Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пермский национальный исследовательский
политехнический университет»

**И.Н. Щапова, В.А. Щапов**

# ПРОГРАММИРОВАНИЕ. ОБРАБОТКА ИНФОРМАЦИИ В ПАКЕТАХ ПРИКЛАДНЫХ ПРОГРАММ

*Лабораторный практикум*
*по дисциплине «Информатика»*

Издательство
Пермского национального исследовательского
политехнического университета
2018

УДК 004.9 + 004.42](076.5)
Щ12

Рецензенты:
канд. физ.-мат. наук *А.В. Евграфова*
(Институт механики сплошных сред УрО РАН, г. Пермь);
канд. техн. наук, доцент кафедры горной электромеханики *С.В. Нусс*
(Пермский национальный исследовательский
политехнический университет)

**Щапова, И.Н.**
Щ12 Программирование. Обработка информации в пакетах прикладных программ : лаб. практикум по дисциплине «Информатика» / И.Н. Щапова, В.А. Щапов. – Пермь : Изд-во Перм. нац. исслед. политехн. ун-та, 2018. – 35 с.

ISBN 978-5-398-02065-6

Приведены лабораторные работы по дисциплине «Информатика» по темам: «Решение задач с использованием методов структурного программирования» и «Обработка информации в пакетах прикладных программ», а также цели работ, задания по вариантам, примеры выполнения заданий, вопросы для контроля и список рекомендуемой литературы. Рассмотрено решение задач в математическом пакете Mathcad.

Предназначено для студентов, обучающихся по направлениям подготовки бакалавров и специалистов.

УДК 004.9 + 004.42](076.5)

ISBN 978-5-398-02065-6 

## Содержание

## Порядок выполнения лабораторных работ

1. Проанализировать задание, установить, какие величины являются входными параметрами, какие – выходными.
2. Составить блок-схему алгоритма решения задачи.
3. По разработанному алгоритму составить программу на языке PascalABC.NET.
4. В редакторе среды программирования PascalABC.NET ввести программу.
5. Осуществить компиляцию и отладку программы.
6. Получить результаты работы программы.
7. Выполнить задания в математическом пакете Mathcad.
8. Оформить отчет по работе и ответить на контрольные вопросы.

## Содержание отчета по лабораторной работе

1. Номер и тема лабораторной работы.
2. Содержание задания.
3. Математическая постановка задачи, т.е. представление ее в виде уравнений, соотношений, ограничений и т.п.
4. Блок-схема алгоритма решения задачи.
5. Текст программы на языке PascalABC.NET с описанием используемых переменных и с комментариями к отдельным частям программы.
6. Результаты выполнения программы на языке PascalABC.NET.
7. Описание решения задачи в математическом пакете Mathcad.

## Лабораторная работа № 1

## Численные методы решения нелинейных уравнений

***Цели работы:***

– научиться использовать численные методы, языки и системы программирования для решения нелинейных уравнений;

– научиться использовать математический пакет Mathcad для решения нелинейных уравнений.

***Задание***.

1. Методом простых итераций и методом половинного деления (бисекции) решить уравнение с точностью $10^{-3}$ и с точностью $10^{-5}$ на языке программирования PascalABC.NET (в программах предусмотреть выполнение проверки правильности полученного значения корня уравнения). Блок-схемы алгоритмов решения уравнения методом простых итераций и методом половинного деления (бисекции) приведены на рис. 1, 2 [1]. Сравнить полученные значения корней уравнения между собой и с его точным значением, приведенным в таблице.

2. Решить уравнение средствами программы Microsoft Excel [2].

3. Решить уравнение в математическом пакете Mathcad [3].

4. Проанализировать все полученные решения уравнения.

***Варианты заданий***

| № варианта | Уравнение | Отрезок [a;b] | Точное значение корня |
|---|---|---|---|
| 1 | $3\sin(\sqrt{x})+0,35x-3,8=0$ | [2;3] | 2,2985 |
| 2 | $0,25x^3+x-1,2502=0$ | [0;2] | 1,0001 |
| 3 | $x+x^{1/2}+x^{1/3}-2,5=0$ | [0,4;1] | 0,7376 |
| 4 | $x-\dfrac{1}{3+\sin 3,6x}=0$ | [0;0,85] | 0,2624 |
| 5 | $3x-4\ln x-5=0$ | [2;4] | 3,2300 |
| 6 | $x-2+\sin\dfrac{1}{x}=0$ | [1,2;2] | 1,3077 |
| 7 | $1-x+\sin x-\ln(1+x)=0$ | [0;1,5] | 1,1474 |
| 8 | $x+\cos(x^{0,52}+2)=0$ | [0,5;1] | 0,9892 |
| 9 | $\ln x-x+1,8=0$ | [2;3] | 2,8459 |
| 10 | $2x+\sin x-1=0$ | [0;1] | 0,3354 |

Примечание: если указанное уравнение невозможно решить каким-либо из методов вследствие ограничений его применимости, то программу протестировать на любом из уравнений других вариантов.

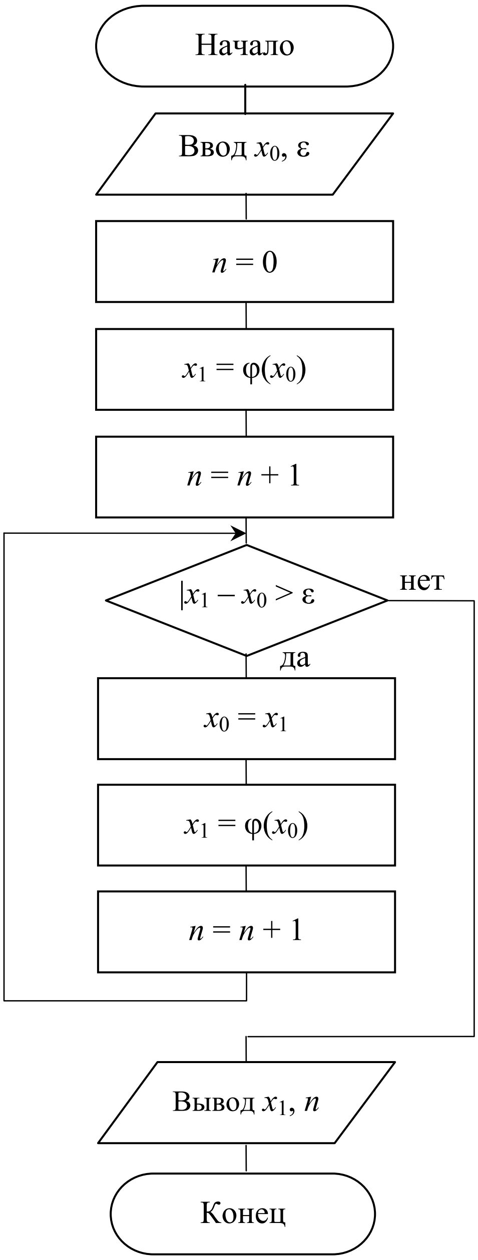



Рис. 1. Блок-схема алгоритма нахождения корня уравнения с точностью ε методом простых итераций ($x_0$ и $x_1$ – предыдущее и последующее приближения к корню соответственно, $n$ – счетчик числа итераций)

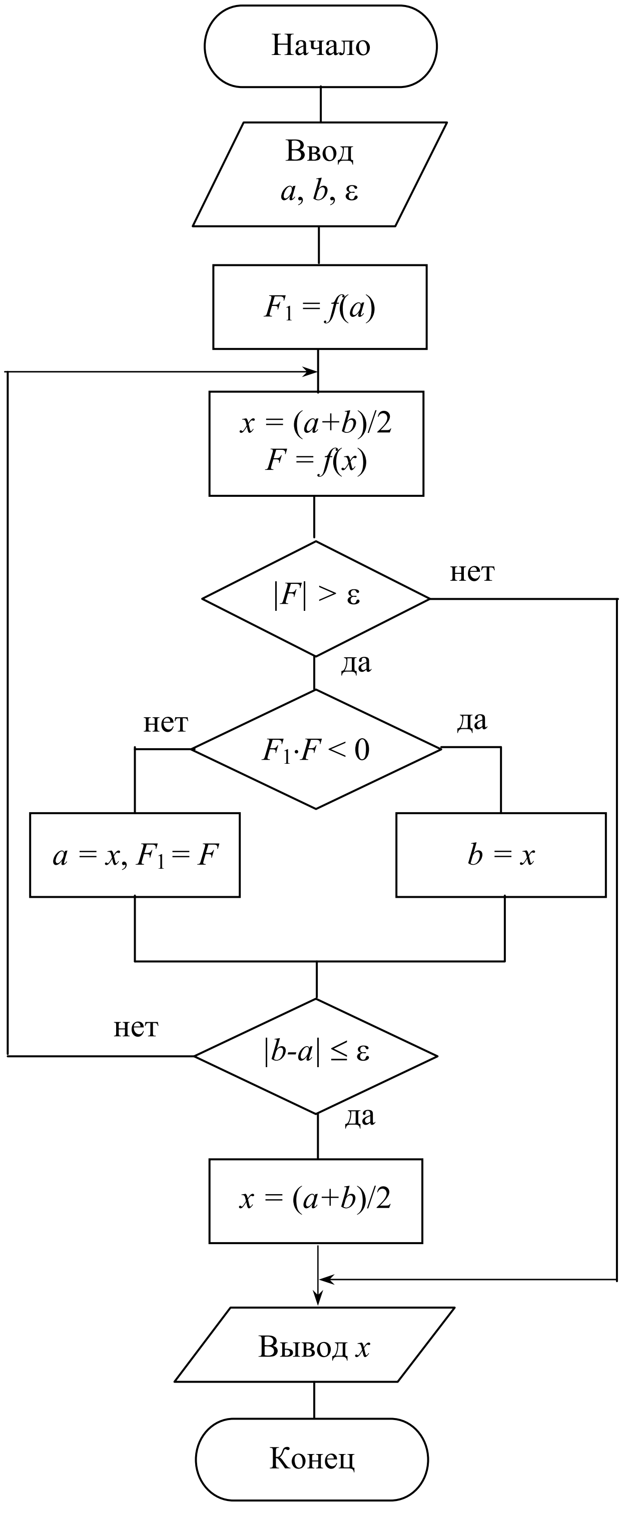



Рис. 2. Блок-схема алгоритма нахождения корня уравнения с точностью ε методом бисекции

## Решение нелинейных уравнений в программе Mathcad

### Решение уравнений вида f(x)=0 с помощью функции `root(f(x),x)`

Для численного поиска корней уравнений вида f(x)=0 в программе Mathcad используется функция `root(f(x), x)`. Перед вызовом функции `root` необходимо присвоить искомой переменной `x` начальное значение.

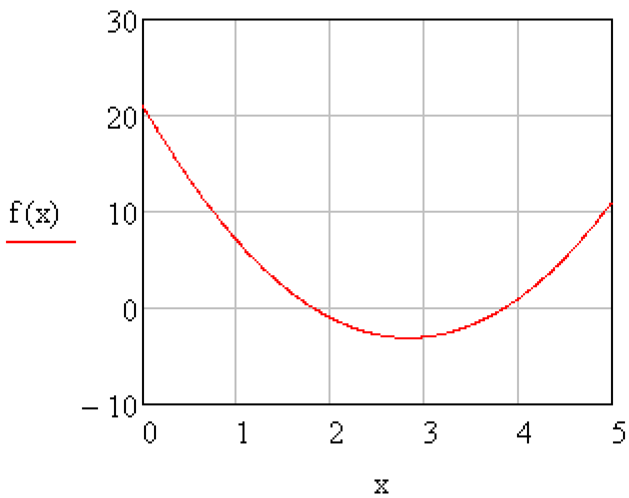
Если уравнение имеет несколько корней, то результат зависит от выбранного начального приближения.

***Пример***: решить уравнение $3(x-3)^2 + x - 6 = 0$.

***Решение***: Для определения начального приближения корня можно построить график функции `f(x)`.

$$f(x) := 3 \cdot (x-3)^2 + x - 6$$

$$x := 0, 0.01 .. 5$$




Нахождение первого корня уравнения:

$$x := 1$$

$$x1 := root(f(x), x)$$

$$x1 = 1.82$$

Нахождение второго корня уравнения:

$$x := 4$$

$$x2 := root(f(x), x)$$

$$x2 = 3.847$$

Проверка:

$$f(x1) = 5.353 \times 10^{-11}$$

$$f(x2) = 1.51 \times 10^{-14}$$

**Нахождение экстремума функции с помощью функции root**

$$x := 1$$

$$x2 := \mathrm{root}\left(\frac{d}{dx} f(x), x\right)$$

$$x2 = 2.833$$

$$f(x2) = -3.083$$

***Контрольные вопросы по лабораторной работе № 1***

1. Сформулируйте задачу о нахождении корней нелинейного уравнения.
2. Опишите метод простых итераций для вычисления корней нелинейного уравнения.
3. Сформулируйте условие сходимости метода простых итераций для нелинейного уравнения.
4. Опишите метод половинного деления (бисекции) для вычисления корней нелинейного уравнения.
5. Сформулируйте критерии остановки итерационного вычислительного процесса при определении корней нелинейного уравнения.
6. В чем заключается алгоритм проверки правильности полученного решения нелинейного уравнения?
7. Опишите последовательность действий для решения нелинейных уравнений в программе Mathcad.

## Лабораторная работа № 2

## Методы решения систем линейных алгебраических уравнений

***Цели работы***:

– научиться использовать методы, языки и системы программирования для решения систем линейных алгебраических уравнений;

– научиться использовать математический пакет Mathcad для решения систем линейных алгебраических уравнений.

***Задание***.

1. Решить систему уравнений методом Гаусса с выбором главного элемента в столбце на языке программирования PascalABC.NET (в программе должна быть предусмотрена проверка правильности полученных решений). Блок-схемы алгоритма решения системы уравнений методом Гаусса с выбором главного элемента в столбце приведены на рис. 3–8 [1].

2. Решить систему уравнений в математическом пакете Mathcad [3].

***Варианты заданий***

| | |
|---|---|
| 1. $\begin{cases} 5x_1 - 3x_2 + 4x_3 = 11 \\ 2x_1 - x_2 - 2x_3 = -6 \\ 3x_1 - 2x_2 + x_3 = 2 \end{cases}$ | 2. $\begin{cases} 3x_1 - x_2 + x_3 = 4 \\ x_1 + 2x_2 - x_3 = 4 \\ 2x_1 + x_2 + 2x_3 = 16 \end{cases}$ |
| 3. $\begin{cases} 3x_1 + x_2 + x_3 = -2 \\ 5x_1 - x_2 - x_3 = 10 \\ x_1 - x_2 + 5x_3 = -12 \end{cases}$ | 4. $\begin{cases} x_1 + 2x_2 - x_3 = -3 \\ 2x_1 + 3x_2 + x_3 = -1 \\ x_1 - x_2 - x_3 = 3 \end{cases}$ |
| 5. $\begin{cases} x_1 + 2x_2 - 3x_3 = 13 \\ 3x_1 + 2x_2 + 2x_3 = 16 \\ 4x_1 - 2x_2 + 5x_3 = 5 \end{cases}$ | 6. $\begin{cases} x_1 - 2x_2 + 4x_3 = 6 \\ 2x_1 - x_2 + 3x_3 = 11 \\ 4x_1 + x_2 - 5x_3 = 9 \end{cases}$ |
| 7. $\begin{cases} x_1 + 2x_2 - x_3 = 9 \\ 2x_1 - x_2 + 3x_3 = 13 \\ 3x_1 + 2x_2 - 5x_3 = -1 \end{cases}$ | 8. $\begin{cases} 2x_1 - 3x_2 + x_3 = 2 \\ 2x_1 + x_2 - 4x_3 = 9 \\ 6x_1 - 5x_2 + 2x_3 = 17 \end{cases}$ |
| 9. $\begin{cases} 2x_1 + x_2 - 3x_3 = -1 \\ x_1 - 3x_2 + 2x_3 = 10 \\ 3x_1 - 4x_2 - x_3 = 5 \end{cases}$ | 10. $\begin{cases} 3x_1 - 2x_2 + x_3 = 10 \\ x_1 + 5x_2 - 2x_3 = -15 \\ 2x_1 - 2x_2 - x_3 = 3 \end{cases}$ |

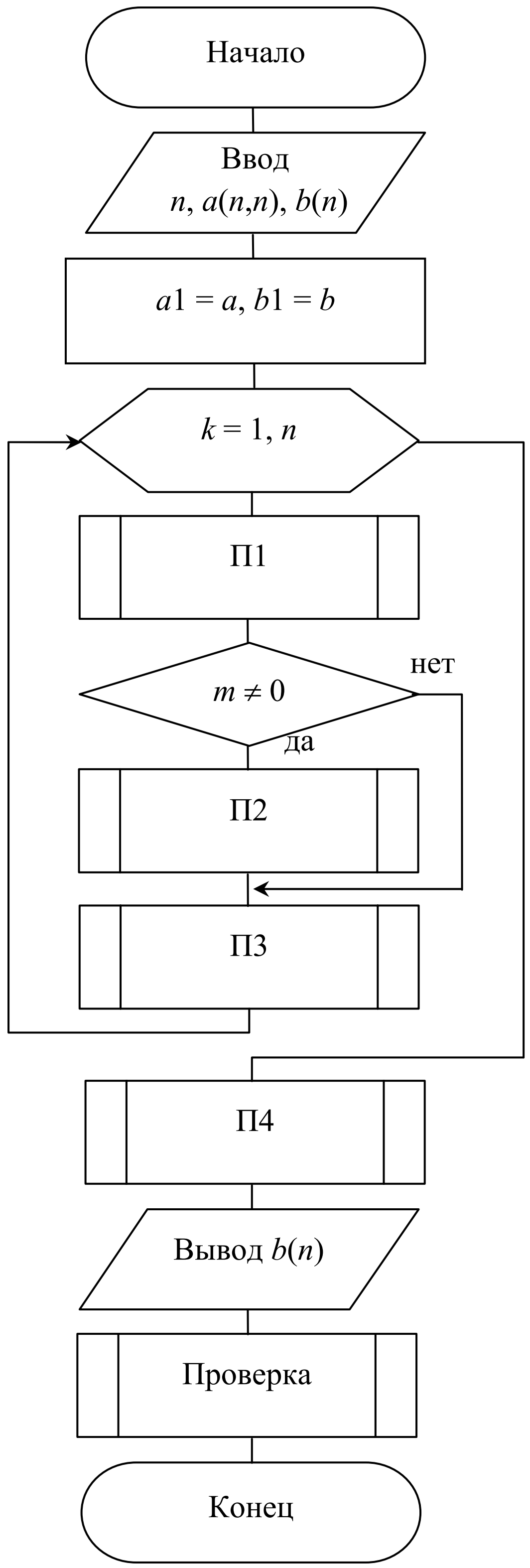



Рис. 3. Блок-схема основной программы

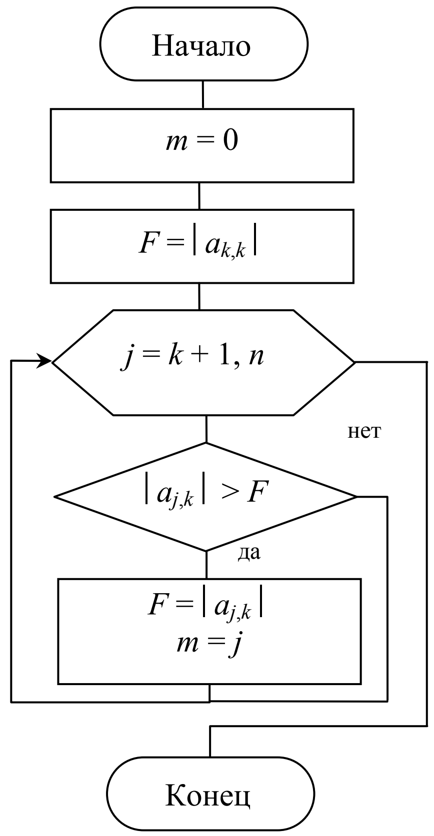



Рис. 4. Блок-схема подпрограммы (П1) поиска главного элемента в $k$-м столбце

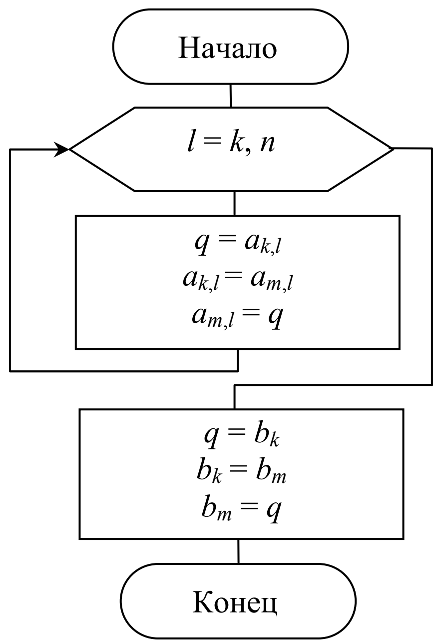



Рис. 5. Блок-схема подпрограммы (П2) перестановки $k$-го и $m$-го уравнений при $m \neq 0$

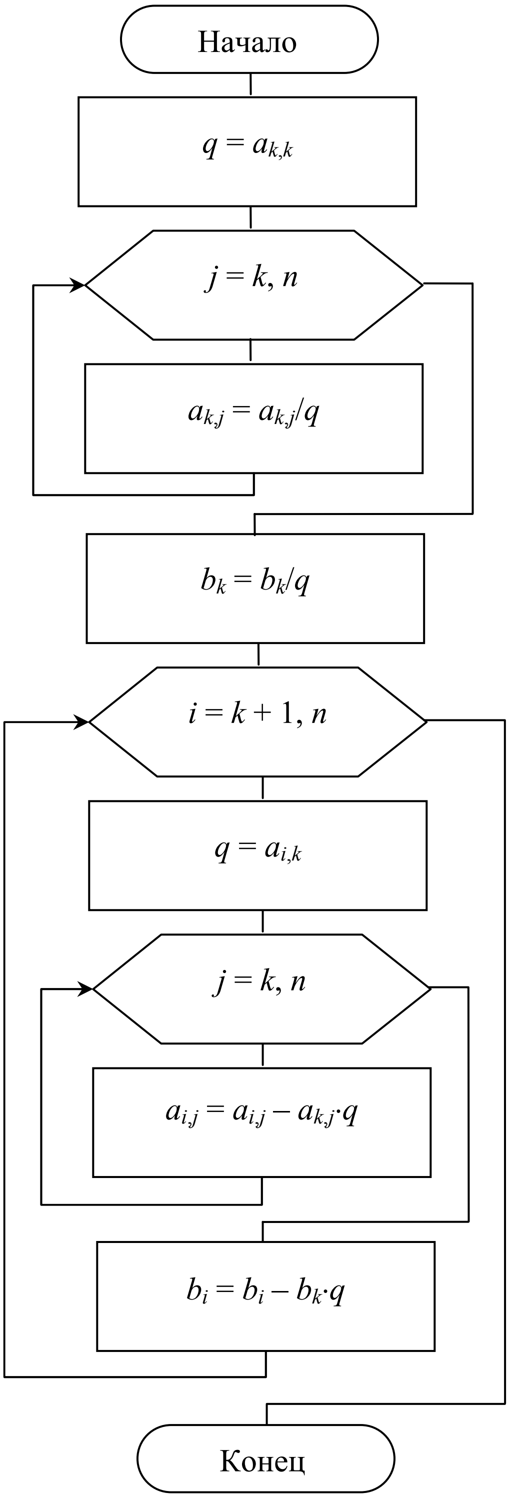



Рис. 6. Блок-схема подпрограммы (П3) прямого хода метода Гаусса (приведение системы к треугольному виду)

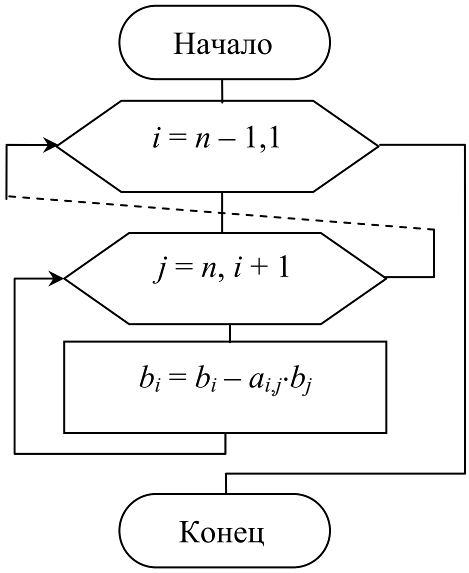



Рис. 7. Блок-схема подпрограммы (П4) обратного хода метода Гаусса

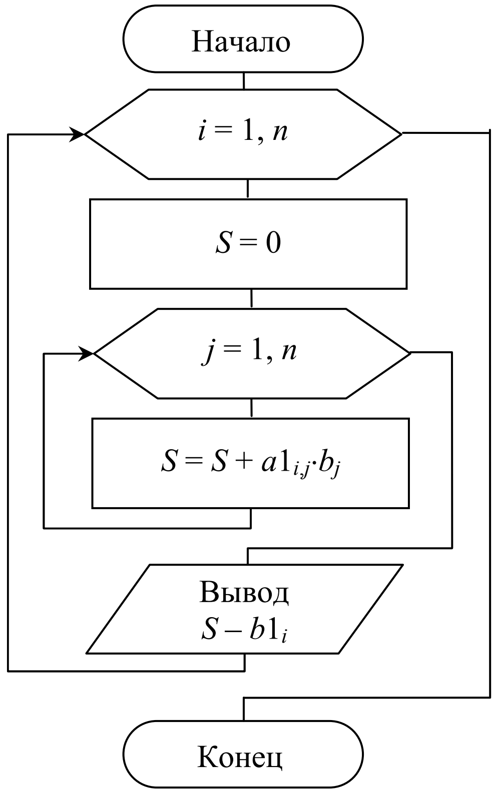



Рис. 8. Блок-схема подпрограммы (Проверка) проверки правильности решения

## Решение систем уравнений в программе Mathcad

### Решение системы линейных алгебраических уравнений с помощью функции `rref(C)`

В программе Mathcad решение системы уравнений методом Гаусса позволяет реализовать встроенная функция `rref(C)`, которая преобразует матрицу `C` в ступенчатый вид.

При решении системы уравнений с помощью функции `rref(C)` необходимо также воспользоваться еще двумя функциями: `augment(A,B)` и `submatrix(A,ir, jr,ic,jc)`.

Функция `augment(A,B)` объединяет матрицы `A` и `B` в одну новую, при этом матрица B располагается справа от матрицы `A`, и обе матрицы должны иметь одинаковое число строк.

Функция `submatrix (A,ir,jr,ic,jc)` формирует матрицу, которая является блоком матрицы `A`, расположенным в строках с `ir` по `jr` и в столбцах с `ic` по `jc`.

Номер первой строки (столбца) матрицы хранится в Mathcad в переменной `ORIGIN`. По умолчанию в Mathcad столбцы и строки матрицы нумеруются, начиная с 0, т.е. `ORIGIN=0`. Поскольку в математической записи чаще используется нумерация с 1, здесь и в дальнейшем перед началом работы с матрицами необходимо присвоить переменной `ORIGIN` значение 1, т.е. выполнить команду `ORIGIN:=1`.

***Пример.*** Решить систему уравнений

$$\begin{cases} 3x_1 - x_2 + x_3 = 4, \\ x_1 + 2x_2 - x_3 = 4, \\ 2x_1 + x_2 + 2x_3 = 16. \end{cases}$$

***Решение.***

1. Присвоим переменной `ORIGIN` значение 1.

$$\mathrm{ORIGIN} := 1$$

2. Запишем матрицу коэффициентов при неизвестных системы уравнений и столбец свободных членов.

$$\mathbf{A} := \begin{pmatrix} 3 & -1 & 1 \\ 1 & 2 & -1 \\ 2 & 1 & 2 \end{pmatrix} \qquad \mathbf{b} := \begin{pmatrix} 4 \\ 4 \\ 16 \end{pmatrix}$$

3. Сформируем расширенную матрицу системы – объединим `A` и `b` в одну матрицу `C` с помощью функции `augment(A,B)`.

$$\mathbf{C} := \mathbf{augment}(\mathbf{A}, \mathbf{b})$$

$$C = \begin{pmatrix} 3 & -1 & 1 & 4 \\ 1 & 2 & -1 & 4 \\ 2 & 1 & 2 & 16 \end{pmatrix}$$

4. Приведем расширенную матрицу системы `C` к ступенчатому виду, используя функцию `rref(C)`.

$$\mathbf{D} := \mathbf{rref}(\mathbf{C})$$

$$D = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 5 \end{pmatrix}$$

5. Получим вектор-столбец решения системы уравнений, выделив из матрицы `D` последний столбец с помощью функции `submatrix(A,ir,jr,ic,jc)`.

$$\mathbf{x} := \mathbf{submatrix}(\mathbf{D}, 1, 3, 4, 4)$$

$$\mathbf{x} = \begin{pmatrix} 1 \\ 4 \\ 5 \end{pmatrix}$$

6. Выполним проверку полученного решения:

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

## Решение системы уравнений с помощью вычислительного блока `Given ... Find`

Для решения системы уравнений (неравенств), в том числе и нелинейных, используют *блок решения*, который начинается с ключевого слова (дано) и заканчивается вызовом функции `Find` (найти) в виде `Find(x1,x2)`, где `x1` и `x2` – неизвестные. Между `Given` и `Find` располагают уравнения (неравенства), входящие в систему. При этом между левой и правой частями уравнений должен стоять знак логического равенства с панели инструментов *Булева алгебра*. Перед решением системы уравнений необходимо задать начальные значения для всех неизвестных.

***Пример***. Решить систему уравнений:

$$\begin{cases} x_1^2 + x_2^2 = 28, \\ x_1 - x_2 = 4. \end{cases}$$

***Решение***.

1. Зададим начальные значения для неизвестных.

$$x1 := 1$$

$$x2 := 0$$

2. Запишем блок решения.

Given

$$x1^2 + x2^2 = 28$$

$$x1 - x2 = 4$$

$$\mathrm{Find}(x1, x2) = \begin{pmatrix} 5.162 \\ 1.162 \end{pmatrix}$$

Если необходимо будет выполнить проверку полученного решения, то для его нахождения используем новые обозначения для неизвестных, как показано ниже:

$$x1 := 1$$

$$x2 := 0$$

Given

$$x1^2 + x2^2 = 28$$

$$x1 - x2 = 4$$

$$\begin{pmatrix} x11 \\ x22 \end{pmatrix} := \mathrm{Find}(x1, x2)$$

$$\begin{pmatrix} x11 \\ x22 \end{pmatrix} = \begin{pmatrix} 5.162 \\ 1.162 \end{pmatrix}$$

Проверка полученного решения:

$$x11^2 + x22^2 = 28$$

$$x11 - x22 = 4$$

Для нахождения второго решения заданной системы уравнений надо задать новые начальные значения для неизвестных и снова использовать блок решения:

$$x1 := -1$$

$$x2 := 0$$

Given

$$x1^2 + x2^2 = 28$$

$$x1 - x2 = 4$$

$$\mathrm{Find}(x1, x2) = \begin{pmatrix} -1.162 \\ -5.162 \end{pmatrix}$$

***Контрольные вопросы по лабораторной работе № 2***

1. Опишите метод Гаусса с выбором главного элемента в столбце (строке) для решения систем линейных алгебраических уравнений.

2. В чем заключается алгоритм проверки правильности полученного решения системы линейных алгебраических уравнений?

3. Опишите последовательность действий для решения систем уравнений в программе Mathcad.

## Лабораторная работа № 3

## Решение задачи интерполяции

***Цели работы***:

– научиться использовать методы, языки и системы программирования для решения задачи интерполяции;

– научиться использовать математический пакет Mathcad для решения задачи интерполяции.

***Задание.***

1. Решить задачу интерполяции для заданной табличной зависимости на языке программирования PascalABC.NET. Блок-схема алгоритма решения задачи интерполяции (интерполяция алгебраическим многочленом) приведена на рис. 9 [1]. Записать зависимость $y = f(x)$ с полученными в результате расчета коэффициентами. Построить график полученной функции. В отчете также привести полученное значение функции $y$ в одном из табличных значений $x$ (проверка) и значение функции $y$ в промежуточном значении $x$.

2. Решить задачу интерполяции в математическом пакете Mathcad [3].

***Варианты заданий***

| Вариант 1 | | Вариант 2 | | Вариант 3 | | Вариант 4 | | Вариант 5 | |
|---|---|---|---|---|---|---|---|---|---|
| $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ |
| 51 | 1655 | 54 | 1176 | 12 | 932 | 19 | 946 | 150 | 512 |
| 52 | 1682 | 55 | 1733 | 13 | 964 | 20 | 909 | 151 | 506 |
| 53 | 1699 | 56 | 1751 | 14 | 986 | 21 | 863 | 152 | 501 |
| 54 | 1716 | 57 | 1751 | 15 | 997 | 22 | 809 | 153 | 495 |
| 55 | 1733 | 58 | 1786 | 16 | 999 | 23 | 746 | 154 | 489 |
| 56 | 1751 | 59 | 1804 | 17 | 992 | 24 | 675 | 155 | 484 |
| 57 | 1768 | 60 | 1822 | | | | | | |

| Вариант 6 | | Вариант 7 | | Вариант 8 | | Вариант 9 | | Вариант 10 | |
|---|---|---|---|---|---|---|---|---|---|
| $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ |
| 170 | 1232 | 13 | 129 | 12 | 89 | 25 | 276 | 100 | 2718 |
| 175 | 1210 | 18 | 179 | 13 | 66 | 35 | 379 | 105 | 2858 |
| 180 | 1179 | 23 | 228 | 14 | 47 | 45 | 475 | 110 | 3004 |
| 185 | 1139 | 28 | 276 | 15 | 34 | 55 | 563 | 115 | 3158 |
| 190 | 1089 | 33 | 324 | 16 | 23 | 65 | 642 | 120 | 3320 |
| 195 | 1028 | 38 | 371 | 17 | 16 | 75 | 711 | 125 | 3490 |
| 200 | 956 | 43 | 417 | 18 | 11 | | | | |

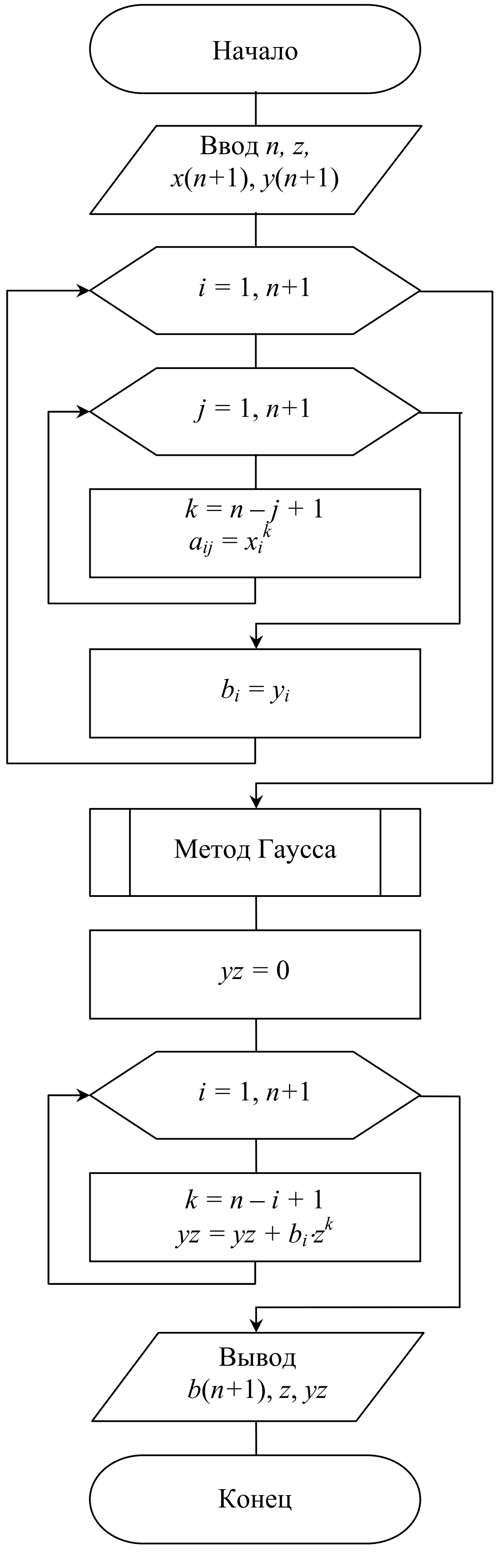



Рис. 9. Блок-схема алгоритма решения задачи интерполяции (интерполяция алгебраическим многочленом)

Блок-схема алгоритма решения задачи интерполяции алгебраическим многочленом приведена на рис. 9.

Решается система уравнений вида (в матричной форме):

$$A \cdot z = b,$$

где $A = \begin{pmatrix} x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \dots & x_2 & 1 \\ x_3^n & x_3^{n-1} & \dots & x_3 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ x_{n+1}^n & x_{n+1}^{n-1} & \dots & x_{n+1} & 1 \end{pmatrix}, \qquad z = \begin{pmatrix} a_n \\ a_{n-1} \\ a_{n-2} \\ \dots\dots \\ a_0 \end{pmatrix}, \qquad b = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \dots\dots \\ y_{n+1} \end{pmatrix},$

$A$ – матрица коэффициентов системы; $z$ – столбец неизвестных; $b$ – столбец свободных членов.

В блок-схеме алгоритма решения задачи интерполяции алгебраическим многочленом (рис. 9) использованы следующие обозначения: $a_{ij}$ – матрица коэффициентов системы уравнений; $b_i$ – вектор свободных членов, векторы $x$ и $y$ – исходные табличные данные; $z$ – значение аргумента $x$, для которого вычисляется промежуточное значение функции $yz$ после нахождения коэффициентов многочлена. В результате выполнения приведенного на рис. 9 алгоритма получаем коэффициенты многочлена: $a_n = b_1$, $a_{n-1} = b_2$, …, $a_0 = b_{n+1}$.

Полученные в результате значения $a_0$, $a_1$, …, $a_n$ подставляют в интерполирующую функцию (в данной лабораторной работе это алгебраический многочлен $P_n(x)$), что позволяет полностью ее определить и в дальнейшем использовать для приближенного вычисления значений функции при значениях аргумента, отличных от таблично заданных, а также исследовать методами математического анализа.

## Решение задачи интерполяции в программе Mathcad

### Линейная интерполяция в программе Mathcad

Для линейной интерполяции в Mathcad используется встроенная функция `linterp(X,Y,t)`, где `X` – вектор экспериментальных значений аргумента, расположенных в порядке возрастания; `Y` – вектор экспериментальных значений функции; `t` – значение аргумента, при котором вычисляется интерполирующее значение функции.

Если необходимо определить несколько интерполирующих значений функции, тогда `t` – вектор значений аргумента, а результат расчета – массив интерполирующих значений функции в этих точках. Может быть, `t` – переменная, тогда результат расчета – функция, которую можно далее интегрировать, дифференцировать и т.д.

***Пример 1.*** Решить задачу интерполяции для массива экспериментальных данных `(X,Y)`.

***Решение.*** Вектор `X` и вектор `Y` создаются с помощью панели инструментов *Матрица*.

$$X := \begin{pmatrix} 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \end{pmatrix} \qquad Y := \begin{pmatrix} 89 \\ 66 \\ 47 \\ 34 \\ 23 \\ 16 \\ 11 \end{pmatrix}$$

$\mathbf{n} := 6$ `(n+1)` – количество экспериментальных точек. Здесь `n=6`, так как в Mathcad по умолчанию отсчет идет от нуля. Отсчет от 1 можно реализовать с помощью переменной `ORIGIN`, присвоив ей значение 1.

1. $\mathbf{Y1 := linterp(X, Y, 15)}$ `Y1` – интерполирующее значение функции в одной точке `X=15`

   $Y1 = 34$

   $\mathbf{Y1 := linterp(X, Y, 15)}$ `Y1` – интерполирующее значение функции в одной точке `X=15.5`

   $Y1 = 28.5$

2. $m := 5$

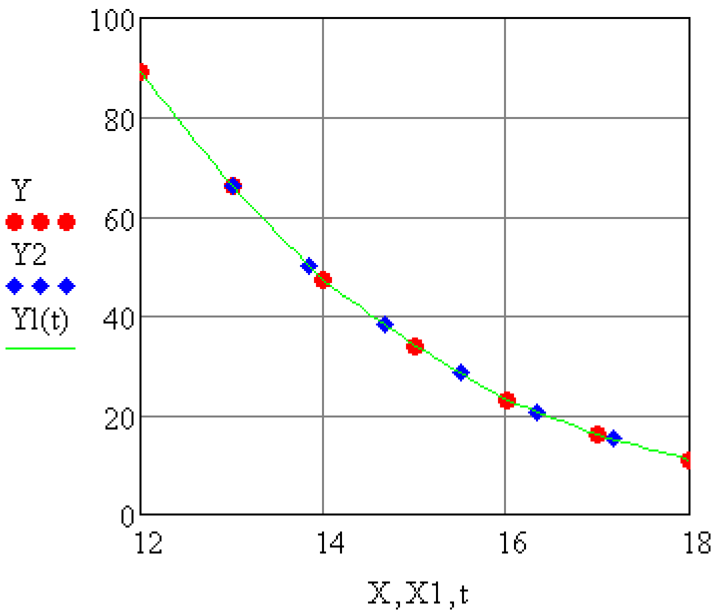
$j := 0..m$

$$X1_j := X_1 + j \cdot \frac{X_n - X_1}{m+1}$$

$Y2 := \mathrm{linterp}(X, Y, X1)$ `Y2` – массив интерполирующих значений функции в `m` точках значений `X`

3. $t := 12, 12.1 .. \max(X)$

$Y1(t) := \mathrm{linterp}(X, Y, t)$ `Y1(t)` – непрерывная интерполирующая функция




## Кубическая сплайн-интерполяция в программе Mathcad

В большинстве случаев желательно соединять экспериментальные точки не ломаной линией, а гладкой кривой (см. также пример 2), для чего используется сплайн-интерполяция. При этом кривая образуется путем создания ряда кубических полиномов, проходящих через наборы из трех соседних экспериментальных точек.

Для кубической сплайн-интерполяции используется встроенная функция

`interp(vs, X,Y,t)`,

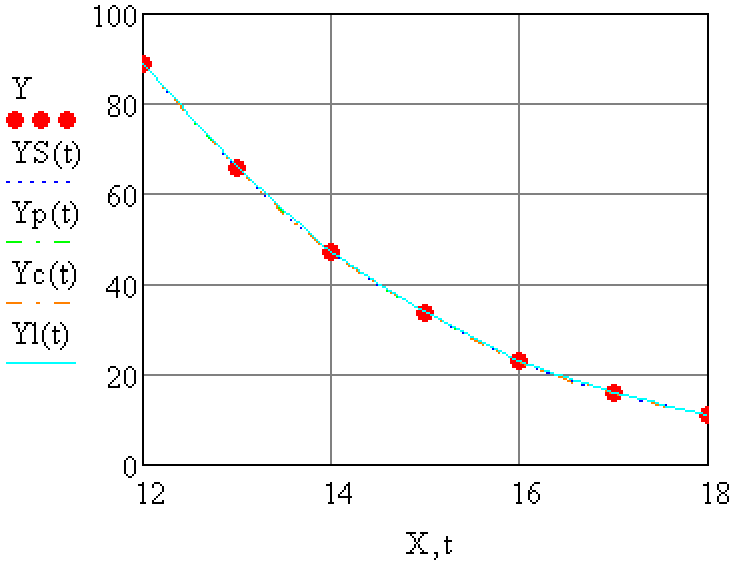
где `vs` – вектор вторых производных, созданный одной из функций: `lspline(X,Y)`, `pspline(X,Y)`, `cspline(X,Y)`; `X` – вектор экспериментальных значений аргумента, расположенных в порядке возрастания; `Y` – вектор экспериментальных значений функции; `t` – значение аргумента, при котором вычисляется интерполирующее значение функции.

***Кубическая сплайн-интерполяция для примера 1:***

$$YS(t) := interp(lspline(X,Y),X,Y,t)$$

$$Yp(t) := interp(pspline(X,Y),X,Y,t)$$

$$Yc(t) := interp(cspline(X,Y),X,Y,t)$$




***Пример 2.*** Решить задачу интерполяции для массива значений `(X,Y)`, заданных с помощью функции случайных чисел `rnd`.

***Решение***.
***Линейная интерполяция для примера 2:***

$$n := 10$$

$$i := 0..n$$

$$X_i := i$$

$$Y_i := rnd(10)$$

1. $Y1 := \mathrm{linterp}(X, Y, 5)$ — Y1 – интерполирующее значение функции в одной точке X=5

$Y1 = 1.741$

2. $m := 6$

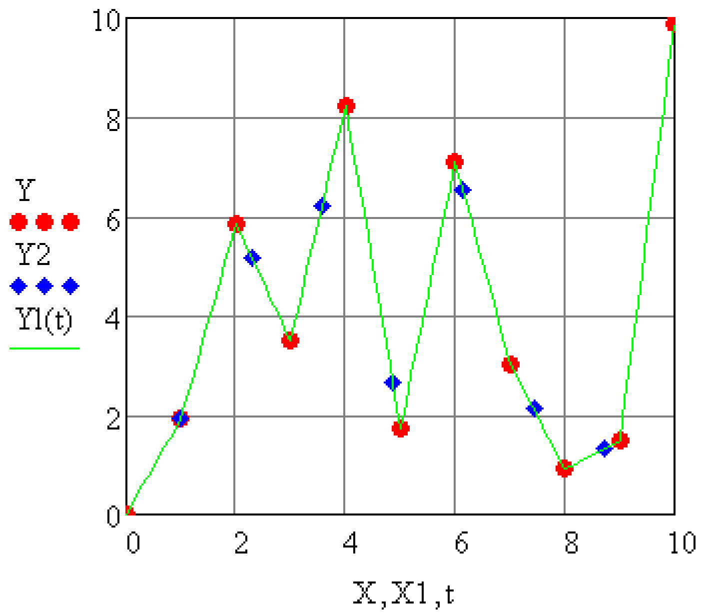
$j := 0 .. m$

$$X1_j := X_1 + j \cdot \frac{X_n - X_1}{m + 1}$$

$Y2 := \mathrm{linterp}(X, Y, X1)$ — Y2 – массив интерполирующих значений функции в m точках значений X

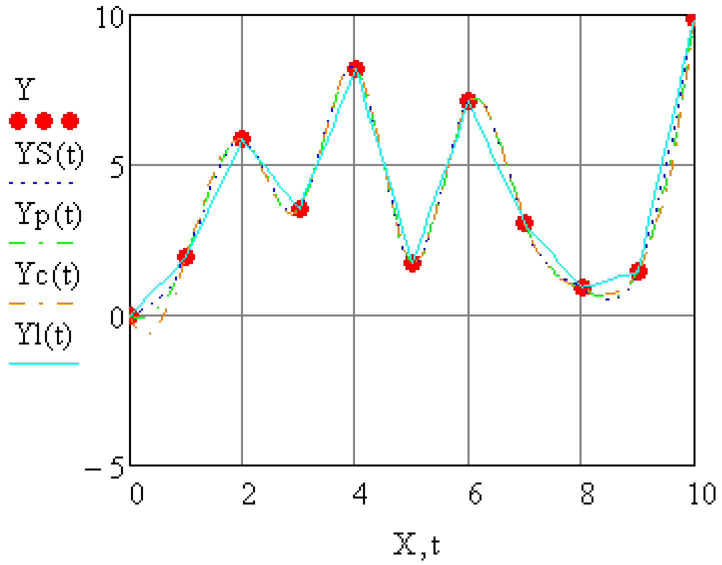
3. $t := 0, 0.1 .. \max(X)$

$Yl(t) := \mathrm{linterp}(X, Y, t)$ — Y1(t) – непрерывная интерполирующая функция



***Кубическая сплайн-интерполяция для примера 2:***

$$YS(t) := interp(lspline(X, Y), X, Y, t)$$
$$Yp(t) := interp(pspline(X, Y), X, Y, t)$$
$$Yc(t) := interp(cspline(X, Y), X, Y, t)$$




***Контрольные вопросы по лабораторной работе № 3***

1. Сформулируйте задачу интерполяции.
2. Укажите требования к интерполирующей функции.
3. В чем заключается алгоритм проверки правильности полученного решения задачи интерполяции?
4. Опишите последовательность действий для решения задачи интерполяции (линейной и кубической сплайн-интерполяции) в программе Mathcad.

## Лабораторная работа № 4

## Аппроксимация табличных зависимостей методом наименьших квадратов

***Цели работы***:

– научиться использовать методы, языки и системы программирования для решения задачи аппроксимации табличных зависимостей;

– научиться использовать математический пакет Mathcad для решения задачи аппроксимации табличных зависимостей.

***Задание.***

1. Аппроксимировать табличную зависимость методом наименьших квадратов на языке программирования PascalABC.NET. Блок-схема алгоритма аппроксимации табличных зависимостей многочленом степени $m$ для $n$ точек методом наименьших квадратов приведена на рис. 10 [1]. Записать зависимость $y = f(x)$ с полученными в результате расчета коэффициентами. В программе должна быть предусмотрена проверка правильности полученных решений (определить разность между табличными значениями $y$ и значениями функции, вычисленными в табличных значениях $x$ по полученной зависимости $y = f(x)$).

2. Решить задачу аппроксимации табличной зависимости в математическом пакете Mathcad [3].

***Варианты заданий***

| Вариант 1 | | Вариант 2 | | Вариант 3 | | Вариант 4 | | Вариант 5 | |
|---|---|---|---|---|---|---|---|---|---|
| *x* | *y* | *x* | *y* | *x* | *y* | *x* | *y* | *x* | *y* |
| 0,2 | 9,9 | 0,2 | 6,9 | 0,2 | 6,9 | 0,8 | 36,4 | 0,2 | 0,017 |
| 0,4 | 5,1 | 0,4 | 4,6 | 0,4 | 2,1 | 0,9 | 27,68 | 0,4 | 0,072 |
| 0,6 | 3,2 | 0,6 | 3,57 | 0,6 | 0,23 | 1,0 | 22,07 | 0,6 | 0,199 |
| 0,8 | 2,6 | 0,8 | 3,35 | 0,8 | –0,4 | 1,5 | 11,48 | 0,8 | 0,276 |
| 1,0 | 1,9 | 1,0 | 2,9 | 1,0 | –1,1 | 2,0 | 8,05 | 1,0 | 0,037 |
| 2,0 | 1,1 | 2,0 | 2,6 | 2,0 | –1,9 | 2,5 | 6,78 | 2,0 | 0,6 |
| 4,0 | 0,4 | 4,0 | 2,15 | 4,0 | –2,6 | 3,0 | 5,74 | 4,0 | 0,789 |
| 6,0 | 0,43 | 6,0 | 2,27 | 6,0 | –2,57 | 3,5 | 5,41 | 6,0 | 0,836 |
| 8,0 | 0,15 | 8,0 | 2,03 | 8,0 | –2,85 | 4,0 | 4,85 | 8,0 | 0,982 |
| 10,0 | 0,3 | 10,0 | 2,2 | 10,0 | –2,7 | 4,5 | 4,78 | 10,0 | 0,905 |

| Вариант 6 | | Вариант 7 | | Вариант 8 | | Вариант 9 | | Вариант 10 | |
|---|---|---|---|---|---|---|---|---|---|
| *x* | *y* | *x* | *y* | *x* | *y* | *x* | *y* | *x* | *y* |
| 0,6 | 11,21 | 0,2 | –2,32 | 0,2 | 2,12 | 0,2 | –1,21 | 0,6 | –2,32 |
| 0,7 | 8,06 | 0,4 | –0,73 | 0,4 | 0,93 | 0,4 | –0,32 | 0,7 | –0,73 |
| 0,8 | 6,35 | 0,6 | –0,12 | 0,6 | 0,02 | 0,6 | –0,01 | 0,8 | –0,12 |
| 0,9 | 4,84 | 0,8 | 0,65 | 0,8 | –0,45 | 0,8 | 0,38 | 0,9 | 0,65 |
| 1,0 | 4,1 | 1,0 | 0,9 | 1,0 | –1,1 | 1,0 | 0,5 | 1,0 | 0,9 |
| 2,0 | 0,9 | 3,0 | 3,3 | 3,0 | –3,1 | 3,0 | 1,7 | 2,0 | 3,3 |
| 3,0 | 0,54 | 5,0 | 4,12 | 5,0 | –4,32 | 5,0 | 2,01 | 3,0 | 4,12 |
| 4,0 | 0,15 | 7,0 | 4,98 | 7,0 | –4,79 | 7,0 | 2,55 | 4,0 | 4,98 |
| 5,0 | 0,26 | 9,0 | 5,29 | 8,0 | –5,26 | 8,0 | 2,48 | 5,0 | 5,29 |
| 6,0 | 0,01 | 11,0 | 5,9 | 9,0 | –5,29 | 9,0 | 2,79 | 6,0 | 5,9 |

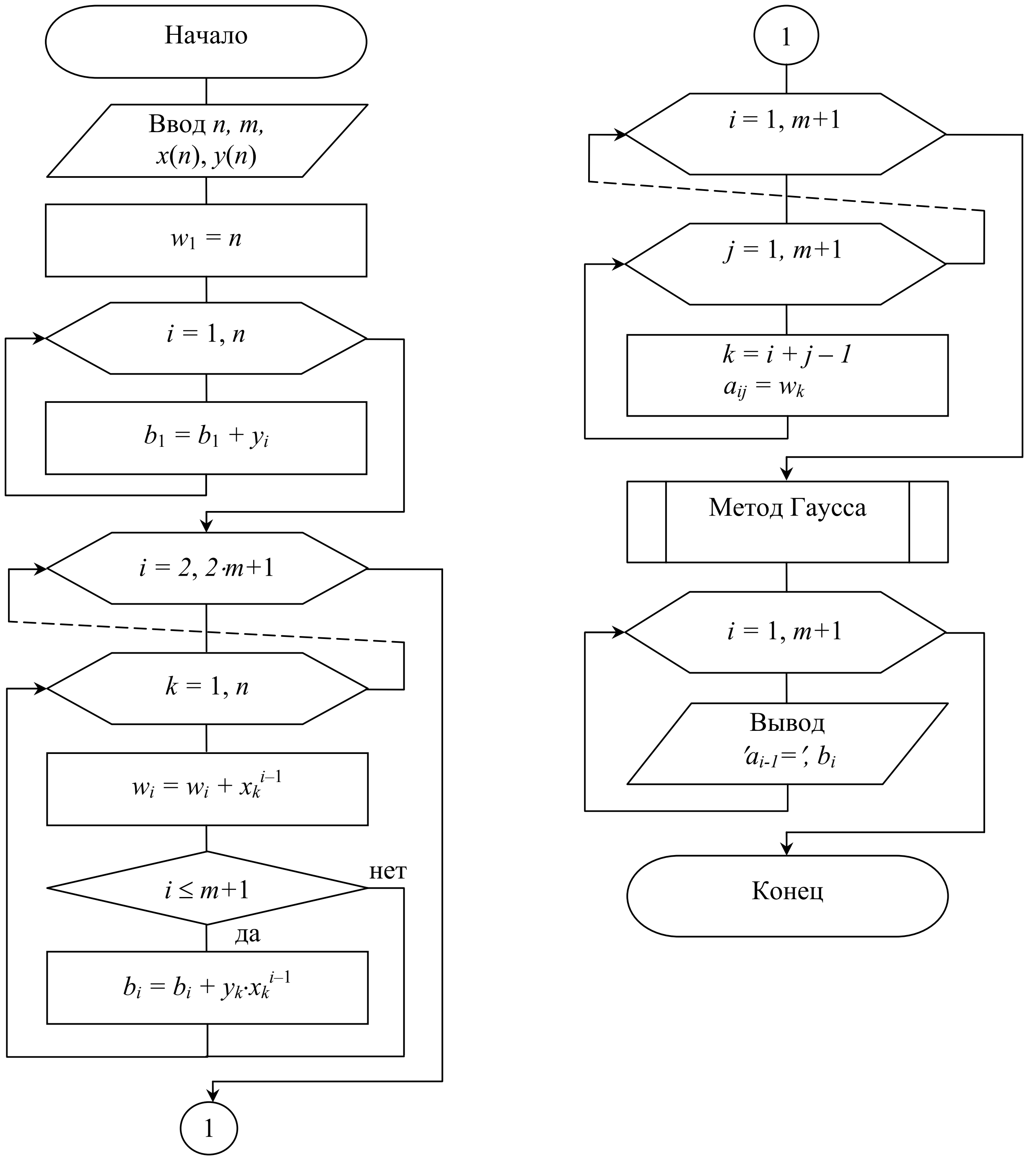



Рис. 10. Блок-схема алгоритма аппроксимации табличных зависимостей многочленом степени *m* для *n* точек методом наименьших квадратов

Блок-схема алгоритма аппроксимации табличных зависимостей многочленом степени m для n точек методом наименьших квадратов приведена на рис. 10.

Решается система уравнений вида (в матричной форме):

$$A \cdot z = b,$$

где

$$A = \begin{pmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \dots & \sum_{i=1}^{n} x_i^m \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \dots & \sum_{i=1}^{n} x_i^{m+1} \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 & \dots & \sum_{i=1}^{n} x_i^{m+2} \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ \sum_{i=1}^{n} x_i^m & \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m+2} & \dots & \sum_{i=1}^{n} x_i^{2\cdot m} \end{pmatrix}, \quad z = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \\ a_m \end{pmatrix}, \quad b = \begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i \cdot x_i \\ \sum_{i=1}^{n} y_i \cdot x_i^2 \\ \dots\dots\dots \\ \sum_{i=1}^{n} y_i \cdot x_i^m \end{pmatrix},$$

$A$ – матрица коэффициентов системы; $z$ – столбец неизвестных; $b$ – столбец свободных членов.

В связи с тем, что матрица коэффициентов системы уравнений $a_{ij}$ содержит попарно повторяющиеся элементы, для ее формирования используется вспомогательный массив $w$ – вектор с количеством элементов, равным $2m+1$. Кроме этого, в блок-схеме использованы следующие обозначения: векторы $x$ и $y$ – исходные табличные данные; $n$ – количество точек таблицы; $m$ – степень многочлена (аппроксимирующей функции); $b_i$ – вектор свободных членов. В результате выполнения приведенного на рис. 10 алгоритма получаем коэффициенты многочлена: $a_0 = b_1$, $a_1 = b_2$, …, $a_m = b_{m+1}$.

Полученные в результате значения $a_0$, $a_1$, …, $a_m$ подставляют в аппроксимирующую функцию (в данной лабораторной работе это алгебраический многочлен $P_m(x)$), что позволяет полностью ее определить и в дальнейшем использовать для приближенного вычисления значений функции при значениях аргумента, отличных от таблично заданных, а также исследовать методами математического анализа.

## Аппроксимация табличных зависимостей в программе Mathcad

При проведении различных экспериментов часто требуется массив экспериментальных данных (табличную зависимость) представить в виде функции, которую можно использовать в дальнейших расчетах. Если кривая, описываемая этой функцией, не должна проходить через все экспериментальные точки и является аппроксимацией (усреднением) исходных данных, операция получения промежуточных точек и расчетной функции называется регрессией. Регрессия сводится к подбору коэффициентов в той или иной аналитической зависимости.

В программе Mathcad есть несколько встроенных функций регрессии двух типов:

1) позволяющих увидеть аналитическую зависимость, т.е. возвращающих набор коэффициентов аппроксимирующей функции;

2) не позволяющих увидеть аналитическую зависимость.

Рассмотрим функцию 2-го типа, которая не выводит коэффициентов и аппроксимирует массив данных одним многочленом степени `n`. В программе Mathcad это реализуется комбинацией встроенных функций интерполяции и регрессии:

```
interp(s,X,Y,t)
regress(X,Y,n)
```

Здесь `X` – вектор экспериментальных значений аргумента, расположенных в порядке возрастания; `Y` – вектор экспериментальных значений функции; `s` – вектор коэффициентов для построения аппроксимирующего многочлена, создаваемый функцией `regress`; `t` – значение аргумента, при котором вычисляется значение функции; `n` – степень аппроксимирующего многочлена.

***Пример***. Решить задачу аппроксимации для массива значений `(X,Y)`, заданных с помощью функции случайных чисел `rnd`.

***Решение***.

$k := 20$

$i := 0..k$

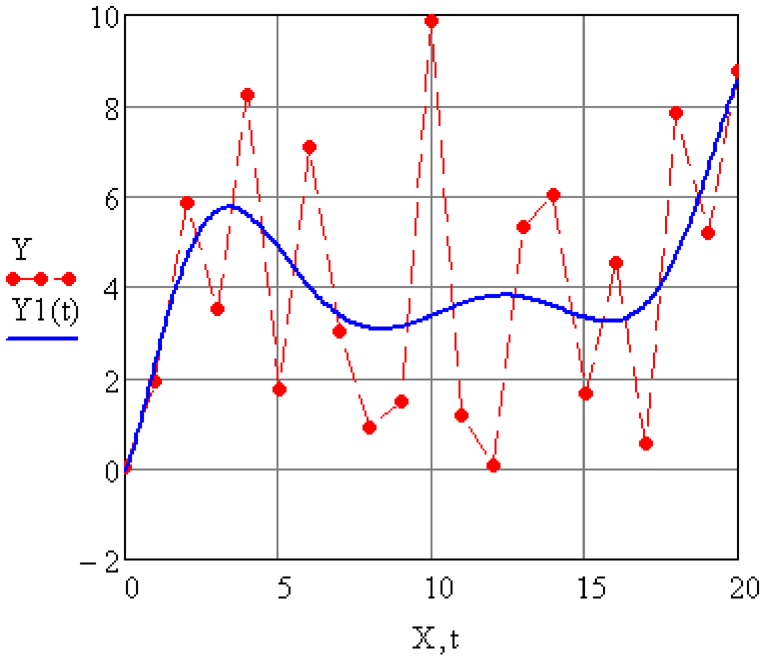
$X_i := i$

$Y_i := \mathrm{rnd}(10)$

$m := 9$

$Y1(t) := \text{interp}(\text{regress}(X, Y, m), X, Y, t)$

$t := 0, 0.1 .. \max(X)$




Поменяйте m – степень многочлена и проанализируйте изменения вида аппроксимирующей функции.

***Контрольные вопросы по лабораторной работе № 4***

1. Сформулируйте задачу аппроксимации табличных зависимостей.
2. Укажите требования к аппроксимирующей функции.
3. Опишите метод наименьших квадратов для аппроксимации таблично заданной функции.
4. В чем заключается алгоритм проверки правильности полученного решения задачи аппроксимации табличных зависимостей?
5. Опишите последовательность действий для аппроксимации табличных зависимостей в программе Mathcad.

# Лабораторная работа № 5

## Численные методы решения дифференциальных уравнений и их систем

***Цели работы:***

– научиться использовать численные методы, языки и системы программирования для решения обыкновенных дифференциальных уравнений и их систем;

– научиться использовать математический пакет Mathcad для решения обыкновенных дифференциальных уравнений и их систем.

***Задание.***

1. Решить дифференциальное уравнение методом Эйлера на языке программирования PascalABC.NET. Блок-схема алгоритма решения дифференциального уравнения 1-го порядка методом Эйлера приведена на рис. 11 [1]. В программе должна быть предусмотрена проверка правильности полученных решений (сравнить полученные значения искомой функции с точным решением, приведенным в таблице заданий).

2. Решить дифференциальное уравнение методом Рунге–Кутты в математическом пакете Mathcad с помощью функций `rkfixed` и `odesolve` [3].

***Варианты заданий***

| № варианта | Дифференциальное уравнение | Начальные условия | Отрезок интегрирования $x_0$; $x_k$ | Шаг интегрирования $\Delta x$ | Точное решение $y=f(x)$ |
|---|---|---|---|---|---|
| 1 | $y'' + y = 1 + e^x$ | $y_o = 2,5$<br>$y_o' = 1,5$ | 0; 1 | 0,05 | $f(x) = \cos x + \sin x + 1 + \frac{1}{2}e^x$ |
| 2 | $y''+4y = \cos 3x$ | $y_o = 0,8$<br>$y_o' = 2$ | 0; 1 | 0,05 | $f(x) = \cos 2x + \sin 2x - 0,2\cos 3x$ |
| 3 | $y''-y'-6y = 2e^{4x}$ | $y_o = 1,433$<br>$y_o' = -0,367$ | 0; 1 | 0,05 | $f(x) = 0,1e^{3x} + e^{-2x} + \frac{1}{3}e^{4x}$ |
| 4 | $y''-2y'+y = 5xe^x$ | $y_o = 1$<br>$y_o' = 2$ | 0; 1 | 0,05 | $f(x) = e^x + xe^x + \frac{5}{6}x^3e^x$ |
| 5 | $y''+y'-6y = 3x^2 - x - 1$ | $y_o = -0,9$<br>$y_o' = 3,2$ | 0; 1 | 0,05 | $f(x) = 0,1e^{2x} - e^{-3x} - 0,5x^2$ |
| 6 | $8y''+2y'-3y = x + 5$ | $y_o = 1/9$<br>$y_o' = -7/12$ | 0; 1 | 0,05 | $f(x) = e^{\frac{x}{2}} + e^{-\frac{3x}{4}} - \frac{x}{3} - \frac{17}{9}$ |

| | | | | | |
|---|---|---|---|---|---|
| 7 | $y''-4y'+5y=3x$ | $y_o=1,48$ <br> $y_o'=3,6$ | 0; 0,5 | 0,025 | $f(x)=e^{2x}(\cos x+\sin x)+\dfrac{3}{5}x+\dfrac{12}{25}$ |
| 8 | $y''+4y'+4y=0$ | $y_o=1$ <br> $y_o'=-1$ | 0; 1 | 0,05 | $f(x)=(1+x)\,e^{-2x}$ |
| 9 | $(1+x^2)y''+(y')^2+1=0$ | $y_o=1$ <br> $y_o'=1$ | 0; 0,5 | 0,025 | $f(x)=1-x+2\ln(1+x)$ |
| 10 | $y''+2y'+2y=2e^{-x}\cos x$ | $y_o=1$ <br> $y_o'=0$ | 0; 1 | 0,05 | $f(x)=e^{-x}(\cos x+\sin x+x\sin x)$ |

Примечание: заданное дифференциальное уравнение 2-го порядка при решении численным методом Эйлера необходимо привести к системе дифференциальных уравнений 1-го порядка. При этом формула метода Эйлера используется для каждого из полученных уравнений. Совместная система уравнений на каждом шаге интегрирования решается одновременно.

Алгоритм решения дифференциального уравнения 1-го порядка методом Эйлера представляет собой циклический процесс вычислений искомой функции $y$ по формуле метода Эйлера при изменении аргумента $x$ от $x_0$ до $x_k$ с шагом $h$ (рис. 11).

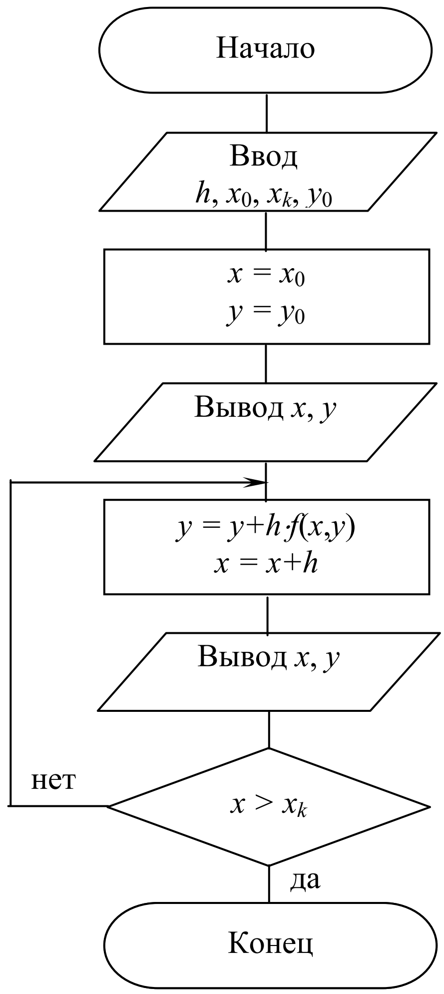



Рис. 11. Блок-схема алгоритма решения дифференциального уравнения 1-го порядка методом Эйлера

## Решение дифференциальных уравнений в программе Mathcad

Для численного решения обыкновенных дифференциальных уравнений и их систем в программе Mathcad существует ряд функций, среди них `rkfixed` и `odesolve`.

Функция `rkfixed(y,x0,xk,k,F)` – возвращает матрицу решений методом Рунге–Кутты системы обыкновенных дифференциальных уравнений с начальными условиями в векторе `y`, первые производные от неизвестных функций (правые части уравнений) записаны в векторе `F`, на интервале от `x0` до `xk`; `k` – число шагов. Функция `rkfixed` ищет решение с постоянным шагом.

***Пример***. Решить дифференциальное уравнение 2-го порядка $y'' - 2 \cdot y + y = 5 \cdot x \cdot e^x$ на отрезке [0;1] с шагом интегрирования 0.1 и начальными условиями $y_o = 1, \quad y_o^{'} = 2$ с использованием функции `rkfixed`.

***Решение***.

$\mathbf{y} := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ вектор начальных значений функции, размерность вектора соответствует числу решаемых уравнений

$\mathbf{x0} := 0 \qquad \mathbf{xk} := 1$ `x0`, `xk` – начальное и конечное значение отрезка интегрирования

$\mathbf{h} := 0.1$ `h` – шаг интегрирования

$\mathbf{k} := \dfrac{\mathbf{xk} - \mathbf{x0}}{\mathbf{h}}$ `k` – число шагов интегрирования

$\mathbf{F(x,y)} := \begin{pmatrix} y_1 \\ 5x \cdot e^x - y_0 + 2y_1 \end{pmatrix}$ `F(x,y)` – функция правых частей дифференциальных уравнений

$Z := rkfixed(y, x0, xk, k, F)$

Z =

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 1 | 2 |
| 1 | 0.1 | 1.217 | 2.349 |
| 2 | 0.2 | 1.474 | 2.817 |
| 3 | 0.3 | 1.785 | 3.439 |
| 4 | 0.4 | 2.168 | 4.257 |
| 5 | 0.5 | 2.645 | 5.324 |
| 6 | 0.6 | 3.243 | 6.705 |
| 7 | 0.7 | 3.999 | 8.48 |
| 8 | 0.8 | 4.955 | 10.742 |
| 9 | 0.9 | 6.167 | 13.608 |
| 10 | 1 | 7.702 | 17.216 |

Z – матрица решений методом Рунге–Кутты системы обыкновенных дифференциальных уравнений

Полученное решение представлено в виде графика (рис. 12).

$x := 0, 0.1 .. 1$

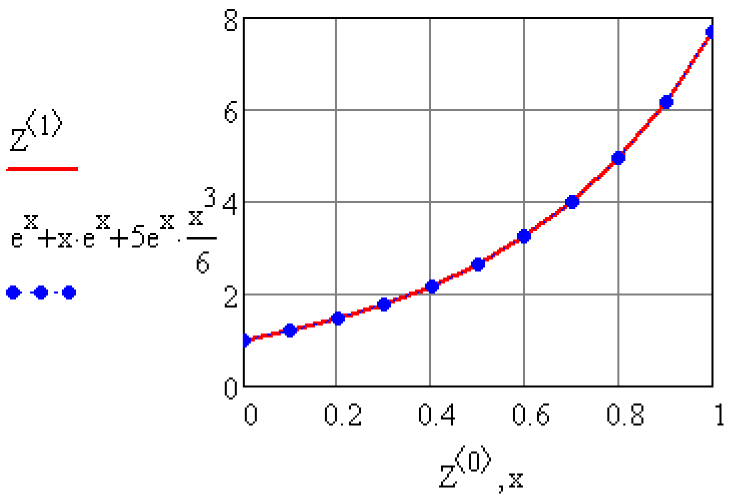



Рис. 12. Графики численного и точного решений дифференциального уравнения

Применение функции odesolve требует записи вычислительного блока, в который входят три части.

– ключевое слово Given,

– дифференциальное уравнение и начальные, или граничные, условия к нему,

– функция odesolve(x,xk,k), где x – имя переменной, относительно которой решается уравнение; xk – конец интервала интегрирования (начало интервала интегрирования указывается в начальных условиях); k – необязательный внутренний параметр, определяющий число шагов интегрирования, на которых вычисляется решение дифференциального уравнения.

Функция `odesolve` возвращает решение дифференциального уравнения в виде функции, а не в виде массива, как `rkfixed`, поэтому найденное решение можно интегрировать и дифференцировать, а также использовать в последующих расчетах как функцию пользователя.

***Пример***. Решить дифференциальное уравнение 3-го порядка $y'''(x)+x^2\cdot y'(x)+$ $+x\cdot y(x)=e^x\cdot\cos(x)$ на отрезке [0;5] с начальными условиями $y_o=-8,\quad y_o^{'}=3,\quad y_o^{''}=3$ с использованием функции `odesolve` (для набора штриха в дифференциальном уравнении служат клавиши Ctrl+F7).

***Решение***.

Given

$$y'''(x)+x^2 y'(x)+x\cdot y(x)=e^x\cos(x)$$

$$y(0)=-8\qquad y'(0)=3\qquad y''(0)=3$$

$$y:=\text{Odesolve}(x,5)$$

Полученное решение представлено в виде графика (рис. 13).

$$x:=0,0.1..5$$

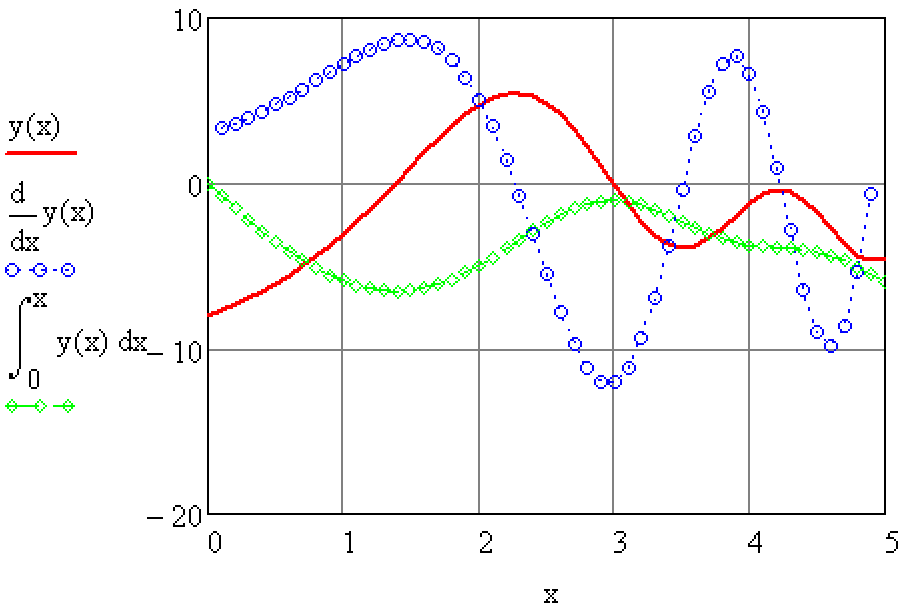



Рис. 13. Графики искомой функции, производной
и первообразной от искомой функции

***Контрольные вопросы по лабораторной работе № 5***

1. Опишите метод Эйлера для решения обыкновенных дифференциальных уравнений.

2. От чего зависит погрешность при решении обыкновенных дифференциальных уравнений методом Эйлера?

3. В чем заключается алгоритм проверки правильности полученного решения обыкновенного дифференциального уравнения методом Эйлера?

4. Опишите последовательность действий для решения обыкновенных дифференциальных уравнений в программе Mathcad.

## Список литературы

1. Щапова И.Н., Щапов В.А. Информатика: учеб. пособие. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2016. – 154 с.

2. Электронные таблицы: метод. указания к выполнению лаб. работ по дисциплине «Информатика» [Электронный ресурс] / сост. И.Н. Щапова. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2015. – 26 с. – URL: http://pstu.ru/title1/faculties/ gnf/?infores=1.

3. Макаров Е. Инженерные расчеты в Mathcad 14. – СПб.: Питер, 2007. – 591 с.

Учебное издание

Щапова Ирина Николаевна,
Щапов Владислав Алексеевич

# ПРОГРАММИРОВАНИЕ. ОБРАБОТКА ИНФОРМАЦИИ В ПАКЕТАХ ПРИКЛАДНЫХ ПРОГРАММ

*Лабораторный практикум*

Редактор и корректор *И.Н. Жеганина*

---

Подписано в печать 20.09.2018. Формат 60×90/8.
Усл. печ. л. 4,5. Тираж 40 экз. Заказ № 191/2018.

---

Издательство
Пермского национального исследовательского
политехнического университета.
Адрес: 614990, г. Пермь, Комсомольский пр., 29, к. 113.
Тел.: + 7 (342) 219-80-33.