

Министерство общего и профессионального образования  
Российской Федерации  
Челябинский государственный университет

МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
к лабораторным работам по курсу  
**ЧИСЛЕННЫЕ МЕТОДЫ**

Челябинск 1997

Одобрено учебно-методической комиссией математического факультета

В методических указаниях изложен весь необходимый теоретический материал для выполнения лабораторных работ студентами 3 курса математического факультета специальности "Прикладная математика". Задания к лабораторным работам находятся в компьютерной сети математического факультета ЧелГУ:  $Y :> MATMODEL > PM\_3 > lab\_i.*$

Предназначены для студентов специальностей "Прикладная математика" и "Математика".

Составители : Н.Б.Медведева, К.А.Рязанов.

Рецензент : Т.Б.Бигильдеева.

# 1 Лабораторная работа.

## Элементы теории погрешностей.

Под погрешностью понимается некоторая величина, характеризующая точность результата. Существуют три вида погрешностей:

- 1) неустранимая погрешность, возникающая из-за неточности исходной информации, например неточности измерений;
- 2) погрешность метода;
- 3) погрешность вычислений, возникающая из-за округлений.

Основная задача теории погрешностей – указание области неопределенности результата.

### 1.1 Абсолютная и относительная погрешности.

*Абсолютной погрешностью* числа  $x$  называется величина  $\Delta_x$ , удовлетворяющая условию

$$|x - \tilde{x}| \leq \Delta_x, \quad (1.1)$$

где  $x$  – точное значение величины, а  $\tilde{x}$  – ее приближенное значение. В этом случае говорят, что  $\tilde{x}$  определено с точностью до  $\Delta_x$ , т.е.  $x = \tilde{x}(\pm\Delta_x)$ .

Отношение абсолютной погрешности к абсолютному значению приближенной величины

$$\frac{\Delta_x}{|\tilde{x}|} = \delta_x$$

называется *относительной погрешностью*  $\tilde{x}$ . Следовательно

$$\tilde{x}(1 - \delta_x) \leq x \leq \tilde{x}(1 + \delta_x).$$

Относительная погрешность чаще всего указывается в процентах. Точность результата лучше характеризует его относительная погрешность, которая показывает какую часть самого числа составляет погрешность. Абсолютные и относительные погрешности числа принято округлять только в большую сторону, так как при округлениях границы неопределенности числа, как правило, увеличиваются. По этой причине вычисления ведут с одним-двумя запасными знаками.

Рассмотрим как определяются верные значащие цифры чисел. *Значащими цифрами числа* называются все цифры в его записи, начиная с первой ненулевой слева. Значащая цифра называется *верной*, если абсолютная погрешность числа не превосходит половины единицы разряда, соответствующего этой цифре. В противном случае цифра называется *сомнительной*.

*Например*, пусть  $\tilde{x} = 12,396$  и известно, что  $\Delta_x = 0,03$ . Тогда число  $\tilde{x}$  имеет верные знаки 1, 2, 3, а 9 и 6 – сомнительные, т.к.  $0,005 < \Delta_x < 0,05$ .

Если число имеет лишь верные цифры, его округленное значение имеет также лишь верные цифры. Приближенное значение числа, имеющего все верные цифры, может не совпадать с его точным значением.

При вычислениях желательно сохранять такое количество значащих цифр, чтобы их число не превышало числа верных цифр более чем на две единицы.

## 1.2 Возникновение вычислительных погрешностей.

При решении задач на ЭВМ возникают так называемые *вычислительные погрешности*. Они обусловлены округлением чисел при записи и при выполнении арифметических операций.

Рассмотрим представление чисел в ЭВМ, которая работает, как правило, в двоичной системе, когда любое число записывается в виде последовательности нулей и единиц. Пусть  $x$  – действительное число. Его можно записать в двоичной системе с плавающей запятой:

$$\tilde{x} = \pm 2^p \sum_{k=1}^t \frac{d_k}{2^k}, \quad (1.2)$$

где параметр  $p$  определяет порядок числа, а параметр  $t$  – количество значащих цифр в числе. Порядок чисел, представимых в ЭВМ, ограничен сверху и снизу, т.е.

$$M_0 \leq |\tilde{x}| \leq M_\infty,$$

где  $M_0 = 2^{-t}$  – машинный ноль,  $M_\infty = 2^p$  – машинная бесконечность.

Все вещественные числа, которые могут быть представлены в данной ЭВМ, расположены по абсолютной величине в диапазоне от  $M_0$  до  $M_\infty$ . Если в процессе счета появится вещественное число, меньшее по модулю чем  $M_0$ , то

ему присваивается нулевое значение. При появлении в процессе счета вещественного числа, большего по модулю чем  $M_\infty$ , происходит так называемое переполнение разрядной сетки, после чего ЭВМ прекращает счет задачи.

Абсолютная погрешность записи числа с плавающей запятой (1.2) не больше единицы последнего разряда, т.е.

$$|x - \tilde{x}| \leq 2^{p-t} = \Delta_x.$$

Из этого соотношения получим относительную погрешность представления (1.2) чисел в ЭВМ:

$$\delta_x = \frac{|x - \tilde{x}|}{|x|} \leq \frac{\Delta_x}{|\tilde{x}|} = \frac{2^{p-t}}{2^p \sum_{k=1}^t \frac{d_k}{2^k}} \leq 2^{-t}.$$

То есть, предельная относительная погрешность представления действительных чисел в ЭВМ  $\delta_0 = 2^{-t}$  зависит только от значения  $t$  – числа разрядов, отведенных под мантиссу в записи числа. Чем больше  $t$ , тем точнее представление чисел.

При выполнении арифметических действий на ЭВМ одна арифметическая операция вносит относительную погрешность не более, чем  $2^{-t}$ .

### 1.3 Прямая задача теории погрешностей.

Пусть в некоторой области  $\Omega$   $n$ -мерного действительного пространства рассматривается непрерывно дифференцируемая функция

$$z = f(x_1, x_2, \dots, x_n).$$

Предположим, что в точке  $(x_1, x_2, \dots, x_n)$  области  $\Omega$  нужно вычислить значение  $z = f(x_1, x_2, \dots, x_n)$  и нам известны лишь приближенные значения  $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$  и их погрешности.

Если воспользоваться формулой Лагранжа, то для приближенного значения функции  $\tilde{z} = f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$  можно получить оценку при малых  $\Delta_{x_i}$

$$\Delta_z = |z - \tilde{z}| \approx \sum_{i=1}^n \Delta_{x_i} \left| \frac{\partial}{\partial x_i} f(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n) \right|. \quad (1.3)$$

Рассмотрим распространение погрешностей при выполнении арифметических операций. Пусть  $x, y$  – точные значения ( $x > 0, y > 0$ ),  $\tilde{x}, \tilde{y}$  – приближенные значения,  $\Delta_x, \Delta_y$  – абсолютные, а  $\delta_x, \delta_y$  – относительные погрешности

значений  $\tilde{x}, \tilde{y}$ .

Если  $f \equiv x + y$ , тогда  $f'_x = f'_y = 1$  и из (1.3) следует, что  $\Delta_{x+y} = \Delta_x + \Delta_y$ . Приведем формулы определяющие погрешности арифметических операций.

1) При выполнении сложения:

$$\Delta_{x+y} = \Delta_x + \Delta_y, \quad \delta_{x+y} = \frac{\delta_x \tilde{x} + \delta_y \tilde{y}}{\tilde{x} + \tilde{y}}.$$

2) При выполнении вычитания:

$$\Delta_{x-y} = \Delta_x + \Delta_y, \quad \delta_{x-y} = \frac{\delta_x \tilde{x} + \delta_y \tilde{y}}{\tilde{x} - \tilde{y}}.$$

3) При выполнении умножения:

$$\Delta_{xy} = \Delta_y \tilde{x} + \Delta_x \tilde{y}, \quad \delta_{xy} = \delta_x + \delta_y.$$

4) При выполнении деления:

$$\Delta_{x/y} = \frac{\Delta_x \tilde{y} + \Delta_y \tilde{x}}{\tilde{y}^2}, \quad \delta_{x/y} = \delta_x + \delta_y.$$

#### 1.4 Обратная задача теории погрешностей.

Обратная задача теории погрешностей состоит в определении допустимой погрешности аргументов по допустимой погрешности функции.

Для функции одной переменной  $z = f(x)$  абсолютную погрешность можно приближенно вычислить по формуле

$$\Delta_x = \frac{1}{|f'(\tilde{x})|} \Delta_z, \quad f'(\tilde{x}) \neq 0.$$

Для функции нескольких переменных  $z = f(x_1, x_2, \dots, x_n)$  задача решается при следующих ограничениях.

1) Если значение одного из аргументов значительно труднее измерить или вычислить с той же точностью, что и значения остальных аргументов, то погрешность именно этого аргумента надо согласовать с требуемой погрешностью функции.





$$x_m = y_m, \quad x_i = y_i - \sum_{j=i+1}^m c_{ij}x_j, \quad i = m - 1, \dots, 1.$$

- 4) В случае, если  $a_{kk}^{(k-1)} = 0$ , меняют местами строки или столбцы и помещают ненулевой элемент на его место. При перестановке столбцов запоминают переименование неизвестных, чтобы затем вернуться к исходному вектору  $x$ .

Если невозможно найти ненулевой элемент, чтобы поставить его на место  $a_{kk}^{(k-1)}$  это означает, что  $\det A = 0$ . В этом случае нужно прервать вычисления и выдать соответствующее сообщение .

### 2.3 Метод Гаусса с выбором главного элемента.

- Наиболее часто используются схемы метода Гаусса с выбором главного элемента:
  - по строке;
  - по столбцу;
  - по всей матрице.

На  $k$ -ом шаге метода Гаусса с выбором главного элемента *по столбцу* выбирают максимальный элемент среди всех элементов  $a_{kk}^{(k-1)}, \dots, a_{mk}^{(k-1)}$ , стоящих в  $k$ -ом столбце, и ставят его на место  $a_{kk}^{(k-1)}$ , производя соответствующую перенумерацию строк. Аналогично делается выбор максимального элемента по столбцу или по всей матрице.

### 2.4 Оценка погрешности и уточнение полученного решения.

Оценка погрешности решения системы линейных уравнений методом Гаусса - задача более сложная, чем решение самой системы. Поэтому мы рассмотрим метод уточнения решения, позволяющий уменьшить влияние ошибок округления на окончательный результат и найти решение с заданной точностью. Будем считать полученное решение нулевым приближением и обозначим его  $x_i^{(0)}$ ,  $i = 1, 2, \dots, m$ . Подставим это приближенное решение в левые части

исходной системы (2.4) и получим

$$Ax^{(0)} = f^{(0)}. \quad (2.5)$$

Вычтем из каждого уравнения исходной системы (2.4) соответствующее уравнение полученной системы (2.5), тогда получим систему уравнений относительно невязок, т.е.

$$A\varepsilon^{(0)} = \beta^{(0)}, \quad (2.6)$$

где  $\varepsilon^{(0)} = x - x^{(0)}$ ,  $\beta^{(0)} = f - f^{(0)}$ .

Полученную систему (2.6) можно решить, описанным выше, методом Гаусса. Новое приближение исходной системы запишем в виде:

$$x_i^{(1)} = x_i^{(0)} + \varepsilon_i^{(0)}, \quad i = 1, 2, \dots, m.$$

Если теперь повторить описанную процедуру, то новую поправку к решению исходной системы (2.4) можно получить из решения системы уравнений:

$$A\varepsilon^{(1)} = \beta^{(1)},$$

где  $\beta^{(1)} = f - f^{(1)}$ . И новое приближение запишется в виде

$$x_i^{(2)} = x_i^{(1)} + \varepsilon_i^{(1)}, \quad i = 1, 2, \dots, m.$$

Такой процесс необходимо продолжить до тех пор, пока все  $\varepsilon_i$  не станут достаточно малыми. Важно отметить, что нельзя останавливать вычисления при малости  $\beta_i$ , т.к. решение при этом может быть плохим.

## 2.5 Вычисление определителя матрицы.

В большинстве существующих стандартных программ одновременно с решением системы линейных алгебраических уравнений (2.4) вычисляется определитель матрицы  $A$ . Если по схеме Гаусса система (2.4) приводится к системе (??), то это достигается с помощью следующих элементарных преобразований:

- 1) деление на "ведущие элементы"  $a_{11}, a_{22}^{(1)}, \dots, a_{mm}^{(m-1)}$ , которые отличны от нуля, если это не так, то необходимо перестановкой строк заменить диагональный элемент и запомнить число перестановок;

2) вычитание из строк матрицы  $A$  и промежуточных матриц чисел, пропорциональных элементам соответствующих ведущих строк.

При первой операции определитель матрицы также делится на соответствующий "ведущий элемент", при второй – не меняется. Поэтому

$$\det C = 1 = \frac{\det A}{(-1)^n a_{11} a_{22}^{(1)} \dots a_{mm}^{(m-1)}},$$

$$\det A = (-1)^n a_{11} a_{22}^{(1)} \dots a_{mm}^{(m-1)},$$

где  $n$  – число перестановок строк.

## 2.6 Вычисление обратной матрицы.

Нахождение матрицы, обратной матрице  $A$ , эквивалентно решению матричного уравнения

$$AX = E, \tag{2.7}$$

где  $E$  – единичная матрица, а  $X$  – искомая квадратная матрица. Пусть  $A = [a_{ij}]$ ,  $X = [x_{ij}]$ . Уравнение (2.7) можно записать в виде системы  $m^2$  уравнений

$$\sum_{k=1}^m a_{ik} x_{kj} = \delta_{ij}, \quad i, j = 1, 2, \dots, m, \tag{2.8}$$

где  $\delta_{ij} = 1$  при  $i = j$  и  $\delta_{ij} = 0$  при  $i \neq j$ .

Для дальнейшего важно заметить, что система (2.8) распадается на  $m$  независимых систем уравнений с одной и той же матрицей  $A$ , но с различными правыми частями. Эти системы имеют вид

$$Ax^{(j)} = \delta^{(j)}, \quad j = 1, 2, \dots, m, \tag{2.9}$$

где  $x^{(j)} = (x_{1j}, x_{2j}, \dots, x_{mj})^T$ , у вектора  $\delta^{(j)}$  равна единице  $j$ -я компонента и равны нулю остальные компоненты.

Для решения систем (2.9) можно использовать метод Гаусса (обычный или с выбором главного элемента). Поскольку все системы (2.9) имеют одну и ту же матрицу  $A$ , достаточно один раз совершить прямой ход метода Гаусса и тем самым обращение матрицы потребует не намного больше времени, чем решение системы уравнений (2.4).

## 2.7 Метод квадратного корня.

Метод предназначен для решения систем уравнений

$$Ax = y \quad (2.10)$$

с симметричной матрицей. Он основан на разложении матрицы  $A$  в произведение

$$A = S^T D S,$$

где  $S$  - верхняя треугольная матрица с положительными элементами на главной диагонали,  $S^T$  - транспонированная к ней матрица,  $D$  - диагональная матрица, на диагонали которой стоят числа, равные  $\pm 1$ .

Элементы матриц  $S$  и  $D$  вычисляются следующим образом

$$\begin{aligned} d_{11} &= \text{sign } a_{11}, \quad s_{11} = \sqrt{|a_{11}|}, \\ s_{1j} &= a_{1j}/s_{11}, \quad j = 2, 3, \dots, m, \\ d_{ii} &= \text{sign} \left( a_{ii} - \sum_{l=1}^{i-1} |s_{li}|^2 d_{ll} \right), \\ s_{ii} &= \left( |a_{ii} - \sum_{l=1}^{i-1} |s_{li}|^2 d_{ll} \right)^{1/2} \quad (i > 1), \\ s_{ij} &= \frac{a_{ij} - \sum_{l=1}^{i-1} s_{li} s_{lj} d_{ll}}{s_{ii} d_{ii}}, \quad j = i+1, \dots, m, \quad i = 2, 3, \dots, m. \end{aligned}$$

Система (2.10) распадается на две системы

$$Sx = \bar{y}, \quad (2.11)$$

$$S^T D \bar{y} = y \quad (2.12)$$

Решая систему (2.12), получаем  $\bar{y}$  по формулам:

$$\begin{aligned} \bar{y}_1 &= y_1 / (s_{11} d_{11}), \\ \bar{y}_i &= \frac{1}{s_{ii} d_{ii}} \left( y_i - \sum_{k=1}^{i-1} s_{ki} d_{kk} \bar{y}_k \right), \quad i = 2, \dots, m. \end{aligned}$$

Подставляя это решение в (2.11), можем найти  $x$  по формулам

$$\begin{aligned} x_m &= \bar{y}_m / s_{mm}, \\ x_i &= \frac{1}{s_{ii}} \left( \bar{y}_i - \sum_{k=i+1}^m s_{ik} x_k \right), \quad i = m-1, \dots, 1. \end{aligned}$$

## 2.8 Метод прогонки.

Для численного решения систем линейных алгебраических уравнений специального вида с трехдиагональными матрицами применяется *метод прогонки*, который представляет собой вариант метода последовательного исключения неизвестных.

В общем случае система линейных алгебраических уравнений с трехдиагональной матрицей имеет вид:

$$a_i y_{i+1} + b_i y_i + c_i y_{i-1} = f_i, \quad i = 1, 2, \dots, m-1, \quad (2.13)$$

$$\alpha_{01} y_0 + \beta_{01} y_1 = \gamma_{01}, \quad (2.14)$$

$$\alpha_{02} y_{m-1} + \beta_{02} y_m = \gamma_{02}, \quad (2.15)$$

где  $a_i, b_i, c_i, f_i, \alpha_{0k}, \beta_{0k}, \gamma_{0k}$  – заданные числа.

Решение системы (2.13) – (2.15) по методу *правой прогонки* состоит в следующем:

- 1) определяем прогоночные коэффициенты (прямой ход прогонки)

$$X_1 = -\beta_{01}/\alpha_{01}, \quad Z_1 = \gamma_{01}/\alpha_{01},$$

$$X_{i+1} = -\frac{a_i}{b_i + c_i X_i}, \quad Z_{i+1} = \frac{f_i - c_i Z_i}{b_i + c_i X_i}, \quad i = 1, 3, \dots, m-1;$$

- 2) определяем неизвестные  $y_i$  (обратный ход прогонки)

$$y_m = \frac{\gamma_{02} - \alpha_{02} Z_m}{\beta_{02} + \alpha_{02} X_m}, \quad y_{i-1} = X_i y_i + Z_i, \quad i = m, m-1, \dots, 1.$$

Метод правой прогонки будет устойчив к погрешностям округления при выполнении условий:

$$|\alpha_{01}| \geq |\beta_{01}|, \quad |b_i| \geq |c_i| + |a_i|, \quad i = 1, 2, \dots, m-1. \quad (2.16)$$

Если условия (2.16) не выполнены, то можно решить систему (2.13) – (2.15) методом *левой прогонки*, который состоит в следующем:

1) определяем прогоночные коэффициенты (прямой ход прогонки)

$$X_{m-1} = -\alpha_{02}/\beta_{02}, \quad Z_{m-1} = \gamma_{02}/\beta_{02},$$

$$X_{i-1} = -\frac{c_i}{b_i + a_i X_i}, \quad Z_{i-1} = \frac{f_i - a_i Z_i}{b_i + a_i X_i}, \quad i = m-1, m-2, \dots, 1;$$

2) определяем неизвестные  $y_i$  (обратный ход прогонки)

$$y_0 = \frac{\gamma_{01} - \alpha_{01} Z_0}{\alpha_{01} + \beta_{01} X_0}, \quad y_{i+1} = X_i y_i + Z_i, \quad i = 0, 1, \dots, m-1.$$

Метод левой прогонки будет устойчив к погрешностям округления при выполнении условий:

$$|\beta_{02}| \geq |\alpha_{02}|, \quad |b_i| \geq |c_i| + |a_i|, \quad i = 1, 2, \dots, m-1. \quad (2.17)$$

В случае невыполнения условий (2.16), (2.17), для решения системы (2.13) – (2.15) можно использовать методы немонотонной или ортогональной прогонки.

### 3 Лабораторная работа.

#### Итерационные методы решения систем линейных алгебраических уравнений.

Итерационные методы решения систем линейных уравнений отличаются самоисправляемостью и простотой реализации на ЭВМ. Рассмотрим систему линейных уравнений:

$$Ax = f, \quad (3.18)$$

где  $A = [a_{ij}]$ ,  $i, j = 1, 2, \dots, m$ ,  $x = (x_1, x_2, \dots, x_m)^T$ ,  $f = (f_1, f_2, \dots, f_m)^T$ .

Итерационный метод называется *одношаговым*, если для нахождения  $(n+1)$ -й итерации  $x^{(n+1)}$  требуется помнить только одну предыдущую итерацию  $x^{(n)}$ . Каноническая форма одношагового итерационного метода решения системы (3.18) имеет вид:

$$B^{(n+1)} \frac{x^{(n+1)} - x^{(n)}}{\tau^{(n+1)}} + Ax^{(n)} = f, \quad n = 0, 1, \dots,$$

где  $B^{(n+1)}$  – матрица, задающая тот или иной итерационный метод,  $\tau^{(n+1)}$  – итерационный параметр. Итерационные методы делятся на *явные* ( $B^{(n)} = E$ ) и  *неявные* ( $B^{(n)} \neq E$ ), *стационарные* ( $B^{(n)} = B$   $\tau^{(n)} = \tau$ ) и *нестационарные*. Рассмотрим некоторые одношаговые итерационные методы.

### 3.1 Метод простой итерации.

*Метод простой итерации* для системы (3.18) имеет вид:

$$\frac{x^{(n+1)} - x^{(n)}}{\tau} + Ax^{(n)} = f, \quad n = 0, 1, \dots,$$

где  $\tau$  - постоянный итерационный параметр. Если  $A$  - симметричная матрица, то этот метод сходится при  $0 < \tau < 2/\lambda_{max}$ , где  $\lambda_{max}$  - максимальное собственное значение матрицы  $A$ .

### 3.2 Метод Якоби.

*Метод Якоби* в координатной форме имеет вид:

$$x_i^{(n+1)} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(n)} - \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j^{(n)} + \frac{f_i}{a_{ii}}, \quad (3.19)$$

$$n = 0, 1, \dots, \quad i = 1, 2, \dots, m.$$

Начальные значения  $x_i^{(0)}$ ,  $i = 1, 2, \dots, m$  задаются произвольно, обычно столбец свободных членов системы (3.19). Окончание вычислений определяется либо заданием максимального числа итераций  $n_0$ , либо условием

$$\max_{1 \leq i \leq m} |x_i^{(n+1)} - x_i^{(n)}| < \varepsilon,$$

где  $\varepsilon > 0$  – заданное число.

**Теорема.** Пусть  $A$  - симметричная положительно определенная матрица с диагональным преобладанием, то есть

$$a_{ii} > \sum_{j \neq i} |a_{ij}|, \quad i = 1, \dots, m.$$

Тогда метод Якоби сходится.

### 3.3 Метод Зейделя.

Итерационный метод Зейделя в координатной форме имеет вид:

$$x_i^{(n+1)} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(n+1)} - \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j^{(n)} + \frac{f_i}{a_{ii}}, \quad (3.20)$$

$$n = 0, 1, \dots, n_0, \quad i = 1, 2, \dots, m.$$

Компоненты вектора  $(n+1)$ -ой итерации  $x^{(n+1)}$  вычисляются последовательно. При вычислении значения  $x_i^{(n+1)}$  используются уже полученные значения  $x_1^{(n+1)}, x_2^{(n+1)}, \dots, x_{i-1}^{(n+1)}$ .

Приведем два достаточных условия сходимости метода Зейделя.

**Теорема 1.** Если  $A$  - матрица с диагональным преобладанием, то метод Зейделя сходится.

**Теорема 2.** Если  $A$  - симметричная положительно определенная матрица, то метод Зейделя сходится.

Если в матрице  $A$  нет диагонального преобладания, его нужно добиться с помощью элементарных преобразований.

**Пример.** Система

$$4.51x_1 - 1.8x_2 + 3.6x_3 = -1.7, \quad (I)$$

$$3.1x_1 + 2.3x_2 - 1.2x_3 = 3.6, \quad (II)$$

$$1.8x_1 + 2.5x_2 + 4.6x_3 = 2.2 \quad (III)$$

приводится к следующей системе с диагональным преобладанием:

$$7.61x_1 + 0.5x_2 + 2.4x_3 = 1.9, \quad (I + II)$$

$$3.19x_1 + 9.1x_2 + 4.4x_3 = 9.7, \quad (2III + II - I)$$

$$-1.3x_1 + 0.2x_2 + 5.8x_3 = -1.4 \quad (III - II)$$

Процесс Зейделя для последней системы сходится.

Для того, чтобы воспользоваться достаточным условием сходимости метода Зейделя, даваемым теоремой 2, применяют следующую процедуру. Умножают левую и правую части системы (3.18) слева на транспонированную матрицу  $A^T$ . Получают систему

$$Cx = d, \quad (3.21)$$

где  $A^T A = C$ ,  $A^T f = d$ . Матрица  $A^T A$  является симметричной и положительно определенной. Процесс Зейделя (3.20) для системы (3.21) с такой матрицей сходится.

### 3.4 Метод верхней релаксации.

Обобщением метода Зейделя является метод *верхней релаксации*, который в координатной форме записывается следующим образом

$$x_i^{(n+1)} + \omega \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(n+1)} = (1 - \omega)x_i^{(n)} - \omega \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j^{(n)} + \omega \frac{d_i}{a_{ii}},$$

$$n = 0, 1, \dots, \quad i = 1, 2, \dots, m.$$

При  $\omega = 1$  этот метод совпадает с методом Зейделя. Как и в методе Зейделя компоненты вектора  $(n+1)$ -ой итерации  $x^{(n+1)}$  вычисляются последовательно. При вычислении значения  $x_i^{(n+1)}$  используются уже полученные значения  $x_1^{(n+1)}, x_2^{(n+1)}, \dots, x_{i-1}^{(n+1)}$ .

**Теорема.** Пусть  $A$  - симметричная положительно определенная матрица, тогда метод верхней релаксации сходится при  $0 < \omega < 2$ .

Скорость сходимости метода верхней релаксации зависит от параметра  $\omega$ .

### 3.5 Итерационные методы с чебышевским набором параметров.

Явный нестационарный итерационный метод

$$x^{(k+1)} = x^{(k)} - \tau^{(k+1)}(Ax^{(k)} - f), \quad k = 0, 1, \dots, \quad (3.22)$$

где  $x^{(0)}$  заданный вектор, а параметры  $\tau^{(k)}$ ,  $k = 1, \dots, n$  выбираются из условий минимума нормы погрешности  $\|x^{(n)} - x\|$ , называется *итерационным методом с чебышевским набором параметров*.

Для симметричной положительно определенной матрицы  $A$ , для которой выполняется оценка

$$\gamma_1 E \leq A \leq \gamma_2 E,$$

в методе (3.22) множество итерационных параметров задается формулой

$$\tau^{(k)} = \frac{\tau^{(0)}}{1 + \rho_0 t_k}, \quad k = 1, 2, \dots, n, \quad (3.23)$$

где

$$\tau^{(0)} = \frac{2}{\gamma_1 + \gamma_2}, \quad \rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\gamma_1}{\gamma_2}, \quad t_k = -\cos \frac{(2k-1)\pi}{2n}, \quad k = 1, 2, \dots, n.$$

Следует учесть, что это множество не является упорядоченным, и параметры в формуле (3.22) можно в принципе брать в произвольном порядке. В качестве констант  $\gamma_1$ ,  $\gamma_2$  годятся любые оценки снизу и сверху спектра матрицы  $A$ :

$$\gamma_1 \leq \lambda_{\min}(A) < \lambda_{\max}(A) \leq \gamma_2.$$

Для метода с чебышевским набором параметров справедлива оценка погрешности

$$\|x^{(n)} - x\|_2 \leq q_n \|x^{(0)} - x\|_2,$$

где

$$q_n = \frac{2\rho_1^n}{1 + \rho_1^{2n}}, \quad \rho_1 = \frac{1 - \sqrt{\xi}}{1 + \sqrt{\xi}}, \quad \xi = \frac{\gamma_1}{\gamma_2},$$

Число итераций, необходимое для достижения заданной точности  $\varepsilon$  равно

$$n \geq n_0(\varepsilon) = \frac{\ln(\frac{2}{\varepsilon})}{\ln(\frac{1}{\rho_1})}.$$

В случае малых  $\xi$  можно получить приближенную оценку для числа итераций

$$n_0(\varepsilon) \cong \frac{\ln(\frac{2}{\varepsilon})}{2\sqrt{\xi}}.$$

При практическом применении метода сначала вычисляют число итераций, необходимых для достижения заданной точности, а затем вычисляют параметры. Порядок выбора параметров  $\tau^{(k)}$  существенно влияет на численную устойчивость метода. Использование параметров в произвольном порядке может привести к недопустимому возрастанию вычислительных погрешностей.

В настоящее время известен алгоритм построения такого упорядоченного набора итерационных параметров (3.23), для которого итерационный метод (3.22) является устойчивым (вычислительные погрешности не нарастают). Укажем способ упорядочения итерационных параметров  $\tau_k$ , которые вычисляются по формуле

$$\tau_k = \frac{\tau_0}{1 + \rho_0 t_k}, \quad t_k = -\cos \left[ \frac{\pi}{2n} \theta_n(k) \right], \quad \theta_n(k) = 2k - 1, \quad k = 1, 2, \dots, n.$$

Таким образом, достаточно упорядочить множество  $\theta_n$  первых  $n$  нечетных чисел. Сделаем это для случая  $n = 2^p$ . При практическом применении следует выбрать ближайшую к  $n_0(\varepsilon)$  степень двойки.

Алгоритм построения множества  $\theta_n$  основан на поэтапных переходах от множеств  $\theta_m$  к множествам  $\theta_{2m}$ ,  $m = 1, \dots, 2^{p-1}$  по формулам

$$\theta_1 = \{1\},$$

$$\theta_{2m}(2i-1) = \theta_m(i), \quad \theta_{2m}(2i) = 4m - \theta_{2m}(2i-1), \quad i = 1, 2, \dots, m. \quad (3.24)$$

Например, если  $n = 16 = 2^4$ , то строятся последовательно множества

$$\theta_2 = \{1, 3\}, \quad \theta_4 = \{1, 7, 3, 5\}, \quad \theta_8 = \{1, 15, 7, 9, 3, 13, 5, 11\},$$

$$\theta_{16} = \{1, 31, 15, 17, 7, 25, 9, 23, 3, 29, 13, 19, 5, 27, 11, 21\}.$$

При переходе от  $\theta_m$  к  $\theta_{2m}$ , согласно (3.24), после каждого члена  $\theta_m(i)$  ставится член  $\theta_{2m}(2i) = 4m - \theta_m(i)$ .

В заключение, рассмотрим некоторые итерационные методы вариационного типа.

### 3.6 Метод минимальных невязок.

Рассмотрим систему уравнений (3.18) с симметричной положительно определенной матрицей  $A$ . *Методом минимальных невязок* называется итерационный метод определяемый формулой

$$x^{(k+1)} = x^{(k)} - \tau^{(k+1)}(Ax^{(k)} - f), \quad (3.25)$$

где параметр  $\tau^{(k+1)}$  выбирается из условия минимума  $\|r_{k+1}\|_2 = \|Ax^{(k+1)} - f\|_2$  при заданной норме  $\|r_k\|_2$ :

$$\tau^{(k+1)} = \frac{(Ar_k, r_k)}{\|Ar_k\|_2^2}, \quad r_k = Ax^{(k)} - f. \quad (3.26)$$

Таким образом, в методе минимальных невязок переход от  $k$ -й итерации к  $(k+1)$ -й осуществляется в таком порядке. По найденному значению  $x^{(k)}$  вычисляется вектор невязки  $r_k = Ax^{(k)} - f$  и по формуле (3.26) находится параметр  $\tau^{(k+1)}$ . Затем по формуле (3.25) досчитывается вектор  $x^{(k+1)}$ .

Для метода минимальных невязок выполняется оценка

$$\|A(x^{(n)} - x)\|_2 \leq \rho_0^n \|A(x^{(0)} - x)\|_2, \quad n = 0, 1, \dots,$$

где

$$\rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)},$$

$\lambda$  – собственное значение матрицы  $A$ .

### 3.7 Метод скорейшего спуска.

Рассмотрим систему уравнений (3.18) с симметричной положительно определенной матрицей  $A$ . Если в методе (3.25) итерационный параметр  $\tau^{(k+1)}$  выбирается из условия минимума  $\|z_{k+1}\|_A = \|x^{(k+1)} - x\|_A$  при заданном  $z_k$ , то метод (3.25) называется *методом скорейшего спуска*. Итерационные параметры вычисляются по формуле

$$\tau^{(k+1)} = \frac{(r_k, r_k)}{(Ar_k, r_k)}, \quad r_k = Ax^{(k)} - f.$$

Для метода скорейшего спуска справедлива оценка

$$\|x^{(n)} - x\|_A \leq \rho_0^n \|x^{(0)} - x\|_A, \quad n = 0, 1, \dots,$$

где

$$\rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}.$$

### 3.8 Метод сопряженных градиентов.

Метод сопряженных градиентов является двухшаговым итерационным методом, т.е. для нахождения новой итерации  $x^{(k+1)}$  используются две предыдущие итерации  $x^{(k)}$  и  $x^{(k-1)}$ .

Явный двухшаговый итерационный метод

$$x^{(k+1)} = \alpha_{k+1}x^{(k)} + (1 - \alpha_{k+1})x^{(k-1)} - \tau^{(k+1)}\alpha_{k+1}r_k, \quad (3.27)$$

где  $r_k = Ax^{(k)} - f$ ,

$$\tau^{(k+1)} = \frac{(r_k, r_k)}{(Ar_k, r_k)}, \quad k = 0, 1, \dots,$$

$$\alpha_{k+1} = \left[ 1 - \frac{\tau^{(k+1)}}{\tau^{(k)}} \frac{1}{\alpha_k} \frac{(r_k, r_k)}{(Ar_{k-1}, r_{k-1})} \right]^{-1}, \quad k = 1, 2, \dots, \quad \alpha_1 = 1,$$

называется *методом сопряженных градиентов*. Для начала счета по формуле (3.27) необходимо задать два начальных приближения  $x^{(0)}$  и  $x^{(1)}$ . Начальное приближение  $x^{(0)}$  будем задавать произвольно, а вектор  $x^{(1)}$  вычислять по одношаговой формуле, которая получается из (3.27) при  $k = 0$ ,  $\alpha_1 = 1$ , т.е. по формуле

$$x^{(1)} = x^{(0)} - \tau^{(1)}(Ax^{(0)} - f).$$

## 4 Лабораторная работа.

### Интерполирование алгебраическими многочленами.

#### 4.1 Постановка задачи.

Пусть на отрезке  $a \leq x \leq b$  заданы точки  $x_k$ ,  $k = 0, 1, \dots, n$  (*узлы интерполирования*), в которых известны значения функции  $f(x)$ . Задача интерполирования состоит в том, чтобы построить многочлен

$$P_n(x) = a_0 + a_1x + \dots + a_nx^n$$

степени  $n$ , значения которого в заданных точках  $x_k$ , совпадают со значениями функции  $f(x)$  в этих точках. Такой полином существует и единственен.

#### 4.2 Интерполяционная формула Ньютона.

Формула Ньютона является разностным аналогом формулы Тейлора и имеет вид

$$\begin{aligned} P_n(x) = & f(x_0) + (x - x_0)f(x_0, x_1) + (x - x_0)(x - x_1)f(x_0, x_1, x_2) + \\ & + \dots + (x - x_0)(x - x_1)\dots(x - x_{n-1})f(x_0, x_1, \dots, x_n), \end{aligned} \quad (4.28)$$

где

$$f(x_i, x_j) = \frac{f(x_j) - f(x_i)}{x_j - x_i}, \quad i, j = 0, 1, \dots, n, \quad i \neq j -$$

разделенные разности первого порядка,

$$\begin{aligned} f(x_i, x_j, x_k) = & \frac{f(x_j, x_k) - f(x_i, x_j)}{x_k - x_i}, \\ & i, j, k = 0, 1, \dots, n, \quad i \neq j \neq k - \end{aligned}$$

разделенные разности второго порядка,

$$f(x_j, x_{j+1}, \dots, x_{j+k}) = \frac{f(x_{j+1}, x_{j+2}, \dots, x_{j+k}) - f(x_j, x_{j+1}, \dots, x_{j+k-1})}{x_{j+k} - x_j} -$$

разделенные разности  $k$ -го порядка.

При выводе формулы Ньютона не накладывается ограничений на порядок узлов  $x_0, x_1, \dots, x_n$ , поэтому множество интерполяционных формул можно получить из (4.28) перенумерацией узлов. Если  $x_0 < x_1 < x_2 < \dots < x_n$ , то (4.28) называется формулой интерполирования вперед. Интерполяционная формула Ньютона при интерполировании назад имеет вид:

$$P_n(x) = f(x_n) + (x - x_n)f(x_n, x_{n-1}) + (x - x_n)(x - x_{n-1})f(x_n, x_{n-1}, x_{n-2}) + \dots + (x - x_n)(x - x_{n-1}) \dots (x - x_1)f(x_n, x_{n-1}, \dots, x_0). \quad (4.29)$$

### 4.3 Погрешность интерполирования.

Заменяя функцию  $f(x)$  интерполяционным многочленом  $P_n(x)$ , мы допускаем погрешность

$$r_n(x) = f(x) - L_n(x),$$

которая называется *погрешностью интерполирования* или *остаточным членом интерполяционной формулы*. Если функция  $f(x)$  имеет непрерывную  $(n + 1)$ -ю производную, то имеет место следующая оценка остаточного члена

$$|r_n(x)| \leq \frac{M_{n+1}}{(n + 1)!} |\omega(x)|, \quad (4.30)$$

где

$$\omega(x) = \prod_{k=0}^n (x - x_k),$$

$$M_{n+1} = \sup_{x \in [a, b]} |f^{(n+1)}(x)|.$$

Погрешность интерполирования можно представить также через разделенную разность следующим образом

$$r_n(x) = \omega(x) f(x, x_0, x_1, \dots, x_n).$$

#### 4.4 Минимизация остаточного члена интерполирования.

Из формулы (4.30) следует, что для данной функции  $f(x)$  погрешность интерполирования зависит от выбора узлов  $x_0, x_1, \dots, x_n$  на отрезке  $[a, b]$ . Величину  $|\omega(x)|$  можно минимизировать за счет выбора узлов интерполирования  $x_i \in [a, b]$ ,  $i = 0, 1, \dots, n$ .

Таковыми оптимальными узлами для отрезка  $[-1, 1]$  являются корни многочлена Чебышева первого рода

$$T_{n+1}(x) = \frac{1}{2^n} \cos((n+1) \arccos x),$$

которые вычисляются по формуле

$$x_k = \cos \frac{(2k+1)\pi}{2(n+1)}, \quad k = 0, 1, \dots, n.$$

В случае произвольного отрезка  $[a, b]$  из этого равенства получим формулу для оптимальных узлов

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{(2k+1)\pi}{2(n+1)}, \quad k = 0, 1, \dots, n.$$

При этом оценка (4.30) примет вид

$$\max_{x \in [a, b]} |\omega(x)| = \frac{(b-a)^{n+1}}{2^{2n+1}} \quad |r_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

#### 4.5 Интерполирование по равноотстоящим узлам.

Приведем некоторые интерполяционные формулы для случая равноотстоящих узлов.

Пусть на отрезке  $[a, b]$  задана равномерная сетка  $x_k = x_0 + kh$ ,  $k = \overline{0, n}$  и значения функции  $f(x_k) = f_k$ ,  $h > 0$ . Пусть  $x = x_0 + th$ - точка интерполирования. Тогда, используя (4.28), получаем *первую интерполяционную формулу Ньютона*:

$$\begin{aligned} P_n(x) = P_n(x_0 + th) = f_0 + \frac{t}{1!} \Delta f_0 + \frac{t(t-1)}{2!} \Delta^2 f_0 + \dots \\ \dots + \frac{t(t-1)\dots(t-n+1)}{n!} \Delta^n f_0, \end{aligned}$$

где

$$\Delta^0 f_i = f_i, \quad \Delta^{k+1} f_i = \Delta^k f_{i+1} - \Delta^k f_i, \quad i = 0, 1, \dots, n-k, \quad -$$

конечные разности.

Положив  $x = x_n + th$ , получаем *вторую интерполяционную формулу Ньютона*

$$P_n(x) = P_n(x_n + th) = f_n + \frac{t}{1!} \Delta f_{n-1} + \frac{t(t+1)}{2!} \Delta^2 f_{n-2} + \dots \\ \dots + \frac{t(t+1)\dots(t+n-1)}{n!} \Delta^n f_0.$$

Для двух последних интерполяционных формул оценка погрешности интерполирования имеет вид

$$|r_n(x)| \leq \frac{h^{n+1} M_{n+1}}{(n+1)!} |t(t-1)\dots(t-n)|, \quad M_{n+1} = \max_{\xi \in [a,b]} |f^{(n+1)}(\xi)|.$$

При малых значениях  $h$  и при условии непрерывности  $f^{(n+1)}(x)$  можно приближенно считать

$$M_{n+1} \simeq \frac{\Delta^{n+1} f}{h^{n+1}}, \quad \Delta^{n+1} f = \max_{0 \leq m \leq n} |\Delta^{n+1} f_m|.$$

Тогда оценка остаточных членов первой и второй интерполяционных формул Ньютона имеет вид

$$|r_n(x)| \simeq \frac{|t(t-1)\dots(t-n)|}{(n+1)!} \Delta^{n+1} f, \quad (4.31)$$

Формула (4.31) удобна тем, что позволяет делать оценку ошибки интерполирования без исследования  $(n+1)$ -й производной функции  $f(x)$ . На окончательную погрешность интерполирования, разумеется, влияет и вычислительная погрешность, поэтому при вычислении интерполяционных многочленов желательно сводить число арифметических операций к минимуму.

## 5 Лабораторная работа.

### Интерполирование сплайнами.

При большом количестве узлов интерполяции возрастает степень интерполяционных многочленов, что приводит к плохому приближению из-за накопления вычислительных погрешностей. Высокой степени многочлена можно

избежать, разбив отрезок  $[a, b]$  на несколько частей с построением на каждой части самостоятельного интерполяционного многочлена (так называемая *кусочно-полиномиальная интерполяция*). В этом случае удобно пользоваться особым видом кусочно-полиномиальной интерполяции – интерполяцией сплайнами (spline – рейка). Преимуществом сплайнов перед обычным интерполированием является, во-первых, их сходимость, во-вторых, устойчивость процесса вычислений. Рассмотрим способ построения сплайнов третьей степени (так называемых кубических сплайнов), наиболее широко распространенных на практике.

### 5.1 Построение кубического сплайна.

Пусть на  $[a, b]$  задана непрерывная функция  $f(x)$ . Введем сетку

$$a = x_0 < x_1 < \dots < x_{N-1} < x_N = b$$

и обозначим  $f_i = f(x_i)$ ,  $i = \overline{0, N}$ .

*Сплайном*, соответствующим данной функции  $f(x)$  и данным узлам  $\{x_i\}_{i=0}^N$  называется функция  $S(x)$ , удовлетворяющая следующим условиям:

- 1) на каждом сегменте  $[x_{i-1}, x_i]$ ,  $i = \overline{1, N}$  функция  $S(x)$  является многочленом третьей степени;
- 2) функция  $S(x)$ , а также ее первая и вторая производные непрерывны на  $[a, b]$ ;
- 3)  $S(x_i) = f(x_i)$ ,  $i = \overline{0, N}$ .

Последнее условие называется *условием интерполирования*, а сплайн, определяемый этими тремя условиями, называется *интерполяционным кубическим сплайном* [1].

Сплайн, определенный таким образом, существует и единственен, если наложить два дополнительных условия на производные функции  $f$ . Покажем способ его построения.

На каждом из отрезков  $[x_{i-1}, x_i]$ ,  $i = \overline{1, N}$ , будем искать функцию  $S(x) = S_i(x)$  в виде многочлена третьей степени

$$S_i(x) = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3, \quad (5.32)$$

$$x_{i-1} \leq x \leq x_i, \quad i = \overline{1, N},$$

где  $a_i = S_i(x_i) = f(x_i)$ - известные величины,  $b_i = S'_i(x_i)$ ,  $c_i = S''_i(x_i)$ ,  $d_i = S'''_i(x_i)$ , - коэффициенты, подлежащие определению.

Для того, чтобы составить систему уравнений для коэффициентов сплайна, необходимо потребовать, чтобы в точке  $x_{i-1}$ ,  $i = \overline{2, N+1}$ , совпадали значения многочленов  $S_{i-1}$  и  $S_i$ , а также значения их первых и вторых производных, то есть

$$S_{i-1}(x_{i-1}) = S_i(x_{i-1}), \quad S'_{i-1}(x_{i-1}) = S'_i(x_{i-1}), \quad S''_{i-1}(x_{i-1}) = S''_i(x_{i-1}).$$

Учитывая выражения для функций  $S_i(x)$  и обозначая  $h_i = x_i - x_{i-1}$ , получаем уравнения

$$h_i b_i - \frac{h_i^2}{2} c_i + \frac{h_i^3}{6} d_i = f_i - f_{i-1}, \quad i = \overline{1, N}, \quad (5.33)$$

$$h_i c_i - \frac{h_i^2}{2} d_i = b_i - b_{i-1}, \quad i = \overline{2, N}, \quad (5.34)$$

$$h_i d_i = c_i - c_{i-1}, \quad i = \overline{2, N}. \quad (5.35)$$

Это система  $3N - 2$  уравнений относительно  $3N$  неизвестных. Два недостающих уравнения получают, задавая те или иные граничные условия для  $S(x)$ . Полученная система  $3N$  уравнений может быть решена тем или иным методом решения систем линейных уравнений (см. Лаб раб.2-3).

**Пример.** Предположим, например, что функция  $f(x)$  удовлетворяет условиям  $f''(a) = f''(b) = 0$ . Отсюда естественно требовать, чтобы  $S''(a) = S''(b) = 0$ . Отсюда получаем  $S''_1(x_0) = 0$ ,  $S''_N(x_N) = 0$ , то есть

$$c_1 - d_1 h_1 = 0, \quad c_N = 0. \quad (5.36)$$

Исключим из системы (5.2)-(5.5) переменные  $b_i, d_i$  и получим трехдиагональную систему, содержащую только переменные  $c_i$ :

$$h_i c_{i-1} + 2(h_i + h_{i+1})c_i + h_{i+1}c_{i+1} = 6 \left( \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right),$$

$$c_N = 0, \quad i = \overline{1, N-1},$$

где положено  $c_0 = 0$ . Эту систему можно решать любым методом, однако, такие трехдиагональные системы удобно решать методом прогонки, которая в данном случае устойчива. По найденным коэффициентам  $c_i$  определим остальные коэффициенты по формулам

$$d_i = \frac{c_i - c_{i-1}}{h_i}, \quad b_i = \frac{h_i}{2}c_i - \frac{h_i^2}{6}d_i + \frac{f_i - f_{i-1}}{h_i},$$

$$a_i = f(x_i), \quad i = \overline{1, N}.$$

## 6 Лабораторная работа.

### Численное интегрирование.

#### 6.1 Методы прямоугольников, трапеций и Симпсона.

Для приближенного вычисления интеграла  $\int_a^b f(x) dx$  используются конечные суммы вида  $\sum_{i=0}^n c_i f(x_i)$ , где  $c_i$  - числовые коэффициенты. Последнее выражение называется квадратурной формулой. Построение квадратурной формулы сопровождается оценкой погрешности, возникающей при замене интеграла квадратурной формулой.

Один из способов построения квадратурной формулы следующий. Отрезок  $[a, b]$  разбивается на несколько частей с построением на каждой части самостоятельного интерполяционного многочлена той или иной степени. Интеграл функции по отрезку разбиения заменяется интегралом от интерполяционного многочлена. Полученные интегралы суммируются по всем отрезкам разбиения. Рассматривая интерполяционные многочлены нулевой степени, получаем метод прямоугольников, первой степени - метод трапеций, второй степени - метод Симпсона.

#### 6.2 Формулы для вычислений

Пусть на  $[a, b]$  задана функция  $f(x)$ . Введем сетку

$$a = x_0 < x_1 < \dots < x_{N-1} < x_N = b,$$

где  $x_i = a + i \cdot h$ ,  $h = (b - a)/N$ . и обозначим  $f_i = f(x_i)$ ,  $i = \overline{0, N}$ .

На каждом сегменте  $[x_{i-1}, x_i]$ ,  $i = \overline{1, N}$  выберем срединную точку  $x_{i-1/2} =$

$x_i - 0.5h$  и обозначим  $f_{i-1/2} = f(x_{i-1/2})$ . Квадратурная формула прямоугольников имеет вид

$$\int_a^b f(x)dx \approx \sum_{i=1}^n f_{i-1/2}h,$$

Если функция  $f(x)$ , а также ее первая и вторая производные непрерывны на  $[a, b]$ , то погрешность формулы прямоугольников оценивается выражением

$$R = M_2 \frac{h^2(b-a)}{24}, \quad M_2 = \sup_{x \in [a,b]} |f''(x)|.$$

Квадратурная формула трапеций имеет вид

$$\int_a^b f(x)dx \approx \sum_{i=1}^n \frac{f_{i-1} + f_i}{2} h,$$

или

$$\int_a^b f(x)dx = h(0.5f_0 + f_1 + \dots + f_{N-1} + 0.5f_N).$$

Если функция  $f(x)$ , а также ее первая и вторая производные непрерывны на  $[a, b]$ , то погрешность этой формулы оценивается выражением

$$R = M_2 \frac{h^2(b-a)}{12}, \quad M_2 = \sup_{x \in [a,b]} |f''(x)|.$$

Квадратурная формула Симпсона имеет вид

$$\int_a^b f(x)dx \approx \sum_{i=1}^n \frac{h}{6} (f_{i-1} + 4f_{i-1/2} + f_i).$$

Чтобы не использовать дробных индексов, можно обозначить

$$x_i = a + 0.5hi, \quad f_i = f(x_i), \quad i = \overline{0, 2N}$$

и записать формулу Симпсона в виде

$$\int_a^b f(x)dx = \frac{b-a}{6N} (f_0 + f_{2N} + 2(f_2 + f_4 + \dots + f_{2N-2}) + 4(f_1 + f_3 + \dots + f_{2N-1})).$$

Если функция  $f(x)$  имеет на отрезке  $[a, b]$  непрерывные производные до четвертого порядка включительно, то погрешность этой формулы оценивается выражением

$$R = M_4 \frac{h^4(b-a)}{2880}, \quad M_4 = \sup_{x \in [a,b]} |f^{(4)}(x)|.$$

Решая неравенство  $R < \varepsilon$  относительно  $h$ , и делая вычисления с таким шагом, получаем заданную точность  $\varepsilon$  вычисления интеграла.

### 6.3 Апостериорная оценка погрешности методом Рунге. Автоматический выбор шага интегрирования.

Пусть  $I_i = \int_{x_{i-1}}^{x_i} f(x)dx \approx I_{h,i}$  – квадратурная формула, примененная на частичном отрезке и имеющая порядок  $m$ . Для формул прямоугольников и трапеций  $m = 3$ , а для формулы Симпсона  $m = 5$ . Проведем на каждом частичном отрезке  $[x_{i-1}, x_i]$  все вычисления дважды, один раз с шагом  $h_i$ , другой раз с шагом  $0.5h_i$ . Тогда справедлива оценка

$$I_i - I_{h/2,i} \approx \frac{I_{h/2,i} - I_{h,i}}{2^m - 1}.$$

Если для заданного  $\varepsilon$  правая часть не превосходит  $\frac{\varepsilon h_i}{b-a}$ ,  $i = 1, 2, \dots, N$ , то получим

$$|I - I_{h/2}| \leq \frac{\varepsilon}{b-a} \sum_{i=1}^N h_i = \varepsilon,$$

то есть будет достигнута заданная точность  $\varepsilon$ .

Если же на каком-то из частичных отрезков эта оценка не будет выполняться, то шаг на этом отрезке надо измельчить еще в два раза и снова оценить погрешность. Заметим, что для некоторых функций такое измельчение может продолжаться слишком долго. Поэтому в соответствующей программе следует предусмотреть ограничение сверху на число измельчений, а также возможность увеличения  $\varepsilon$ .

## 7 Лабораторная работа.

### Квадратурные формулы интерполяционного типа

Для приближенного вычисления интеграла  $\int_a^b \rho(x)f(x)dx$ , где  $\rho$  – положительная плотность, можно построить квадратурную формулу вида  $\sum_{k=0}^n c_k f(x_k)$ , заменяя  $f(x)$  на всем отрезке интерполяционным многочленом степени  $n$ . Построенная таким способом квадратурная формула называется *квадратурной формулой интерполяционного типа*. Коэффициенты  $c_k$  такой квадратурной формулы вычисляются по формулам

$$c_k = \int_a^b \frac{\rho(x)\omega(x)dx}{(x - x_k)\omega'(x_k)}, \quad k = 0, \dots, n,$$

где

$$\omega(x) = \prod_{j=0}^n (x - x_j), \quad \omega'(x_k) = \prod_{\substack{j=0 \\ j \neq k}}^n (x_k - x_j).$$

Квадратурная формула интерполяционного типа точна для всех многочленов степени  $n - 1$ .

## 7.1 Квадратурные формулы Гаусса

В предыдущем пункте предполагалось, что узлы квадратурной формулы заданы заранее. Оказывается, что за счет выбора узлов можно получить квадратурные формулы интерполяционного типа, которые будут точны для всех многочленов степени  $2n - 1$ . Они называются *формулами Гаусса*.

## 7.2 Некоторые квадратурные формулы Гаусса

Квадратурная формула Гаусса

$$\int_{-1}^1 f(x) dx \approx \sum_{k=1}^n A_k f(x_k) \quad (7.37)$$

имеет своими узлами корни многочлена Лежандра. Таблица узлов и коэффициентов этой формулы для  $n$  от 1 до 16 приводится ниже.

	Узлы	Коэффициенты
$n = 1 :$	0.0	2
$n = 2 :$	$\pm 0,5773502692$	1
$n = 3 :$	$\pm 0,7745966692$	5/9
	0.0	8/9
$n = 4 :$	$\pm 0,8611363116$	0.6521451549
	$\pm 0.3399810436$	0.6521451549
$n = 5 :$	$\pm 0.9061798459$	0.2369268851
	$\pm 0.5384693101$	0.4786286705
	0.0	0.5688888899
$n = 6 :$	$\pm 0.9324695142$	0.1713244924
	$\pm 0.6612093865$	0.3607615730
	$\pm 0.2386191861$	0.4679139346
$n = 7 :$	$\pm 0.4058451513$	0.3818300505
	$\pm 0.7415311855$	0.2797053914
	$\pm 0.9491079123$	0.1294849661
	0.0	0.4179591836
$n = 8 :$	$\pm 0.1834346424$	0.3626837833
	$\pm 0.5255324099$	0.3137066458
	$\pm 0.7966664774$	0.2223810344
	$\pm 0.9602898564$	0.1012285362
$n = 9 :$	$\pm 0.3242534234$	0.3123470770
	$\pm 0.6133714327$	0.2606106964
	$\pm 0.8360311073$	0.1806481606
	$\pm 0.9681602395$	0.0812743883
	0.0	0.3302393550
$n = 10 :$	$\pm 0.1488743389$	0.2955242247
	$\pm 0.4333953941$	0.2692667193
	$\pm 0.6794095682$	0.2190863625
	$\pm 0.8650633666$	0.1494513491
	$\pm 0.9739065285$	0.0666713443
$n = 11 :$	$\pm 0.2695431559$	0.2628045445
	$\pm 0.5190961292$	0.2331937645
	$\pm 0.7301520055$	0.1862902109
	$\pm 0.8870625997$	0.1255803694

	$\pm 0.9782286581$	0.0556685671
	0.0	0.2729250867
$n = 12 :$	$\pm 0.1252334085$	0.2491470458
	$\pm 0.3678314989$	0.2334925365
	$\pm 0.5873179542$	0.2031674267
	$\pm 0.7699026741$	0.1600783285
	$\pm 0.9041172563$	0.1069393259
	$\pm 0.9815606342$	0.0471753363
$n = 13 :$	$\pm 0.2304583159$	0.2262831802
	$\pm 0.4484927510$	0.2078160475
	$\pm 0.6423493394$	0.1781459807
	$\pm 0.8015780907$	0.1388735102
	$\pm 0.9175983992$	0.0921214998
	$\pm 0.9841830547$	0.0404840047
	0.0	0.2325515532
$n = 14 :$	$\pm 0.1080549487$	0.2152638534
	$\pm 0.3191123689$	0.2051984637
	$\pm 0.5152486363$	0.1855383974
	$\pm 0.6872929048$	0.1572031671
	$\pm 0.8272013150$	0.2115185706
	$\pm 0.9284348836$	0.0801580871
	$\pm 0.9862838086$	0.0351194603
$n = 15 :$	$\pm 0.2011940939$	0.1984314853
	$\pm 0.3941513470$	0.1861610000
	$\pm 0.5709721726$	0.1662692058
	$\pm 0.7244177313$	0.1395706779
	$\pm 0.8482065834$	0.1071592204
	$\pm 0.9372733924$	0.0703660474
	$\pm 0.9879925180$	0.0307532419
	0.0	0.2025782419
$n = 16 :$	$\pm 0.0950125098$	0.1894506104
	$\pm 0.2816035507$	0.1826034150
	$\pm 0.4580167776$	0.1691565193
	$\pm 0.6178762444$	0.1495959888
	$\pm 0.7554044083$	0.1246289712
	$\pm 0.8656312023$	0.0951585116
	$\pm 0.9445750230$	0.0622535239
	$\pm 0.9894009349$	0.1255803604

Для остатка квадратурной формулы (7.1) имеет место представление

$$R_n(f) = \frac{2^{2n+1}}{(2n+1)(2n)!} \left[ \frac{(n!)^2}{(2n)!} \right]^2 f^{(2n)}(\xi), \quad \xi \in [-1, 1].$$

Для вычисления интеграла

$$\int_a^b f(x) dx$$

следует сделать замену переменной интегрирования  $x = \frac{b-a}{2}t + \frac{a+b}{2}$ .

Формулами Эрмита называются формулы Гаусса

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx \approx \frac{\pi}{n} \sum_{k=1}^n f(x_k), \quad (7.38)$$

узлами которых являются корни многочлена Чебышева

$$x_k = \cos \frac{(2k-1)\pi}{2n}, \quad k = 1, \dots, n.$$

Остаточный член квадратурной формулы (7.2) равен

$$R_n(f) = \frac{\pi}{2^{2n-1}(2n)!} f^{(2n)}(\xi), \quad -1 \leq \xi \leq 1.$$

## 8 Лабораторная работа.

### Решение нелинейных уравнений

#### 8.1 Отделение корней.

Отделить корень уравнения

$$f(x) = 0 \quad (8.39)$$

значит найти такой отрезок области определения функции  $f(x)$  который содержит только один корень этого уравнения.

Для отделения корней уравнения (8.1) можно использовать следующий критерий: если на отрезке  $[a, b]$  функция  $f(x)$  непрерывна и монотонна, а ее значения на концах отрезка имеют разные знаки, то на этом отрезке существует и притом только один корень данного уравнения. Достаточным признаком монотонности функции  $f(x)$  на отрезке  $[a, b]$  является сохранение знака производной. При отделении корней стараются определить отрезок как можно меньшей длины.

Отделение корней уравнения (8.1) можно выполнить также графически. Найти корень уравнения (8.1) значит найти абсциссу точки пересечения графика функции  $f(x)$  с осью абсцисс. Если построить график функции  $f(x)$  затруднительно, то уравнение (8.1) следует представить в эквивалентном виде

$$f_1(x) = f_2(x) \quad (8.40)$$

с таким расчетом, чтобы графики функций  $f_1(x)$ ,  $f_2(x)$  строились проще. Корни же уравнения (8.2) определяются как абсциссы точек пересечения графиков функций  $f_1(x)$  и  $f_2(x)$ .

**Замечание.** Известно, что все корни алгебраического уравнения

$$f(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n = 0$$

расположены в кольце

$$\frac{|a_n|}{b + |a_n|} \leq |z| \leq 1 + \frac{c}{|a_0|},$$

где  $b = \max\{|a_0|, |a_1|, \dots, |a_{n-1}|\}$ ,  $c = \max\{|a_1|, |a_2|, \dots, |a_n|\}$ .

## 8.2 Метод деления отрезка пополам.

Простейшим алгоритмом уточнения корня на отрезке  $[a, b]$ , если  $f(x)$  непрерывная функция и  $f(a)f(b) < 0$  является метод деления отрезка пополам. Очевидно, что середина отрезка служит приближением к корню уравнения (8.1) с точностью  $\varepsilon = \frac{b-a}{2}$ . В средней точке отрезка  $[a, b]$  определяется знак функции  $f(x)$ , затем выбирается та половина отрезка, на концах которой функция принимает значения разных знаков, и деление повторяется. Если требуется найти корень с точностью до  $\varepsilon$ , то деление отрезка пополам продолжается до тех пор, пока длина отрезка не станет меньше  $\varepsilon/2$ . Тогда середина последнего отрезка даст значение корня с требуемой точностью.

## 8.3 Метод простой итерации.

Он состоит в том, что уравнение (8.1) заменяется эквивалентным уравнением вида

$$x = s(x) \quad (8.41)$$

и итерации образуются по правилу

$$x_{n+1} = s(x_n), \quad n = 0, 1, \dots,$$

причем задается начальное приближение  $x_0$ . Для сходимости большое значение имеет выбор функции  $s(x)$ .

Метод простой итерации сходится при надлежащем выборе начального приближения  $x_0$ , если

$$|s'(x)| < 1 \tag{8.42}$$

в некоторой окрестности корня. Более точно:

**Теорема.** Если  $|s'(x)| \leq q < 1$  при  $x \in [a - r, a + r]$ , причем  $|s(a) - a| \leq (1 - q)r$ , то уравнение (8.2) имеет на отрезке  $[a - r, a + r]$  единственное решение  $x_*$  и метод простой итерации сходится к  $x_*$  при любом начальном приближении  $x_0 \in [a - r, a + r]$ .

Для метода простой итерации можно пользоваться следующей оценкой погрешности

$$|x_k - x_*| \leq \frac{q^k}{1 - q} |s(x_0) - x_0|, \quad k = 1, 2, \dots$$

где  $x_*$  - истинное значение корня,  $|s'(x)| \leq q < 1$  для  $x \in [a, b]$ ,  $x_0 \in [a, b]$ .

Если функцию  $s(x)$  в уравнении (8.3) берем в виде

$$s(x) = x + \tau f(x), \quad \tau = const,$$

то получаем так называемый *метод релаксации*. Параметр  $\tau$  выбирается таким образом, чтобы выполнялась оценка (8.4). Если в некоторой окрестности корня выполняются условия

$$f'(x) < 0, \quad 0 < m_1 < |f'(x)| < M_1,$$

то метод релаксации сходится при  $\tau \in (0, 2/M_1)$ . Наиболее быстрая скорость сходимости достигается при оптимальном значении параметра  $\tau_0 = 2/(M_1 + m_1)$ . При этом значении для погрешности справедлива оценка

$$|x_n - x_*| \leq \rho_0^n |x_0 - x_*|,$$

где

$$\rho_0 = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{m_1}{M_1}.$$

**Пример.** Решим уравнение

$$f(x) = x^3 + 3x^2 - 1 = 0$$

методом простой итерации с точностью  $\varepsilon = \frac{1}{2} \cdot 10^{-4}$ .

По графику находим, что уравнение имеет три корня, расположенных на отрезках  $[-3; -2]$ ,  $[-1; 0]$  и  $[0; 1]$ . Найдем корень на отрезке  $[-3; -2]$ . Поделим уравнение на  $x^2$  и приведем к виду

$$x = \frac{1}{x^2} - 3. \quad (8.43)$$

Положим  $s(x) = \frac{1}{x^2} - 3$  и возьмем  $x_0 = -2.5$ . На отрезке  $[-3; -2]$  ( $r=0.5$ ) выполняются условия теоремы:

$$\max_{-3 \leq x \leq -2} |s'(x)| = q = 1/4, \quad |s(x_0) - x_0| = 0.34 < (1 - q)r = 0.375.$$

Найдем число итераций, необходимых для достижения заданной точности:

$$\frac{|s(x_0) - x_0|}{1 - q} q^n \leq \varepsilon \Rightarrow n \geq 6.$$

После шести итераций получаем приближенное значение корня  $-2.87938$ .

На отрезках  $[-1; 0]$  и  $[0; 1]$  представление (8.5) не годится для нахождения корней, поскольку производная функции  $s(x)$  больше 1 на этих отрезках. Для нахождения корней здесь удобно использовать представления

$$x = \pm 1/\sqrt{x + 3}.$$

## 8.4 Метод Ньютона и метод секущих.

*Метод Ньютона* в случае простого вещественного корня имеет вид

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \dots, \quad (8.44)$$

в случае корня кратности  $p$  -

$$f'(x_k) \frac{x_{k+1} - x_k}{p} + f(x_k) = 0.$$

Оценка погрешности следующая:

$$|x_k - x_*| \leq q^{2k} |x_0 - x_*|, \quad k = 1, 2, \dots$$

где

$$q = \frac{M_{p+1}|x_0 - x_*|}{m_p p(p+1)} < 1.$$

Можно пользоваться оценкой погрешности как в методе простой итерации учитывая что для метода Ньютона

$$s(x) = x - p \frac{f(x)}{f'(x)}.$$

*Модифицированный метод Ньютона*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}, \quad k = 0, 1, \dots,$$

применяют в том случае, когда хотят избежать многократного вычисления производной  $f'(x_k)$ .

В методе Ньютона требуется вычислять производную функции, что не всегда удобно. Можно заменить производную первой разделенной разностью, найденной по двум последним итерациям. Тогда вместо метода Ньютона (8.6) получим *метод секущих*

$$x_{k+1} = x_k - \frac{(x_k - x_{k-1})f(x_k)}{f(x_k) - f(x_{k-1})}.$$

Для начала процесса требуется задать значения  $x_0$  и  $x_1$ .

## 9 Лабораторная работа.

### Итерационные методы решения систем нелинейных уравнений

Рассмотрим систему нелинейных уравнений

$$F(x) = 0, \tag{9.45}$$

где  $F(x) = (f_1(x), f_2(x), \dots, f_m(x))$ ,  $x = (x_1, x_2, \dots, x_m)$ .

#### 9.1 Метод простой итерации.

Метод простой итерации применим к системам, которые предварительно приведены к виду

$$x = S(x). \tag{9.46}$$

Пусть  $x_0$  - начальное приближение. Последующие приближения в методе простой итерации находятся по формулам

$$x^{k+1} = S(x^k).$$

**Теорема.** Пусть оператор  $S$  определен на множестве

$$U_r(a) = \{x \in H : \|x - a\| \leq r\}$$

и является сжимающим на этом множестве с коэффициентом сжатия  $0 < q < 1$ , то есть для любых  $x', x'' \in U_r(a)$  выполняется неравенство

$$\|S(x') - S(x'')\| \leq q\|x' - x''\|.$$

Пусть кроме того

$$\|S(a) - a\| \leq (1 - q)r$$

Тогда в  $U_r(a)$  уравнение (9.2) имеет единственное решение  $x_*$  и к нему сходится последовательность  $x^k$ . Погрешность метода простой итерации оценивается неравенством

$$\|x^k - x_*\| \leq \frac{q^k}{1 - q} \|S(x^0) - x^0\|.$$

Следующее условие является достаточным для того, чтобы оператор  $S$  был сжимающим на множестве  $D$ :

$$\sum_{i=1}^n \max_D \left| \frac{\partial s_k}{\partial x_i} \right| < 1, \quad k = 1, \dots, m. \quad (9.47)$$

В случае  $n = 2$  можно применять следующий прием для приведения системы (9.1) к виду (9.2). Положим

$$\begin{aligned} s_1(x, y) &= x + \alpha f_1(x, y) + \beta f_2(x, y), \\ s_2(x, y) &= y + \gamma f_1(x, y) + \delta f_2(x, y), \end{aligned}$$

где  $\alpha\delta - \beta\gamma \neq 0$ . Коэффициенты  $\alpha, \delta, \beta, \gamma$  находят как решение системы уравнений

$$\begin{aligned} 1 + \alpha \frac{\partial f_1}{\partial x}(x_0, y_0) + \beta \frac{\partial f_2}{\partial x}(x_0, y_0) &= 0, \\ \alpha \frac{\partial f_1}{\partial y}(x_0, y_0) + \beta \frac{\partial f_2}{\partial y}(x_0, y_0) &= 0, \\ \gamma \frac{\partial f_1}{\partial x}(x_0, y_0) + \delta \frac{\partial f_2}{\partial x}(x_0, y_0) &= 0, \\ 1 + \gamma \frac{\partial f_1}{\partial y}(x_0, y_0) + \delta \frac{\partial f_2}{\partial y}(x_0, y_0) &= 0. \end{aligned}$$

Нетрудно убедиться в том, что при таком выборе параметров условие (9.3) будет соблюдено в некоторой малой окрестности начального приближения  $(x_0, y_0)$ .

*Метод релаксации* представляет собой частный случай метода простой итерации, когда  $S(x) = x - \tau F(x)$ . Величину  $\tau$  выбирают так, чтобы выполнялось условие (9.3).

Для удачного выбора начального приближения рекомендуется отделять корни системы (9.1) графически.

Итерационный процесс заканчивают, когда расстояние между соседними приближениями станет меньше заданной точности.

## 9.2 Метод Ньютона.

Пусть  $x^0$  - начальное приближение. Последующие приближения в методе Ньютона находятся по формуле

$$F(x^k) + J(x^k)(x^{k+1} - x^k) = 0,$$

где  $J(x)$  - матрица Якоби системы (9.1).

Для системы второго порядка

$$f(x, y) = 0, \quad g(x, y) = 0$$

последовательные приближения по методу Ньютона вычисляются по формулам

$$x^{k+1} = x^k - \frac{A_k}{J_k}, \quad y^{k+1} = y^k - \frac{B_k}{J_k},$$

$$A_k = \begin{vmatrix} f(x^k, y^k) & f'_y(x^k, y^k) \\ g(x^k, y^k) & g'_y(x^k, y^k) \end{vmatrix}$$

$$B_k = \begin{vmatrix} f'_x(x^k, y^k) & f(x^k, y^k) \\ g'_x(x^k, y^k) & g(x^k, y^k) \end{vmatrix}$$

$$J_k = \begin{vmatrix} f'_x(x^k, y^k) & f'_y(x^k, y^k) \\ g'_x(x^k, y^k) & g'_y(x^k, y^k) \end{vmatrix} \neq 0.$$

Метод Ньютона сходится, если начальное приближение выбрано удачно и матрица  $J(x_*)$  невырождена. На практике итерации обычно оканчивают, если  $\|x^{k+1} - x^k\| \leq \varepsilon$ .

Для того, чтобы на каждом шаге не обращать матрицу Якоби, применяют *модифицированный метод Ньютона*, который имеет вид

$$F(x^k) + J(x^k)(x^{k+1} - x^k) = 0.$$

Этот метод сходится медленнее, чем метод Ньютона, им пользуются в случае, когда начальное приближение удастся взять достаточно близким к истинному решению.

### 9.3 Нелинейные итерационные методы

*Нелинейный метод Якоби* для системы (9.1) имеет вид

$$f_i(x_1^k, x_2^k, \dots, x_{i-1}^k, x_i^{k+1}, x_{i+1}^k, \dots, x_m^k) = 0, \quad i = 1, \dots, m.$$

Здесь для отыскания  $x^{k+1}$  необходимо решить  $m$  независимых скалярных уравнений любым из изученных ранее итерационных методов решения нелинейных уравнений.

*Нелинейный метод Зейделя* состоит в последовательном решении уравнений

$$f_i(x_1^{k+1}, x_2^{k+1}, \dots, x_{i-1}^{k+1}, x_i^{k+1}, x_{i+1}^k, \dots, x_m^k) = 0, \quad i = 1, \dots, m.$$

относительно переменной  $x_i^{k+1}$ .

## 10 Лабораторная работа.

### Задача Коши для обыкновенных дифференциальных уравнений.

Пусть требуется найти решение задачи Коши

$$y' = f(x, y), \quad a \leq x \leq b, \quad y(a) = y_0. \quad (10.48)$$

На отрезке  $[a, b]$  зададим конечное множество точек  $\{x_i\}_{i=0}^N$  ( $a = x_0 < x_1 < \dots < x_N = b$ ). Будем искать приближенное решение  $y_i$  задачи (10.1) в выбранных точках  $x_i$ .

### 10.1 Метод Эйлера.

В методе Эйлера приближенные значения искомой функции  $y_i \approx y(x_i)$  вычисляются последовательно по формуле:

$$y_{i+1} = y_i + h_i f(x_i, y_i), \quad i = \overline{0, N-1},$$

где  $h_i = x_{i+1} - x_i$ . При этом искомая интегральная кривая  $y(x)$ , проходящая через точку  $(x_0, y_0)$ , заменяется ломаной с вершинами в точках  $(x_i, y_i)$ .

Оценка погрешности производится двойным пересчетом. Расчет на отрезке  $[x_i, x_{i+1}]$  повторяют с шагом  $h_i/2$  и погрешность более точного решения  $y_{i+1}^*$  (при шаге  $h_i/2$ ) оценивают по формуле

$$|y_{i+1}^* - y(x_{i+1})| \approx |y_{i+1}^* - y_{i+1}| \quad (10.49)$$

### 10.2 Метод Эйлера-Коши.

При решении задачи методом Эйлера-Коши, который иногда называют методом Хьюна, приближенное значение  $y(i+1)$  решения  $y(x)$  в точке  $x(i+1)$  вычисляется по формуле

$$y(i+1) = y(i) + (f(x(i), y(i)) + f(x(i+1), y(i) + hf(x(i), y(i))))h_i/2.$$

Остаточный член на каждом шаге в методе Эйлера-Коши имеет порядок  $O(h^3)$ . Оценка погрешности может быть получена с помощью двойного пересчета, аналогично (10.2):

$$|y_i^* - y(x_i)| \approx \frac{1}{3} |y_i^* - y_i|.$$

### 10.3 Метод Рунге-Кутты.

Пусть на отрезке  $[a, b]$  выбрана равномерная сетка

$$x_i = x_0 + ih, \quad i = \overline{1, N}, \quad h > 0; \quad N = \left[ \frac{b-a}{h} \right].$$

Предположим, что приближенное значение  $y_i$  решения исходной задачи в момент  $x = x_i$  уже известно.

*Метод Рунге-Кутты второго порядка точности* состоит в следующем. Сначала, используя схему Эйлера

$$\frac{y_{i+1/2} - y_i}{0.5h} = f(x_i, y_i)$$

вычисляют промежуточное значение  $y_{i+1/2}$ , затем из разностного уравнения

$$\frac{y_{i+1} - y_i}{h} = f(x_i + 0.5h, y_{i+1/2})$$

находят искомое значение  $y_{i+1}$ .

Приведем примеры методов Рунге-Кутты более высокого порядка точности.

*Метод Рунге-Кутты третьего порядка точности* состоит в последовательном вычислении следующих величин для вычисления значения  $y_{i+1}$  через значение  $y_i$ .

$$\begin{aligned} K_1^i &= hf(x_i, y_i), \\ K_2^i &= hf(x_i + 0.5h, y_i + 0.5K_1^i), \\ K_3^i &= hf(x_i + h, y_i - K_1^i + 2K_2^i), \\ \Delta y_i &= \frac{1}{6}(K_1^i + 4K_2^i + K_3^i) \\ y_{i+1} &= y_i + \Delta y_i \end{aligned}$$

*Метод Рунге-Кутты четвертого порядка точности* состоит в следующей последовательности вычислений.

$$\begin{aligned} K_1^i &= hf(x_i, y_i), \\ K_2^i &= hf(x_i + 0.5h, y_i + 0.5K_1^i), \\ K_3^i &= hf(x_i + 0.5h, y_i + 0.5K_2^i), \\ K_4^i &= hf(x_i + h, y_i + K_3^i), \quad h = const, \\ \Delta y_i &= \frac{1}{6}(K_1^i + 2K_2^i + 2K_3^i + K_4^i), \\ y_{i+1} &= y_i + \Delta y_i \end{aligned}$$

Оценка погрешности производится двойным пересчетом, т.е. сначала вычисляется решение с шагом  $h$ , затем с шагом  $h/2$ . Погрешность определяется аналогично (10.2).

#### 10.4 $m$ - шаговый метод Адамса.

При решении задачи Коши (10.1)  $m$  - шаговым методом Адамса считается, что найдены приближенные значения  $y_{n-1}, y_{n-2}, \dots, y_{n-m}$  решения задачи

Коши в точках  $x_{n-1}, x_{n-2}, \dots, x_{n-m}$  равномерной сетки с шагом  $h$ . Приближенное значение  $y_n = y(x_n)$  в точке  $x_n$  выражается через них по следующей формуле

$$\frac{y_n - y_{n-1}}{\tau} = b_0 f_n + b_1 f_{n-1} + \dots + b_m f_{n-m}, \quad n = m, m+1, \dots,$$

где  $b_k$  - числовые коэффициенты, не зависящие от  $n$ ,  $f_k = f(x_k, y_k)$ . Для начала расчета необходимо задать  $m$  начальных значений  $y_0, y_1, \dots, y_{m-1}$ , которые могут быть вычислены методом Рунге-Кутты или Эйлера. Это "начало решения" должно быть вычислено с большей точностью по сравнению с основной задачей. Расчет начинается с  $n = m$ , то есть с уравнения

$$\frac{y_m - y_{m-1}}{\tau} = b_0 f_m + b_1 f_{m-1} + \dots + b_m f_0.$$

Если  $b_0 = 0$  метод Адамса называется *явным*, в противном случае - *неявным*. Наивысший порядок аппроксимации  $m$ -шагового метода Адамса равен  $m+1$ , а явного метода Адамса -  $m$ .

Приведем примеры методов  $m$ -го порядка аппроксимации.

$$\frac{y_n - y_{n-1}}{h} = \frac{3}{2} f_{n-1} - \frac{1}{2} f_{n-2}, \quad (m = 2),$$

$$\frac{y_n - y_{n-1}}{h} = \frac{1}{12} (23f_{n-1} - 16f_{n-2} + 5f_{n-3}), \quad (m = 3),$$

$$\frac{y_n - y_{n-1}}{h} = \frac{1}{24} (55f_{n-1} - 59f_{n-2} + 37f_{n-3} - 9f_{n-4}), \quad (m = 4),$$

$$\frac{y_n - y_{n-1}}{h} = \frac{1}{720} (1901f_{n-1} - 2774f_{n-2} + 2616f_{n-3} - 1274f_{n-4} + 251f_{n-5}), \quad (m = 5),$$

$$\frac{y_n - y_{n-1}}{h} = \frac{1}{12} (5f_n + 8f_{n-1} - f_{n-2}), \quad (m = 3),$$

$$\frac{y_n - y_{n-1}}{h} = \frac{1}{24} (9f_n + 19f_{n-1} - 5f_{n-2} + f_{n-3}), \quad (m = 4),$$

$$\frac{y_n - y_{n-1}}{h} = \frac{1}{720} (251f_n + 646f_{n-1} - 264f_{n-2} + 106f_{n-3} - 19f_{n-4}), \quad (m = 5).$$

Последние три метода являются неявными и содержат искомое значение нелинейно, поэтому для их реализации необходимо применять итерационные методы.

## 11 Лабораторная работа.

### Задача Коши для систем обыкновенных дифференциальных уравнений.

Задача Коши для системы обыкновенных дифференциальных уравнений

$$\begin{aligned} y' &= f(x, y), \quad a \leq x \leq b, \quad y = (y_1, \dots, y_m)^T, \quad f = (f_1, \dots, f_m)^T \\ y(a) &= y_0 = (y_1^0, \dots, y_m^0)^T \end{aligned}$$

решается теми же методами, что и задача Коши для одного уравнения: методами *Эйлера*, *Эйлера-Коши*, *Рунге-Кутта*. При этом все равенства следует считать векторными. Константы  $K_j^i$  в методе Рунге-Кутта также являются векторами длины  $m$ .

Рассмотрим, например, метод Эйлера для системы двух дифференциальных уравнений первого порядка

$$\begin{cases} y' = f_1(x, y, z), & a \leq x \leq b, \\ z' = f_2(x, y, z), \end{cases}$$

с условиями  $y(a) = y_0$ ,  $z(a) = z_0$ . Приближенные решения  $y_i \approx y(x_i)$ ,  $z_i \approx z(x_i)$  этой системы в точках  $x_{i+1}$  вычисляются последовательно по формулам

$$\begin{cases} y_{i+1} = y_i + h_i f_1(x_i, y_i, z_i), & i = \overline{0, N-1}, \\ z_{i+1} = z_i + h_i f_2(x_i, y_i, z_i). \end{cases}$$

## 12 Лабораторная работа.

### Метод сеток решения граничных задач для обыкновенных дифференциальных уравнений.

Граничная задача для обыкновенного дифференциального уравнения второго порядка ставится следующим образом: найти функцию  $u(x)$ , которая внутри отрезка  $[a, b]$  удовлетворяет уравнению

$$u''(x) + p(x)u'(x) - q(x)u(x) = f(x), \quad (12.50)$$

а на концах отрезка - граничным условиям

$$\beta_0 u'(a) + \alpha_0 u(a) = \gamma_0, \quad (12.51)$$

$$\beta_1 u'(b) + \alpha_1 u(b) = \gamma_1, \quad (12.52)$$

где  $p(x), q(x), f(x)$  - известные функции;  $\alpha_i, \beta_i, \gamma_i$ , ( $i = 1, 2$ ) - заданные постоянные.

Метод сеток решения граничной задачи (12.1) - (12.3) состоит в следующем:

1. На отрезке  $[a, b]$  выбирается сетка, то есть конечное множество точек:

$$a = x_0 < x_1 < \dots < x_N = b.$$

Часто сетка выбирается равномерной, то есть

$$x_i = a + ih, \quad i = 0, 1, \dots, N, \quad h = (b - a)/N.$$

2. Граничная задача (12.1) - (12.3) заменяется системой алгебраических уравнений, неизвестные в которой являются приближенными значениями решения граничной задачи в узлах сетки. Эту систему алгебраических уравнений называют *разностной схемой*.

3. Находится решение разностной схемы и тем самым определяется приближенное решение граничной задачи.

Простейшим способом построения разностной схемы является замена производных в узлах сетки разностными производными. Часто используются следующие разностные производные первого и второго порядка для сеточной функции  $y_i = y(x_i)$  :

$$\begin{aligned} y'(x_i) &\approx y_{x,i} = \frac{y_{i+1} - y_i}{h}, \\ y'(x_i) &\approx y_{\bar{x},i} = \frac{y_i - y_{i-1}}{h}, \\ y'(x_i) &\approx y_{x^\circ,i} = \frac{y_{i+1} - y_{i-1}}{2h}, \\ y''(x_i) &\approx y_{\bar{x}x,i} = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}. \end{aligned}$$

Полагая в уравнении (12.1)  $x = x_i$  и заменяя  $u(x_i), u'(x_i), u''(x_i)$  на  $y_i, y_{x,i}, y_{\bar{x}x,i}$  соответственно, получим разностное уравнение относительно сеточной функции  $y_i$ :

$$a_i y_{i+1} + b_i y_i + c_i y_{i-1} = h^2 f(x_i), \quad i = 1, \dots, N - 1, \quad (12.53)$$

где  $a_i = 1 + \frac{h}{2}p(x_i)$ ,  $b_i = -(2 + h^2q(x_i))$ ,  $c_i = 1 - \frac{h}{2}p(x_i)$ .

Заменяя в граничных условиях  $u(a), u(b)$  на  $y_0, y_N$ , а  $u'(a), u'(b)$  на  $y_{x,0}, y_{x,N}$  получим разностные граничные условия

$$\left(\alpha_0 - \frac{\beta_0}{h}\right)y_0 + \frac{\beta_0}{h}y_1 = \gamma_0, \quad -\frac{\beta_1}{h}y_{N-1} + \left(\alpha_1 + \frac{\beta_1}{h}\right)y_N = \gamma_1. \quad (12.54)$$

Разностная схема (12.4)-(12.5) имеет первый порядок аппроксимации из-за первого порядка аппроксимации производных в граничных условиях. Системе (12.4)-(12.5) можно решать методом прогонки, проверив предварительно условия сходимости.

Заменяя в граничных условиях (12.2)-(12.3)

$$u'(a) \approx \frac{-y_2 + 4y_1 - 3y_0}{2h},$$

$$u'(b) \approx \frac{3y_N - 4y_{N-1} + y_{N-2}}{2h},$$

получим разностную схему, имеющую второй порядок аппроксимации. Эту разностную схему можно преобразовать к трехдиагональному виду и решать методом прогонки в случае выполнения условий сходимости.

Если наложить условия

$$q(x) \geq 0, \quad x \in [a, b]; \quad \alpha_0 \beta_0 \leq 0, \quad \alpha_1 \beta_1 \geq 0,$$

то порядок точности разностной схемы будет совпадать с порядком аппроксимации.

Один из способов построения разностных схем носит название *интегроинтерполяционного метода*. Например для граничной задачи

$$(k(x)u'(x))' - q(x)u(x) + f(x) = 0, \quad 0 < x < l, \quad (12.55)$$

$$-k(0)u'(0) + \beta u(0) = \mu_1, \quad u(l) = \mu_2, \quad (12.56)$$

где  $k(x) > 0$ ,  $q(x) \geq 0$ ,  $\beta \geq 0$ ,  $\mu_1, \mu_2 \in R$ , этот метод дает разностную схему

$$\frac{a_{i+1}y_{x,i} - a_i y_{\bar{x},i}}{h} - d_i y_i + \phi_i = 0, \quad i = 1, \dots, N-1, \quad (12.57)$$

$$-a_1 y_{x,0} + (\beta + 0.5d_0)y_0 = \mu_1 + 0.5h\phi_0, \quad y_N = \mu_2, \quad (12.58)$$

где коэффициенты выражаются через интегралы от функций, участвующих в исходной дифференциальной задаче. Если применить к этим интегралам формулы прямоугольников и трапеций, то получим следующие два варианта формул для коэффициентов:

$$(a) \quad a_i = k(x_i - 0.5h), \quad d_i = q(x_i), \phi_i = f(x_i);$$

$$(b) \quad a_i = \frac{2k(x_i)k(x_{i-1})}{k(x_i) + k(x_{i-1})}, \quad d_i = \frac{q(x_{i-1/2}) + q(x_{i+1/2})}{2},$$

$$\phi_i = \frac{f(x_{i-1/2}) + f(x_{i+1/2})}{2} \quad (i \geq 1), \quad d_0 = q(a), \quad \phi_0 = f(a).$$

Полученная таким способом разностная схема имеет второй порядок аппроксимации и второй порядок точности.

Если первое из граничных условий (12.7) имеет вид  $u(a) = \mu_1$ , то граничные условия разностной схемы будут иметь вид

$$y_0 = \mu_1, \quad y_N = \mu_2. \quad (12.59)$$

Разностная схемы (12.8)-(12.9) и (12.8),(12.10) решаются методом прогонки. Действительно, разностную схему (12.8)-(12.9) можно записать в виде

$$\begin{aligned} A_i y_{i-1} - C_i y_i + B_{i+1} &= -F_i, \quad i = 1, 2, \dots, N-1, \\ y_0 &= \kappa_1 y_1 + \tilde{\mu}_1, \quad y_N = \kappa_2 y_2 + \mu_2, \end{aligned}$$

где

$$\begin{aligned} A_i &= a_i, \quad B_i = a_{i+1}, \quad C_i = a_i + a_{i+1} + h^2 d_i, \quad F_i = h^2 \phi_i, \quad \kappa_2 = 0, \\ \kappa_1 &= \frac{1}{1 + ha_1^{-1}(\beta + 0.5hd_0)}, \quad \tilde{\mu}_1 = \frac{h(\mu_1 + 0.5h\phi_0)}{a_1}. \end{aligned}$$

Нетрудно проверить, что условия устойчивости прогонки в данном случае выполнены.

## Список литературы

- [1] Самарский А.А., Гулин А.В. Численные методы. – М., Наука, 1989.
- [2] Бахвалов Н.С., Численные методы. – М., Наука, 1973.
- [3] Сборник задач по методам вычислений. Под ред. Монастырного П.И. – Минск: БГУ, 1983.