единицу в многомерных матрицах плоского поворота. В соответствии с (48), бóльшая часть[7] коэффициентов матрицы $A^{(k)}$ остаётся без изменений:

$$a_{\alpha\beta}^{(k+1)} = a_{\alpha\beta}^{(k)} \quad \text{при} \quad \alpha \neq i_k,\ \alpha \neq j_k,\ \beta \neq i_k,\ \beta \neq j_k. \tag{48}$$

(Греческими буквами α, β обозначены индексы, пробегающие в цикле значения от 1 до n за исключением i_k и j_k; в противоположность этому, индексы i_k и j_k определяются выражением (41) и являются фиксированными в пределах одной итерации.) Следующие две формулы определяют преобразование большинства коэффициентов в двух строках с индексами i_k и j_k:

$$\begin{aligned} a_{\alpha i_k}^{(k+1)} &= a_{i_k \alpha}^{(k+1)} = c a_{\alpha i_k}^{(k)} + s a_{\alpha j_k}^{(k)}, \quad \text{при} \quad \alpha \neq i_k,\ \alpha \neq j_k; \\ a_{\alpha j_k}^{(k+1)} &= a_{j_k \alpha}^{(k+1)} = c a_{\alpha j_k}^{(k)} - s a_{\alpha i_k}^{(k)}, \quad \text{при} \quad \alpha \neq i_k,\ \alpha \neq j_k. \end{aligned} \tag{49}$$

Наконец, последние три формулы определяют преобразование коэф-

[7] В предположении большого размера матриц $n \gg 1$.

фициентов на пересечении строк и столбцов с номерами i_k, j_k:

$$\begin{aligned} a_{i_k i_k}^{(k+1)} &= c^2 a_{i_k i_k}^{(k)} + 2cs a_{i_k j_k}^{(k)} + s^2 a_{j_k j_k}^{(k)}; \\ a_{j_k j_k}^{(k+1)} &= s^2 a_{i_k i_k}^{(k)} - 2cs a_{i_k j_k}^{(k)} + c^2 a_{j_k j_k}^{(k)}; \\ a_{i_k j_k}^{(k+1)} &= a_{j_k i_k}^{(k+1)} = (c^2 - s^2) a_{i_k j_k}^{(k)} + cs\big(a_{j_k j_k}^{(k)} - a_{i_k i_k}^{(k)}\big). \end{aligned} \tag{50}$$

Для удобства использования перечислим основные шаги метода вращений Якоби:

1) поиск максимального по модулю наддиагонального элемента в матрице $A^{(k)}$ — определение индексов i_k, j_k (41);
2) проверка условия малости $|a_{i_k,j_k}|$: если наибольший по модулю недиагональный элемент матрицы $A^{(k)}$ мал, завершаем итерационный процесс;
3) вычисление $c = \cos\varphi_k$, $s = \sin\varphi_k$ по формулам (46, 47);
4) переход от матрицы $A^{(k)}$ к $A^{(k+1)}$ по формулам (48–50);
5) вычисление матрицы перехода $T = V_{(i_0 j_0)} V_{(i_1 j_1)} \cdot \ldots \cdot V_{(i_k j_k)}$, столбцы которой при $k \to \infty$ дают собственные векторы матрицы A.

1.7.6. Сравнение методов

Степенной метод исключительно прост в реализации, однако достаточно эффективен лишь для ленточных матриц при $|\lambda_1/\lambda_2| \gg 1$. В случае близких по модулю собственных значений скорость сходимости степенного метода падает обратно пропорционально разности $1 - |\lambda_2/\lambda_1|$. Однако даже если собственные значения заметно различаются, другие рассмотренные выше методы, как правило, демонстрируют более высокую эффективность, что ограничивает практическую применимость степенного метода.

Методы линеаризации и обратных итераций со сдвигом обеспечивают достаточно быструю сходимость (квадратичную вблизи корня) и особенно эффективны в случае ленточных матриц. Дополнительным преимуществом по сравнению со степенным методом является возможность нахождения *любого* собственного значения, для которого известно начальное приближение. Как и в случае трансцендентных уравнений, сходимость метода Ньютона не гарантируется: при неудачно выбранном начальном приближении итерационный процесс может быть расходящимся или сходиться не к искомому собственному значению, а к другому.

Метод вращений Якоби годится лишь для симметричных вещественных матриц. Хотя он уступает в эффективности QR-алгоритму,

авторы монографии [4, с. 571] рекомендуют его к использованию при работе с не слишком большими матрицами ввиду высокой точности и простоты программирования.

Как и при рассмотрении других тем данного курса, в этой главе мы познакомились лишь с узким кругом базовых методов, наиболее простых в понимании и реализации в программном коде. Вместе с тем данный раздел численных методов имеет и свою специфику. Спектральные задачи, возникающие на практике, зачастую сопряжены с обработкой огромных массивов информации — матриц большого размера, возникающих при дискретизации непрерывных функций и операторов. Вследствие этого вопросы эффективности используемых численных методов и их программных реализаций становятся критически важными, что заставляет задуматься об использовании программных решений из готовых библиотек. Авторы монографии [4, с. 567] пишут по этому поводу: «Возможно, вы уже пришли к заключению, что решение спектральных задач — достаточно непростое дело. И это действительно так. Это одна из немногочисленных тем в данной книге, в которой мы не советуем вам избегать готовых решений».

Большинство использованных до настоящего времени пакетов для решения спектральных задач восходят корнями к монографии [A5]. Реализация методов, изложенных в этой книге, получила широкое распространение в виде открытого набора програм EISPACK на Фортране. Более современной библиотекой, пришедшей ей на смену, стала LAPACK, также написанная на Фортране с использованием подпрограмм линейной алгебры LINPACK. Библиотека LAPACK отличается более высокой производительностью на современных компьютерах, портирована на Си (CLAPACK), имеет расширения для параллельных вычислений (ScaLAPACK).

Упражнения

1) Используя метод прогонки для решения систем линейных уравнений с трёхдиагональными матрицами, напишите программный код для интерполяции таблицы значений кубическими сплайнами [1, с. 83]. Исследуйте зависимость погрешности аппроксимации функции Рунге от количества узлов равномерной сетки, сравните с результатом полиномиальной интерполяции — см. задачу 7 на с. 85 [1].

2) Найти уровень энергии и волновую функцию $\psi_0(x)$ основного состояния в потенциальной яме $U(x)$, решая конечномерный аналог

спектральной задачи для одномерного стационарного уравнения Шрёдингера

$$\left(-\frac{1}{2}\frac{\partial^2}{\partial x^2} + U(x) - E\right)\psi(x,t) = 0, \qquad |\psi(x)| \to 0 \quad \text{при} \quad x \to 0.$$

Для поиска наименьшего собственного значения $\hat{H}\psi = E_0\psi$ трёхдиагональной матрицы $\hat{H}$ использовать метод обратных итераций. Проверить работу программы, сравнив с точным решением для $U(x) = \frac{1}{2}x^2$.

3) В условиях предыдущей задачи найти волновую функцию и уровень энергии первого возбуждённого состояния.

4) Что получится, если при поиске первого возбуждённого состояния методом обратных итераций выполнять ортогонализацию с $|\psi_0\rangle$ (37) только перед началом итерационного процесса? Только по окончании итерационного процесса? Коммутируют ли итерации (33) с вычислением проекции (37)? Как это соотносится с тем фактом, что и матрица $\hat{H}$, и проектор на ортогональное дополнение к $|\psi_0\rangle$ имеют диагональный вид в базисе $|\psi_0\rangle, \ldots, |\psi_n\rangle$?

5) Пусть собственные значения матрицы A удовлетворяют условию $|\lambda_1| > |\lambda_2| > |\lambda_3| > \ldots$, причем $|\lambda_1/\lambda_2| = |\lambda_2/\lambda_3| = \ldots = 1{,}1$. Считая, что для нахождения каждого из собственных значений использовалось 100 итераций степенного метода (33), оцените погрешность вычисления λ_1, $\lambda_2, \ldots$, полагая, что в начальном приближении (32) $a_1 \approx a_2 \approx \ldots$

6) Пусть требуется найти наибольшее по модулю собственное значение λ_1 матрицы A порядка $n = 500$ с точностью $\varepsilon = 10^{-4}$. Известно, что $|\lambda_1|/|\lambda_2| \approx 1{,}01$. Что быстрее: использовать степенной метод (33) непосредственно для матрицы A или предварительно вычислить A^2? Как изменится ответ, если требуется точность $\varepsilon = 10^{-12}$? Если при этом размер матрицы $n = 1000$?

7) Как зависит число арифметических операций на каждом шаге метода вращений Якоби от размера матрицы n? Можно ли сократить число операций в $\mathcal{O}(n)$ раз на каждом шаге метода (за исключением первого), используя тот факт, что на каждой итерации меняются только две строки и два столбца матрицы A?

8) Получите априорную оценку погрешности вычисления собственных значений в методе Якоби для симметричной матрицы $n \times n$

после m вращений. Сравните результат с апостериорной оценкой погрешности, полученной для случайной симметричной матрицы.

9) Используя метод вращений Якоби, в условиях первой задачи найдите n уровней энергии и соответствующих им волновых функций, исследуйте точность численных решений в зависимости от номера уровня.

2. Дискретное преобразование Фурье

Преобразование Фурье широко используется в теоретической физике и научно-технических расчётах, существенно упрощая анализ и понимание многих физических процессов и явлений в оптике, акустике, классической и квантовой механике, термодинамике и других областях. Два ключевых источника информации об окружающем нас мире — зрение и слух — позволяют непосредственно воспринимать спектры оптических и акустических колебаний. Многие линейные дифференциальные уравнения могут быть легко решены сведением к алгебраическим уравнениям в Фурье-представлении. Кроме того, быстрое преобразование Фурье лежит в основе эффективных алгоритмов расщепления по физическим процессам для интегрирования нелинейных дифференциальных уравнений в частных производных. Всё это обуславливает необходимость знакомства с базовыми понятиями и алгоритмами преобразования Фурье для дискретного набора данных на ЭВМ, чему и посвящена данная глава.

Оговоримся сразу, что анализ ряда аспектов дискретного преобразования Фурье выполнен в данной главе последовательно в спектральном и временно́м представлении. В этом смысле представленное изложение может показаться кому-то из читателей излишне сложным и избыточным. Разумеется, можно было бы ограничиться лишь одним из альтернативных объяснений каждого эффекта, однако опыт показывает, что на первый взгляд простые вопросы, связанные с преобразованием Фурье, способны вызвать затруднение не только у студентов, но и аспирантов и научных сотрудников. В этой связи мы осознанно допустили некоторую избыточность изложения в данной главе в надежде на то, что умение посмотреть на рассматриваемые здесь вопросы с разных точек зрения окажется полезным.

2.1. Преобразование *vs.* ряд Фурье

Из курса анализа известно интегральное преобразование Фурье

$$\tilde{f}(\omega) = \int\limits_{-\infty}^{+\infty} f(t)e^{i\omega t}\,dt, \qquad f(t) = \frac{1}{2\pi}\int\limits_{-\infty}^{+\infty} \tilde{f}(\omega)e^{-i\omega t}\,d\omega, \tag{51}$$

а также ряды Фурье τ-периодической функции:

$$f(x) = \sum_{m=-\infty}^{+\infty} a_m e^{ik_m x}, \qquad a_m = \frac{1}{\tau}\int\limits_0^{\tau} f(x)e^{-ik_m x}\,dx, \tag{52}$$

где $k_m = 2\pi m/\tau$. Напомним, что выбор знака в показателе экспоненты прямого преобразования Фурье, как и выбор коэффициента перед интегралом является произвольным; важно лишь, чтобы знаки в показателе в определении прямого и обратного преобразования были противоположными, а произведение коэффициентов перед интегралами было равно $(2\pi)^{-1}$.

Напомним также о тесной связи между интегральным преобразованием (51) и рядами Фурье (52). Для этого рассмотрим интегральное преобразование Фурье от τ-периодической функции $f(t)$; распишем интеграл по всей числовой оси[8] на сумму интегралов по отрезкам длины τ и воспользуемся условием $f(t+\tau) = f(t) \quad \forall t$:

$$\tilde{f}(\omega) = \sum_{n=-\infty}^{+\infty} \int\limits_{n\tau}^{(n+1)\tau} f(t)e^{i\omega t}\,dt = \lim_{N\to\infty} \sum_{n=-N}^{N} e^{i\omega n\tau} \int\limits_0^{\tau} f(t')e^{i\omega t'}\,dt'.$$

В соответствии с (51) интеграл в полученном выражении есть преобразование Фурье от одного периода функции $f(t)$, т. е. от $f_1(t)$: $f_1 \equiv f(t)$ при $0 \le t < \tau$ и $f_1 \equiv 0$ при $t \notin [0,\tau)$. Вычисляя сумму экспонент и переходя к пределу $N \to \infty$, имеем:

$$\tilde{f}(\omega) = \tilde{f}_1(\omega) \cdot \frac{2\pi}{\tau} \sum_{m=-\infty}^{+\infty} \delta\left(\omega - \frac{2\pi m}{\tau}\right). \tag{53}$$

Полученное выражение (53) иллюстрирует хорошо известный в физике факт: спектр регулярной последовательности большого числа N

[8] Мы представляем читателю возможность самостоятельно провести более строгие выкладки, регуляризовав выражения с помощью множителя $\exp(-\alpha|t|)$ с последующим переходом к пределу $\alpha \to 0$ для обеспечения сходимости интегралов и возможности смены порядка суммирования и интегрирования.

одинаковых сигналов имеет вид эквидистантного набора узких линий. Интервал между спектральными линиями равен частоте повторения сигналов, а их огибающая совпадает со спектром единичного сигнала в последовательности. В пределе к $N \to \infty$ последовательность сигналов переходит в периодическую функцию времени $f(t)$, а её спектр может быть описан дискретным бесконечным набором коэффициентов Фурье (суть коээфициенты перед δ-функциями в (53)). В таком случае обычно оперируют с рядами Фурье, что избавляет от необходимости работать с обобщёнными функциями. Однако полезно понимать и помнить, что по сути в обоих случаях выполняется одно и то же преобразование.

2.2. Наводящие соображения

Зададимся теперь вопросом, как выполнить преобразование Фурье численно и какое определение для этого следует использовать. Однако прежде чем давать формальное определение, как это обычно делается в учебниках по математике, рассмотрим наводящие соображения и придём к этому определению путём логических рассуждений. Тем читателям, для которых ближе и понятнее более строгий математический подход, рекомендуем сразу перейти к следующему параграфу 2.3.

Обратим внимание, что данные ранее определения (51) и (52) включают интегралы по бесконечной области либо суммы бесконечного ряда. Очевидно, что в численных расчётах область определения функций и количество коэффициентов ряда будут конечными.

Возьмём за основу формулу (51) и заменим в ней интеграл суммой, используя простейшую квадратурную формулу прямоугольников. При этом здесь и далее будем предполагать, что значения функции $f(t)$ известны нам в n узлах равномерной сетки t_k, $k = 0, 1, n-1$:

$$\tilde{f}_j = \frac{T}{n} \sum_{k=0}^{n-1} f_k \exp(i\omega_j t_k), \tag{54}$$

где T — ширина сетки. Обратное преобразование запишем по аналогии с (51) и (54) в виде

$$f_k = \frac{1}{T} \sum_j \tilde{f}_j \exp(-i\omega_j t_k). \tag{55}$$

Обратим внимание, что функция $f(t)$, определяемая своими значениями f_k в узлах сетки, является периодической (55). Подчеркнём, что данное обстоятельство имеет более глубокую природу и не связано напрямую со сделанными выше предположениями при записи формулы

(55). В самом деле, как бы мы ни определили дискретное преобразование Фурье, оно непременно должно выражаться через сумму экпонент с некоторыми коэффициентами, причём число членов в такой сумме обязано быть *конечным* в силу ограниченности памяти и быстродействия ЭВМ — именно это обстоятельство и обуславливает периодичность функции $f(t)$, получаемой в результате преобразования. Таким образом, хотя мы и брали за основу формулы интегрального преобразования Фурье (51), дискретное преобразование Фурье записывается в виде суммы комплексных экспонент и приводит к периодическим функциям, что делает его похожим на ряды Фурье (52).

В силу симметрии прямого и обратного преобразований очевидно, что выражение (54) описывает периодическую функцию частоты ω. Действительно, если в формуле (54) заменить ω_j на $\omega_j + 2\pi n/T$, где $T \equiv t_n - t_0$ — ширина t-сетки, спектральные амплитуды $\tilde{f}_j$ не изменятся либо приобретут фиксированный фазовый сдвиг, не зависящий от j. Данный эффект, известный как подмена частот (aliasing), подробнее рассмотрен в п. 2.5.

До сих пор мы ничего не сказали об узлах ω-сетки: формально выражение (54) допускает вычисление функции $\tilde{f}(\omega)$ в произвольных точках ω_j. Однако при этом нужно иметь в виду следующие обстоятельства:

1) как уже было сказано выше, функция $\tilde{f}(\omega)$ имеет период, равный $2\pi n/T$, поэтому имеет смысл ограничить значения ω_j одним (первым) периодом;

2) нулевая частота имеет важный физический смысл (среднее значение сигнала), поэтому один из узлов сетки должен соответствовать $\omega = 0$ (для определённости возьмём $\omega_0 = 0$);

3) дискретное преобразование Фурье (54) можно рассматривать как линейное преобразование n комплексных коэффициентов f_k в коэффициенты $\tilde{f}_j$ и обратно. Мы хотим определить преобразования таким образом, чтобы они были однозначными и взаимно обратными. Следовательно, они должны осуществляться квадратными комплексными матрицами $n \times n$, т. е. число узлов на ω- и t-сетках должно быть одинаковым. Далее, чтобы прямое и обратное преобразования имели симметричный вид, ω-сетка также должна быть равномерной, откуда $\omega_j = 2\pi j/T$, а $t_k = Tk/n$.

2.3. Определение и свойства

С учётом сказанного выше можно переписать формулы (54, 55) и дать формальное определение дискертного преобразования Фурье:

$$\tilde{f}_j = C_+ \sum_{k=0}^{n-1} f_k \exp\left(\frac{2\pi i j k}{n}\right), \quad f_k = C_- \sum_{j=0}^{n-1} \tilde{f}_j \exp\left(-\frac{2\pi i j k}{n}\right), \qquad (56)$$

где C_+ и C_- — коэффициенты, причём $C_+ C_- = 1/n$. Прежде чем переходить к рассмотрению эффектов и особенностей использования дискретного преобразования Фурье в практике вычислений, полезно вначале формально посмотреть на выражения (56) как на взаимно однозначное отображение набора коэффициентов $f_k \leftrightarrow \tilde{f}_j$. Перечислим основные свойства этих преобразований:

1) Дискретное преобразование Фурье есть линейное преобразование векторного пространства $\mathbb{C}^n$ с матрицами $F^+_{jk} = C_+ \exp(2\pi i j k/n)$ и $F^-_{jk} = C_- \exp(-2\pi i j k/n)$.

2) Матрицы прямого и обратного дискретного преобразования Фурье взаимно обратны: $F^+_{jk} F^-_{kl} = \delta_{kl}$, так что суперпозиция прямого и обратного преобразований есть тождественное преобразование.

3) При симметричном выборе коэффициентов $C_+ = C_- = 1/\sqrt{n}$ матрицы преобразования (56) унитарны: $(F^\pm)^\dagger = F^\mp = (F^\pm)^{-1}$.

4) Как следствие, дискретное преобразование Фурье (56) сохраняет квадратичную норму (энергию). Кроме того, дискретное преобразование Фурье является хорошо обусловленным, т. е. последовательное применение прямого и обратного преобразований оставляет вектор f_k неизменным с точностью порядка машинного ε.

5) Применение формул (56) с предварительно вычисленными комплексными экспонентами требует выполнения $\mathcal{O}(n^2)$ арифметических операций; в п. 2.10 будет рассмотрена идея более эффективного алгоритма, позволяющего уменьшить число операций до $\mathcal{O}(n \log n)$.

Хотя в некоторых случаях и бывает полезным представлять себе дискретное преобразование Фурье в виде унитарной матрицы над $\mathbb{C}^n$, в большинстве задач продуктивнее смотреть на (56) как на отображение функций $f(t) \leftrightarrow \tilde{f}(\omega)$, каждая из которых задана на своей сетке, содержащей n узлов. Такой подход позволяет увидеть совершенно иной набор свойств и особенностей использования преобразований (56), о которых пойдёт речь в следующих параграфах.

2.4. Периодичность по времени

На с. 31 мы уже упоминали о том, что дискретное преобразование Фурье (55) порождает периодические функции времени. Рассмотрим сейчас этот вопрос более подробно на примере распространения сигнала $A_0(t)$ вдоль оси z в линии связи:

$$\frac{\partial A}{\partial z} + \frac{1}{v_g}\frac{\partial A}{\partial t} = 0. \tag{57}$$

Здесь A — комплексная огибающая передаваемого сигнала, v_g — групповая скорость распространения сигнала вдоль линии связи, z и t — пространственная координата и время.

Общее решение и решение задачи Коши $A(z,t) = f(z - v_g t) = A_0(t - z/v_g)$ немедленно выписывается методом характеристик: при распространении вдоль линии связи на расстояние z сигнал задерживается во времени на z/v_g. Для построения численного решения следует применить к задаче (57) метод функции Грина. С его помощью можно получить решение более общего уравнения:

$$\frac{\partial A}{\partial z} + \frac{1}{v_g}\frac{\partial A}{\partial t} + \frac{i}{2}\beta\frac{\partial^2 A}{\partial t^2} = 0, \tag{58}$$

имеющего второй порядок по времени и учитывающего также эффект дисперсии групповых скоростей (см. коэффициент β в (58)). Аналитическое решение уравнения (58) легко получить, выполнив преобразование Фурье по времени:

$$\tilde{A}(z,\omega) = \tilde{A}(0,\omega)\exp\left(\frac{i\omega}{v_g} + \frac{i}{2}\beta\omega^2\right) \quad \Rightarrow$$

$$A(z,t) = \frac{1}{2\pi}\int\limits_{-\infty}^{+\infty} d\omega\, \exp\left(\frac{i\omega}{v_g} + \frac{i}{2}\beta\omega^2 - i\omega t\right) \int\limits_{-\infty}^{+\infty} dt'\, A(0,t')e^{i\omega t'}. \tag{59}$$

На рис. 1 представлены результаты расчёта распространения импульса с огибающей sech^2 в соответствии с выражением (59) при $\beta = 0$. На графике 1 (а) показано начальное условие $A(z = 0; t)$ — импульс с огибающей sech^2. Стрелкой указано направление смещения импульса по времени при его распространении вдоль z. График (б) соответствует некоторой точке $z > 0$. Видно, что импульс сместился по времени и дошёл до правого края сетки ширины $T = 1$, при этом его «хвост» появился на левом краю сетки.

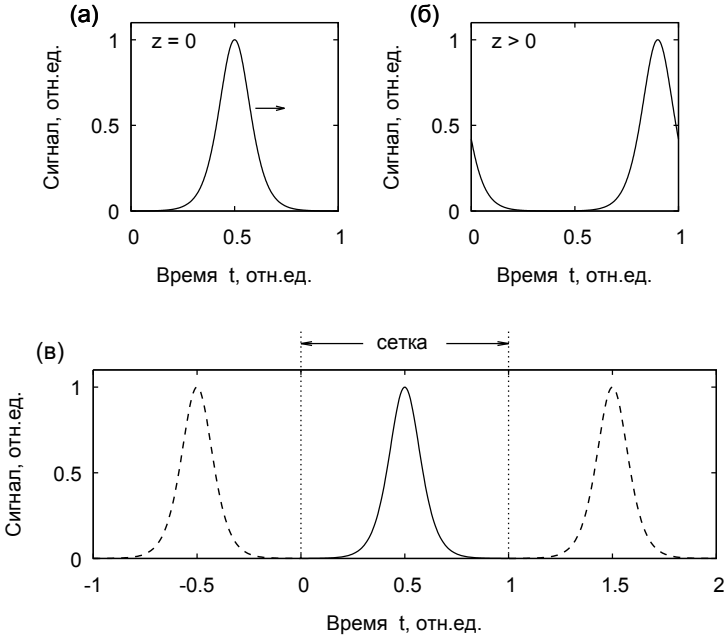

Рис. 1. При использовании дискретного преобразования Фурье численное решение на сетке является периодической функцией

Другими словами, в моделировании, использующем дискретное преобразование Фурье, всегда имеют дело с периодическими функциями времени. Это обстоятельство иллюстрирует рис. 1 (в), где показана периодическая последовательность импульсов и расчётная сетка, ширина которой равна периоду последовательности.

Также можно представлять себе график сеточной функции $f(t)$ нарисованным на листе бумаги, который свернули в цилиндр так, что правый и левый концы графика оказались склеены друг с другом. При этом очевидно, что импульс не может выйти за пределы сетки, но может лишь перемещаться по кругу. У сетки, как и у окружности, нет начала и конца, все её узлы топологически эквивалентны.

Неявно заданные периодические граничные условия, возникающие при выполнении дискретного преобразования Фурье, обязательно нужно учитывать при выполнении численного моделирования. Ширину сетки следует выбирать достаточно большой, чтобы исключить нежелательное наложение и взаимодействие переднего и заднего фронтов сигнала друг с другом.

2.5. Подмена частот

Из симметрии прямого и обратного преобразований Фурье очевидно, что аналогичная периодичность численного решения должна иметь место и в частотном представлении $\tilde{f}(\omega)$. Подобно тому как каждый узел временно́й сетки t_k соответствует бесконечному набору моментов времени $t_k + mT$, $m = 0, \pm 1, \pm 2 \ldots$, так и каждый узел ω_j *обратной* (частотной) сетки соответствует различным физическим частотам $\omega_j + 2\pi m/T$, $m \in \mathbb{Z}$. Данный эффект называется *подменой*, *переналожением*, или *маскировкой частот*. Для его обозначения также очень часто употребляется англицизм *элайзинг*, или *алиасинг* (от англ. *alias* — псевдоним, прозвище).

Природа данного явления достаточно проста. Если гармонический сигнал с частотой ω измеряется в дискретные моменты времени с равными интервалами τ, то частота ω по результатам таких измерений может быть определена лишь по модулю $\omega \mod \frac{2\pi}{\tau}$. Действительно, регистрируя изменение фазы $\delta\varphi$ сигнала между моментами времени измерений $t = 0, \tau, 2\tau, \ldots$, мы никак не можем отличить его от $\delta\varphi + 2\pi k$ при произвольном целом k. Сказанное иллюстрирует рис. 2 (а), на котором приведены графики $\sin t$ и $\sin 5t$: их значения совпадают в узлах сетки с шагом $\tau = \pi/2$ (показаны круглыми маркерами), поскольку аргумент первой гармоники возрастает между узлами сетки на величину $\delta\varphi = \pi/2$, а пятой гармоники — на $\frac{\pi}{2} + 2\pi$. Как следствие, частоты $\omega = 1$ и $\omega = 5$ оказываются неотличимыми на данной сетке.

На том же принципе основано действие стробоскопа; из-за подмены частот колёса автомобилей в кинофильмах вращаются в противоположную движению сторону, а экраны мониторов на видеозаписях непривычно сильно мерцают. Переналожение пространственных частот можно наблюдать, сделав фотографию компьютерного монитора, при этом частота измерений (пространственная) равна обратному расстоянию между пикселями матрицы цифрового фотоаппарата, а частотой сигнала является обратное расстояние между пикселями экрана на оптическом изображении, формируемом объективом фотоаппарата. Ещё одно «компьютерное» проявление эффекта подмены частот, которое легко можно увидеть в числе первых результатов поискового запроса *aliasing* в Интернете, связано с отображением шрифтов либо других геометрических форм с резкими наклонными краями. Поскольку идеально резкие края фигуры соответствуют медленно убывающей спектральной функции, пространственный спектр компьютерных шрифтов содержит неограниченно высокие частоты, которые при любой частоте дискретизации (при любом разрешении экрана или принтера) способ-

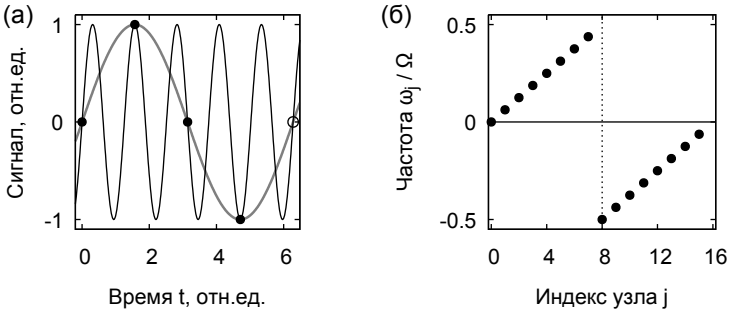

Рис. 2. (а) Эффект подмены частот; (б) частота ω_j узлов обратной сетки

ны приводить к эффекту подмены частот. Для преодоления данного эффекта при отображении шрифтов, воспроизведении и съёмке фото- и видео применяются разнообразные цифровые и аналоговые сглаживающие фильтры, предотвращающие появление резких краёв и связанных с ними слишком высоких частот в пространственном спектре (см. *anti-aliasing* в Google). При работе с гладкими функциями (в частности, при оцифровке сигнала в эксперименте, а также численном решении дифференциальных уравнений) необходимо следить за тем, чтобы частота дискретизации (обратная величина шага сетки) была как минимум вдвое выше верхней границы спектра регистрируемого сигнала.

2.6. Узлы обратной сетки

Как было показано в п. 2.5, каждый узел обратной сетки в общем случае может соответствовать различным физическим частотам:

$$\omega_j \sim \frac{2\pi j}{T} + m\Omega, \qquad \Omega \equiv \frac{2\pi}{\tau}, \tag{60}$$

где τ и $T = n\tau$ — шаг и ширина временно́й сетки, n — количество узлов, Ω — частота дискретизации (циклическая[9]), m — произвольное целое число. Как устранить неоднозначность по m в (60)? Такой вопрос возникает в физических расчётах при использовании уравнений, коэффициенты которых явно зависят от частоты ω. Например, в уравнении переноса (57) скорость распространения сигнала v_g может быть функцией ω (эффект дисперсии групповых скоростей).

[9]Напомним, что *циклическая* частота ω связана с *линейной* частотой ν соотношением $\omega = 2\pi\nu$. Как правило, для них используются не только различные буквы, но и даже разные размерности: циклическая частота обычно измеряется в обратных секундах, тогда как линейная — в Герцах.

Определение дискретного преобразования Фурье (56) подсказывает самый простой (но при этом ошибочный) способ решения проблемы, который состоит в том, чтобы при вычислении в (60) положить $m = 0$. Тогда, в полном соответствии с формулой (56), частоты ω_j узлов обратной сетки будут пропорциональны индексу узла $j = 0, 1, \ldots, n-1$. Чтобы понять, почему такой способ неправилен, вспомним, что спектр вещественного сигнала $f(t) = f^*(t)$ симметричен относительно нуля: $\tilde{f}(\omega) = \tilde{f}^*(-\omega)$. С другой стороны, выбрав $m = 0$ в (60), мы получим в спектре лишь неотрицательные частоты, т. е. будем не способны выполнить преобразование Фурье с ожидаемыми результатами ни для какой вещественной функции за исключением $f(t) \equiv \mathrm{const}$.

Чтобы найти правильное решение, рассмотрим гармонический сигнал $f(t) = \cos \omega_* t$ в «идеальном» случае: пусть мы можем регистрировать этот сигнал в течение длительного времени T, кратного периоду косинуса: $T = 2\pi q/\omega_*, \quad q \in \mathbb{N}$, что позволит нам избежать эффекта частокола, см. п. 2.7. Пусть также мы можем ипользовать сетку со сколь угодно малым шагом τ и сколь угодно большим числом узлов $n = T/\tau$, чтобы избежать эффекта подмены частот (см. п. 2.5). Что даст дискретное преобразование Фурье (56) от косинуса в этом заведомо «хорошем» предельном случае? Подставляя $\omega_* = 2\pi q/T$, $f_k = \cos(\omega_* t_k)$, $t_k = kT/n$ в определение (56) и пользуясь ортогональностью комплесных экспонент, легко увидеть, что мы получим в спектре лишь два ненулевых коэффициента:

$$\tilde{f}_j = \frac{C_+}{2} \sum_{k=0}^{n-1} \left(e^{2\pi i q k/n} + e^{-2\pi i q k/n} \right) e^{2\pi i j k/n} = \frac{C_+ n}{2} \left(\delta_{j,q} + \delta_{j,n-q} \right).$$

Как видно из формулы (60), отличные от нуля коэффициенты соответствуют частотам $\omega_q = m\Omega + \omega_*$ и $\omega_{n-q} = (m+1)\Omega - \omega_*$. С другой стороны, поскольку речь идёт о преобразовании Фурье функции $\cos \omega_* t$, мы ожидаем увидеть в спектре две линии: $\omega = \pm\omega_*$. Для этого нам нужно положить $m_1 = 0$ и $m_2 = -1$ в первом и во втором случае соответственно. Если же в формуле (60) мы положим $m \equiv 0$ для всех узлов обратной сетки, полученный в результате спектр не будет соответствовать нашим ожиданиям. В этой связи для половины узлов ($0 \le j < n/2$) мы должны положить $m = 0$, для оставшейся половины считать $m = -1$. При этом формула (60) станет однозначной и перепишется в виде

$$\omega_j = \begin{cases} j\Omega/n, & \text{при } j < n/2, \\ (-1 + j/n)\Omega, & \text{при } j \ge n/2, \end{cases} \tag{61}$$

где $\Omega = 2\pi/\tau = 2\pi n/T$ — частота дискретизации (циклическая). На рис. 2 (б) показан график, иллюстрирующий зависимость (61) частоты от индекса узлов обратной сетки на примере $n = 16$. До середины обратной сетки частота ω_j нарастает пропорционально индексу j от 0 до $\Omega/2$; в середине сетки происходит скачок на $-\Omega$, после которого частота вновь изменяется линейно по j с тем же наклоном. Максимальное по модулю значение частоты вдвое меньше частоты дискретизации ($\max \omega_j = \Omega/2$) и называется *частотой Найквиста*.

Несмотря на кажущуюся на первый взгляд сложность, полученный ответ (61) имеет достаточно простой физический смысл. В каждом узле сетки j мы должны разрешить неоднозначность выбора частоты ω_j (60) таким образом, чтобы абсолютное значение $|\omega_j|$ было минимальным. Здесь уместна аналогия с вращающимся колесом автомобиля в кинофильме: мы не знаем, на какой угол $\varphi + 2\pi m$ *в действительности* повернулось колесо между моментами последовательных измерений (съёмкой двух последовательных кадров киноплёнки), но мы интерпретируем увиденное на экране так, чтобы модуль угла поворота $|\varphi + 2\pi m|$ принимал бы минимальное возможное значение, т. е. чтобы $-\pi \leq \varphi + 2\pi m < \pi$.

Разумеется, сказанное выше относительно правильного способа разрешения неоднозначности по m в (60) справедливо лишь в случае, когда измерения f_k проводятся достаточно часто, т. е. шаг сетки по времени или пространству достаточно мал по сравнению с масштабом, на котором изменяется исследуемая функция f. Данный способ не может быть использован для предотвращения эффекта наложения частот; если в расчётах произошла подмена частот, следует повысить частоту дискретизации сигнала[10].

2.7. Эффект частокола

Эффект частокола, известный в англоязычной литературе как *picket fence effect*, связан с потерей спектральной информации, попадающей между узлами обратной сетки (61). Чтобы понять суть явления, удобно провести аналогию между анализом спектров, полученных в результате дискретного преобразования Фурье, и взглядом на мир через изгородь: в обоих случаях получаемая картина является фрагментарной, что иллюстрирует рис. 3. Видимые в промежутках между

[10] При оцифровке аналоговых сигналов для экономии ресурсов также целесообразно применение спектральных аналогвых фильтров низких частот (при обработке изображений — сглаживающих фильтров) при условии, что обрезаемые высокочастотные компоненты сигнала не содержат важной информации.

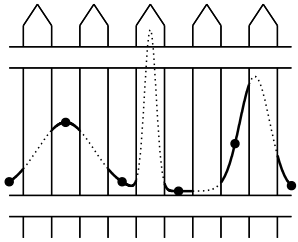

Рис. 3. Эффект частокола

досками частокола участки графика (спектра) показаны сплошной линией, скрытые — пунктиром; круглые маркеры соответствуют узловым точкам ω-сетки.

В показанном на рис. 3 примере после наложения частокола на некоторую функцию (спектр) только один из трёх пиков (первый) был передан без искажений. Второй (средний) пик полностью исчез, а высота третьего пика после наложения сетки оказалась в два раза меньше истинного значения.

Очевидно, что наличие в спектре сигнала пиков с шириной меньше шага обратной сетки (как в случае среднего пика на рис. 3), свидетельствует о недостаточно большой ширине сетки T. Таким образом, для устранения указанной проблемы необходимо увеличить время T регистрации сигнала, тем самым уменьшив шаг обратной сетки и повысив разрешение спектральной функции.

Однако одно лишь увеличение ширины сетки T не всегда позволяет полностью решить проблему. Чтобы понять это, вновь обратимся к преобразованию Фурье от гармонической функции. На этот раз вместо косинуса используем всего одну комплексную экспоненту для максимального упрощения выкладок:

$$f(t) = \exp(-i\omega_* t), \quad \omega_* = (2q+1)\frac{\pi}{T} \tag{62}$$

Частоту ω_* комплексной экспоненты выберем так, чтобы на сетке ширины T укладывалось $q + \frac{1}{2}$ периодов осцилляций. Положив $C_+ = 1/n$ в определении (56), в результате дискретного преобразования Фурье сигнала (62) получим:

$$\tilde{f}_j = \frac{1}{n}\sum_{k=0}^{n-1} \exp\left(\frac{2\pi i(j-q-1/2)k}{n}\right). \tag{63}$$

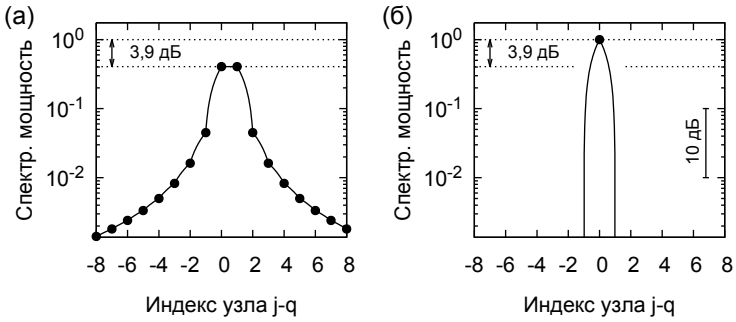

Рис. 4. Результат дискретного преобразования Фурье для гармонического сигнала с полуцелой (а) и целой (б) частотой на сетке с $n = 256$ узлами; сплошная кривая соответствует линейной интерполяции между узлами

Поскольку частота ω_* является полуцелой[11], комплексные экспоненты не являются ортогональными. Вычисляя сумму (63) по формулам для геометрической прогрессии, получаем для спектральной мощности сигнала:

$$\left|\tilde{f}_j\right|^2 = \frac{1}{n^2}\left|\frac{\sin\left(\pi\left(j-q-\frac{1}{2}\right)\right)}{\sin\left(\frac{\pi}{n}\left(j-q-\frac{1}{2}\right)\right)}\right|^2 = \frac{1}{n^2\sin^2\left(\frac{\pi}{n}\left(j-q-\frac{1}{2}\right)\right)}. \qquad (64)$$

График полученного выражения (64) в окрестности спектральной линии ($|j-q| \le 8$ при общем числе узлов сетки $n = 256$) показан на рис. 4 (а) на логарифмической шкале спектральных мощностей. Круглые маркеры соответствуют значениям функции в узлах обратной сетки, соединяющие их кривые — линейной интерполяции между узлами. Как и следовало ожидать, максимум спектральной мощности достигается в узлах ω_q и ω_{q+1}, ближайших к частоте ω_*. Вместе с тем обратим внимание на две неприятных особенности полученного ответа (64), которые представляют реальную проблему при спектральном анализе сигнала.

Для понимания первого аспекта проблемы вычислим максимальное значение спектральной мощности. В соответствии с (64), оно равно

$$\max_j |\tilde{f}_j|^2 = \left(n\sin\left(\frac{\pi}{2n}\right)\right)^{-2} = \frac{4}{\pi^2} + \mathcal{O}(n^{-2}) \text{ при } n \gg 1. \qquad (65)$$

[11] Здесь и далее, говоря о «целых» и «нецелых» частотах ω, мы имеем в виду отсутствие или наличие дробной части у безразмерной частоты $\omega T/(2\pi)$, равной отношению частоты ω к шагу обратной сетки $2\pi/T$.

Если бы частота ω_* в (62) была не полуцелой, а целой ($\omega_* = 2\pi q/T$), то, заменив $q - \frac{1}{2}$ на q в (63), мы бы получили $\max|\tilde{f}|^2 = \max|\delta_{j,q}|^2 = 1$, что на 3,9 дБ (в 2,5 раза) больше, чем в случае полуцелой частоты (65). Таким образом, если в спектре регистрируемого сигнала есть две линии с равными мощностями, то это ещё не означает, что при выполнении дискретного преобразования Фурье мы действительно увидим линии равной высоты! Так, в случае если одна из линий совпадёт с узлом обратной сетки, а другая окажется посередине между соседними узлами, мы увидим, что одна из линий выше другой на 3,9 дБ, ср. рис. 4 (а) и (б). Другими словами, проблема заключается в искажении формы спектра, полученного в результате дискретного преобразования Фурье. Мы уже ожидали возникновения этой проблемы, используя простые качественные соображения: см. рис. 3, на котором *видимая* высота третьего пика значительно ниже его *истинной* высоты. Математические выкладки (62–65) позволили нам количественно оценить масштаб проблемы (3,9 дБ). Кроме того, полученный ответ (65) позволяет сделать вывод о том, что величина, на которую уменьшается высота спектральной линии (3,9 дБ), не зависит от времени регистрации сигнала T (при условии, что частота ω_* остаётся полуцелой при использовании сеток с разными ширинами T). Таким образом, мы можем увеличить время регистрации сигнала в 3 или даже в 33 раза, повысив в соответствующее число раз разрешение спектральной функции, но это не поможет ни решить, ни даже уменьшить указанную проблему[12].

Куда делась спектральная энергия при уменьшении высоты линии на 3,9 дБ? В самом деле, как уже говорилось в п. 2.3, дискретное преобразование Фурье описывается унитарными матрицами и, следовательно, сохраняет квадратичную норму векторов. Таким образом, если при изменении частоты сигнала ω_* с $2\pi q/T$ на $2\pi(q + \frac{1}{2})/T$ происходит уменьшение высоты линии в 2,5 раза, то это должно сопровождаться увеличением ширины спектральной линии. И действительно, в соответствии с полученным выше выражением (64) и графиком на рис. 4 (а), спектральная мощность сигнала на полуцелой частоте демонстрирует медленный (степенной) спад из точки максимума, тогда как в случае целой частоты $\tilde{f}_j \sim \delta_{j,q}$, т. е. спектральная функция зануляется уже в соседних с пиком узлах сетки (рис. 4 (б)). Это составляет второй аспект проблемы: сигнал на частоте, расположенной между узлами обратной сетки, приводит к появлению широких фантомных линий в спектрах, полученных в результате дискретного преобразования Фурье. Спек-

[12] Это связано с тем, что спектр исследуемого сигнала (62) представляет собой δ-функцию, и, следовательно, его ширина всегда будет меньше шага обратной сетки при любом значении T.

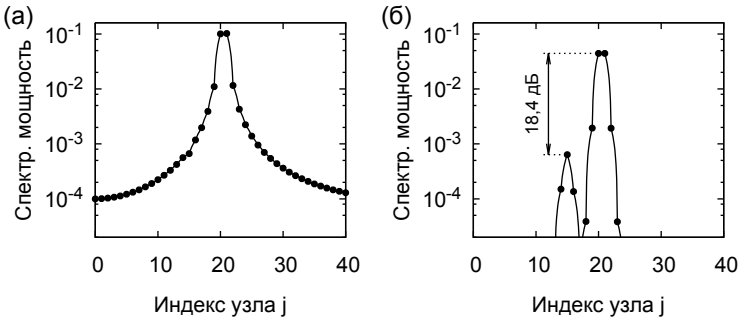

Рис. 5. Результат дискретного преобразования Фурье для сигнала (66), вычисленный с прямоугольным окном (а) и оконной функцией Ханна (б)

тральная энергия как бы «перетекает» из линии в соседние узлы, в связи с чем данное явление обозначается в англоязычной литературе термином *spectral leakage*.

2.8. Окно Ханна

Рассмотренный в предыдущем пункте эффект частокола приводит к искажению регистрируемой спектральной информации в случае, если частота сигнала не является целой (не кратна $2\pi/T$). Указанные искажения включают в себя уменьшение высоты и увеличение ширины спектральных линий, а также маскировку слабых спектральных линий за счёт слияния их с расположенными рядом более мощными линиями, уширенными за счёт эффекта частокола. В качестве иллюстрации на рис. 5 (а) показана спектральная мощность сигнала

$$f(t) = 0{,}1\sin(15t) + \cos(20{,}5t), \tag{66}$$

вычисленная на сетке шириной $T = 2\pi$ с числом узлов $n = 100$. Поскольку сигнал $f(t)$ вещественный, приведена лишь половина спектра при $\omega > 0$. Спектральная мощность сигнала отложена на вертикальной оси в логарифмическом масштабе. Обратим внимание: в спектре на рис. 5 (а) отчётливо видна всего одна спектральная линия, тогда как в соответствии с выражением (66) сигнал является суперпозицией двух гармоник!

Данная проблема легко решается за счёт использования оконных функций, рис. 5 (б). Чтобы понять смысл этого решения, удобно посмотреть на эффект частокола не в спектральном представлении, как мы делали выше в п. 2.7, а во времени.

Во временно́м представлении регистрация сигнала $f(t)$ на сетке с шириной T может быть представлена в виде произведения $f(t) \cdot h(t;T)$, где $h(t;T)$ — ступенчатая функция, значения которой во всех узлах сетки равны 1, а за пределами сетки — нулю:

$$h_k = h(t_k) = \begin{cases} 1, & \text{при} \quad 0 \le k < n, \\ 0, & \text{при} \quad k < 0 \ \text{и} \ k \ge n. \end{cases} \tag{67}$$

При этом функцию $h(t;T)$ называют также *прямоугольной оконной функцией* ввиду того, что она ограничивает окно длительности T, внутри которого исследуется сигнал $f(t)$, и имеет прямоугольный (ступенчатый) профиль.

Для примера на рис. 6 (а) показан гармонический сигнал $f(t) = \cos(5\pi t)$ и прямоугольное окно шириной $T = 1$. Поскольку частота сигнала не кратна шагу обратной сетки (их отношение равно 2,5), мы ожидаем появления эффекта частокола, см. п. 2.7. Снизу на панели (б) того же графика сплошной линией показано произведение $f(t) \cdot h(t;T)$ — сеточная функция. Как мы говорили выше в п. 2.4, при выполнении дискретного преобразоавния Фурье на сеточную функцию накладываются периодические граничные условия. В этой связи на рис. 6 (б) пунктиром показано также периодическое продолжение функции $f(t) \cdot h(t;T)$. Видно, что периодическое продолжение сеточной функции терпит разрывы в точках $t = mT$, $m \in \mathbb{Z}$. Как известно из курса анализа, спектральная мощность разрывных функций $|\tilde{f}(\omega)|^2$ медленно убывает с ростом частоты ω, что и приводит к уширению спектральных линий на рис. 4 (а) и 5 (а).

Данное рассмотрение даёт ключ к решению проблемы. Действительно, если эффект частокола возникает вследствие разрывности периодического продолжения сеточной функции $f(t) \cdot h(t;T)$, нам следует использовать другую оконную функцию, которая бы занулялась на границах окна (при $t = 0$ и $t = T$) и, следовательно, обеспечивала бы непрерывность произведения $f(t)h(t)$. Очевидно, существует бесчисленное множество функций $h(t)$, удовлетворяющих нулевым граничным условиям. Одним из часто используемых вариантов является *окно Ханна*, или *Ханнинга*[13]:

[13]Название предложено американскими математиками Ральфом Блэкмэном и Джоном Тьюки в честь австрийского метеоролога Джулиуса фон Ханна (Julius von Hann); применение окна Ханна к сигналу также обозначают в англоязычной литературе герундием *hanning*; не следует путать его с созвучным с *окном Хэмминга*, предложенного другим американским математиком, Ричардом Хэммингом (Hamming, Richard).

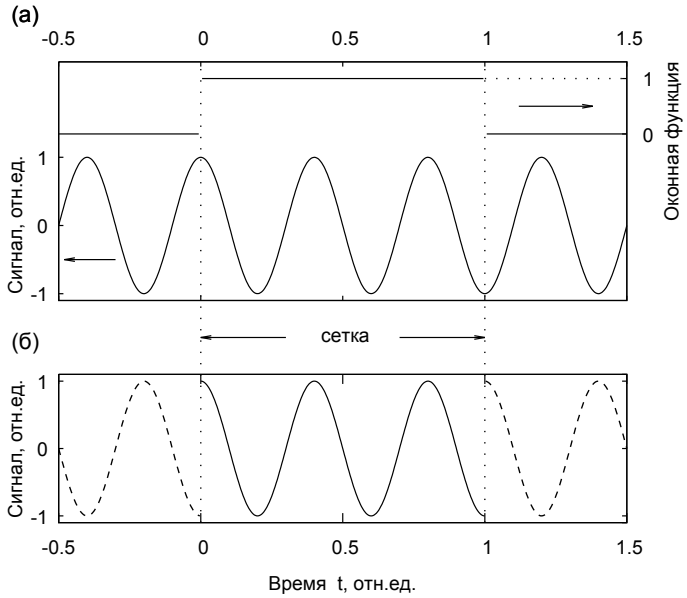

Рис. 6. Сигнал на сетке можно рассматривать как произведение $f(t)$ и ступенчатой оконной функции $h(t)$ (а), при этом периодическое продолжение сеточной функции fh может быть разрывным (б)

$$h_k = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi k}{n}\right)\right), \quad 0 \le k < n. \tag{68}$$

Данная функция зануляется на границах периода вместе со своей первой производной, что делает периодическое продолжение сеточной функции непрерывным и гладким.

Заметим, что часто встречается альтернативное определение окна Ханна с заменой $n \to n-1$ в знаменателе:

$$h_k = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi k}{n-1}\right)\right), \quad 0 \le k < n. \tag{69}$$

В первом случае период оконной функции (68) равен ширине сетки; такие оконные функции называют *периодическими*. Функции вида (69) называют *симметричными* ввиду симметричности набора из n коэффициентов h_k в (69). Когда число узлов сетки n велико, отличие между двумя способами определения практически несущественно.

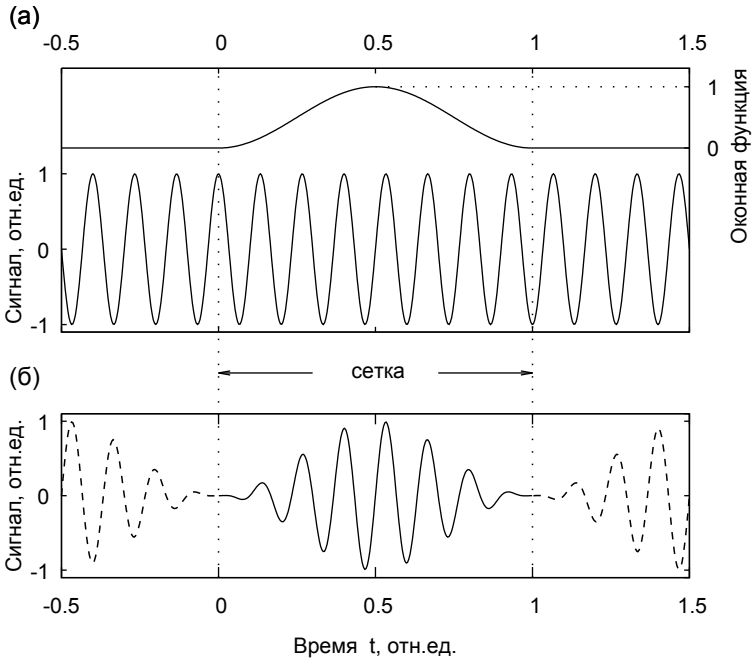

Рис. 7. Использование окна Ханна делает периодическое продолжение сеточной функции непрерывным и гладким

На рис. 7 показано применение окна Ханна к периодическому сигналу $f(t) = \cos(7{,}5 \cdot 2\pi t/T)$, $T = 1$. Из сравнения рис. 6 и 7 легко увидеть, что использование окна Ханна позволяет устранить разрыв на границе периода. Как следствие, это позволяет в значительной степени избавиться от эффекта частокола, что иллюстрирует рис. 5 (б), на котором показан спектр сигнала (66), полученный с использованием окна Ханна (68). В спектре отчётливо видны две спектральные линии, что соответствует суперпозиции двух гармоник в (66).

2.9. Другие оконные функции

Хотя поставленный в начале предыдущего параграфа вопрос о разрешении двух спектральных линий в спектре сигнала (66) был решён, внимательный читатель наверняка заметил, что высота спектральных линий на рис. 5 (б) отличается на 18,4 дБ, тогда как в соответствии с (66), амплитуды линий соотносятся как 1:10, что позволяет ожидать

отличия спектральных мощностей линий в 100 раз (на 20 дБ). Таким образом, использование оконной функции Ханна хотя и позволяет *в большинстве случаев* уменьшить искажения спектров, получаемых в результате дискретного преобразования Фурье оцифрованных сигналов, но всё же не решает проблему полностью. Использование оконной функции всегда предполагает некоторый компромисс между различными видами искажений. Соответственно, выбор той или иной оконной функции определяется спецификой задачи: тем, какие именно характеристики спектра представляют наибольшую ценность (ширина спектральных линий, относительная высота спектральных линий и соотношение шум/сигнал). В контексте данного курса обсуждаемые здесь вопросы определённо можно отнести к разряду «дополнительных», поэтому в ходе первого знакомства с темой можно смело переходить к следующему параграфу 2.10 на с. 49.

Для более детального количественного исследования свойств *оконного* преобразования Фурье удобно воспользоваться пропорциональностью спектра произведения $f(t)h(t)$ свёртке спектров сомножителей:

$$(\tilde{fh})(\omega) = \frac{1}{2\pi} \int\limits_{-\infty}^{+\infty} d\omega' \, \tilde{f}(\omega - \omega')\tilde{h}(\omega'). \tag{70}$$

Таблица значений именно этой спектральной функции (но отнюдь не «истинного» спектра $\tilde{f}(\omega)$ непрерывного и не ограниченного во времени сигнала $f(t)$) получается на обратной сетке в результате выполнения дискретного преобразования Фурье. Именно к спектральной функции (70) напрямую и непосредственно применимо понимание эффекта частокола в том виде, как оно представлено на рис. 3.

В предыдущих параграфах в качестве примеров мы рассматривали дискретное преобразование Фурье от гармонических сигналов, «истинный» спектр которых представляет собой одну или несколько δ-функций. Свёртка $\tilde{f}(\omega) = \delta(\omega - \omega_*)$ со спектром оконной функции $\tilde{h}(\omega)$ в (70) даёт $\tilde{h}(\omega - \omega_*)$, т. е. в примерах выше, хотя мы и не говорили об этом явно, мы каждый раз имели дело со спектрами $\tilde{h}(\omega - \omega_*)$, вычисленными на частотных сетках $\omega_j = +2\pi j/T$. В этой связи рассмотрим спектры $\tilde{h}(\omega)$ для разных оконных функций h более подробно и систематически.

Начнём с прямоугольного окна (67). Спектр прямоугольной (ступенчатой) функции ширины T есть

$$\left|\tilde{h}(\omega)\right| = T \operatorname{sinc} \frac{\omega T}{2}, \tag{71}$$

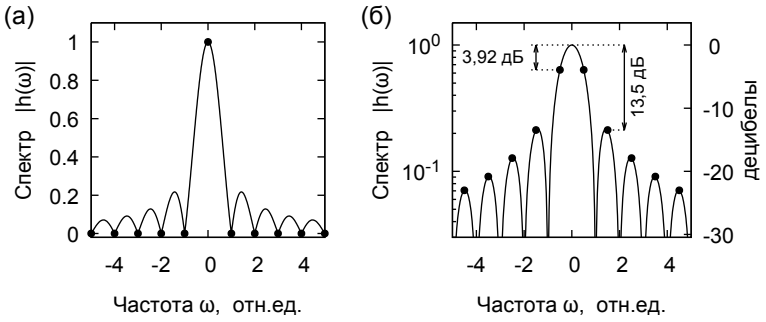

Рис. 8. Спектр прямоугольной оконной функции $|\tilde{h}(\omega)|$ (71) в линейном (а) и логарифмическом (б) масштабе; круглыми маркерами показаны примеры расположения узлов обратной сетки

где $\operatorname{sinc} x \equiv \sin(x)/x$. График спектральной функции прямоугольного окна $\tilde{h}(\omega)$ (71) показан на рис. 8 в линейном (а) и логарифмическом (б) масштабе. По горизонтальной оси в обоих случаях отложено отношение частоты ω к шагу обратной сетки $2\pi/T$.

Заметим, что спектр (71) является непрерывной функцией ω, тогда как в результате дискретного преобразования Фурье мы получим таблицу значений $\tilde{h}_j = \tilde{h}(\omega_j - \omega_*)$. Узлы сетки $\omega_j = 2\pi j/T$ могут быть по-разному расположены относительно нулей функции sinc. Так, если мы вычисляем спектр гармонического сигнала с «целой» частотой $\omega_p = 2\pi p/T$, то узлы обратной сетки будут совпадать с нулями функции sinc (за исключением узла с индексом p, который будет находиться в максимуме функции sinc), см. рис. 8 (а). Если рассмотреть гармонический сигнал на «полуцелой» частоте $\omega_{p+0,5} = 2\pi(p + \frac{1}{2})/T$, то расположение узлов обратной сетки будет таким, как показано на рис. 8 (б). Таким образом, дискретизация спектра (71) на сетках, сдвинутых друг относительно друга на половину шага, будет давать качественно отличающиеся результаты, см. рис. 4 на с. 41.

Обратим внимание на количественные характеристики спектра, показанные на рис. 8 (б). Изменение частоты ω_* сигнала $f(t) = \exp(i\omega_* t)$ может приводить к изменению наблюдаемой высоты линий в дискретном преобразовании Фурье на 3,92 дБ[14], см. рис. 4. Помимо основного лепестка в центре, спектры оконных функций содержат также *побочные* лепестки, что приводит к появлению шумовой подложки на реги-

[14] Децибелы здесь, как это обычно принято, характеризуют отношение *мощностей*: $k_{dB} = 10\log_{10}(P_1/P_2)$. Для вычисления отношения *амплитуд* в спектре на рис. 8 использована формула $k_{dB} = 20\log_{10}(A_1/A_2)$.

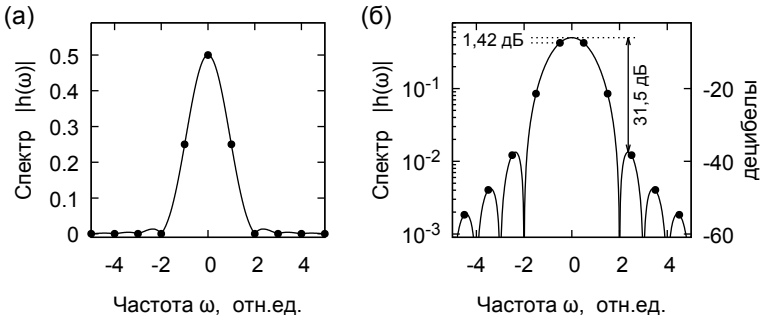

Рис. 9. Спектр оконной функции Ханна $|\tilde{h}(\omega)|$ (72) в линейном (а) и логарифмическом (б) масштабе

стрируемых спектрах. В случае прямоугольного окна отношение высот главного и первого побочного лепестка составляет всего 13,5 дБ.

Аналогично для оконной функции Ханна можем записать:

$$h(t) = \frac{1}{2}\left(1 - \cos\frac{2\pi t}{T}\right), \quad |\tilde{h}(\omega)| = 2\pi^2 T \left|\frac{\operatorname{sinc}(\omega T/2)}{(\omega T)^2 - 4\pi^2}\right|. \qquad (72)$$

График спектральной функции (72) показан на рис. 9 в линейном и логарифмическом масштабе. По сравнению со спектром прямоугольного окна (см. рис. 8) видно несколько отличий: соотношение высоты главного и побочного пика в спектре возросло с 13,5 до 31,5 дБ, что позволяет значительно уменьшить эффект частокола при использовании окна Ханна. Однако при этом мы наблюдаем также увеличение ширины центрального пика, что приводит к снижению разрешения в спектре, содержащем лишь «целые» частоты (период которых делит нацело ширину сетки).

Таким образом, мы видим, что использование оконных функций является компромиссным решением. Использование любой оконной функции подразумевает внесение некоторых искажений в регистрируемый спектр сигнала, а выбор той или иной оконной функции обусловлен спецификой конкретной задачи и зависит от допустимости тех или иных искажений.

2.10. Быстрое преобразование Фурье

Преобразование Фурье играет чрезвычайно важную роль и повсеместно используется для цифровой обработки сигналов, в том числе

потоков аудио- и видеоданных, ставших неотъемлемой частью современной жизни. Необходимость больших объёмов вычислений (многие из которых требуется выполнять в реальном масштабе времени) диктует спрос на экономичные алгоритмы и их эффективные программные и аппаратные реализации. Заметим, что возникновение практического интереса к эффективным вычислительным алгоритмам возникло за полвека до появления Youtube. Актуальные задачи тех дней включали анализ сейсмоданных для обнаружения испытаний атомного оружия вероятным противником.

Вычисление дискретного преобразование Фурье непосредственно по определению (56) на с. 33 требует $\mathcal{O}(n^2)$ арифметических операций. Алгоритмы, позволяющие решить данную задачу за меньшее число шагов, принято называть *быстрым преобразованием Фурье* (БПФ). Начало активного развития данного направления связано с публикацией работы [A4] в 1965 году, хотя, как выяснилось уже после выхода статьи, базовые идеи были высказаны значительно раньше.

К настоящему времени разработано значительное количество алгоритмов быстрого преобразования Фурье (и ещё большее количество их разнообразных описаний). Для наших целей, однако, будет вполне достаточным поверхностного знакомства с основными идеями, позволяющими достичь значительной экономии при выполнении дискретного преобразования Фурье, а также с одной из наиболее эффективных универсальных программных реализаций, о которой пойдёт речь в п. 2.11.

Достаточно компактная формальная запись формул, позволяющих уменьшить число арифметических операций за счёт перегруппировки слагаемых в (56), может быть найдена в пионерской работе [A4][15]. Однако для наших целей представляется более полезным и интересным начать рассмотрение основной идеи алгоритмов БПФ, проведя аналогию с вычислением полиномов, как это сделано в замечательной монографии [A6, гл. 2].

2.10.1. БПФ и вычисление полиномов

Заметим, что матрица $n \times n$ дискретного преобразования Фурье $\Lambda^{(n)}_{jk} \propto \omega_n^{jk}$, где $\omega_n \equiv \exp(2\pi i/n)$, с точностью до коэффициента пропорциональности совпадает с матрицей Вандермонда $\Delta = x_j^k$ при $x_j = \omega_n^j$. Другими словами, выполнение дискретного преобразования Фурье для вектора $(a_0, a_1, \ldots, a_{n-1})$ в соответствии с выражением

[15] Бесплатную копию можно найти в Интернет, используя сервис Google Scholar.

$$\begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & \omega_n^1 & \dots & \omega_n^{n-2} & \omega_n^{n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & \omega_n^{n-2} & \dots & \omega_n^{(n-2)(n-2)} & \omega_n^{(n-2)(n-1)} \\ 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)(n-2)} & \omega_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-2} \\ a_{n-1} \end{pmatrix}$$

эквивалентно вычислению значений полинома[16]

$$P_n(x) = a_0 + a_1 x^1 + \ldots + a_{n-1} x^{n-1}$$

в n точках — комплексных корнях n-й степени из единицы:

$$\{x_j\} = \{\omega_n^j\}, \quad j = 0, 1, \ldots, n-1, \quad \omega_n = \exp(2\pi i/n). \tag{73}$$

Непосредственное вычисление значений $P_n(\omega_n^j)$ полинома потребует приблизительно $2n^2$ арифметических операций над комплексными числами (полагаем, что таблица значений ω_n^j построена предварительно и используется для многократного вычисления полиномов с разными a_j). Как добиться экономии при вычислении значений полинома? Для простоты ограничимся рассмотрением случая $n = 2^r$, где r — целое число. Разобьём полином P_n на сумму двух многочленов разной чётности:

$$\begin{aligned} P_n(x) &= F_{n/2}(x^2) + x\, G_{n/2}(x^2) = \\ &= \left(a_0 + a_2 x^2 + \ldots + a_{n-2} x^{n-2}\right) + \\ &+ \left(a_1 + a_3 x^2 + \ldots + a_{n-1} x^{n-2}\right) \cdot x. \end{aligned} \tag{74}$$

Несложно увидеть (рис. 10), что при чётных n для каждой точки x_j в наборе (73) присутствует также симметрично расположенная точка $x_k = -x_j, \quad k = (j + n/2) \mod n$. Следовательно, вычисление значений полинома $P_n(\omega_n^j)$ можно оптимизировать, разбив набор точек (73) на $n/2$ пар: $\pm x_0, \pm x_1, \ldots, \pm x_{\frac{n}{2}-1}$. В каждой паре для нахождения $P(\pm x)$ необходимо вычислить значения полиномов $F_{n/2}$ и $G_{n/2}$:

$$\begin{aligned} P_n(x) &= F_{n/2}(x^2) + x \cdot G_{n/2}(x^2), \\ P_n(-x) &= F_{n/2}(x^2) - x \cdot G_{n/2}(x^2). \end{aligned} \tag{75}$$

Использование соотношений (75) позволяет получить ответ всего за $2 \times 2(\frac{n}{2})^2 + \frac{3}{2}n \approx n^2$ операций, что почти вдвое меньше первоначальной оценки $2n^2$.

[16] В данной задаче в качестве нижнего индекса n в обозначении полиномов P_n удобнее использовать количество полиномиальных коэффициентов (размерность пространства), а не степень полиномов.

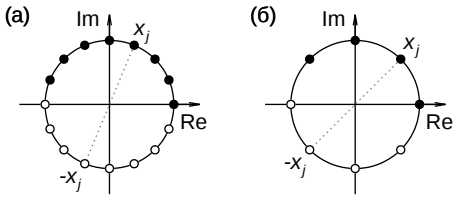

Рис. 10. Симметричное расположение точек $x_j = \omega_n^j$ при $n = 2^r$ на примере (а) $n = 16$, (б) $n = 8$

Однако самое замечательное свойство использованного подхода состоит в том, что его можно применять многократно, что позволяет не просто уменьшить коэффициент в $\mathcal{O}(n^2)$, но даже изменить асимптотику числа операций. Действительно, формулы (75) сводят задачу о вычислении полинома P_n в точках $x_j = \omega_n^j$ к двум задачам о вычислении многочленов $F_{n/2}$ и $G_{n/2}$ в точках $x_j^2 = \omega_n^{2j}$. Поскольку $\omega_n^2 = \exp(2\pi i/(n/2)) = \omega_{n/2}$, легко заметить, что использование формул (75) приводит нас к условиям первоначальной задачи с заменой n на $n' = n/2 = 2^{r-1}$, ср. рис. 10 (а) и (б). Это позволяет использовать данный подход многократно, на каждом шаге удваивая число задач и уменьшая вдвое размерность пространства.

Посчитаем полное число арифметических операций, которые необходимо совершить для вычисления P_n в $n = 2^r$ точках $\{x_j\}$ (73) при рекурсивном использовании формул (75). Поскольку на каждом шаге рекурсии число полиномов удваивается, а число коэффициентов уменьшается вдвое, на k-м шаге ($k = 0, 1, \ldots, r-1$) будем иметь 2^k полиномов с $n_k = n/2^k$ коэффициентами в каждом. На последнем шаге алгоритма будем иметь $k = r - 1$, $n_{r-1} = 2$, $\omega_2 = e^{2\pi i/2} = -1$, а для вычисления каждого из 2^{r-1} полиномов P_2 в точках $\{\omega_2^0, \omega_2^1\}$ потребуется две арифметических операции:

$$\begin{aligned} P_2(+1) &= a_0 + a_1, \\ P_2(-1) &= a_0 - a_1. \end{aligned} \tag{76}$$

На остальных шагах ($k = 0, \ldots, r-2$) в соответствии с выражением (75) потребуется 3 операции над полиномами F и G (умножение $x \cdot G$ и вычисление суммы и разности $F \pm x \cdot G$) для каждого из 2^k полиномов в каждой из $n_k/2 = n/2^{k+1}$ пар точек. Следовательно, всего на k-м шаге алгоритма необходимо сделать $3 \times 2^k \times (n/2^{k+1}) = \frac{3}{2}n$ арифметических операций. Учитывая, что алгоритм включает $(r-1)$ таких шагов плюс

самый последний шаг при $k = r - 1$ (76), получаем для полного числа операций $\mathcal{O}(n \times r) = \mathcal{O}(n \log n)$.

2.10.2. Запись БПФ через матрицы

Сформулируем описанный выше алгоритм также на языке матриц, более привычном в контексте данной задачи. Запишем дискретное преобразование Фурье n-мерного вектора $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ в матричной форме, по-прежнему полагая $n = 2^r$:

$$\mathbf{b} = \Lambda^{(n)}\mathbf{a}, \qquad \text{где} \quad \Lambda^{(n)}_{jk} = \omega_n^{jk}, \quad \omega_n = \exp(2\pi i/n). \tag{77}$$

Для использования симметрии матрицы $\Lambda^{(n)}$ дискретного преобразования Фурье перейдём теперь к другому базису в пространстве исходных данных, изменив порядок элементов. Расположим вначале $n/2$ чётных компонент вектора $\mathbf{a}$, а затем оставшиеся $n/2$ нечётных, что является аналогом записи полинома P_n в виде суммы чётной и нечётной частей (74). Изменение порядка элементов базиса приведёт к соответствующей перестановке столбцов матрицы $\Lambda^{(n)}$. Записав матрицу $\Lambda^{(n)}$ в новом базисе в блочном виде (78), получим, что блоки $A^{(n/2)}$ и $C^{(n/2)}$ будут включать чётные столбцы $\Lambda^{(n)}$, а блоки $B^{(n/2)}$ и $D^{(n/2)}$ — нечётные:

$$\begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{pmatrix} = \left(\begin{array}{c|c} A^{(n/2)} & B^{(n/2)} \\ \hline C^{(n/2)} & D^{(n/2)} \end{array}\right) \begin{pmatrix} a_0 \\ a_2 \\ \vdots \\ a_{n-3} \\ a_{n-1} \end{pmatrix} \tag{78}$$

Для матричных элементов блока $A^{(n/2)}$ имеем:

$$A^{(n/2)}_{jk} = \Lambda^{(n)}_{j,2k} = \omega_n^{2jk} = \omega_{n/2}^{jk}, \quad \text{где} \quad j,\, k = 0, \ldots, \frac{n}{2} - 1. \tag{79}$$

Из определения ω_n (77) следует, что $\omega_n^{n/2} = -1$, а $\omega_n^n = 1$, что позволяет выразить три других блока через $A^{(n/2)}$:

$$\begin{aligned} C^{(n/2)}_{jk} &= \Lambda^{(n)}_{j+n/2,2k} = \omega_n^{2(j+n/2)k} = A^{(n/2)}_{jk}, \\ B^{(n/2)}_{jk} &= \Lambda^{(n)}_{j,2k+1} = \omega_n^{j\cdot(2k+1)} = \omega_n^j A^{(n/2)}_{jk}, \\ D^{(n/2)}_{jk} &= \Lambda^{(n)}_{j+n/2,2k+1} = \omega_n^{(j+n/2)(2k+1)} = -\omega_n^j A^{(n/2)}_{jk}. \end{aligned} \tag{80}$$

Формулы (78–80) сводят задачу о дискретном преобразовании Фурье n-мерного вектора $\mathbf{a}$ к двум задачам вдвое меньшей размерности

для чётных и нечётных компонент вектора **a**, что даёт матричное описание рекурсивного алгоритма, рассмотренного выше в терминах полиномов.

Несложно построить альтернативную версию рассмотренного алгоритма без использования рекурсии и показать, что быстрое преобразование Фурье позволяет сократить число операций без использования дополнительной памяти для хранения промежуточных вычислений. Однако это выходит за рамки поставленных нами целей, в связи с чем мы направляем интересующихся читателей к монографии [A6].

2.10.3. Общий случай составных n

Выше мы рассматривали алгоритм БПФ для частного случая массивов длины $n = 2^r$. Указанное ограничение на размер таблиц данных хотя и позволяет достичь наибольшей экономии числа операций, но не является обязательным. Чтобы построить более общий алгоритм БПФ заметим, что экономия числа операций достигается за счёт использования симметрии расположения комплексных корней из единицы $\{x_j\} = \{\omega_n^j\}$. В рассмотренном выше частном случае $n = 2^r$ мы воспользовались симметрией относительно замены $x \to -x$, что эквивалентно домножению x на $e^{i\pi}$, или повороту комплексной плоскости на π. Аналогичным образом можно использовать симметрию относительно поворота на угол $2\pi/m$, где m — делитель числа точек n. Рассмотрим данную мысль на примере $m = 3$. Полагая n кратным 3 и действуя по аналогии с (74), разобьём полином P_n на сумму трёх многочленов, каждый из которых является собственной функцией преобразования $x \to xe^{2\pi i/3}$ с собственным значением 1 либо $e^{\pm 2\pi i/3}$:

$$\begin{aligned} P_n(x) &= F_{n/3}(x^3) + x\,G_{n/3}(x^3) + x^2\,H_{n/3}(x^3) = \\ &= \left(a_0 + a_3x^3 + \ldots + a_{n-3}x^{n-3}\right) + \\ &+ \left(a_1 + a_4x^3 + \ldots + a_{n-2}x^{n-3}\right)\cdot x\ + \\ &+ \left(a_2 + a_5x^3 + \ldots + a_{n-1}x^{n-3}\right)\cdot x^2. \end{aligned}$$

Поскольку $n \bmod 3 = 0$, для каждой точки x_j в наборе (73) присутствует также симметрично расположенные точки $x_k = x_j e^{\pm 2\pi i/3}$, $k = (j \pm n/3) \bmod n$. Следовательно, вычисление значений полинома $P_n(\omega_n^j)$ можно оптимизировать, если разбить набор точек (73) на $n/3$ троек. В каждой тройке для нахождения P_n необходимо вычислить значения полиномов $F_{n/3}$, $G_{n/3}$ и $H_{n/3}$:

$$
\begin{aligned}
P_n(x) &= F_{n/3}(x^3) + \quad x \cdot G_{n/3}(x^3) + \quad x^2 \cdot H_{n/3}(x^3), \\
P_n(xe^{\frac{2\pi i}{3}}) &= F_{n/3}(x^3) + e^{\frac{2\pi i}{3}} x \cdot G_{n/3}(x^3) + e^{\frac{4\pi i}{3}} x^2 \cdot H_{n/3}(x^3), \qquad (81) \\
P_n(xe^{\frac{4\pi i}{3}}) &= F_{n/3}(x^3) + e^{\frac{4\pi i}{3}} x \cdot G_{n/3}(x^3) + e^{\frac{2\pi i}{3}} x^2 \cdot H_{n/3}(x^3).
\end{aligned}
$$

Использование соотношений (81) позволяет получить ответ всего за $3 \times 2(\frac{n}{3})^2 + 12\frac{n}{3} \approx \frac{2}{3}n^2$ операций (вместо $2n^2$ при непосредственном вычислении по формулам (56) на с. 33).

Аналогичные рассуждения можно произвести для произвольного составного числа точек n. Краткое формальное рассмотрение общего случая $n = r_1 \cdot r_2$ можно найти в статье [A4]. Перегруппировка слагаемых в сумме позволяет свести исходную задачу о выполнении дискретного преобразования Фурье массива длины n к вычислению суммы из r_1 слагаемых, каждое из которых есть результат БПФ от подмножества исходного массива длины r_2.

2.10.4. Случай простого n

Работа описанного выше алгоритма основана на разделении задачи для n-точечного дискретного преобразования Фурье на последовательность подзадач с кратно меньшим числом точек. Каждый шаг деления n даёт выигрыш в быстродействии, так что общая экономия будет увеличиваться с ростом числа шагов. Последнее определяется, очевидно, числом сомножителей, на которые раскладывается длина n исходного массива. Можно ли достичь какого-либо ускорения в случае, если n — простое число, т. е. непредставимо в виде произведения целочисленных сомножителей? Через три года после выхода работы Кули и Тьюки было показано [A7], что в случае простых n дискретное преобразование Фурье может быть сведено к вычислению циклической корреляции (либо свёртки) двух последовательностей длины $(n-1)$. Данная задача может быть решена с помощью дискретных преобразований Фурье[17]. Поскольку $(n-1)$ не является простым числом (предполагаем, что n простое и $n > 3$), задача сводится к рассмотренному выше алгоритму быстрого преобразования Фурье, что позволяет решить её за $\mathcal{O}(n \log n)$ арифметических операций, хотя коэффициент пропорциональности может быть значительно больше, чем в случае составных n и тем более $n = 2^r$.

[17] Фурье-образ от свёртки $f * g$ равен произведению Фурье-образов f и g.

Рассмотрим идею данного алгоритма более подробно. Данный параграф предназначен лишь для самых увлечённых и любознательных читателей, тогда как при первом знакомстве с предметом рекомендуется переходить к п. 2.11. Для наглядности, помимо общих формул, выпишем также матрицы для частного случая $n = 5$. Прежде всего, обратим внимание, что все элементы первой строки и первого столбца матрицы дискретного преобразования Фурье $\Lambda_{jk} = \exp(2\pi ijk)$ равны 1 (см. матрицу на с. 51). В этой связи для коэффициента Фурье с индексом 0 имеем:

$$b_0 = \sum_{k=0}^{n-1} a_k. \tag{82}$$

Для остальных коэффициентов можем записать:

$$b_j = a_0 + \sum_{k=1}^{n-1} a_k \exp\left(\frac{2\pi i}{n} jk\right), \quad j = 1, \ldots, n-1. \tag{83}$$

В частном случае $n = 5$, обозначая для краткости $\omega \equiv \exp(2\pi i/5)$, имеем:

$$\begin{pmatrix} b_1 - a_0 \\ b_2 - a_0 \\ b_3 - a_0 \\ b_4 - a_0 \end{pmatrix} = \begin{pmatrix} \omega^1 & \omega^2 & \omega^3 & \omega^4 \\ \omega^2 & \omega^4 & \omega^1 & \omega^3 \\ \omega^3 & \omega^1 & \omega^4 & \omega^2 \\ \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}.$$

Выполним перестановку строк и столбцов полученной матрицы, заменив индексы j и k в (83) так, чтобы получилось взаимно однозначное отображение $\alpha = 0, 1, \ldots, n-2$ на $j = 1, 2, \ldots, n-1$ и $\beta = 0, 1, \ldots, n-2$ на $k = 1, 2, \ldots, n-1$:

$$\begin{aligned} j &= g^\alpha \mod n, \quad \alpha = 0, \ldots, n-2, \\ k &= g^\beta \mod n, \quad \beta = 0, \ldots, n-2. \end{aligned} \tag{84}$$

(Числа g, обеспечивающие указанное взаимно однозначное отображение, называют *примитивными корнями*.) Например, при $n = 5$ можно выбрать $g = 2$, что даст следующее отображение индексов:

α, β	0	1	2	3
j, k	1	2	4	3

Выполнив замену индексов (84) в выражении (83), получим:

$$\begin{aligned} &b_{g^\alpha \bmod n} - a_0 = \sum_{\beta=0}^{n-2} a_{g^\beta \bmod n} \exp\left(\frac{2\pi i}{n} \cdot g^{\alpha+\beta} \bmod n\right), \\ &\alpha = 0, \ldots, n-2. \end{aligned} \tag{85}$$

Выражение (85) задаёт произведение некоторой матрицы и вектора $\mathbf{a}$, компоненты которого переставлены в соответствии с (84). Обратим внимание на структуру матрицы: заменяя в (85) $\alpha \to \alpha + 1$, несложно увидеть, что каждая следующая строка содержит те же элементы, что и предыдущая, со сдвигом столбцов на 1 ($\beta \to \beta - 1$). Сдвиг является циклическим ввиду того, что $g^{\alpha+\beta}$ входит в (85) по модулю n. В частном случае $n = 5$ имеем:

$$\begin{pmatrix} b_1 - a_0 \\ b_2 - a_0 \\ b_4 - a_0 \\ b_3 - a_0 \end{pmatrix} = \begin{pmatrix} \omega^1 & \omega^2 & \omega^4 & \omega^3 \\ \omega^2 & \omega^4 & \omega^3 & \omega^1 \\ \omega^4 & \omega^3 & \omega^1 & \omega^2 \\ \omega^3 & \omega^1 & \omega^2 & \omega^4 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_4 \\ a_3 \end{pmatrix}.$$

Полученное выражение носит название *корреляции* пары векторов $(\omega^1, \omega^2, \omega^4, \omega^3)$ и (a_1, a_2, a_4, a_3), либо *свёртки* вектора $(\omega^3, \omega^4, \omega^2, \omega^1)$ с (a_1, a_2, a_4, a_3). Как уже отмечалось в начале данного параграфа, данная операция может быть сведена к обратному дискретному преобразованию Фурье от произведения Фурье-образов сворачиваемых векторов. Всего требуется три преобразования Фурье, однако одно из них — от вектора $(\omega^3, \omega^4, \omega^2, \omega^1)$ — может быть выполнено предварительно, сохранено в памяти и использовано многократно для различных $\mathbf{a}$. Поскольку свёртка вычисляется в пространстве $\mathbb{C}^{n-1}$, а число $(n-1)$ является составным для простых $n \geq 5$, можно использовать описанные в пп. 2.10.1-2.10.3 алгоритмы быстрого преобразования Фурье для составных n.

Таким образом, мы свели задачу о вычислении дискретного преобразования Фурье для простых n к использованию алгоритмов быстрого преобразования Фурье плюс $\mathcal{O}(n)$ дополнительных операций, связанных с компонентами b_0 и a_0, см. (82) и (85). Совместное использование алгоритмов БПФ для простых и составных n формально позволяет выполнить дискретное преобразование Фурье для произвольного n за $\mathcal{O}(n \log n)$ арифметических операций. Для желающих изучить вопрос более глубоко, рекомендуем обратиться к оригинальным работам и монографии [A8].

2.11. Библиотека FFTW

В заключение познакомимся с одной из наиболее эффективных кросс-платформенных библиотек для выполнения дискретного преобразования Фурье — FFTW[18] [A9].

[18] От англ. Fastest Fourier Transform in the West, самое быстрое преобразование Фурье на Западе.

Библиотека FFTW является бесплатной (в том числе для коммерческого использования), распространяется по лицензии GNU GPL и имеет целый ряд достоинств, в числе которых:

- высокая скорость работы на различных аппаратных платформах;
- возможность дополнительного повышения быстродействия за счёт использования различных наборов векторных (SIMD) инструкций и параллельных вычислений в системах с симметричной мультипроцессорностью (SMP) и массивно-параллельной архитектурой (MPP);
- выполнение одно- и многомерных преобразований Фурье;
- возможность работы с массивами произвольной длины (хотя использование таблиц длины $n = 2^a$ обеспечивает лучшее быстродействие);
- возможность использования как комплексных, так и вещественных входных / выходных данных;
- интерфейс на языках Си и Фортран;
- наличие подробной, хорошо структурированной документации с примерами использования.

Указанные достоинства библиотеки FFTW обусловливают её широкое применение. Использование библиотеки FFTW вместо самописного программного кода в некоторых случаях может дать ускорение вплоть до порядка величины даже без использования параллельных вычислений в SMP/MPP системах. В данном параграфе мы кратко познакомимся с основными принципами использования FFTW.

2.11.1. Установка

Для установки библиотеки необходимо скачать дистрибутив FFTW со страницы Download официального сайта [A9].

Пользователям операционной системы Microsoft Windows следует перейти с основной страницы загрузок по ссылке Go here for Windows и скачать архив с уже готовыми к использованию dll-файлами[19] библиотеки FFTW. Обратим внимание на наличие в архиве сразу трёх dll-файлов: `libfftw3-3.dll`, `libfftw3f-3.dll` и `libfftw3l-3.dll`, содержащих реализацию для типа данных `double` (файл без суффикса

[19] Сокращение от *Dynamic Link Library*, динамически подключаемая библиотека в операционных системах семейства MS Windows.

в имени), `float` (файл с суффиксом `f`) и `long double` (файл с суффиксом `l`). Требуемый программе dll-файл необходимо разместить в рабочем каталоге рядом с создаваемым exe-файлом либо в одном из стандартных каталогов[20], используемых для размещения динамически подключаемых библиотек (`C:\Windows`, `C:\Windows\System32`).

Пользователям Microsoft Visual Studio для сборки программы, использующей библиотеку FFTW, необходимо дополнительно создать lib-файлы с помощью команд[21]

```
lib /def:libfftw3-3.def
lib /def:libfftw3f-3.def
lib /def:libfftw3l-3.def
```

Для запуска программы `lib.exe` нужно открыть консоль разработчика Microsoft Visual Studio (Developer Command Prompt for VS). Ярлык для запуска командной строки находится в папке Visual Studio Tools в меню программ. Также можно перейти в каталог с установленной Microsoft Visual Studio на жёстком диске и запустить файл `Common7\Tools\VsDevCmd.bat`. Более подробную информацию о способе запуска консоли в различных версиях операционной системы Microsoft Windows можно найти в статье «Командная строка разработчика для Visual Studio» на сайте msdn.microsoft.com.

Для включения библиотеки в проект Microsoft Visual Studio необходимо в свойствах проекта выбрать в левом верхнем углу диалогового окна Configuration: All Configurations, после чего на вкладке Linker ▷ Input вписать нужный lib-файл в поле Additional Dependencies. Путь к lib-файлу можно указать в поле Additional Library Directories на вкладке Linker ▷ General. Путь к h-файлу следует указать в поле Additional Include Directories на вкладке C/C++ ▷ General.

Если для сборки программ с FFTW под Windows используется компилятор MinGW, то dll-файл будет подключён компоновщиком непосредственно (без использования дополнительных lib-файлов). При этом путь к h-файлу указывается на стадии компиляции с использованием опции `-I`.

Пользователям операционных систем семейства Unix необходимо скачать исходные коды библиотеки FFTW, после чего выполнить установку, состоящую из трёх шагов:

[20] Также возможно разместить dll-файлы в произвольном месте, добавив путь к ним в переменную PATH.

[21] Предполагается, что dll-файлы лежат в текущем рабочем каталоге, в котором открыта консоль разработчика; если это не так, предварительно нужно либо сменить текущий рабочий каталог с помощью команды cd, либо скопировать dll-файлы в каталог по умолчанию.

```
./configure
make
make install
```

В качестве опций скрипта `configure` можно указать используемый компилятор (например, `CC=gcc`), необходимость установки библиотеки для типа `float`, `long double` либо нестандартного `__float128` (`--enable-float`, `--enable-long-double` и `--enable-quad-precision`[22] соответственно), возможность использования SIMD-инструкций процессора (`--enable-sse2`, `--enable-avx`, `--enable-avx2` и т. п.). В случае отсутствия прав администратора в системе необходимо использовать опцию `--prefix=`*путь_для_установки*; в противном случае следует выполнять установку с использованием `sudo`. Для подключения библиотеки на стадии компоновки программы в `gcc`/`g++` используется опция `-lfftw3`, для указания пути к библиотеке — опция `-L`*путь*.

Более подробную информацию об опциях скрипта `configure` при установке в Unix-системах, а также о способах сборки библиотеки из исходных кодов под Windows можно найти на официальном сайте библиотеки [A9] в разделе «Installation and Customization».

2.11.2. Использование

Общая структура программы с использованием библиотеки FFTW состоит из следующих основных частей: подключение `fftw3.h` наряду с другими заголовочными файлами, инициализация (выделение памяти, создание плана преобразований), основная часть (выполнение расчётов с использованием быстрого преобразования Фурье) и финализация (освобождение памяти):

```
#include <fftw3.h>
#include <complex>
typedef std::complex<double> Cplx;
...
{
    //размер таблиц с данными (число узлов сетки):
    const int n = 1 << 10;          //2**10 == 1024
    //инициализация:
    //выделяем память под исходные данные и Фурье-образа:
    Cplx *in  = (Cplx*) fftw_malloc(sizeof(Cplx) * n);
```

[22] Доступно при использовании компилятора `gcc` начиная с версии 4.6, при этом на стадии компоновки программы нужно подключать библиотеки `-lfftw3q -lquadmath -lm`.

```
    Cplx *out = (Cplx*) fftw_malloc(sizeof(Cplx) * n);
    //создаем план для выполнения одномерного DFT:
    fftw_plan p = fftw_plan_dft_1d(n, (fftw_complex*)in,
        (fftw_complex*)out, FFTW_FORWARD, FFTW_ESTIMATE);
    ...
    //основная часть программы: выполняем DFT
    fftw_execute(p);  //необходимое число раз
    ...
    //финализация: освобождаем память:
    fftw_destroy_plan(p);
    fftw_free(in);
    fftw_free(out);
}
```

Хотя библиотека FFTW определяет собственный тип для записи комплексных чисел (`typedef double fftw_complex[2]`), рекомендуется использовать стандартные типы данных: `complex` в программах на языке C (стандарт C99, `#include <complex.h>`) либо шаблон `std::complex<T>` в C++ (`#include <complex>`), см. пример выше. Для стандартных типов комплексных чисел определены арифметические операции и основные математические функции, что существенно упрощает разработку и чтение программного кода. Поскольку стандартные типы обладают битовой совместимостью с `fftw_complex`, для вызова функций библиотеки FFTW достаточно выполнять приведение типа указателей, как это сделано в примере выше при вызове функции `fftw_plan_dft_1d`.

Для выделения памяти разработчики FFTW рекомендуют использовать функцию `fftw_malloc` (и комплементарную к ней `fftw_free` для освобождения памяти) вместо стандартных средств C и C++, таких как `malloc` / `free`, `new[]` / `delete[]`, `std::vector<T>` и статических массивов фиксированного размера. Хотя использование «обычных» средств выделения памяти также возможно, вызов `fftw_malloc` является предпочтительным, обеспечивая выравнивание выделяемых фрагментов в памяти, что необходимо для использования SIMD-инструкций[23] процессора и заметного повышения скорости работы FFTW. Напомним, что выделение памяти занимает достаточно много машинного времени, поэтому рекомендуется выделять память в на-

[23] *Single Instruction, Multiple Data* — набор специальных команд (инструкций), поддерживаемых рядом современных процессоров, позволяющих выполнить одну операцию сразу над несколькими числами. Для включения поддержки различных наборов SIMD-инструкций при установке FFTW необходимо указывать соответствующие опции, см. п. 2.11.1.

чале работы программы, затем многократно использовать её в цикле численного моделирования и освобождать только перед выходом из программы.

Помимо выделения памяти программный код инициализации также создаёт *план* преобразования Фурье (см. вызов `fftw_plan_dft_1d` в приведённом выше примере). Определяемый библиотекой FFTW тип `fftw_plan` представляет собой указатель на структуру данных, содержащую всю необходимую информацию для выполнения дискретного преобразования Фурье, включая указатели на исходную таблицу данных и массив-получатель (указатели `in` и `out` в примере выше). Поскольку вся необходимая информация сохранена в плане, для выполнения дискретного преобразования Фурье в программе достаточно вызвать функцию `fftw_execute`, передав ей единственный параметр — план преобразования.

Однако необходимость создания плана преобразования Фурье продиктована в первую очередь стремлением не к лаконичности кода, но к высокой скорости его выполнения. Построение плана призвано определить наиболее эффективный алгоритм дискретного преобразования Фурье при использовании заданных параметров (таких как тип преобразования, размерность пространства, число узлов сетки) и конкретного аппаратного обеспечения, на котором выполняется программа.

Библиотека FFTW включает в себя широкий набор функций для создания планов одно- и многомерных дискретных преобразований Фурье над комплексным и вещественным набором значений. Для целей нашего курса будет достаточно ограничиться рассмотрением лишь одной функции из данного набора:

```
fftw_plan fftw_plan_dft_1d(int n,
                           fftw_complex *in, fftw_complex *out,
                           int sign, unsigned flags);
```

Обратим внимание на стиль именования функций FFTW. Название функции начинается с префикса `fftw_`, указывающего на её принадлежность к библиотеке и тип данных. Так, при использовании типа данных одинарной точности `float` префикс должен быть заменён на `fftwf_`, для `long double` — на `fftwl_`, тогда как нестандартный тип четверной точности `__float128` соответствует префиксу `fftwq_`. Последующая часть имени состоит из отдельных слов и сокращений, раскрывающих решаемую задачу. В данном случае это `plan_dft_1d` — создание плана DFT (дискретного преобразования Фурье) в одномерном случае.

Первый параметр (`int n`) функции `fftw_plan_dft_1d` определяет размер входного и выходного массивов `*in` и `*out` комплексных значений, второй и третий параметры — указатели на первые элементы этих массивов. Очевидно, что на этапе создания плана память под массивы должна быть уже выделена, при этом какая-либо инициализация (очистка памяти и заполнение массивов) не требуется. Следует также подчеркнуть, что размер массива `n` может быть любым целым числом[24]. Если решаемая задача не накладывает ограничений на количество узлов сетки, налучшим выбором с точки зрения повышения быстродействия выполнения преобразования Фурье будет степень двойки, $n = 2^a$. Хорошие результаты также получаются для $n = 2^a 3^b 5^c 7^d 11^e 13^f$ при $e + f = 0$ или 1 и произвольных значениях показателей $a, \ldots, d$, однако даже в самом худшем случае — при использовании простых n — реализованные в FFTW алгоритмы требуют $\mathcal{O}(n \log n)$ арифметических операций.

Четвёртый параметр (`int sign`) определяет знак в показателе экспоненты (см. определение (56) на с. 33). Для повышения читаемости кода, написанного с использованием FFTW, в файле `fftw3.h` определены константы `FFTW_FORWARD` и `FFTW_BACKWARD`, равные соответственно -1 и $+1$.

Последний, пятый параметр `unsigned flags` представляет собой набор флагов (переключателей), определяющих выбор алгоритма, который будет использоваться для выполнения преобразования Фурье при последующих вызовах `fftw_execute`. К числу таких переключателей относится выбор между необходимостью сохранения значений во входном массиве (`FFTW_PRESERVE_INPUT`[25] и возможностью перезаписи входных значений для использования более эффективных алгоритмов (`FFTW_DESTROY_INPUT`). Другой переключатель определяет уровень оптимизации при выборе алгоритма DFT, для чего используются следующие константы: `FFTW_ESTIMATE` — быстрый выбор субоптимального алгоритма для выполнения дискретного преобразования Фурье; `FFTW_MEASURE`[26], `FFTW_PATIENT` и `FFTW_EXHAUSTIVE` — выбор оптимального алгоритма путём непосредственного измерения и сравнения времени, затраченного на выполнение преобразования Фурье различными способами. Как можно догадаться из названий, каждая следующая константа в этом ряду соответствует поиску минимума времени выполнения в более широком классе алгоритмов, что требует

[24] Разумеется, с учётом ограничений на объём доступной памяти и необходимую скорость выполнения преобразования Фурье.

[25] Используется по умолчанию для комплекснозначных преобразований.

[26] Используется по умолчанию.

бóльших затрат времени на этапе создания плана, но позволяет сократить общее время выполнения программы при использовании очень большого числа преобразований Фурье (типичный случай в численном моделировании физических задач). Для комбинации двух переключателей (флагов) используется операция «побитовое ИЛИ» — стандартный приём в языке C. Например, чтобы скомбинировать исчерпывающий способ оптимизации (`FFTW_EXHAUSTIVE`) с разрешением на перезапись входных значений (`FFTW_DESTROY_INPUT`), следует передавать в качестве пятого параметра функции `fftw_plan_dft_1d` значение `FFTW_EXHAUSTIVE | FFTW_DESTROY_INPUT`.

Заметим, что построение плана может занимать достаточно много времени, особенно при использовании больших массивов данных совместно с опциями `FFTW_PATIENT` или `FFTW_EXHAUSTIVE`. Например, для таблиц длиной $n = 2^{20} \approx 10^6$ создание плана одномерного преобразования Фурье на процессоре Intel(R) Core(TM) i7-4770 3,40 ГГц занимает более получаса при использовании уровня оптимизации `FFTW_EXHAUSTIVE` и 7,5 секунд при использовании `FFTW_MEASURE`. Для экономии времени и повышения быстродействия программы библиотека FFTW позволяет сохранить информацию о найденном оптимальном способе выполнения преобразования Фурье в файл. Данный механизм носит название *wisdom* (мудрость) и подробно описан в разделе «Words of Wisdom — Saving Plans» официальной документации FFTW [A9].

Таблица 1

	$n = 2^{10}$		$n = 2^{20}$	
	План, сек	БПФ, мкс	План, сек	БПФ, мс
`FFTW_MEASURE`	0,02	8	7,5	30
`FFTW_PATIENT`	0,3	7	600	23
`FFTW_EXHAUSTIVE`	6,5	7	1860	23

Необходимость использования механизма wisdom возникает только для сохранения планов между запусками программы. В пределах одного запуска использование «накопленной мудрости» (информации о наиболее эффективном алгоритме выполнения дискретного преобразования Фурье) происходит автоматически, без участия пользователя FFTW. При необходимости создать в программе план того же типа и с тем же числом точек n, что был создан ранее, повторный вызов `fftw_plan_dft_1d` выполняется за пренебрежимо малое время.

(Использование нескольких планов FFTW в одной программе встречается достаточно часто для выполнения прямого и обратного преобразования Фурье либо для преобразования различных величин, сохранённых в разных массивах: поскольку `fftw_plan` содержит информацию о входном и выходном массиве данных, использование нескольких пар входных и выходных массивов предполагает использование различных планов FFTW[27].)

Решив проблемы, связанные с установкой библиотеки FFTW, её подключением на этапе компиляции и компоновки программы, разобравшись с выделением памяти, созданием планов и соответствующими операциями финализации (освобождения ресурсов в конце работы программы), нам осталось приложить последнее усилие собственно для выполнения преобразования Фурье. Хотя с точки зрения программирования данный шаг тривиален и сводится к вызову функции `void fftw_execute(const fftw_plan plan)`, это место зачастую становится источником *математических* ошибок в расчётах. Типичная ошибка связана с использованием различных определений дискретного преобразования Фурье, используемых пользователем и разработчиками библиотеки. Действительно, в соответствии с выражением (56) на с. 33, возможны различные определения, отличающиеся знаками в показателе экспонент и значениями коэффициентов C_+, C_-. В FFTW преобразование со знаком «минус» в экспоненте считается прямым (`DFT_FORWARD`), со знаком «плюс» — обратным (`DFT_BACKWARD`). Константы $C_+ = C_- = 1$, так что суперпозиция прямого и обратного преобразований не даёт единицу, но эквивалентна (с точностью до погрешности вычислений) домножению таблицы значений на её размер n. Чтобы избежать трудно обнаруживаемых ошибок, перед написанием программы необходимо перепроверить свои обозначения и привести их в соответствие с использованными в FFTW.

В заключение заметим, что возможности библиотеки FFTW значительно превосходят объём данной главы, не претендующей на всеобъемлющее руководство, но призванной лишь познакомить читателя с FFTW и облегчить начало работы с этим замечательным и эффективным инструментом. Более подробную информацию, включая описание многомерных преобразований, различных типов преобразований вещественных данных, использования параллельных вычислений, а также ответы на целый ряд других вопросов по использованию FFTW можно найти на официальном сайте библиотеки [A9] в разделе Documentation.

[27] Об альтернативном решении см. раздел «New-array Execute Functions» официальной документации FFTW [A9].

Упражнения

1) Выберите параметры временно́й сетки (число узлов, ширина) для моделирования распространения импульсов длительностью 10^{-12} с при требуемом уровне точности не хуже 0,1%. Как следует изменить параметры, если в процессе распространения ожидается увеличение ширины спектра импульсов в десять раз?

2) Постройте спектр мощности гауссовского импульса $\exp(-t^2)$ и пары импульсов $\exp(-t^2) + \exp(-(t-5)^2)$, объясните вид графиков.

3) Постройте спектр мощности гармонического сигнала $f(t) = A\cos\omega_0 t$ при разных ω_0, используя прямоугольное окно и окно Ханна.

4) Какое число N членов ряда Фурье позволит аппроксимировать осцилляции температуры воздуха возле НГУ [A10] за последние 3 дня (10 дней) с точностью 5°C? 10°C? Как убывает амплитуда коэффициентов Фурье с ростом частоты? (Для повышения точности аппроксимации следует использовать сумму линейной функции и конечного ряда Фурье.)

5) Используя библиотеку FFTW, выполните дискретное преобразование Фурье от функции $f(x) = 1 + e^{ix} + ie^{2ix}$ на промежутке $0 \le x < 2\pi$. Напечатайте и проанализируйте таблицу значений функции и её спектра, используя $n = 8$. Выполните обратное преобразование Фурье, сравните с исходной таблицей значений функции.

6) Сравните время выполнения 1000 преобразований Фурье с использованием библиотеки FFTW для таблиц длиной $n = 4093$, 4095 и 4096. Объсните результат. (Для вычисления времени выполнения программного кода на языке C можно использовать функцию `clock`, объявленную в заголовочном файле `<time.h>`.)

7) Вычисляя конечную разность, найдите производную по времени $\frac{d}{dt}\int v(x,t)e^{ikx}dx$, где $v(x,t)$ — численное решение однородного уравнения теплопроводности $u_t - u_{xx} = 0$, построенное по явной схеме (89). Сравните график зависимости $(\partial_t\tilde{v}(k,t))/|\tilde{v}(k,t)|$ от k с аналитическим решением при разных значениях отношения τ/h^2.

8) Решите аналогичную задачу, используя неявную схему Кранка–Николсона (100). Какие спектральные компоненты затухают медленнее всего? Как зависит ответ от величины шага τ?

3. Уравнение теплопроводности

В данной главе мы познакомимся с основами применения разностных схем для решения уравнений в частных производных на примере параболического уравнения второго порядка, описывающего теплопроводность и диффузию. Начнём с наиболее простого случая, когда имеется всего одна пространственная координата:

$$\partial_t u = \partial_x^2 u + f(x,t). \tag{86}$$

(Здесь и далее мы будем использовать обозначения ∂_t и ∂_x для дифференциальных операторов $\frac{\partial}{\partial t}$ и $\frac{\partial}{\partial x}$ соответственно.) Если неоднородность $f(x,t)$ имеет достаточно простой вид, уравнение (86) может быть проинтегрировано аналитически. Однако при включении в (86) нелинейных членов, зависимости коэффициента теплопроводности от координат и времени, а также при поиске решения в областях сложной формы оказываются востребованными численные методы. Исключительно в целях упрощения изложения мы будем рассматривать применение численных методов на наиболее простых примерах, допускающих также точное аналитическое решение. При этом следует иметь в виду, что такое упрощение не принципиально, и рассматриваемые методы могут быть использованы в более сложных случаях, когда получение точного решения невозможно.

Будем искать решение уравнения (86) в области $0 \le x \le 1$, $t > 0$. Начальное распределение температуры $u(x,0) = u_0(x)$. Граничные условия будут рассмотрены ниже, в п. 3.1. Численное решение уравнения (86) будем строить на равномерной сетке с шагом $h = 1/N$ по пространственной координате x и шагом τ по времени, рис. 11 (а). Условимся использовать $u(x,t)$ для обозначения точного решения (86), а $v(x,t)$ — для численного решения. При этом для краткости будем обозначать значения функций в узлах сетки соответствующей буквой с двумя индексами: нижним (пространственным) и верхним (временны́м). Например, $v_j^m \equiv v(x_j, t_m)$, $f_j^m \equiv f(x_j, t_m)$ и т. п. Пространственный индекс j пробегает значения от 0 до N, что соответствует $0 \le x \le 1$.

3.1. Граничные условия

Условия на границе будем задавать в одном из двух видов, известных как задача Дирихле ($u(0,t) = q_0(t)$, $u(1,t) = q_1(t)$) и задача Неймана ($u_x(0,t) = p_0(t)$, $u_x(1,t) = p_1(t)$, где $u_x \equiv \frac{\partial}{\partial x} u$). Легко увидеть, что заменой неизвестной функции можно свести каждую из этих задач (либо их линейную суперпозицию) к аналогичной задаче с нулевыми

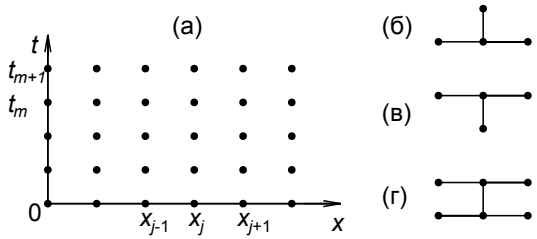

Рис. 11. (а) Равномерная сетка для построения численного решения; шаблоны численных схем: (б) явной, (в) неявной, (г) Кранка — Николсона

условиями на границе. Так, в случае задачи Дирихле можно сделать подстановку:

$$u(x,t) = (1-x)q_0(t) + xq_1(t) + w(x,t),$$

где функция w удовлетворяет уравнению

$$\partial_t w(x,t) = \partial_x^2 w(x,t) + f(x,t) - (1-x)q_0'(t) - xq_1'(t)$$

с нулевыми граничными условиями $w(0,t) = v(1,t) = 0$. Аналогично, для задачи Неймана можно использовать подстановку:

$$u(x,t) = (q_1(t) - q_0(t))\,\frac{x^2}{2} + q_0(t)x + \omega(x,t), \quad \omega_x(0,t) = \omega_x(1,t) = 0.$$

Использование нулевых граничных условий позволяет упростить процедуру численного решения.

Чтобы учесть граничные условия при решении задачи Дирихле, очевидно, следует положить $v_0^m = 0$, $v_N^m = 0$ для всех $m = 1, 2, \ldots$ Как применить условие $v_x = 0$ при решении задачи Неймана? Наиболее простой способ состоит в замене производной v_x на уже известную нам конечную разность: $\partial_x u(x=0) \approx (v_1 - v_0)/h$. Однако использование данного соотношения приводит к появлению погрешности $\mathcal{O}(h)$. Как мы увидим дальше, сравнительно легко можно построить разностную схему второго порядка точности по шагу сетки h, поэтому применение формулы первого порядка для производной $\partial_x u$ нежелательно: это снизит порядок точности численного решения. В этой связи выведем разностную формулу, аппроксимирующую производную на краю сетки со вторым порядком точности. Для этого запишем невязку R между значением производной $v'(0)$ и разностной формулой с неопределёнными коэффициентами:

$$R = v'(0) - \frac{av_0 + bv_1 + cv_2}{h}.$$

Наличие h в знаменателе формулы позволяет сделать коэффициенты a, b и c безразмерными. Разложив решение $v(x)$ в ряд Тейлора, выберем коэффициенты так, чтобы занулить невязку R в первых трёх порядках по шагу сетки h:

$$R = -v_0\frac{a+b+c}{h} + v'(0)(1-b-2c) - v''(0)\frac{h}{2}\big(b+4c\big) + \mathcal{O}(h^2).$$

Разрешая систему линейных уравнений относительно коэффициентов a, b и c, получаем искомую разностную формулу:

$$v'(0) = \frac{-3v_0 + 4v_1 - v_2}{2h} + \mathcal{O}(h^2). \tag{87}$$

Действуя аналогичным образом либо записывая полученное выражение (87) для производной $\partial_x v(1-x)$ при $x = 0$, несложно получить выражение для производной на правом краю сетки:

$$v'(x_N) = \frac{3v_N - 4v_{N-1} + v_{N-2}}{2h} + \mathcal{O}(h^2). \tag{88}$$

Выражения (87) и (88) понадобятся нам для построения численных схем второго порядка точности для решения задачи Неймана.

3.2. Явная схема

Заменяя в уравнении (86) производные ∂_t и ∂_x^2 в точке (x_j, t_m) конечными разностями, несложно выписать наиболее простую схему для численного решения уравнения теплопроводности:

$$\frac{v_j^{m+1} - v_j^m}{\tau} = \frac{v_{j+1}^m - 2v_j^m + v_{j-1}^m}{h^2} + f_j^m. \tag{89}$$

Если значения v_j^m известны на m-м слое по времени, используя соотношение (89) и граничные условия, можно вычислить значения численного решения v_j^{m+1} на следующем, $(m+1)$-м, слое по времени по *явным* формулам, ввиду чего (89) относят к классу *явных* численных схем. Например, для задачи Дирихле имеем:

$$v_0^{m+1} = 0, \quad v_N^{m+1} = 0, \tag{90}$$

$$v_j^{m+1} = v_j^m + \frac{\tau}{h^2}\big(v_{j+1}^m - 2v_j^m + v_{j-1}^m\big) + \tau f_j^m, \quad 1 \le j \le N-1. \tag{91}$$

При решении задачи Неймана вначале нужно применить формулу (91) для вычисления v_j^{m+1} во внутренних точках $1 \le j \le N-1$, а затем использовать граничные условия, вычислив значения искомого решения

на границах v_0^{m+1} и v_N^{m+1} с использованием полученных выше разностных соотношений (87) и (88).

Соотношение (91) связывает одно неизвестное значение сеточной функции v_j^{m+1} с тремя известными значениями v_{j-1}^m, v_j^m и v_{j+1}^m. Схематически данную связь удобно изображать с помощью *шаблона* численной схемы, показанного на рис. 11 (б). Точки на рисунке соответствуют четырём узлам сетки, связь значений численного решения в которых определяется соотношением (91).

Обратим внимание на чрезвычайную простоту явной схемы (89). Действительно, чтобы получить схему (89), нам потребовалось лишь заменить частные производные в уравнении (86) соответствующими конечными разностями. Сделав это вполне очевидным способом, мы получили формулу (91), позволяющую нам совместно с граничными условиями вычислять искомые значения численного решения слой за слоем по времени.

Однако использование явной схемы на практике для построения численного решения оказывается не столь простым, как вывод и программирование формул (90), (91). Для понимания основной проблемы, связанной с использованием явной схемы, полезно посмотреть на систему разностных уравнений (89) как на схему Эйлера для системы ОДУ первого порядка на $(N-1)$ функцию времени $v_1(t), \ldots, v_{N-1}(t)$. Ранее мы уже видели [1, с. 101], что численное решение, полученное с использованием явных схем и недостаточно малого шага сетки, оказывается неустойчивым. В этой связи исследуем вопрос устойчивости схемы (89). Для простоты рассмотрим задачу Дирихле $u(0,t) = u(1,t) = 0$ для однородного уравнения ($f(x,t) \equiv 0$). Общее решение такой задачи может быть получено методом разделения переменных:

$$u(x,t) = \sum_{q=1}^{\infty} c_q \sin(\pi q x) \exp(-\pi^2 q^2 t),$$

где c_q — коэффициенты Фурье начального распределения температуры при $t = 0$. Каждый член ряда представляет собой гармонику Фурье, равную нулю на границах отрезка $[0,1]$, с амплитудой, экспоненциально убывающей во времени. Посмотрим, сохранится ли это свойство базисных решений при переходе к разностной схеме (89). Для этого рассмотрим вначале действие разностного оператора

$$\hat{L} v_j^m = \frac{v_{j+1}^m - 2v_j^m + v_{j-1}^m}{h^2} \tag{92}$$

на сеточную функцию $v_j^m = A^m \sin(\pi q x_j) = A^m \sin(\pi q j h)$, где A^m — амплитуда гармоники с номером q на m-м шаге по времени, h — шаг

сетки по x:

$$\hat{L}v_j^m = \frac{A^m}{h^2}\big(\sin(\pi qjh + \pi qh) - 2\sin(\pi qjh) + \sin(\pi qjh - \pi qh)\big).$$

Раскрывая синус суммы и разности, приводя подобные члены и используя тождество $\cos(\pi qh) - 1 \equiv -2\sin^2(\pi qh/2)$, имеем:

$$\hat{L}v_j^m = -\frac{4A^m}{h^2}\sin^2\left(\frac{\pi qh}{2}\right)\sin(\pi qjh) = -\frac{4}{h^2}\sin^2\left(\frac{\pi qh}{2}\right)\cdot v_j^m. \qquad (93)$$

Таким образом, мы получили, что $v_j^m = A^m \sin(\pi qjh)$ является собственной функцией разностного оператора (92). Подставляя полученный результат в численную схему $(v^{m+1} - v^m)/\tau = \hat{L}v^m$ (89), получаем выражение для амплитуды синуса на $(m+1)$-м слое:

$$A^{m+1} = A^m \cdot \left(1 - \frac{4\tau}{h^2}\sin^2\left(\frac{\pi qh}{2}\right)\right). \qquad (94)$$

Поскольку квадрат синуса не превосходит единицы, из полученного соотношения следует, что амплитуда A^m численного решения будет затухать в геометрической прогрессии (экспоненциально по t) независимо от номера гармоники q при условии

$$\tau < \frac{h^2}{2}. \qquad (95)$$

Следовательно, явная схема (89) является *условно устойчивой*. В случае, если условие (95) не выполнено, происходит развитие неустойчивости в численном решении. Как видно на рис. 12 (а), вначале численное решение может хорошо согласовываться с точным, однако начиная с некоторого момента t в численном решении становятся заметны высокочастотные осцилляции (см. рис. 12 (б)), амплитуда которых возрастает экспоненциально по времени (в геометрической прогрессии с увеличением числа шагов). В соответствии с (94), быстрее всего возрастает амплитуда гармоники Фурье с максимальной частотой, близкой к частоте Найквиста: $v_j \sim \sin(\frac{N-1}{N}\pi j)$. Это кардинально противоречит физическим ожиданиям, в соответствии с которыми высокочастотные компоненты решения должны затухать быстрее всего. Дальнейший рост неустойчивости приводит к появлению очень больших значений численного решения и скорому выходу за пределы разрядной сетки — в выводе программы появляются значения inf и nan.

Почему при нарушении соотношения (95) численное решение, построенное по явной схеме (89), оказывается неустойчивым и не аппроксимирует точное решение уравнения (86) даже в пределе τ, $h \to 0$?

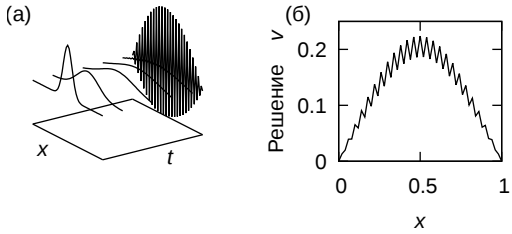

Рис. 12. (а) Динамика развития и (б) проявление неустойчивости

Чтобы ответить на этот вопрос, сравним распространение возмущений в уравнении теплопроводности и численной схеме (89). В непрерывном случае распространение возмущений описывается функцией Грина

$$G(x-x',t-t') = \frac{\theta(t-t')}{\sqrt{4\pi(t-t')}} \exp\left(-\frac{(x-x')^2}{4(t-t')}\right), \qquad (96)$$

где θ — функция Хевисайда. Формально $G(x) > 0 \quad \forall x$ при $t > t'$, что означает наличие дальнодействия в уравнении теплопроводности (86). Однако, хотя функция Грина G нигде не обращается в ноль при $t > t'$, она быстро убывает с ростом $|x - x'|$, что с физической точки зрения ограничивает область влияния возмущения масштабом расстояний $h_* \approx 2\sqrt{\tau}$. Тем не менее, даже несмотря на конечный размер области возмущений h_*, скорость распространения возмущений остаётся неограниченной: $dh_*/d\tau \to \infty$ при $\tau \to 0$. Принципиально иначе происходит распространение возмущений в численном решении, построенном по явной схеме (89). Если в некоторой точке (x_j, t_m) произошло возмущение (например, выделилось тепло), то в соответствии с (89) через время τ это приведёт к изменению температуры всего в трёх узлах сетки: v_j^{m-1}, v_j^m и v_j^{m+1}. На каждом шаге τ область возмущения (показана крестиками на рис. 13 (а)) расширяется ровно на один шаг сетки h влево и вправо, что соответствует постоянной и конечной скорости распространения возмущений $2h/\tau$. Следовательно, чтобы схема (89) корректно воспроизводила точное решение с неограниченно высокой скоростью распространения возмущений, необходимо положить $\tau = o(h) = \mathcal{O}(h^2)$, что согласуется с условием устойчивости (95) явной схемы (89).

Наконец, исследуем вопрос порядка аппроксимации численной схемы (89). Запишем невязку точного решения на численной схеме (89):

$$\psi = \frac{u_j^{m+1} - u_j^m}{\tau} - \frac{u_{j+1}^m - 2u_j^m + u_{j-1}^m}{h^2} - f_j^m.$$

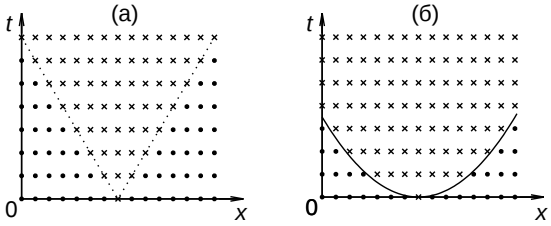

Рис. 13. Распространение возмущения в численном и точном решении

Раскладывая $u(x,t)$ в ряд Тейлора в точке (x_j, t_m), получаем:

$$\psi = \frac{u_t\tau + u_{tt}\tau^2/2 + \ldots}{\tau} - \frac{2u_{xx}h^2/2 + 2u_{xxxx}h^4/24 + \ldots}{h^2} - f,$$

где нижними индексами обозначены производные по t и x. В силу (86) $u_t - u_x x - f = 0$, откуда

$$\psi = u_{tt}\frac{\tau}{2} - u_{xxxx}\frac{h^2}{12} + \ldots = \mathcal{O}(\tau + h^2). \tag{97}$$

Следовательно, явная схема (89) имеет первый порядок точности по времени и второй — по пространственной координате x.

Таким образом, платой за простоту явной схемы является её низкая эффективность, связанная с условной устойчивостью и первым порядком аппроксимации по t. Как следствие, использование явной схемы требует большого количества шагов для построения численного решения. В следующих двух пунктах мы покажем, как можно исправить указанные недостатки.

3.3. Неявная схема

Если заменить частные производные u_t и u_{xx} в уравнении (86) конечными разностями не в точке (x_j, t_m), как это было сделано в п. 3.2, но в (x_j, t_{m+1}), получим схему

$$\frac{v_j^{m+1} - v_j^m}{\tau} = \frac{v_{j+1}^{m+1} - 2v_j^{m+1} + v_{j-1}^{m+1}}{h^2} + f_j^{m+1}. \tag{98}$$

Разностные соотношения (98) при $j = 1, \ldots, N-1$ образуют систему уравнений на $(N-1)$ неизвестную величину $v_1^{m+1}, \ldots, v_{N-1}^{m+1}$. Обратим внимание, что количество уравнений и неизвестных в системе на два меньше количества узлов сетки: входящие в (98) величины v_0^{m+1} и v_N^{m+1}

определяются условиями на границе. Значения температуры на следующем, $(m+1)$-м, слое по времени получаются в результате решения системы уравнений и уже не могут быть выписаны явно, как это было сделано в предыдущем параграфе (см. формулу (91)). В этой связи схему (98) относят к классу *неявных* численных схем.

В каждое уравнение системы (98) входят три неизвестных величины (v_j^{m+1}, $v_{j\pm1}^{m+1}$) на слое $t = t_{m+1}$ и известное значение v_j^m, что соответствует шаблону на рис. 11 (в), с. 68.

Система уравнений на величины v_j^{m+1} $(j = 1, \ldots, N-1)$ имеет трёхдиагональную матрицу и может быть решена методом прогонки, см. п. 1.5, с. 15. Коэффициенты трёхдиагональной матрицы (18) при решении задачи Дирихле с нулевыми условиями на границе имеют вид:

$$\begin{aligned}
a_i &= \begin{cases} 0, & i = 1; \\ -\tau/h^2, & i = 2, \ldots, N-1; \end{cases} \\
b_i &= 1 + \frac{2\tau}{h^2}; \\
c_i &= \begin{cases} -\tau/h^2, & i = 1, \ldots, N-2; \\ 0, & i = N-1; \end{cases} \\
d_i &= v_i^m + \tau f_i^{m+1}.
\end{aligned}$$

В случае решения задачи Неймана с условием $u_x = 0$ на границах значения v_0^{m+1} и v_N^{m+1} следует подставлять в (98) из соотношений (87,88), что приведёт к изменению матричных коэффициентов системы в первом и последнем уравнениях:

$$\begin{aligned}
a_i &= \begin{cases} 0, & i = 1, \\ -\tau/h^2, & i = 2, \ldots, N-2, \\ -\frac{2}{3}\tau/h^2, & i = N-1; \end{cases} \\
b_i &= \begin{cases} 1 + 2\tau/h^2, & i = 2, \ldots, N-2, \\ 1 + \frac{2}{3}\tau/h^2, & i = 1 \quad \text{и} \quad i = N-1; \end{cases} \\
c_i &= \begin{cases} -\frac{2}{3}\tau/h^2, & i = 1, \\ -\tau/h^2, & i = 2, \ldots, N-2, \\ 0, & i = N-1; \end{cases} \\
d_i &= v_j^m + \tau f_j^{m+1}.
\end{aligned}$$

Заметим, что в обоих рассмотренных случаях матричные коэффициенты удовлетворяют строгому условию диагонального преобладания ($|b_i| \geq |a_i| + |c_i|$), что гарантирует устойчивость решения системы линейных уравнений: малое изменение распределения температуры v_j^m

при $t = t_m$ будет приводить к малому изменению решения v_j^{m+1} на следующем слое при $t = t_{m+1}$.

Решение задачи с периодическими граничными условиями $u(0) = u(1)$, $u_t(0) = u_t(1)$ потребует применения соответствующей модификации метода прогонки, рассмотренной в п. 1.6 на с. 16.

Исследуем устойчивость неявной схемы (98). Как и раньше, проследим за изменением амплитуды A^m базисного решения $v_j^m = A^m \sin(\pi q j h)$. Используя полученное выше выражение (93), имеем:

$$A^{m+1} \cdot \left(1 + \frac{4\tau}{h^2} \sin^2\left(\frac{\pi q h}{2}\right)\right) = A^m,$$

откуда немедленно следует, что $|A^{m+1}| < |A^m| \quad \forall \tau, h > 0$, т. е. неявная схема (98) *безусловно устойчива*.

Очевидно, что неявная схема обеспечивает первый порядок аппроксимации по времени и второй — по x. При этом накопление ошибки может происходить ещё быстрее, чем при использовании явной схемы, что связано с решением системы большого количества $(n-2)$ линейных уравнений на каждом шаге τ.

Действуя аналогично изложенному в конце п. 3.2, несложно получить выражение для невязки ψ точного решения на схеме (98):

$$\psi = -u_{tt}\frac{\tau}{2} - u_{xxxx}\frac{h^2}{12} + \ldots = \mathcal{O}(\tau + h^2). \tag{99}$$

Из сравнения (99) с (97) видно, что коэффициенты при τ в невязке противоположны для явной (89) и неявной (98) схем. Это позволяет построить численную схему, обеспечивающую второй порядок аппроксимации по x и t.

3.4. Схема Кранка — Николсона

Записывая полусумму явной (89) и неявной (98) схем и используя для краткости обозначение $\hat{L}$ для разностного оператора (92), получим схему *Кранка — Николсона*:

$$\frac{v_j^{m+1} - v_j^m}{\tau} = \frac{1}{2}\hat{L}v_j^m + \frac{1}{2}\hat{L}v_j^{m+1} + \frac{f_j^m + f_j^{m+1}}{2}. \tag{100}$$

Неявная схема (100) также известна как *симметричная* схема и схема *с полусуммой* [2] (также встречаются названия *полунеявная* и *явно-неявная* схема, которые могут ввести в заблуждение неподготовленного читателя). Шаблон схемы (100) показан на рис. 11 (г). Из симметрии

схемы, а также полученных выше выражений для невязки явной (97) и неявной (99) схем несложно понять, что схема Кранка — Николсона обеспечивает второй порядок аппроксимации по x и t, т. е. подстановка точного решения в (100) даст невязку $\psi = \mathcal{O}(\tau^2 + h^2)$.

Исследуем устойчивость схемы (100). Подставляя в неё $v_j^m = A^m \sin(\pi q j h)$, полагая $f \equiv 0$ и используя (93), имеем:

$$A^{m+1} \cdot \left(1 + \frac{2\tau}{h^2} \sin^2\left(\frac{\pi q h}{2}\right)\right) = A^m \cdot \left(1 - \frac{2\tau}{h^2} \sin^2\left(\frac{\pi q h}{2}\right)\right).$$

Вводя для краткости обозначение $\varkappa = (2\tau/h^2)\sin^2(\pi q h/2)$ и учитывая, что $\varkappa > 0 \quad \forall \tau, h$, получаем

$$\left|\frac{A^{m+1}}{A^m}\right| = \left|\frac{1-\varkappa}{1+\varkappa}\right| < 1 \quad \forall \tau, h.$$

Таким образом, амплитуда всех базисных решений монотонно убывает, и схема (100) является безусловно устойчивой.

Как и в случае рассмотренной в предыдущем параграфе неявной схемы (98), расчёт по схеме Кранка — Николсона (100) требует использования метода прогонки для решения системы линейных уравнений на величины v_j^{m+1} $(j = 1, \ldots, N-1)$ на каждом шаге τ. Коэффициенты трёхдиагональной матрицы (18) при решении задачи Дирихле с нулевыми условиями на границе имеют вид:

$$\begin{aligned}
a_i &= \begin{cases} 0, & i = 1, \\ -\frac{1}{2}\tau/h^2, & i = 2, \ldots, N-1; \end{cases} \\
b_i &= 1 + \frac{\tau}{h^2}; \\
c_i &= \begin{cases} -\frac{1}{2}\tau/h^2, & i = 1, \ldots, N-2, \\ 0, & i = N-1; \end{cases} \\
d_i &= v_j^m + \frac{\tau}{2}\left(\hat{L} v_j^m + f_j^m + f_j^{m+1}\right).
\end{aligned}$$

В заключение данного параграфа сравним рассмотренные выше численные схемы для решения одномерного уравнения теплопроводности (86). На рис. 14 показана зависимость погрешности численного решения от количества узлов временно́й сетки:

$$R(T/\tau) \equiv \max_x \left|u(x,T) - v(x,T)\right|,$$

где $T = \text{const}$ — верхний предел интегрирования по t. На каждом графике показаны результаты расчётов по трём численным схемам: явной

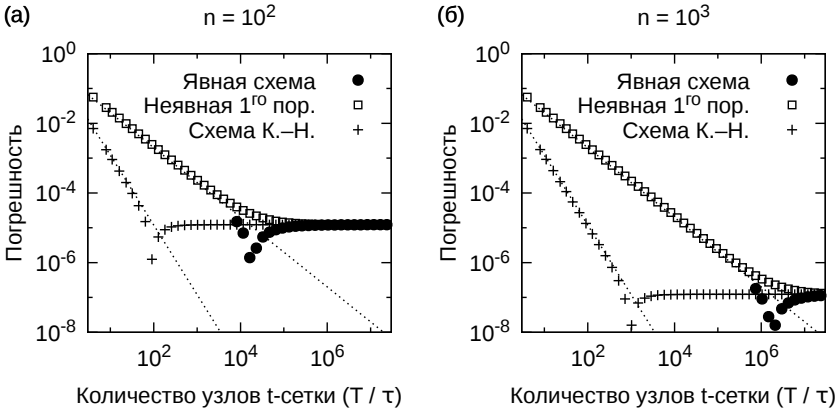

Рис. 14. Зависимость погрешности численного решения, построенного по явной схеме (89), неявной схеме первого порядка (98) и схеме Кранка — Николсона (100), от числа узлов временно́й сетки для (а) $n = 10^2$ и (б) $n = 10^3$.

схеме (89), неявной схеме первого порядка (98) и схеме Кранка — Николсона (100). Графики (а, б) соответствуют расчётам с различным числом узлов пространственной сетки: $n = 10^2$ и 10^3 соответственно.

Обратим внимание, что график погрешности для явной схемы построен начиная с достаточно большого числа шагов ($T/\tau \gtrsim 8 \times 10^3$ для $n = 10^2$ и $T/\tau \gtrsim 7 \times 10^6$ для $n = 10^3$): поскольку явная схема (89) является неустойчивой при $\tau > h^2/2$, построение численного решения по явной схеме возможно только при использовании достаточно малой величины шага τ (достаточно большого числа шагов T/τ).

Неявная схема (98) в силу абсолютной устойчивости позволяет получать численное решение при любой величине шага τ. Однако медленное (линейное по величине шага τ) убывание погрешности практически нивелирует её преимущество перед явной схемой (89). Действительно, сокращение числа шагов по времени за счёт перехода от явной схемы (89) к неявной схеме (98), хотя и возможно, но ведёт к пропорциональному снижению точности (увеличению погрешности численного решения).

И только применение неявной схемы более высокого (второго) порядка точности по t позволяет полностью реализовать преимущество абсолютно устойчивых схем. Использование схемы Кранка — Николсона (100) позволяет существенно (до двух-трёх порядков величины для $n = 10^2$ и 10^3) сократить число шагов по времени и, соответственно, повысить скорость выполнения расчётов без потери точности численного решения, см. рис. 14.

Наконец, обратим внимание, что при достаточно большом числе шагов (T/τ) погрешности численных решений, полученных различными методами, выходят на одну и ту же асимптоту (при фиксированном n)[28], см. рис. 14. Очевидно, это связано с тем, что в погрешности численного решения $R = \mathcal{O}(\tau^\alpha + h^2)$ первое слагаемое при малых τ становится пренебрежимо мало, так что погрешность R определяется шагом h пространственной сетки и не зависит от порядка точности α использованной схемы. При увеличении n с 10^2 до 10^3 асимптотическая ошибка уменьшается с $R \approx 10^{-5}$ до $R \approx 10^{-7}$, что согласуется с вторым порядком точности рассматриваемых численных схем по шагу пространственной сетки h.

3.5. Обобщение на двумерный случай

Рассмотрим уравнение теплопроводности в двумерном случае:

$$\partial_t u = \partial_x^2 u + \partial_v^2 u + f(x, y, t). \tag{101}$$

Численное решение уравнения (101) будем искать на равномерной сетке с шагом h_x и h_y по пространственным координатам x, y и τ по времени t. Аналогично одномерному уравнению, для обозначения сеточных функций условимся использовать нижние индексы для указания пространственных координат и верхний индекс для времени: $v_{jk}^m \equiv v(x_j, y_k, t_m)$.

Несложно построить двумерное обобщение явной схемы из п. 3.2 Для этого, как и в одномерном случае, заменим частные производные конечными разностями в точке (x_j, y_k, t_m), совершая при этом ошибку $\mathcal{O}(\tau + h_x^2 + h_y^2)$:

$$\frac{v_{jk}^{m+1} - v_{jk}^m}{\tau} = \hat{L}_x v_{jk}^m + \hat{L}_y v_{jk}^m + f_{jk}^m, \tag{102}$$

где $\hat{L}_x$, $\hat{L}_y$ обозначают разностные операторы, аппроксимирующие частные производные ∂_x^2 и ∂_y^2 соответственно:

$$\hat{L}_x v_{jk}^m \equiv \frac{v_{j+1,k}^m - 2v_{jk}^m + v_{j-1,k}^m}{h_x^2}, \tag{103}$$

$$\hat{L}_y v_{jk}^m \equiv \frac{v_{j,k+1}^m - 2v_{jk}^m + v_{j,k-1}^m}{h_y^2}. \tag{104}$$

[28] При дальнейшем уменьшении шага τ погрешность численного решения вновь начнёт возрастать из-за ошибок округления.

Для того, чтобы сделать один шаг по времени с использованием схемы (102) необходимо выполнить $\mathcal{O}(N_x N_y)$ арифметических операций, где N_x и N_y — число узлов сетки по x и y соответственно.

Подобно явной схеме для одномерного уравнения, схема (102) является условно устойчивой. В этом несложно убедиться, рассматривая эволюцию во времени амплитуды $A^m \equiv A(t_m)$ базисной функции

$$v_{jk}^m = A^m \sin(\pi q_x j h_x) \sin(\pi q_y k h_y). \tag{105}$$

Повторяя выкладки из п. 3.2 на с. 71, получим, что функция (105) является собственной для разностных операторов $\hat{L}_x$, $\hat{L}_y$ (ср. с формулой (93) на с. 71):

$$\hat{L}_{x,y} v_{jk}^m = -\frac{4}{h_{x,y}^2} \sin^2\left(\frac{\pi q_{x,y} h_{x,y}}{2}\right) \cdot v_{jk}^m. \tag{106}$$

Подставляя полученный результат в численную схему (102), получаем выражение для амплитуды базисной функции (105) на $(m+1)$-м слое:

$$A^{m+1} = A^m \cdot \left(1 - \frac{4\tau}{h_x^2} \sin^2\left(\frac{\pi q_x h_x}{2}\right) - \frac{4\tau}{h_y^2} \sin^2\left(\frac{\pi q_y h_y}{2}\right)\right).$$

Амплитуда всех базисных решений $(\forall q_x, q_y)$ будет убывать с ростом t при условии

$$\tau < \frac{1}{2}\left(\frac{1}{h_x^2} + \frac{1}{h_y^2}\right)^{-1}, \tag{107}$$

что является обобщением условия устойчивости (95) явной схемы на двумерный случай. Таким образом, схема (102) является условно устойчивой и обеспечивает первый порядок точности по времени и второй — по пространственным координатам.

Обобщения неявной схемы (98) и схемы Кранка — Николсона (100) на случай двух и более пространственных переменных, хотя и могут быть легко построены, но на практике не используются, уступая в эффективности более экономичным численным схемам, которые будут рассмотрены нами ниже. Поймём причины низкой эффективности двумерного обобщения схем (98), (100) на примере неявной схемы:

$$\frac{v_{jk}^{m+1} - v_{jk}^m}{\tau} = \hat{L}_x v_{jk}^{m+1} + \hat{L}_y v_{jk}^{m+1} + f_{jk}^{m+1}. \tag{108}$$

Для того, чтобы сделать шаг по времени и найти численное решение на следующем слое по времени необходимо решить систему из

$N_x \times N_y$ линейных алгебраических уравнений на величины v_{jk}^{m+1} при $0 \le j < N_x$ и $0 \le k < N_y$. Записывая систему (108) в базисе $(v_{0,0}, v_{0,1}, \ldots, v_{0,N_y-1}, v_{1,0}, v_{1,1}, v_{1,2}, \ldots, v_{1,N_y-1}, v_{2,0}, v_{2,1}, \ldots, v_{N_x-1,N_y-1})$, легко понять, что матрица системы является ленточной с шириной ленты $2N_y + 1$. Следовательно, для выполнения одного шага по времени с использованием схемы (108) потребуется $\mathcal{O}((2N_y+1)N_xN_y) = \mathcal{O}(N^3)$ арифметических операций при $N_x \approx N_y \equiv N$. К аналогичному выводу можно прийти и при рассмотрении двумерного обобщения схемы Кранка — Николсона. В трёхмерном случае проблема ещё более усугубится: ширина ленты матрицы системы будет $\approx N^2$, и, следовательно, число арифметических операций будет расти как $\mathcal{O}(N^5)$. Ниже мы рассмотрим так называемые *экономичные* схемы, позволяющие повысить эффективность построения численного решения уравнения теплопроводности.

3.6. Продольно-поперечная схема

Построим безусловно устойчивую численную схему, требующую выполнения $\mathcal{O}(N^2)$ операций для перехода на следующий слой по времени. Для этого разобьём шаг τ численного интегрирования на два за счёт введения дополнительного (вспомогательного, полуцелого) слоя по времени $t = t_{m+1/2}$. Вначале сделаем половину шага по времени, $\tau/2$, используя неявную схему по x и явную по y. Затем делаем оставшуюся половину шага $\tau/2$, используя схему, явную по x и неявную по y:

$$\frac{v_{jk}^{m+\frac{1}{2}} - v_{jk}^{m}}{\tau/2} = \hat{L}_x v_{jk}^{m+\frac{1}{2}} + \hat{L}_y v_{jk}^{m} + f_{jk}^{m+\frac{1}{2}}, \tag{109}$$

$$\frac{v_{jk}^{m+1} - v_{jk}^{m+\frac{1}{2}}}{\tau/2} = \hat{L}_x v_{jk}^{m+\frac{1}{2}} + \hat{L}_y v_{jk}^{m+1} + f_{jk}^{m+\frac{1}{2}}. \tag{110}$$

Проследим за зоной влияния в схеме (109, 110): предположим, что в узле v_{jk}^m возникло возмущение (рис. 15 (а)) и проследим, на какие узлы сетки оно распространится при $t = t_{m+1}$. Поскольку на первой половине шага (109) используется неявная по x схема, возмущение в узле v_{jk}^m распространится вдоль всей оси x, так что окажутся затронутыми узлы $v_{0k}^{m+1/2}, \ldots, v_{Nk}^{m+1/2}$. В направлении y схема (109) явная, так что по y возмущение распространится всего на два узла $v_{j,k\pm1}$ промежуточного слоя $t_{m+1/2}$ (рис. 15 (б)). На второй половине шага (110) используется схема, неявная в направлении y, благодаря чему возмущение из узлов $v_{0k}^{m+1/2}, \ldots, v_{Nk}^{m+1/2}$ передастся вдоль оси y на все без

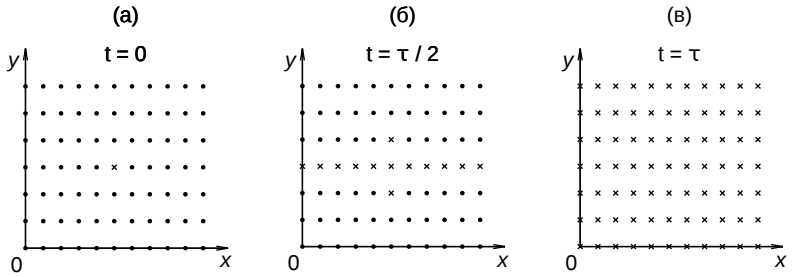

Рис. 15. Распространение возмущения при использовании продольно-поперечной схемы

исключения узлы сетки при $t = t_{m+1}$ (рис. 15 (в)). Таким образом, зона влияния продольно-поперечной схемы (109, 110) неограничена, что позволяет ожидать безусловной устойчивости данной схемы.

Исследуем устойчивость более формально, проследив за эволюцией амплитуды A^m базисной функции (105). Используя (106), получаем для амплитуды после первой половины шага (109):

$$A^{m+\frac{1}{2}} \cdot \left(1 + \frac{8\tau}{h_x^2} \sin^2 \frac{\pi q_x h_x}{2}\right) = A^m \cdot \left(1 - \frac{8\tau}{h_y^2} \sin^2 \frac{\pi q_y h_y}{2}\right).$$

Аналогично на второй половине шага (110):

$$A^{m+1} \cdot \left(1 + \frac{8\tau}{h_y^2} \sin^2 \frac{\pi q_y h_y}{2}\right) = A^{m+\frac{1}{2}} \cdot \left(1 - \frac{8\tau}{h_x^2} \sin^2 \frac{\pi q_x h_x}{2}\right).$$

Поделив полученные выражения друг на друга, вычислим отношение амплитуд базисной функции на соседних слоях по времени:

$$\frac{A^{m+1}}{A^m} = \frac{1 - \frac{8\tau}{h_x^2} \sin^2 \frac{\pi q_x h_x}{2}}{1 + \frac{8\tau}{h_x^2} \sin^2 \frac{\pi q_x h_x}{2}} \cdot \frac{1 - \frac{8\tau}{h_y^2} \sin^2 \frac{\pi q_y h_y}{2}}{1 + \frac{8\tau}{h_y^2} \sin^2 \frac{\pi q_y h_y}{2}}.$$

Вводя обозначение

$$\varkappa_{x,y} \equiv \frac{8\tau}{h_{x,y}^2} \sin^2 \frac{\pi q_{x,y} h_{x,y}}{2}$$

и учитывая, что $\varkappa_{x,y} > 0 \quad \forall \tau, h_x, h_y > 0$, имеем:

$$\left|\frac{A^{m+1}}{A^m}\right| = \left|\frac{1 - \varkappa_x}{1 + \varkappa_x}\right| \cdot \left|\frac{1 - \varkappa_y}{1 + \varkappa_y}\right| < 1 \quad \forall \tau, h_{x,y} > 0.$$

Таким образом, амплитуда всех базисных решений монотонно убывает во времени при любых значениях шага сетки, что означает безусловную устойчивость продольно-поперечной схемы.

Найдем порядок точности продольно-поперечной схемы. Вычитая уравнение (110) из (109), выразим значение численного решения на полуцелом слое:

$$v_{jk}^{m+\frac{1}{2}} = \frac{1}{2}\left(v_{jk}^{m} + v_{jk}^{m+1}\right) + \frac{\tau}{4}\hat{L}_y\left(v_{jk}^{m} - v_{jk}^{m+1}\right).$$

Суммируя (109) и (110) и подставляя $v_{jk}^{m+1/2}$ из полученного выше выражения, имеем:

$$\frac{v_{jk}^{m+1} - v_{jk}^{m}}{\tau} = \left(\hat{L}_x + \hat{L}_y\right)\frac{v_{jk}^{m} + v_{jk}^{m+1}}{2} + f_{jk}^{m+\frac{1}{2}} - \frac{\tau}{4}\hat{L}_x\hat{L}_y\left(v_{jk}^{m+1} - v_{jk}^{m}\right).$$

Первые два члена в полученном равенстве аппроксимируют частные производные u_t и $u_{xx} + u_{yy}$ в точке $(x_j, y_k, t_{m+1/2})$ со вторым порядком точности. Следовательно, при подстановке в полученное равенство точного решения $u(x, y, t)$ уравнения теплопроводности вместо численного $v(x, y, t)$, первые три члена сократятся с точностью до невязки $\psi = \mathcal{O}(\tau^2 + h_x^2 + h_y^2)$. Тот же порядок малости имеет и последний член в полученном равенстве, $(\tau/4)\hat{L}_x\hat{L}_y(u_{jk}^{m+1} - u_{jk}^{m})$. Действительно, $u_{jk}^{m+1} \approx u_{jk}^{m} + u'\tau$, откуда следует, что $\tau \cdot (u_{jk}^{m+1} - u_{jk}^{m}) = \mathcal{O}(\tau^2)$. Таким образом, продольно-поперечная схема (109, 110) обеспечивает второй порядок аппроксимации по времени и пространственным координатам.

Для организации счёта по продольно-поперечной схеме на первой половине шага необходимо в цикле по индексу k делать шаг $\tau/2$, используя неявную схему (109). На второй половине шага следует организовать цикл по индексу j, выполняя для каждого значения j шаг $\tau/2$ по неявной схеме (110). Использование неявной схемы описано в п. 3.3 на с. 73.

3.7. Локально одномерный метод

Рассмотренная выше продольно-поперечная схема не обобщается на случай трёх и более измерений без потери порядка точности и абсолютной устойчивости [2, с. 395]. В этой связи для численного интегрирования многомерных параболических уравнений используют другую эффективную схему — *локально одномерную*. Идея данного метода также основана на разделении шага по времени на несколько частей и выполнении каждой части шага с использованием схемы, неявной по одной

пространственной координате. В отличие от продольно-поперечной схемы, здесь на каждой части шага учитывается распространение тепла только вдоль одной оси (что и отражено в названии метода), а в качестве неявной схемы используется схема с полусуммой (100) на с. 75.

Введём p промежуточных слоёв (где p — количество пространственных координат). Для сокращения записи будем опускать нижние (пространственные) индексы, а также использовать w для обозначения численного решения на промежуточных слоях по времени:

$$w^j \equiv v^{k+j/p}, \quad j = 0, \ldots, p.$$

Величины w^j, зависящие от p пространственных координат, подчиняются уравнениям:

$$\frac{w^{j+1} - w^j}{\tau} = \hat{L}_j \frac{w^{j+1} + w^j}{2} + f^j, \quad \sum f^j = f, \quad j = 1, \ldots, p. \qquad (111)$$

Здесь $\hat{L}_j$ — разностный аналог дифференциального оператора $\partial^2/\partial x_j^2$. Поскольку переход к каждому следующему промежуточному слою (от w^j к w^{j+1}) выполняется по абсолютно устойчивой неявной схеме, все малые высокочастотные возмущения затухают, что обусловливает абсолютную устойчивость схемы (111) в целом.

Можно показать [2, с. 397], что в частном случае отсутствия зависимости от пространственных координат коэффициента теплопроводности и распределения источников тепла f схема (111) имеет второй порядок аппроксимации по времени и пространственным координатам, $\mathcal{O}(\tau^2 + h_1^2 + h_2^2 + \ldots + h_p^2)$. В общем случае, при наличии зависимости от координат, схема имеет первый порядок точности по времени, $\mathcal{O}(\tau + \sum h_j^2)$ [A11], что позволяет использовать вместо схемы с полусуммой (100) чуть более компактную в записи неявную схему (98), см. с. 73.

Упражнения

1) Запишите численную схему, соответствующую шаблону 
К какому классу численных схем — явных или неявных — она относится?

2) Исследуйте устойчивость схемы из задачи 1.

3) Температура воздуха на улице осциллирует по закону из задачи 4 на с. 66, внутри помещения поддерживается температура +20°C. Пренебрегая краевыми эффектами, вычислить средний

поток тепла через участок стены площадью 50 м^2, не освещаемый прямыми солнечными лучами. Толщина стены 150 мм, теплопроводность 0,05 Вт/м·К, плотность 75 кг/м^3, теплоёмкость 1,3 кДж/(кг·К).

4) Рассчитайте амплитуду суточных и годовых осцилляций температуры в сухом песчаном грунте. На какой глубине амплитуда осцилляций не превышает 1°C? 0,1°C? Теплопроводность грунта принять равной 1,2 Вт/м·К, теплоёмкость 1 кДж/кг·К, плотность 1,4 т/м^3.

5) Горячая вода $t = 95°$ C прокачивается по трубе $L = 100$ м, $r = 3/4''$ со средним объёмным расходом $V = 0{,}1$ л/с. Считая течение в трубе ламинарным, исследовать зависимость температуры воды на выходе от коэффициента теплоотдачи между трубой и окружающей средой.

6) На границах квадратной области $0 \le x, y \le L$ поддерживается температура $T = 0$, внутри области происходит тепловыделение $f(x, y) = 1 - (1 - x/L)(1 - y/L)xy/L^2$. Исследовать зависимость температуры $T(x, y, t)$ от времени в центре квадрата, полагая начальную температуру $T(x, y, 0)$ равной нулю. Как меняется во времени величина

$$\delta = \max_{x,y} |T(x, y, t) - T_0(x, y)|,$$

где $T_0(x, y)$ — равновесное распределение температуры ($T_0 = \lim_{t\to\infty} T(x, y)$)?

7) Два длинных цилиндрических электрода расположены параллельно друг другу в электропроводящем пространстве. Исследуйте эволюцию распределения температуры в пространстве от времени после подключения к электродам постоянного напряжения; постройте графики распределения плотности тока и стационарной температуры. Указания: для простоты рассмотрите двумерную задачу, пренебрегая краевыми эффектами. Используйте нулевые граничные условия, полагая температуру на поверхности электродов и большом удалении от них равной нулю.

4. Нелинейное уравнение Шрёдингера

Рассмотрим ещё одно параболическое уравнение в частных производных второго порядка — нелинейное уравнение Шрёдингера:

$$i\partial_z A = \frac{\beta_2}{2}\partial_t^2 A - \gamma|A|^2 A, \tag{112}$$

где A — огибающая волнового пакета[29], распространяющегося в среде с дисперсией β_2 и нелинейностью γ вдоль z — пространственной координаты; t — время в *бегущей* системе координат, см. п. 4.1 и формулу (122) на с. 90. Своё название уравнение (112) получило ввиду формального сходства с квантовомеханическим уравнением Шрёдингера, в которое оно переходит в линейном пределе $\gamma = 0$ после переобозначений $z \leftrightarrow t$.

Нелинейное уравнение Шрёдингера (112) описывает эволюцию огибающей волнового пакета $A(t)$ при распространении в среде с дисперсией групповых скоростей и кубической нелинейностью. Соответствующие задачи возникают, например, в физике плазмы, где наличие кубического члена может быть обусловлено действием пондеромоторных сил либо нагревной нелинейностью. Другим важным примером является нелинейная оптика, где уравнение (112) широко используется для описания распространения коротких (длительностью $\gtrsim 10^{-12}$ с) оптических импульсов. В настоящее время такие импульсы повсеместно применяются для передачи информации на магистральных линиях связи. Ввиду быстрого роста потребления Интернет-трафика во всём мире актуальной проблемой является повышение пропускной способности информационных каналов. Поскольку экстенсивное решение — прокладка новых линий телекоммуникаций — связано с большими финансовыми затратами, приоритетным направлением является использование новых способов передачи данных по уже имеющимся волоконно-оптическим линиям связи. В этой связи нелинейное уравнение Шрёдингера (112) активно используется для решения широкого круга инженерных оптимизационных задач в области телекоммуникаций.

В отличие от многих других уравнений математической физики, описывающих эволюцию состояния некоторой системы во времени, начальные условия для (112) ставятся при $z = 0$: $A(0,t) = A_0(t)$, при этом в роли эволюционной координаты выступает z — расстояние, пройденное волновым пакетом в среде (например, волоконно-оптической линии связи).

[29] Модуль электрического поля волны $|\mathbf{E}| \propto \mathrm{Re}(Ae^{i\omega_0 t})$, где ω_0 — несущая оптическая частота. В оптике нормировку удобно выбирать так, чтобы $|A|^2$ был равен мощности излучения.

Аналитические решения задачи Коши для уравнения (112) могут быть получены лишь в относительно узком классе частных случаев с помощью метода обратной задачи рассеяния [А12]. В общем случае произвольных начальных условий для решения уравнения (112) используют численные методы, с наиболее употребительными из которых мы познакомимся в п. 4.5. Однако перед этим в пп. 4.1–4.3 будут рассмотрены основные предельные случаи, допускающие построение точного решения, что необходимо для понимания физики явлений, описываемых уравнением (112), и верификации результатов численного моделирования.

4.1. Линейный канал

Рассмотрим распространение оптических импульсов в линейном канале связи. При $\gamma = 0$ уравнение (112) переходит в линейное дифференциальное уравнение второго порядка: $iA_z = \frac{1}{2}\beta_2 A_{tt}$. Данное уравнение с точностью до переобозначений совпадает с квантовомеханическим уравнением Шрёдингера для свободной частицы. Его решение может быть легко выписано через функцию Грина после выполнения преобразования Фурье по времени: $\partial_t \to -i\omega \Rightarrow \tilde{A}_z(z,\omega) = \frac{i}{2}\beta_2\omega^2\tilde{A}(z,\omega) \Rightarrow$

$$\tilde{A}(z,\omega) = \tilde{A}(0,\omega)\exp\left(\frac{i}{2}\beta_2\omega^2 z\right). \tag{113}$$

Выполняя обратное преобразование Фурье, можно выписать решение $A(z,t) = \int A(0,t')\,G(t,t')\,dt'$ через функцию Грина G:

$$G(t,t') = \frac{1}{2\pi}\int\limits_{-\infty}^{+\infty}\exp\left(\frac{i}{2}\beta_2\omega^2 z + i\omega t' - i\omega t\right)d\omega. \tag{114}$$

В случае $\beta_2 = 0$ выражение (114) переходит в $G(t,t') = \delta(t,t')$, так что огибающая волнового пакета не изменяется при распространении по z: $A(z,t) = A(z,0)$.

При $\beta_2 \neq 0$ распространение пакета сопровождается набегом фазы в соответствии с (113), при этом непосредственно измеряемая физическая величина — спектральная плотность мощности $|\tilde{A}(\omega)|^2$ — остаётся постоянной: $\partial_z|\tilde{A}(z,\omega)|^2 = 0$. Это соответствует неизменности частоты (цвета) излучения при распространении оптических волн в линейном режиме (в пределе малой мощности). В противоположность этому, временна́я структура волновых пакетов при распространении в линейной среде с дисперсией претерпевает изменения: происходит искажение

формы импульсов и/или изменение их длительности. В частности, гауссовы импульсы $A(0,t) = A_0 \exp(-\frac{1}{2}(t/\tau_0)^2)$ после прохождения в среде расстояния z остаются гауссовыми, изменяя длительность и приобретая частотную модуляцию:

$$A(z,t) = \frac{A_0}{\sqrt{1 - i\frac{z}{L_D}\mathrm{sgn}\beta_2}} \exp\left(\frac{-\frac{1}{2}(\frac{t}{\tau_0})^2}{1 + (\frac{z}{L_D})^2}\right) \exp\left(\frac{-\frac{i}{2}(\frac{t}{\tau_0})^2 \cdot z/L_D}{1 + (\frac{z}{L_D})^2}\right),$$

где

$$L_D = \tau_0^2/|\beta_2| \tag{115}$$

обозначает *дисперсионную длину* — масштаб расстояния, на котором проявляются изменения огибающей волнового пакета за счёт дисперсии групповых скоростей.

Как следует из полученного выражения, длительность гауссового импульса увеличивается при распространении в линейном канале связи по закону

$$\tau(z) = \tau_0 \sqrt{1 + (z/L_D)^2}.$$

При малых z асимптотика квадратична по z, поскольку при $z = 0$ импульс является *спектрально ограниченным*, и его длительность минимальна при заданной ширине спектра. При $z \gg L_D$ асимптотика линейна, $\tau(z) \sim \tau_0 z/L_D$: излучение на переднем и заднем фронтах импульса имеет разные частоты и распространяется с различными скоростями из-за дисперсии, что и приводит к росту длительности, пропорциональному ширине спектра $1/\tau_0$, дисперсии $|\beta_2|$ и пройденному расстоянию z. Из данных рассуждений очевидно, что линейная асимптотика длительности $\tau \propto z$ при $z \gg L_D$ имеет место для импульсов произвольной формы.

4.2. Бездисперсионный канал

Рассмотрим уравнение (112) в противоположном предельном случае: $\gamma \neq 0$, $\beta_2 = 0$. При этом уравнение (112) перейдёт в $A_z = i\gamma|A|^2 A$, что позволяет выписать его точное решение:

$$A(z) = A(0)\exp(i\gamma|A|^2 z). \tag{116}$$

Видно, что при распространении импульса в оптическом волокне с нулевой дисперсией временна́я форма сигнала $|A(z)|^2$ остаётся неизменной. При этом импульс приобретает нелинейный набег фазы $\varphi =$

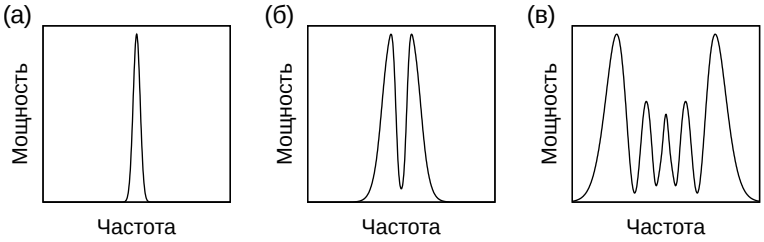

Рис. 16. Эволюция спектра при распространении гауссовского импульса в нелинейной среде без дисперсии: (а) $z = 0$, (б) $z = 5L_N$, (в) $z = 15L_N$

$\gamma|A|^2 z = \gamma P(t) z$, где P — мгновенная мощность. Набег фазы φ приводит к сдвигу частоты:

$$\delta\omega = \dot{\varphi} = \gamma \dot{P} z.$$

Данный эффект известен как *фазовая самомодуляция* и обусловлен зависимостью показателя преломления оптической среды от мощности распространяющегося в среде излучения. Нелинейный сдвиг частоты максимален на фронтах импульса, уменьшаясь на краях (в силу $P \approx 0$) и обращаясь в ноль в центре импульса (в точке максимума), где $\dot{P} = 0$. По аналогии с L_D (115) введём *нелинейную* длину, на которой проявляется фазовая самомодуляция:

$$L_N = \frac{1}{\gamma P}. \tag{117}$$

На рис. 16 показаны спектры гауссовского импульса после прохождения различных расстояний z в нелинейной среде без дисперсии.

4.3. Солитоны

Используя метод обратной задачи рассеяния [A12], можно получить аналитическое решение уравнения (112) ещё в одном частном случае, когда нелинейный и дисперсионный масштабы длины в точности равны. В случае аномальной дисперсии групповых скоростей ($\beta_2 < 0$) дисперсия и нелинейность компенсируют друг друга для оптических *солитонов* — импульсов с огибающей в форме гиперболического секанса:

$$A(0,t) = \frac{A_0}{\cosh(t/\tau)}, \quad |A_0|^2 = \frac{-\beta_2}{\gamma\tau^2}, \quad \beta_2 < 0. \tag{118}$$

Солитоны (118) распространяются в среде с нелинейностью и дисперсией без изменения формы и спектра на большие расстояния[30], приобретая при этом лишь набег общей фазы (аргумента комплекснозначной амплитуды A):

$$A(z,t) = A(0,t)e^{i\varkappa z}, \tag{119}$$

в чём несложно убедиться, подставив (118) в (119) и далее в (112).

Помимо *фундаментальных* солитонов (118) уравнение (112) имеет решения в виде солитонов *высших порядков*:

$$A(0,t) = \frac{A_0}{\cosh(t/\tau)}, \quad |A_0|^2 = \frac{-\beta_2 N^2}{\gamma\tau^2}, \quad \beta_2 < 0, \tag{120}$$

где $N = 1, 2, \ldots$ называют *порядком* солитона. При $N \geq 2$ солитоны имеют осциллирующую динамику, при распространении вдоль z периодически распадаясь и вновь восстанавливая свою форму[31]. Период осцилляций равен $L_D \cdot \pi/2$ [A13, с. 115].

Точные аналитические решения, полученные выше в пп. 4.1–4.3, понадобятся нам для проверки правильности численного моделирования. Другим регулярным способом контроля правильности реализации численных схем и точности вычислений является проверка сохранения интегралов движения. Очевидным примером величины, сохраняющейся в уравнении (112), является энергия $\int |A|^2\,dt$. Чтобы убедиться в сохранении энергии, необходимо умножить уравнение (112) на A^* и прибавить к нему комплексно-сопряжённое уравнение, умноженное на A:

$$\begin{array}{l|l} A^*\times & \partial_z A \;\; = -\dfrac{i}{2}\beta_2\partial_t^2 A \;\; + i\gamma|A|^2 A, \\ A\;\times & \partial_z A^* = +\dfrac{i}{2}\beta_2\partial_t^2 A^* - i\gamma|A|^2 A^*. \end{array}$$

Проинтегрировав сумму полученных уравнений по времени, получим

$$\frac{d}{dz}\int\limits_{-\infty}^{+\infty} |A|^2\,dt = \frac{i}{2}\beta_2 \int\limits_{-\infty}^{+\infty} (A_{tt}^* A - A^* A_{tt})\,dt.$$

[30] В соответствии с уравнением (112) солитоны могут распространяться на сколь угодно большие расстояния, однако в реальных линиях связи имеются оптические потери, комбинационное рассеяние и другие физические эффекты, ограничивающие дальность распространения солитонов.

[31] Комбинационное рассеяние и другие физические эффекты, не учитываемые в уравнении (112), могут приводить к разрушению солитонов высоких порядков, в результате чего решение $A(z,t)$ не будет периодической функцией z.

Интегрируя по частям и учитывая ограниченность волнового пакета во времени ($|A(t)| \to 0$ при $t \to \pm\infty$), получаем $\frac{d}{dz}\int |A|^2\,dt = 0$, что и требовалось показать.

4.4. Физическое обоснование

Скажем ещё несколько слов о физическом смысле членов в правой части нелинейного уравнения Шрёдингера (112). Обратим внимание, что набег фазы $\varphi = \frac{1}{2}\beta_2\omega^2$ в (113) за счёт дисперсии групповых скоростей есть не что иное, как квадратичный член ряда Тейлора для разложения волнового числа[32] $\beta(\omega)$. Таким образом, обобщение решения (113) имеет вид:

$$\tilde{A}(z,\omega) = \tilde{A}(0,\omega)e^{i\beta(\omega)z} = \tilde{A}(0,\omega)\exp\left(iz\sum_{k=0}^{\infty}\frac{\beta_k\omega^k}{k!}\right). \qquad (121)$$

Член суммы с $k = 0$ в показателе экспоненты (121) описывает общий набег фазы волнового пакета и может быть устранён заменой $A \to A\exp(-i\beta_0 z)$. Следующий член ($k = 1$) описывает запаздывание волнового пакета при распространении. В этом несложно убедиться, если рассмотреть $\beta(\omega) = \beta_1\omega$, что приводит нас к функции Грина

$$G(t,t') = \frac{1}{2\pi}\int\limits_{-\infty}^{+\infty}\exp\left(i\beta_1\omega z + i\omega t' - i\omega t\right)\,d\omega = \delta(t - t' - \beta_1 z),$$

откуда немедленно следует $A(z,t) = A(0, t-\beta_1 z)$. Таким образом, член $\beta_1\omega$ в (121) описывает распространение волнового пакета без изменения формы с групповой скоростью $v_g = 1/\beta_1$. Для упрощения уравнений этот член также исключают из рассмотрения, переходя в систему координат (z,T), бегущую вместе с пакетом с его групповой скоростью:

$$T = t - \frac{z}{v_g} = t - \beta_1 z. \qquad (122)$$

Ввиду того, что неподвижная лабораторная система координат в расчётах не используется, далее мы будем всюду подразумевать под (z,t) бегущие координаты.

[32] При рассмотрении задачи распространения плоских волн в однородном пространстве $\mathbf{E} \propto \exp(ikz - i\omega t)$ аналогичная величина есть $k(\omega) = n(\omega)\omega/c$. В нелинейной волоконной оптике постоянная распространения $\beta(\omega)$, помимо материального вклада в дисперсию групповых скоростей $n(\omega)$, имеет также волноводный вклад, являясь решением спектральной задачи на моды волоконного волновода.

После указанных замен и переобозначений суммирование ряда Тейлора в (121) начинается с $k = 2$. В случае, когда ширины рассматриваемых оптических спектров невелики (что справедливо для большинства телекоммуникационных задач), высшими членами ряда Тейлора в (121) пренебрегают, в результате чего в уравнение (112) входит единственное слагаемое, описывающее хроматическую дисперсию групповых скоростей.

Аналогично кубическая нелинейность $|A|^2 A$ в (112) также является первым ненулевым нелинейным членом в разложении нелинейного оператора по степеням A. Квадратичный по A член, который в общем случае даёт более сильный вклад в нелинейность, равен нулю в средах с центром инверсии. Важным с практической точки зрения примером таких сред является плавленый кварц SiO_2, из которого изготавливают телекоммуникационные оптические волокна. Молекула кварца линейна (O = Si = O) и обладает центром инверсии (переходит сама в себя при преобразовании инверсии $\mathbf{r} \to -\mathbf{r}$). В отсутствии квадратичной нелинейности в средах с центром инверсии несложно убедиться, записав вектор поляризации среды $\mathbf{P}$:

$$P_i = \chi^{(1)}_{ij} E_j + \chi^{(2)}_{ijk} E_j E_k + \chi^{(3)}_{ijkl} E_j E_k E_l + \ldots,$$

где $\mathbf{E}$ — вектор электрического поля, $\chi^{(1)}$ — линейная диэлектрическая восприимчивость (поляризуемость) среды, $\chi^{(2)}$ и $\chi^{(3)}$ — нелинейная восприимчивость среды второго и третьего порядка (суть коэффициенты разложения поляризации среды $\mathbf{P}$ в ряд Тейлора по напряжённости поля $\mathbf{E}$). Нижние индексы обозначают компоненты тензоров, по повторяющимся индексам подразумевается суммирование. Поскольку $\mathbf{E}$ и $\mathbf{P}$ являются *истинными* векторами и меняют знак при инверсии $\hat{I}$, должно выполняться

$$\hat{I}\chi^{(2)} = -\chi^{(2)}. \tag{123}$$

С другой стороны, поскольку молекулы среды переходят сами в себя при инверсии $\hat{I}$, справедливо равенство

$$\hat{I}\chi^{(2)} = \chi^{(2)}. \tag{124}$$

Одновременное выполнение (123) и (124) возможно только при равенстве нулю всех компонентов тензора $\chi^{(2)}$, что и требовалось показать. Таким образом, в средах с центром инверсии кубический член является младшим нелинейным порядком разложения поляризации среды $\mathbf{P}$ по полю $\mathbf{E}$.

Строгий вывод уравнения (112) и более подробное рассмотрение описываемых им физических эффектов можно найти в монографии

[A14] (первое издание также есть в русском переводе [A13]). Альтернативный вывод нелинейного уравнения Шрёдингера без использования приближения медленно меняющейся амплитуды приведён в статье [A15].

4.5. Численные методы

Прежде чем переходить к рассмотрению различных методов численного интегрирования нелинейного уравнения Шрёдингера (112), обратим внимание на сходство его линейной части $iA_z = \frac{1}{2}\beta_2 A_{tt}$ с уравнением теплопроводности. Как мы помним из гл. 3, основная сложность построения численного решения уравнения теплопроводности была связана с построением устойчивых численных схем. В этой связи начнём наше рассмотрение именно с вопросов устойчивости.

Несложно заметить, что, в отличие от уравнения теплопроводности, спектр линейного оператора $\hat{D} = -\frac{i}{2}\beta_2\partial_t^2$ в правой части уравнения (112) целиком лежит на мнимой оси, что соответствует набегу фазы при распространении волн вместо диссипации энергии в задаче о теплопроводности. Как следствие, разностный аналог уравнения Шрёдингера напрямую не попадает под определение жёстких систем [1, с. 103]. Тем не менее, вполне очевидно, что появление множителя i не влияет на ключевые выводы, сделанные в гл. 3 для уравнения теплопроводности. А именно, при использовании простейших явных разностных схем для уравнения (112) шаг численного интегрирования по z должен выбираться достаточно малым для предотвращения взрывного роста амплитуды самой высокочастотной гармоники на сетке $\exp(i\omega_{\max}t)$. При этом проявляется характерная черта жёстких систем: малость шага h численного интегрирования определяется не масштабом, на котором эволюционирует физическое решение, но свойствами уравнения и сетки, используемой для построения численного решения.

Как и в случае уравнения теплопроводности, гармоника с частотой Найквиста характеризуется самым большим по модулю собственным значением:

$$\hat{D}e^{i\omega t} = \lambda e^{i\omega t}, \quad \max_{\omega}|\lambda| = \left|\frac{i}{2}\beta\omega_{\max}^2\right| = \frac{\pi^2|\beta_2|}{2\tau^2}, \tag{125}$$

где τ — шаг сетки по t. Максимальное собственное значение (125) быстро (квадратично) возрастает с уменьшением шага сетки τ. При использовании явных схем это приводит к необходимости соответствующего уменьшения шага h сетки по z, $h \propto \tau^2$. Ниже мы рассмотрим альтернативное решение — использование безусловно устойчивых численных

схем. Хотя для уравнения (112) можно использовать неявные схемы, подобно тому как это было сделано в гл. 3 для уравнения теплопроводности, гораздо интереснее рассмотреть два принципиально иных подхода, позволяющих построить эффективные безусловно устойчивые численные схемы.

4.5.1. Метод расщепления по физическим процессам

Запишем уравнение (112) в виде

$$\partial_z A = (\hat{D} + \hat{N})A, \tag{126}$$

где $\hat{D}$ и $\hat{N}$ — дисперсионный и нелинейный операторы соответственно:

$$\hat{D} = -\frac{i}{2}\beta_2\partial_t^2, \qquad \hat{N}A = i\gamma|A|^2A. \tag{127}$$

Будем обозначать посредством A и Ψ точное и численное решение уравнения (126) соответственно.

Основная идея метода расщепления по физическим процессам[33] заключается в разделении шага h численного интегрирования по координате z на две или более части, на каждой из которых учитывается только дисперсия либо только нелинейность. Похожий приём использовался нами в локально одномерном методе для решения уравнения теплопроводности (см. п. 3.7 на с. 82), хотя, как мы увидим далее, данные методы основаны на различных механизмах обеспечения устойчивости численного решения.

Самый простой способ продолжения численного решения из точки z до $z+h$ методом расщепления может быть записан следующим образом:

$$\Psi(z+h) = \exp\left(h\hat{D}\right)\exp\left(h\hat{N}\right)\Psi(z), \tag{128}$$

тогда как точное решение эволюционирует на шаге h по закону

$$A(z+h) = \exp\left(h\big(\hat{D}+\hat{N}\big)\right)A(z). \tag{129}$$

Ключевым моментом является *точный* учёт действия дисперсии в Фурье-представлении, см. п. 4.1 и формулу (113) на с. 86. Использование точной аналитической формулы вместо разностной схемы даёт абсолютную (не зависящую от шага сетки h) устойчивость численной схемы. Подставляя (113) в (128), имеем:

$$\Psi(z+h) = \hat{F}^{-}\exp\left(\frac{i}{2}\beta_2\omega^2 h\right)\hat{F}^{+}\exp\left(h\hat{N}\right)\Psi(z),$$

[33] Step-split Fourier method в англоязычной литературе.

где $\hat{F}^+$ и $\hat{F}^-$ — прямое и обратное дискретное преобразование Фурье. Применение эффективных программных реализаций алгоритмов быстрого преобразования Фурье позволяет выполнять вычисления быстрее, чем при использовании разностных схем.

В отличие от дисперсионного оператора, нелинейность не приводит к развитию неустойчивости, поэтому действие нелинейного оператора $\exp(h\hat{N})$ может быть реализовано любым способом: как точно, в соответствии с формулой (116), так и с помощью какой-либо разностной схемы для уравнения $\partial_z A = i\gamma|A|^2 A$.

Таким образом, для выполнения шага численного интегрирования в соответствии с (128) необходимо выполнить следующие шаги (символ $\leftarrow$ в выражениях ниже соответствует оператору присваивания в программе):

1) нелинейный шаг: $\Psi(t) \leftarrow \exp(h\hat{N})\Psi(t)$;

2) прямое преобразования Фурье: $\tilde{\Psi}(\omega) \leftarrow \hat{F}^+\,\Psi(t)$;

3) линейный шаг в Фурье-представлении: $\tilde{\Psi}(\omega) \leftarrow \tilde{\Psi}(\omega)\exp\left(\frac{i}{2}\beta_2\omega^2 h\right)$;

4) обратное преобразования Фурье: $\Psi(t) \leftarrow \hat{F}^-\,\tilde{\Psi}(\omega)$.

Исследуем погрешность схемы (128). В случае, если действие оператора $\exp(h\hat{N})$ в (128) вычисляется точно ($\Psi(t) \leftarrow \Psi(t)\exp(i\gamma|\Psi|^2 h)$), единственным источником ошибки (не считая неизбежных погрешностей округления) является замена оператора $\exp(h(\hat{D}+\hat{N}))$ на суперпозицию $\exp(h\hat{D})\exp(h\hat{N})$. Данная замена приводит к ошибке, связанной с некоммутативностью[34] операторов $\hat{D}$ и $\hat{N}$. Вычислим погрешность схемы (128) на одном шаге, вычитая из точного решения $A(z+h)$ численное, построенное в соответствии с (128):

$$R_1 = A(z+h) - e^{h\hat{D}}e^{h\hat{N}}A(z) = \left[e^{h(\hat{D}+\hat{N})} - e^{h\hat{D}}e^{h\hat{N}}\right]A(z).$$

Расписывая разность в квадратных скобках по определению через ряд Тейлора для экспоненты, имеем:

[34] Отличие от нуля коммутатора $[\hat{D},\hat{N}]$ легко понять из физических соображений: как мы видели в п. 4.1, действие дисперсии приводит к увеличению длительности импульсов и пропорциональному уменьшению их пиковой мощности P. Поскольку нелинейный набег фазы $\delta\varphi = \gamma P z$ пропорционален P, перестановка $\hat{D}$ и $\hat{N}$ будет приводить к изменению $\delta\varphi \Rightarrow \hat{D}\hat{N} \neq \hat{N}\hat{D}$, что и требовалось показать.

$$R_1 = \left[1 + h\left(\hat{D} + \hat{N}\right) + \frac{h^2}{2}\left(\hat{D} + \hat{N}\right)^2 + \ldots - \right.$$
$$\left. - \left(1 + h\hat{D} + \frac{h^2}{2}\hat{D}^2 + \ldots\right)\left(1 + h\hat{N} + \frac{h^2}{2}\hat{N}^2 + \ldots\right)\right] A(z) =$$
$$= \frac{h^2}{2}\left(\hat{N}\hat{D} - \hat{D}\hat{N}\right) A(z) + \mathcal{O}(h^3).$$

Таким образом, погрешность численной схемы (128) на одном шаге есть $\mathcal{O}(h^2)$. При интегрировании по отрезку длины $L = \text{const}$ с использованием L/h шагов погрешность возрастает до $\mathcal{O}(h)$, так что схема (128) обеспечивает первый порядок точности по шагу z-сетки.

Точность метода расщепления можно повысить за счёт использования симметричной схемы:

$$\Psi(z + h) = \exp\left(\frac{h}{2}\hat{D}\right) \exp\left(h\hat{N}\right) \exp\left(\frac{h}{2}\hat{D}\right) \Psi(z). \tag{130}$$

Расписывая аналогичным образом экспоненты от операторов через сумму рядов Тейлора, несложно показать, что погрешность симметричной схемы (130) на одном шаге есть:

$$R_1 = \left[e^{h(\hat{D}+\hat{N})} - e^{\frac{1}{2}h\hat{D}} e^{h\hat{N}} e^{\frac{1}{2}h\hat{D}}\right] A(z) = \mathcal{O}(h^3).$$

Следовательно, схема (130) имеет второй порядок точности по шагу сетки h. Абсолютная устойчивость схемы (130) достигается за счёт использования точного (аналитического) ответа для эволюции решения под действием дисперсии групповых скоростей $\hat{D}$.

Формально сравнивая формулы (128) и (130), можно прийти к выводу о том, что повышение порядка точности достигается ценой дополнительного действия дисперсионного оператора на каждом шаге. Однако в действительности объём вычислений при переходе от (128) к (130) почти не возрастает. В самом деле, для выполнения m шагов по формуле (130) необходимо подействовать на численное решение Ψ оператором

$$\hat{P}^{(2)}_{m\times h} = \left(e^{\frac{1}{2}h\hat{D}} e^{h\hat{N}} e^{\frac{1}{2}h\hat{D}}\right)^m = e^{\frac{1}{2}h\hat{D}} \left(e^{h\hat{N}} e^{h\hat{D}}\right)^{m-1} e^{h\hat{N}} e^{\frac{1}{2}h\hat{D}}. \tag{131}$$

Как видно из (131), группировка операторов $\exp(\frac{1}{2}h\hat{D})$ в конце и начале промежуточных шагов позволяет записать оператор эволюции $\hat{P}_{m\times h}$ в

виде суперпозиции m нелинейных и $m+1$ дисперсионных операторов. По сравнению со схемой первого порядка (128), m-кратное применение которой даёт оператор $(\exp(h\hat{D})\exp(h\hat{N}))^m$, произведение в формуле (131) содержит всего на один оператор больше и, соответственно, требует одного дополнительного быстрого преобразования Фурье, что почти незаметно при большом числе шагов m. Данная ситуация вполне аналогична переходу от квадратурной формулы правых (левых) прямоугольников к формуле трапеций, также позволяющей повысить на единицу порядок точности ценой всего одного дополнительного вычисления подынтегральной функции, см. [1, с. 59].

Реализовав в программном коде преобразования численного решения под действием операторов $\exp(h\hat{D})$ и $\exp(h\hat{N})$, совмещённые с вызовом функций для быстрого преобразования Фурье, достаточно просто запрограммировать метод расщепления четвёртого порядка точности по h [A15]. Для этого нужно, используя симметричную схему (130), сделать 4 шага h вперёд, один шаг $2h$ в обратном направлении и ещё 4 шага h вперёд. Соответствующее преобразование численного решения Ψ может быть записано в виде:

$$\Psi(z+6h) = \hat{P}^{(2)}_{4\times h}\hat{P}^{(2)}_{1\times(-2h)}\hat{P}^{(2)}_{4\times h}\Psi(z), \tag{132}$$

где оператор $\hat{P}^{(2)}_{m\times h}$ определён в соответствии с выражением (131). Группировка дисперсионных операторов в (132) позволяет сократить объём вычислений до 19 преобразований на шаге $6h$:

$$\hat{P}^{(4)}_{1\times 6h} = e^{\frac{h}{2}\hat{D}}\left(e^{h\hat{N}}e^{\frac{h}{2}\hat{D}}\right)^3 e^{h\hat{N}}e^{-\frac{h}{2}\hat{D}}e^{-2h\hat{N}}e^{-\frac{h}{2}\hat{D}}\left(e^{h\hat{N}}e^{\frac{h}{2}\hat{D}}\right)^3 e^{h\hat{N}}e^{\frac{h}{2}\hat{D}}.$$

Это приблизительно втрое увеличивает объём вычислений в расчёте на один шаг сетки h, однако позволяет повысить общую скорость моделирования благодаря увеличению h (сокращению числа шагов).

4.5.2. Переход к представлению взаимодействия

Рассмотрим ещё один подход к построению абсолютно устойчивой численной схемы для интегрирования нелинейного уравнения Шрёдингера (112). Данный метод [A16] аналогичен переходу к *представлению взаимодействия*, или представлению Дирака, используемому в квантовой механике наряду с представлениями Шрёдингера и Гейзенберга.

Основная идея, лежащая в основе данного метода: выполнить замену переменной и перейти к новой искомой функции, которая бы не

эволюционировала под действием дисперсионного оператора[35] $\hat{D}$:

$$A_I = \exp\big(-(z-z_0)\hat{D}\big)A, \tag{133}$$

где A_I — решение нелинейного уравнения Шрёдингера в представлении взаимодействия, $z_0 = \text{const}$ — точка, в которой представления совпадают. Дифференцирование (133) по z даст вид уравнения (112) в представлении взаимодействия:

$$\frac{\partial A_I}{\partial z} = \hat{N}_I A_I, \tag{134}$$

где

$$\hat{N}_I = e^{-(z-z_0)\hat{D}}\hat{N}e^{(z-z_0)\hat{D}}. \tag{135}$$

Исключение дисперсионного оператора $\hat{D}$ из уравнения (134) в представлении взаимодействия позволяет интегрировать (134) с помощью явных разностных схем, которые в данном случае являются устойчивыми. Вместе с тем переход между представлениями с помощью оператора $\exp(\pm h\hat{D})$ в (135) должен выполняться *точно*, путём выполнения быстрого преобразования Фурье, домножения на $\exp(\pm\frac{1}{2}i\beta_2\omega^2 h)$ и последующего обратного преобразования Фурье. В таком случае мы получим устойчивое численное решение, точность которого будет определяться точностью разностной схемы, используемой для интегрирования (134).

Рассмотрим алгоритм построения решения на примере трёх разностных схем, начав с наиболее простой схемы ломаных и переходя далее к схемам более высокого порядка точности. В каждом случае будем полагать, что численное решение $\Psi(z,t)$ известно нам в точке $z=a$, и покажем, как продолжить его в точку $z=a+h$.

Метод Эйлера. Положив $z_0 = a$ в (133), получим начальное условие для уравнения (134) в представлении взаимодействия: $\Psi_I(a) = \Psi(a)$. Используя для решения (134) метод Эйлера [1, с. 86], имеем:

$$\Psi_I(a+h) = \Psi_I(a) + h\hat{N}_I\Psi_I\big|_{z=a} = \Psi(a) + h\hat{N}\Psi(a).$$

Возвращаясь от представления взаимодействия к исходному[36] в соответствии с (133):

$$\Psi(a+h) = e^{h\hat{D}}\Psi_I(a+h) = e^{h\hat{D}}\big(1+h\hat{N}\big)\Psi(a). \tag{136}$$

[35] Напомним, что в квантовой механике делается аналогичная замена, позволяющая исключить из рассмотрения эволюцию решения под действием невозмущённого гамильтониана $\hat{H}_0$, что упрощает анализ эффектов, связанных с поправкой $\hat{H}_1$, описывающей взаимодействие в исследуемой системе.

[36] Аналог представления Шрёдингера в квантовой механике.

Обратим внимание, что выражение в круглых скобках с точностью до членов $\mathcal{O}(h^2)$ совпадает с $\exp(h\hat{N})$, а формула (136) в целом — с выражением (128).

Расчёт по формуле (136) требует однократного вычисления нелинейного оператора $\hat{N}$, однократного действия $\exp(h\hat{D})$ и пары дискретных преобразований Фурье на каждом шаге, что эквивалентно вычислительной сложности схемы (128).

Модифицированный метод Эйлера. Снова выберем $z_0 = a$ в (133), что даст $\Psi_I(a) = \Psi(a)$ в качестве начальных условий для (134). Для построения численного решения (134) будем использовать модифицированный метод Эйлера [1, с. 92]:

$$\Psi_I\left(a + \frac{h}{2}\right) = \Psi_I(a) + \frac{h}{2}\hat{N}_I\Psi_I\Big|_{z=a},$$

$$\Psi_I(a+h) = \Psi_I(a) + h\hat{N}_I\Psi_I\Big|_{a+\frac{h}{2}}.$$

С учётом (133) и (135) имеем:

$$\Psi(a+h) = e^{h\hat{D}}\Psi(a) + he^{\frac{1}{2}h\hat{D}}\hat{N}e^{\frac{1}{2}h\hat{D}}\left(\Psi(a) + \frac{h}{2}\hat{N}\Psi(a)\right). \tag{137}$$

Расчёт по формуле (137) требует двукратного вычисления нелинейного оператора $\hat{N}$, троекратного действия $\exp(h\hat{D})$ и шести дискретных преобразований Фурье на каждом шаге, что делает более предпочтительным использование формулы (130) и в особенности экономичной составной формулы (131), также обеспечивающих второй порядок точности по шагу сетки h.

Метод Рунге—Кутты четвёртого порядка.

При использовании для интегрирования уравнения (134) метода Рунге—Кутты четвёртого порядка [1, с. 95] удобно выбрать $z_0 = a + \frac{1}{2}h$. С учётом этого можем записать [A16]:

$$\begin{aligned}
&\Psi_I(a) = e^{\frac{h}{2}\hat{D}}\Psi(a),\\
&k_{1I} = he^{\frac{h}{2}\hat{D}}\hat{N}\Psi(a),\\
&k_{2I} = h\hat{N}\left(\Psi_I(a) + \frac{k_{1I}}{2}\right),
\end{aligned} \tag{138}$$

$$k_{3I} = h\hat{N}\left(\Psi_I(a) + \frac{k_{2I}}{2}\right),$$
$$k_4 = h\hat{N}e^{\frac{h}{2}\hat{D}}\left(\Psi_I(a) + k_{3I}\right), \qquad (139)$$
$$\Psi(x+h) = e^{\frac{h}{2}\hat{D}}\left(\Psi_I(a) + \frac{k_{1I}}{6} + \frac{k_{2I}}{3} + \frac{k_{3I}}{3}\right) + \frac{k_4}{6}.$$

Расчёт по формулам (138), (139) требует четырёхкратного вычисления нелинейного оператора $\hat{N}$ на каждом шаге, четырёхкратного вычисления дисперсионного оператора $\exp(\frac{1}{2}h\hat{D})$ и восьми преобразований Фурье. Скорость счёта с использованием данного метода сопоставима либо даже чуть выше [A16] скорости метода расщепления по физическим процессам (132), также имеющего четвёртый порядок точности по шагу сетки h.

Упражнения

1) Численно исследуйте распространение оптических солитонов в уравнении $i\partial_z A = \frac{1}{2}\partial_t^2 A - |A|^2 A$ с начальными условиями $A(0,t) = N/\mathrm{ch}\, t$ на различные расстояния $z < 10$. Проследите за неизменностью формы фундаментального солитона ($N = 1$) и осциллирующей динамикой солитонов целых порядков $N \geq 2$.

2) В предыдущей задаче выберите начальные условия в виде суммы (разности) двух фундаментальных солитонов

$$A(0,t) = \frac{1}{\mathrm{ch}\,(t - t_0)} \pm \frac{1}{\mathrm{ch}\,(t + t_0)},$$

разделённых небольшим временны́м интервалом $t_0 \approx 2 \ldots 3$. Наблюдайте притяжение (отталкивание) солитонов при распространении вдоль z.

3) Напишите программный код для интегрирования уравнения (112) по схеме (130), вычисляя действие дисперсионного и нелинейного операторов точно, как $\exp(h\hat{D})$ и $\exp(h\hat{N})$. Насколько изменится скорость работы программы, если вычислять комплексные экспоненты $\exp(h\hat{D})$ только на первом шаге h, используя на последующих шагах ранее вычисленные и сохранённые в памяти значения? Если заменить $\exp(h\hat{N})$ на $\cos\varphi + i(1 - \cos^2\varphi)^{1/2}$, а $\cos\varphi$ вычислять как сумму первых нескольких членов ряда Тейлора?

4) Интернет-трафик передаётся по оптическому волокну $\beta_2 = -2 \times 10^{-26}$ с^2/м, $\gamma = 1{,}3$ (Вт·км)$^{-1}$ последовательностями по 100 импульсов, форма огибающих которая близка к гауссовой $A_0 \exp(-\frac{1}{2}(t/\tau)^2)$, $\tau = 10^{-11}$ с, $|A_0|^2 = 10^{-3}$ Вт. Импульсы в последовательности изначально разделены временны́м интервалом 5×10^{-11} с. Фаза каждого импульса ($\arg\{A_0\}$) может принимать одно из четырёх значений 0, $\pi/2$, π, $-\pi/2$. Проследите за эволюцией мгновенной мощности $|A(t)|^2$ последовательности импульсов после прохождения расстояния $z = L_D$, $3L_D$, $10L_D$.

5) В условиях предыдущей задачи оцените отношение L/L_D для магистральной оптической волоконной линии связи длиной $L = 800$ км. Как выглядит временно́е распределение мощности передаваемой последовательности импульсов при $z = L$?

6) Для восстановления информации на принимающем конце линии связи (при $z = L$) необходима компенсация хроматической дисперсии групповых скоростей, что может быть учтено в моделировании путём домножения комплекснозначной функции $\hat{F}^+ A(L,t) = \tilde{A}(L,\omega)$ на $\exp(-\frac{i}{2}\beta_2\omega^2 L)$. В условиях задачи 4 постройте график мгновенной мощности $|A(L,t)|^2$ последовательности импульсов на выходе из волокна $L = 800$ км после компенсации дисперсии. Выполните декодирование сигнала — вычислите фазу $\arg\{A(L,t)\}$ в центре импульсов и восстановите передаваемую битовую последовательность.

7) Одним из ключевых факторов, ограничивающих пропускную способность линий связи, является нелинейность, приводящая к смешению различных спектральных и/или временны́х компонент передаваемого сигнала. Промоделируйте данный эффект, используя параметры из задачи 6. Постройте *констелляционную диаграмму*, отобразив точками на комплексной плоскости амплитуды $A(L)$ импульсов, принимаемых после прохождения линии связи и компенсации дисперсии. Увеличьте мощность импульсов до 2, 5, 10 мВт. Как изменится констелляционная диаграмма и почему? Какую роль играет разброс точек на диаграмме при передаче сигнала? Какие физические эффекты, не учитываемые в используемой модели, могут привести к увеличению разброса точек на диаграмме?

8) Четырёхпортовый сплавной волоконный ответвитель (рис. 17 (а)) представляет собой линейный пассивный оптический элемент, выполняющий унитарное преобразование амплитуд сигнала (A_1, A_2)

на своих входах 1, 2 в амплитуды сигналов $(A_{1'}, A_{2'})$ на двух выходных портах $1'$, $2'$:

$$\begin{pmatrix} A_{1'} \\ A_{2'} \end{pmatrix} = \begin{pmatrix} \cos\alpha & i\sin\alpha \\ i\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix},$$

где α — параметр ответвителя. При соединении портов $1'$ и $2'$ ответвителя отрезком оптического волокна длины L (рис. 17 (б)), получается нелинейное петлевое волоконное зеркало (Nonlinear optical loop mirror). Преобразование амплитуды сигнала A_1 на входе волоконного зеркала в выходной сигнал A_2 может быть описано выражением $A_2 = \cos\alpha\hat{P}\{\cos\alpha A_1\} - \sin\alpha\hat{P}\{\sin\alpha A_1\}$, где $\hat{P} = \exp(L(\hat{D}+\hat{N}))$ — оператор распространения сигнала в оптическом волокне длины L. Получите аналитическое выражение для зависимости коэффициента пропускания $T \equiv |A_2/A_1|^2$ от мгновенной мощности на входе $|A_1|^2$ в пределе $L_D \to \infty$. Постройте график зависимости коэффициента пропускания по энергии $T_W = (\int |A_2(t)|^2 dt)/(\int |A_1(t)|^2 dt)$ от пиковой мощности гауссовых импульсов длительностью 5×10^{-12} с, используя точный ответ в приближении $\beta_2 = 0$ и численный счёт для следующих параметров нелинейного петлевого зеркала: $\beta_2 = 2 \times 10^{-26}$ с^2/м, $\gamma = 1$ (Вт·км)$^{-1}$, $L = 100$ м, $\cos^2\alpha = 0{,}6$.

9) В условиях задачи 8 смените знак дисперсии на противоположный ($\beta_2 = -2 \times 10^{-26}$ с^2/м), а в качестве входных импульсов используйте $A_1(t) = A_0/\cosh(t/T)$ с различным уровнем мощности $|A_0|^2$. Выберите длительность импульсов T так, чтобы она была близка к длительности фундаментального солитона. Как изменилось максимальное значение коэффициента пропускания $\max\limits_{|A_0|^2}[T_W(|A_0|^2)]$ нелинейного петлевого зеркала и почему?

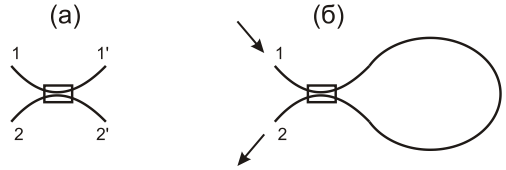

Рис. 17. (а) Четырёхпортовый сплавной волоконный ответвитель и (б) нелинейное петлевое волоконное зеркало

Рекомендованная литература

[1] *Смирнов С. В.* Основы вычислительной физики. Новосибирск : Новосибирский государственный университет, 2015. Часть 1. 113 с.

[2] *Калиткин Н. Н.* Численные методы. М. : Наука, 1978. 512 с.

[3] *Самарский А. А.* Введение в численные методы. М. : Наука, 1987. 286 с.

[4] *Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.* Numerical recipes The art of scientific computing. N. Y. : Cambridge University Press, 2007. 1235 p.

Дополнительная литература и Интернет-ресурсы

[A1] *Абрамов А. А., Андреев В. Б.* О применении метода прогонки к нахождению периодических решений дифференциальных и разностных уравнений // Вычисл. матем. и матем. физ. 1963. Т. 3, № 2. С. 377–381.

[A2] *Rice J. R.* Numerical methods, software, and analysis. San Diego : Academic Press, 2^{nd} ed., 1993. 720 p.

[A3] *Уилкинсон Дж. Х.* Алгебраическая проблема собственных значений. М. : Наука, 1970. 564 с.

[A4] *Cooley J. W., Tukey J. W.* An algorithm for the machine calculation of complex Fourier series // Math. of computation. 1965. V. 19, № 90. P. 297–301.

[A5] *Wilkinson, J.H., Reinsch, C.* Linear Algebra, vol. II of Handbook for Automatic Computation. N. Y. : Springer-Verlag, 1971. 439 p.

[A6] *Дасгупта С., Пападимитриу Х., Вазирани У.* Алгоритмы. М. : МЦНМО, 2014. 320 с.

[A7] *Rader C. M.* Discrete Fourier transforms when the number of data samples in prime // Proceedings of the IEEE. 1968. V. 56, №. 6. P. 1107–1108.

[A8] *Науссбаумер Г.* Быстрое преобразование Фурье и алгоритмы вычисления свёрток. М. : Радио и связь, 1985. 248 с.

[A9] FFTW homepage. URL: http://fftw.org/.

[A10] Температура воздуха около НГУ. URL: http://weather.nsu.ru/.

[A11] *Самарский А. А.* Об одном экономичном разностном методе решения многомерного параболического уравнения в произвольной области // Вычисл. матем. и матем. физ. 1962. Т. 2, № 5. С. 787–811.

[A12] *Захаров В. Е., Шабат А. Б.* Точная теория двумерной самофокусировки и одномерной самомодуляции волн в нелинейных средах // Эксп. и теор. физ. 1971. Т. 61. С. 118–134.

[A13] *Агравал Г.* Нелинейная волоконная оптика. Пер. с. англ. М. : Мир, 1996. 323 с.

[A14] *Agrawal G. P.* Nonlinear fiber optics. Oxford : Academic Press, 5^{th} ed., 2013. 648 p.

[A15] *Blow K. J., Wood D.* Theoretical description of transient stimulated Raman scattering in optical fibers // IEEE J. Quantum Electronics. 1989. V. 25, № 12. P. 2665–2673.

[A16] *Hult J.* A fourth-order Runge—Kutta in the interaction picture method for simulating supercontinuum generation in optical fibers // J. Lightwave Technology. 2007. V. 25, № 12. P. 3770–3775.

Учебное издание

Смирнов Сергей Валерьевич

ОСНОВЫ ВЫЧИСЛИТЕЛЬНОЙ ФИЗИКИ
Часть II

Учебное пособие

Редактор Я. О. Козлова
Обложка Е. В. Неклюдовой

Подписано в печать .09.2017
Формат 60x84 1/16. Уч.-изд. л. 7. Усл. печ. л. 6,1
Тираж 200 экз. Заказ №

Издательско-полиграфический центр НГУ
630090, г. Новосибирск, ул. Пирогова, 2