

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение  
высшего профессионального образования  
«Томский государственный университет систем управления и  
радиоэлектроники»

Кафедра электронных приборов

## **КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И ПРОЕКТИРОВАНИЕ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ. Часть 1**

Методические указания к лабораторным работам  
для студентов направления «Фотоника и оптоинформатика» и  
«Электроника и микроэлектроника»  
(специальность «Электронные приборы и устройства»)

## **Саликаев Юрий Рафаельевич**

Компьютерное моделирование и проектирование. Лабораторный практикум. Часть 1: Методические указания к лабораторным работам для студентов направления «Фотоника и оптоинформатика» и направлению «Электроника и микроэлектроника» (специальность 210105 – Электронные приборы и устройства) / Ю.Р. Саликаев; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Томский государственный университет систем управления и радиоэлектроники, Кафедра электронных приборов. - Томск : ТУСУР, 2012. – 40 с.

В методических указаниях приведено описание семи лабораторных работ, охватывающих материал по изучению компьютерного моделирования и проектирования. Работы № 1, 3 и 6 выполняются с использованием программной системы MathCad. Работы № 2, 4, 5 и 7 предполагают написание студентом программ на алгоритмическом языке Pascal.

Предназначено для студентов очной и заочной форм, обучающихся по направлению «Фотоника и оптоинформатика» по дисциплине «Компьютерное моделирование и проектирование оптических систем» и по направлению «Электроника и микроэлектроника» (специальность 210105 – Электронные приборы и устройства) по дисциплине «Компьютерное моделирование и проектирование электронных приборов и устройств»

© Саликаев Юрий Рафаельевич, 2012

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Томский государственный университет систем управления и  
радиоэлектроники»

Кафедра электронных приборов

УТВЕРЖДАЮ  
Зав.кафедрой ЭП  
\_\_\_\_\_ С.М. Шандаров  
« \_\_\_\_ » \_\_\_\_\_ 2012 г.

**КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ  
И ПРОЕКТИРОВАНИЕ.  
ЛАБОРАТОРНЫЙ ПРАКТИКУМ. Часть 1**

Методические указания к лабораторным работам  
для студентов направления «Фотоника и оптоинформатика» и  
«Электроника и микроэлектроника»

Разработчик

\_\_\_\_\_ Ю.Р. Саликаев  
« \_\_\_\_ » \_\_\_\_\_ 2012 г

## Содержание

Введение .....	5
Лабораторная работа № 1. Интерполяция и аппроксимация функций .....	6
1.1 Введение .....	6
1.2 Теоретическая часть .....	6
1.3 Контрольные вопросы .....	8
1.4 Порядок проведения вычислительных экспериментов .....	9
Лабораторная работа № 2. Методы вычисления определенного интеграла .....	11
2.1 Введение .....	11
2.2 Теоретическая часть .....	11
2.3 Контрольные вопросы .....	13
2.4 Порядок проведения вычислительных экспериментов .....	13
Лабораторная работа № 3. Разностная аппроксимация производных .....	15
3.1 Введение .....	15
3.2 Теоретическая часть .....	15
3.3. Контрольные вопросы .....	16
3.4 Порядок проведения вычислительных экспериментов .....	16
Лабораторная работа № 4. Методы решения систем линейных алгебраических уравнений (СЛАУ) .....	17
4.1 Введение .....	17
4.2 Теоретическая часть .....	17
4.3 Контрольные вопросы .....	20
4.4 Порядок проведения вычислительных экспериментов .....	21
Лабораторная работа № 5. Методы решения нелинейных уравнений .....	22
5.1 Введение .....	22
5.2 Теоретическая часть .....	22
5.3 Контрольные вопросы .....	25
5.4 Порядок проведения вычислительных экспериментов .....	25
Лабораторная работа № 6. Методы решения краевых задач .....	26
6.1 Введение .....	26
6.2 Теоретическая часть .....	26
6.3 Контрольные вопросы .....	30
6.4 Порядок проведения вычислительных экспериментов .....	31
Лабораторная работа № 7. Методы безусловной оптимизации .....	32
7.1 Введение .....	32
7.2 Теоретическая часть .....	32
7.3 Контрольные вопросы .....	37
7.4 Порядок проведения вычислительных экспериментов .....	38
Список рекомендуемой литературы .....	38

## ВВЕДЕНИЕ

В методических указаниях приведено описание семи лабораторных работ, охватывающих материал по изучению компьютерного моделирования и проектирования.

Работы № 1, 3 и 6 выполняются с использованием программной системы MathCad.

Работы № 2, 4, 5 и 7 предполагают написание студентом программ на алгоритмическом языке Pascal.

По каждой из предложенных лабораторных работ необходимо составить отчет, который должен содержать:

- титульный лист;
- цель работы;
- краткие сведения из теории, содержащие расчетные формулы;
- блок-схемы программ;
- результаты расчетов и экспериментов в виде таблиц и графиков;
- ответы на контрольные вопросы;
- выводы по проведенной работе.

## Лабораторная работа № 1. Интерполяция и аппроксимация функций

### 1.1 Введение

Цель лабораторной работы - изучение и сравнение различных способов обработки экспериментальных данных представленных таблично.

### 1.2 Теоретическая часть

Пусть  $y=f(x)$  – некоторая функция, для которой известны лишь табличные значения, т.е. известно, что при значениях аргумента  $x = x_0, x_1, \dots, x_n$  функция принимает значения  $y = y_0, y_1, \dots, y_n$ :

$$\begin{cases} f(x_0) = y_0 \\ f(x_1) = y_1 \\ \dots \\ f(x_n) = y_n \end{cases} \quad (1.1)$$

Функция типа (1) может являться результатом проведения измерительного либо вычислительного эксперимента, проведенного с некоторым дискретным шагом.

Возникает задача нахождения промежуточных значений функции и отыскания простого аналитического представления функции. Если на искомую функцию накладываются условия ее точного совпадения с заданными значениями в исходных точках, называемых узловыми, то различают задачу интерполяции

Очевидно, что через заданные точки можно провести бесконечно множество различных кривых  $\phi(x)$ . Чаще всего требуется, чтобы  $\phi(x)$  была многочленом, однозначное определение коэффициентов которого требует, чтобы его степень  $m$  была равна  $n$ . В этом случае говорят о глобальной интерполяции, поскольку один многочлен

$$\phi(x_i) = a_0 + a_1x + \dots + a_nx^n \quad (1.2)$$

используется для интерполяции функции  $f(x)$  на всем рассматриваемом интервале изменения аргумента  $x$ . Коэффициенты  $a_j$  многочлена (2) находятся из условия прохождения функции  $\phi(x)$  через узловые точки. Такой путь построения интерполяционного многочлена требует значительного объема вычислений, особенно при большом числе узлов. Существуют более экономичные алгоритмы построения интерполяционные многочленов. Формула

$$L(x) = \sum_{j=0}^n y_j \prod_{\substack{i=0, \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i} = \sum_{i=0}^n y_i \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \quad (1.3)$$

называется *интерполяционным многочленом Лагранжа*, а

$$N(x) = y_0 + \sum_{i=1}^n \frac{\Delta^i y_0}{i! h^i} \prod_{j=1}^i (x - x_{j-1}) = y_0 + \frac{\Delta y_0}{1! h} (x - x_0) + \dots + \frac{\Delta^n y_0}{n! h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (1.4)$$

- первой интерполяционной формулой Ньютона для равноотстоящих узлов, где

$$\Delta^k y_0 = \sum_{i=0}^k \left[ (-1)^{k-i} \frac{k!}{(k-i)! i!} y_i \right] = y_k - k y_{k-1} + \frac{k(k-1)}{2!} y_{k-2} - \dots + (-1)^{k-1} k y_1 + (-1)^k y_0 \quad (1.5)$$

- конечные *разности порядка k*.

Разные способы построения канонического многочлена, Лагранжа и Ньютона дают тождественные интерполяционные формулы при заданной таблице значений функции. Принципиальное различие формул Ньютона и Лагранжа заключается в том, что в формуле Лагранжа каждое из слагаемых представляет собой многочлен  $n$ -й степени, и все слагаемые равноправны и поэтому нельзя пренебрегать ни одним из слагаемых. В формулу же Ньютона в качестве слагаемых входят многочлены увеличивающихся степеней. Ввиду этого существует возможность не учитывать слагаемые более высоких степеней, которые вносят незначительный вклад в сумму.

При большом количестве узлов интерполяции получается высокая степень многочлена (2), что часто приводит к плохому приближению из-за сильного накопления погрешностей в процессе вычислений. Кроме того, из-за расходимости процесса интерполяции увеличение числа узлов не обязательно приводит к повышению точности.

Интерполяционные многочлены могут строиться отдельно для разных частей рассматриваемого интервала изменения  $x$ . В этом случае имеем *кусочную* (или *локальную*) интерполяцию. Одним из таких способов является линейная интерполяция, когда на каждом из отрезков строится полином первой степени и точки соединяются прямыми линиями, что в результате дает ломанную. Другим, дающим более удобоваримый результат, способом является интерполирование с помощью сплайн-функций.

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

а) на каждом сегменте  $[x_{i-1}, x_i], i = 1, 2, \dots, N$ , функция  $s(x)$  является многочленом третьей степени;

б) функция  $s(x)$ , а также ее первая и вторая производные непрерывны на  $[a, b]$

в)  $s(x_i) = f(x_i), i = 0, 1, \dots, N$ .

Сплайн, определяемый условиями а) - в), называется также *интерполяционным кубическим сплайном*. Сплайн, определяемый перечисленными условиями, существует и единственен.

Преимуществом сплайнов перед обычной интерполяцией является, во-первых, их сходимость и, во-вторых, устойчивость процесса вычислений.

Кубический сплайн задается в виде

$$s_i(x) = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3, \quad x_{i-1} \leq x \leq x_i, \quad i = 1, 2, \dots, N \quad (1.6)$$

где коэффициенты

$$a_i = f(x_i), \quad i = 1, 2, \dots, N, \quad (1.7)$$

а  $c_i$  определяются методом прогонки из системы уравнений

$$h_i c_{i-1} + 2(h_i + h_{i+1})c_i + h_{i+1}c_{i+1} = 6 \left( \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right), \quad (1.8)$$

$$i = 1, 2, \dots, N-1, \quad c_0 = c_N = 0.$$

По найденным коэффициентам  $c_i$  коэффициенты  $b_i$  и  $d_i$  определяются с помощью явных формул

$$d_i = \frac{c_i - c_{i-1}}{h_i}, \quad b_i = \frac{h_i}{2}c_i - \frac{h_i^2}{6}d_i + \frac{f_i - f_{i-1}}{h_i}, \quad i = 1, 2, \dots, N. \quad (1.9)$$

Итак, при интерполяции основным условием является прохождение графика интерполяционного многочлена через данные значения функции в узлах интерполяции. Однако в ряде случаев выполнение этого условия затруднительно или даже нецелесообразно, например, если исходная таблица значений функции получена в результате эксперимента и неизбежно уже содержит измерительную погрешность. Если не требуется строгого выполнения условия прохождения функции через все узловые точки, а приоритет отдается простой записи функции, то для такого приближения используют аппроксимацию - наилучшее приближение функции, заданной таблично.

В качестве аппроксимирующей функции  $\phi(x)$  можно выбрать любую, качественное поведение которой напоминает  $f(x)$ . Разности

$$r_k = \phi(x_k) - f(x_k), \quad k = 0, 1, \dots, m, \quad (1.10)$$

характеризуют отклонение в узлах  $x_k$  точного значения функции  $f(x)$  от ее приближенного значения  $\phi(x)$  и образуют вектор погрешности  $\mathbf{r}$ . Для него можно ввести ту или иную норму, например,

$$\|\mathbf{r}\| = \left( \sum_{k=0}^m r_k^2 \right)^{1/2} = \left( \sum_{k=0}^m (\phi(x_k) - f(x_k))^2 \right)^{1/2} \quad (1.11)$$

или

$$\|\mathbf{r}\| = \max_{0 \leq k \leq m} |r_k| = \max_{0 \leq k \leq m} |\phi(x_k) - f(x_k)| \quad (1.12)$$

Задача о наилучшем приближении функции  $f(x)$  заданной таблично, состоит в нахождении коэффициентов  $c_0, c_1, \dots, c_n$ , минимизирующих норму вектора  $\mathbf{r}$ . В зависимости от выбора нормы получим различные задачи. Так, норме (1.11) соответствует задача о наилучшем среднеквадратичном приближении, а норме (1.12) - задача о наилучшем равномерном приближении функции, заданной таблично.

### 1.3 Контрольные вопросы

1. Дайте определение интерполяции.
2. Какая интерполяция называется глобальной и какая локальной?



3. В случае, когда интерполируется одна и та же функция  $f(x)$ , но число узлов интерполяции постепенно увеличивается какой из способов глобальной интерполяции уместнее применять?

4. Если узлы интерполяции фиксированы и интерполируется не одна, а несколько функций, то каким из способов глобальной интерполяции удобнее пользоваться?

5. Что такое сплайн?

6. Каковы преимущества сплайнов перед глобальной интерполяцией?

7. Дайте определение аппроксимации.

8. Запишите нормы для вектора погрешностей, соответствующие наилучшим среднеквадратичному и равномерному приближениям функции, заданной таблично.

9. Если в качестве аппроксимирующей функции используется полином, то в чем его принципиальное отличие от полинома, используемого для глобальной интерполяции?

#### **1.4 Порядок проведения вычислительных экспериментов**

1. Задание следует выполнять в программной системе MathCad. Необходимо определить конечные разности  $n$ -порядка, интерполяционные многочлены Ньютона и Лагранжа.

2. Для соответствующего варианта данных провести линейную аппроксимацию; аппроксимацию другими наиболее подходящими функциями исходя из критериев наилучших среднеквадратичного и равномерного приближений; провести линейную интерполяцию; интерполяцию кубическим сплайном; многочленом Лагранжа – для данных таблицы 1.1; многочленом Ньютона второй, четвертой и шестой степени – для данных таблицы 1.2. Полученные функции представить совместно с исходными данными на двух графиках.

Таблица 1.1 - Исходные данные для интерполяции многочленом Лагранжа

Первая строка - это значения аргумента  $x_0 - x_9$ .

В каждой строке 1 - 25 - значения функции ( $y_0 - y_9$ ) для соответствующего массива значений  $x_0 - x_9$ .

Таблица 1.2 - Исходные данные для интерполяции многочленом Ньютона

Первая строка - это значения аргумента  $x_0 - x_9$ .

В каждой строке 1 - 25 - значения функции ( $y_0 - y_9$ ) для соответствующего массива значений  $x_0 - x_9$

## Лабораторная работа № 2. Методы вычисления определенного интеграла

### 2.1 Введение

Целью данной работы являются: вычисление определенного интеграла численными методами; оценка возникающей при этом погрешности; применение алгоритмов уменьшения погрешности.

### 2.2 Теоретическая часть

Численное вычисление определенных интегралов

$$I = \int_a^b f(x) dx \quad (2.1)$$

основано на замене интеграла конечной суммой, называемой квадратурной формулой. В зависимости от вида последней различают несколько методов. Геометрическая интерпретация определенного интеграла – это площадь, ограниченная подынтегральной функцией и осью абсцисс, а также прямыми  $x=a$  и  $x=b$ . Разбивая интервал от  $a$  до  $b$  на множество  $N$  равных частей, подынтегральную функцию на каждом из них приближенно заменяют константой; прямой; параболой.

Методу прямоугольников соответствует квадратурная формула

$$\int_a^b f(x) dx \approx \sum_{i=1}^N f(x_{i-1/2})h. \quad (2.2)$$

Ее погрешность определяется формулой

$$|\Psi| \leq \frac{M_2 N h^3}{24} = \frac{h^2 (b-a) M_2}{24}, \quad h = \frac{b-a}{N}, \quad M_2 = \max_{x \in [a,b]} |f''(x)|. \quad (2.3)$$

т. е. величиной  $O(h^2)$ . В этом случае говорят, что квадратурная формула имеет второй *порядок точности*.

Методу трапеций соответствует квадратурная формула

$$\int_a^b f(x) dx \approx \sum_{i=1}^N \frac{f(x_i) + f(x_{i-1})}{2} h = h(0.5f_0 + f_1 + \dots + f_{n-1} + 0.5f_N), \quad (2.4)$$

Погрешность этой формулы имеет тот же второй порядок точности и оценивается величиной в два раза большей (15).

Методу Симпсона соответствует квадратурная формула

$$\int_a^b f(x) dx \approx \frac{b-a}{12} [f_0 + f_{2N} + 4(f_1 + f_3 + \dots + f_{2N-1}) + 2(f_2 + f_4 + \dots + f_{2N-2})]. \quad (2.5)$$

Ее погрешность оценивается как:

$$|\Psi| \leq \frac{h^4 (b-a)}{2880} M_4, \quad M_4 = \sup_{x \in [a,b]} |f^{IV}(x)|. \quad (2.6)$$

Таким образом, формула Симпсона существенно точнее, чем формулы прямоугольников и трапеций; она имеет четвертый порядок точности.

Для всех трех методов приведены оценки погрешности. Может оказаться, что величина погрешности велика, а уменьшать ее путем уменьшения шага не всегда уместно, т.к. при этом пропорционально возрастает объем вычислений.

Известный математик Рунге показал путь оценки возникающей вычислительной погрешности путем двойного просчета интеграла с шагами  $h$  и  $kh$ . Это позволяет сделать формула для главного члена погрешности

$$R_0 = \frac{I_h - I_{kh}}{k^p - 1}, \quad (2.7)$$

которая называемая первой формулой Рунге. Здесь  $p$  – порядок точности метода. Формула (19) имеет большое практическое значение, так как позволяет провести апостериорную оценку погрешности без изменения алгоритма используемого вычислительного процесса. После определения  $R_0$  можно вычислить уточненное значение искомой величины

$$I_{\text{уточн}} = I_h + R_0 + O(h^{p+1}), \quad (2.8)$$

с погрешностью на один порядок меньше, чем  $R_0$ . Последнее соотношение называют второй формулой Рунге.

Порядок точности метода  $P$  можно также вычислить апостериорно. Для этого необходимо в третий раз вычислить значение интеграла с шагом  $k^2h$  и воспользоваться формулой Эйткена

$$p = \ln \left( \frac{I_{kh} - I_{k^2h}}{I_h - I_{kh}} \right) / \ln k. \quad (2.9)$$

С определенной долей погрешности априорный и апостериорный порядок точности метода должны совпадать.

Возможность апостериорно оценивать погрешность позволяет вычислять интеграл (2.1) с заданной точностью  $\varepsilon > 0$  путем автоматического выбора шага интегрирования  $h_i$ . Пусть используется квадратурная формула, например, прямоугольников (2.2). Проведем на каждом частичном отрезке  $[x_{i-1}, x_i]$  суммы в (2.2) все вычисления дважды, один раз с шагом  $h_i$  и второй раз – с шагом  $0,5h_i$  и оценим погрешность по первой формуле Рунге (19). Если для заданного  $\varepsilon > 0$  будут выполняться неравенства

$$R_0 \leq \frac{\varepsilon h_i}{b-a}, \quad i=1,2,\dots,N, \quad (2.10)$$

то будет достигнута заданная точность  $\varepsilon$ . Если же на каком-то из частичных отрезков оценка (2.10) не будет выполняться, то шаг на этом отрезке надо измельчить еще в два раза и снова оценить погрешность. Измельчение сетки на данном отрезке следует проводить до тех пор, пока не будет достигнута оценка вида (2.10). Заметим, что для некоторых функций  $f(x)$  такое измельчение может продолжаться слишком долго, поэтому в соответствующей программе следует предусмотреть ограничение сверху на число измельчений, а так же возможность увеличения  $\varepsilon$ .

Таким образом, автоматический выбор шага интегрирования приводит к тому, что интегрирование ведется с крупным шагом на участках плавного изменения функции  $f(x)$  и с мелким шагом – на участках быстрого изменения  $f(x)$ . Это позволяет при заданной точности  $\varepsilon$  уменьшить количество вычислений значений  $f(x)$  по сравнению с расчетом на сетке с постоянным шагом.

### 2.3 Контрольные вопросы

1. Какой зависимостью заменяется подынтегральная функция в методе Симпсона?
2. Какой порядок точности имеет формула прямоугольников?
3. Какой порядок точности имеет формула трапеций?
4. Какой порядок точности имеет формула Симпсона?
5. Как соотносятся погрешности формул прямоугольников и трапеций?
6. Какая величина вычисляется по первой формуле Рунге?
7. Какая величина вычисляется по второй формуле Рунге?
8. Какая величина вычисляется по формуле Эйткена?
9. Назовите антоним понятия априорный. Раскройте оба понятия.
10. Какую формулу можно использовать для тестирования программ, реализующих вычислительные методы с известной априорной погрешностью.
11. Какая формула позволяет вычислять определенный интеграл с заданной точностью  $\varepsilon > 0$  путем автоматического выбора переменного шага интегрирования?

### 2.4 Порядок проведения вычислительных экспериментов

1. На алгоритмическом языке Pascal написать процедуры, реализующие методы прямоугольников, трапеций и Симпсона. Все необходимые для работы процедур данные передавать через список формальных параметров. Создать функцию соответствующую выбранному варианту определенного интеграла.
2. Объединить созданные процедуры и функцию в программу, реализующую вычисление определенного интеграла с числом разбиений интервала интегрирования на 10, 20 и 40 частей. Сравнить полученные результаты с точным значением интеграла, вычисленным аналитически, сделать выводы о точности методов и влиянии числа разбиений на погрешность.
3. При разбиении интервала интегрирования на 10 частей применить в данной программе формулы Рунге и Эйткена для апостериорной оценки погрешности вычислений, уточнения значения интеграла и оценки порядка точности всех методов. Сравнить полученную погрешность с теоретической и с полученной в эксперименте 2. Сравнить уточненное значение интеграла с точным значением интеграла, вычисленным аналитически. Сравнить

вычисленный апостериорно порядок точности методов с теоретическим. Сделать выводы.

4. В последней части программы вычислить интеграл (2.1) с заданной точностью  $\varepsilon = 10^{-6}$  путем автоматического выбора шага интегрирования.

### Варианты заданий

$$1. \int_1^{10} 3x^2 + x^3 dx$$

$$2. \int_2^7 x^4 + x^5 dx$$

$$3. \int_0^5 x^3 + x^4 dx$$

$$4. \int_3^7 \frac{4}{3}x^3 + \frac{5}{2}x^4 dx$$

$$5. \int_{-1}^3 \frac{x^2}{3} + x dx$$

$$6. \int_{-5}^5 x^3 dx$$

$$7. \int_{-4}^5 \frac{x^5}{4} + \frac{x^4}{5} dx$$

$$8. \int_0^4 x^2 - \frac{x^3}{4} dx$$

$$9. \int_{0.1}^{0.2} x^5 + x dx$$

$$10. \int_{-0.5}^{0.5} x - \frac{x^2}{3} dx$$

$$11. \int_0^2 \frac{x^6 - x^2}{x} dx$$

$$12. \int_{-1}^3 \frac{x^2 - 1}{x + 1} dx$$

$$13. \int_2^{-2} x^3 - 8 dx$$

$$14. \int_3^5 \frac{x^2 - 1}{x + 1} dx$$

$$15. \int_{-1}^3 x^3 - \frac{x^2}{4} dx$$

$$16. \int_{-1}^1 \frac{1}{2}x^3 dx$$

$$17. \int_{-2}^2 3x^4 + 4x^5 dx$$

$$18. \int_0^5 \frac{3}{x^2} + x^2 dx$$

$$19. \int_{-3}^4 x^2 dx$$

$$20. \int_0^1 x^3 + x^2 dx$$

$$21. \int_0^{2\pi} \sin\left(\frac{x}{2}\right) dx = 4$$

$$22. \int_0^{\frac{\pi}{4}} \frac{1}{\cos(x)^2(1 + \tan(x))} dx$$

$$23. \int_0^1 \frac{e^{3x}}{1 + e^{3x}} dx = 0.785$$

$$24. \int_0^{\frac{\pi}{4}} \operatorname{tg}(x) dx$$

$$25. \int_0^2 \frac{e^x}{1 + e^{2x}} dx$$

$$26. \int_0^{2\pi} \cos(4x) dx = 0$$

$$27. \int_0^{\pi} x \cdot \sin(x^2) dx = 0.951$$

$$28. \int_0^{2\pi} -2x \cdot \cos(x^2) dx$$

$$29. \int_0^{\pi} e^x \cdot \sin(x) dx$$

$$30. \int_0^{\pi} e^x \cdot \cos(x) dx$$

$$31. \int_{-2}^2 e^x x^2 dx = 13.425$$

$$32. \int_{-1}^1 e^x + \sin(2x) dx$$

$$33. \int_{-1}^1 e^{2x} \cos(2x) dx$$

$$34. \int_{-0.5}^2 e^x + \cos(2x) dx$$

$$35. \int_0^1 e^x + e \cdot \cos(2x) dx$$

$$36. \int_{\frac{\pi}{3}}^{\frac{\pi}{6}} \frac{\cos(x)}{\sin(x)} dx$$

$$37. \int_0^{2\pi} \sin(4x) dx$$

$$38. \int_0^{2\pi} \cos(4x) dx$$

$$39. \int_0^{\pi} \cos(6x) - x dx$$

$$40. \int_0^{\pi} \sin(6x) + x^2 dx$$

## Лабораторная работа № 3. Разностная аппроксимация производных

### 3.1 Введение

Целью данной работы являются: вычисление производной численными методами, оценка возникающей при этом погрешности, применение алгоритмов уменьшения погрешности.

### 3.2 Теоретическая часть

Численное вычисление производных основано на введении на отрезке  $[a, b]$  сетки и замене на ней дифференциала конечной разностью, которая использует значения сеточной функции слева и справа от рассматриваемой точки  $x = x_i$ :

$$\begin{aligned} u_i &= y(x_i), & u_{x,i}^- &= (u_i - u_{i-1})/h, \\ u_{x,i} &= (u_{i+1} - u_i)/h, & u_{x,i}^o &= (u_{i+1} - u_{i-1})/2h. \end{aligned} \quad (3.1)$$

Приведенные здесь разностные отношения называются, соответственно, *левой, правой и центральной разностными производными функции  $u(x)$  в точке  $x = x_i$* . Если точка  $x_i$  фиксирована, а шаг  $h$  стремится к нулю (при этом количество разбиений  $N^i \rightarrow \infty$ ), то каждое из этих разностных соотношений стремится к значению производной функции  $u(x)$  в точке  $x_i$ , поэтому в качестве приближенного значения  $u'(x)$  можно взять любое из них.

Погрешность  $u_{x,i}^- - u'(x_i)$ , возникающая при замене дифференциального выражения  $u'(x)$  разностным выражением  $u_{x,i}^-$ , называется *погрешностью аппроксимации*. Для левой и правой разностных производных погрешность аппроксимации является величиной  $O(h)$  при  $h \rightarrow 0$ . В этом случае говорят, что имеет место *аппроксимация первого порядка*. Центральная разностная производная аппроксимирует  $u'(x)$  со вторым порядком и, следовательно, является более точным приближением к  $u'(x)$ , чем левая или правая разностные производные.

Вторую производную  $u''(x)$  можно приближенно заменить в точке  $x_i \in \omega_h$  *второй разностной производной*

$$u_{xx,i}^- = \frac{1}{h}(u_{x,i} - u_{x,i}^-) = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}. \quad (3.2)$$

Здесь также имеет место аппроксимация второго порядка.

Записывая интерполяционный многочлен Лагранжа  $L(x)$  и его остаточный член  $R_L(x)$  для случая пяти узлов интерполяции ( $n=4$ ) и дифференцируя их соответствующее количество раз, получают следующие выражения для производных функции в центральном узле  $x = x_i$ :

$$y'(x_i) = \frac{1}{12h}(y_{i-2} - 8y_{i-1} + 8y_{i+1} - y_{i+2}) + \frac{h^4}{30}y_*''', \quad (3.3)$$

$$y''(x_i) = \frac{1}{12h^2}(-y_{i-2} + 16y_{i-1} - 30y_i + 16y_{i+1} - y_{i+2}) + O(h^4). \quad (3.4)$$

которые являются формулами повышенного порядка аппроксимации.

Для численного вычисления производной функции на концах отрезка  $[a, b]$  (где не определены, например  $y_{i-2}$  или  $y_{i+1}$ ) можно воспользоваться левой и правой разностными производными соответственно, а также формулами повышенного порядка аппроксимации, которые можно получить, записав продифференцированный многочлен Лагранжа не в центральном, а в левом или правом узле интерполяции.

### 3.3. Контрольные вопросы

1. Какой порядок точности имеют левая и правая разностные производные?
2. Какой порядок точности имеет центральная разностная производная?
3. Какой порядок точности имеет вторая разностная производная?
4. Какой порядок точности имеют пятиузловые разностные производные?
5. Какая величина вычисляется по первой формуле Рунге?
6. Какая величина вычисляется по второй формуле Рунге?
7. Какая величина вычисляется по формуле Эйткена?
8. Назовите антоним понятия априорный. Раскройте оба понятия.
9. Какую формулу можно использовать для тестирования программ, реализующих вычислительные методы с известной априорной погрешностью.

### 3.4 Порядок проведения вычислительных экспериментов

1. В системе Mathcad написать программу, реализующую вычисление первой и второй производных функции всеми рассмотренными методами с числом разбиений отрезка  $[a, b]$  на 10, 20 и 40 частей. Результаты представить графически. В одной из внутренних точек отрезка сравнить полученные результаты с точным значением производной, вычисленным аналитически, сделать выводы о точности методов и влиянии величины шага на погрешность.

2. При разбиении интервала интегрирования на 10 частей применить в данной программе формулы Рунге и Эйткена для апостериорной оценки погрешности вычислений, уточнения значения производной и оценки порядка точности всех методов. Сравнить полученную погрешность с теоретической и с полученной в эксперименте 1. В одной из внутренних точек отрезка сравнить уточненное значение производной с точным значением, вычисленным аналитически. Сравнить вычисленный апостериорно порядок точности методов с теоретическим. Сделать выводы.



## Лабораторная работа № 4. Методы решения систем линейных алгебраических уравнений (СЛАУ)

### 4.1 Введение

Целью данной работы являются: изучение алгоритмов численного решения СЛАУ, реализация методов решения СЛАУ на алгоритмических языках.

### 4.2 Теоретическая часть

Пусть имеется система линейных уравнений

$$Ax = b, \quad (4.1)$$

где  $A$  – матрица размера  $(n \times n)$  с постоянными коэффициентами;  $b$  –  $n$ - мерный вектор известных констант и  $x$  –  $n$ - мерный вектор неизвестных:

$$\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}. \quad (4.2)$$

Для корректности задачи необходима устойчивость по исходным данным, которыми в данном случае являются коэффициенты матрицы и правая часть. Задача (4.2) устойчива по правой части если  $\det A \neq 0$  и чем ближе к нулю  $\det A$ , тем сильнее погрешность правой части может исказить решение. Число  $M_A = \|A^{-1}\| \cdot \|A\|$  называется *числом обусловленности* матрицы  $A$  и характеризует степень зависимости  $\delta x$  от  $\delta f$ , оно не связано с каким-либо численным алгоритмом, а характеризует только свойства задачи (4.2). Если еще дополнительно учесть возмущение коэффициентов матрицы  $A$ , то будет справедлива полная оценка относительной погрешности решения:

$$\|\delta x\| \leq \frac{M_A}{1 - M_A \frac{\|\Delta A\|}{\|A\|}} \left( \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta f\|}{\|f\|} \right)$$

Методы численного решения СЛАУ делятся на две группы: прямые методы и итерационные. В прямых методах решение системы (4.2) находится за конечное число арифметических действий. Вследствие погрешностей округления при решении задач на ЭВМ прямые методы на самом деле не приводят к точному решению. На практике они применяются для матриц умеренного порядка ( $10^2 \div 10^3$ ). Для матриц более высокого порядка целесообразнее применять итерационные методы, например, методы Якоби и Зейделя.

### Метод исключения Гаусса

Метод состоит из прямого и обратного хода. Общую формулу прямого хода метода исключения Гаусса можно представить следующим образом:

$$a_{kj}^{(k)} = a_{kj}^{(k-1)} / a_{kk}^{(k-1)}; a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} a_{kj}^{(k)}; k = 1, 2, \dots, n; i = k + 1, \dots, n; j = k + 1, \dots, n + 1. \quad (4.3)$$

В результате система уравнений приводится к верхнедиагональному виду.

Для определения вектора неизвестных необходимо произвести обратную подстановку:

$$x_i = b_{i,n+1} - \sum_{j=i+1}^n a_{ij} x_j, \quad i = n - 1, n - 2, \dots, 1. \quad (4.4)$$

Здесь верхние индексы опущены.

Из-за накопления погрешностей округления решение определяется с относительной погрешностью  $\delta x = O(M_A \cdot n \cdot 2^{-t})$ , где  $t$  – число разрядов мантиссы.

В вычислительной математике принято оценивать эффективность алгоритмов числом *операций*, причем каждая операция представляет собой комбинацию умножения и вычитания. Можно показать, что исключение по Гауссу требует  $n^3/3$  операций, где  $n$  — порядок матрицы, а обратная подстановка может быть выполнена приблизительно  $n^2/2$  операций.

### Метод LU-разложения

Лучшим методом решения СЛАУ является метод *разложения на треугольные матрицы*, или метод *LU-факторизации*. Алгоритмы этого метода близки к методу исключения Гаусса, хотя вычисления элементов матриц **L** и **U** могут проводиться в различной последовательности.

Матрицу системы уравнений (4.1) можно разложить на два сомножителя:

$$\mathbf{A} = \mathbf{LU} \quad (4.5)$$

и

Здесь матрица **L** является *нижней треугольной*, а матрица **U** - *верхней треугольной*. Заметим, что на главной диагонали матрицы **U** стоят единицы. Это означает, что определитель матрицы **A** равен произведению диагональных элементов матрицы **L**.

**LU**-разложение может проводиться, например, в соответствии с алгоритмом Краута по следующим формулам:

$$(4.8)$$

Его выполнение осуществляется при задании  $k = 1, 2, \dots, n$ . Заметим, что

требуемые в этих соотношениях значения элементов матриц  $\mathbf{L}$  и  $\mathbf{U}$  рассчитываются на предыдущих этапах процесса. Далее, каждый элемент  $a_{ij}$  матрицы  $\mathbf{A}$  требуется для вычисления только соответствующих элементов матриц  $\mathbf{L}$  и  $\mathbf{U}$ . Так как нулевые элементы матриц  $\mathbf{L}$  и  $\mathbf{U}$ , а также единичную диагональ матрицы  $\mathbf{U}$  запоминать не нужно, в процессе вычислений матрицы  $\mathbf{L}$  и  $\mathbf{U}$  могут быть записаны на месте матрицы  $\mathbf{A}$ , причем  $\mathbf{L}$  расположена в нижнем треугольнике ( $i \geq j$ ), а  $\mathbf{U}$  — соответственно в верхнем треугольнике ( $i < j$ ) матрицы  $\mathbf{A}$ . Обобщив все вышесказанное, опишем алгоритм  $\mathbf{LU}$ -разложения следующим образом:

*Шаг 1.* Положим  $k = 1$  и перейдем к шагу 3.

*Шаг 2.* Используя (4.8), рассчитываем  $k$ -й столбец матрицы  $\mathbf{L}$ . Если  $k = n$ . Закончим процедуру разложения.

*Шаг 3.* Используя (4.9), рассчитываем  $k$ -ю строку матрицы  $\mathbf{U}$ .

*Шаг 4.* Положим  $k = k + 1$  и перейдем к шагу

После того, как разложение на матрицы  $\mathbf{L}$  и  $\mathbf{U}$  осуществлено решение системы проводится в два приема, эквивалентных обратному ходу метода Гаусса.

$$(4.10)$$

$$(4.11)$$

где  $\mathbf{z}$  – вспомогательный вектор эквивалентный  $\mathbf{Ux}$ . Этот процесс называют *прямым исключением* (*прямой подстановкой* или *прямым ходом*. А в ходе *обратной подстановки* или *обратного хода* находятся уже непосредственно неизвестные

Число операций  $M$ , требуемых для проведения  $\mathbf{LU}$ -разложения, определяется как  $\frac{n^3}{3} - \frac{n}{3}$ , а для выполнения как прямой, так и обратной подстановок, равно примерно  $n^2/2$ . С точки зрения объема вычислений метод  $\mathbf{LU}$ -разложения вместе с прямой и обратной подстановками эквивалентен методу исключения Гаусса.

Изучение уравнений (4.10) показывает, что компоненты  $b_i$  используются только для определения величин  $z_i$  и позднее не требуются. Аналогично в (4.11) величина  $z_i$  не нужна после вычисления переменных  $x_i$ . Следовательно, при такой системе расчетов векторы  $\mathbf{b}$ ,  $\mathbf{z}$ , и  $\mathbf{x}$  могут быть размещены в одних и тех же ячейках памяти ЭВМ.

Важные черты метода  $\mathbf{LU}$ -разложения состоят в следующем:

1. Легко вычисляется определитель матрицы  $\mathbf{A}$

$$\det \mathbf{A} = \prod_{i=1}^n l_{ii} \quad (4.12)$$

2. Элементы матриц  $\mathbf{L}$  и  $\mathbf{U}$  могут быть записаны на месте элементов

матрицы  $\mathbf{A}$  и занесены в те же ячейки памяти (запоминать единичные элементы на главной диагонали матрицы  $\mathbf{U}$  нет необходимости).

3. Если требуется найти решение для другого вектора  $\mathbf{b}$  в правой части, то не нужно повторно проводить разложение матриц на треугольные, а достаточно только произвести прямую и обратную подстановки.

4. Для уравнений с транспонированной матрицей  $\mathbf{A}^t \mathbf{x} = \mathbf{c}$  решение находится при том же LU-разложении. Анализ таких систем уравнений необходим при расчете чувствительностей.

### Метод Зейделя

Метод Зейделя состоит в последовательном решении уравнений

$$f_i(x_1^{k+1}, x_2^{k+1}, \dots, x_i^{k+1}, x_{i+1}^k, \dots, x_n^k) = 0 \quad (4.13)$$

относительно переменной  $x_i^{k+1}$ ,  $i=1, 2, \dots, n$ ;  $k$  – номер итерации.

Здесь для отыскания  $\mathbf{x}^{k+1}$  необходимо решить  $n$  независимых скалярных уравнений вида  $x_i^{k+1} = x_i^k + b_i f_i(x_1^{k+1}, x_2^{k+1}, \dots, x_i^k, x_{i+1}^k, \dots, x_n^k)$ , которые в зависимости от удачного выбора коэффициента  $b$  могут как сходиться так и расходиться. В последнем случае при численной реализации метода это приведет к переполнению разрядной сетки ЭВМ и аварийной остановке программы. Поэтому в ходе выполнения алгоритма метода Зейделя необходимо следить за возможной расходимостью, например, используя критерий  $\max_{1 \leq i \leq m} |x_i^{k+1} - x_i^k| < \max_{1 \leq i \leq m} |x_i^k - x_i^{k-1}|$ , нарушение которого является признаком расходимости итерационного процесса.

Окончание итераций определяется либо заданием максимального числа итераций  $k_{\max}$ , либо условием

$$\max_{1 \leq i \leq m} |x_i^{k+1} - x_i^k| < \varepsilon, \quad (4.14)$$

где  $\varepsilon > 0$  – заданное число, определяющее погрешность метода.

### 4.3 Контрольные вопросы

1. Назовите условие необходимое для корректности решения задачи СЛАУ.
2. Что такое *число обусловленности* матрицы  $\mathbf{A}$  и что оно характеризует?
3. Какие ограничения накладываются на применение прямых методов решения СЛАУ?
4. Как оценивается число операций в методе Гаусса?
5. Выделите части алгоритмов методов Гаусса и LU-разложения, которые можно оформить одной общей процедурой?
6. Как оценивается число операций в методе LU-разложения?
7. Назовите преимущества метода LU-разложения.
8. Чем регулируется сходимость в методе Зейделя?
9. Сформулируйте несколько признаков расходимости вычислительного алгоритма.
10. Сформулируйте возможные условия окончания итерационного процесса.

#### 4.4 Порядок проведения вычислительных экспериментов

1. На языке Pascal написать процедуры методов Гаусса, LU-разложения и Зейделя со списком формальных параметров, содержащих исходные коэффициенты, количество итераций, требуемую погрешность (если необходимо), определитель матрицы, вектор неизвестных и т. д. При написании процедуры LU-разложения сделать доступными все преимущества метода, экономно расходовать память ЭВМ. При написании процедуры метода Зейделя предусмотреть адекватные действия программы при возможной расходимости алгоритма.

2. Написать основную программу, реализующую обращения к процедурам методов по выбору пользователя, представляющую результаты расчета в удобном виде. Предусмотреть возможность работы программы с тестовой СЛАУ; со СЛАУ пользователя, вводимой как с клавиатуры так и из файла данных.

3. Организовать раздел “Help” программы, содержащий ограничения по применению методов; выбору того или иного метода; рекомендации по порядку следования уравнений и т. п.

## Лабораторная работа № 5. Методы решения нелинейных уравнений

### 5.1 Введение

Целью данной работы является: изучение алгоритмов численного решения нелинейных уравнений и реализация методов решения нелинейных уравнений на алгоритмических языках.

### 5.2 Теоретическая часть

Рассмотрим задачу нахождения действительных корней  $f(x)=0$ , расположенных на заданном отрезке. Для ее численного решения используются итерационные методы, т.е. методы последовательных приближений.

В общем случае, когда нет информации о поведении  $f(x)$  алгоритм нахождения корней уравнения с помощью итерационного метода можно представить состоящим из двух этапов:

- 1) отыскания приближенного значения корня или содержащего его отрезка;
- 2) уточнения приближенного значения до некоторой заданной степени точности.

#### Метод отделения действительных корней

Предположим, что  $f(x)$  определена и непрерывна на  $[a,b]$ . Вычислим таблицу значений функции  $f(x)$  в заданных точках  $x_k \in [a,b]$ ,  $k = 0, 1, \dots, n$ . Число разбиений  $n$  не следует брать ни очень малым, т. к. тогда увеличивается вероятность пропустить расположенную рядом пару корней, ни очень большим, чтобы не увеличивать количество вычислений. Если обнаружится, что при некотором  $k$  числа  $f(x_k)$ ,  $f(x_{k+1})$  имеют разные знаки, то это будет означать, что на интервале  $[x_k, x_{k+1}]$  уравнение имеет по крайней мере один действительный корень, точнее, имеет нечетное число корней на  $[x_k, x_{k+1}]$ . В качестве начального приближения для реализации следующего этапа уточнения корня можно брать любые, точки принадлежащие найденному интервалу.

На втором этапе, используя заданное начальное приближение, строится итерационный процесс, позволяющий уточнить значение отыскиваемого корня. Таким образом, находится один из корней наиболее близкий к выбранному начальному приближению. Чтобы отыскать другие корни, надо менять начальные приближения.

Может оказаться, что и при других начальных данных метод сходится к тому же корню. Тогда целесообразно использовать прием выделения корней: если корень  $x = x_*$  кратности  $m$  найден, то рассматривается функция

$$g(x) = \frac{f(x)}{(x - x_*)^m} \quad (5.1)$$

и для неё повторяется процесс нахождения корня.

Рассмотрим далее несколько методов уточнения корней.

### Метод дихотомии

Начальным приближением для метода дихотомии или, как его еще называют, метода деления отрезка пополам является отрезок, на концах которого  $f(x)$  имеет разные знаки. Такой отрезок может быть получен рассмотренным выше методом отделения корней.

Предположим, что на  $[a, b]$  расположен лишь один корень  $x_*$  уравнения (11). Пусть для определенности  $f(a) > 0$ ,  $f(b) < 0$ . Положим  $x_0 = (a + b)/2$  и вычислим  $f(x_0)$ . Если  $f(x_0) < 0$ , то искомый корень находится на интервале  $[a, x_0]$ , если же  $f(x_0) > 0$ , то  $x_* \in [x_0, b]$ . Другими словами, из двух интервалов  $[a, x_0]$  и  $[x_0, b]$  выбираем тот, на границах которого функция  $f(x)$  имеет различные знаки. Далее находим точку  $x_1$  – середину выбранного интервала, вычисляем  $f(x_1)$  и повторяем указанный процесс. В результате получаем последовательность интервалов, содержащих искомый корень  $x_*$ , причем длина каждого последующего интервала вдвое меньше, чем предыдущего. Процесс заканчивается, когда длина вновь полученного интервала станет меньше заданного числа  $\varepsilon > 0$ , и в качестве ее корня  $x_*$  приближенно принимается середина этого интервала.

Заметим, что если на  $[a, b]$  имеется несколько корней, то указанный процесс сойдется к одному из корней, но заранее неизвестно, к какому именно.

К недостаткам метода дихотомии относят его медленную сходимость, скорость которой можно оценить как  $2^n$ , где  $n$  – количество итераций.

Для более быстрого получения решения нелинейного уравнения можно использовать одношаговые итерационные методы, общую формулу которых можно представить в следующем виде:

$$x^{k+1} = x^k + b_k f(x^k) \quad (5.2)$$

где  $k=0, 1, \dots$  – номер итерации,  $x^0$  – заданное начальное приближение,  $b_k$  – параметр, в зависимости от значения которого различают методы релаксации, Ньютона, секущих. Окончание итераций во всех этих методах определяется либо заданием максимального числа итераций  $k_{\max}$ , либо условием (38):  $|x^{k+1} - x^k| < \varepsilon$ , где  $\varepsilon > 0$  – заданное число.

**В методе релаксации**  $b_k = \text{const}$ . Это стационарный итерационный метод, который можно записать в виде

$$x^{k+1} = x^k + b f(x^k) \quad (5.3)$$

Метод сходится, если выполняется неравенство

$$-2 < b f'(x) < 0 \quad (5.4)$$

которое обеспечивается выбором коэффициента  $b$ .

Рассмотренный метод называют также **методом простых итераций** или **методом Зейделя**.

**В методе Ньютона**  $b_k = -1/f'(x^k)$ . Формула метода

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)} \quad (5.5)$$

Метод Ньютона называют также методом касательных, так как новое приближение  $x^{k+1}$  является абсциссой точек пересечения касательных, проведенных в точках  $(x^k, f(x^k))$  к графику функции  $f(x)$ , с осью  $Ox$ .

Метод Ньютона требует на каждой итерации вычисления производной. Нетрудно убедиться, что приведенное условие сходимости (5.4) имеет для метода Ньютона абсолютный минимум, следовательно, метод Ньютона не просто сходится, а имеет наивысшую скорость сходимости. Погрешность на каждой следующей итерации пропорциональна квадрату погрешности на предыдущей. Все эти утверждения справедливы, если начальное приближение выбрано достаточно хорошо. Если начальное приближение выбрано неудачно, то метод может сходиться медленно, либо не сойдется вообще. Формула метода Ньютона работает и в комплексной плоскости, причем можно не прибегать в (5.5) к разделению действительных и мнимых частей для работы только с действительными числами.

**Модифицированный метод Ньютона** имеет вид

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^0)} \quad (5.6)$$

и обладает линейной сходимостью. Упрощение в численной реализации по сравнению с обычным методом Ньютона состоит в том, что производную нужно вычислять не на каждой итерации, а лишь один раз. Возможно циклическое применение модифицированного метода Ньютона, когда  $f'(x^k)$  вычисляется через определенное число итераций.

**Метод секущих** получается из метода Ньютона заменой производной  $f'(x^k)$  разделенной разностью  $\frac{f(x^k) - f(x^{k-1})}{x^k - x^{k-1}}$ , вычисленной по известным значениям на двух предыдущих итерациях. В результате получаем итерационный метод

$$x^{k+1} = x^k - \frac{(x^k - x^{k-1})f(x^k)}{f(x^k) - f(x^{k-1})} \quad (5.7)$$

который в отличие от рассмотренных ранее методов является двухшаговым, т.е. новое приближение определяется двумя предыдущими итерациями. Соответственно необходимо задавать два начальных приближения - вектора  $x^0$  и  $x^1$ .



### 5.3 Контрольные вопросы

1. Сколько корней позволяет найти каждый из методов уточнения корней?
2. Перечислите приемы для отыскания всех корней нелинейного уравнения?
3. Какие условия накладываются на исходный отрезок в методе дихотомии?
4. Какой метод имеет наивысшую скорость сходимости?
5. Какой метод может расходиться с наибольшей вероятностью?
6. Какие методы требуют двух начальных приближений?
7. Каким образом можно найти комплексные корни нелинейного уравнения?

### 5.4 Порядок проведения вычислительных экспериментов

1. На языке Pascal написать процедуры методов дихотомии, Ньютона, секущих, простых итераций, со списком формальных параметров, содержащих начальные приближения, требуемую погрешность, количество итераций (если необходимо), и т. д. При написании процедуры простых итераций предусмотреть автоматическое задание коэффициента  $b$  из некоторого диапазона чисел; проверку сходимости в ходе итераций; автоматическое изменение  $b$  путем перебора в случае необходимости.
2. На языке Pascal написать процедуру метода отделения корней.
3. Написать основную программу, реализующую обращения к процедуре метода отделения корней, к процедурам методов по выбору пользователя, представляющую результаты расчета в удобном виде; позволяющие сделать выводы по точности и скорости нахождения решения различными методами. Предусмотреть возможность работы программы с тестовым уравнением, имеющим 3 действительных корня; с уравнением пользователя вида полинома, задаваемого вводом с клавиатуры его коэффициентов.
4. Организовать раздел “Help” программы.

## Лабораторная работа № 6. Методы решения краевых задач

### 6.1 Введение

Цель данной работы: изучение методов численного решения задачи Коши и граничной задачи для дифференциальных уравнений, применение встроенных функций системы MathCad для решения начальных и граничных задач.

### 6.2 Теоретическая часть

Краевые задачи для систем обыкновенных дифференциальных уравнений делятся на два класса.

#### 1 Задача Коши (начальная задача)

$$\frac{du_i(t)}{dt} = f_i(t, u_1, u_2, \dots, u_m), \quad t > 0, \quad i = 1, 2, \dots, m. \quad (6.1)$$

$$u_i(0) = u_i^{(0)}, \quad i = 1, 2, \dots, m, \quad (6.2)$$

где (6.2) представляют собой начальные условия.

Существование и единственность решения задачи Коши гарантируется требованиями к функциям  $f_i$ ,  $i = 1, 2, \dots, m$ : они должны быть непрерывны по всем аргументам в рассматриваемой области и удовлетворять условию Липшица по аргументам  $u_1, u_2, \dots, u_m$ .

Рассматривая алгоритмы методов решения задачи Коши для простоты изложения ограничимся одним уравнением системы

$$\frac{du}{dt} = f(t, u), \quad t > 0, \quad u(0) = u_0. \quad (6.3)$$

Введем по переменному  $t$  равномерную сетку с шагом  $\tau > 0$ , т.е. рассмотрим множество точек

$$\omega_\tau = \{t_n = n\tau, n = 0, 1, 2, \dots\}$$

Будем обозначать через  $u(t)$  точное решение задачи (48), а через  $y_n = y(t_n)$  – приближенное решение. Заметим, что приближенное решение является сеточной функцией, т.е. определено только в точках сетки  $\omega_\tau$ .

Отметим, что сетка  $\omega_\tau$  является равномерной, что достаточно для большинства практических задач. Однако для неравномерно меняющихся функций с выраженным резонансом целесообразно вводить неравномерную сетку.

#### Метод Эйлера

Уравнение (6.3) заменяется разностным уравнением

$$\frac{y_{n+1} - y_n}{\tau} - f(t_n, y_n) = 0, \quad n = 0, 1, 2, \dots, \quad y_0 = u_0. \quad (6.4)$$

Решение этого уравнения находится явным образом по рекуррентной формуле

$$y_{n+1} = y_n + \tau f(t_n, y_n), \quad n = 0, 1, 2, \dots, y_0 = u_0. \quad (6.5)$$

При использовании приближенных методов вводится понятие невязки или погрешности аппроксимации разностного уравнения (6.4) на решении исходного уравнения (6.3). Невязка представляет собой результат подстановки точного решения  $u = u(t)$  в левую часть разностного уравнения (6.4):

$$\psi_n^{(1)} = -\frac{u_{n+1} - u_n}{\tau} + f(t_n, u_n) \quad (6.6)$$

Если бы приближенное решение  $y_n$  совпадало с точным  $u(t_n)$ , то невязка равнялась бы нулю. Говорят, что разностный метод *аппроксимирует исходное дифференциальное уравнение*, если  $\psi_n^{(1)} \rightarrow 0$  при  $\tau \rightarrow 0$ . Разностный метод имеет  $p$ -й порядок аппроксимации, если  $\psi_n^{(1)} = O(\tau^p)$ . Можно показать, что при очень общих предположениях порядок точности разностного метода совпадает с порядком аппроксимации.

Порядок аппроксимации метода Эйлера (6.5) нетрудно найти, используя разложение по формуле Тейлора. Поскольку

$$\frac{u_{n+1} - u_n}{\tau} = u'(t_n) + O(\tau),$$

то в силу уравнения (6.3)

$$\psi_n^{(1)} = -u'(t_n) + f(t_n, u_n) + O(\tau) = O(\tau),$$

т.е. метод Эйлера имеет первый порядок аппроксимации. При выводе предполагалась ограниченность  $u''(t)$ .

### Симметричная схема

Уравнение (6.3) заменяется разностным уравнением

$$\frac{y_{n+1} - y_n}{\tau} - \frac{1}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1})) = 0, \quad n = 0, 1, \dots, \quad y_0 = u_0. \quad (6.7)$$

Данный метод более сложен в реализации, чем метод Эйлера (6.5), так как новое значение  $y_{n+1}$  определяется по найденному ранее  $y_n$  путем решения уравнения

$$y_{n+1} - 0.5\tau f(t_{n+1}, y_{n+1}) = F_n,$$

где

$$F_n = y_n + 0.5\tau f(t_n, y_n).$$

По этой причине метод называется  *неявным* .

### Методы Рунге-Кутты

Методы Рунге-Кутты отличаются от разностных методов тем, что в них допускается вычисление правых частей  $f(t, u)$  не только в точках сетки, но и в промежуточных точках.

Различают два метода Рунге-Кутты второго порядка: со средней производной:

$$\begin{aligned} k_1 &= f(t_n, y_n), k_2 = f(t_n + \tau, y_n + \tau k_1), \\ y_{n+1} &= y_n + 0.5\tau(k_1 + k_2), \end{aligned} \quad (6.8)$$

и с производной в средней точке.

$$\begin{aligned} k_1 &= f(t_n, y_n), \quad k_2 = f(t_n + 0.5\tau, y_n + 0.5\tau k_1), \\ y_{n+1} &= y_n + \tau k_2. \end{aligned} \quad (6.9)$$

Поскольку требуется вычислить две промежуточные функции  $k_1$  и  $k_2$ , данные методы относятся к *двухэтапным методам*.

Реализация двухэтапных методов Рунге-Кутты (6.8) и (6.9) соответствует методу *предиктор – корректор* (предсказывающе – исправляющему), поскольку на первом этапе приближенное значение предсказывается с невысокой точностью  $O(\tau)$ , а на втором этапе это предсказанное значение исправляется, так что результирующая погрешность имеет второй порядок по  $\tau$ .

Наиболее точными являются методы Рунге-Кутты четвертого порядка, в качестве примера которых можно привести такую разновидность:

$$\begin{aligned} k_1 &= f(t_n, y_n), \quad k_2 = f(t_n + \frac{\tau}{2}, y_n + \frac{\tau k_1}{2}), \\ k_3 &= f(t_n + \frac{\tau}{2}, y_n + \frac{\tau k_2}{2}), \quad k_4 = f(t_n + \tau, y_n + \tau k_3), \quad (6.10) \\ \frac{y_{n+1} - y_n}{\tau} &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned}$$

Этот метод имеет четвертый порядок аппроксимации (т.е.  $\psi_n^{(1)} = O(\tau^4)$ ) и, как можно показать, четвертый порядок точности.

В заключение заметим, что всюду фигурировали дифференциальные уравнения первого порядка. Следовательно если на практике необходимо решить дифференциальное уравнение  $m$ -го порядка, то его необходимо предварительно преобразовать к эквивалентной системе  $m$  уравнений первого порядка.

## 2. Граничная задача

Граничная задача отличается от рассмотренной ранее задачи Коши тем, что ни в одной из точек  $t$  не известны условия на все сразу функции  $u_i$ , ( $i=1,2,\dots,m$ ), и состоит в отыскании функций  $u_i(t)$ , непрерывно дифференцируемых на интервале  $(a,b)$ , непрерывных на отрезке  $[a,b]$ , удовлетворяющих уравнениям (6.1) при  $t \in (a,b)$  дополненных граничными условиями

$$u_i(t_i) = u_i^{(0)}, \quad t_i \in [a,b], \quad i=1,2,\dots,m. \quad (6.11)$$

Очевидно, что граничная задача определена минимум для двух уравнений (6.1). Как правило,  $n$  условий известно в начальной точке  $a$  интервала и  $m-n$  условий – в конечной точке  $b$  интервала, т.е. на его границах. Ни один из методов решения задачи Коши не удастся применить к решению граничной задачи в силу недостатка условий на  $m-n$  функций в начальной точке.

## Метод стрельбы

Метод стрельбы основывается на рассмотренных ранее методах решения задачи Коши и состоит в построении предположения о недостающих значениях неизвестных функций в начальной точке интервала и их пошаговом уточнении в дальнейшем.

Рассмотрим для простоты систему двух ОДУ ( $m=2$ )

$$\begin{aligned} \frac{du_1(t)}{dt} &= f_1(t, u_1, u_2), \\ \frac{du_2(t)}{dt} &= f_2(t, u_1, u_2), \quad a \leq t \leq b, \end{aligned} \quad (6.12)$$

дополненную граничными условиями

$$u_1(a) = \mu_1, u_2(b) = \mu_2, \quad (6.13)$$

где  $\mu_1, \mu_2$  – заданные числа.

Сделаем предположение, что  $u_2(a) = C_0$  и решим одним из известных методов сформированную таким образом задачу Коши. В результате получим некоторое значение  $u_2(b) = D_0$ , отличающееся от заданного второго граничного условия (6.13). Введем функцию  $g(C_i) = D_i - \mu_2$ , характеризующую неточность проведенного решения. Если бы мы сразу точно угадали значение  $u_2(a)$ , соответствующее данной граничной задаче, то функция  $g$  обратилась бы в ноль. Однако, благодаря произвольному выбору  $C_0$ , она в этой точке  $C_0$  отлична от нуля и нам как раз необходимо найти ноль функции  $g(C)$ , который даст истинное значение  $C$ . Это можно сделать одним из методов нахождения корней, например, методом секущих, но для этого необходима вторая точка функции  $g(C)$ .

Изменим выбранное значение  $C_0$  на небольшую величину  $\Delta$  и решим такую задачу Коши. В результате получим новое значение  $u_2(b) = D_i^\Delta$  и функции  $g(C_0 + \Delta) = D_0^\Delta - \mu_2$ . Теперь можно применить метод секущих:

$$C_{i+1} = C_i - \frac{g(C_i) \cdot \Delta}{g(C_i + \Delta) - g(C_i)} = C_i - \frac{(D_i - \mu_2) \cdot \Delta}{D_i^\Delta - D_i}. \quad (6.14)$$

Вычисления по этой рекуррентной формуле проводятся до тех пор, пока полученное приращение  $C_{i+1} - C_i$  не станет меньше требуемой точности  $\varepsilon$ .

Обычно при уточнении значения  $C$  используют малозатратный метод невысокого порядка точности, а когда значение  $C$  найдено с требуемой точностью, задачу Коши решают методом Рунге-Кутты высокого порядка точности.

Описанный метод получил название метода стрельбы потому, что задание  $C_0$  и построение соответствующей траектории напоминает пробный выстрел, а уточнение начального условия по формуле (6.14), соответствующее сведению к нулю функции  $g(C)$ , – серию выстрелов, постепенно приближающихся к цели.

Метод стрельбы может быть применим только к задачам, устойчивым по начальным условиям.

### Разностный метод

Разностный метод основан на введении на отрезке  $[a, b]$  сетки, а в большинстве случаев - равномерной сетки  $\omega_\tau = \{t_n = n\tau, n = 0, 1, 2, \dots\}$ , и замене всех входящих в уравнения производных  $u''(x_i)$  разностными производными  $u_{x,i}$ ,  $u_{xx,i}$  и т.д. (3.1). Он приводит к замене системы  $m$  дифференциальных уравнений первого порядка (6.1) системой  $m \cdot N$  алгебраических уравнений относительно  $m \cdot (N+1)$  неизвестных значений функций в  $N+1$  точках сетки. Недостающие  $m$  уравнений дают граничные условия (6.13).

Нелинейные дифференциальные уравнения приводят к системе нелинейных алгебраических уравнений, а линейные – к СЛАУ.

Рассмотрим в качестве примера дифференциальное уравнение второго порядка (эквивалентное системе двух уравнений первого порядка)

$$u''(x) = -f(x) \quad (6.15)$$

при  $x \in (a, b)$ , дополненное условиями

$$u(a) = \mu_1, \quad u(b) = \mu_2. \quad (6.16)$$

Заменим  $u''(x_i)$  второй разностной производной  $u_{xx,i}$  (3.1) и получим разностное уравнение

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = -f_i. \quad (6.17)$$

Это уравнение можно записать для  $i = 1, 2, \dots, N-1$ , т.е. во всех внутренних точках сетки  $\omega_h$ ; общее количество неизвестных  $N+1$ .

В граничных точках в соответствии с (6.16) следует положить

$$u_0 = \mu_1, \quad u_N = \mu_2. \quad (6.18)$$

Таким образом, (6.3) как раз и представляют два недостающих уравнения для определения  $N+1$  неизвестных. Система уравнений (6.17), (6.18) называется *разностной схемой* или *разностной краевой задачей*, соответствующей исходной дифференциальной задаче (6.15)–(6.16). В дальнейшем, чтобы не было путаницы в обозначениях, будем через  $u(t)$  обозначать решение дифференциальной задачи и через  $y_i = y(t_i)$  - решение разностной задачи.

Итак, мы получили разностную схему

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} = -f_i, \quad i = 1, 2, \dots, N-1, \quad (6.19)$$

$$y_0 = \mu_1, \quad y_N = \mu_2,$$

решать далее которую можно методом прогонки.

### 6.3 Контрольные вопросы

1. Сформулируйте условия существования и единственности решения задачи Коши.
2. В каких случаях целесообразно использовать неравномерную сетку?
3. Дайте определение невязки разностного метода.
4. Как определяется порядок аппроксимации разностного метода?

5. Как определяется порядок точности разностного метода?
6. Какой порядок аппроксимации имеет метод Эйлера?
7. В чем заключаются преимущества и недостатки симметричной схемы?
8. Назовите особенности методов Рунге-Кутты.
9. Перечислите методы Рунге-Кутты второго порядка. В чем заключаются их отличия от метода Эйлера и симметричной схемы?
10. Какого максимального порядка используются методы Рунге-Кутты?
11. Запишите дифференциальное уравнение второго порядка и преобразуйте его к системе уравнений первого порядка.
12. Какие ограничения имеет метод стрельбы?
13. Сколько алгебраических уравнений возникает при использовании разностного метода.
14. Назовите и опишите следующий этап решения, следующий за разностным методом
15. Почему нельзя применять методы решения задачи Коши к граничной задаче?

#### 6.4 Порядок проведения вычислительных экспериментов

1. Записать выражение для невязки методов Рунге Кутты второго порядка и симметричной схемы. Используя разложение в ряд Тэйлора с учетом достаточного числа его членов, найти порядок аппроксимации каждого из этих трех методов.
2. Изучить список параметров функций программной системы MathCad: Rkfixed, Rkadapt. Используя обе эти функции найти решения задачи Коши для соответствующего варианта задания, результаты представить на графике и сравнить.
3. Изучить список параметров функций программной системы MathCad: Sbval. Найти решение граничной для соответствующего варианта задания, результаты представить на графике.

#### Варианты заданий

1.  $y'' - y = x, \quad x \in [0,1]$

Начальные условия  $y(0) = 0.5, \quad y'(0) = 0.1$

Граничные условия  $2y(0) - y'(0) = 1, \quad y(1) = 2$

2.  $y'' + \frac{1}{x}y' - \frac{3}{x^2}y = -\frac{3}{4\sqrt{x}}, \quad x \in [1,4]$

Начальные условия  $y(1) = 1, \quad y'(1) = 0.5$

Граничные условия  $y(1) = 1, \quad y(4) - 2y'(4) = 2$

$$3. y'' + 2y' - 3xy = \frac{2-8x}{x^3}, \quad x \in [1, 2] \quad \text{на сетке с шагом } h = 0.2$$

Начальные условия  $y(1) = 1, \quad y'(1) = 0.5$

Граничные условия  $y(1) = 1, \quad y(2) = 0.5$

$$4. y'' - y' \ln(x) - 2y = 1, \quad x \in [0.5, 1.5], \quad \text{шаг сетки } h = 0.125.$$

Начальные условия  $y(0.5) = 0.5, \quad y'(0.5) = 0.5$

Граничные условия  $y(0.5) + y'(0.5) = 1, \quad y(1.5) - y'(1.5) = 0$

$$5. x^2 y'' \ln(x) - xy' + y = 0, \quad x \in [1, e]$$

Начальные условия  $y(1) = 0, \quad y'(1) = 1$

Граничные условия  $y(1) = 0, \quad y(e) = e - 2$

$$6. y'' = \sqrt{y'}, \quad x \in [0, 2]$$

Начальные условия  $y(0) = 0, \quad y'(0) = 0.1$

Граничные условия  $y(0) = 0, \quad y(2) = \frac{2}{3}$

## Лабораторная работа № 7. Методы безусловной оптимизации

### 7.1 Введение

Цель данной работы: изучение алгоритмов методов безусловной оптимизации нулевого и первого порядка, реализация методов оптимизации на алгоритмических языках.

### 7.2 Теоретическая часть

Задача оптимизации формулируется в виде задачи математического программирования:

$$\left. \begin{array}{l} \text{extr}_{X \in XD} F(X), \\ XD = \{X : \Phi(X) \geq 0, \psi(X) = 0\} \end{array} \right\}, \quad (7.1)$$

где  $F(X)$  — целевая функция,  $X$  — вектор управляемых параметров,  $XD$  — область варьирования  $X$ .

Выражение (7.1) означает, что необходимо найти экстремум целевой функции  $F(X)$  в пределах области  $XD$  пространства управляемых параметров, заданной ограничениями типа неравенств  $\Phi(X) \geq 0$  и равенств  $\psi(X) = 0$ . Под экстремумом в практических задачах подразумевается максимум или минимум.

Выходной параметр — величина, характеризующая определенное свойство проектируемого объекта. Поэтому функции  $F(X)$ ,  $\Phi(X)$  и  $\psi(X)$  в (7.1) должны быть непосредственно или косвенно связаны с выходными



параметрами  $y_j(X)$ ,  $j=1,m$ . Способ отражения  $y_i(X)$  в функциях  $F(X)$ ,  $\Phi(X)$ ,  $\psi(X)$  является основной характеристикой постановки задачи оптимизации. Неоднозначность постановки задачи оптимизации определяется ее *многокритериальностью*, под которой подразумевается наличие более одного параметра, претендующего на роль критерия оптимальности, которыми являются выходные параметры. В силу этих причин не удастся найти экстремум функции  $F(X)$  обычными средствами математического анализа.

Методы оптимизации имеют поисковую природу и сводятся к многошаговому вычислительному процессу последовательного приближения к искомому экстремуму. Каждый шаг процесса заключается в переходе из точки  $X_{k-1}$  в пространстве управляемых параметров в точку  $X_k$ . Для такого перехода нужно определить направление  $g_k$  перемещения и величину шага  $h_k$  в этом направлении, такие, что  $X_k = X_{k-1} + h_k g_k$ . Получающаяся последовательность точек  $(X_0, X_1, X_2, \dots, X_N)$ , называемая *траекторией поиска*, должна сходиться в экстремальной точке  $X^*$ . Окончание поиска связывается с попаданием текущей точки поиска  $X_k$  в заданную окрестность экстремальной точки. Однако точка  $X^*$  неизвестна заранее, поэтому возможны различные варианты интерпретации условия окончания поиска.

Управляемые параметра в своей исходной трактовке являются величинами с различными физическими размерностями. В процессе поиска необходимо их сопоставление, определение расстояний между точками и другие операции, возможные только при условии нормирования управляемых параметров. Изменение способа нормирования приводит к деформации пространства управляемых параметров, к изменению траекторий поиска.

Таким образом, содержанием любого метода или алгоритма поисковой оптимизации должны быть способы выбора: направление поиска  $g_k$ ; величины шага  $h_k$ ; формул для нормирования управляемых параметров; критерия окончания поиска. Эффективность поиска зависит от того, как сделан этот выбор. Составляющими эффективности являются надежность, точность, экономичность. *Надежность* определяется как вероятность достижения заданной  $\varepsilon$ -окрестности экстремальной точки при применении данного метода; *точность* характеризуется гарантированным значением  $\varepsilon$ ; *экономичность* отождествляется с потерями на поиск. Потери на поиск выражают трудоемкость процедуры оптимизации, которую в большинстве случаев оценивают количеством обращений к ММ объекта.

Рассматриваемые методы являются методами поиска локальных экстремумов. Это основные методы, так как методов глобальной оптимизации, обеспечивающих нахождение глобального экстремума с приемлемыми потерями на поиск, для задачи математического программирования общего вида (7.1) не существует. Поиск глобального экстремума осуществляется путем локальной оптимизации из нескольких исходных точек, выбираемых случайным образом в пределах области, задаваемой прямыми ограничениями. В многоэкстремальных задачах возможно получение нескольких локальных экстремумов, из которых выбирается наилучший. Вероятность определения

глобального экстремума при подобном подходе тем меньше, чем меньше объем области притяжения глобального экстремума. Малый объем этой области, как правило, свидетельствует о низкой стабильности выходных параметров в точке экстремума, следовательно, глобальный экстремум может оказаться малополезным. Поэтому оптимизация на основе небольшого числа вариантов локального поиска является достаточной.

Способ выбора направления поиска  $g_k$  является определяющим для методов безусловной оптимизации, которые бывают нулевого, первого и второго порядков. В *методах нулевого порядка* для определения  $g_k$  производные целевой функции  $F(X)$  по управляемым параметрам  $X$  не используются. В *методах первого и второго порядков* используются соответственно первые и вторые производные  $F(X)$  по  $X$ .

Способ выбора величины шага  $h_k$  бывает двух видов. В способе оптимального шага на каждом  $k$ -м шаге поиска экстремума исходной задачи  $\min_x F(X)$  решается вспомогательная задача одномерной оптимизации:

$$\min_{h_k} F(X_{k-1} + h_k g_k) \quad (7.2)$$

Решением (7.2) является оптимальная величина шага  $h_k$ , минимизирующая  $F(X)$  на луче, выходящем в направлении  $g_k$  из точки  $X_{k-1}$ . В *способе задаваемого шага* величина  $h_k$  является постоянной на всей траектории поиска или только на ее частях, выделяемых в зависимости от близости к искомому экстремуму. При выполнении условий попадания в заданную окрестность экстремальной точки  $X^*$  значение  $h_k$  уменьшается, поскольку мелкие шаги позволяют точнее определить ее положение.

Нормирование управляемых параметров осуществляется *способом логарифмической нормализации* параметров

$$u_i = \ln(x_i / \xi_i), \quad (7.3)$$

или *способом линейной нормализации*

$$u_i = x_i / \xi_i, \quad (7.4)$$

где  $u_i$  и  $x_i$  — соответственно нормализованный и ненормализованный управляемый параметр;  $\xi_i$  — константа, равная единице измерения величины  $x_i$ . Преимущество логарифмической нормализации — оперирование относительными приращениями параметров.

В качестве условия попадания в заданную окрестность оптимальной точки  $X^*$  принимается условие малости расстояния, на которое произошло продвижение в пространстве управляемых параметров в результате последних  $r$  шагов. Это расстояние является нормой разности векторов нормированных управляемых параметров  $U_k$  и  $U_{k-r}$ :

$$\|U_k - U_{k-r}\| < \delta, \quad (7.5)$$

где  $\delta$  — заданная положительная константа.

Критерием прекращения поиска являются условия (7.5) и  $h_k \leq h_{\min}$ , где  $h_{\min}$  — нижний предел  $h_k$  в способе задаваемого шага. Вместо (7.5) иногда

используется  $F(X_k) - F(X_{k-r}) < \delta$ .

Стратегия поиска в методах нулевого порядка основывается на переборе ограниченного множества избранных направлений поиска или на случайном выборе.

Метод покоординатного спуска (метод Гаусса -Зейделя) характеризуется тем, что в нем избранное множество направлений поиска составляют направления вдоль  $n$  координатных осей пространства управляемых параметров. Для определения  $h_k$  используется способ оптимального шага. В условии (7.5)  $r$  принимается равным  $n$ , т.е. поиск заканчивается, если в цикле из  $n$  шагов вдоль каждой из координатной осей пройдено расстояние, меньшее  $\delta$ . Метод покоординатного спуска малонадежен. Так, если целевая функция имеет овражный характер и линия дна оврага на параллельна какой-либо из координатных осей, то все разрешенные направления (вдоль координатных осей) в точках дна оврага могут оказаться неудачными — вести по склону оврага вверх. Например, из точки  $A$  (рис.7.1) невозможно продолжать покоординатный спуск, хотя она находится далеко от точки минимума  $X^*$ . Числами 8, 10, 12, 14 отмечены значения целевой функции, соответствующие линиям равного уровня.

### Рисунок 7.1 - Овражная функция

Поэтому применяются другие методы перебора направлений, улучшающие стратегию покоординатного спуска, например такие как методы конфигураций и Розенброка.

Метод конфигураций включает выполнение циклов из  $n$  шагов покоординатного спуска. После каждого цикла делается один дополнительный шаг, определяемый с помощью одномерной оптимизации (7.2), вдоль направления  $U_k - U_{k-n}$ , где  $k = n, 2n, 3n, \dots$  т.е. по линии, проходящей через конечные точки двух смежных циклов шагов.

Рисунок 7.2 - Траектория поиска по методу конфигураций

На рис. 7.2 показан поиск по методу конфигураций: точки  $X_1$  и  $X_2$  получены из начальной  $X_0$  с помощью покоординатного спуска, точка  $X_d$  — результат дополнительного шага.

Метод Розенброка характеризуется тем, что в нем реализована идея поворота координатных осей после каждого цикла из  $n$  шагов покоординатного спуска таким образом, чтобы одна из осей новой системы координат оказалась параллельной линии, соединяющей точки  $U_k$  и  $U_{k-n}$ , т.е. заняла положение, близкое к параллельному по отношению к линии дна оврага. Тогда заметно повышается вероятность того, что следующий цикл шагов покоординатного спуска будет успешным.

Методы случайного поиска характеризуются большим числом реализующих их алгоритмов. В качестве примера отметим алгоритм, в котором элементы вектора  $g_k$  являются равномерно распределенными в диапазоне  $[-1, +1]$  случайными величинами и используется способ задаваемого шага. Неудачные шаги, приводящие к ухудшению целевой функции не выполняются.

Методы первого порядка, называемые градиентами, приводят к цели за меньшее число шагов по сравнению с методами нулевого порядка, однако выполнение каждого шага более трудоемко, так как вычисляется градиент целевой функции  $gradF(X_{k-1})$ , где  $X_{k-1}$  — результат предыдущего шага.

Метод наискорейшего спуска предписывает вести поиск в направлении

$$g_k = -gradF(X_{k-1}), \quad (7.6)$$

$h_k$  выбирается по способу оптимального шага.

Метод сопряженных градиентов:

$$g_k = -gradF(X_{k-1} + \beta_k g_{k-1}), \quad (7.7)$$

где  $\beta_k$  — коэффициент, учитывающий предысторию поиска. Благодаря такому учету «сглаживается» траектория поиска и быстрее достигается экстремум. Коэффициент  $\beta_k$  рассчитывается как отношение скалярных произведений векторов:

$$\beta_k = \frac{\langle \text{grad}F(X_{k-1}), \text{grad}F(X_{k-1}) \rangle}{\langle \text{grad}F(X_{k-2}), \text{grad}F(X_{k-2}) \rangle}.$$

Рисунок 7.3 - Траектория поиска по методам наискорейшего спуска (сплошная линия) и сопряженных градиентов (пунктирная линия)

На рис. 7.3 сплошной линией показана траектория поиска минимума квадратичной целевой функции методом наискорейшего спуска. Поиск по методу сопряженных градиентов (пунктирная линия) дает более быстрое определение экстремума квадратичной функции.

Метод переменной метрики:

$$g_k = -H_k \text{grad}F(X_{k-1}), \quad (7.8)$$

где  $H_k$  — матрица, рассчитываемая по специальному алгоритму.

Заметим, что (7.8) применимо и к методу наискорейшего спуска, если  $H_k$  — единичная матрица. Если принять  $H_k = \mathcal{Y}_k^{-1}$ , где  $\mathcal{Y}_k^{-1}$  — обратная матрица вторых частных производных  $F(X)$  по  $X$ , называемая матрицей Гессе, то мы имеем метод Ньютона, относящийся к методам второго порядка. Методы второго порядка практически не применяются из-за трудностей расчета матрицы Гессе. Поэтому вместо  $\mathcal{Y}_k^{-1}$  используется ее приближение, рассчитываемое в методе переменной метрики без использования вторых производных  $F(X)$  по  $X$ .

### 7.3 Контрольные вопросы

1. Сформулируйте причины, из-за которых в задачах оптимизации не удастся использовать обычные средства математического анализа?
2. На чем основаны и к чему сводятся методы оптимизации?
3. Что называется называется *траекторией поиска*?
4. Что является содержанием любого метода или алгоритма поисковой оптимизации?
5. Как осуществляется поиск глобального экстремума?
6. Покажите преимущество логарифмической нормализации параметров.

7. Приведите варианты критериев прекращения поиска.
8. Отметьте недостатки метод покоординатного спуска.
9. Выполните геометрические построения, иллюстрирующие отличия метода конфигураций от метода Розенброка.

#### 7.4 Порядок проведения вычислительных экспериментов

1. На языке Pascal написать процедуры методов координатного спуска, Розенброка и конфигураций, случайного поиска, методов наискорейшего спуска и сопряженных градиентов
2. Написать основную программу, реализующую обращения к процедурам методов по выбору пользователя, представляющую результаты расчета в удобном виде; позволяющие сделать выводы по точности и скорости нахождения решения различными методами. Предусмотреть возможность работы программы с тестовой целевой функцией; с  $n$ -координатной целевой функцией пользователя вида полинома, задаваемого вводом с клавиатуры его коэффициентов.
3. Организовать раздел “Help” программы.

#### Список рекомендуемой литературы

1. Влах И., Сингхал К. Машинные методы анализа и проектирования электронных схем: Пер. с англ.- М.: Радио и связь. 1988. - 560 с.
2. Дьяконов В. П. MathCAD 2001: Учебный курс. – СПб.: Питер, 2001. – 621 с.
3. Дьяконов В. П. Абраменкова Н. В. MathCAD 7.0 в математике, физике и в Internet. - М.: Нелидж., 1998. - 352 с.
4. Крылов В. И., Бобков В. В., Монастырский П. И. Начала теории вычислительных методов. Дифференциальные уравнения. – Минск: Наука и техника, 1982. –288 с.
5. Мудров А. Е. Численные методы для ПЭВМ на языках Бэйсик, Фортран и Паскаль. – Томск: МП “РАСКО”. 1991. – 272 с.
6. Очков В. Ф. Mathcad PLUS 6.0 для студентов и инженеров. М.: Компьютер-Пресс, 1996. – 238 с.
7. Самарский А. А., Гулин А. В. Численные методы: Учеб. пособ. для вузов.- М.: Наука. 1989.- 432 с.
8. Турчак Л. И. Основы численных методов. Учебное пособие для вузов / Под ред. В. В. Щерникова. – М.: Наука. 1987.- 320 с.

Учебное пособие

Саликаев Ю.Р.

Компьютерное моделирование и проектирование.  
Лабораторный практикум. Часть 1

Методические указания к лабораторным работам

Усл. печ. л. \_\_\_\_\_. Препринт  
Томский государственный университет  
систем управления и радиоэлектроники  
634050, г.Томск, пр.Ленина, 40