

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ
УНИВЕРСИТЕТ "МИФИ"

В.И. Ращиков

ЧИСЛЕННЫЕ МЕТОДЫ. КОМПЬЮТЕРНЫЙ ПРАКТИКУМ

*Рекомендовано УМО «Ядерные физика и технологии»
в качестве учебно-методического пособия
для студентов высших учебных заведений*

Москва 2010

УДК 519.2(075)
ББК 22.193я7
P28

Рашиков В.И. **Численные методы. Компьютерный практикум:**
Учебно-методическое пособие. – М.: НИЯУ МИФИ, 2010. – 132 с.

В данном пособии представлены основные численные методы решения физических задач: аппроксимация и интерполяция функций, численное интегрирование и дифференцирование, решение нелинейных уравнений и систем, задачи линейной алгебры, обыкновенные дифференциальные уравнения и уравнения в частных производных, методы оптимизации. Для иллюстрации каждого метода подобрано большое число типовых задач, наиболее часто встречающихся в инженерно-физических расчетах. Приведенные блок-схемы программ и практические рекомендации по их написанию, позволяют детально разобраться в алгоритме решения задачи и облегчить процесс программирования.

Пособие предназначено для студентов дневного и вечернего факультетов МИФИ, а также может быть полезно студентам других вузов физического профиля.

Подготовлено в рамках Программы создания и развития НИЯУ МИФИ.

Рецензент канд. физ.-мат. наук, доцент *М.В. Скачков*

ISBN 978-5-7262-1223-4

© *Национальный исследовательский
ядерный университет «МИФИ», 2010*

СОДЕРЖАНИЕ

Предисловие.....	4
Задание № 1. Анализ последовательности данных	7
Задание № 2. Решение нелинейных уравнений	13
Задание № 3. Интерполирование	21
Задание № 4. Аппроксимация	27
Задание № 5. Численное дифференцирование.....	33
Задание № 6. Численное интегрирование	40
Задание № 7. Вычисление кратных интегралов методом Монте-Карло	48
Задание № 8. Решение систем линейных алгебраических уравнений	54
Задание № 9. Частичная проблема собственных значений	65
Задание № 10. Поиск минимума функции одной переменной.....	71
Задание № 11. Поиск минимума функции двух переменных.....	78
Задание № 12. Численное решение задачи Коши для обыкновенных дифференциальных уравнений.	83
Задание № 13. Численное решение линейной краевой задачи для обыкновенного дифференциального уравнения второго порядка	89
Задание № 14. Численное решение линейного уравнения переноса	95
Задание № 15. Численное решение одномерного уравнения теплопроводности	102
Задание № 16. Численное решение уравнения теплопроводности в прямоугольнике	109
Задание № 17. Численное решение одномерного волнового уравнения	116
Задание № 18. Численное решение уравнения Пуассона в прямоугольнике	123
Библиографический список	131

ПРЕДИСЛОВИЕ

Данное учебно-методическое пособие составлено на основе многолетнего опыта проведения автором практических занятий и чтения лекций по курсу «Численные методы» на дневном и вечернем факультетах МИФИ. Компьютерный практикум является одним из основных факторов для практического овладения численными методами решения физических задач, в силу чего для него выделена большая часть отведенного курса учебного времени.

Пособие состоит из 18 заданий, которые охватывают практически все основные разделы курса: интерполяция и аппроксимация функций, численное интегрирование и дифференцирование, решение нелинейных уравнений и систем, задачи линейной алгебры, обыкновенные дифференциальные уравнения и уравнения в частных производных, численные методы поиска минимума функций одной и нескольких переменных.

В каждом задании приводятся необходимые для понимания метода теоретические сведения, варианты задач для самостоятельного выполнения и практические рекомендации по составлению программ, подкреплённые блок-схемами вычислительных алгоритмов. В конце каждого задания приведен список контрольных вопросов, позволяющих проверить степень усвоения изучаемого материала.

Блок-схемы нужны для более наглядного представления алгоритма задачи, что позволяет существенно облегчить понимание метода решения и написания самой программы. При выполнении схем использовался ГОСТ 19.701-90 [1], который регулирует правила построения схем и внешний вид их элементов. Основные элементы, которые в дальнейшем будут использоваться в пособии, имеют следующий вид:



– терминатор; элемент отображает вход из внешней среды или выход из нее (наиболее частое применение – начало и конец программы). Внутри фигуры записывается соответствующее действие;



– процесс; выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположения). Внутри фигуры записывают непосредственно

сами операции;



– решение; отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления

условий, определенных внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Используется для иллюстрации условных операторов *if* (два выхода: *true*, *false*) и *case* (множество выходов);



– предопределённый процесс; символ отображает выполнение процесса, состоящего из одной или нескольких операций, который определен в другом месте программы (в подпрограмме, модуле). Внутри

символа записывается название процесса и передаваемые в него данные. Используется для обозначения вызова процедуры или функции;



– данные (ввод-вывод); преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Данный символ не определяет носителя данных (для указания

типа носителя данных используются специфические символы);




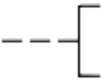
– граница цикла; символ состоит из двух частей – соответственно начало и конец цикла, операции, выполняемые внутри цикла, размещаются между



ними. Условия цикла и приращения записываются внутри символа начала или конца цикла в зависимости от типа организации цикла. Часто для изображения на блок-схеме цикла вместо данного символа используют символ решения, указывая в нем

условие, а одну из линий выхода замыкают выше в блок-схеме (перед операциями цикла);

 – соединитель; символ отображает выход в часть схемы и вход из другой части этой схемы. Используется для обрыва линии и продолжения ее в другом месте (пример: разделение блок-схемы, не помещающейся на листе);

 – комментарий; используется для более подробного описания шага, процесса или группы процессов. Описание помещается со стороны квадратной скобки и охватывается по всей высоте. Пунктирная линия идет к описываемому элементу или группе элементов (при этом группа выделяется замкнутой пунктирной линией). Также символ комментария используется в тех случаях, когда объем текста в каком-либо другом символе (например, символ процесса, символ данных и др.) превышает его объем.

Порядок выполнения действий задается путем соединения вершин, что позволяет рассматривать блок-схемы не только как наглядную интерпретацию алгоритма, удобную для восприятия человеком, но и как ориентированный граф.

При написании данного пособия использовался в основном материал из [2–4]. Для более полного изучения численных методов в библиографическом списке приведены необходимые учебные пособия [5–8].

Задание № 1

АНАЛИЗ ПОСЛЕДОВАТЕЛЬНОСТИ ДАННЫХ

Цель работы — построение развилки и циклических конструкций в программах, составление программ анализа потоков данных.

ПОСТАНОВКА ЗАДАЧИ

В работе анализируется последовательность, имитирующая поток экспериментальных данных. Пусть $f(x_i)$ — экспериментальная зависимость, снятая на отрезке $[a, b]$ с фиксированным шагом

$$h = \frac{b-a}{N-1},$$

где N — число точек в экспериментальной зависимости.

Значения абсцисс этих точек определим по формуле

$$x_i = a + ih, \quad i = 0, 1, \dots, n.$$

Необходимо вычислить:

1) максимальное значение функции $f_{\max} = \max_i f_i$ и номер узла i_{\max} , в котором достигается это значение;

2) минимальное значение функции $f_{\min} = \min_i f_i$;

3) среднее значение \bar{f} , средний квадрат \bar{f}^2 и среднеквадратичное значение f_m функции:

$$\bar{f} = \frac{1}{n+1} \sum_{i=0}^n f_i, \quad \bar{f}^2 = \frac{1}{n+1} \sum_{i=0}^n f_i^2, \quad f_m = \sqrt{\bar{f}^2};$$

4) относительное число положительных p_+ и отрицательных p_- значений функции $f_i (i = 0, 1, \dots, n)$

$$p_{\pm} = n_{\pm} / (n+1),$$

где n_+ и n_- — числа положительных и отрицательных значений $f_i (i = 0, 1, \dots, n)$;

5) среднеквадратичное отклонение от среднего значения

$$\sigma = \sqrt{\frac{1}{n+1} \sum_{i=0}^n (f_i - \bar{f})^2}.$$

ВАРИАНТЫ ФУНКЦИЙ $f(x)$

- 1) $f(x) = -\cos^k(\pi x^m / 2) + 2x^j$;
- 2) $f(x) = \sin^k(\pi x^m / 2) - (1-x)^j$;
- 3) $f(x) = \operatorname{sh} x^j - \cos^k(\pi x^m)$;
- 4) $f(x) = 1 - x^j - \operatorname{tg}^k(\pi x^m / 4)$;
- 5) $f(x) = (1-x)^j - \operatorname{tg}^k(\pi x^m / 4)$;
- 6) $f(x) = (1-x)^{1/j} - \operatorname{tg}^k(\pi x^m / 4)$;
- 7) $f(x) = e^{-x^j} - \sin(\pi x^m / 2)$;
- 8) $f(x) = e^{-x^j} - x^k \sin(\pi x^m / 2)$;
- 9) $(1-x)^m - 2x^j$;
- 10) $(1-x)^{1/m} - 2x^{1/j}$;
- 11) $f(x) = (1-x)^j \operatorname{ch}^k x^m - x^l$;
- 12) $f(x) = (1-x)^j \cos^k(\pi x^m / 2) - x^l$;
- 13) $f(x) = (1-x)^j - \operatorname{sh}^k x^m$;
- 14) $f(x) = (1-x)^{1/j} - \operatorname{sh}^k x^m$;
- 15) $f(x) = (1-x)^j - \operatorname{sh}^{1/k} x^m$;
- 16) $f(x) = (1-x)^j \operatorname{ch}^k x^m - x^l$;
- 17) $f(x) = \arcsin x^m - (1-x^j)^k$;
- 18) $f(x) = k \cos(\pi x^m) + x^l(1-x)$;
- 19) $f(x) = \ln^m(1+x^j) - (1-x)^k$;
- 20) $f(x) = \ln^m(1+x^j) - (1-x)^k \operatorname{ch} x^l$.

В функциях параметры l, k, j, m принимают значения от 1 до 4, $0 \leq x \leq 1$, рекомендуемые значения n от 50 до 1000.

ПРОГРАММИРОВАНИЕ

Практически во всех современных реализациях универсальных языков программирования, таких, как Фортран, Си, Паскаль, присутствуют одинаковые структурные элементы, с помощью которых строятся программы.

Основными структурами, которые изучаются в настоящей программе, являются развилка и цикл. Развилку можно представить в блок-схеме как показано на рис. 1.1.

Здесь P (решение) – некоторое логическое условие, которое может принимать значение «истина» (Да, True) или «ложь» (Нет, False). В зависимости от этого будет выполнен либо блок операторов A , либо блок операторов B .

Цикл может быть реализован либо с помощью развилки, либо с помощью специальных операторов цикла с условием (условие выполнения операторов тела цикла проверяется при входе в цикл), с постусловием (условие выполнения операторов тела цикла проверяется при выходе из цикла), со счетчиком (переменная, называемая счетчиком цикла, меняется с заданным шагом, пока не достигнет фиксированного значения). На блок-схеме цикл представляется либо развилкой (рис. 1.2), либо состоящим из двух частей символом, отображающим начало и конец цикла (рис. 1.3). Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т.д. помещаются внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие.

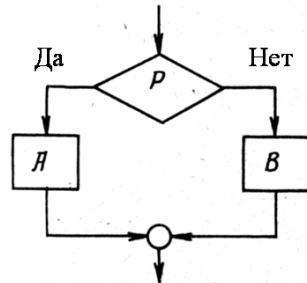


Рис. 1.1. Блок-схема развилки («решение», «выбор»)

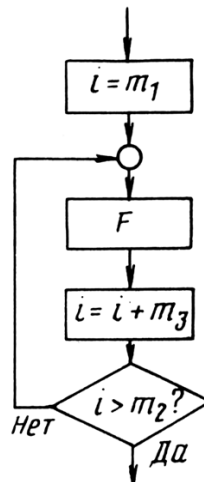


Рис. 1.2. Схема цикла с развилкой

В примере на рис. 1.2 i – простая переменная целого типа, называемая переменной цикла; m_1 – начальное значение переменной цикла, m_3 — шаг изменения i , а m_2 определяет конечное значение i , F – тело цикла.

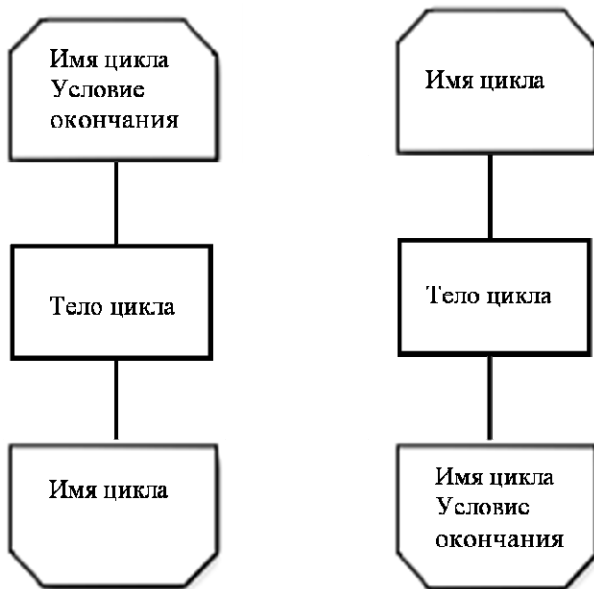


Рис. 1.3. Схема цикла со специальным символом

В нашем задании вычисленные в узлах x_i значения функции u_i следует поместить в одномерный массив, заранее описав его тип и размерность. Блок-схема программы представлена на рис. 1.4.

В блоке 2 вводятся начальные данные, 3 - 4 - 5 – цикл вычисления основных величин, за исключением среднеквадратичного отклонения от среднего, вычисление которого выделено в отдельный цикл 6 - 7 - 8, так как оно опирается на результаты предыдущих операций. В блоке 9 результаты приводятся к необходимому виду и в 10 осуществляется их вывод.

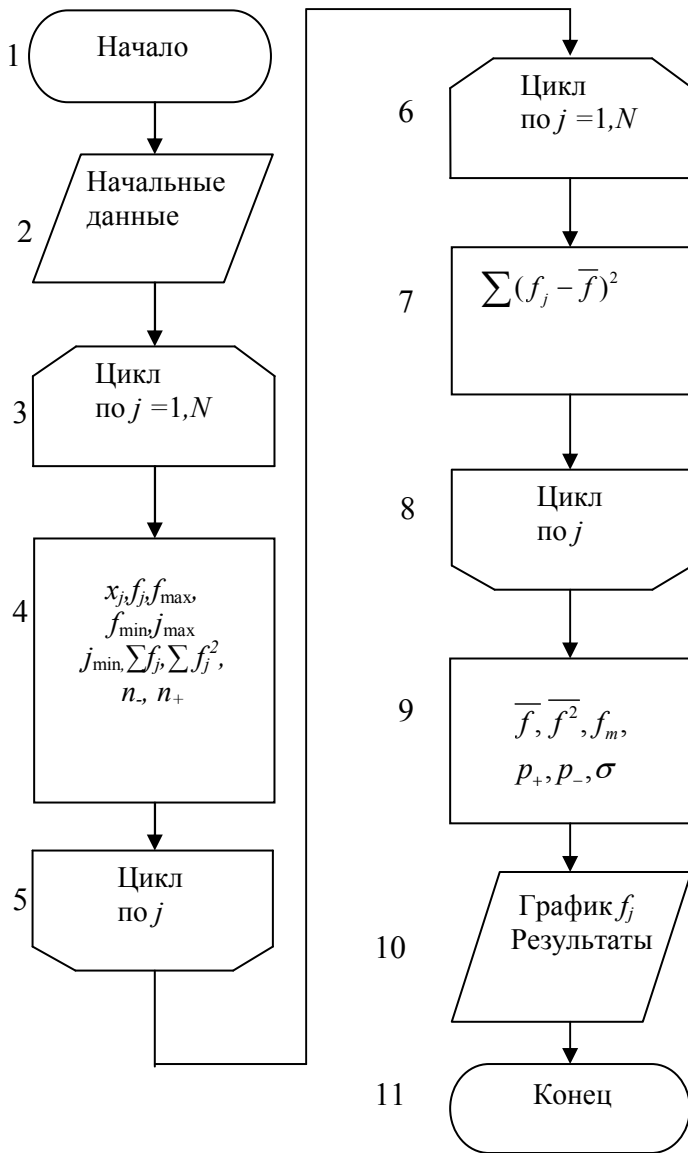


Рис. 1.4. Блок-схема программы анализа потока данных

Для проверки правильности программы рекомендуется предварительно исследовать тестовую функцию

$$u(x) = x(1-x) - 1/8, \quad N = 101,$$

для которой должны получиться следующие результаты:

$$u_{\max} = 0.125, \quad j_{\max} = 51, \quad u_{\min} = -0.125, \quad j_{\min} = 1;$$

$$\bar{u} = 0.0400, \quad \overline{u^2} \approx 0.0074, \quad u_m \approx 0.0859,$$

$$p_- \approx 0.2970, \quad p_+ \approx 0.7030, \quad \sigma \approx 0.0760.$$

Рекомендуется выполнить вычисления для нескольких значений n и проанализировать, как при этом изменяются результаты.

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы, параметры и график функции $u(x)$ для конкретного варианта;
- текст программы;
- результаты расчётов.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как описываются массивы?
2. Как записывается и выполняется оператор цикла?
3. Какие ограничения накладываются на оператор цикла?
4. Как записываются и выполняются операторы ввода и вывода информации?

Задание № 2

РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Цель работы: изучение условно и безусловно сходящихся итерационных методов решения нелинейных уравнений.

ПОСТАНОВКА ЗАДАЧИ

Одной из наиболее частых задач, с которыми сталкивается физик, — это решение уравнений вида

$$f(x) = 0. \quad (2.1)$$

Решения ищутся методами последовательных приближений или итерационными методами. Начальное приближение может находиться из физических соображений, из опыта решения аналогичных задач, с помощью графических методов и т. д.

Поиск корня уравнения математически осуществляется при помощи построения последовательности Коши $\{x_i\}$, когда при заданном ε существует такое N , что для всех n и p , превышающих N , выполняется $|x_n - x_p| < \varepsilon$, причем корень находится внутри этого отрезка неопределённости.

Метод половинного деления (дихотомия)

Пусть известно, что на отрезке $[a, b]$ находится искомое значение корня уравнения (2.1), т. е. $x \in [a, b]$ (рис. 2.1). В качестве начального приближения возьмем середину отрезка $x_0 = \frac{a+b}{2}$.

Теперь исследуем значения функции $f(x)$ на концах образовавшихся отрезков $[a, x_0]$ и $[x_0, b]$.

Выберем из них тот, на концах которого функция принимает значения разного знака, так как

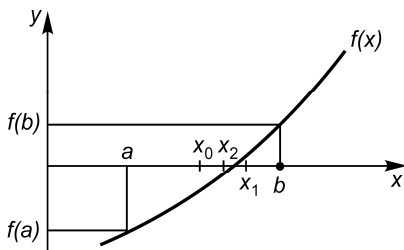


Рис. 2.1. Иллюстрация метода дихотомии

он и содержит искомый корень. Вторую половину отрезка можно не рассматривать. Затем делим новый отрезок пополам и приходим вновь к двум отрезкам, на концах одного из которых функция меняет знак, т. е. содержит корень. Таким образом, после каждой итерации исходный отрезок сокращается вдвое, т. е. после n итераций он сократится в 2^n раз. Процесс итераций будет продолжаться до тех пор, пока значение модуля функции не окажется меньше заданной точности ε , т. е. $|f(x_n)| < \varepsilon$, или длина исследуемого отрезка станет меньше удвоенной допустимой δ : $|x_{n+1} - x_n| < 2\delta$. Тогда в качестве приближенного значения корня можно принять середину последнего отрезка $0,5(x_n + x_{n+1})$.

Как видно из сказанного, метод довольно медленный, однако он безусловно сходящийся, т. е. всегда сходится к корню.

Метод касательных

Метод касательных (метод Ньютона) можно отнести к методам, в основе которых лежит замена исследуемой функции $f(x)$ более простой функцией, в данном случае касательной.

Геометрически в начальной точке x_0 проводится касательная к кривой $f(x)$ и находится точка ее пересечения с осью абсцисс (рис. 2.2).

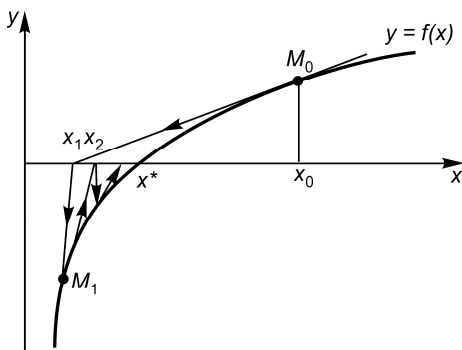


Рис. 2.2. Иллюстрация метода касательных

Уравнение касательной к кривой $f(x)$ в точке $M_0(x_0, f(x_0))$ имеет вид

$$y - f(x_0) = f'(x_0)(x - x_0),$$

а следующее приближение x_1 , являющееся точкой пересечения касательной с осью абсцисс, дается формулой

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Аналогичным образом можно найти и приближения

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

строая касательные последовательно из точек M_1, \dots, M_{n-1} , не забывая, что $f'(x_n) \neq 0$.

Метод касательных является условно сходящимся методом, то есть для его сходимости

$$\lim_{i \rightarrow \infty} x_i = x^*$$

должно быть выполнено следующее условие в области поиска корня

$$|ff''| < (f')^2,$$

x^* - искомое значение корня.

При произвольном нулевом приближении итерации будут сходиться, если всюду будет выполнено полученное выше условие. В противном случае сходимость будет лишь в некоторой окрестности корня.

Для окончания итерационного процесса могут быть использованы следующие критерии.

1. Максимальное число итераций. Этот критерий необходим в случае, если методы не сходятся. Тем не менее трудно заранее определить, сколько итераций будет необходимо для получения удовлетворительной точности.

2. Слабая вариация приближения к корню:

$$|x_{n+1} - x_n| < \varepsilon \text{ или } |x_{n+1} - x_n| < \varepsilon |x_n|. \quad (2.2)$$

3. Достаточно малое значение функции $|f(x_n)| < \varepsilon$.

Метод Ньютона обладает сходимостью второго порядка. Это означает, что вблизи корня погрешность уменьшается по закону

$$\varepsilon_i = \text{const} \cdot \varepsilon_{i-1}^2.$$

Поэтому итерации по методу Ньютона сходятся очень быстро, так что для достижения условия (2.2) достаточно нескольких итераций.

Невыполнение условия (2.2) при $i > 10$ обычно указывает на отсутствие сходимости, ошибку в формулах или в программе.

Метод секущих

Вычисление производной функции $f(x)$, необходимой в методе Ньютона, не всегда удобно или возможно. Замена производной первой разделенной разностью, которую находят по двум последним итерациям (т. е. замена касательной на секущую) приводит нас к методу секущих. С точки зрения аналитических методов, в качестве аппроксимирующей взята прямая, проходящая через две последние точки x_n и x_{n-1} , т. е. вместо производной в методе касательных необходимо подставить

$$f' = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}},$$

тогда придем к формуле метода секущих:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n).$$

Метод секущих является двухшаговым методом, т. е. требует двух начальных (разгонных) точек x_0 и x_1 . Графически метод иллюстрируется рис. 2.3.

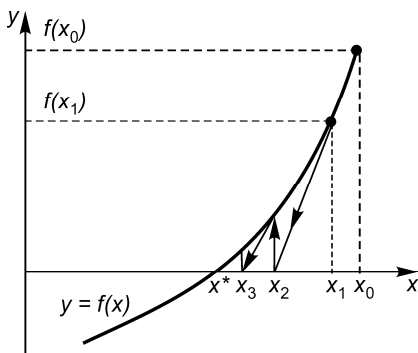


Рис. 2.3. Иллюстрация метода секущих

Сначала через выбранные точки $(x_0, f(x_0))$, $(x_1, f(x_1))$ проводим прямую до пересечения с осью абсцисс и определяем x_2 , а вертикальная прямая в точке x_2 дает $f(x_2)$. Далее прямая проводится через точки $(x_1, f(x_1))$ и $(x_2, f(x_2))$ и т. д., пока не будет выполнено одно из трех условий окончания итерационного процесса (2.2).

Обычно в методе секущих требуется больше итераций, чем в методе касательных, но зато каждая итерация выполняется значительно быстрее, так как не требуется вычислять $f'(x)$, и поэтому часто при том же объеме вычисле-

ний можно сделать больше итераций и получить более высокую точность.

ВАРИАНТЫ ЗАДАНИЙ

Используя безусловно сходящийся метод дихотомии и один из условно сходящихся методов (касательных или секущих), найти на отрезке $0 \leq x \leq 1$ корень одной из функции, приведенных ниже. В функциях параметры l, k, j, m принимают значения от 1 до 4. Рекомендуется исследовать ту же функцию, что и в задании № 1.

- 1) $f(x) = -\cos^k(\pi x^m / 2) + 2x^j$;
- 2) $f(x) = \sin^k(\pi x^m / 2) - (1-x)^j$;
- 3) $f(x) = \operatorname{sh} x^j - \cos^k(\pi x^m)$;
- 4) $f(x) = 1 - x^j - \operatorname{tg}^k(\pi x^m / 4)$;
- 5) $f(x) = (1-x)^j - \operatorname{tg}^k(\pi x^m / 4)$;
- 6) $f(x) = (1-x)^{1/j} - \operatorname{tg}^k(\pi x^m / 4)$;
- 7) $f(x) = e^{-x^j} - \sin(\pi x^m / 2)$;
- 8) $f(x) = e^{-x^j} - x^k \sin(\pi x^m / 2)$;
- 9) $(1-x)^m - 2x^j$;
- 10) $(1-x)^{1/m} - 2x^{1/j}$;
- 11) $f(x) = (1-x)^j \operatorname{ch}^k x^m - x^l$;
- 12) $f(x) = (1-x)^j \cos^k(\pi x^m / 2) - x^l$;
- 13) $f(x) = (1-x)^j - \operatorname{sh}^k x^m$;
- 14) $f(x) = (1-x)^{1/j} - \operatorname{sh}^k x^m$;
- 15) $f(x) = (1-x)^j - \operatorname{sh}^{1/k} x^m$;
- 16) $f(x) = (1-x)^j \operatorname{ch}^k x^m - x^l$;
- 17) $f(x) = \arcsin x^m - (1-x^j)^k$;
- 18) $f(x) = k \cos(\pi x^m) + x^l(1-x)$;
- 19) $f(x) = \ln^m(1+x^j) - (1-x)^k$;
- 20) $f(x) = \ln^m(1+x^j) - (1-x)^k \operatorname{ch} x^l$.

ПРОГРАММИРОВАНИЕ

При составлении программы целесообразно задаться максимально допустимым числом итераций i_{\max} , прерывая итерационный процесс, если $i = i_{\max}$. Это предохраняет от так называемого «зацикливания» программы, которое иногда случается вследствие ошибок в формулах или программе, а также при неудачном выборе начальной итерации. В данной задаче достаточно положить $i_{\max} = 30$, так как при отсутствии ошибок сходимость достигается гораздо раньше. Значение ε рекомендуется выбирать в диапазоне $10^{-5} - 10^{-7}$.

В качестве начальной итерации можно принять $x_0 = 0,2$. Если же итерации не сойдутся, это значение можно уменьшить или увеличить, оставаясь в диапазоне $0 < x_0 < 1$.

Блок-схемы программ решения нелинейных уравнений методом дихотомии и секущих приведены на рис. 2.4 и 2.5. Они составлены так, чтобы на каждой итерации вычислялось лишь одно новое значение $f(x)$. Это обеспечивает максимальную скорость решения, так как основное время решения занимает вычисление значений функции $f(x)$. В качестве «разгонных» значений в методе секущих можно принять $x_0 = 0,2$; $x_1 = x_0 + (10 - 100)\varepsilon$.

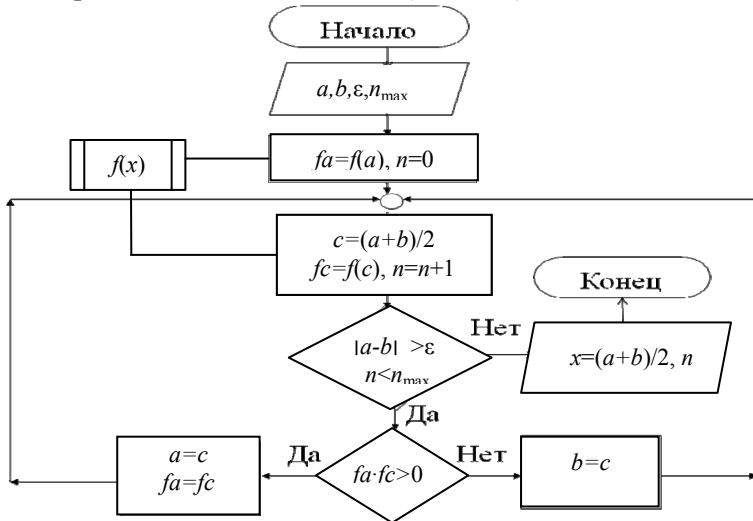


Рис. 2.4. Блок-схема программы решения нелинейного уравнения методом дихотомии

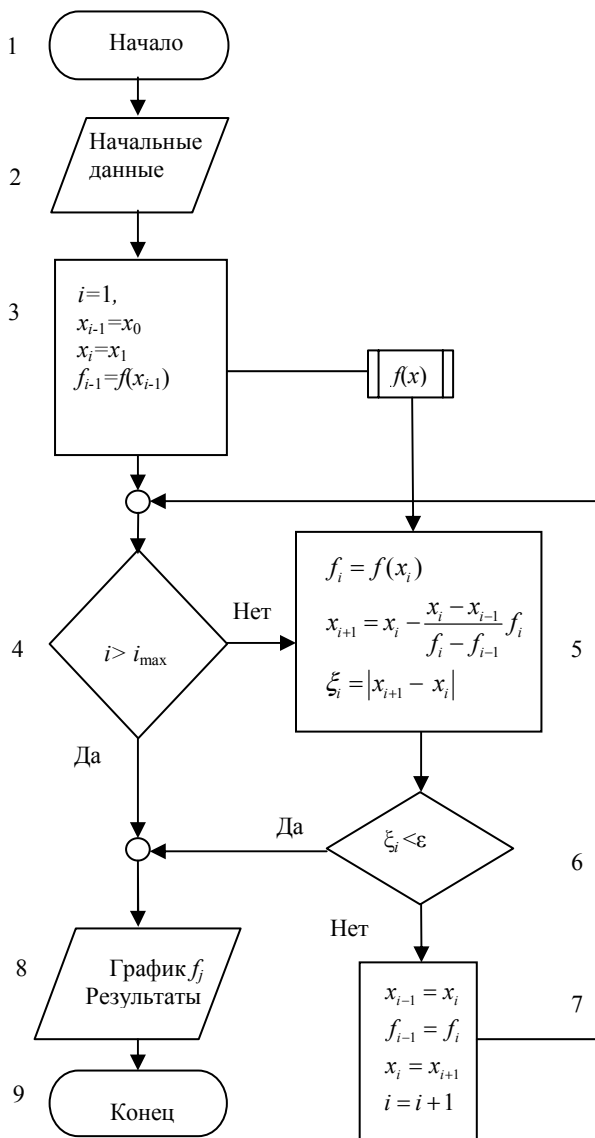


Рис. 2.5. Блок-схема программы решения нелинейного уравнения методом секущих

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулу функции $f(x)$ для конкретного варианта;
- заданное значение ε и начальные значения x_0 ;
- текст программы;
- найденные приближенные значения корня и число итераций для обоих методов;
- построенный в предыдущем задании график функции $f(x)$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как строится решение нелинейных уравнений методом касательных, каковы его характеристики?
2. Получите условие сходимости метода касательных.
3. Получите оценку скорости (порядка) сходимости метода касательных.
4. Как строится решение нелинейных уравнений методом секущих, каковы его характеристики?
5. Какие еще существуют методы решения нелинейных уравнений?
6. Как строится решение нелинейных уравнений методом дихотомии, каковы его характеристики?
7. Сравните методы решения нелинейных уравнений по скорости сходимости на примере полученных вами результатов.

Задание № 3

ИНТЕРПОЛИРОВАНИЕ

Цель работы — изучение методов интерполирования, построение интерполяционного многочлена Ньютона.

ПОСТАНОВКА ЗАДАЧИ

Численное моделирование большинства физических задач сопряжено, как правило, с необходимостью учета факторов, которые не могут быть описаны аналитически. Имеется лишь ряд экспериментальных зависимостей, полученных в фиксированном числе точек интересующего нас диапазона переменных. Так, при решении широко распространенной задачи о динамике макро- и микрообъектов во внешних гравитационных или электромагнитных полях информацию о поле часто бывает невозможно получить в виде аналитических функций без ввода дополнительных упрощающих предположений, которые могут существенно повлиять на результат. В этом случае необходимо прибегнуть к экспериментальным характеристикам, причем эксперимент может быть проведен лишь конечное число раз. Таким образом, мы приходим к физической задаче, в которой ряд функций задан на конечном числе точек x_i фиксированной области изменения аргумента $x_i \in [a, b]$. Численный метод, однако, может потребовать знания этих функций для всех значений аргумента из этой области. В этом случае возникает задача восстановления функции $y(x)$ для всех значений $x \in [a, b]$, если известны ее значения в некотором фиксированном числе точек x_i этого отрезка.

Наиболее простым и распространенным способом решения этой задачи является интерполяция между соседними значениями, которая сводится к построению функции $\varphi(x)$, совпадающей с функцией $y(x)$ в точках x_i , т. е.

$$\varphi(x_i) = y(x_i) = y_i, \quad i = 0, 1, 2, \dots, n,$$

где $n + 1$ — число заданных на отрезке $[a, b]$ точек, а x_i — узлы интерполяции.

При выборе интерполирующей функции $\varphi(x)$ необходимо ограничить поиск функциями, легко и быстро вычисляющимися на компьютере, так как их, как правило, приходится вычислять многократно.

Существует много интерполяционных многочленов и способов их построения, пригодных для различного расположения узлов. При построении интерполяционных многочленов обычно подразумевается, что множество используемых узлов известно. Однако часто бывает известна лишь требуемая точность, а число узлов не фиксировано.

Интерполяционный полином Ньютона, изучению которого и посвящена данная работа, отличается тем, что число используемых узлов можно легко увеличить или уменьшить без повторения всего цикла вычислений, изменяя тем самым точность интерполяции.

Интерполирование производится по таблице с равноотстоящими узлами, хотя интерполяционный многочлен Ньютона применим при любом расположении узлов.

Задание включает следующие этапы.

1. Вычислить таблицу значений $y_j = y(x_j)$ заданной функции $y(x)$ в равноотстоящих узлах

$$x_j = jh \quad (j = 0, 1, \dots, n), \quad h = 1/n,$$

отрезка $[0, 1]$.

2. Составить таблицу первых разностей функции

$$y(x_{j+1}, x_j) = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} = \frac{y_{j+1} - y_j}{h} \quad (j = 0, 1, \dots, n-1).$$

3. Составить таблицу вторых разделённых разностей

$$y(x_{j+2}, x_{j+1}, x_j) = \frac{y(x_{j+2}, x_{j+1}) - y(x_{j+1}, x_j)}{x_{j+2} - x_j} = \frac{y_{j+2} - 2y_{j+1} + y_j}{2h^2}$$

$$(j = 0, 1, \dots, n-2).$$

4. По этим таблицам, используя интерполяционный многочлен Ньютона второго порядка

$$\begin{aligned} P(x) &= y_j + (x - x_j)y(x_{j+1}, x_j) + (x - x_j)(x - x_{j+1})y(x_{j+2}, x_{j+1}, x_j) = \\ &= y_j + (x - x_j)\frac{y_{j+1} - y_j}{h} + (x - x_j)(x - x_{j+1})\frac{y_{j+2} - 2y_{j+1} + y_j}{2h^2}, \end{aligned}$$

вычислить значения $P(x)$ в точках (узлах) с полуцелыми индексами

$$x_{j+1/2} = (j+1/2)h \quad (j = 0, 1, \dots, n-2).$$

5. Найти погрешность интерполирования в этих узлах

$$\varepsilon_{j+1/2} = \left| y(x_{j+1/2}) - P(x_{j+1/2}) \right| \quad (j = 0, 1, \dots, n-2),$$

максимальную погрешность ε_{\max} , средний квадрат погрешности и среднеквадратичную погрешность ε_m :

$$\varepsilon_{\max} = \max_j \left| \varepsilon_{j+1/2} \right|, \quad \overline{\varepsilon^2} = \frac{1}{n-1} \sum_{j=0}^{n-2} \varepsilon_{j+1/2}^2, \quad \varepsilon_m = \sqrt{\overline{\varepsilon^2}}.$$

6. Исследовать, как изменяются погрешности ε_{\max} и ε_m с изменением n .

ВАРИАНТЫ ФУНКЦИЙ $y(x)$

$$(a, b = 1 \div 5; \quad k, m = 1 \div 4; \quad n = 20 \div 100)$$

1) $y(x) = \sin^k(\pi x^m)$;

2) $y(x) = \cos^k(\pi x^m)$;

3) $y(x) = \sin^k(\pi x^{1/m})$;

4) $y(x) = \cos^k(\pi x^{1/m})$;

5) $y(x) = \operatorname{tg}^k(\pi x^m / 4)$;

6) $y(x) = \operatorname{tg}^k(\pi x^{1/m} / 4)$;

7) $y(x) = \pm ax^4 \pm bx^3$;

8) $y(x) = (a + bx^m)^{-k}$;

9) $y(x) = (a + bx^m)^{-1/k}$;

10) $y(x) = (a + bx^{1/m})^{-k}$;

11) $y(x) = (a + bx^{1/m})^{-1/k}$;

12) $y(x) = x^k / (a + bx^m)$;

13) $y(x) = x^k / (a + bx)^m$;

14) $y(x) = x^{1/k} / (a + bx)^m$;

15) $y(x) = x^k / (a + bx)^{1/m}$;

16) $y(x) = x^{1/k} / (a + bx)^{1/m}$;

17) $y(x) = (a + x^2)^k / (b + x^4)^m$;

18) $y(x) = (a + x^2)^{1/k} / (b + x^4)^m$;

19) $y(x) = (a + x^2)^k / (b + x^4)^{1/m}$;

20) $y(x) = (a + x^2)^{1/k} / (b + x^4)^{1/m}$;

21) $y(x) = x^k / (a + bx^m)$;

22) $y(x) = x^{1/k} / (a + bx^m)$;

23) $y(x) = x^k / (a + bx^{1/m})$;

24) $y(x) = x^{1/k} / (a + bx^{1/m})$;

25) $y(x) = \ln^k(1 + x^m)$;

26) $y(x) = \ln^{1/k}(1 + x^m)$;

- 27) $y(x) = \ln^k(1 + x^{1/m})$; 28) $y(x) = \ln^{1/k}(1 + x^{1/m})$;
- 29) $y(x) = x^k e^{-2x^2}$; 30) $y(x) = x^{1/k} e^{-2x^2}$;
- 31) $y(x) = x^k e^{-2x^{1/m}}$; 32) $y(x) = x^{1/k} e^{-2x^{1/m}}$;
- 33) $y(x) = e^{-8(x-0.5)^2}$; 34) $y(x) = x^k e^{-8(x-0.5)^2}$;
- 35) $y(x) = x^{1/k} e^{-8(x-0.5)^2}$; 36) $y(x) = (a + bx^m)^{1/k}$;
- 37) $y(x) = (a + bx^{1/m})^{1/k}$; 38) $y(x) = (a + bx^m)^{-k}$;
- 39) $y(x) = (a + bx^m)^{-1/k}$; 40) $y(x) = (a + bx^{1/m})^{-1/k}$;
- 41) $y(x) = \arcsin[\cos^k(0.5\pi x^m)]$; 42) $y(x) = \arccos[\sin^k(0.5\pi x^m)]$;
- 43) $y(x) = \arcsin[\cos^k(0.5\pi x^{1/m})]$; 44) $y(x) = \arccos[\sin^k(0.5\pi x^m)]$;
- 45) $y(x) = \operatorname{arctg}(a \pm bx^m)$; 46) $y(x) = \operatorname{arctg}(a \pm bx^{1/m})$;
- 47) $y(x) = \operatorname{sh}(a \pm bx)^{-m}$; 48) $y(x) = \operatorname{sh}(a \pm bx)^{-1/m}$;
- 47) $y(x) = \operatorname{ch}(a \pm bx)^{-m}$; 50) $y(x) = \operatorname{ch}(a \pm bx)^{-1/m}$.

ПРОГРАММИРОВАНИЕ

Блок-схема программы представлена на рис. 3.1. Основу программы составляют три последовательных цикла — блоки 3-4- 5, 6-7-8, 9-10-11. Для хранения вычисляемых в этих циклах значений функции, первых и вторых разностей, а также погрешностей следует описать соответствующие массивы.

Для проверки правильности программы рекомендуется предварительно выполнить вычисления для тестовой функции

$$y(x) = x^2,$$

для которой ε_{\max} и ε_m должны обращаться в ноль.

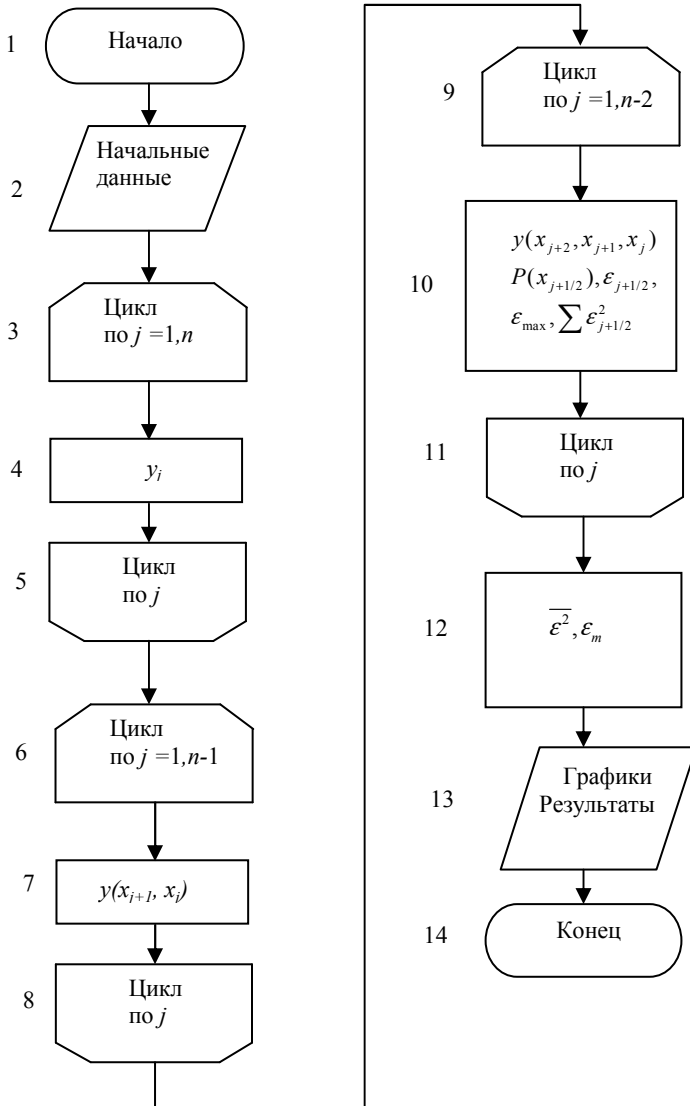


Рис.3.1. Блок-схема программы интерполяции

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулу и график функции $y(x)$ для конкретного варианта;
- текст программы;
- таблицу погрешностей $\varepsilon_{j+1/2}$ ($j = 0, 1, \dots, n - 2$);
- значения ε_{\max} и ε_m .

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как ставится задача интерполирования?
2. Какие интерполяционные многочлены вы знаете?
3. Как определяются разделенные разности различных порядков?
4. Как строился интерполяционный многочлен Ньютона?
5. Какова погрешность (остаточный член) интерполяционного многочлена?
6. Как можно практически оценить погрешность интерполирования?

Задание № 4

АППРОКСИМАЦИЯ

Цель работы: изучение аппроксимации функций на примере метода наименьших квадратов.

ПОСТАНОВКА ЗАДАЧИ

При замене функции интерполяционным многочленом необходимым условием является прохождение интерполяционного многочлена через значения функции в узлах интерполяции. В случае использования экспериментальных зависимостей значения функции в узлах получены с определенной погрешностью (часто достаточно большой), поэтому нецелесообразно прибегать к интерполяции, заставляя интерполяционный полином повторять эти ошибки. В этом случае лучше воспользоваться аппроксимацией, т. е. подбором функции, близко проходящей от заданных точек, заранее определив критерии «близости». В зависимости от выбранного способа приближения можно получить сильно отличающиеся друг от друга результаты: кривая может точно проходить через все заданные точки и в то же время сильно отличаться от сглаженной аппроксимирующей функции рис. 4.1.

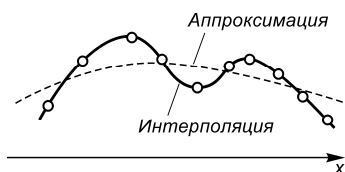


Рис. 4.1. Иллюстрация аппроксимации и интерполяции

Будем аппроксимировать функции многочленом степени m :

$$f(x) = c_0 + c_1x + c_2x^2 + \dots + c_mx^m,$$

коэффициенты которого c_i подберем так, чтобы минимизировать отклонение многочлена от данной функции.

Воспользуемся среднеквадратичным приближением функции $y(x)$ многочленом $\varphi(x)$ на множестве (x_i, y_i) , $(i = 0, 1, 2, \dots, n)$, при котором мерой отклонения является величина S , равная сумме квадратов разностей между значениями многочлена и функции в данных точках:

$$S = \sum_{i=0}^n \varepsilon_i^2 = \sum_{i=0}^n [\varphi(x_i, c_0, c_1, \dots, c_m) - y_i]^2.$$

Для построения аппроксимирующего многочлена нужно подобрать коэффициенты c_0, c_1, \dots, c_m так, чтобы величина S была наименьшей. В этом и состоит метод наименьших квадратов. Если отклонение ε_i подчиняется нормальному закону распределения, то полученные таким образом значения параметров наиболее вероятны. Как уже упоминалось выше, среднеквадратичное приближение сглаживает неточности функции, давая правильное представление о ней.

Поскольку c_i выступают в роли независимых переменных функции S , то минимум найдем, приравнявая нулю частные производные по этим переменным:

$$\frac{\partial S}{\partial c_0} = 0, \frac{\partial S}{\partial c_1} = 0, \frac{\partial S}{\partial c_2} = 0, \dots, \frac{\partial S}{\partial c_n} = 0,$$

т. е. приходим к системе уравнений для определения c_i . Если в качестве аппроксимирующей функции взять многочлен, то выражение для квадратов отклонений примет вид:

$$S = \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 + \dots + c_m x_i^m - y_i)^2.$$

Приравнявая нулю частные производные, приходим к системе:

$$\begin{aligned} \frac{\partial S}{\partial c_0} &= 2 \sum_{i=0}^n (c_0 + c_1 x_i + \dots + c_m x_i^m - y_i) = 0; \\ \frac{\partial S}{\partial c_1} &= 2 \sum_{i=0}^n (c_0 + c_1 x_i + \dots + c_m x_i^m - y_i) x_i = 0; \\ &\dots\dots \\ \frac{\partial S}{\partial c_m} &= 2 \sum_{i=0}^n (c_0 + c_1 x_i + \dots + c_m x_i^m - y_i) x_i^m = 0. \end{aligned}$$

Собирая коэффициенты при неизвестных c_0, c_1, \dots, c_m , получаем систему уравнений:

$$(n+1)c_0 + c_1 \sum_{i=0}^n x_i + c_2 \sum_{i=0}^n x_i^2 + \dots + c_m \sum_{i=0}^n x_i^m = \sum_{i=0}^n y_i;$$

$$c_0 \sum_{i=0}^n x_i + c_1 \sum_{i=0}^n x_i^2 + c_2 \sum_{i=0}^n x_i^3 + \dots + c_m \sum_{i=0}^n x_i^{m+1} = \sum_{i=0}^n x_i y_i;$$

.....

$$c_0 \sum_{i=0}^n x_i^m + c_1 \sum_{i=0}^n x_i^{m+1} + c_2 \sum_{i=0}^n x_i^{m+2} + \dots + c_m \sum_{i=0}^n x_i^{2m} = \sum_{i=0}^n x_i^m y_i.$$

Решая систему, находим неизвестные параметры c_0, c_1, \dots, c_m . В более компактном виде можно записать:

$$\begin{aligned} b_{00}c_0 + b_{01}c_1 + \dots + b_{0m}c_m &= a_0; \\ b_{10}c_0 + b_{11}c_1 + \dots + b_{1m}c_m &= a_1; \\ &\dots \\ b_{m0}c_0 + b_{m1}c_1 + \dots + b_{mm}c_m &= a_m, \end{aligned} \tag{4.1}$$

если ввести обозначения $b_{kl} = \sum_{i=0}^n x_i^{k+l}$, $a_k = \sum_{i=0}^n x_i^k y_i$; $k, l = 0, 1, \dots, m$.

В данной работе, для того чтобы облегчить решение системы, ограничимся значением $m=2$.

Обозначим чертой усреднение по множеству узлов x_i

$$\bar{u} = \frac{1}{n+1} \sum_{i=0}^n u_i$$

и введем также обозначения:

$$m_k = \overline{x^k} \quad (k=1, 2 \dots), \quad K_{ii} = \overline{x^i y^i}.$$

Тогда систему (4.1) можно записать в виде:

$$\begin{aligned} c_0 + m_1 c_1 + m_2 c_2 &= K_{01}; \\ m_1 c_0 + m_2 c_1 + m_3 c_2 &= K_{11}; \\ m_2 c_0 + m_3 c_1 + m_4 c_2 &= K_{21}. \end{aligned}$$

Систему уравнений можно решить любым из рассматриваемых далее прямых методов. Поскольку система имеет симметричную матрицу, то можно воспользоваться методом квадратных корней, расчетные формулы которого приведены ниже:

$$\begin{aligned}
s_{11} &= 1, s_{12} = m_1, s_{13} = m_2; \\
s_{22} &= \sqrt{m_2 - m_1^2}, s_{23} = (m_3 - m_1 m_2) / s_{22}; \\
s_{33} &= \sqrt{m_4 - (s_{13}^2 + s_{23}^2)} = \sqrt{m_4 - (m_2^2 + s_{23}^2)}; \\
z_1 &= K_{01}, z_2 = (K_{11} - m_1 K_{01}) / s_{22}; \\
z_3 &= [K_{21} - (s_{13} z_1 + s_{23} z_2)] / s_{33}; \\
c_2 &= \frac{z_3}{s_{33}}, c_1 = \frac{z_2 - z_{23} c_2}{s_{22}}, c_0 = z_1 - (s_{12} c_1 + s_{13} c_2).
\end{aligned}$$

(Для системы трех линейных уравнений вместо метода квадратных корней несложно воспользоваться известной схемой Крамера.)

Найдя коэффициенты c_0, c_1, c_2 , вычислим значения полинома $\varphi(x_j)$ ($j = 0, 1, \dots, n$) и погрешность аппроксимации

$$\varepsilon_{\max} = \max_j \varepsilon_j, \varepsilon_j = y(x_j) - \varphi(x_j), \varepsilon_m = \sqrt{\overline{\varepsilon^2}},$$

где средний квадрат погрешности $\overline{\varepsilon^2} = S / (n + 1)$.

ВАРИАНТЫ ЗАДАНИЙ

Параметры: $a = 0, b = 1, n = 10 \div 50$.

Узлы: $x_j = jh$ ($j = 0, 1, \dots, n$), $h = 1 / n$.

ВАРИАНТЫ ФУНКЦИЙ $y(x)$

- 1) $y(x) = \sin x^q$ ($q = 1 \div 3$);
- 2) $y(x) = \sin x^{1/q}$ ($q = 2, 3$);
- 3) $y(x) = e^x$;
- 4) $y(x) = \ln(1 + x^q)$ ($q = 1 \div 3$);
- 5) $y(x) = \cos x^{1/q}$ ($q = 2, 3$);
- 6) $y(x) = \cos x^q$ ($q = 1 \div 3$);
- 7) $y(x) = e^{x^{1/q}}$ ($q = 2, 3$);
- 8) $y(x) = \ln(1 + x^{1/q})$ ($q = 1 \div 3$);

- 9) $y(x) = x(1-x) \pm 0.01x^q (q = 3 \div 5)$;
- 10) $y(x) = x(1-x) \pm x^{1/q} (q = 2 \div 5)$;
- 11) $y(x) = \operatorname{tg} x^{1/q} (q = 1 \div 3)$;
- 12) $y(x) = \sqrt{1+x^q} (q = 1 \div 4)$;
- 13) $y(x) = (1+x^q)^{-1} (q = 1 \div 4)$;
- 14) $y(x) = (1+x^2)^{-1/q} (q = 1 \div 3)$;
- 15) $y(x) = (1+x^2) / (1+x^3)$;
- 16) $y(x) = x(1+x)^{-1/q} (q = 2 \div 4)$;
- 17) $y(x) = e^{-x^2}$;
- 18) $y(x) = e^{-x}$;
- 19) $y(x) = \arcsin x^{1/q} (q = 1 \div 3)$;
- 20) $y(x) = (1+x)^{1/q} (q = 2 \div 4)$;
- 21) $y(x) = (1+x)^{-1/q} (q = 1 \div 3)$;
- 22) $y(x) = (1+x^2)^{1/q} (q = 2 \div 4)$;
- 23) $y(x) = x / (1+x^q) (q = 1 \div 4)$;
- 24) $y(x) = x^q e^{-x^2} (q = 1, 2)$.

ПРОГРАММИРОВАНИЕ

Блок-схема программы представлена на рис. 4.2. Для хранения y_i , $\phi(x_i)$ следует отвести массивы, имеющие не менее $n+1$ элементов. Вычисления рекомендуется выполнить для нескольких значений n , обращая внимание на изменение погрешности с ростом n . На экран достаточно выдать результаты для одного значения n . Для проверки правильности программы рекомендуется в качестве тестовой задачи аппроксимировать функцию

$$y(x) = (1+x^2),$$

для которой погрешности ε_{\max} , ε_m , должны быть равны нулю.

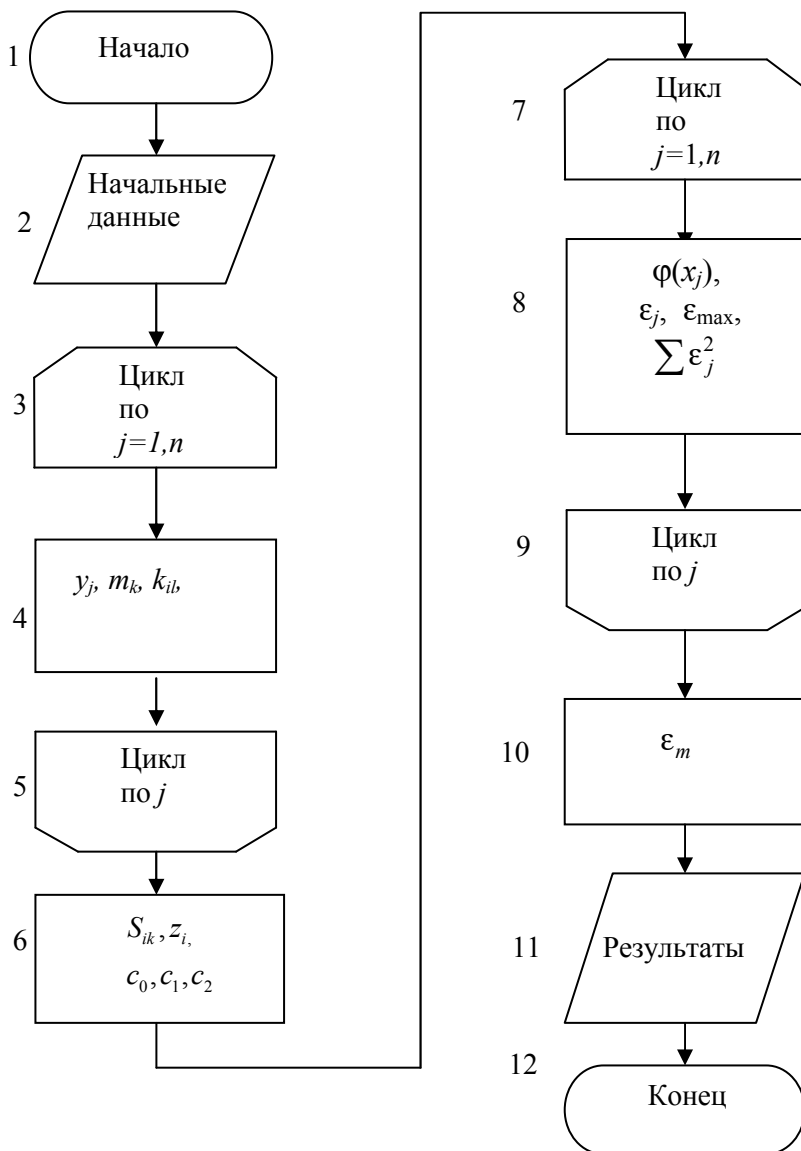


Рис. 4.2. Блок-схема программы аппроксимации

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулу функции $y(x)$ и параметры для конкретного варианта;
- текст программы;
- значения c_0, c_1, c_2 ; массивы и графики $y_i, \varphi(x_i)$; погрешности $\varepsilon_{\max}, \varepsilon_m, \overline{\varepsilon^2}$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как ставится задача аппроксимирования функций?
2. Что такое среднеквадратичное аппроксимирование?
3. Что такое равномерное аппроксимирование?
4. Как строится метод наименьших квадратов?
5. Чему равна погрешность (остаточный член) аппроксимирования степенными функциями?
6. Каковы условия полноты системы функций?

Задание № 5

ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ

Цель работы — изучение методов численного дифференцирования, вычисление первой и второй производных заданной функции с использованием интерполяционного многочлена Ньютона.

ПОСТАНОВКА ЗАДАЧИ

Одной из распространенных задач вычислительной математики, имеющей разнообразные приложения, является численное дифференцирование. Пусть задана сеточная функция

$$y_j = y(x_j) \quad (j = 0, 1, \dots, n), \quad (5.1)$$

определенная на множестве узлов x_j , ($j = 0, 1, \dots, n$). Для вычисления производной $y^{(k)}(x)$ порядка k ($k=1, 2, \dots$) в некоторой точке x выберем $m+1$ ($m \geq k$) узлов в окрестности этой точки и построим интерполяционный многочлен $P_m(x)$ степени m (например, многочлен Ньютона (см. задание № 3)), проходящий через все выбранные узлы:

$$y(x) = P_m(x) + R_m(x), \quad (5.2)$$

где $R_m(x)$ — остаточный член (погрешность) интерполяционного многочлена $P_m(x)$. Дифференцируя равенство (5.2), находим

$$y^{(k)}(x) = P_m^{(k)}(x) + R_m^{(k)}(x) \quad (k = 1, 2, \dots). \quad (5.3)$$

Примем теперь в качестве приближенного значения производной $\tilde{y}^{(k)}(x)$ производную многочлена:

$$\tilde{y}^{(k)}(x) = P_m^{(k)}(x). \quad (5.4)$$

Тогда остаточный член (погрешность) производной $Q_{m,k}(x)$ равняется производной остаточного члена (погрешности) интерполяционного многочлена:

$$Q_{m,k}(x) = y^{(k)}(x) - P_m^{(k)}(x) = R_m^{(k)}(x).$$

Производные (5.4) называются конечно-разностными. На практике наиболее часто используются равномерные сетки, т.е. сетки с равноотстоящими узлами. На таких сетках полученные указанным методом первая и вторая конечно-разностные производные в узлах x_j , с погрешностью $O(h^2)$ относительно шага сетки h даются формулами:

$$\left[\begin{array}{l} \tilde{y}'_j = \tilde{y}'(x_j) = \frac{y_{j+1} - y_{j-1}}{2h}, \quad y'_j = \tilde{y}'_j + O(h^2); \\ \tilde{y}''_j = \tilde{y}''(x_j) = \frac{y_{j+1} - 2y_j + y_{j-1}}{h^2}, \quad y''_j = \tilde{y}''_j + O(h^2); \\ (j = 1, 2, \dots, n-1). \end{array} \right.$$

В граничных узлах с номерами $j = 0$ и $j = n$ приходится вычислять так называемые односторонние производные, выбирая узлы интерполирования только с одной стороны от граничного узла. На равномерной сетке формулы второго порядка для первой и второй производных имеют вид:

$$\left[\begin{array}{l} \tilde{y}'_0 = \tilde{y}'(x_0) = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2); \\ \tilde{y}'_n = \tilde{y}'(x_n) = \frac{y_{n-2} - 4y_{n-1} + 3y_n}{2h} + O(h^2); \\ \tilde{y}''_0 = \tilde{y}''(x_0) = \frac{12y_0 - 30y_1 + 24y_2 - 6y_3}{6h^2} + O(h^2); \\ \tilde{y}''_n = \tilde{y}''(x_n) = \frac{-6y_{n-3} + 24y_{n-2} - 30y_{n-1} + 12y_n}{6h^2} + O(h^2). \end{array} \right.$$

Как видно из формул, в односторонних производных для достижения той же точности требуется больше узлов.

Для выполнения лабораторной работы предварительно составляется таблица значений (5.1) одной из заданных ниже функций $y(x)$ в равноотстоящих узлах

$$x_j = jh \quad (j = 0, 1, \dots, n), \quad h = 1/n,$$

на отрезке $0 \leq x \leq 1$. Значения n выбираются в диапазоне $n = 20 \div 100$.

Затем вычисляются точные y'_j, y''_j (аналитически) и приближенные \tilde{y}'_j и \tilde{y}''_j значения первой и второй производных, полученные по приведенным выше формулам. Во всех узлах находятся максимальные

$$\varepsilon_{k, \max} = \max_i \left| y_j^{(k)} - \tilde{y}_j^{(k)} \right| \quad (k = 1, 2)$$

и среднеквадратичные

$$\bar{\varepsilon}_k = \sqrt{\frac{1}{n+1} \sum_{j=0}^n (y_j^{(k)} - \tilde{y}_j^{(k)})^2} \quad (k = 1, 2)$$

значения погрешности численного дифференцирования, а также номера узлов $j_{k \max}$, в которых достигаются значения $\varepsilon_{k \max}$ ($k=1, 2$).

ВАРИАНТЫ ФУНКЦИЙ $y(x)$

- | | | |
|----------------------------------|----------------------|--|
| 1) $\sin(\pi x^2 / 2)$; | 2) $x^2 e^x$; | 3) $x \operatorname{sh} x$; |
| 4) $\cos(\pi x^2 / 2)$; | 5) $x^2 e^{-x}$; | 6) $x \operatorname{ch} x$; |
| 7) $\operatorname{sh} x^2$; | 8) $e^{x^2/2}$; | 9) $x^2 \operatorname{sh} x$; |
| 10) $\operatorname{ch} x^2$; | 11) $e^{-x^2/2}$; | 12) $x^2 \operatorname{ch} x^2$; |
| 13) $x \sin x$; | 14) $\sin^2 x$; | 15) $x \operatorname{sh} x^2$; |
| 16) $x \cos x$; | 17) $\cos^2 x$; | 18) $\operatorname{tg}(\pi x^2 / 4)$; |
| 19) $x^2 \sin x$; | 20) $x \sin^2 x$; | 21) $(x+1) \ln(x+1)$; |
| 22) $x^2 \cos x$; | 23) $x \cos^2 x$; | 24) $x \ln(x^2 + 1)$; |
| 25) $e^x \sin x$; | 26) $x e^{x^2/2}$; | 27) $\arcsin(x^2 / 2)$; |
| 28) $\operatorname{arctg} x^2$; | 29) $x e^{-x^2/2}$; | 30) $(x+1)^2 \ln(x+1)$. |

ПРОГРАММИРОВАНИЕ

Блок-схема программы численного дифференцирования представлена на рис. 5.1. Для хранения значений сеточной функции, точных и приближенных значений производных, а также их погрешностей следует отвести массивы длиной не менее $n + 1$. Поскольку в данной работе $n < 100$, достаточно описать массивы, каждый из которых имеет 101 элемент. Функцию $y(x)$, а также ее аналитические и конечно-разностные производные удобно описать в виде подпрограмм-функций. В цикле 3-4-5 вычисляются узлы и значения функции в них, а в цикле 6-7-8 аналитические и численные значения производных и погрешности.

Правильность программы можно проверить, задавая, например, тестовую функцию

$$y = x^2/2,$$

для которой должны получаться точные значения

$$\tilde{y}'_j = j/i, \quad \tilde{y}''_j = 1 \quad (j = 0, 1, \dots, n).$$

Рабочие варианты рекомендуется рассчитать для нескольких значений n , наблюдая за уменьшением погрешности с ростом n . На экран достаточно вывести результаты вычислений для какого-либо одного значения n .

СОДЕРЖАНИЕ ОТЧЁТА

Отчет должен содержать:

- функцию $y(x)$ и расчетные формулы для конкретного варианта;
- текст программы;
- результаты вычислений и графики функций:

$$y'_j, \tilde{y}'_j, y''_j, \tilde{y}''_j, (j = 0, 1, \dots, n),$$

причем точные и приближенные значения соответствующих производных должны быть изображены разными цветами в одном окне.

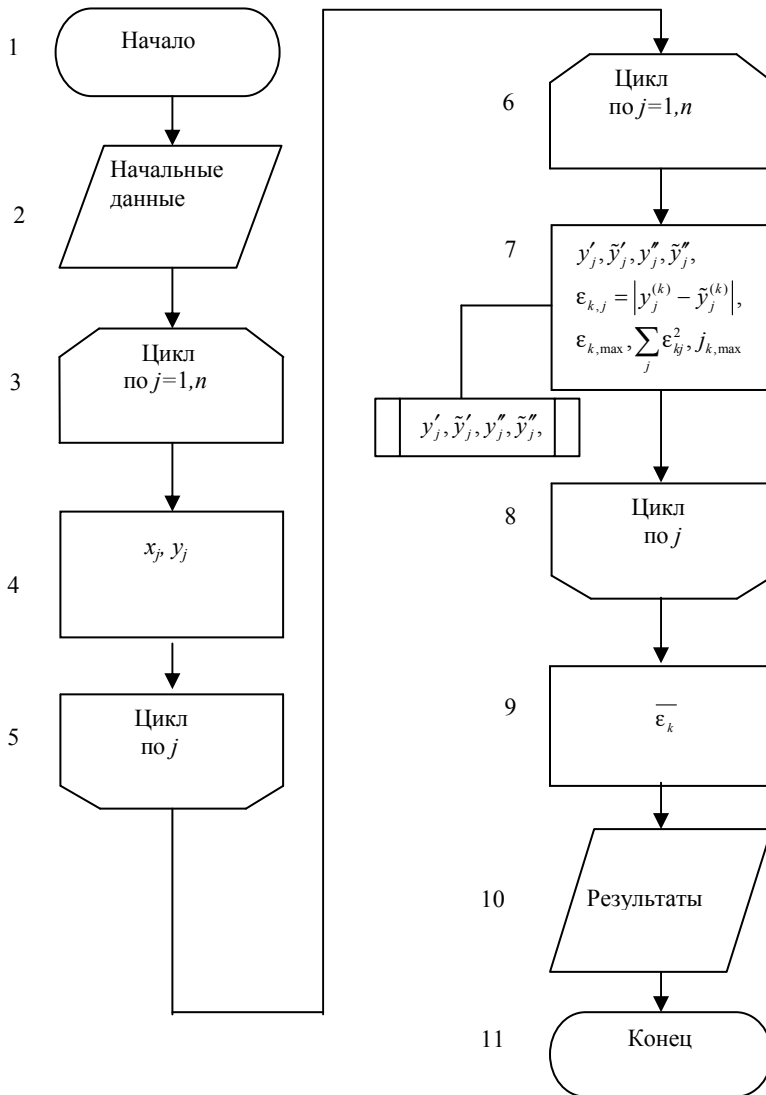


Рис. 5.1. Блок-схема программы численного дифференцирования.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как ставится задача численного дифференцирования?
2. Как строятся формулы численного дифференцирования, какова их погрешность?
3. Оцените погрешность используемых вами формул.
4. Как понижается порядок погрешности численного дифференцирования с ростом порядка производной при том же числе узлов?
5. Как можно построить формулы численного дифференцирования повышенной точности?
6. В чем проявляется некорректность постановки задачи численного дифференцирования?

Задание № 6

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Цель работы — изучение методов численного интегрирования, вычисление определенного интеграла от заданной функции методами прямоугольников и Гаусса.

ПОСТАНОВКА ЗАДАЧИ

Пусть на отрезке $[a, b]$ в точках $x_0 = a, x_1, \dots, x_n = b$ задана функция $y_i = f(x_i)$. Нам необходимо вычислить определенный интеграл вида

$$\int_a^b f(x) dx. \quad (6.1)$$

Используя определение интеграла как предела интегральной суммы, имеем:

$$\int_a^b f(x) dx = \lim_{\max \Delta x_i \rightarrow 0} \sum_{i=1}^{n-1} f(\xi_i) \Delta x_i, \quad \Delta x_i = x_{i+1} - x_i,$$

где $x_i \leq \xi_i \leq x_{i+1}$ — некая средняя точка интервала x_i, x_{i+1} . Задача интегрирования графически сводится к нахождению площади под графиком функции $f(x)$ на заданном отрезке рис. 6.1.

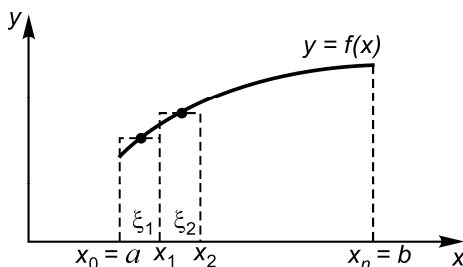


Рис. 6.1. Иллюстрация численного интегрирования

Ось x делится на n отрезков длиной Δx_i и на каждом отрезке по определенному критерию выбирается точка ξ_i и вычисляется в этой

точке значение функции $f(\xi_i)$. Площадь определяется суммой площадей полученных прямоугольников. Когда длины отрезков $\Delta x_i \rightarrow 0$, сумма площадей прямоугольников стремится к значению интеграла.

Для численного интегрирования функцию $f(x)$ заменяют такой аппроксимирующей функцией $\varphi(x)$, интеграл от которой легко бы вычислялся. Наиболее часто в качестве аппроксимирующих выступают обобщенные интерполяционные многочлены. Поскольку такая аппроксимация линейна относительно параметров, то функцию при этом заменяют неким линейным выражением, коэффициентами которого служат значения функции в узлах:

$$f(x) = \sum_{i=0}^n f(x_i)\Phi_i(x) + r(x),$$

где $r(x)$ — остаточный член аппроксимации.

Подставляя это выражение для функции в исходный интеграл (6.1), получим

$$\int_a^b f(x)dx \approx \sum_{i=0}^n q_i f(x_i) + R, \quad (6.2)$$

где $q_i = \int_a^b \Phi_i(x)dx$, $R = \int_a^b r(x)dx$.

Формула (6.2) называется квадратурной формулой с весами q_i и узлами x_i . Как видно из формулы, веса q_i зависят лишь от расположения узлов, но не от вида функции $f(x)$. Говорят, что квадратурная формула точна для многочленов степени m , если при замене функции $f(x)$ произвольным алгебраическим многочленом степени m остаточный член становится равным нулю.

Наиболее известные квадратурные формулы получают, если выбирать узлы x_i равноотстоящими на отрезке интегрирования. Такие формулы называются формулами Ньютона – Котеса. К формулам этого типа относятся известные формулы прямоугольников, трапеций, парабол (Симпсона) и некоторые другие.

В методе прямоугольников рис. 6.2 функцию $f(x)$ аппроксимируем полиномом нулевой степени

$$f(x) \approx f(x_0) = f_0.$$

Для вычисления интеграла на отрезке $[a, b]$ разобьем его на маленькие отрезки длиной h , а интеграл — на сумму интегралов на отдельных участках.

Тогда для одного участка

$$\int_{-h/2}^{h/2} f(x)dx \approx hf_0,$$

где f_0 — значение функции в середине отрезка. Таким образом, площадь криволинейной трапеции аппроксимируется прямоугольником, причем функция вычислена в средней точке отрезка.

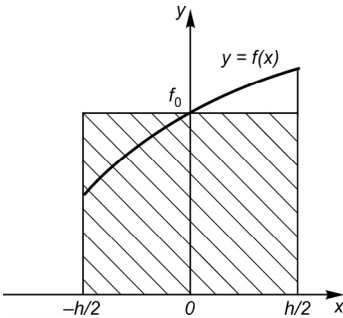


Рис. 6.2. Метод прямоугольников

Для i -го отрезка

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx hf_{i+1/2},$$

где $f_{i+1/2} = f(a + (i + 1/2)h)$.

Тогда, окончательно, значение интеграла на $[a, b]$

$$\int_a^b f(x)dx = h(f_{1/2} + f_{3/2} + \dots + f_{n-1/2}) + r(x).$$

Если узлы x_i фиксированы (расположены равномерно на $[a, b]$), то в квадратурной формуле (6.2) и веса q_i фиксированы. Тогда для построения интерполяционного полинома, аппроксимирующего функцию $f(x)$ на $[a, b]$, остается лишь $(n + 1)$ независимое условие, т. е. известные значения функции в узлах интерполяции $f(x_i)$. Таким образом, используя эти условия, можно построить многочлен не выше n -й степени. Если же не фиксировать положение узлов, а следовательно, и q_i , то в нашем распоряжении оказываются $(2n + 2)$

условия, с помощью которых можно построить многочлен $(2n + 1)$ -й степени.

Так возникла задача нахождения среди всех квадратурных формул с $(n + 1)$ узлами формулы с таким расположением узлов x_i на $[a, b]$ и с такими весами q_i , при которых она точна для многочленов максимальной степени. Интуитивно ясно, что погрешность метода тем меньше, чем выше порядок многочлена, при численном интегрировании которого получается точный результат.

Выполним замену переменной интегрирования в исходном интеграле (6.1)

$$x = a + (b - a)t \quad (0 \leq t \leq 1)$$

и преобразуем его к виду $I = (b - a)J$, где

$$J = \int_0^1 f(t)dt, \quad f(t) = f(x(t)).$$

Таким образом, мы приводим интеграл на любом отрезке к фиксированному интервалу $[0, 1]$, где и будем искать оптимальное расположение узлов. Такая задача успешно решена, и в справочниках для данного интервала приведены расположение узлов t_i и весов A_i , где $i=1, \dots, m$.

Для вычисления интеграла воспользуемся квадратурной формулой следующего вида:

$$J = \int_0^1 f(t)dt = \sum_{i=1}^m A_i f(t_i) + R_m.$$

Остаточный член формулы Гаусса с m узлами имеет вид

$$R_m = M_m \max_{0 \leq t \leq 1} |f^{(2m)}(t)|, \quad M_m = \frac{(m!)^4}{(2m+1)|(2m)!|^3}.$$

В частности $M_3 = 4.960 \cdot 10^{-7}$, $M_5 = 3.945 \cdot 10^{-13}$, $M_7 = 6.492 \cdot 10^{-20}$, $M_9 = 3.478 \cdot 10^{-27}$, $M_{10} = 5.734 \cdot 10^{-31}$ и т.д.

Веса A_i и узлы t_i квадратурных формул Гаусса имеют значения:

$$m = 3$$

$$t_1 = 1 - t_3 = 0.112701665,$$

$$t_2 = 0.500000000,$$

$$A_1 = A_3 = 0.277777778,$$

$$A_2 = 0.444444444.$$

$m=5$

$$\begin{aligned}t_1 &= 1 - t_5 = 0.046910077, \\A_1 &= A_5 = 0.118463443, \\t_3 &= 0.500000000,\end{aligned}$$

$$\begin{aligned}t_2 &= 1 - t_4 = 0.230765345, \\A_2 &= A_4 = 0.239314335, \\A_3 &= 0.284444444.\end{aligned}$$

$m=7$

$$\begin{aligned}t_1 &= 1 - t_7 = 0.025446044, \\t_3 &= 1 - t_5 = 0.297077424, \\A_1 &= A_7 = 0.064742483, \\A_3 &= A_5 = 0.190915025,\end{aligned}$$

$$\begin{aligned}t_2 &= 1 - t_6 = 0.129234407, \\t_4 &= 0.500000000, \\A_2 &= A_6 = 0.139852696, \\A_4 &= 0.208979592.\end{aligned}$$

$m=9$

$$\begin{aligned}t_1 &= 1 - t_9 = 0.015919880, \\t_3 &= 1 - t_7 = 0.193314284, \\t_5 &= 0.500000000, \\A_1 &= A_9 = 0.040637194, \\A_3 &= A_7 = 0.130305348,\end{aligned}$$

$$\begin{aligned}t_2 &= 1 - t_8 = 0.081934446, \\t_4 &= 1 - t_6 = 0.337873288, \\A_5 &= 0.165119678, \\A_2 &= A_8 = 0.090324080, \\A_4 &= A_6 = 0.156173539.\end{aligned}$$

$m=11$

$$\begin{aligned}t_1 &= 1 - t_{11} = 0.010885671, \\t_3 &= 1 - t_9 = 0.134923997, \\t_5 &= 1 - t_7 = 0.365228422, \\A_1 &= A_{11} = 0.027834284, \\A_3 &= A_9 = 0.093145105, \\A_5 &= A_7 = 0.131402272,\end{aligned}$$

$$\begin{aligned}t_2 &= 1 - t_{10} = 0.056468700, \\t_4 &= 1 - t_8 = 0.240451935, \\t_6 &= 0.500000000, \\A_2 &= A_{10} = 0.062790185, \\A_4 &= A_8 = 0.116596882, \\A_6 &= 0.136462543.\end{aligned}$$

ВАРИАНТЫ ЗАДАНИЙ

Для вычисления интеграла используем метод прямоугольников с числом узлов от m до 100 и квадратурную формулу Гаусса $sm=5 \div 11$ узлами. В исходные данные включаются: функция $f(x)$; пределы интегрирования a , b ; число узлов m , веса A_i и узлы t_i квадратурной формулы Гаусса.

Вычислить интеграл вида $\int_a^b E_0(\xi) d\xi$ по данным, приведенным в

табл. 6.1.

Таблица 6.1

№	$E_0(\xi)$	a	b
1	$\sqrt{e^\xi - 0.5}$	0.2	2.5
2	$e^\xi \sin(\xi/2)$	0	π
3	$(\xi^2 - 1)e^{-2\xi}$	0	2
4	$1/ 3 + 2\cos(\xi/2) $	0	π
5	$\xi^2 \sqrt{1 + \xi^2}$	1	4
6	$\xi / \ln^2 \xi$	1.5	3
7	$1/ 4 - 3\sin(\xi/2) $	0	π
8	$\xi^3 e^{3\xi}$	0	2
9	$\xi^3 e^{-3\xi}$	0	2
10	$\arcsin \sqrt{\xi(\xi + \sqrt{\xi(1 + \xi)})}$	0.1	1
11	$\cos^2(0.5\xi + 0.125\xi^2)$	0	π
12	$\xi \operatorname{arctg} \sqrt{\xi}$	0	4
13	$\operatorname{sh} \sqrt{\xi} + 2\xi$	0.1	3
14	$\operatorname{ch} \sqrt{\xi} + \xi/2$	0.1	4
15	$\xi \cos^2 \xi + e^\xi$	0	π

ПРОГРАММИРОВАНИЕ

Для хранения весов A_i , узлов t_i квадратурной формулы Гаусса и значений функций в центрах выбранных отрезков $f_{i+1/2}$ в методе прямоугольников, следует описать массивы соответствующей длины. Вычисления в методе Гаусса можно упростить, учитывая симметрию весов и узлов относительно середины отрезка $t=0.5$. Значения A_i, t_i ($i=1,2,\dots,11$) должны быть предварительно введены в массивы $A(i)$ и $T(i)$ с помощью операторов присваивания или оператора ввода начальных данных.

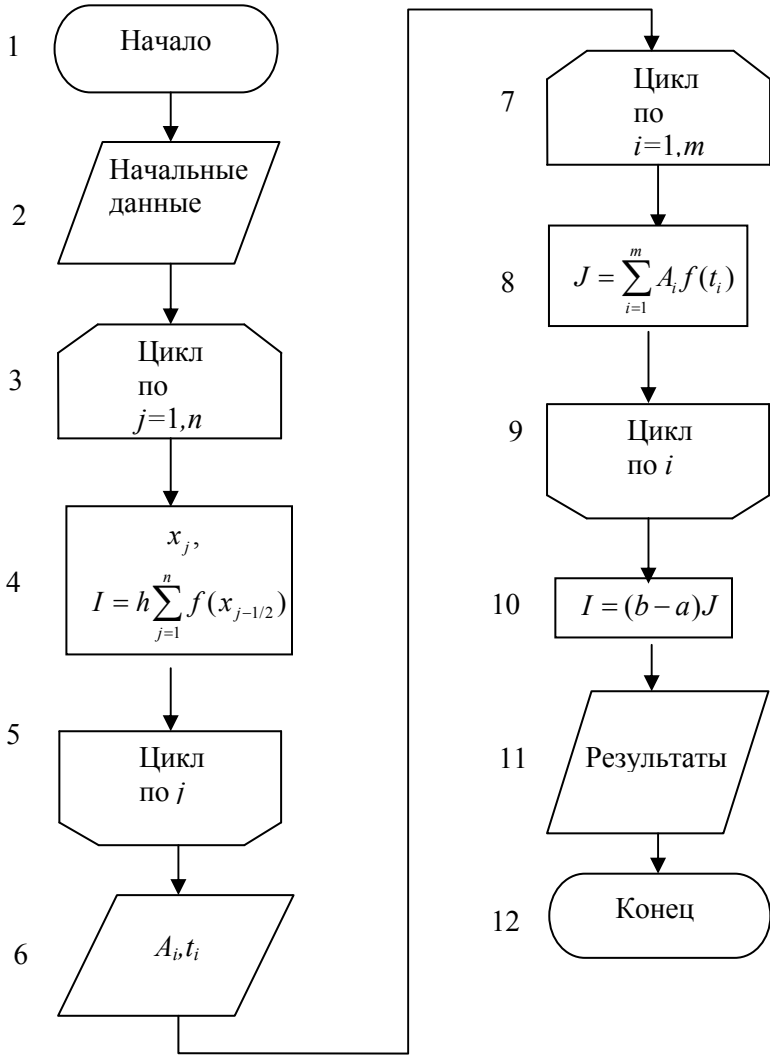


Рис. 6.3. Блок-схема программы интегрирования

Блок-схема программы вычисления интеграла методом прямоугольников и методом Гаусса приведена на рис. 6.3. В цикле 3-4-5 реализован метод прямоугольников, а в 7-8-9 – метод Гаусса. Правильность интегрирования можно проверить, вычисляя в качестве теста интеграл

$$I = (n + 1) \int_0^1 x^n dx \quad (n \leq 2m - 1),$$

для которого должно получиться точное значение 1, или

$$I = 4 \int_0^1 (1 - x^2)^{1/2} dx,$$

значение которого равно π .

СОДЕРЖАНИЕ ОТЧЁТА

Отчет должен содержать:

- подынтегральную функцию и пределы интегрирования конкретного варианта;
- число узлов в методах прямоугольников и Гаусса;
- текст программы;
- график подынтегральной функции;
- значение интеграла, полученное двумя методами.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как ставится задача численного интегрирования?
2. Как строятся интерполяционные квадратурные формулы, какова их погрешность (остаточный член)?
3. Как строятся квадратурные формулы Гаусса, какова их погрешность (остаточный член)?
4. Как строятся составные (большие) квадратурные формулы (прямоугольников, трапеций, парабол), какова их погрешность (остаточный член)?
5. Сравнить по точности метод прямоугольников и метод Гаусса при одинаковом числе узлов.

Задание № 7

ВЫЧИСЛЕНИЕ КРАТНЫХ ИНТЕГРАЛОВ МЕТОДОМ МОНТЕ-КАРЛО

Цель работы — знакомство с численными методами Монте-Карло, вычисление методом Монте-Карло кратного интеграла от заданной функции в выпуклой области.

ПОСТАНОВКА ЗАДАЧИ

Рассмотрим задачу вычисления n -мерного интеграла

$$I = \int_{(V)} f(x) dx, \quad X = (x_1, x_2, \dots, x_n), \quad dx = dx_1, dx_2, \dots, dx_n$$

в области V с границей Γ , вложенной в n -мерный параллелепипед

$$W = [a_i \leq x_i \leq b_i; i = 1, 2, \dots, n],$$

имеющий объем

$$|W| = (b_1 - a_1)(b_2 - a_2) \dots (b_n - a_n). \quad (7.1)$$

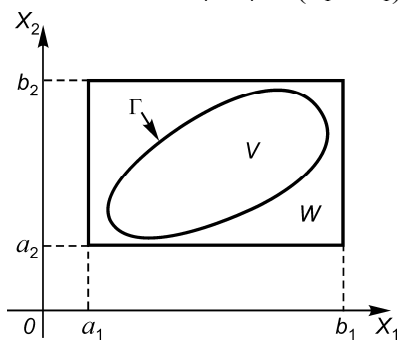


Рис. 7.1. Область интегрирования в случае двух переменных

На рис. 7.1 изображена область интегрирования V с границей Γ , вложенная в прямоугольник W ($n = 2$).

По теореме о среднем в интегральном исчислении среднее значение \bar{f} функции $f(x)$ по области V дается равенством

$$\bar{f} = \frac{1}{|V|} \int_{(V)} f(x) dx,$$

где $|V|$ — объем области V . Для вычисления \bar{f} с помощью генератора случайных чисел R , равновероятных на $(0, 1)$, будем генерировать случайные точки $X^{(k)}$ с координатами

$$X_i^{(k)} = a_i + (b_i - a_i)R_i^{(k)}, \quad (i = 1, 2, \dots, n; k = 1, 2, \dots, N),$$

равномерно распределенные в параллелепипеде W . Значение \bar{f} находится усреднением $f(x)$ по точкам $X^{(k)}$, принадлежащим области V :

$$\bar{f} = \frac{1}{M} \sum_{X^{(k)} \in V} f(X^{(k)}),$$

где M — число точек $X^{(k)}$, принадлежащих V .

Объем $|V|$ области V можно приближенно оценить по формуле

$$|\tilde{V}| \approx \frac{M}{N} |W|,$$

где N — общее число случайных точек $X^{(k)}$ в W , а $|W|$ определяется формулой (7.1). Исходя из формулы (7.2) находим оценку интеграла I :

$$\tilde{I} = |\tilde{V}| \bar{f} \approx \frac{1}{N} |W| \sum_{X^{(k)} \in V} f(X^{(k)}). \quad (7.3)$$

Принадлежность точки $X^{(k)}$ области V можно устанавливать по заданной границе Γ . Пусть, например, граница Γ имеет уравнение $\Gamma(x) = 0$, и точки x , удовлетворяющие условию $\Gamma(x) \leq 0$, принадлежат области V . Тогда в сумму (7.3) следует включить все точки $X^{(k)}$, удовлетворяющие условию

$$\Gamma(X^{(k)}) \leq 0.$$

В теории вероятностей показано, что погрешность интегрирования методом Монте-Карло имеет порядок $O(N^{-1/2})$, т.е. очень медленно убывает с ростом N . Тем не менее, при той же погрешности интегрирование методом Монте-Карло выполняется на компьютере быстрее, чем другими методами уже при $n > 3$, а при $n > 6$ — это единственно приемлемый метод интегрирования.

ВАРИАНТЫ ЗАДАНИЙ

1. Методом Монте-Карло найти объемный заряд

$$Q = \rho_0 \int_{(V)} \rho(x_1, x_2, x_3) dx_1 dx_2 dx_3 = \rho_0 \int_{(V)} \rho(x) dx$$

в трехмерной области V , ограниченной эллипсоидом

$$\sum_{i=1}^3 \left(\frac{x_i}{c_i}\right)^2 = 1 \quad (c_i = 0.1 \div 2)$$

с нормированной плотностью заряда $\rho(x_1, x_2, x_3)$.

ВАРИАНТЫ ФУНКЦИЙ

$$1) \rho(x) = \prod_{i=1}^3 |x_i - \alpha_i|^{\beta_i} \quad (\alpha_i = -5 \div 5, \beta_i = 0.5 \div 4);$$

$$2) \rho(x) = \prod_{i=1}^3 e^{-p_i |x_i - \alpha_i|^{q_i}} \quad (\alpha_i = -1 \div 1, p_i = 0.1 \div 0.5, q_i = 0.1 \div 2);$$

$$3) \rho(x) = \sum_{i=1}^3 |x_i - \alpha_i|^{\beta_i} \quad (\alpha_i = -5 \div 5, \beta_i = 0.5 \div 4);$$

$$4) \rho(x) = \prod_{i=1}^3 |x_i - \alpha_i|^{\beta_i} e^{-p_i |x_i - \alpha_i|^{q_i}} \quad (\alpha_i = -5 \div 5, \beta_i = 0.5 \div 4, p_i = 0.1 \div 0.5, q_i = 0.1 \div 2);$$

$$5) \rho(x) = \sum_{i=1}^3 e^{-p_i |x_i - \alpha_i|^{q_i}} \quad (\alpha_i = -5 \div 5, p_i = 0.1 \div 2, q_i = 0.1 \div 2);$$

$$6) \rho(x) = \sum_{i=1}^3 |x_i - \alpha_i|^{\beta_i} e^{-p_i |x_i - \alpha_i|^{q_i}} \quad (\alpha_i = -5 \div 5, \beta_i = 0.5 \div 4, p_i = 0.1 \div 0.5, q_i = 0.1 \div 2).$$

2. Температура T в точках $x=(x_1, x_2, x_3)$ шара

$$V_s = \left\{ \sum_{i=1}^3 x_i^2 \leq R^2 \right\}$$

изменяется со временем t по закону

$$T = T(x_1, x_2, x_3, t) = T(x, t).$$

Найти среднюю температуру шара за время $[t_1, t_2]$ по формуле

$$\bar{T} = \frac{1}{|V_s|(t_2 - t_1)} \int_{t_1}^{t_2} \int_{(V_s)} T(x_1, x_2, x_3, t) dx_1 dx_2 dx_3 dt,$$

где $|V_s| = \frac{4}{3} \pi R^3$ - объём шара.

В качестве $T(x, t)$ принять

$$T(x_1, x_2, x_3, t) = \rho(x_1 + g_1 t, x_2 + g_2 t, x_3 + g_3 t),$$

где $\rho(x)$ — одна из функции предыдущего задания, $g_i = 0.1 \div 0.9$ ($i = 1, 2, 3$).

3. Вычислить объем $|V|$ тела, ограниченного шестимерным эллипсоидом

$$\Gamma = \left\{ \sum_{i=1}^6 \left(\frac{x_i}{c_i} \right)^2 - 1 = 0 \right\} \quad (c_i = 0.1 \div 2).$$

Вычисление выполнить методом Монте-Карло по формуле

$$|V| = \int_{(V)} dx = \int_{(V)} dx_1 dx_2 dx_3 dx_4 dx_5 dx_6.$$

ПРОГРАММИРОВАНИЕ

Вычисление интеграла проведем для нескольких значений N , заданных отдельным массивом $N(L)$ в головной программе. Соответственно, в программе следует организовать выдачи результатов по достижении числа случайных чисел очередного значения N из массива. Это даст возможность наблюдать за изменением результатов и сходимостью интегрирования с ростом N .

Блок-схема программы представлена на рис. 7.2. Основу программы составляет цикл (блоки 3-10) по l от 1 до L , где L — заданное число вариантов с различным количеством случайных чисел N_l , для которых осуществляется выдача результатов. В блоке 4 происходит обращение к датчику случайных чисел для вычисления ξ . На рисунке j — текущее число случайных точек, M — число случайных точек в области V , \tilde{I} и $|\tilde{V}|$ — оценки интеграла I и объема области V .

В качестве теста необходимо вычислить объём эллипсоида в трёхмерной области в соответствии с пунктом 3 задания. Известно аналитическое решение

$$|V| = \frac{4}{3} \pi c_1 c_2 c_3.$$

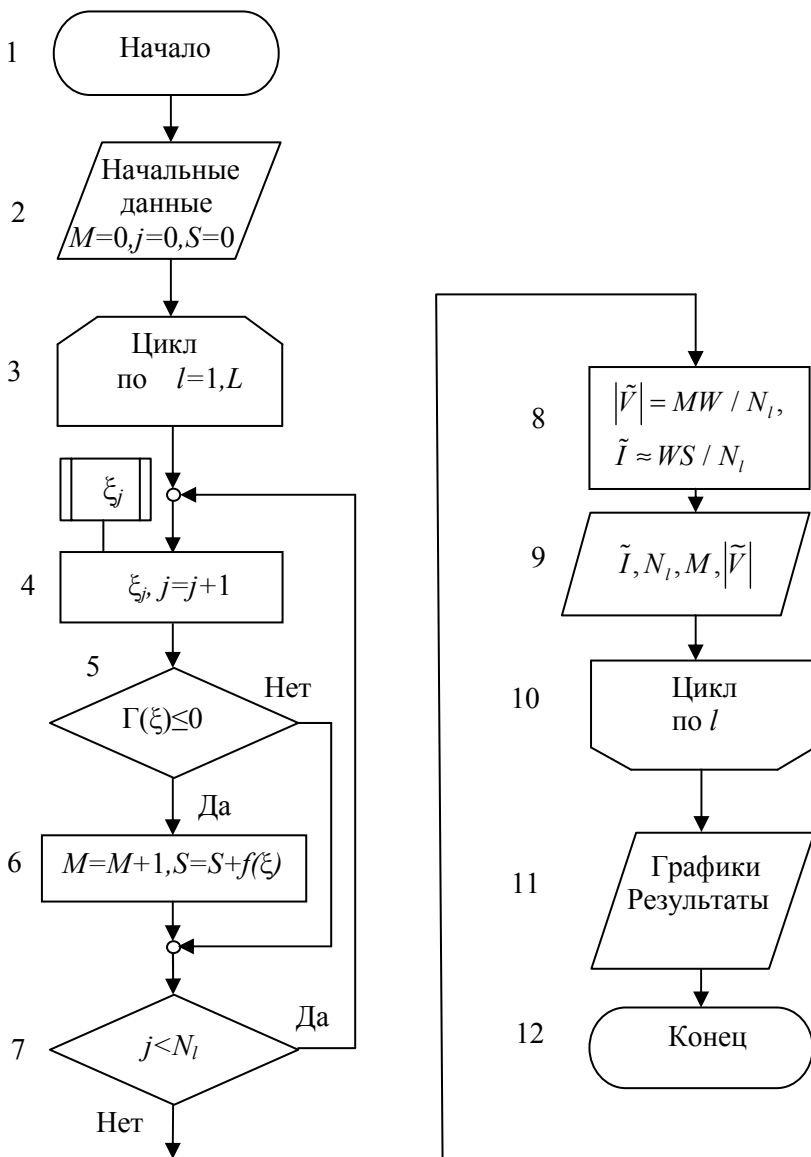


Рис. 7.2. Блок-схема программы вычисления интеграла методом Монте-Карло

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы подынтегральной функции и границы области интегрирования для конкретного варианта; текст программы;
- значения интеграла \tilde{I} для нескольких значений N , выбранных в диапазоне $N \sim 10^2 - 10^4$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие численные методы называются методами Монте-Карло?
2. Как вычисляются кратные интегралы методом Монте-Карло?
3. Как получена формула (7.3)?
4. Каков порядок погрешности интегрирования методом Монте-Карло?

$$a_{1j}^{(1)} = \frac{a_{1j}}{a_{11}} \quad (j = 2, 3, \dots, n), \quad b_1^{(1)} = \frac{b_1}{a_{11}}.$$

Умножая затем это уравнение на $-a_{i1}$ и складывая с i -м уравнением ($i = 2, 3, \dots, n$), исключим x_1 из всех уравнений, получая систему $(n - 1)$ уравнений. Далее аналогично исключим x_2, x_3 и т. д. Этот процесс исключения называется прямым ходом метода Гаусса. Наконец, остается одно уравнение, из которого находим x_n , а затем последовательно остальные неизвестные $x_{n-1}, x_{n-2}, \dots, x_1$; этот этап решения называют обратным ходом.

Одним из наиболее эффективных прямых методов является используемый в данной работе метод факторизации, называемый также схемой Халецкого (Холесского) или компактной схемой Гаусса. В этой схеме матрица A представляется в факторизованном виде

$$A = LU, \tag{8.2}$$

где L (lower) — нижняя треугольная матрица, U (upper) — верхняя треугольная матрица с единицами вдоль главной диагонали:

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 1 & u_{12} & \dots & u_{1n} \\ 0 & 1 & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

Элементы матриц L и U находятся из матричного уравнения (8.1). Матрица U может быть получена из (8.1) как результат прямого хода метода Гаусса, а матрица L хранит коэффициенты, на которые умножались уравнения в процессе исключения неизвестных в прямом ходе.

После этого исходная система (8.2) сводится к последовательному решению двух систем

$$Ly = b, \tag{8.3}$$

$$Ux = y \tag{8.4}$$

с треугольными матрицами.

Введем расширенные матрицы, дополняя A и U еще одним столбцом:

$$A + b, \quad a_{i, n+1} = b_i, \quad U + y, \quad u_{i, n+1} = y_i. \tag{8.5}$$

то в левой части получим новое значение вектора $x^{(1)} = (x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$.

Таким образом мы получаем итерационный метод Якоби, имеющий в компонентах вид

$$x_i^{(k)} = -\sum_{j \neq i} a_{ij} x_j^{(k-1)} / a_{ii} + b_i / a_{ii} \quad (i = 1, 2, \dots, n).$$

Метод Якоби сходится, если матрица A обладает диагональным преобладанием, т. е.

$$a_{ii} > \sum_{j \neq i} |a_{ij}| \quad (i = 1, 2, \dots, n)$$

(все диагональные элементы $a_{ii} > 0$, так как матрица A положительно определенная).

Представим матрицу A в виде

$$A = L + D + U, \tag{8.10}$$

где L — нижняя, а U — верхняя треугольные матрицы с нулями вдоль главных диагоналей, D — диагональная матрица:

$$L = \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix}, \quad D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix},$$

$$U = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

Итерационный метод Зейделя имеет вид

$$x^{(k)} = -D^{-1}(Lx^{(k)} + Ux^{(k-1)} - b) \tag{8.11}$$

или в компонентах

$$x_i^{(k)} = -a_{ii}^{-1} \left(\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} + \sum_{j=i+1}^n a_{ij} x_j^{(k-1)} - b_i \right). \tag{8.12}$$

Как мы видим, в этом методе в отличие от метода Якоби в первой части используются значения $x_j^{(k)}$, уже вычисленные в ходе данной итерации.

Здесь индекс k – номер итерации. Начальное приближение $x^{(0)}$ в итерационных методах может быть выбрано произвольно, например, можно принять $x^{(0)} = 0$ или $x^{(0)} = b$. Итерации проводятся до выполнения одного из условий:

$$\max_i |x_i^{(k)} - x_i^{(k-1)}| < \delta; \quad \|\xi^{(k)}\| < \delta; \quad k \geq k_{\max},$$

где δ — заданная малая величина, k_{\max} — предельное число итераций. Число арифметических операций в итерационных методах имеет порядок $O(n^2k)$, где k — число выполненных итераций.

ВАРИАНТЫ ЗАДАНИЙ

$$n = 5 \div 20; \quad p, q = 1 \div 4; \quad r, t = 0.1 \div 1.2.$$

$$1) a_{ii} = 5i^{p/2}, \quad a_{ij} = \pm 0.01(i^{p/2} \pm j^{q/2}) \quad (i \neq j), \quad b_i = 4.5i^{p/2};$$

$$2) a_{ii} = 15i^p, \quad a_{ij} = \pm i^{-p/2} j^{-q/2} \quad (i \neq j), \quad b_i = 3.5i^p;$$

$$3) a_{ii} = 10i^{p/2}, \quad a_{ij} = \pm 10^{-3}(i/j^r)^{1/q} \quad (i \neq j), \quad b_i = 9i^{p/2};$$

$$4) a_{ii} = 8i^{p/3}, \quad a_{ij} = \pm 10^{-2}(i^p \pm j^q)^{-1} \quad (i \neq j), \quad b_i = 7i^{p/3};$$

$$5) a_{ii} = 9i^{p/3}, \quad a_{ij} = \pm 10^{-2}(i^{p/2} \pm j^{q/2})^{-1} \quad (i \neq j), \quad b_i = 8i^{p/3};$$

$$6) a_{ii} = 8.5i^{p/3}, \quad a_{ij} = \pm 10^{-3} \exp\left(-\frac{r}{|i \pm j|^t}\right) \quad (i \neq j), \quad b_i = 7.5i^{t/2};$$

$$7) a_{ii} = 12i^r, \quad a_{ij} = \pm 0.1(ij)^{\frac{r}{2}} \quad (i \neq j), \quad b_i = 10.5i^{t/2};$$

$$8) a_{ii} = 14i^{r/3}, \quad a_{ij} = 10^{-2} \operatorname{sh} \frac{r}{|i \pm j|^t} \quad (i \neq j), \quad b_i = 9.5i^{t/2};$$

$$9) a_{ii} = 11i^{r/2}, \quad a_{ij} = \pm 10^{-2} \sin \frac{\pi r / 2}{|i \pm j|^t} \quad (i \neq j), \quad b_i = 8.5i^{t/3}.$$

ПРОГРАММИРОВАНИЕ

Компактная схема Гаусса. Для экономии памяти компьютера треугольные матрицы L , U и вектор u можно объединить в единую матрицу с элементами

$$\begin{pmatrix} l_{11} & u_{12} & u_{13} & \cdots & u_{1n} & u_{1,n+1} \\ l_{12} & l_{22} & u_{23} & \cdots & u_{2n} & u_{2,n+1} \\ l_{31} & l_{32} & l_{33} & \cdots & u_{3n} & u_{3,n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ l_{1n} & l_{2n} & l_{3n} & \cdots & l_{nn} & u_{n,n+1} \end{pmatrix}$$

(поэтому этот метод и называют компактной схемой Гаусса).

При таком объединении главная диагональ матрицы U в памяти компьютера исчезает, однако равенство единице всех ее элементов уже учтено в формуле обратного хода (8.9) тем, что коэффициент перед $u_{i,n+1} = y_i$ равен 1. Удобнее, однако, сохранить обозначения в формулах (8.2) — (8.8), но совместить матрицы L и U в памяти компьютера, пользуясь оператором эквивалентности.

При составлении программы некоторые осложнения возникают из-за того, что суммирование в формулах (8.6) — (8.8) допустимо лишь тогда, когда верхний предел суммы не меньше нижнего.

Кроме того, вычисление решения x в (8.8) можно организовать так, чтобы переменная цикла i не убывала, а возрастала. Составленная с учетом этих особенностей блок-схема программы приведена на рис. 8.1.

Отметим, что из внутреннего цикла допускается принудительный выход как во внешний цикл, так и к другим операторам программы, а из внешнего цикла принудительный выход допускается лишь к операторам вне вложенного цикла, т. е. и здесь соблюдается общее правило «вход в цикл возможен лишь через заголовок».

В блоках 3-9 вычисляются матрица A и правая часть b системы уравнений, блоки 10-29 составляют прямой ход, блоки 31-38 — обратный ход, в блоке 39 вычисляется норма невязки.

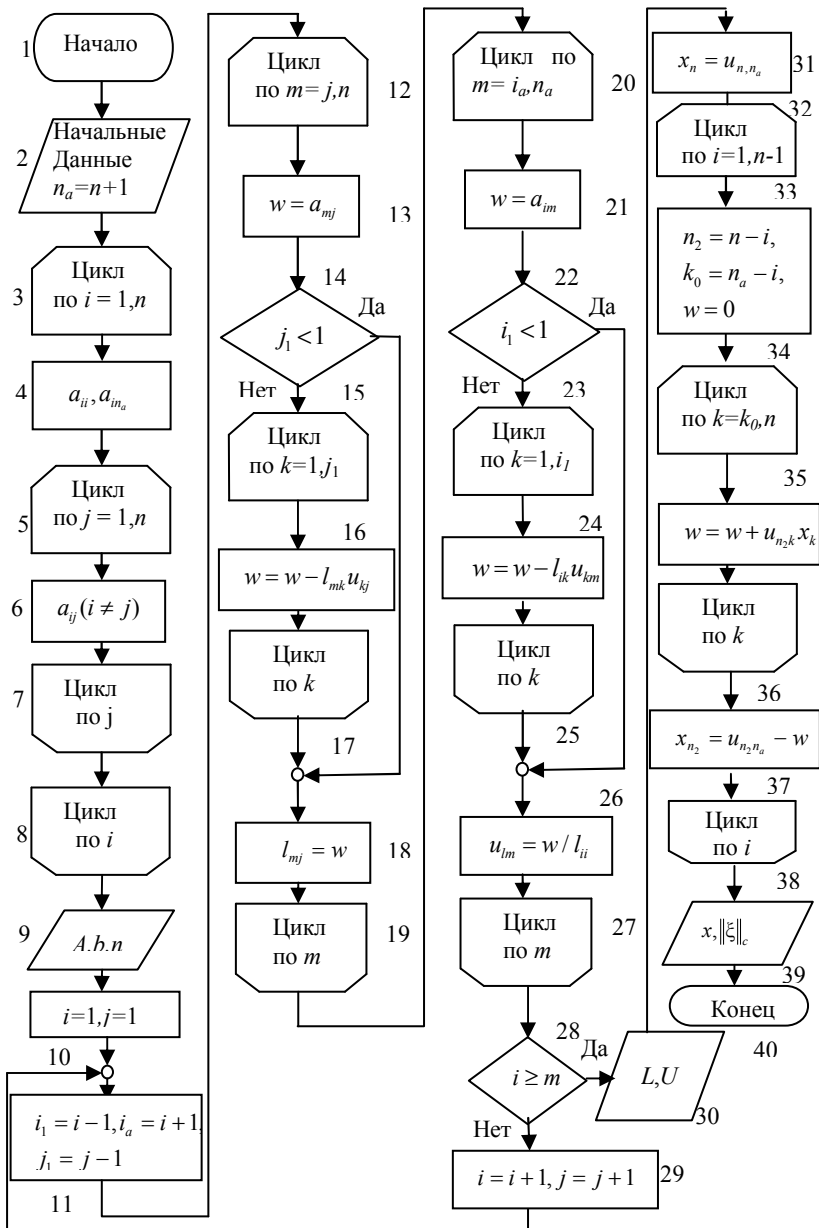


Рис. 8.1. Блок-схема метода факторизации

В программе организован вывод A , b , промежуточных и окончательных результатов. При выводе на экран достаточно ограничиться значением $n = 5$, при этом невязка должна иметь норму порядка $10^{-5} \div 10^{-6}$, большие значения нормы указывают на ошибки в программе.

При $n=10 \div 20$ порядок невязки может возрасти до 10^{-4} из-за накопления вычислительных погрешностей.

Итерационный метод Зейделя. Матрицу A и вектор b сначала преобразуем к так называемому приведенному виду:

$$C = \begin{bmatrix} 0 & a_{12}/a_{11} & -a_{13}/a_{11} & \cdots & -a_{1n}/a_{11} \\ -a_{21}/a_{22} & 0 & -a_{23}/a_{22} & \cdots & -a_{2n}/a_{22} \\ & \cdots & \cdots & \cdots & \cdots \\ -a_{n1}/a_{nn} & -a_{n2}/a_{nn} & -a_{n3}/a_{nn} & \cdots & 0 \end{bmatrix},$$

$$d = \begin{bmatrix} b_1/a_{11} \\ b_2/a_{22} \\ \cdots \\ b_n/a_{nn} \end{bmatrix}.$$

Чтобы не портить матрицу A и вектор b , для C и d отводятся отдельные массивы.

Для хранения элементов вектора x на текущей k -й и предыдущей $(k-1)$ итерациях отведем единый массив, имеющий не менее n элементов. Перед началом k -й итерации

$$x = (x_i^{(k-1)}; i = 1, 2, \dots, n).$$

После вычисления очередного значения $x_i^{(k)}$ по формуле (8.12) оно помещается на место $x_i^{(k-1)}$ ($i = 1, 2, \dots, n$). При такой организации вычислений часть элементов массива x будет содержать новые значения $x_i^{(k)}$, а часть — предыдущие $x_i^{(k-1)}$. т.е. после вычисления значения $x_i^{(k)}$ массив имеет вид

$$X = (x_1^{(k)}, x_2^{(k)}, \dots, x_i^{(k)}, x_{i+1}^{(k-1)}, \dots, x_n^{(k-1)}).$$

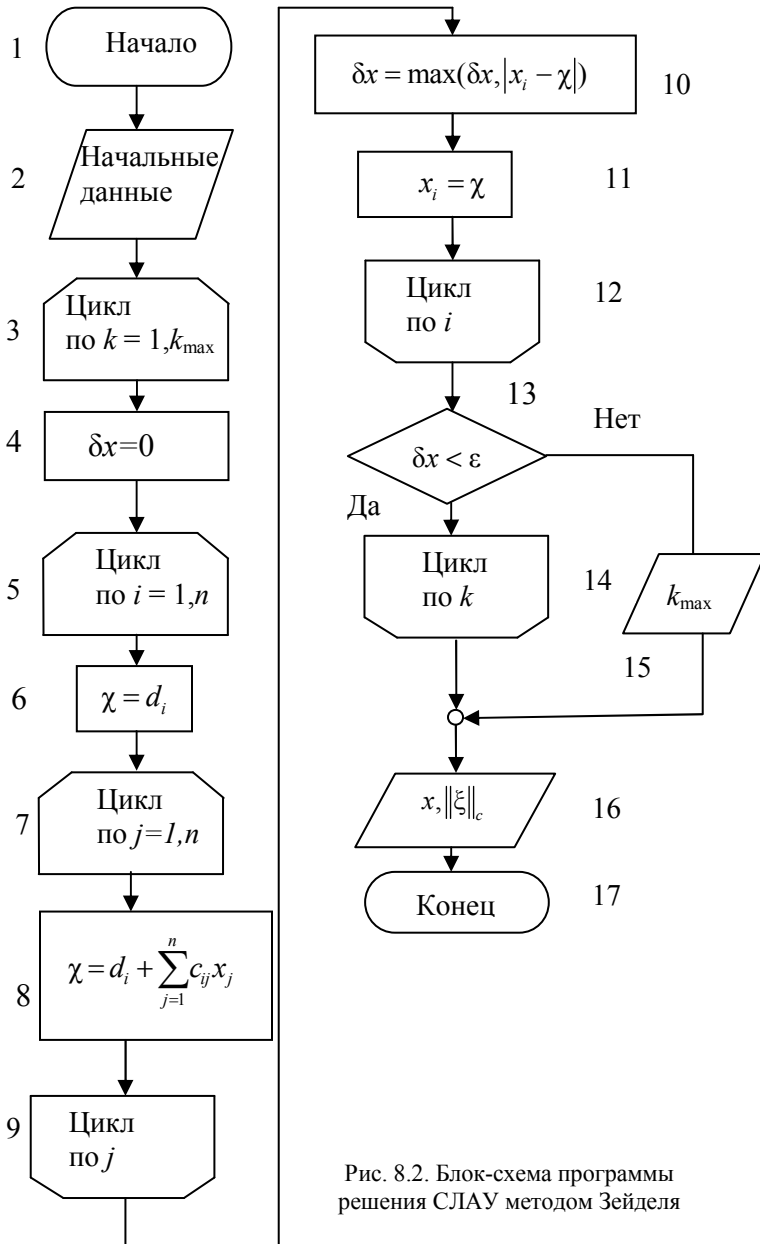


Рис. 8.2. Блок-схема программы решения СЛАУ методом Зейделя

В результате организация итераций упрощается и выполняется по формулам

$$\begin{aligned} \delta x &= 0; \\ \left[\begin{array}{l} \chi = d_i + \sum_{j=1}^n c_{ij} x_j; \\ \delta x = \max(\delta x, |x_i - \chi|); \\ x_i = \chi (i = 1, 2, \dots, n) \end{array} \right. \end{aligned}$$

(номера итераций опущены, χ — промежуточная рабочая величина). После завершения k -й итерации

$$\begin{aligned} X &= (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}), \\ \delta x &= \max_{1 \leq i \leq n} |x_i^{(k)} - x_i^{(k-1)}|. \end{aligned}$$

В качестве начального приближения рекомендуется принять

$$x_i^{(0)} = d_i \quad (i = 1, 2, \dots, n),$$

Ошибки в программе могут привести к заикливанию, т.е. к бесконечному, не сходящемуся итерационному процессу. Чтобы избежать этого, рекомендуется установить максимально допустимое число итераций k_{\max} , по достижении которого итерации прерываются (цикл 3-14). Составленная в соответствии с этими замечаниями блок-схема программы приведена на рис. 8.2.

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы для конкретного варианта;
- текст программы;
- результаты решения СЛАУ двумя методами.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие численные методы решения СЛАУ вы знаете, каковы их характерные особенности?
2. Как строится решение систем линейных алгебраических уравнений методом исключения Гаусса?

3. Как строится решение систем линейных алгебраических уравнений методом Зейделя?
4. Как можно проверить правильность решения системы линейных алгебраических уравнений?
5. Чем отличается невязка от погрешности решения?
6. Всегда ли можно получить решение системы линейных алгебраических уравнений?
7. Каковы достаточные условия сходимости метода Зейделя?
8. Что такое обусловленность системы?

Задание № 9

ЧАСТИЧНАЯ ПРОБЛЕМА СОБСТВЕННЫХ ЗНАЧЕНИЙ

Цель работы: изучение частичной проблемы собственных значений, вычисление максимального собственного значения самосопряженной матрицы.

ПОСТАНОВКА ЗАДАЧИ

Пусть A – квадратная матрица размера $n \times n$. Частичной проблемой собственных значений называется нахождение некоторой части собственных значений λ и, возможно, соответствующих собственных векторов x матрицы A из однородного уравнения

$$Ax = \lambda x.$$

Одной из задач проблемы собственных значений является рассматриваемая в данной работе задача нахождения максимального по модулю собственного значения матрицы A .

Далее будем полагать, что A – действительная симметричная матрица. Самосопряженная матрица A имеет полную систему (базис) ортогональных нормированных собственных векторов e_i ($i = 1, 2, \dots, n$), являющихся решением задачи

$$Ae_i = \lambda_i e_i,$$

$$\sqrt{e_i e_j} = \delta_{ij} = \begin{cases} 1 & (i = j), \\ 0 & (i \neq j), \end{cases}$$

где δ_{ij} – символ Кронекера. Предположим также, что максимальное собственное значение матрицы A $|\lambda_1(A)| = \max_i |\lambda_i(A)|$ – простое (не кратное).

Тогда $\lambda_1(A)$ можно найти с помощью следующего итерационного процесса, называемого степенным методом:

$$\left\{ \begin{array}{l} \|x^{(k-1)}\| = (x^{(k-1)}, x^{(k-1)})^{1/2}, \\ e_1^{(k-1)} = \frac{x^{(k-1)}}{\|x^{(k-1)}\|}, \\ x^{(k)} = A e_1^{(k-1)}, \\ \lambda_1^{(k)} = (x^{(k)}, e_1^{(k-1)}) \quad (k=1, 2, \dots). \end{array} \right.$$

Здесь k - номер итерации, а в качестве $x^{(0)}$ принимается произвольный, например случайный вектор. При $k \rightarrow \infty$ итерационный процесс сходится линейно, т.е.

$$\lambda_1^{(k)} \rightarrow \lambda_1(A),$$

со скоростью геометрической прогрессии, имеющей знаменатель $|\lambda_2(A) / \lambda_1(A)|$, где $\lambda_2(A)$ - следующее по модулю собственное значение матрицы A .

Если значения $|\lambda_2(A)|$ и $|\lambda_1(A)|$ близки, то сходимость будет медленной.

Смысл данного итерационного процесса можно пояснить следующим образом. Поскольку собственные векторы в нашем случае образуют полную систему, то любой вектор можно представить в виде

$$y = c_1 x_1 + c_2 x_2 + \dots + c_n x_n,$$

где x_i - собственные векторы, а c_i - константы.

Умножим обе части равенства на матрицу A

$$Ay = \sum_{i=1}^n c_i A x_i = \sum_{i=1}^n c_i \lambda_i x_i.$$

Производя такое умножение k раз, получим

$$A^k y = \sum_{i=1}^n c_i A^k x_i = \sum_{i=1}^n c_i \lambda_i^k x_i.$$

Таким образом, после k умножений вектор y имеет компоненту вдоль каждого собственного вектора, пропорциональную k -й степени соответствующего значения λ_i .

Теперь, если $|\lambda_1| > |\lambda_i|$ для $i=2, 3, \dots, n$, то при $k \rightarrow \infty$

$$A^k y \approx c_1 \lambda_1^k x_1,$$

т.е. вектор y оказывается почти коллинеарен собственному вектору x_1 .

Рассмотрим отношение

$$\frac{A^{k+1} y}{A^k y} \approx \frac{c_1 \lambda_1^{k+1} x_1}{c_1 \lambda_1^k x_1} = \lambda_1.$$

Можно ожидать, что в пределе каждая компонента нашего вектора будет умножаться на одно и то же число λ_i . На практике рекомендуется нормировать получающийся вектор на каждом шаге так, чтобы наибольшая его компонента была равна 1.

На последнем шаге нормирующий множитель даст величину наибольшего по модулю собственного значения. При этом заодно получается и собственный вектор.

ВАРИАНТЫ ЗАДАНИЙ

$$(n=5 \div 10; p, q=1 \div 4; r, t=0.1 \div 1.2; b=0.01 \div 0.1)$$

1. $a_{ii} = 5i^{p/2}, \quad a_{ij} = b(i^{p/2} + j^{p/2})^{1/q}, \quad (i \neq j);$
2. $a_{ii} = 15i^{p/3}, \quad a_{ij} = b(ij)^{-q/2}, \quad (i \neq j);$
3. $a_{ii} = 10i^{p/2}, \quad a_{ij} = b(i / j^p + j / i^p)^{1/q}, \quad (i \neq j);$
4. $a_{ii} = 8i^{p/3}, \quad a_{ij} = b(i^p + j^p)^{-1/q}, \quad (i \neq j);$
5. $a_{ii} = 6i^{p/3}, \quad a_{ij} = b(i^{p/2} + j^{p/2})^{-1/q}, \quad (i \neq j);$
6. $a_{ii} = 6.5i^{p/3}, \quad a_{ij} = b \exp\left(-\frac{r}{(i+j)^t}\right), \quad (i \neq j);$

$$7. a_{ii} = 12i^{p/3}, \quad a_{ij} = b \cdot \operatorname{sh} \left(\frac{r}{(i+j)^t} \right), \quad (i \neq j);$$

$$8. a_{ii} = 11i^{p/4}, \quad a_{ij} = b \sin \left(\frac{\pi r}{2(i+j)^t} \right), \quad (i \neq j);$$

$$9. a_{ii} = 8.5i^{p/4}, \quad a_{ij} = b \cdot \operatorname{sh} |i-j|^{-1/q}, \quad (i \neq j);$$

$$10. a_{ii} = 13i^{p/3}, \quad a_{ij} = b |i^p - j^p|^{-1/q}, \quad (i \neq j).$$

ПРОГРАММИРОВАНИЕ

Блок-схема программы представлена на рис. 9.1. В блоках 3-8 вычисляются элементы матрицы A и начальный вектор $x^{(0)}$, в блоках 10-13 реализован степенной метод, в блоках 9,14 выводятся промежуточные и конечные результаты.

В качестве $x^{(0)}$ следует принять случайный вектор, элементы которого представляют собой случайные числа, равномерно распределенные в интервале $(0,1)$. Для выработки таких чисел в программном обеспечении компьютера имеется специальный генератор (датчик). Итерации следует заканчивать при выполнении какого-либо из двух условий

$$|\lambda^{(k)} - \lambda^{(k-1)}| < \varepsilon, \quad k > k_{\max},$$

где $\varepsilon = 0.1 \div 0.001$ - заданная погрешность, k_{\max} - максимальное число итераций. Окончание итераций при выполнении последнего условия означает, что за k_{\max} итераций заданная погрешность не достигнута.

Проверку программы рекомендуется провести на тестовой диагональной матрице с элементами

$$a_{ii} = i, \quad a_{ij} = 0 \quad (i, j = 1, 2, \dots, n),$$

для которой $\lambda_1(A) = n$.

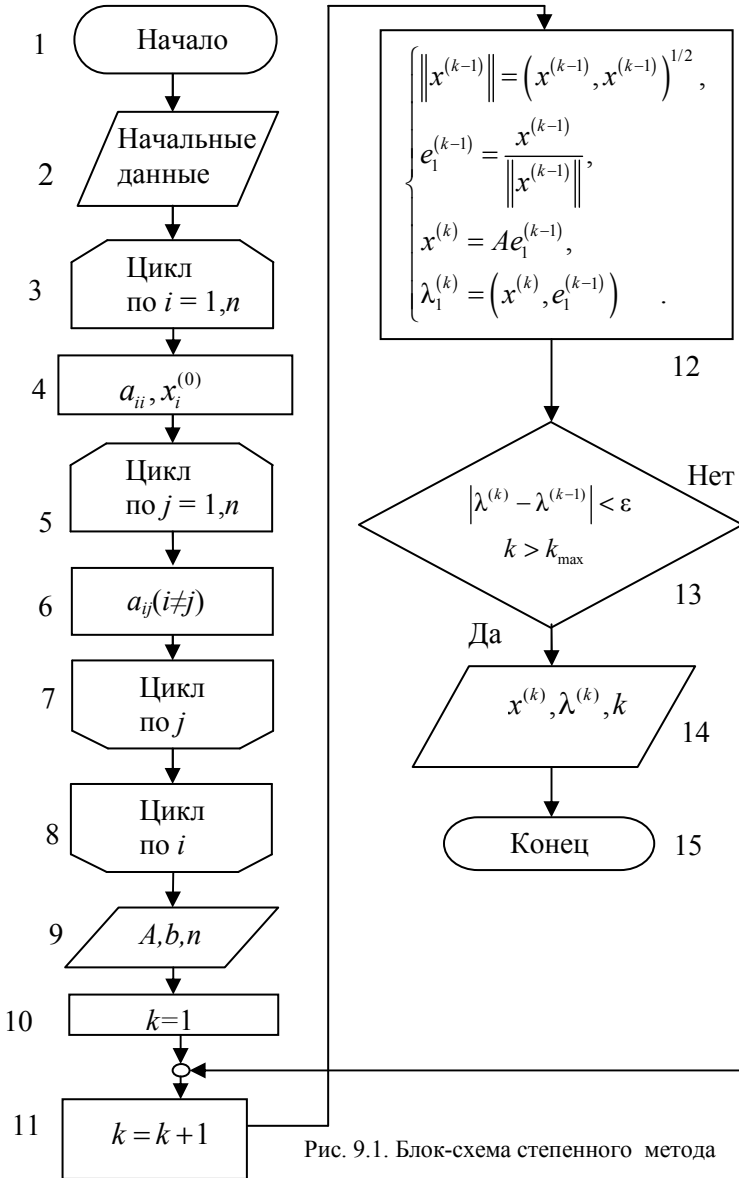


Рис. 9.1. Блок-схема степенного метода

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы для конкретного варианта;
- текст программы;
- результаты решения: исходная матрица, значение $\lambda_1(A)$ и число итераций k .

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что понимается под частичной проблемой собственных значений?
2. Как строится степенной метод вычисления максимального по модулю собственного значения?
3. Какие методы решения полной проблемы собственных значений вы знаете?
4. Что такое характеристический полином?

Задание № 10

ПОИСК МИНИМУМА ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ

Цель работы: изучение методов одномерной минимизация, нахождение минимума унимодальной функции одной переменной методом "золотого сечения".

ПОСТАНОВКА ЗАДАЧИ

Пусть непрерывная функция $\Phi(x)$ имеет в интересующей нас области

$$a \leq x \leq b$$

единственный минимум \bar{x} . Значение \bar{x} называют модой, а функцию $\Phi(x)$, имеющую единственную моду, - унимодальной. Для отыскания минимума обычно используются так называемые симметричные методы, которые строятся следующим образом:

1. На отрезке $[a, b]$ выбирается точка

$$a < y < b.$$

2. Далее берется симметричная ей точка z (рис. 10.1) такая, что

$$y - a = b - z.$$

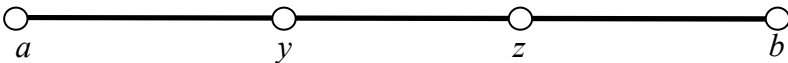


Рис. 10.1. Расположение точек в симметричных методах поиска минимума

Отсюда находим:

$$z = a + b - y.$$

Если оказывается, что

$$\Phi(y) < \Phi(z),$$

то для дальнейшего рассмотрения выбирается отрезок $[a, z]$ в противном случае - отрезок $[y, b]$. Далее итерационный процесс повторяется.

Наиболее экономичными являются симметричные методы, в которых требуется однократное вычисление функции $\Phi(x)$ на каждой

итерации, кроме первой, на которой $\Phi(x)$, вычисляется дважды. Рассмотрим схему таких симметричных методов. Пусть на k -й итерации минимум отыскивается на отрезке $[a_{k-1}, b_{k-1}]$ ($k = 1, 2, \dots$), где $a_0 = a$, $b_0 = b$, причём уже известны из предыдущей итерации одна из двух внутренних точек, например, y_k (рис. 10.2) и соответствующее значение $\Phi(y_k)$.

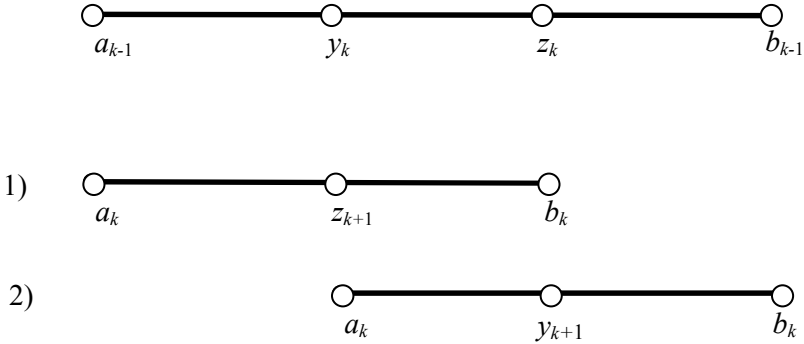


Рис. 10.2. Выполнение итераций в симметричных методах поиска минимума:

1. случай $\Phi(z_k) \geq \Phi(y_k)$; 2. случай $\Phi(z_k) < \Phi(y_k)$

Положим, что

$$y_k = a_{k-1} + c_{k-1} (b_{k-1} - a_{k-1}) \quad (k=1, 2, \dots),$$

где c_k - некоторый параметр:

$$0 < c_k < 0.5 \quad (k = 0, 1, \dots).$$

Находим симметричную точку z_k :

$$z_k = a_{k-1} + b_{k-1} - y_k = a_{k-1} + (1 - c_{k-1})(b_{k-1} - a_{k-1}).$$

Далее вычисляем $\Phi(z_k)$ и полагаем (рис. 10.2):

1) если $\Phi(z_k) \geq \Phi(y_k)$, то

$$a_k = a_{k-1}, \quad b_k = z_k, \quad z_{k+1} = y_k, \quad \Phi(z_{k+1}) = \Phi(y_k);$$

2) если $\Phi(z_k) < \Phi(y_k)$, то

$$a_k = y_k, \quad b_k = b_{k-1}, \quad y_{k+1} = z_k, \quad \Phi(y_{k+1}) = \Phi(z_k).$$

Для дальнейшего рассмотрения выбираем отрезок $[a_k, b_k]$, длина которого

$$b_k - a_k = (1 - c_{k-1})(b_{k-1} - a_{k-1}).$$

Итерации прекращаются, если

$$b_k - a_k < 2\varepsilon,$$

где ε - заданная погрешность, и в качестве \bar{x} принимается значение

$$\bar{x} = 0.5(a_k + b_k).$$

Различные симметричные методы различаются выбором параметров c_k . Методы, в которых параметры c_k постоянны, т.е.

$$c_k = c = \text{const},$$

называются стационарными, а методы, в которых параметры c_k зависят от номера итерации k , - нестационарными.

Наиболее быстрым из всех симметричных методов с однократным вычислением $\Phi(x)$ на итерации является нестационарный метод Фибоначчи. Незначительно уступает ему в скорости стационарный метод "золотого сечения", в котором

$$c = \text{const} = \frac{2}{3 + \sqrt{5}} = 0.381966. \quad (10.1)$$

В этом методе выполняется равенство

$$\frac{z_k - a_{k-1}}{b_{k-1} - a_{k-1}} = \frac{b_{k-1} - z_k}{z_k - a_{k-1}}, \quad (10.2)$$

о котором говорят, что точка z_k делит отрезок $[a_{k-1}, b_{k-1}]$ "в среднем и крайнем отношении". Из (10.2) с учетом

$$b_{k-1} - z_k = y_k - a_{k-1}$$

и вытекает равенство (10.1).

Метод "золотого сечения" обеспечивает линейную сходимость к минимуму \bar{x} , т.е. сходимость со скоростью геометрической прогрессии, имеющей знаменатель

$$q = 1 - c = 0.618034 \approx 0.62$$

(сходимость первого порядка) и применим к не дифференцируемым функциям $\Phi(x)$.

ВАРИАНТЫ ЗАДАНИЙ

Параметры: $a=-1$, $b=1$, $\varepsilon=10^{-3} \div 10^{-4}$.

- 1) $\Phi(x) = \sum_{i=0}^5 c_i |x|^i \quad (0 \leq c_i \leq 1; i = 0 \div 5);$
- 2) $\Phi(x) = \left| \sin |x|^m \right|^n \quad (n, m = 1 \text{ или } 2);$
- 3) $\Phi(x) = |x|^m e^{n|x|} \quad (n, m = 1, 2, 3);$
- 4) $\Phi(x) = \left(\arcsin \left(|x|^m / 2 \right) \right)^n \quad (n, m = 1 \text{ или } 2);$
- 5) $\Phi(x) = \left(\operatorname{tg} |x|^m \right)^n \quad (n, m = 1 \text{ или } 2);$
- 6) $\Phi(x) = \left| \operatorname{arctg} |x|^m \right|^n \quad (n, m = 1 \text{ или } 2);$
- 7) $\Phi(x) = |x|^m \operatorname{arctg} |x|^n \quad (n, m = 1 \text{ или } 2);$
- 8) $\Phi(x) = |x|^m \left(\sin |x| \right)^n \quad (n, m = 1 \text{ или } 2);$
- 9) $\Phi(x) = |x|^m \left(\operatorname{tg} |x| \right)^n \quad (n, m = 1 \text{ или } 2);$
- 10) $\Phi(x) = |x|^m \sin \left(|x|^n \right) \quad (n, m = 1 \text{ или } 2);$
- 11) $\Phi(x) = |x|^m \operatorname{tg} \left(|x|^n \right) \quad (n, m = 1 \text{ или } 2);$
- 12) $\Phi(x) = |x|^m \left| \operatorname{arctg} x \right|^n \quad (n, m = 1 \text{ или } 2);$
- 13) $\Phi(x) = |x|^m e^{|x|^n} \quad (n, m = 1, 2, 3);$
- 14) $\Phi(x) = \left| \ln \left(1 + |x|^m \right) \right|^n \quad (n, m = 1 \text{ или } 2);$
- 15) $\Phi(x) = |x|^m \operatorname{sh} |x|^n \quad (n, m = 1, 2, 3);$
- 16) $\Phi(x) = |x|^m \operatorname{sh}^n x \quad (n, m = 1, 2, 3);$
- 17) $\Phi(x) = \left| \sin^m \left(x^n \right) \right| \quad (n, m = 1 \text{ или } 2);$
- 18) $\Phi(x) = \left| 1 - \cos^m \left(x^n \right) \right| \quad (n, m = 1, 2, 3);$

$$19) \quad \Phi(x) = \left| 1 - \operatorname{ch}^m(x^n) \right| \quad (n, m = 1 \text{ или } 2);$$

$$20) \quad \Phi(x) = \left| 1 - e^{m|x|} \right|^n \quad (n, m = 1, 2, 3);$$

$$21) \quad \Phi(x) = \left| \operatorname{sh}|x|^n \right|^m \quad (n, m = 1 \text{ или } 2);$$

$$22) \quad \Phi(x) = \left| \sin x^m \cos x^n \right| \quad (n, m = 1 \text{ или } 2).$$

ПРОГРАММИРОВАНИЕ

Все приведенные в задании варианты функций $\Phi(x)$ имеют нулевой минимум в точке $x = 0$. Вычисление $\Phi(x)$ целесообразно оформить в виде функции, а метод "золотого сечения" - в виде процедуры, одним из аргументов которой является имя функции.

Чтобы предотвратить "зацикливание" программы в случае отсутствия сходимости (например, из-за каких-либо ошибок в программе или в исходных данных), рекомендуется указать максимальное число итераций $k_{\max} \approx 50$, по достижении которого итерации прекращаются.

Блок-схема алгоритма программы приведена на рис. 10.3. Вычисление минимума организовано в цикле 3-10, блок 8 представляет собой подпрограмму вычисления функции $\Phi(x)$. Для наглядного представления процесса сходимости удобно построить зависимость

$$l(k) = 0.5(b_k - a_k),$$

или зависимость

$$\delta(k) = -\lg l(k) = -\lg \frac{b_k - a_k}{2} \quad (k = 0, 1, 2, \dots).$$

В соответствии с теорией метода зависимость $\delta(k)$ должна быть близка к линейной.

В качестве тестовой функции использовать

$$\Phi(x) = 1 - 4x(1-x),$$

для которой

$$x_{\min} = 0.5, \quad \Phi_{\min} = 0, \quad a = -1, \quad b = 1, \quad \varepsilon = 10^{-3} \div 10^{-4}.$$

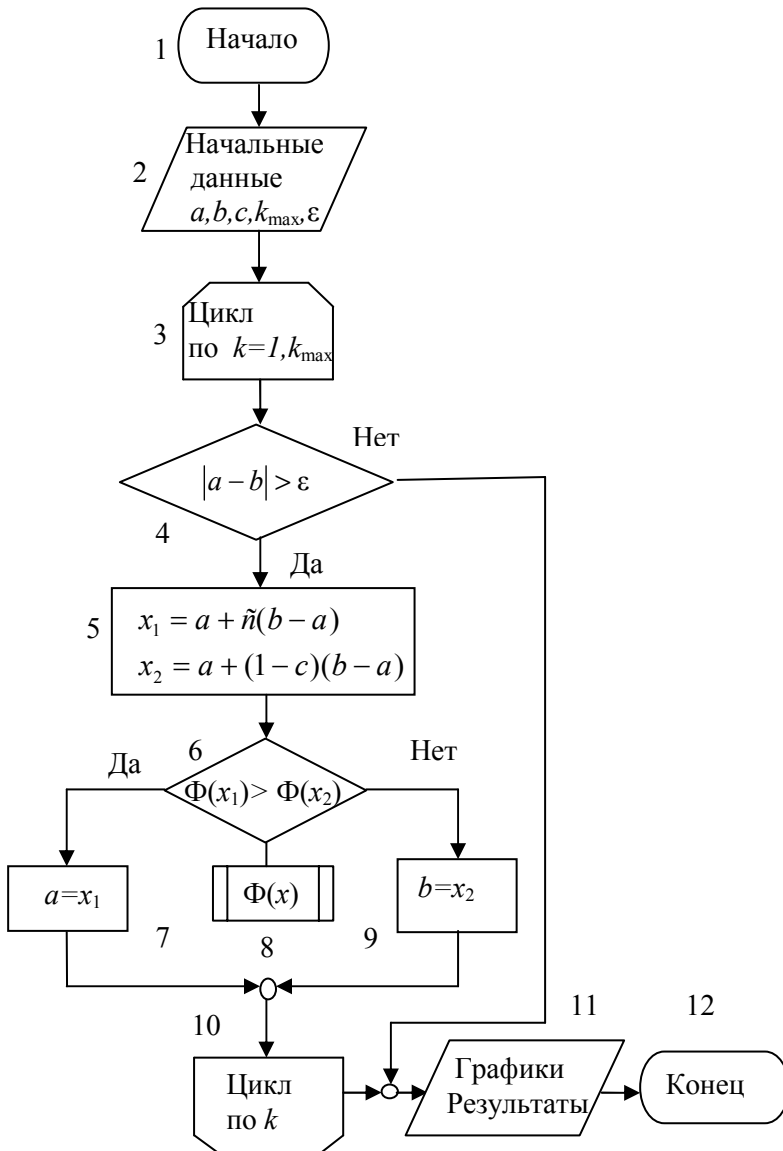


Рис. 10.3. Блок-схема программы поиска минимума функции одной переменной методом золотого сечения

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулу и график функции $\Phi(x)$, параметры для конкретного варианта;
- текст программы;
- значения \bar{x} , k (число итераций), зависимость $\delta(k)$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как строятся симметричные методы поиска минимума функций одной переменной?
2. Как строится метод "золотого сечения", каковы его особенности?
3. Какие еще существуют методы поиска минимума функций одной переменной, каковы их сравнительные характеристики?
4. Как связаны задача поиска минимума и задача решения нелинейных уравнений?

Задание № 11

ПОИСК МИНИМУМА ФУНКЦИИ ДВУХ ПЕРЕМЕННЫХ

Цель работы: изучение методов поиска минимума функций нескольких переменных.

ПОСТАНОВКА ЗАДАЧИ

Для поиска минимума функций нескольких переменных часто используются методы спуска. Простейшим из этих методов является координатный спуск, который строится следующим образом. Пусть $x^{(0)}$ — начальное приближение к моде \bar{x} , которое должно быть задано. Зафиксируем координаты $x_2^{(0)}, x_3^{(0)}, \dots, x_n^{(0)}$ и найдем значение $x_1^{(1)}$ как решение задачи

$$x_1^{(1)} = \arg \min \Phi(x_1, x_2^{(0)}, x_3^{(0)}, \dots, x_n^{(0)})$$

каким-либо методом поиска минимума функции одной переменной, например, методом «золотого сечения».

Геометрически это означает, что минимум отыскивается вдоль прямой, проходящей через точку $x^{(0)}$ параллельно оси x_1 , причем $x_1^{(1)}$ — координата точки, в которой эта прямая касается некоторой поверхности уровня (рис. 11.1).

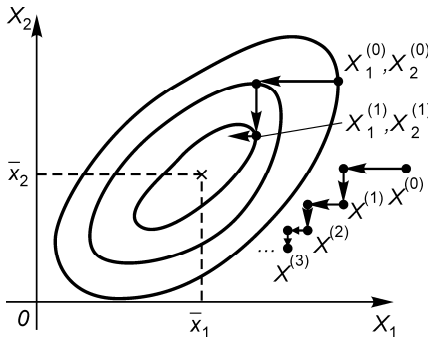


Рис. 11.1. Выполнение итераций и траектория в методе координатного спуска

Далее, аналогично находим $x_2^{(1)}, x_3^{(1)}, \dots, x_n^{(1)}$ как решения задач одномерной минимизации:

$$\begin{cases} x_2^{(1)} = \arg \min \Phi(x_1^{(1)}, x_2, x_3^{(0)}, \dots, x_n^{(0)}), \\ x_3^{(1)} = \arg \min \Phi(x_1^{(1)}, x_2^{(1)}, x_3, x_4^{(0)}, \dots, x_n^{(0)}), \\ \dots \\ x_n^{(1)} = \arg \min \Phi(x_1^{(1)}, x_2^{(1)}, \dots, x_{n-1}^{(1)}, x_n). \end{cases}$$

На этом первая итерация заканчивается. Как видно, итерационный процесс строится как последовательность решений одномерных задач минимизации по отдельным координатам. Далее, аналогично строятся следующие итерации.

Доказано, что в котловине метод координатного спуска сходится к минимуму со скоростью геометрической прогрессии, имеющей знаменатель $q < 1$:

$$\|x^{(k)} - \bar{x}\| \leq q \|x^{(k-1)} - \bar{x}\| \quad (0 < q < 1)$$

(сходимость первого порядка). Однако во многих случаях (например, если рельеф близок к *разрешимому оврагу*) знаменатель прогрессии может быть близок к 1:

$$q \approx 1,$$

и сходимость оказывается очень медленной.

ВАРИАНТЫ ЗАДАНИЙ

Методом координатного спуска найти минимум функции двух переменных

$$f(x, y) = \left(\frac{\alpha x - \beta y}{a} \right)^m + \left(\frac{\alpha y + \beta x}{b} \right)^n$$

$$\alpha = \sqrt{1 - \beta^2}$$

$$a, b = 1 \div 0.001; \quad m, n = 2, 4, 6; \quad \beta = 0 \div 1.$$

Точные значения: $x_{\min} = y_{\min} = 0; f(x_{\min}, y_{\min}) = 0$.

В качестве начальной точки (0-й итерации) принять

$$x^{(0)}=y^{(0)}=d,$$

где

$$d = \sqrt{a^2 + b^2},$$

а в качестве интервала поиска минимумов по x и по y — интервал $[-d, d]$.

Условия окончания итераций:

$$1) \quad \delta^{(k)} = \sqrt{(x^{(k)} - x^{(k-1)})^2 + (y^{(k)} - y^{(k-1)})^2} < \varepsilon = 10^{-2} \div 10^{-4}$$

($k = 1, 2, 3 \dots$ — номер итерации);

$$2) \quad k > k_{\max} = 10000.$$

ПРОГРАММИРОВАНИЕ

Блок-схема программы поиска минимума функции двух переменных методом координатного спуска приведена на рис. 11.2. Метод реализован в цикле 4-10, блок 7 представляет собой подпрограмму вычисления минимума функции одной переменной, такую, как, например, в задании № 10.

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- исходные данные $a, b, \beta, m, n, \varepsilon$;
- текст программы;
- график функции $f(x, y)$;
- результаты решения: $k, x^{(k)}, y^{(k)}$, а также зависимость от k величины

$$q^{(k)} = -\ln(\delta^{(k)}) / \ln 10.$$

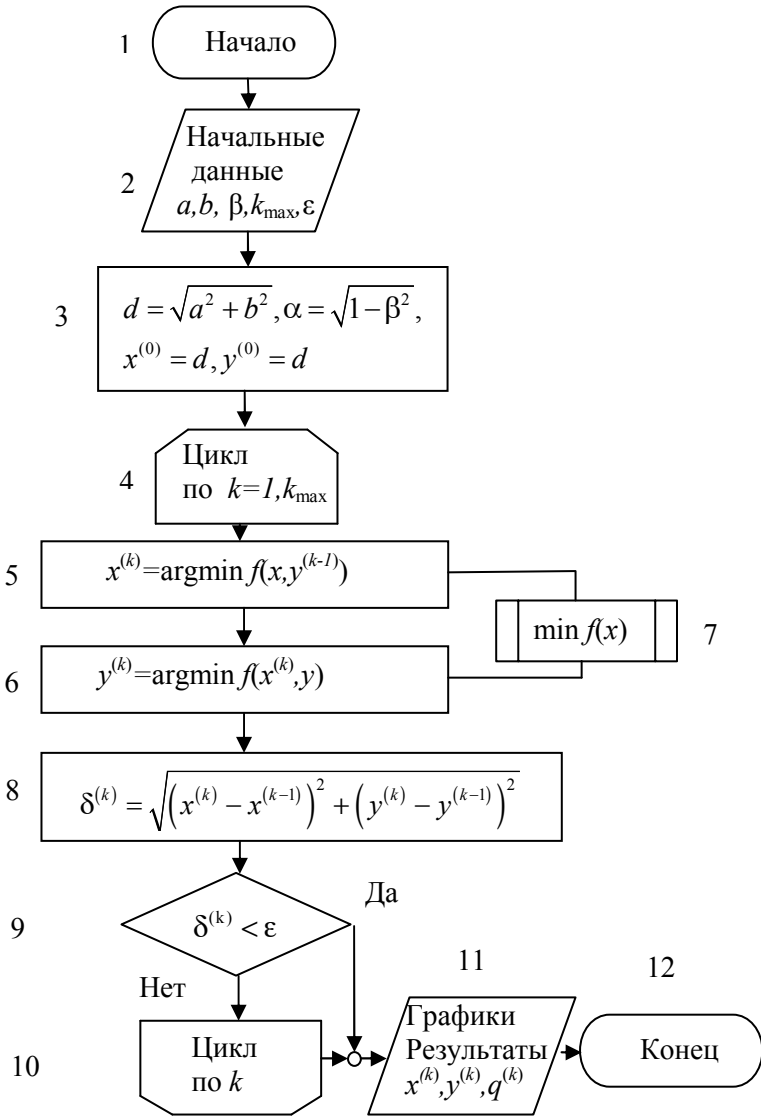


Рис. 11.2. Блок-схема программы метода координатного спуска

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие методы поиска минимума функций нескольких переменных вы знаете?
2. Как можно ускорить сходимость итераций в методе координатного спуска?
3. В чём суть метода наискорейшего спуска?
4. В чём идея метода сопряженных направлений?
5. Какие методы поиска минимума функций нескольких переменных можно использовать, если функция имеет овражный рельеф?

Задание № 12

ЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧИ КОШИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Цель работы — изучение численных методов решения систем обыкновенных дифференциальных уравнений, решение задачи Коши для систем двух уравнений методом Рунге-Кутты второго порядка.

ПОСТАНОВКА ЗАДАЧИ

Решение простейшего обыкновенного дифференциального уравнения

$$\frac{du}{dt} = f(t) \quad (12.1)$$

начиная с некоторого момента $t=t_0$ имеет вид

$$u(t) = \int_{t_0}^t f(\tau) d\tau + c, \quad t_0 \leq t \leq T. \quad (12.2)$$

То есть очевидна связь между задачами интегрирования и решения дифференциальных уравнений. Однако есть и существенные отличия. Одно из важнейших заключается в том, что при интегрировании результатом является единственное число — величина интеграла, а решая дифференциальное уравнение мы ищем функцию, определённую на некотором отрезке.

Поскольку функция может быть вычислена лишь в конечном числе точек (компьютер не может выполнить бесконечное число операций), то при численном интегрировании дифференциальных уравнений необходимо перейти от непрерывного решения на отрезке к дискретному. Таким образом, решение $u(t)$ будет вычисляться лишь в конечном числе точек t_i (в фиксированные моменты времени, если t - время), а не на всём непрерывном отрезке; $t_i = t_0 + \Delta t_i$, где Δt_i - шаг интегрирования.

Простейшим методом решения дифференциальных уравнений является метод Эйлера, основанный на предположении, что правая

часть уравнения (12.1) слабо изменяется на шаге интегрирования, то есть в (12.2) $f(\tau)=\text{const}$. Иными словами, решение можно получить, используя для численного интегрирования в (12.2) метод прямоугольников:

$$u(t_{i+1}) = u(t_i) + \Delta t_i f(t_i). \quad (12.3)$$

К формуле (12.3) можно также прийти, разложив функцию $u(t)$ в точке t_i в ряд Тейлора с шагом Δt_i и отбросив все члены с производной выше первого порядка. Оценивая остаточный член разложения, можно сделать вывод о том, что метод Эйлера имеет первый порядок погрешности относительно шага интегрирования, то есть уменьшение шага интегрирования вдвое, приводит к такому же уменьшению погрешности.

Точность метода можно увеличить, оставляя в разложении в ряд Тейлора члены со второй производной и т.д.

Одним из наиболее распространенных методов численного решения задачи Коши для систем обыкновенных дифференциальных уравнений вида

$$\begin{cases} \frac{du}{dt} = f(u, t); \\ u = (u_1, u_2, \dots, u_n), \quad f(u, t) = (f_1(u, t), f_2(u, t), \dots, f_n(u, t)), \\ u(t^0) = u^0, \end{cases}$$

является метод «предиктор—корректор» второго порядка. В этом методе решение на каждом временном шаге τ складывается из двух этапов:

Первый этап («предиктор»), на котором вычисляется предварительное значение на половинном шаге

$$u^+ = u + \frac{\Delta t}{2} f.$$

Второй этап («корректор»), на котором вычисляется окончательное значение на следующем шаге

$$\hat{u} = u + \Delta t f^+, \quad f^+ = f\left(u^+, t + \frac{\Delta t}{2}\right).$$

Здесь обозначено

$$u = u(t), \quad u^+ = u\left(t + \frac{\Delta t}{2}\right), \quad \hat{u} = u(t + \Delta t).$$

Описанный метод относится к методам типа Рунге-Кутта. Поскольку он имеет на шаге погрешность $O(\Delta t^3)$, а на всем интервале интегрирования — погрешность $O(\Delta t^2)$, его называют также методом Рунге-Кутта второго порядка.

ВАРИАНТЫ ЗАДАНИЙ

Задание 1. Вычислить траекторию заряженной частицы с зарядом q и массой покоя m_0 в потенциальном поле $\varphi(x)$.

Формулы уравнений движения:

$$\begin{cases} \frac{dx}{dt} = v, & (0 \leq t \leq T), \\ \frac{dp}{dt} = qE, & E = -\frac{d\varphi}{dx}, \end{cases}$$

$$x(0) = 0, \quad v(0) = v_0, \quad p = \gamma m_0 v, \quad \gamma = \sqrt{1 + \left(\frac{p}{m_0 c}\right)^2}.$$

Сорта частиц: электроны, протоны, α -частицы.

ВАРИАНТЫ ЗАВИСИМОСТЕЙ $\varphi(x)$

($\sigma, r, s = 1 \div 3$; $\varphi_0 = \pm(100 \div 1000)$, $v_0 = (0.5 \div 1)10^6$, $\tau = 10^{-12} \div 10^{-8}$)

- | | |
|---|--|
| 1) $\varphi_0 x^\sigma (1 + \sigma x^r)^s$ | 2) $\varphi_0 x^\sigma (\text{sh}^\sigma x^r - 1)^2$ |
| 3) $\varphi_0 \text{sh}^\sigma x^r$ | 4) $\varphi_0 [\exp(x^\sigma)]^r$ |
| 5) $\varphi_0 (\text{ch}^\sigma x^r - 1)^s$ | 6) $\varphi_0 (\text{sh} x^2 - x)^\sigma$ |
| 7) $\varphi_0 \arctg^\sigma x^r$ | 8) $\varphi_0 \exp(b \cdot \arctg x^\sigma)$ |
| 9) $\varphi_0 \text{tg}^\sigma x^r$ | 10) $\varphi_0 (1 \pm ax^r \pm bx^3)^2$ |
| 11) $\varphi_0 \sin^2 x^\sigma$ | 12) $\varphi_0 \cos^2 x^\sigma$ |

Задание 2. Промоделировать колебания в контуре.

Вынужденные колебания в электрическом RLC -контуре описываются системой обыкновенных дифференциальных уравнений

$$\frac{dQ}{dt} = I \quad (0 \leq t \leq t_k);$$

$$\frac{dI}{dt} = -2\beta I - \omega_0^2 Q + E(Q, I, t);$$

$$I(0) = 0; \quad Q(0) = Q_0;$$

$$\beta = 0.5R/L; \quad \omega_0^2 = 1/(CL).$$

Здесь R — сопротивление, L — индуктивность, C — ёмкость контура; Q — заряд на конденсаторе, I — ток в контуре, ω_0 — собственная частота контура, β — коэффициент затухания, E — вынуждающая сила, $\beta = 0 \div 0.2$, $\omega_0 = 1 \div 2$, $Q_0 = 1 \div 10$, $\tau = 0.05 \cdot 2\pi/\omega_0$.

ВАРИАНТЫ ЗАВИСИМОСТЕЙ $E(Q, I, t)$

$$(p, q, r, s = 0 \div 3; \omega = 1 \div 4, \varphi_0 = -\pi/2 \div \pi/2)$$

- 1) $(a+b|Q|^p+c|I|^q) t^r \sin(\omega t + \varphi_0)$;
- 2) $(a+b|Q|^p+c|I|^q+d|Q|^s) t^r \sin(\omega t + \varphi_0)$;
- 3) $a|Q|^p |I|^q e^{\pm rt} \sin(\omega t + \varphi_0)$;
- 4) $a|Q|^p |I|^q t^r \sin(\omega t + \varphi_0)$;
- 5) $(a+b|Q|^p+c|I|^q) e^{\pm rt} \sin(\omega t + \varphi_0)$;
- 6) $(a+b|Q|^p+c|I|^q+d|Q|^s) e^{\pm rt} \sin(\omega t + \varphi_0)$.

ПРОГРАММИРОВАНИЕ

Блок-схема программы для первого варианта задания приведена на рис. 12.1. Метод «предиктор—корректор» второго порядка реализован в блоке 5. Зависимости координаты и импульса частицы накапливаются в одномерных массивах $x(i), p(i)$ и затем используются для построения фазовых траекторий.

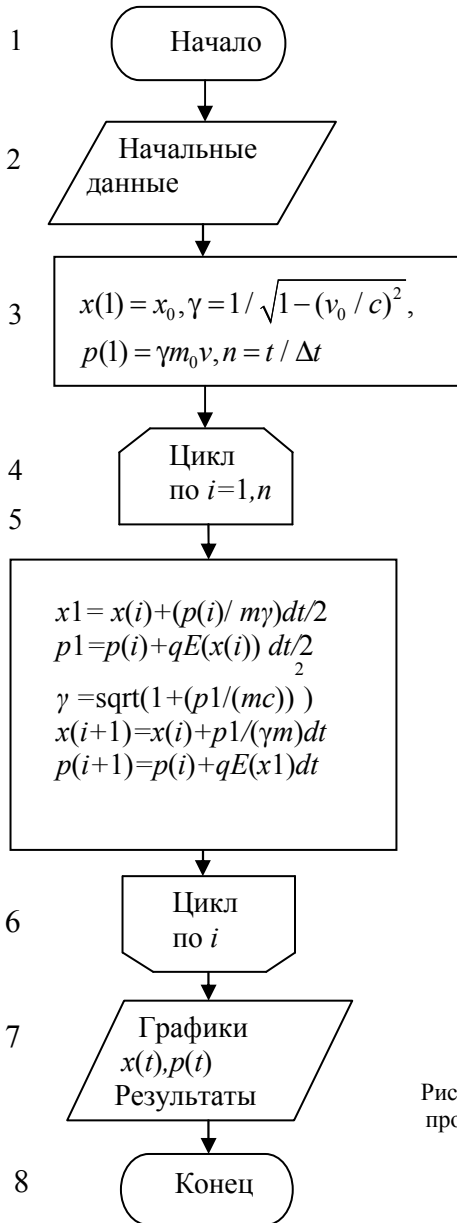


Рис. 12.1. Блок-схема программы решения задачи Коши

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы и параметры для конкретного варианта;
- текст программы;
- результаты решения, графики зависимостей.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как выполняется решение систем обыкновенных дифференциальных уравнений методом "предиктор-корректор"?
2. Каковы особенности метода Рунге—Кутты второго порядка?
3. Какие ещё существуют методы численного решения систем обыкновенных дифференциальных уравнений?
4. Чем отличается локальная ошибка от глобальной?
5. Какие из методов, явные или неявные, более устойчивы?

Задание № 13

ЧИСЛЕННОЕ РЕШЕНИЕ ЛИНЕЙНОЙ КРАЕВОЙ ЗАДАЧИ ДЛЯ ОБЫКНОВЕННОГО ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ ВТОРОГО ПОРЯДКА

Цель работы — изучение численных методов решения линейных краевых задач для обыкновенных дифференциальных уравнений, численное решение линейной краевой задачи для уравнения второго порядка.

ПОСТАНОВКА ЗАДАЧИ

Краевой задачей называется задача нахождения решения системы (13.1) ($n \geq 2$), для которой дополнительные условия задаются более чем в одной точке. Здесь ограничимся линейной краевой задачей (линейными называются краевые задачи для линейных уравнений с линейными краевыми условиями) для одного уравнения второго порядка:

$$\begin{aligned} y'' + p(x)y' + q(x)y &= f(x) \quad (x_0 \leq x \leq x_N), \\ y(x_0) &= y_0, \quad y(x_N) = y_N. \end{aligned} \quad (13.1)$$

Приближенное решение будем искать в виде линейной комбинации базисных функций:

$$y(x) = \varphi_0(x) + a_1 \varphi_1(x) + a_2 \varphi_2(x) + \dots + a_n \varphi_n(x).$$

В зависимости от способа определения коэффициентов разложения a_i можно выделить методы коллокаций, наименьших квадратов, метод Галеркина и его модификации. Все они сводятся к построению системы линейных алгебраических уравнений, из которой, если существует ее решение, находятся неизвестные коэффициенты разложения.

Другим возможным путём решения (13.1) является сведения её к последовательности задачи с начальными условиями

$$y(x_0) = y_0, \quad y'(x_0) = \operatorname{tg} \alpha.$$

Здесь y_0 — точка на оси ординат, в которой помещается начало искомой интегральной кривой; α — угол наклона касательной к интегральной кривой в этой точке. Считая решение задачи Коши $y = y(x, \alpha)$ зависящим от параметра α , будем искать такую интегральную кривую $y = y(x, \alpha^*)$, которая выходит из точки (x_0, y_0) и попадает в точку (x_N, y_N) .

Следовательно, для нахождения параметра α получим уравнение вида $F(\alpha) = 0$, где $F(\alpha) = y(x_N, \alpha) - y_N$. Это уравнение отличается от привычной записи задания № 2 тем, что функцию $F(\alpha)$ нельзя представить в виде некоторого аналитического выражения, поскольку она является решением задачи Коши. Тем не менее, для решения уравнения может быть использован любой из рассмотренных ранее методов решения нелинейных уравнений. Описанный алгоритм называется *методом стрельбы*, поскольку в нем как бы проводится «пристрелка» по углу наклона интегральной кривой α в начальной точке.

В методе конечных разностей производные, входящие в дифференциальное уравнение заменяются их конечно-разностными аппроксимациями. Таким образом, краевая задача для дифференциального уравнения сводится к системы алгебраических уравнений относительно значений искомой функции на заданном множестве точек.

Решение (13.1) будем отыскивать на сетке с постоянным шагом h (такие сетки называются равномерными) и узлами

$$x_j = x_0 + jh \quad (j = 0, 1, \dots, N), \quad h = (x_N - x_0)/N,$$

где N — число шагов.

Аппроксимируя производные конечно-разностными формулами

$$y_j'' = \frac{y_{j+1} - 2y_j + y_{j-1}}{h^2} + O(h^2), \quad y_j' = \frac{y_{j+1} - y_{j-1}}{2h} + O(h^2),$$

$$y_j = y(x_j) \quad (j=0, 1, 2, \dots, N),$$

приведем краевую задачу к системе линейных алгебраических уравнений

$$a_j y_{j-1} - b_j y_j + c_j y_{j+1} = d_j \quad (j=0, 1, 2, \dots, N-1), \quad (13.2)$$

где

$$\begin{cases} a_j = 1 - 0.5p_j h, & b_j = 2 - q_j h^2, \\ c_j = 1 + 0.5p_j h, & d_j = f_j h^2, \\ p_j = p(x_j), & q_j = q(x_j), & f_j = f(x_j). \end{cases}$$

Система уравнений имеет трехдиагональную матрицу, так как каждое уравнение содержит лишь три соседних неизвестных $y_j, y_{j\pm 1}$. Экономным методом решения таких систем является метод прогонки, представляющий собой разновидность метода исключения Гаусса. В этом методе решение отыскивается в виде

$$y_j = \xi_{j+1} y_{j+1} + \eta_{j+1} \quad (j = 1, 2, \dots, N-1), \quad (13.3)$$

где ξ_{j+1}, η_{j+1} — пока не известные коэффициенты, называемые прогоночными.

Заменим j на $j-1$ и, подставляя $y_{j-1} = \xi_j y_j + \eta_j$ в уравнение (13.2), преобразуем его к виду

$$y_j = \frac{c_j}{b_j - a_j \xi_j} y_{j+1} + \frac{a_j \eta_j - d_j}{b_j - a_j \xi_j}.$$

Сравнивая теперь полученное y_j с (13.3), находим рекуррентные формулы для прогоночных коэффициентов

$$\xi_{j+1} = \frac{c_j}{b_j - a_j \xi_j}, \quad \eta_{j+1} = \frac{a_j \eta_j - d_j}{b_j - a_j \xi_j} \quad (j = 1, 2, \dots, N-1). \quad (13.4)$$

При $j = 0$, находим, что

$$\xi_1 = 0, \quad \eta_1 = y_0. \quad (13.5)$$

Таким образом, метод решения складывается из двух этапов.

На первом этапе (прямой ход прогонки) вычисляются прогоночные коэффициенты по рекуррентным формулам (13.4) с известными начальными значениями (13.5).

На втором этапе (обратный ход прогонки) по формуле (13.3) вычисляются y_j ($j = N-1, N-2, \dots, 1$) с учетом заданного краевого условия y_N .

Погрешность решения имеет тот же порядок, с которым аппроксимируются производные, т. е. $O(h^2)$.

ВАРИАНТЫ ЗАДАНИЙ

Вычислить установившееся распределение концентрации газа в одномерной системе из краевой задачи

$$D \frac{d^2 u}{dx^2} = f(x) \quad (0 < x < l),$$

$$D = \text{const};$$

$$u(0) = u_0, \quad u(l) = u_l,$$

где D — коэффициент диффузии, $f(x)$ — плотность источников газовой выделения.

ВАРИАНТЫ ФУНКЦИЙ

$$(0 \leq \alpha \leq x \leq \beta \leq l, \beta > \alpha, \beta - \alpha < l, l, n, r, s = 1 \div 3)$$

$$1) f(x) = \begin{cases} l & (\alpha \leq x \leq \beta), \\ 0 & (0 \leq x \leq \alpha, \beta \leq x \leq l); \end{cases}$$

$$2) f(x) = \begin{cases} c(x - \alpha) / (\gamma - \alpha) & (\alpha \leq x \leq \gamma) \\ c(\beta - x) / (\beta - \gamma) & (\gamma \leq x \leq \beta) \\ 0 & (0 \leq x < \alpha, \beta < x \leq l); \end{cases}$$

$$3) f(x) = |\sin(\pi n x / l)|^s;$$

$$4) f(x) = |\cos(\pi n x / l)|^s;$$

$$5) f(x) = \begin{cases} (x - \alpha)^r (\beta - x)^s & (\alpha \leq x \leq \beta) \\ 0 & (0 \leq x < \alpha, \beta < x \leq l). \end{cases}$$

ПРОГРАММИРОВАНИЕ

Функцию $f(x)$ в правой части удобно описать с помощью подпрограммы-функции. Для d_1, ξ_1, η_1 надо задать одномерные массивы.

Для проверки программы можно предварительно решить тестовую задачу при $f(x) \equiv 0$, имеющую аналитическое решение

$$u(x) = u_0 + (u_1 - u_0)x/l,$$

или тестовую задачу при $f(x) \equiv D$, имеющую аналитическое решение

$$u(x) = x^2/2 + Ax + u_0, \quad A = (u_1 - u_0 - 0.5l^2)/l.$$

Блок-схема программы приведена на рис. 13.1. В цикле 3-5 вычисляются элементы трёхдиагональной матрицы и правые части системы. В цикле 7-9 находятся прогоночные коэффициенты (аналог прямого хода метода Гаусса для трёхдиагональной матрицы), а в цикле 10-12 – искомые величины, которые затем выводятся в виде графика.

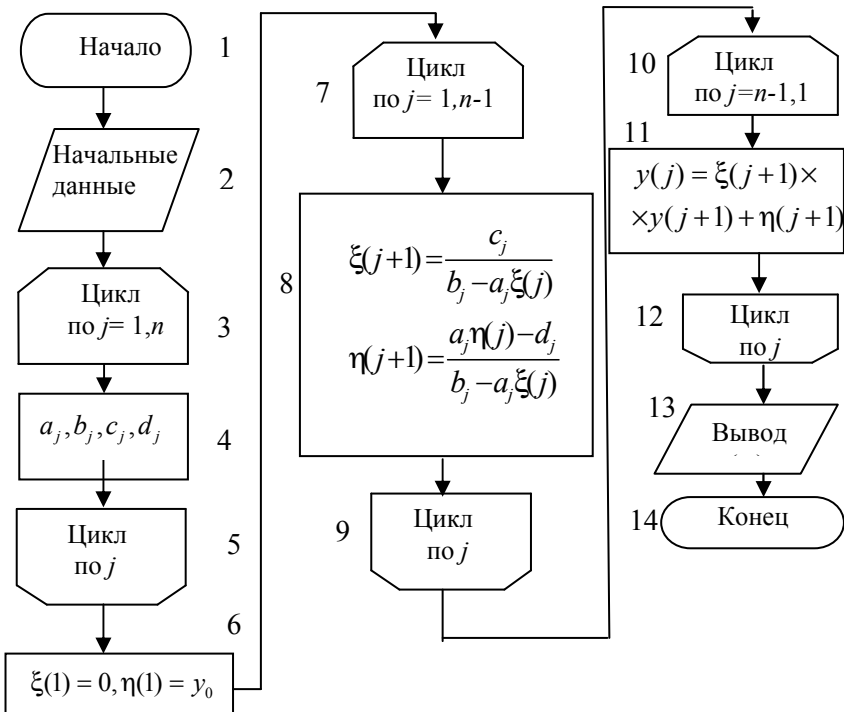


Рис. 13.1. Блок-схема программы решения краевой задачи

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы и параметры для конкретного варианта;
- текст программы;
- результаты решения, графики зависимостей.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как ставится и решается линейная краевая задача для обыкновенного дифференциального уравнения?
2. Как строится метод прогонки?
3. Какова погрешность данного метода?
4. Каковы условия устойчивости метода прогонки?
5. Какие ещё существуют методы численного решения линейных краевых задач для обыкновенных дифференциальных уравнений?

Задание № 14

ЧИСЛЕННОЕ РЕШЕНИЕ ЛИНЕЙНОГО УРАВНЕНИЯ ПЕРЕНОСА

Цель работы — изучение разностных схем и конечно-разностных методов численного решения линейного одномерного уравнения переноса, решение уравнения переноса по схеме бегущего счета.

ПОСТАНОВКА ЗАДАЧИ

В физических приложениях часто встречаются уравнения эволюционного типа:

$$\frac{\partial u}{\partial t} + L_0 u = f,$$

где дифференциальный оператор L_0 уже не содержит производных по времени t ; такие уравнения как бы описывают эволюцию начальных условий. Если L_0 — оператор первого порядка, то уравнение называется уравнением переноса и типично для задач механики сплошной среды. Примером уравнения переноса может служить известное уравнение непрерывности (закон сохранения заряда)

$$\frac{\partial \rho}{\partial t} + \operatorname{div} \vec{j} = 0, \quad \vec{j} = \rho \vec{v},$$

где ρ — плотность объемного заряда, \vec{j} — плотность тока, \vec{v} — скорость движения заряженной среды.

Простейшая линейная одномерная задача для уравнения переноса:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = f(x, t),$$

$$c = \text{const} \quad (0 < x < l), \quad u(0, t) = \varphi(t), \quad u(x, 0) = u^0(x),$$

описывает движение сплошной среды с плотностью u вдоль оси x со скоростью c ; правая часть f описывает приток (отток) вещества вследствие каких-либо сторонних процессов. Будем также для простоты полагать, что

$$\varphi(0) = u^0(0).$$

Для получения численного решения введем равномерную сетку с шагами h , τ и узлами (x_j, t^m) , где

$$\begin{aligned} x_j &= jh \quad (j = 0, 1, \dots, N), \\ h &= l/N, \quad t^m = m\tau \quad (m = 0, 1, 2, \dots). \end{aligned}$$

Значения сеточных функций принято обозначать как

$$u_j^m = u(x_j, t^m),$$

т. е. нижние индексы относятся к пространственным переменным, а верхний указывает номер момента времени. Для краткости принято в разностных схемах верхний индекс, по возможности, опускать, используя обозначения:

$$\begin{cases} u_j = u(x_j, t), & \hat{u}_j = u(x_j, t + \tau), & \check{u}_j = u(x_j, t - \tau); \\ u_j^\pm = u(x_j, t \pm \tau / 2). \end{cases}$$

Расположение узлов, на котором строится разностная схема, называют шаблоном. Рассмотрим два шаблона, показанные на рис. 14.1; кружками отмечены узлы шаблона, а звездочкой — центр соответствующей сеточной ячейки, ограниченной штриховыми линиями.

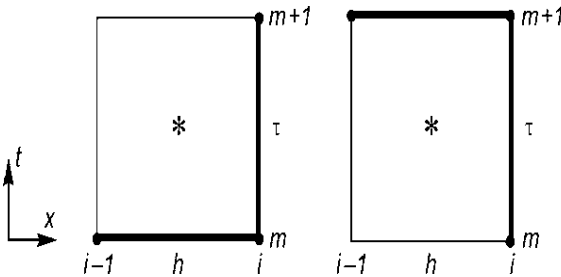


Рис. 14.1. Расположение узлов разностных схем для линейного уравнения переноса

Соответствующие двум шаблонам разностные схемы имеют вид:

$$\frac{\hat{u}_j - u_j}{\tau} + c \frac{u_j - u_{j-1}}{h} = f_j \quad (j = 1, 2, \dots, N); \quad (14.1)$$

$$\frac{\hat{u}_j - u_j}{\tau} + c \frac{\hat{u}_j - \hat{u}_{j-1}}{h} = f_j \quad (j = 1, 2, \dots, N). \quad (14.2)$$

Различия между двумя схемами заключается в том, что производная по пространственной координате берётся в разные моменты времени. Разностная схема (14.1) явная, так как из нее можно явно выразить неизвестное \hat{u}_j через уже известные значения u_j, u_{j-1} и правую часть f_j . Схема (14.2) формально неявная, так как включает два неизвестных значения \hat{u}_j, \hat{u}_{j-1} на $(m + 1)$ -м временном слое. Однако преобразовав ее к виду

$$\hat{u}_j = \frac{1}{1+r} (u_j + r\hat{u}_{j-1} + \tau f_j) \quad (r = c\tau/h, j = 1, 2, \dots, N),$$

можно явно вычислить \hat{u}_j последовательно, начиная с левой границы, на которой \hat{u}_0 известно (это заданное краевое условие). Такие схемы называют схемами «бегущего счета».

Рассмотрим теперь, что принять в качестве сеточной проекции f_j правой части f . Поскольку f зависит от x, t , наименьшая погрешность аппроксимации получается при замене $f(x, t)$ на среднее значение по площади сеточной ячейки, т. е. на значение

$$\bar{f}_j = \frac{1}{h\tau} \int_{x_{j-1}}^{x_j} dx \int_t^{t+\tau} f(x, t) dt = f(x_j - h/2, t + \tau/2) + O(h^2, \tau^2).$$

Следовательно, в уравнениях в качестве f_j будем принимать $f(x_j - h/2, t + \tau/2) = f_{j-1/2}^+$, т. е. правая часть в разностных схемах вычисляется в отмеченном звездочкой центре сеточной ячейки. Заметим, что если в качестве f_j просто принять значение в j -м узле, т. е. $f(x_j, t)$, то погрешность аппроксимации сразу возрастает на порядок.

Анализ устойчивости разностных схем показывает [3], что схема (14.1) устойчива при выполнении условия

$$0 \leq r \leq 1 \quad (r = c\tau/h), \quad (14.3)$$

т. е. это условно устойчивая схема. Условие устойчивости имеет простой физический смысл: вещество среды распространяется в положительном направлении оси абсцисс, причем расстояние $c\tau$, на которое смещается вещество за один временной шаг τ , не должно превышать шаг сетки h . Информация распространяется по сетке со скоростью h/τ (сеточная скорость). Эта скорость должна быть больше физической скорости (в данном случае – скорости переноса).

са). Сформулированный таким образом критерий (14.3) иногда называют критерием Куранта, а число $r = c\tau/h$ — числом Куранта.

Схема (14.2) устойчива для всех $c > 0$. Следовательно, данная схема безусловно устойчивая. Безусловная устойчивость — очень полезное качество разностной схемы, так как позволяет независимо подбирать временной и пространственный шаги сетки, руководствуясь лишь требуемой погрешностью решения.

Используя безусловно устойчивую разностную схему (14.2) найти решение уравнения переноса с заданными ниже краевыми и начальными условиями.

ВАРИАНТЫ ЗАДАНИЙ

ВАРИАНТЫ НАЧАЛЬНЫХ УСЛОВИЙ

$$(0 \leq \alpha < 1, \alpha < \beta \leq l; l, q, s, n = 1 \div 3)$$

$$1) u^0(x) = 0;$$

$$2) u^0(x) = \begin{cases} 1 & (0 \leq \alpha \leq x \leq \beta \leq l, \beta - \alpha < l), \\ 0 & (0 \leq x < \alpha, \beta < x \leq l); \end{cases}$$

$$3) u^0(x) = \begin{cases} (x - \alpha) / (\gamma - \alpha) & (\alpha \leq x \leq \gamma), \\ (\beta - x) / (\beta - \gamma) & (\gamma \leq x \leq \beta), \\ 0 & (0 \leq x < \alpha, \beta < x \leq l); \end{cases}$$

$$4) u^0(x) = \left| \sin \left(\pi n \left(\frac{x}{l} \right)^4 \right) \right|^3;$$

$$5) u^0(x) = \left| \cos \left(\pi n \left(\frac{x}{l} \right)^4 \right) \right|^3;$$

$$6) u^0(x) = \begin{cases} \left(\frac{x - \alpha}{l} \right)^4 \left(\frac{\beta - x}{l} \right)^3 & (\alpha \leq x \leq \beta), \\ 0 & (0 \leq x < \alpha, \beta < x \leq l). \end{cases}$$

ВАРИАНТЫ КРАЕВОГО УСЛОВИЯ $\varphi(t)$

$$1) \varphi(t) = 0;$$

$$2) \varphi(t) = \begin{cases} 1 & (0 \leq t \leq t_u), t_u = 0.1 \div 0.5 \frac{l}{c}, \\ 0 & (t > t_u); \end{cases}$$

$$3) \varphi(t) = e^{-pt} \quad (p = 0.1 \div 2);$$

$$4) \varphi(t) = e^{-pt^2} \quad (p = 0.1 \div 1);$$

$$5) \varphi(t) = te^{-pt} \quad (p = 0.1 \div 2);$$

$$6) \varphi(t) = te^{-pt^2} \quad (p = 0.1 \div 1).$$

ВАРИАНТЫ ФУНКЦИЙ $f(x,t)$ ($a, b = 0.1 \div 2$)

$$1) f(x,t) = \begin{cases} 1 & (0 \leq \beta_1 \leq x \leq \beta_2 \leq l), \\ 0 & (0 \leq x < \beta_1, \beta_2 < x \leq l); \end{cases}$$

$$2) f(x,t) = \begin{cases} 1 - e^{-at} & (0 \leq \beta_1 \leq x \leq \beta_2 \leq l), \\ 0 & (0 \leq x < \beta_1, \beta_2 < x \leq l); \end{cases}$$

$$3) f(x,t) = \begin{cases} e^{-at} & (0 \leq \beta_1 \leq x \leq \beta_2 \leq l), \\ 0 & (0 \leq x < \beta_1, \beta_2 < x \leq l); \end{cases}$$

$$4) f(x,t) = \frac{x}{l} \left(1 - \frac{x}{l} \right) e^{-at};$$

$$5) f(x,t) = e^{-a|x-bct|/l};$$

$$6) f(x,t) = \frac{1}{l} |x - bct| e^{-a(x-bct)^2/l^2};$$

$$7) f(x,t) \equiv 0.$$

ПРОГРАММИРОВАНИЕ

Для хранения текущих значений u_j достаточно отвести один одномерный массив размера $N + 1$ ($N = 40 \div 100$), записывая новые значения \hat{u}_j на место предыдущих. Временной шаг τ , длительность моделирования T и число временных шагов N_t рекомендуется определить из условий

$$c\tau = (0.5 \div 0.9)h, \quad T = (5 \div 10)l / c, \quad N_t = T / \tau.$$

Выдачи результатов u_j ($j=0, 1, \dots, N$) следует организовать через каждые $M = 10 \div 50$ временных шагов, включая в выдачу также номер временного шага N_t и текущее время $t = N_t \tau$. Соответственно, выдача производится, если N_t делится нацело на M .

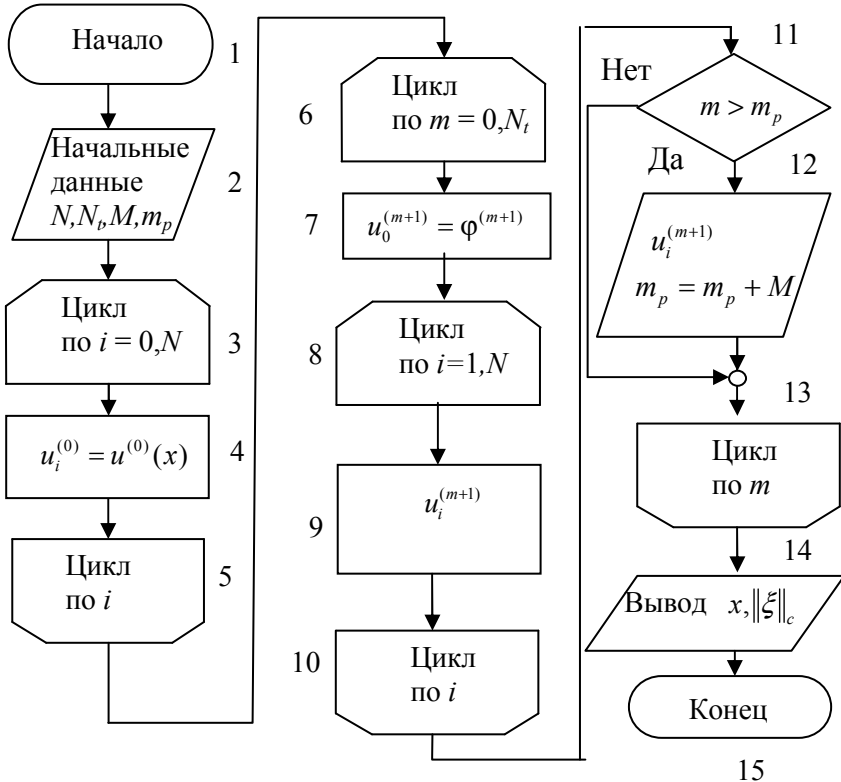


Рис. 14.2. Блок-схема программы решения уравнения переноса методом бегущего счета

В заданиях функции $f(x, t)$ и $\varphi(t)$ выбраны так, что при $T \gg l/c$ устанавливается стационарное состояние с u_j , не зависящими от t . Для проверки программы можно предварительно решить тестовую

задачу, полагая $f(x, t) \equiv 0$, $u^0(x) \equiv 0$, $\varphi(t) \equiv 1$, и исследуя установление очевидного решения

$$u(x, t) \rightarrow 1.$$

Блок-схема программы приведена на рис. 14.2. В цикле 3-4-5 задаются начальные, в 6-7 граничные условия, а в блоках 8-9-10 вычисляются \hat{u} .

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы и параметры для конкретного варианта;
- текст программы;
- результаты решения и график распределения $u(x, t)$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какова погрешность разностной схемы (14.2), какими свойствами она обладает?
2. Нарисуйте расположение сеточных узлов («шаблонов»), на котором построена схема (14.2).
3. Как надо изменить схему (14.2) и шаблон для расчета течений со скоростью $c < 0$?
4. Как можно построить схему бегущего счета второго порядка?
5. Какие разностные схемы называются монотонными? Обладает ли схема (14.2) монотонностью?

Задание № 15

ЧИСЛЕННОЕ РЕШЕНИЕ ОДНОМЕРНОГО УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ

Цель работы — изучение разностных схем для численного решения уравнения теплопроводности, численное решение одномерного уравнения теплопроводности по схеме «с полусуммой».

ПОСТАНОВКА ЗАДАЧИ

Распространение тепла в одномерной области описывается уравнением эволюционного типа

$$\frac{du}{dt} = a^2 \frac{d^2u}{dx^2} + f(x, t) \quad (0 \leq x \leq l, 0 \leq t \leq T) \quad (15.1)$$

с начальным условием

$$u(x, 0) = u^0(x) \quad (0 < x < l)$$

и краевыми условиями

$$u(0, t) = \varphi_1(t), \quad u(l, t) = \varphi_2(t) \quad (0 \leq t \leq T).$$

Здесь u — температура, x — координата, t — время, a^2 — коэффициент теплопроводности; функция $f(x, t)$ описывает источники тепла.

Уравнением описывается также процесс диффузии газа; при этом u — плотность газа, $a^2 = D$ — коэффициент диффузии, $f(x, t)$ — внутренние источники газа (выделение и поглощение).

Заметим, что для существования и единственности решения уравнения не требуется согласования начального и краевых условий, т. е. не требуется выполнения равенств $\varphi_1(0) = u^0(0)$, $\varphi_2(0) = u^0(l)$, иначе говоря, допускаются разрывы $u(x, 0)$ на границах в начальный момент $t = 0$.

Заменяя производные конечными разностями, построим следующие разностные схемы:

$$\frac{\hat{u}_j - u_j}{\tau} = a^2 \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + f_j^+, \quad (15.2)$$

$$\frac{\hat{u}_j - u_j}{\tau} = a^2 \frac{\hat{u}_{j+1} - 2\hat{u}_j + \hat{u}_{j-1}}{h^2} + f_j^+, \quad (15.3)$$

$$\frac{\hat{u}_j - u_j}{\tau} = \frac{a^2}{2} \left(\frac{\hat{u}_{j+1} - 2\hat{u}_j + \hat{u}_{j-1}}{h^2} + \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} \right) + f_j^+, \quad (15.4)$$

где

$$u_j = u(x_j, t), \hat{u}_j = u(x_j, t + \tau), f_j^+ = f(x_j, t + \frac{\tau}{2}),$$

$$x_j = jh \quad (j=0, 1, \dots, N), \quad h = l / N,$$

$$u(x_j, 0) = u_j^0 \quad (j=1, 2, \dots, N-1), \quad u_0 = \varphi_1, \quad u_N = \varphi_2.$$

Соответствующие им шаблоны приведены на рис. 15.1.

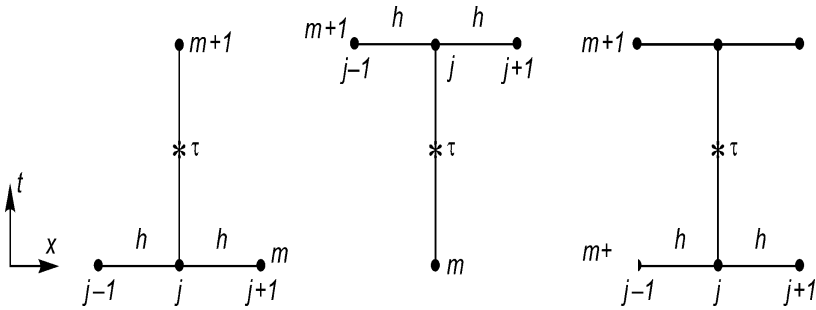


Рис. 15.1. Расположение узлов разностных схем для одномерного уравнения теплопроводности

Первая схема (15.2) является явной, имеет погрешность аппроксимации $O(\tau, h^2)$ и устойчива при выполнении условия $0 \leq \tau \leq 0.5h^2/a^2$. Вторая схема (15.3) является неявной, имеет такую же погрешность аппроксимации и безусловно устойчива. Если для увеличения пространственного разрешения потребуется в первой схеме уменьшить шаг h на порядок, то шаг τ для сохранения устойчивости придется уменьшить на два порядка, а число арифметиче-

ских операций и длительность счета, составляющие $O(1/(\tau h)) = O(h^{-3}) = O(N^3)$, возрастут на три порядка. Поэтому, несмотря на простоту вычислений, явными схемами стараются не пользоваться в задачах, в которых требуется высокое пространственное разрешение, т. е. малый шаг h . Неявная разностная схема (15.3) лишена этого недостатка.

В обеих разностных схемах — низкий, т. е. первый порядок погрешности по τ . Рассмотрим теперь схему (15.4), построенную на третьем шаблоне (см. рис. 15.1)), которую называют разностной схемой «с полусуммой», или схемой Кранка–Николсона. Можно показать [3], что данная разностная схема имеет более высокий порядок погрешности аппроксимации $O(\tau^2, h^2)$ и безусловно устойчива.

Преобразуем её к следующему виду:

$$\hat{u}_{j+1} - \sigma \hat{u}_j + \hat{u}_{j-1} = -\nu f_j^+ - (u_{j+1} - \theta u_j + u_{j-1});$$

$$\sigma = 2 + \mu, \quad \mu = \frac{2h^2}{\tau a^2}, \quad \nu = \frac{2h^2}{a^2}, \quad \theta = 2 - \mu;$$

т.е.

$$\hat{u}_{j+1} - \sigma \hat{u}_j + \hat{u}_{j-1} = g_j \quad (j=1, 2, \dots, N-1);$$

$$g_j = -\nu f_j^+ - (u_{j+1} - \theta u_j + u_{j-1}).$$

Решение на каждом временном шаге находится методом прогонки (смотри задание № 13). Результаты u_j ($j = 0, 1, \dots, N$) выдаются через каждые $M = 10 \div 50$ временных шагов. Прогонка строится в два этапа.

Первый этап — вычисление прогоночных коэффициентов по рекуррентным формулам (прямой ход)

$$\xi_{j+1} = 1/(\sigma - \xi_j), \quad \eta_{j+1} = (\eta_j - g_j)\xi_{j+1} \quad (j=1, 2, \dots, N-1),$$

$$\xi_1 = 0, \quad \eta_1 = \hat{\varphi}_1.$$

Второй этап — вычисление \hat{u} , (обратный ход)

$$\hat{u}_j = \xi_{j+1}\hat{u}_{j+1} + \eta_{j+1}; \quad (j = N-1, N-2, \dots, 1); \quad \hat{u}_N = \hat{\varphi}_2.$$

ВАРИАНТЫ ЗАДАНИЙ

ВАРИАНТЫ НАЧАЛЬНЫХ УСЛОВИЙ

$$1) u^0(x) = \begin{cases} c & (0 \leq \alpha_1 \leq x \leq \alpha_2 \leq l), \\ 0 & (0 \leq x \leq \alpha_1, \alpha_2 < x \leq l); \end{cases}$$

$$2) u^0(x) = cx/l;$$

$$3) u^0(x) = c(l-x)/l;$$

$$4) u^0(x) = c \left(\frac{x}{l} \right)^q \left(1 - \frac{x}{l} \right)^s \quad (q, s = 1 \div 3);$$

$$5) u^0(x) = \begin{cases} cx/\alpha & (0 \leq x \leq \alpha), \\ c(l-x)/(l-\alpha) & (\alpha \leq x \leq l); \end{cases}$$

$$6) u^0(x) = \left| \sin(\pi n(x/l^q)) \right|^5 \quad (n, q, s = 1 \div 3);$$

$$7) u^0(x) = \left| \cos(\pi n(x/l^q)) \right|^5 \quad (n, q, s = 1 \div 3).$$

ВАРИАНТЫ КРАЕВЫХ УСЛОВИЙ

$$(b=0.1 \div 2)$$

$$1) \varphi_1(t) = p_1, \quad \varphi_2(t) = p_2;$$

$$2) \varphi_1(t) = p_1, \quad \varphi_2(t) = p_2(1 - e^{-bt});$$

$$3) \varphi_1(t) = p_1(1 - e^{-bt}), \quad \varphi_2(t) = p_2;$$

$$4) \varphi_1(t) = p_1, \quad \varphi_2(t) = p_2 e^{-bt};$$

$$5) \varphi_1(t) = p_1 e^{-bt}; \quad \varphi_2(t) = p_2;$$

$$6) \varphi_1(t) = p_1 e^{-bt}; \quad \varphi_2(t) = p_2(1 - e^{-bt});$$

$$7) \varphi_1(t) = p_1(1 - e^{-bt}), \quad \varphi_2(t) = p_2 e^{-bt}.$$

ВАРИАНТЫ ФУНКЦИЙ ИСТОЧНИКОВ ТЕПЛА

$$(x=0.1 \div 2)$$

$$1) f(x, t) = \begin{cases} f_0 & (0 \leq \beta_1 \leq x \leq \beta_2 \leq l), \\ 0 & (0 \leq x < \beta_1, \beta_2 < x \leq l); \end{cases}$$

$$2) f(x, t) = \begin{cases} f_0(1 - e^{-xt}) & (0 \leq \beta_1 \leq x \leq \beta_2 \leq l), \\ 0 & (0 \leq x < \beta_1, \beta_2 < x \leq l); \end{cases}$$

$$3) f(x, t) = f_0 \frac{x}{l} \left(1 - \frac{x}{l}\right);$$

$$4) f(x, t) = f_0 \frac{x}{l} \left(1 - \frac{x}{l}\right) e^{-xt};$$

$$5) f(x, t) = \begin{cases} f_0 e^{-xt} & (0 \leq \beta_1 \leq x \leq \beta_2 \leq l), \\ 0 & (0 \leq x < \beta_1, \beta_2 < x \leq l). \end{cases}$$

ПРОГРАММИРОВАНИЕ

Функции f , φ_1 , φ_2 целесообразно оформить в виде подпрограмм-функций. Для хранения u_j и \hat{u}_j , следует отвести два одномерных массива. Шаг τ можно оценить из условия $\tau a^2/h = 0.2 \div 1$, число пространственных шагов N – несколько десятков, число временных шагов – несколько сотен. Особенности прогонки обсуждались в задании № 13, организация вычислений – в предыдущем задании.

Блок-схема программы представлена на рис. 15.2. В блоке 4 задаются граничные условия, в цикле 5-6-7 вычисляются правые части уравнения, в 12-17 организован метод прогонки.

Для проверки программы можно предварительно решить тестовую задачу, полагая

$$a^2 = 1, \quad u^0(x) = x/l, \quad \varphi_1(t) \equiv 1,$$

$$\varphi_2(t) \equiv 0, \quad f(x, t) \equiv 0$$

и исследуя установление стационарного состояния

$$u(x, t) = 1 - x/l.$$

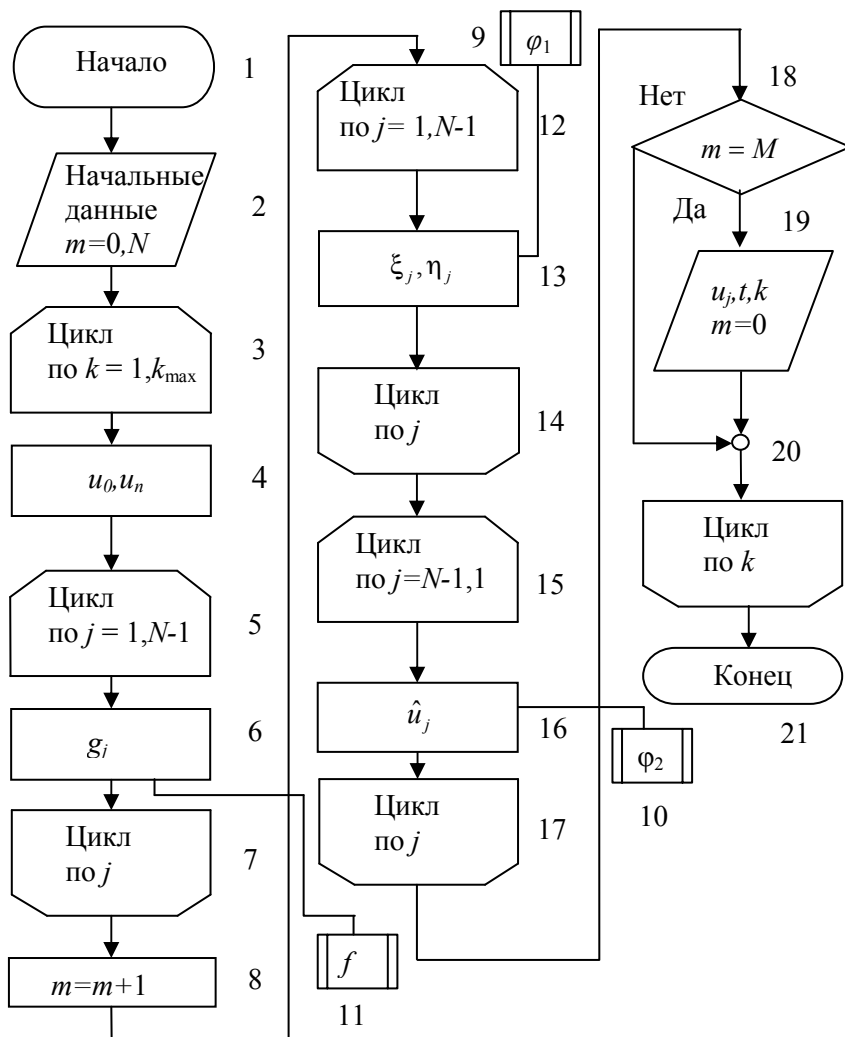


Рис. 15.2. Блок-схема программы решения одномерного уравнения теплопроводности

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы и параметры для конкретного варианта;
- текст программы;
- результаты решения.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков порядок погрешности приведенных разностных схем?
2. Покажите, что неявные разностные схемы (15.3), (15.4) безусловно устойчивы.
3. Как строится решение разностной схемы (15.4)?
4. Изобразите расположение узлов («шаблон»), на котором построена разностная схема (15.4).

Задание №16

ЧИСЛЕННОЕ РЕШЕНИЕ УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ В ПРЯМОУГОЛЬНИКЕ

Цель работы: изучение разностных схем для численного решения двумерного уравнения теплопроводности, численное решение уравнения теплопроводности в прямоугольнике по явной схеме.

ПОСТАНОВКА ЗАДАЧИ

Распространение тепла в прямоугольнике описывается уравнением теплопроводности

$$\frac{\partial u}{\partial t} = a^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f \quad (16.1)$$

$$(0 \leq x \leq l_x, \quad 0 \leq y \leq l_y, \quad 0 \leq t \leq T)$$

с начальным условием

$$u(x, y, 0) = u^0(x, y), \quad (0 \leq x \leq l_x, \quad 0 \leq y \leq l_y) \quad (16.2)$$

и краевыми условиями

$$\begin{aligned} u(0, y, t) = \psi_1(y, t), \quad u(l_x, y, t) = \psi_2(y, t), \\ u(x, 0, t) = \psi_3(x, t), \quad u(x, l_y, t) = \psi_4(x, t), \quad (0 \leq t \leq T). \end{aligned} \quad (16.3)$$

Здесь u — температура, x, y — координаты, t — время, a^2 — коэффициент теплопроводности, функция $f(x, y, t)$ описывает внутренние источники тепла.

Для решения воспользуемся явной двухслойной разностной схемой

$$\frac{\hat{u}_{ji} - u_{ji}}{\tau} = a^2 \left(\frac{u_{j+1,i} - 2u_{ji} + u_{j-1,i}}{h_x^2} + \frac{u_{j,i+1} - 2u_{ji} + u_{j,i-1}}{h_y^2} \right) + f_{ji}^+, \quad (16.4)$$

где

$$\begin{aligned}
 u_{ji} &= u(x_j, y_i, t), \widehat{u}_{ji} = u(x_j, y_i, t + \tau), \\
 f_{ji}^+ &= f(x_j, y_i, t + \tau / 2), \\
 x_j &= jh_x \quad (j = 0, 1, \dots, N), h_x = l_x / N, \\
 y_j &= jh_y \quad (i = 0, 1, \dots, M), h_y = l_y / M, \\
 u(x_j, y_i, 0) &= u^0(x_j, y_i), \quad (j = 1, \dots, N - 1, i = 1, \dots, M - 1), \\
 u_{0i} &= \Psi_1(y_i, t), u_{Ni} = \Psi_2(y_i, t), \quad (i = 1, \dots, M - 1), \\
 u_{j0} &= \Psi_3(x_j, t), u_{jM} = \Psi_4(x_j, t), \quad (j = 1, \dots, N - 1).
 \end{aligned}$$

Разностная схема преобразуется в расчётную формулу

$$\begin{aligned}
 \widehat{u}_{ji} &= u_{ji} + \tau a^2 \left(\frac{u_{j+1,i} - 2u_{ji} + u_{j-1,i}}{h_x^2} + \frac{u_{j,i+1} - 2u_{ji} + u_{j,i-1}}{h_y^2} \right) + \tau f_{ji}^+, \\
 & \quad (j = 1, 2, \dots, N - 1, i = 1, 2, \dots, M - 1),
 \end{aligned} \tag{16.5}$$

которой находится решение на следующем временном шаге в момент $t + \tau$.

Схема (16.4) устойчива при условии

$$\tau < \tau_{\max} = \frac{1}{2a^2(h_x^{-2} + h_y^{-2})}. \tag{16.6}$$

В частном случае, когда $h_x = h_y = h$, условие устойчивости принимает вид

$$\tau < \tau_{\max} = \frac{h^2}{4a^2}.$$

Для дальнейшего введём обозначение ступенчатой функции

$$\eta(z) = \begin{cases} 1 & (z \geq 0), \\ 0 & (z < 0). \end{cases}$$

ВАРИАНТЫ ЗАДАНИЙ

$$0 < \alpha_i < \beta_i < l_x, \quad 0 < \mu_i < \nu_i < l_y \quad (i = 0 \div 4);$$

$$0 < p_i, q_i, r_i, s_i \leq 3 \quad (i = 0 \div 4);$$

$$\gamma_i = 0.1 \div 10 \quad (i = 1 \div 5), \quad c_i = 0.1 \div 2 \quad (i = 1 \div 5).$$

ВАРИАНТЫ НАЧАЛЬНЫХ УСЛОВИЙ

$$1) u^0(x, y) = 1;$$

$$2) u^0(x, y) = c_0 \eta(x - \alpha_0) \eta(\beta_0 - x) \eta(y - \mu_0) \eta(\nu_0 - y);$$

$$3) u^0(x, y) = c_0 \left(\frac{x}{l_x} \right)^{p_0} \left(1 - \frac{x}{l_x} \right)^{q_0} \left(\frac{y}{l_y} \right)^{r_0} \left(1 - \frac{y}{l_y} \right)^{s_0};$$

$$4) u^0(x, y) = c_0 \left(\frac{x}{l_x} \right)^{p_0} \left(1 - \frac{x}{l_x} \right)^{q_0} \sin^{r_0} \left[\pi \left(\frac{y}{l_y} \right)^{s_0} \right];$$

$$5) u^0(x, y) = c_0 \left(\frac{x}{l_x} \right)^{p_0} \left(1 - \frac{y}{l_y} \right)^{q_0} \sin^{r_0} \left[\pi \left(\frac{x}{l_x} \right)^{s_0} \right].$$

ВАРИАНТЫ КРАЕВЫХ УСЛОВИЙ

$$(i=1,2)$$

$$1) \psi_i(y, t) = \eta(y - \mu_i) \eta(\nu_i - y) e^{-\gamma_i t};$$

$$2) \psi_i(y, t) = c_i \eta(y - \mu_i) \eta(\nu_i - y) (1 - e^{-\gamma_i t});$$

$$3) \psi_i(y, t) = c_i \left(\frac{y}{l_y} \right)^{p_i} \left(1 - \frac{y}{l_y} \right)^{q_i} (1 - e^{-\gamma_i t});$$

$$4) \psi_i(y, t) = c_i \left(\frac{y}{l_y} \right)^{p_i} \left(1 - \frac{y}{l_y} \right)^{q_i} e^{-\gamma_i t};$$

$$5) \psi_i(y, t) = c_i \sin^{r_i} \left[\pi \left(\frac{y}{l_y} \right)^{s_i} \right] e^{-\gamma_i t};$$

$$6) \psi_i(y, t) = c_i \sin^{r_i} \left[\pi \left(\frac{y}{l_y} \right)^{s_i} \right] (1 - e^{-\gamma_i t}).$$

ВАРИАНТЫ КРАЕВЫХ УСЛОВИЙ

($i=3,4$)

$$1) \psi_i(x, t) = c_i \eta(x - \alpha_i) \eta(\beta_i - x) e^{-\gamma_i t};$$

$$2) \psi_i(x, t) = c_i \eta(x - \alpha_i) \eta(\beta_i - x) (1 - e^{-\gamma_i t});$$

$$3) \psi_i(x, t) = c_i \left(\frac{x}{l_x} \right)^{p_i} \left(1 - \frac{x}{l_x} \right)^{q_i} e^{-\gamma_i t};$$

$$4) \psi_i(x, t) = c_i \left(\frac{x}{l_x} \right)^{p_i} \left(1 - \frac{x}{l_x} \right)^{q_i} (1 - e^{-\gamma_i t});$$

$$5) \psi_i(x, t) = c_i \sin^{r_i} \left[\pi \left(\frac{x}{l_x} \right)^{s_i} \right] e^{-\gamma_i t};$$

$$6) \psi_i(x, t) = c_i \sin^{r_i} \left[\pi \left(\frac{x}{l_x} \right)^{s_i} \right] (1 - e^{-\gamma_i t}).$$

ВАРИАНТЫ ИСТОЧНИКОВ ТЕПЛА

$$1) f(x, y, t) = c_5 u^0(x, y) e^{-\gamma_5 t},$$

$$2) f(x, y, t) = c_5 u^0(x, y) (1 - e^{-\gamma_5 t}),$$

где $u^0(x, y)$ – одно из возможных начальных условий, указанных выше.

ПРОГРАММИРОВАНИЕ

Функции f , u^0 , ψ целесообразно оформить в виде подпрограмм – функций. Для хранения u_{ji} и \hat{u}_{ji} можно отвести два двумерных массива размерности $(N+1)(M+1)$, хотя при более искусственном программировании достаточно одного двумерного массива

размерности $(N+1)(M+1)$ и одного - двух одномерных рабочих массивов размерности $N+1$. Число пространственных шагов N, M должно составлять несколько десятков, временной шаг τ следует выбирать равным $(0.2 \div 0.8)\tau_{\max}$, где τ_{\max} дается формулой (16.6). Число временных шагов N_t следует выбирать таким, чтобы процесс успел установиться, полагая, например,

$$\gamma_3 N_t \tau \approx 10 \div 100.$$

Для наблюдения за процессом целесообразно через несколько десятков шагов τ выводить на экран распределение температуры $u(x, t)$ в исследуемой области. Как альтернативу можно вывести распределение $u(x, y, t)$ вдоль средней линии $y = l_y/2$. Для этого надо описать вспомогательный одномерный массив v , имеющий $N+1$ элементов, перед выдачей пересылать в него значения $u(j, M/2)$ ($j = 0, 1, \dots, N$).

Для проверки программы рекомендуется предварительно решить тестовую задачу, полагая

$$\begin{aligned} a^2 = 1, l_x = l_y = 1, \psi_1 = \psi_2 = \psi_3 = \psi_4 = 0; \\ u^0(x, y) = 1, f(x, y, t) = 32[x(1-x) + y(1-y)]. \end{aligned}$$

При этих данных установившееся распределение температуры (при больших t) имеет вид

$$u(x, y, t) = 16x(1-x)y(1-y),$$

а вдоль средней линии $y = 0.5$ решение имеет вид параболы

$$u(x, 0.5, t) = 4x(1-x)$$

с вершиной $u=1$ в точке $x = 0.5$.

Блок-схема программы представлена на рис. 16.1.

В блоке 2 вводятся параметры $\alpha_i, \beta_i, \mu_i, \nu_i, p_i, q_i, r_i, s_i, \gamma_i, c_i$. В цикле 3-4-5 задаётся начальное распределение потенциала в области. Цикл 6-13 является основным в программе, в нём с шагом τ осуществляется изменение времени t . В этот цикл вложены циклы 7-8-9 и 10-11-12, в которых соответственно задаются граничные условия и вычисляется распределение потенциала внутри области в текущий момент времени. В блоке 14 выводятся полученные результаты.

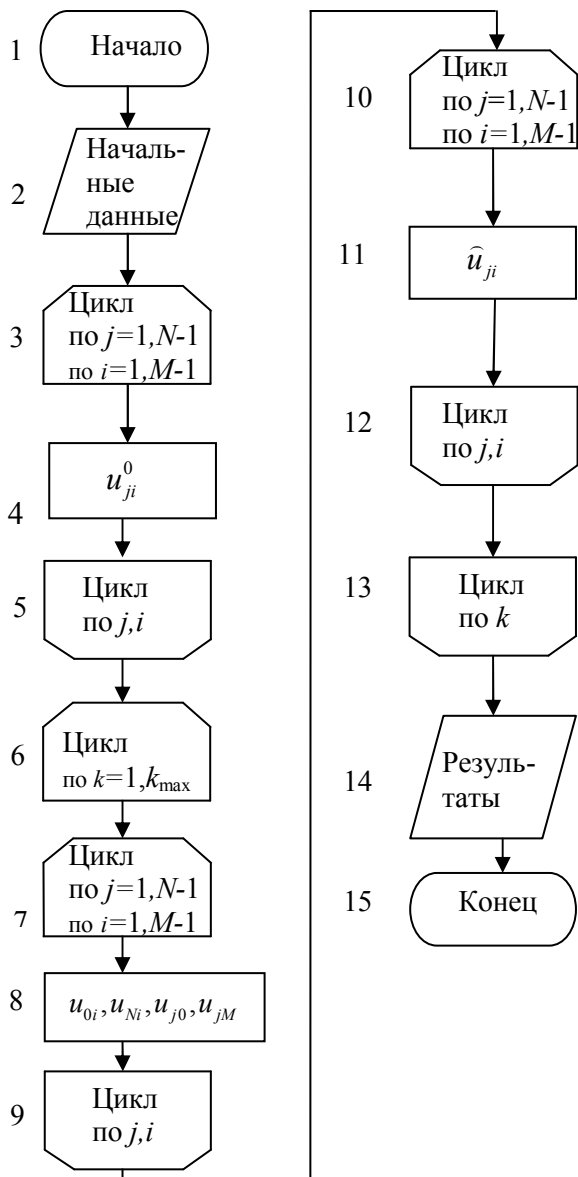


Рис. 16.1. Блок-схема программы решения уравнения теплопроводности в прямоугольнике

СОДЕРЖАНИЕ ОТЧЁТА

Отчет должен содержать:

- формулы и параметры для конкретного варианта;
- текст программы;
- результаты решения и графики.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков порядок погрешности разностной схемы (16.4)?
2. Получите условие устойчивости (16.6) используя спектральный критерий.
3. Как строится решение разностной схемы (16.4)?
4. Изобразите расположение узлов ("шаблон"), на котором построена разностная схема (16.4).
5. Каков порядок числа арифметических операций, требующихся для решения задачи с использованием расчетной формулы (16.5)?

Задание № 17

ЧИСЛЕННОЕ РЕШЕНИЕ ОДНОМЕРНОГО ВОЛНОВОГО УРАВНЕНИЯ

Цель работы: изучение разностных схем для численного решения уравнений в частных производных гиперболического типа, численное решение одномерного волнового уравнения.

ПОСТАНОВКА ЗАДАЧИ

Распространение волн в одномерном приближении описывается уравнением

$$\frac{d^2 u}{dt^2} = a^2 \frac{d^2 u}{dx^2} + f$$
$$(0 \leq x \leq l, 0 \leq t \leq T)$$

с начальными условиями

$$u(x, 0) = \mu_1(x) \quad (0 < x < l),$$
$$u'_t(x, 0) = \mu_2(x) \quad (0 < x < l),$$

и краевыми условиями

$$u(0, t) = \mu_3(t),$$
$$u(l, t) = \mu_4(t)$$
$$(0 \leq t \leq T).$$

Здесь x - координата, t - время, a - фазовая скорость волны; функция $f(x, t)$ описывает влияние среды, т.е. поглощение или возбуждение волн в среде.

Для решения воспользуемся неявной трехслойной разностной схемой

$$\frac{\hat{u}_j - 2u_j + \check{u}_j}{\tau^2} = \Lambda \left(\frac{\hat{u}_j + \check{u}_j}{2} \right) + f_j \quad (j = 1, 2, \dots, N-1). \quad (17.1)$$

Здесь

$$\begin{aligned} u_j &= u(x_j, t), \quad \widehat{u}_j = u(x_j, t + \tau), \quad \check{u}_j = u(x_j, t - \tau); \\ f_j &= f(x_j, t), \quad x_j = jh \quad (j = 0, 1, \dots, N), \quad h = l / N; \\ u_0 &= \mu_3, u_N = \mu_4, \end{aligned}$$

а оператор

$$\Lambda u_j = \frac{a^2}{h^2} (u_{j+1} - 2u_j + u_{j-1}).$$

Разностную схему (17.1) можно преобразовать к виду

$$\begin{aligned} \widehat{u}_{j+1} - \sigma \widehat{u}_j + \widehat{u}_{j-1} &= g_j \quad (j = 1, 2, \dots, N-1), \\ g_j &= -2bu_j - (\check{u}_{j+1} - \sigma \check{u}_j + \check{u}_{j-1}) - \frac{2h^2}{a^2} f_j, \\ \sigma &= 2 + b, \quad b = \frac{2h^2}{a^2 \tau^2}. \end{aligned}$$

Полученное уравнение решается методом прогонки, как и в выполненных ранее работах. Прогонка строится в два этапа.

Первый этап - вычисление прогоночных коэффициентов по рекуррентным формулам (прямой ход прогонки):

$$\begin{aligned} \xi_{j+1} &= 1 / (\sigma - \xi_j), \quad \eta_{j+1} = (\eta_j - g_j) \xi_{j+1} \quad (j = 1, 2, \dots, N-1), \\ \xi_1 &= 0, \quad \eta_1 = \mu_3. \end{aligned}$$

Второй этап - вычисление значений \widehat{u}_j , (обратный ход прогонки):

$$\begin{aligned} \widehat{u}_j &= \xi_{j+1} \widehat{u}_{j+1} + \eta_{j+1} \quad (j = N-1, N-2, \dots, 1), \\ \widehat{u}_N &= \mu_4. \end{aligned}$$

Вместо схемы (17.1) можно также воспользоваться более простой явной трехслойной разностной схемой

$$\begin{aligned} \frac{\widehat{u}_j - 2u_j + \check{u}_j}{\tau^2} &= a^2 \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + f_j, \quad (j = 1, 2, \dots, N-1), \\ u_0 &= \mu_3, \quad u_N = \mu_4, \end{aligned} \tag{17.2}$$

которая преобразуется в расчетную формулу

$$\begin{aligned}\hat{u}_j &= 2(1 - c^2)u_j - \tilde{u}_j + c^2(u_{j+1} + u_{j-1}) + \tau^2 f_j, \\ c &= (\tau a / h),\end{aligned}$$

где c - число Куранта.

Данная схема устойчива при выполнении условия Куранта:

$$|c| < 1.$$

Для начала вычислений необходимо, помимо $u(x, 0)$, располагать значением $u(x, \tau)$. Поскольку решение $u(x, t)$ на каждом временном шаге находится с погрешностью $O(\tau^3)$ значение $u(x, t)$ также необходимо иметь с той же погрешностью $O(\tau^3)$. Для нахождения воспользуемся рядом Тейлора:

$$u(x, \tau) = u(x, 0) + \left. \frac{du}{dt} \right|_{t=0} \tau + \left. \frac{d^2u}{dt^2} \right|_{t=0} \frac{\tau^2}{2} + O(\tau^3).$$

Из исходного уравнения следует, что

$$\left. \frac{d^2u}{dt^2} \right|_{t=0} = a^2 \left. \frac{d^2u}{dx^2} \right|_{t=0} + f(x, 0),$$

и из начальных условий

$$\left. \frac{du}{dt} \right|_{t=0} = \mu_2(x), u(x, 0) = \mu_1(x),$$

$$\left. \frac{d^2u}{dx^2} \right|_{t=0} = \frac{d^2\mu_1}{dx^2}.$$

Окончательно имеем:

$$u(x_j, \tau) = \mu_1(x_j) + \tau \mu_2(x_j) + \frac{\tau^2}{2} \left[a^2 \left. \frac{d^2\mu_1(x)}{dx^2} \right|_{x=x_j} + f(x_j, 0) \right]$$

$$(j = 1, 2, \dots, N-1), \quad u(0, \tau) = \mu_3(\tau), \quad u(l, \tau) = \mu_4(\tau).$$

Затем из уравнения (17.1) находим значения $u(x, t)$ в моменты времени $t = 2\tau, 3\tau$ и т.д.

В дальнейшем ограничимся случаем, когда $f = 0$, т.е. среда (внутренние источники) отсутствует. Таким образом, будут рассматриваться лишь свободные колебания и вынужденные колебания под действием краевой возбуждающей силы. Характерными временами задачи являются:

$$T_0 = \frac{l}{a}, \quad \tau_0 = \frac{h}{a},$$

где T_0 - время прохождения волны вдоль системы, τ_0 - время прохождения волной шага h . Можно показать, что диапазоны длин волн λ и частот ν , которые могут наблюдаться в рассматриваемой системе, составляют:

$$2h \leq \lambda \leq 2l, \quad 0.5 / T_0 \leq \nu \leq 0.5 / \tau_0.$$

ВАРИАНТЫ ЗАДАНИЙ

$$c_i = 1 \div 10 (i = 1 \div 4).$$

ВАРИАНТЫ ФУНКЦИЙ $\mu_1(x)$

- 1) $\mu_1 = c_1(x/l)(1-x/l)$;
- 2) $\mu_1 = c_1 \sin(\pi nx/l) \quad (n = 1 \div 3)$;
- 3) $\mu_1 = 0$;
- 4) $\mu_1 = c_1(x/l)^2(1-x/l)$;
- 5) $\mu_1 = c_1 \cos(\pi nx/l) \quad (n = 1 \div 3)$.

ВАРИАНТЫ ФУНКЦИЙ $\mu_2(x)$

- 1) $\mu_2 = 0$;
- 2) $\mu_2 = \begin{cases} c_2(x \in [\alpha, \beta]), (0 < \alpha < \beta < l), \\ 0(x \notin [\alpha, \beta]); \end{cases}$
- 3) $\mu_2 = c_2(x/l)(1-x/l)$;
- 4) $\mu_2 = c_2 \sin(\pi mx/l) \quad (m = 1 \div 3)$;
- 5) $\mu_2 = \begin{cases} c_2 x/l & (0 < x \leq l/2), \\ c_2(1-x/l) & (l/2 < x < l). \end{cases}$

ВАРИАНТЫ ФУНКЦИЙ $\mu_i(x)$ ($i = 3, 4$)

$$(p_i = 0.2 \div 2, \quad \nu_i = 1 \div 10).$$

- 1) $\mu_i = c_i e^{-p_i t/T_0}$;
- 2) $\mu_i = c_i(1 - e^{-p_i t/T_0})$;
- 3) $\mu_i = c_i e^{-p_i t/T_0} \cos \frac{2\pi t}{T_0} \nu_i$;
- 4) $\mu_i = c_i(1 - e^{-p_i t/T_0}) \cos \frac{2\pi t}{T_0} \nu_i$;

$$5) \mu_i = \begin{cases} c_i(t/v_i T_0)(1-t/v_i T_0) & (0 \leq t \leq v_i T_0), \\ 0 & (t > v_i T_0). \end{cases}$$

ПРОГРАММИРОВАНИЕ

Функции μ_i ($i = 1$ до 4) удобно оформить в виде подпрограмм - функций. Для хранения $\hat{u}_j, u_j, \check{u}_j$ целесообразно отвести три одномерных массива по $N+1$ элементов в каждом, и после завершения очередного временного шага с помощью оператора цикла переписывать u_j на место \check{u}_j , а \hat{u}_j на место u_j .

Значения l и a можно положить равными 1, а число пространственных шагов N , временной шаг τ и длительность счета T рекомендуется выбирать из условий:

$$N=25-100, \tau = (0.2-0.8) \tau_0, T = (5-10) T_0.$$

Для наблюдения за ходом вычислений через каждые $M=20 \div 100$ шагов следует выдавать график u_j .

Для проверки правильности программы рекомендуется предварительно решить тестовую задачу, полагая

$$\mu_1 = \mu_3 = \mu_4 = 0, \quad \mu_2(x) = \frac{\pi a}{l} \sin \frac{\pi x}{l}.$$

Решение этой задачи должно быть близко к стоячей волне вида

$$u(x, t) = \sin \frac{\pi x}{l} \sin \frac{\pi a t}{l}.$$

Блок-схема программы для явной схемы представлена на рис. 17.1. В блоке 2 вводятся параметры $\mu_i, v_i, p_i, c_i, k_{\max}, M$. В цикле 3-4-5 задаются начальные распределения в моменты времени 0 и τ . Цикл 6-15 является основным в программе, в нём с шагом τ осуществляется изменение времени t . В этот цикл вложены циклы 7-8-9 и 12-13-14, в которых соответственно вычисляются значения u_j в текущий момент времени и перезапись массивов для перехода к следующему временному шагу. В блоке 10 анализируется необходимость выдачи промежуточного распределения u_j , которое осуществляется в блоке 11, если это необходимо. В блоке 16 выводятся полученные результаты.

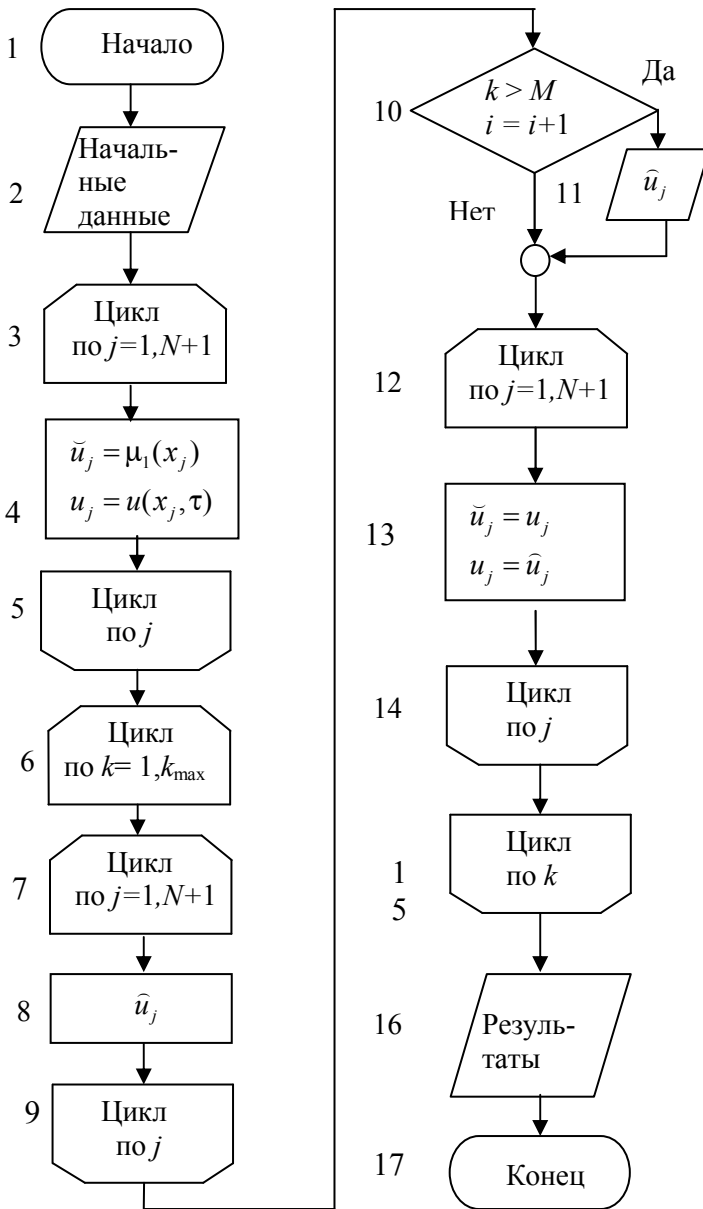


Рис. 17.1. Блок-схема программы решения волнового уравнения

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы и параметры для конкретного варианта;
- текст программы;
- результаты решения.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков порядок погрешности разностной схемы (17.1)?
2. Покажите, что разностная схема (17.1) безусловно устойчива.
3. Как строится решение разностной схемы (17.1)?
4. Изобразите расположение узлов ("шаблон"), на котором построена разностная схема (17.1)?
5. Каков порядок погрешности разностной схемы (17.2)?
6. Получите условие устойчивости разностной схемы (17.2).
7. Изобразите расположение узлов, на котором построена разностная схема (17.2).

Задание № 18

ЧИСЛЕННОЕ РЕШЕНИЕ УРАВНЕНИЯ ПУАССОНА В ПРЯМОУГОЛЬНИКЕ

Цель работы — изучение разностных схем для уравнений эллиптического типа, численное решение уравнения Пуассона в прямоугольнике методами Зейделя или последовательной верхней релаксации.

ПОСТАНОВКА ЗАДАЧИ

Типичным уравнением в частных производных эллиптического типа является уравнение вида

$$-\Delta u + \mu u = f.$$

При $\mu \neq 0$, $f \neq 0$ уравнение представляет собой неоднородное уравнение Гельмгольца, при $\mu \neq 0$, $f = 0$ — однородное уравнение Гельмгольца, при $\mu = 0$ и $f \neq 0$ — уравнение Пуассона

$$\Delta u = -f$$

и, наконец, при $\mu = 0$ и $f = 0$ — уравнение Лапласа.

Так, однородным уравнением Гельмгольца

$$\Delta \vec{E} + k^2 \vec{E} = 0$$

описывается стационарное распределение в резонаторе электрического поля \vec{E} с волновым числом k ; уравнение Пуассона

$$\Delta u = -\rho/\epsilon$$

описывает распределение электрического потенциала u в среде с плотностью заряда ρ и электрической проницаемостью ϵ и т. п.

Установившееся распределение температуры (или плотности газа) также описывается уравнением Пуассона, которое получается из уравнения теплопроводности (диффузии) при $\partial u/\partial t = 0$.

Поскольку уравнения эллиптического типа часто встречаются в важных физических приложениях, для их решения разработано много численных методов, которые можно разделить на конечно-разностные и проекционные [3].

В дальнейшем ограничимся так называемой задачей Дирихле, или первой краевой задачей для уравнения Пуассона

$$\Delta u(\vec{r}) = -f(\vec{r}) \quad (\vec{r} \in G), \quad u(\vec{r}) = \psi(\vec{r}) \quad (\vec{r} \in \Gamma)$$

в прямоугольнике $G = (0 \leq x \leq l_x, 0 \leq y \leq l_y)$ с границей Γ , используя равномерную сетку.

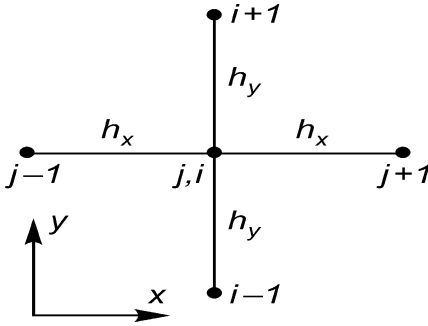


Рис. 18.1. Расположение узлов пространственной сетки для двумерного уравнения эллиптического типа

Вводя в прямоугольнике G равномерную сетку, имеющую N шагов по x и M шагов по y , аппроксимируем вторые производные в операторе $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ конечно-разностными формулами на пятиточечном шаблоне «крест» (рис. 18.1) и построим разностную схему

$$\frac{u_{j+1,i} - 2u_{j,i} + u_{j-1,i}}{h_x^2} + \frac{u_{j,i+1} - 2u_{j,i} + u_{j,i-1}}{h_y^2} = -f_{j,i};$$

$$u_{j,i} = u(x_j, y_i), \quad x_j = jh_x \quad (j = 0, 1, \dots, N), \quad h_x = l_x / N;$$

$$y_i = ih_y \quad (i = 0, 1, \dots, M), \quad h_y = l_y / M;$$

$$f_{j,i} = f(x_j, y_i) \quad (j = 1, 2, \dots, N-1; \quad i = 1, 2, \dots, M-1).$$

Приведём её к виду

$$u_{j,i} = d^{-1} \left[h_x^{-2} (u_{j+1,i} + u_{j-1,i}) + h_y^{-2} (u_{j,i+1} + u_{j,i-1}) + f_{j,i} \right],$$

$$d = 2(h_x^{-2} + h_y^{-2}).$$

Здесь h_x — шаг сетки по x , h_y — шаг по y , (x_j, y_i) — узлы сетки.

Схема имеет погрешность аппроксимации $O(h_x^2, h_y^2)$, т. е. это схема второго порядка; значения $u_{j,i}$ при $j = 0, N$ и $i = 0, M$ задаются краевыми условиями. Уравнения представляют собой систему

$(N - 1) (M - 1)$ линейных алгебраических уравнений, которую можно записать в виде

$$Au = b.$$

Здесь u — вектор значений u_{ji} во всех внутренних узлах сетки, имеющий $(N - 1) (M - 1)$ элементов; b — вектор правой части, включающий как значения f_{ji} во внутренних узлах сетки, так и краевые значения; A — симметричная редкая матрица размерности $(N - 1) (M - 1) \times (N - 1) (M - 1)$.

Для решения полученной системы можно использовать рассмотренный ранее в задании № 8 метод Зейделя или же применить метод последовательной верхней релаксации, ускорение сходимости с использованием Чебышевского набора итерационных параметров и др.[3].

Решим уравнение итерационным методом Зейделя:

$$\begin{cases} u_{ji}^{(k)} = d^{-1} \left[h_x^{-2} (u_{j+1,i}^{(k-1)} + u_{j-1,i}^{(k)}) + h_y^{-2} (u_{j,i+1}^{(k-1)} + u_{j,i-1}^{(k)}) + f_{ji} \right] \\ (j = 1, 2, \dots, N - 1; i = 1, 2, \dots, M - 1), \end{cases} \quad (18.1)$$

где $k=1, 2, \dots$ — номер итерации. Этот метод называют также схемой Либмана.

В качестве начального приближения можно принять любые значения, например:

$$u_{ji}^{(0)} = d^{-1} f_{ji},$$

или даже

$$u_{ji}^{(0)} \equiv 0.$$

Одним из более эффективных методов решения уравнения (18.1) является предложенный Янгом и Франкелом итерационный метод последовательной верхней релаксации. В этом методе каждая итерация в узлах (j, i) складывается из двух этапов.

Первый этап — вычисление предварительного значения методом Зейделя:

$$\begin{cases} \tilde{u}_{ji}^{(k)} = d^{-1} \left[h_x^{-2} (u_{j+1,i}^{(k-1)} + u_{j-1,i}^{(k)}) + h_y^{-2} (u_{j,i+1}^{(k-1)} + u_{j,i-1}^{(k)}) + f_{ji} \right] \\ (j = 1, 2, \dots, N - 1; i = 1, 2, \dots, M - 1), \end{cases}$$

где $k=1, 2, \dots$ — номер итерации.

Второй этап — вычисление окончательного значения в узле (j, i) по релаксационной формуле:

$$u_{ji}^{(k)} = \omega \tilde{u}_{ji}^{(k)} + (1 - \omega) u_{ji}^{(k-1)},$$

где $0 < \omega < 2$ — релаксационный параметр.

Объединяя эти два этапа получаем расчетную формулу:

$$u_{ji}^{(k)} = (1 - \omega) u_{ji}^{(k-1)} + \omega d^{-1} \left[h_x^{-2} (u_{j+1,i}^{(k-1)} + u_{j-1,i}^{(k-1)}) + h_y^{-2} (u_{j,i+1}^{(k-1)} + u_{j,i-1}^{(k-1)}) + f_{ji} \right]$$

$$(j = 1, 2, \dots, N - 1; i = 1, 2, \dots, M - 1).$$

Максимальная скорость сходимости итераций достигается при оптимальном значении релаксационного параметра

$$\omega = \omega_0 = 2 / (1 + \sqrt{1 - \lambda_{\max}^2}),$$

где λ_{\max} — модуль максимального собственного значения матрицы метода простой итерации в правой части исходного уравнения, т. е. λ — это решение проблемы собственных значений

$$d^{-1} \left[h_x^{-2} (u_{j+1,i} + u_{j-1,i}) + h_y^{-2} (u_{j,i+1} + u_{j,i-1}) \right] = \lambda u_{ji}.$$

Отыскивая решение уравнения в прямоугольнике G в виде

$$u_{ji} \sim \sin \frac{\pi n j}{N} \sin \frac{\pi m i}{M},$$

находим собственные значения

$$\lambda = \left[h_x^{-2} \cos(\pi n / N) + h_y^{-2} \cos(\pi m / M) \right] / (h_x^{-2} + h_y^{-2})$$

$$(n = 1, 2, \dots, N - 1; m = 1, 2, \dots, M - 1).$$

Максимальное значение λ достигается при $n = m = 1$:

$$\lambda_{\max} = \left[h_x^{-2} \cos(\pi / N) + h_y^{-2} \cos(\pi / M) \right] / (h_x^{-2} + h_y^{-2}).$$

В частности, в случае квадратной области G при $N = M$:

$$\lambda_{\max} = \cos(\pi / N), \quad \omega_0 = 2 / [1 + \sin(\pi / N)].$$

В качестве начального приближения можно по-прежнему принять приведенные ранее значения.

Итерации прекращаются при выполнении условия

$$\delta u^{(k)} = \left\| u^{(k)} - u^{(k-1)} \right\|_c = \max_{ij} \left| u_{ji}^{(k)} - u_{ji}^{(k-1)} \right| < \varepsilon, \quad (18.2)$$

где ε — заданная малая величина.

ВАРИАНТЫ ЗАДАНИЙ

ВАРИАНТЫ ПРАВОЙ ЧАСТИ $f(x,y)$ УРАВНЕНИЯ

$$1) f(x, y) = \begin{cases} c & (0 < \alpha_1 \leq x \leq \alpha_2 \leq l_x, 0 < b_1 \leq y \leq b_2 < l_y), \\ 0 & (0 < x < \alpha_1, \alpha_2 < x \leq l_x, 0 < y < b_1, b_2 < y < l_y); \end{cases}$$

$$2) f(x, y) = c(l_x - x)^p x^q y^r (l_y - y)^s \quad (p, q, r, s = 0.5 \div 2);$$

$$3) f(x, y) = c \sin^p \pi \left(\frac{x}{l_x} \right)^q \sin^r \pi \left(\frac{y}{l_y} \right)^s \quad (p, q, r, s = 0.5 \div 2).$$

ВАРИАНТЫ КРАЕВЫХ УСЛОВИЙ

$$1) \psi \equiv 0;$$

$$2) \psi \equiv 1;$$

$$3) \psi(x, y) = \begin{cases} 1 & (x = 0), \\ 0 & (x \neq 0); \end{cases}$$

$$4) \psi(x, y) = \begin{cases} 1 & (y = 0), \\ 0 & (y \neq 0); \end{cases}$$

$$5) \psi(x, y) = \begin{cases} 1 & (x = l_x), \\ 0 & (x \neq l_x); \end{cases}$$

$$6) \psi(x, y) = \begin{cases} 1 & (y = l_y), \\ 0 & (y \neq l_y); \end{cases}$$

$$7) \psi(x, y) = \begin{cases} 1 & (x = 0 \text{ è è } y = 0), \\ 0 & (x \neq 0 \text{ è } y \neq 0); \end{cases}$$

$$8) \psi(x, y) = \begin{cases} 1 & (x = l_x \text{ è è } y = l_y), \\ 0 & (x \neq l_x \text{ è } y \neq l_y). \end{cases}$$

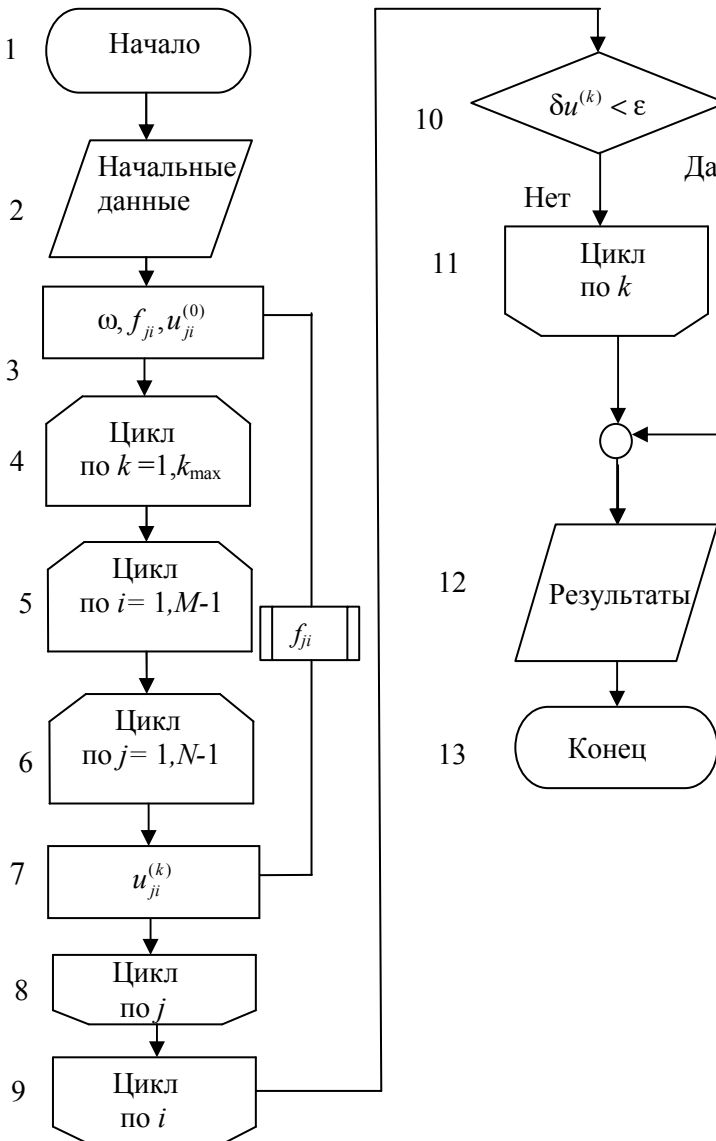


Рис. 18.2. Блок-схема программы решения уравнения Пуассона методом последовательной верхней релаксации

ПРОГРАММИРОВАНИЕ

Для хранения правой части f_{ji} и текущей итерации u_{ji}^k следует описать двумерные массивы. Правую часть удобно оформить в виде подпрограммы-функции. Норма (18.2) вычисляется на этапе релаксации перед тем, как новое значение u_{ji}^k посылается на место прежнего.

Значения N, M достаточно взять небольшими в диапазоне $10 \div 50$ для ускорения расчетов. Чтобы избежать заикливания из-за ошибок в программе, следует задать максимальное число итераций k_{\max} порядка нескольких сотен, по достижении которого итерации прерываются.

Блок-схема решения уравнения Пуассона методом последовательной верхней релаксации приведена на рис. 18.2.

Для проверки программы можно предварительно решить тестовую задачу при

$$f \equiv 0, \quad \psi \equiv 0, \quad u_{ji}^0 \equiv 1, \quad (j = 1, 2, \dots, N-1; i = 1, 2, \dots, M-1)$$

и исследовать сходимость итераций к тривиальному решению

$$u_{ij} \equiv 0 \quad (j = 0, 1, \dots, N; i = 0, 1, \dots, M).$$

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать:

- формулы и параметры для конкретного варианта;
- текст программы;
- результаты решения, число итераций и графики распределения потенциала в области.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какова погрешность конечно-разностного уравнения, построенного на шаблоне "крест"?
2. Как строится численное решение уравнения Пуассона методом Зейделя?

3. Как строится численное решение уравнения Пуассона методом последовательной верхней релаксации?
4. Получите формулу для собственного значения матрицы исходной системы уравнений.
5. Какие еще существуют методы решения уравнения Пуассона?
6. Как определить скорость сходимости итераций?
7. Что такое невязка и погрешность численного решения?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
2. Рашиков В.И., Рошаль А.С. Численные методы решения физических задач. - М.: МИФИ, 1992.
3. Рашиков В.И., Рошаль А.С. Численные методы решения физических задач. - СПб: Лань, 2005.
4. Рошаль А.С. Лабораторный практикум по курсу «Численные методы и программирование». – М: МИФИ, 1992.
5. Турчак Л.И. Основы численных методов. - М.: Наука, 1987, 2002.
6. Калиткин Н.Н. Численные методы. - М.: Наука, 1978.
7. Бахвалов Н.С., Жидков Н.П., Козельков Г.М. Численные методы. -М.: Наука, 1987, 2003.
8. Самарский А.А., Гулин А.В. Численные методы. - М.: Наука, 1989.

Владимир Иванович Ращиков

ЧИСЛЕННЫЕ МЕТОДЫ.
КОМПЬЮТЕРНЫЙ ПРАКТИКУМ

Редактор *Е.Н. Кочубей*

Подписано в печать 20.12.2009. Формат 60×84 1/16

Объем 8,25 п.л. Уч.-изд. л. 5,0. Тираж 500 экз.

Изд. № 1/3/24. Заказ № 18.

Национальный исследовательский ядерный университет «МИФИ».
115409, Москва, Каширское шоссе, 31.

ООО «Полиграфический комплекс «Курчатовский».
144000, Московская область, г. Электросталь, ул. Красная, д. 42