

**Министерство образования и науки, молодежи и спорта
Украины
Донбасская государственная машиностроительная академия**

Составитель: А. А. Костиков

**ЧИСЛЕННЫЕ МЕТОДЫ И МОДЕЛИРОВАНИЕ
КОНСПЕКТ ЛЕКЦИЙ**

Для студентов направлений подготовки

6.050202 «Автоматизация и компьютерно-интегрированные технологии»

6.092501 «Автоматизированное управление технологическими
процессами»

Утверждено
на заседании
методического семинара кафедры
Протокол №5 от 19.01.2012

Краматорск 2012

УДК 681.31:001.8

Конспект лекций по дисциплине «Численные методы и моделирование» для студентов направлений подготовки 6.050202 «Автоматизация и компьютерно интегрированные технологии», 6.092501 «Автоматизированное управление технологическими процессами» / сост. А. А. Костиков. – Краматорск, ДГМА, 2012. – 55с.

Рассмотрены примеры решения задач различного типа: построение графиков функций, нахождение производных и интегралов, интерполяция, решение обыкновенных дифференциальных уравнений с применением пакета Mathcad. Приведены алгоритмы численных методов и программы в Mathcad, реализующие эти методы

Составители:

Костиков А.А., доцент.

Отв. за выпуск

Черномаз В.Н., доцент

Введение

При решении инженерных задач инженер сталкивается с необходимостью применять математические знания: решать алгебраические уравнения и системы уравнений, находить экстремумы функций, вычислять производные и интегралы, находить решения дифференциальных уравнений и т.д.

Развитие компьютерной техники позволило инженеру решать сложные вычислительные задачи, возникающие при моделировании технических устройств и процессов. В настоящее время разработаны математические пакеты, позволяющие решать подобные задачи. Одним из таких математических пакетов является Mathcad.

Главными достоинствами пакета Mathcad являются:

- легкость и наглядность программирования задач;
- запись сложных математических выражений в Mathcad совпадает с привычной математической записью;
- простота в использовании;
- возможность создания высококачественных технических отчетов с таблицами, графиками, текстом.

В данном методическом пособии рассмотрены методы решения математических задач с помощью пакета Mathcad. Приведены примеры решения алгебраических и дифференциальных уравнений, численного и аналитического дифференцирования и интегрирования, интерполяции функций. Изложены алгоритмы численных методов и дана их программная реализация в среде Mathcad.. Изложение сопровождается примерами. Решение каждого примера сопровождается подробными комментариями.

1 ЭЛЕМЕНТЫ ТЕОРИИ ПОГРЕШНОСТЕЙ

1.1 Источники погрешностей

Есть 4 источника погрешностей численного метода:

- математическая модель
- исходные данные
- приближенный метод
- округления при вычислениях

Измерение любой физической величины (расстояния, напряжения, интервала времени и др.) не может быть выполнено точно. Погрешность вычислений в физике и технике достигает 1-10%. Если, например, величина напряжения равна 220,6483759 В, то можно сказать, что несколько младших цифр этого значения недостоверны, что обусловлено точностью измерительного прибора. Если рациональное число $\frac{1}{3}$ представить в виде конечной десятичной дроби, то такое представление не будет точным, так как это число представляется в виде бесконечной периодической десятичной дроби.

1.2 Абсолютная и относительная погрешности

Определение 1

Разность между точным числом a и его приближенным значением a^* называется погрешностью данного приближенного числа. Если известно, что $|a - a^*| \leq \Delta(a^*)$, то величина $\Delta(a^*)$ называется предельной абсолютной погрешностью приближенной величины a^* ; отношение $\delta(a^*) = \left| \frac{\Delta(a^*)}{a^*} \right|$ называется предельной относительной погрешностью (часто выражается в %).

Пример 1.1

Число 3,14 – приближенное значение числа $\pi = 3,14159$. Его погрешность $\Delta(a^*) = |3,14159 - 3,14| = 0,00159$. Предельную абсолютную погрешность можно считать равной 0,0016, а предельную относительную погрешность равной $\frac{0,0016}{3,14} = 0,00051 = 0,051(\%)$.

(Для краткости слово «предельная» опускается).

Определение 2.

Значащими цифрами числа называют все цифры в его записи, начиная с первой ненулевой слева.

Пример 1.2

1 куб. фут=0,0283 м³, 1 дюйм=2,5400 см. В первом случае имеется три значащие цифры, во втором – пять.

Значащую цифру называют верной, если модуль погрешности числа не превосходит единицы разряда, соответствующей этой цифре. Если число a^* имеет n верных значащих цифр, то его относительная погрешность

$$\delta(a^*) \leq \frac{1}{z \cdot 10^{n-1}}, \text{ где } z - \text{ первая значащая цифра числа } a^* .$$

У числа a^* с относительной погрешностью $\delta(a^*)$ верны n значащих, где n – наибольшее целое число, удовлетворяющее неравенству

$$(1 + z)\delta(a^*) \leq 10^{1-n} .$$

Пример 1.3

Если число $a^* = 47,542$ получено в результате действий над приближенными числами и известно, что $\delta(a^*) = 0,1\%$, то a^* имеет три верных значащих цифры, так как $(1 + 4) \cdot 0,001 < 10^{-2}$.

Числа $a^*, \Delta(a^*)$ принято записывать с одинаковым количеством знаков после запятой:

$$a = 1,123 \pm 0,004;$$

$$a = 1,123 \pm 4 \cdot 10^{-3};$$

$$1,123 - 0,004 \leq a \leq 1,123 + 0,004 .$$

1.3 Погрешность округленного числа.

Если приближенное число содержит лишние(или неверные знаки), то его следует округлить. При округлении сохраняются только верные знаки; лишние знаки отбрасываются, причем:

- 1 если первая отбрасываемая цифра больше четырех, то последняя сохраняемая цифра увеличивается на единицу
- 2 если отбрасываемая часть состоит только из цифры 5 , то округляют так, чтобы последняя цифра оставалась четной.

Пример 1.4.

Округляя число $x = 1,1426$ до четырех значащих цифр, определить абсолютную и относительную погрешности полученных приближений.

Решение:

Округлим число x до четырех значащих цифр $x_1 = 1,143$. По определению верной цифры абсолютная погрешность числа x равна $e_x = 0,0001$. Погрешность округления $\Delta_{окр} = |1,143 - 1,1426| = 0,0004$.

Следовательно, погрешность округленного числа x_1 равна $e_{x_1} = 0,0004 + 0,0001 = 0,0005$. Поэтому абсолютная погрешность равна

$$\delta_{x_1} = \left| e_{x_1} / x_1 \right| = 0,0005 / 1,143 = 0,000437 = 0,04\% .$$

1.4 Вычислительная погрешность

Будем обозначать абсолютную погрешность числа x как e_x , относительную погрешность δ_x

Приведем формулы для вычисления погрешностей, возникающих при выполнении арифметических операций над числами x и y

1 Погрешность суммы

Абсолютная погрешность:

$$z = (x \pm e_x) + (y \pm e_y) = (x + y) \pm (e_x + e_y)$$

Относительная погрешность:

$$\delta_z = \frac{e_x + e_y}{x + y} = \frac{|x|}{|x + y|} \cdot \delta_x + \frac{|y|}{|x + y|} \cdot \delta_y$$

Аналогично определяется погрешность разности.

2 Погрешность произведения

Абсолютная погрешность

$$z = (x \pm e_x) \cdot (y \pm e_y) = x \cdot y \pm y \cdot e_x \pm x \cdot e_y$$

Относительная погрешность

$$\delta_z = \frac{|y| \cdot e_x + |x| \cdot e_y}{|x \cdot y|} = \delta_x + \delta_y$$

3 Погрешность частного

Абсолютная погрешность

$$z = \frac{x \pm e_x}{y \pm e_y} \approx \frac{x}{y} + \frac{y \cdot e_x + x \cdot e_y}{y^2}$$

Относительная погрешность:

$$\delta_z = \delta_x + \delta_y$$

4 Погрешность функции, зависящей от одной переменной

Абсолютная погрешность:

$$f(x \pm e_x) \approx f(x) + f'(x) \cdot e_x$$

$$\Delta f = |f'(x)| \cdot e_x$$

Относительная погрешность:

$$\frac{|\Delta f|}{|f|} = \frac{|f'(x)|}{|f(x)|} \cdot e_x$$

2 РЕШЕНИЕ УРАВНЕНИЙ С ОДНОЙ ПЕРЕМЕННОЙ.

2.1 Основные определения.

Нелинейное уравнение имеет вид

$$f(x) = 0,$$

где $f(x)$ – функция, определенная на отрезке $[a, b]$. Здесь a и b – действительные числа.

Определение 1

Всякое число $\xi \in [a, b]$, при котором $f(x) = 0$, называется корнем уравнения $f(x) = 0$.

Определение 2

Если $f(x)$ – многочлен, отличный от нулевого, то уравнение называется алгебраическим, иначе – трансцендентным (показательным, логарифмическим, тригонометрическим и т.п.).

Задача приближенного вычисления корней уравнения $f(x) = 0$ распадается на две:

- 1 Отделение корней уравнения – процедура нахождения отрезков, на которых уравнение $f(x) = 0$ имеет только одно решение.
- 2 Вычисление корня с заданной точностью ε .

Справедлива следующая теорема:

Теорема Больцано-Коши:

Если непрерывная функция $f(x)$ принимает на концах отрезка $[a, b]$ значения разных знаков, т.е. $f(a) \cdot f(b) < 0$, то внутри этого отрезка содержится, по крайней мере, один корень. Этот корень будет единственным, если производная $f'(x)$ существует и сохраняет постоянный знак внутри отрезка $[a, b]$.

Рассмотрим приближенные методы нахождения корней.

2.2 Метод половинного деления (дихотомии).

Пусть уравнение $f(x) = 0$ имеет на отрезке $[a, b]$ единственный корень, и $f(x)$ на данном отрезке непрерывна.

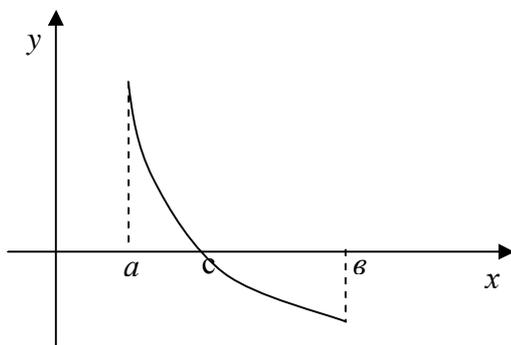


Рисунок 2.1 – Метод половинного деления

Разделим отрезок $[a, b]$ пополам точкой $c = \frac{a+b}{2}$ (рис.2.1).

Если $f(c) \neq 0$, то возможны 2 случая:

- 1) корень лежит на отрезке $[a, c]$
- 2) корень лежит на отрезке $[c, b]$.

Если $f(a) \cdot f(c) < 0$, то корень лежит на отрезке $[a, c]$, а если $f(c) \cdot f(b) < 0$, то корень лежит на отрезке $[c, b]$. Выбираем в каждом случае тот отрезок, на котором функция меняет знак и продолжаем процесс половинного деления. В результате получим сколь угодно малый отрезок, содержащий корень уравнения. Блок схема алгоритма половинного деления приведена на рис. 2.2.

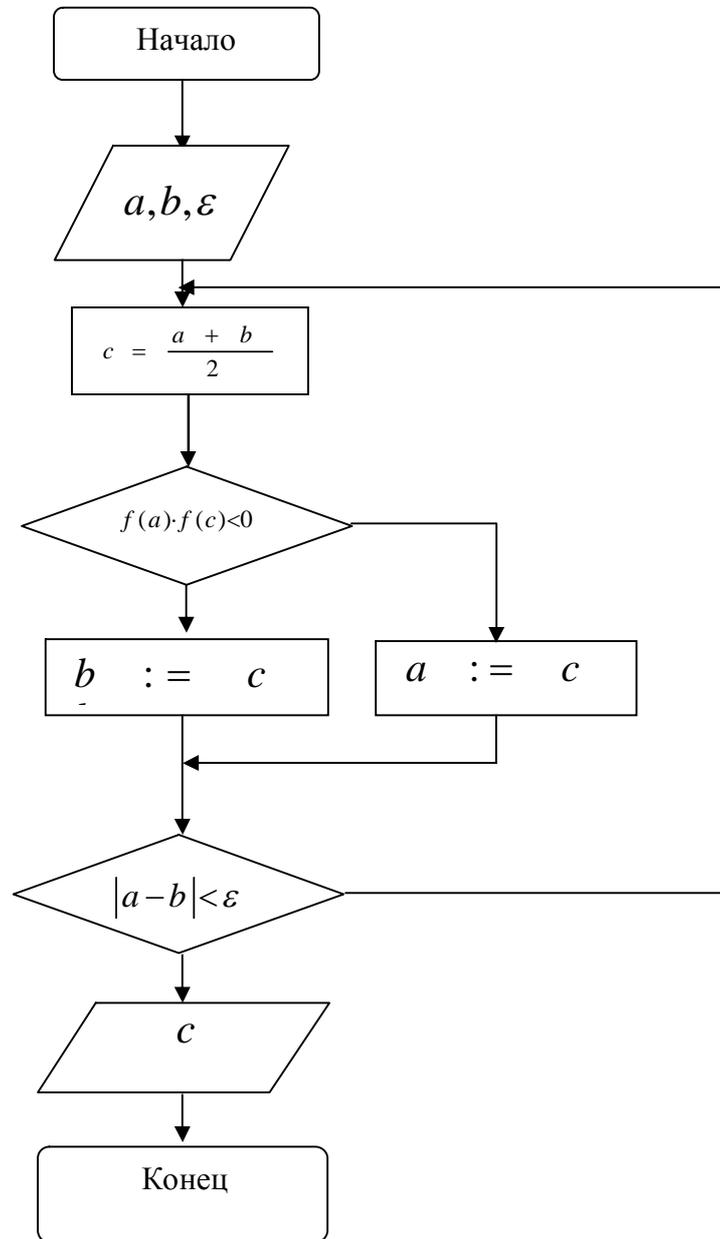


Рис. 2.2 Блок схема алгоритма метода половинного деления

Рассмотрим пример нахождения корня методом половинного деления в Mathcad.

Пример 2.1

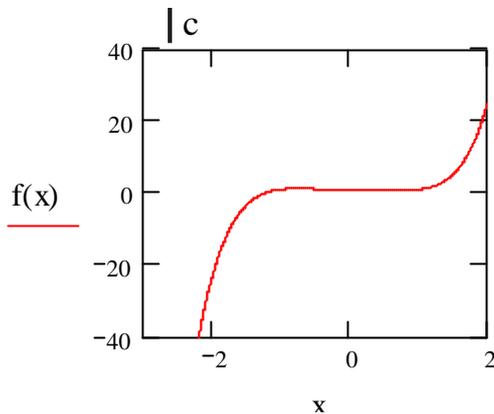
Найти решение уравнения $x^5 - x^3 + 1 = 0$ методом половинного деления.

Решение:

- 1 Определяем функцию $f(x)$, равную левой части уравнения

$$f(x) := x^5 - x^3 + 1$$

- 2 Определяем интервал, на котором находится корень уравнения. Для этого строим график функции $f(x)$



- 3 Задаем функцию $\mathbf{mpd}(a, b, \varepsilon)$, определяющую корень уравнения методом половинного деления. Эта функция имеет следующие аргументы: $\mathbf{a, b}$ -границы интервала, на котором находится корень уравнения, ε -точность вычисления корня.

$$\mathbf{mpd}(a, b, \varepsilon) := \left| \begin{array}{l} L \leftarrow b - a \\ \mathbf{while} \ L > \varepsilon \\ \quad \left| \begin{array}{l} c \leftarrow \frac{a + b}{2} \\ b \leftarrow c \ \text{if} \ f(a) \cdot f(c) < 0 \\ a \leftarrow c \ \text{otherwise} \\ L \leftarrow b - a \end{array} \right. \\ \mathbf{c} \end{array} \right.$$

- 4 Вычисление значения корня

$$\mathbf{mpd}(-2, 2, 0.01) = -1.242 \blacksquare$$

Таким образом, корень уравнения равен -1.242 . Из графика видно, что корень уравнения находится на отрезке $[-2; 2]$, поэтому границы этого отрезка взяты в качестве параметров \mathbf{a} и \mathbf{b} функции \mathbf{mpd} . Корень найден с точностью $\varepsilon=0,01$.

2.3 Метод хорд

Вместо деления отрезка $[a, b]$ пополам, как в методе половинного деления, делим его в отношении $f(a) : f(b)$. Таким образом первое приближение корня находится в точке пересечения отрезка $[a, b]$ хордой, проходящей через точки с координатами $(a, f(a))$ и $(b, f(b))$. Если функция меняет знак на отрезке $[a, c]$, т.е. $f(a) \cdot f(c) < 0$, то корень лежит на отрез-

ке $[a, c]$, в противном случае корень лежит на отрезке $[c, b]$. Выбирая в качестве нового отрезка отрезок $[a, c]$ или $[c, b]$ продолжаем процесс до тех пор, пока длина отрезка, на котором находится корень, не станет меньше любого наперед заданного числа ε . Точку c находим следующим образом. Запишем уравнение прямой, проходящей через точки с координатами $(a, f(a))$ и $(b, f(b))$:

$$y = \frac{f(b) - f(a)}{b - a} \cdot (x - a) + f(a)$$

Точка пересечения этой прямой с осью Ox и будет искомой точкой c .
Итак,

$$c = a - \frac{f(a)}{f(b) - f(a)} \cdot (b - a)$$

Блок-схема алгоритма метода хорд приведена на рис.2.1

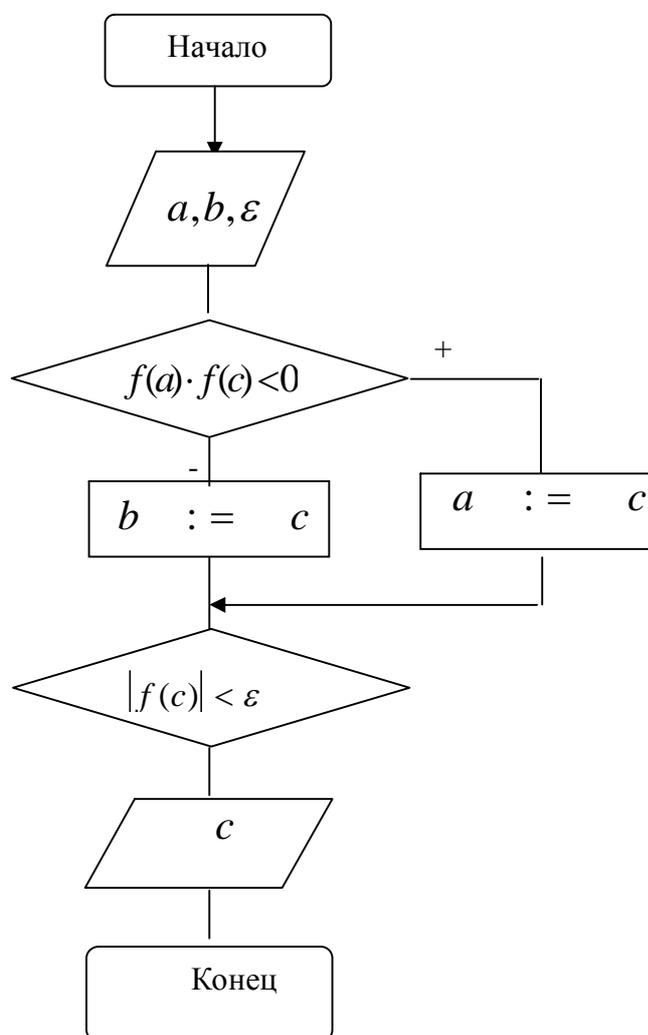


Рис 2.3 Блок-схема метода хорд

Таким образом, алгоритм метода хорд заключается в следующем:

1 Находим точку c по формуле

$$c = a - \frac{f(a)}{f(b) - f(a)} \cdot (b - a)$$

2 Если условие пункта 2 не выполняется, то вычисляем произведение $f(a) \cdot f(c)$. Если $f(a) \cdot f(c) < 0$, то полагаем $b = c$, в противном случае полагаем $a = c$.

3 Если $|f(c)| < \varepsilon$, то завершаем алгоритм. Искомый корень равен c .
Корень найден с точностью ε . В противном случае переходим к пункту 1.

Рассмотрим пример нахождения корня методом хорд.

Пример 2.2

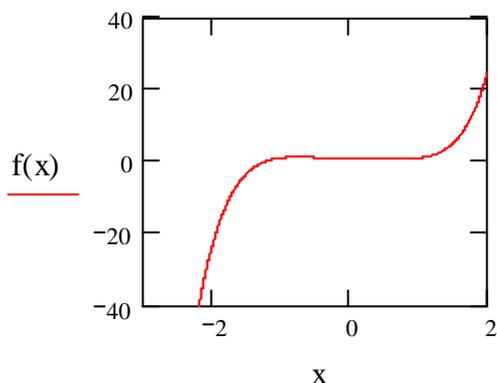
Найти решение уравнения $x^5 - x^3 + 1 = 0$ методом

Решение:

1 Определяем функцию $f(x)$, равную левой части уравнения

$$f(x) := x^5 - x^3 + 1$$

2 Определяем интервал, на котором находится корень уравнения.
Для этого строим график функции $f(x)$



3 Задаем функцию **horda(a,b,ε)**, определяющую корень уравнения методом хорд.

$$\text{horda}(a, b, \varepsilon) := \left| \begin{array}{l} k \leftarrow 0 \\ \text{while } 1 \\ \quad \left| \begin{array}{l} c \leftarrow a - \frac{f(a)}{f(b) - f(a)} \cdot (b - a) \\ b \leftarrow c \text{ if } f(c) \cdot f(a) < 0 \\ a \leftarrow c \text{ otherwise} \\ k \leftarrow k + 1 \\ \text{break if } |f(c)| < \varepsilon \end{array} \right. \\ \left(\begin{array}{c} c \\ k \end{array} \right) \end{array} \right.$$

Эта функция имеет следующие аргументы: **a, b**-границы интервала, на котором находится корень уравнения, **ε**-точность вычисления корня. Функция **horda** возвращает значение корня и количество итераций **k**, необходимое для достижения заданной точности.

4. Находим корень уравнения с точностью $\varepsilon = 0,001$, используя метод хорд

$$\text{horda}(-2, -1, 0.001) = \left(\begin{array}{c} -1.236 \\ 30 \end{array} \right) \blacksquare$$

Таким образом, искомое значение корня равно -1.236 . Для достижения заданной точности потребовалось 30 итераций.

2.4 Метод касательных(метод Ньютона)

Пусть уравнение $f(x) = 0$ имеет корень на отрезке $[a, b]$, причем $f'(x)$ и $f''(x)$ непрерывны и сохраняют определенный знак при $a \leq x \leq b$. Пусть $x_0 = b$, причем $f(x_0) \cdot f''(x_0) = b$. Проведем касательную к графику функции в точке с координатами $(b, f(b))$. Ее уравнение будет иметь вид:

$$y = f(b) + f'(b) \cdot (x - b) \quad (2.1)$$

Полагая в уравнении (2.1) $y = 0$ и учитывая, что $f'(b) \neq 0$, решим его относительно x . В результате получим

$$x = b - \frac{f(b)}{f'(b)}$$

Это значение выбираем в качестве первого приближения корня x_1 . Проведем касательную к графику функции в точке с координатами $(x_1, f(x_1))$. Найдем абсциссу x_2 точки пересечения касательной с осью Ox :

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Это значение выберем в качестве второго приближения корня. Аналогично определяем $k+1$ -е приближение:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (2.2)$$

Итерации (2.2) проводим до тех пор, пока не будет достигнута заданная точность. Геометрическая иллюстрация метода касательных приведена на рис. 2.4.

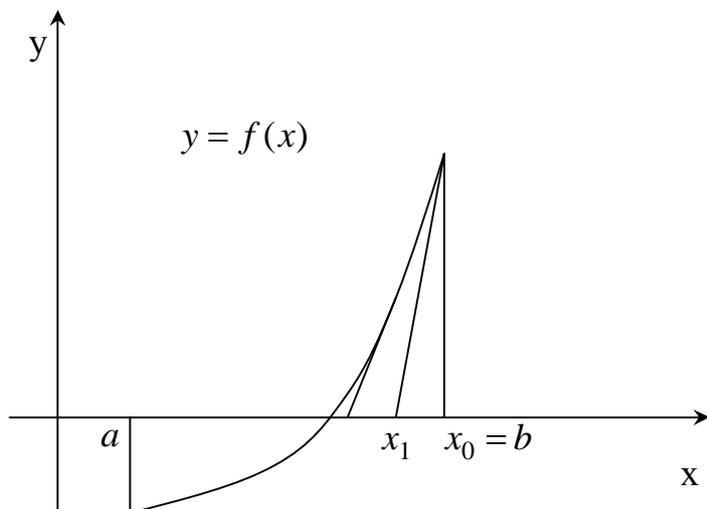


Рис 2.4 Геометрическая иллюстрация метода касательных

Начальное приближение x_0 в методе касательных выбирается следующим образом: если $f(a) \cdot f'(a) > 0$, то $x_0 = a$, в противном случае $x_0 = b$.

Пример 2.3

Найти корень уравнения

$$\frac{e^x}{5} - 2 \cdot (x-1)^2 = 0$$

методом касательных.

Решение:

1 Определяем функцию $f(x)$, равную левой части уравнения $f(x) = 0$ и ее первую и вторую производную.

$$f(x) := \frac{\exp(x)}{5} - 2 \cdot (x-1)^2 \quad f1(x) := \frac{d}{dx}f(x) \quad f2(x) := \frac{d^2}{dx^2}f(x)$$

2 Определяем интервал, на котором находится корень уравнения.

Для этого строим график функции $f(x) = \frac{e^x}{5} - 2 \cdot (x-1)^2 = 0$ (рис.2.1)

3 Задаем функцию **Newton(a,b,ε)** (рис.2.4), определяющую корень уравнения методом касательных. Эта функция имеет следующие аргументы: **a,b**-границы интервала, на котором находится корень уравнения, **ε**-точность вычисления корня. Функция **Newton** возвращает значение корня и количество итераций **k**, необходимое для достижения заданной точности.

4 Находим корень уравнения с точностью $\varepsilon = 0,001$, используя метод касательных

$$\text{Newton}(-1, 1, 0.001) = \begin{pmatrix} 0.578 \\ 4 \end{pmatrix}$$

Таким образом, искомое значение корня равно 0.578. Для достижения заданной точности потребовалось 4 итерации.

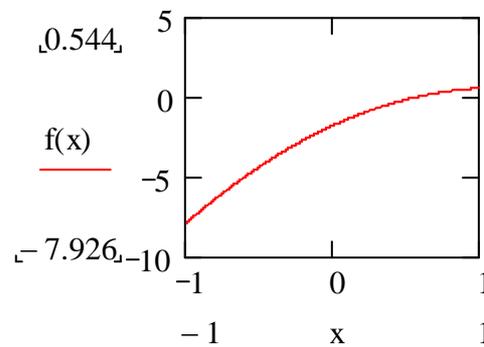


Рис 2.5 График функции $f(x) = \frac{e^x}{5} - 2 \cdot (x-1)^2 = 0$

```

Newton(a, b, ε) :=
  k ← 0
  c ← a if f(a)·f2(a) > 0
  c ← b otherwise
  while 1
    c ← c - f(c)/f1(c)
    k ← k + 1
    break if |f(c)| < ε
  ( c
    k )

```

Рис 2.6. Программа, реализующая метод касательных.

2.4 Встроенные функции Mathcad для нахождения корней

Для решения одного уравнения с одним неизвестным используется функция **root**. Аргументами этой функции является выражение и переменная, входящая в выражение. Ищется значение переменной, при котором выражение обращается в ноль. Функция возвращает значение переменной, которое обращает выражение в ноль.

Для определения корней уравнения необходимо привести уравнение к виду $F(x)=0$ и использовать функцию поиска корней следующим образом: $\text{root}(F(x),x)$, где $F(x)$ – заданное уравнение, x – переменная, относительно которой это уравнение решается. Для поиска корней Mathcad использует приближенные методы вычислений, поэтому перед использованием функции **root** необходимо задать начальное приближение для переменной, относительно которой решается уравнение. Перед решением уравнения желательно построить график функции, чтобы определить, есть ли корни. Начальное приближение лучше всего выбирать по графику поближе к значению корня.

Пример 2.6.

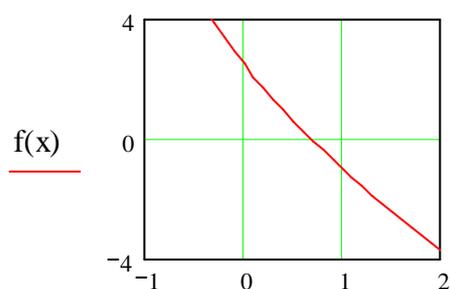
Найти решение уравнения $\arccos^2 x = x$

Решение:

Строим график функции $f(x) = \arccos^2 x - x$

$$f(x) := \arccos(x)^2 - x$$

$$x := -2, -1.9..2$$



$$AX = B,$$

где A –прямоугольная матрица размером $m \times n$:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

X – вектор порядка n :

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

B – вектор порядка m :

$$B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Определение 1:

Решением системы линейных уравнений называется такая упорядоченная совокупность чисел $x_1 = c_1, x_2 = c_2 \dots x_n = c_n$, которая обращает все уравнения системы в верные равенства.

Определение 2:

Прямыми методами решения систем линейных уравнений называются методы, дающие решение системы за конечное число арифметических операций. Если отсутствуют ошибки округления, то полученные решения являются точными. Рассмотрим прямые методы решения системы линейных алгебраических уравнений.

Правило Крамера.

Правило Крамера используют для нахождения решения системы линейных алгебраических уравнений. Если определитель $\Delta = \det A$ матрицы системы из n уравнений с n неизвестными $AX = B$ отличен от нуля, то система имеет единственное решение x_1, x_2, \dots, x_n , определяемое по формулам Крамера $x_i = \Delta_i / \Delta$, где Δ_i – определитель матрицы, получаемой из матрицы системы A заменой i -го столбца столбцом свободных членов B .

Пример 3.1:

Решить систему линейных уравнений при помощи правила Крамера.

$$\begin{cases} 2x_1 + x_2 = 8 \\ x_1 - 3x_2 = 9 \end{cases}$$

Ниже приведен фрагмент документа Mathcad, решающий поставленную задачу.

Определение матрицы A и вектора B

$$A := \begin{pmatrix} 2 & 1 \\ 1 & -3 \end{pmatrix} \quad B := \begin{pmatrix} 8 \\ 9 \end{pmatrix}$$

Вычисление определителя системы

$$\Delta := |A| \quad \Delta = -7$$

Определение вспомогательных матриц

$$\Delta_1 := |A_1| \quad \Delta_2 := |A_2|$$

$$\Delta_1 = -33 \quad \Delta_2 = 10$$

Нахождение решения системы линейных уравнений методом Крамера

$$x_1 := \frac{\Delta_1}{\Delta} \quad x_2 := \frac{\Delta_2}{\Delta}$$

$$x_1 = 4.714 \quad x_2 = -1.429$$

3.2 Метод обратной матрицы.

Для решения системы линейных уравнений методом обратной матрицы запишем ее в матричном виде

$$A \cdot X = B \tag{3.1}$$

Умножая (3.1) на A^{-1} , получим

$$A^{-1}AX = A^{-1}B,$$

откуда следует

$$X = A^{-1}B$$

Пример 3.2

Решить систему из предыдущего примера методом обратной матрицы

Решение:

1 Определяем матрицу коэффициентов и вектор правых частей системы

$$A := \begin{pmatrix} 2 & 1 \\ 1 & -3 \end{pmatrix} \quad B := \begin{pmatrix} 8 \\ 9 \end{pmatrix}$$

2 Находим решение системы методом обратной матрицы

$$X := A^{-1} \cdot B \quad X = \begin{pmatrix} 4.714 \\ -1.429 \end{pmatrix}$$

3 Делаем проверку

$$A \cdot X - B = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \blacksquare$$

Эту же задачу можно решить с помощью встроенной функции Mathcad **lsolve(A,B)**.

Аргументы функции: **A** –матрица системы , **B**-вектор правых частей.

Ниже приведено обращение к функции lsolve в Mathcad.

$$X := \text{lsolve}(A, B)$$

$$X = \begin{pmatrix} 4.714 \\ -1.429 \end{pmatrix}$$

3.3 Метод прогонки

На практике часто встречаются системы линейных уравнений вида

$$\begin{pmatrix} c_1 & d_1 & 0 & 0 & \dots & 0 & 0 \\ b_2 & c_2 & d_2 & 0 & \dots & 0 & 0 \\ 0 & b_3 & c_3 & d_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & b_{n-1} & c_{n-1} & d_{n-1} \\ \dots & \dots & \dots & \dots & b_n & c_n \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \dots \\ r_{n-1} \\ r_n \end{pmatrix}, \quad (3.2)$$

которые называются системами с трехдиагональной матрицей коэффициентов. Каждое из уравнений такой системы содержит 3 неизвестных и может быть записано в виде:

$$b_i \cdot x_{i-1} + c_i \cdot x_i + d_i \cdot x_{i+1} = r_i, \quad (3.3)$$

где $i=1,2,\dots,n$, $b_1 = 0$, $d_n = 0$.

Для решения системы уравнений (3.2) наиболее экономичным по объему затрачиваемой работы является метод прогонки, который учитывает трех-диагональность матрицы системы (3.2).

Будем искать решение системы (3.2) в виде

$$x_i = \delta_i \cdot x_{i+1} + \lambda_i, i = 1, 2, \dots, n, \quad (3.4)$$

где δ_i и λ_i - неизвестные пока функции. Подставляя $x_{i-1} = \delta_{i-1} \cdot x_i + \lambda_{i-1}$ в (3.3), исключим x_{i-1} и получим

$$b_i \cdot \delta_{i-1} \cdot x_i + b_i \cdot \lambda_{i-1} + c_i \cdot x_i + d_i \cdot x_{i+1} = r_i \quad (3.5)$$

Запишем (3.6) в виде

$$x_i = -\frac{d_i}{c_i + b_i \cdot \delta_{i-1}} \cdot x_{i+1} + \frac{r_i - b_i \cdot \lambda_{i-1}}{c_i + b_i \cdot \delta_{i-1}} \quad (3.6)$$

Из (3.6) следует, что соотношение (1.30) имеет место, если для всех $i = 1, 2, \dots, n$ выполняются рекуррентные соотношения:

$$\delta_i = \frac{d_i}{c_i + b_i \cdot \delta_{i-1}}, \lambda_i = \frac{r_i - b_i \cdot \lambda_{i-1}}{c_i + b_i \cdot \delta_{i-1}} \quad (3.7)$$

Т.к. $b_1 = 0$, то процесс вычисления δ_i и λ_i по формулам (1.6) можно начать со значений

$$\delta_1 = -\frac{d_1}{c_1}, \lambda_1 = \frac{r_1}{c_1} \quad (3.8)$$

и продолжать для $i = 2, 3, \dots, n$. В случае, когда $i = n$ и $d_n = 0$ получим, что

$$x_n = \lambda_n = \frac{r_n - b_n \cdot \lambda_{n-1}}{c_n + b_n \cdot \delta_{n-1}}, \delta_n = 0, \quad (3.9)$$

где δ_{n-1} и λ_{n-1} были вычислены на предыдущем шаге. Далее по формулам (1.5) находят значения $x_{n-1}, x_{n-2}, \dots, x_1$. Таким образом, метод прогонки состоит из 2-х шагов:

1. Находят прогоночные коэффициенты δ_i и λ_i по формулам (3.7). Этот процесс называется прямой прогонкой.
2. Определяют неизвестные $x_{n-1}, x_{n-2}, \dots, x_1$ по формулам (3.6). Этот процесс называется обратной прогонкой.

На рис. 3.1 приведена блок-схема метода прогонки.

Рассмотрим пример решения системы линейных уравнений методом прогонки в среде Mathcad.

Пример 3.3

Найти решение системы линейных уравнений методом прогонки

$$\begin{cases} 2x_1 + x_2 = -10 \\ 2x_1 + 9x_2 + 2x_3 = -26 \\ 4x_2 + 17x_3 - 4x_4 = -16 \\ 4x_3 + 15x_4 - 8x_5 = -2 \\ 2x_4 + 3x_5 = 16 \end{cases}$$

Решение:

1 Определяем векторы b, c и d , содержащие элементы трехдиагональной матрицы системы

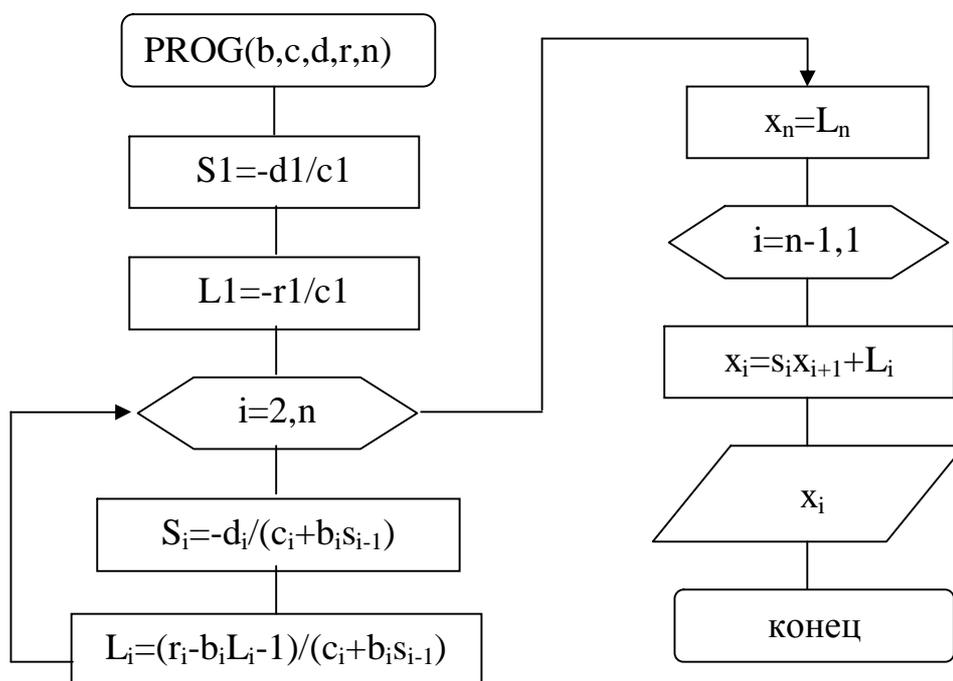


Рис 3.1 Блок-схема метода прогонки

$$b := \begin{pmatrix} 0 \\ 2 \\ 4 \\ 4 \\ 2 \end{pmatrix} \quad c := \begin{pmatrix} 2 \\ 9 \\ 17 \\ 15 \\ 3 \end{pmatrix} \quad d := \begin{pmatrix} 1 \\ 2 \\ -4 \\ -8 \\ 0 \end{pmatrix} \quad r := \begin{pmatrix} -10 \\ -26 \\ -16 \\ -2 \\ 16 \end{pmatrix}$$

2 Задаем функцию **PROGON(b,c,d,r,n)**, реализующую метод прогонки. Аргументы функции **PROGON**: **b,c,d** – векторы, содержащие элементы матрицы системы, **r**-вектор правой части, **n** – число неизвестных.

где F_1, F_2, \dots, F_n - заданные нелинейные вещественные функции n вещественных переменных x_1, x_2, \dots, x_n .

Будем считать совокупность аргументов системы n - мерным вектором

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix},$$

а совокупность функций – n -мерным вектором – функций

$$F = \begin{pmatrix} F_1 \\ F_2 \\ \dots \\ F_n \end{pmatrix}$$

Тогда систему (5.1) можно представить в виде

$$F(x) = 0 \tag{4.2}$$

4.2 Метод Ньютона решения системы нелинейных уравнений

Предположим, известное k -е приближение для одного из корней x^k . Тогда точный корень (4.2) можно представить в виде :

$$x = x^k + \varepsilon^k \tag{4.3}$$

ε^k - погрешность корня

Подставим (4.3) в (4.2). Тогда получим

$$F(x^k + \varepsilon^k) = 0 \tag{4.4}$$

Если $F(x)$ непрерывно дифференцируема в некоторой области, которой принадлежат x и x^k , то

$$F(x^k + \varepsilon) = F(x^k) + F'(x^k) \cdot \varepsilon^k, \tag{4.5}$$

где

$$F'(x) = W(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \dots & \frac{\partial F_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \dots & \frac{\partial F_n}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial F_1}{\partial x_j} \end{pmatrix} - \text{матрица Якоби,}$$

$i, j, = 1, 2, \dots, n.$

Соотношение (4.5) может быть записано в виде

$$F(x^k) + W(x^k) \cdot \varepsilon^k = 0 \quad (4.6)$$

Соотношение (5.6) представляет собой систему линейных уравнений относительно неизвестных ε^k , матрица которой равна $W(x^k)$.

Если $\det W(x^k) \neq 0$, тогда

$$\varepsilon^k = -W^{-1}(x^k) \cdot F(x^k) \quad (4.7)$$

Отсюда следует основная формула Ньютона:

$$x^{k+1} = x^k + W^{-1}(x^k) \cdot F(x^k), \quad k = 1, 2, \dots \quad (4.8)$$

где за x^0 можно взять грубое значение искомого корня.

Практически прекращают вычисления по (7), когда

$$|x^{k+1} - x^k| < \varepsilon$$

Алгоритм метода Ньютона:

1. Определяем начальное приближение x^0
2. Уточняем значение корня по формуле $x^{k+1} = x^k - W^{-1}(x^k) \cdot F(x^k)$
3. Если условие $|x^{k+1} - x^k| < \varepsilon$ выполняется, то задача решена и x^{k+1} - корни векторного уравнения $F(x)=0$, иначе переходим к п.2

Блок схема алгоритма приведена на рис. 4.1.

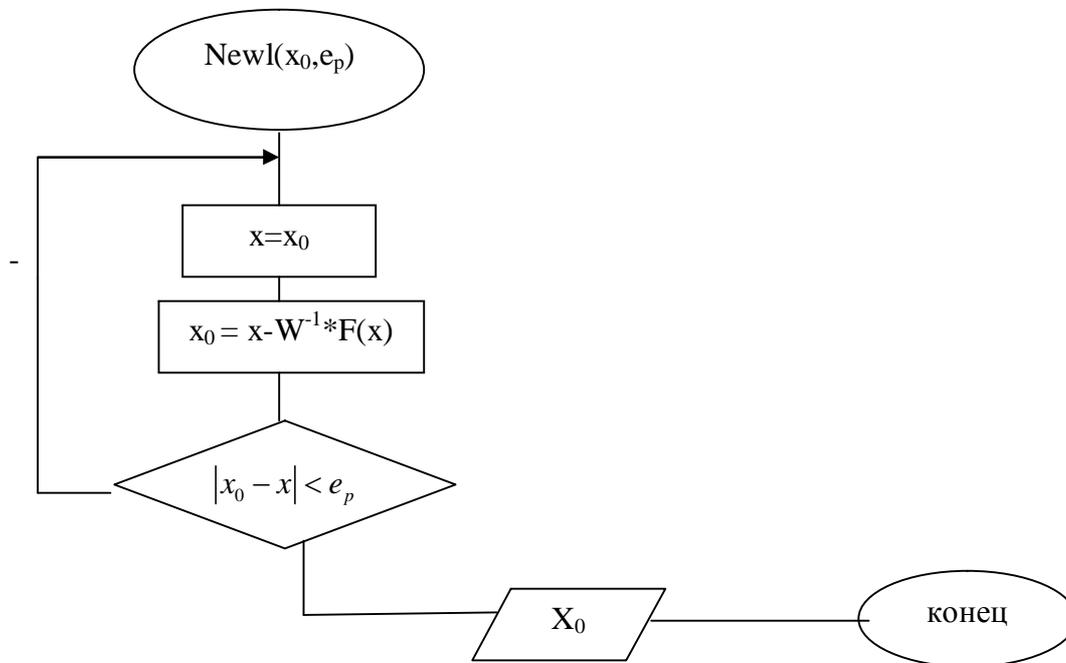


Рис.4.1 Блок схема метода Ньютона.

Пример 4.1

Решить систему нелинейных уравнений методом Ньютона

$$\begin{aligned} x_1 + \lg(x_1) - x_2^2 &= 0 \\ 2x_1^2 - x_1 \cdot x_2 - 5x_1 + 1 &= 0 \end{aligned} \quad (4.9)$$

Решение:

1. Задаем функции, стоящие в левых частях системы

$$\begin{aligned} \text{ORIGIN} &:= 1 \\ F1(x_1, x_2) &:= x_1 + \log(x_1) - x_2^2 \\ F2(x_1, x_2) &:= 2 \cdot x_1^2 - x_1 \cdot x_2 - 5 \cdot x_1 + 1 \end{aligned}$$

2. Определяем элементы матрицы Якоби

$$\begin{aligned} W11(x_1, x_2) &:= \frac{d}{dx_1} F1(x_1, x_2) & W12(x_1, x_2) &:= \frac{d}{dx_2} F1(x_1, x_2) \\ W21(x_1, x_2) &:= \frac{d}{dx_1} F2(x_1, x_2) & W22(x_1, x_2) &:= \frac{d}{dx_2} F2(x_1, x_2) \end{aligned}$$

3. Задаем функцию newt(x0, ε), реализующую метод Ньютона. Аргументы этой функции – вектор начальных приближений и точность вычислений

```

newt(x0,ε) := | while 1
                | x ← x0
                | W ← ( W11(x1,x2)  W12(x1,x2) )
                    ( W21(x1,x2)  W21(x1,x2) )
                | b ← ( F1(x1,x2) )
                    ( F2(x1,x2) )
                | x0 ← x - lsolve(W,b)
                | break if min(|x0 - x|) < ε
                | x0

```

4. Находим решение системы (4.9) с помощью функции newt.

$$X := \text{newt}\left[\begin{pmatrix} 3 \\ 4 \end{pmatrix}, 0.0001\right] \quad X = \begin{pmatrix} 3.332 \\ 1.963 \end{pmatrix}$$

5. Делаем проверку

$$F1(X_1, X_2) = -4.406 \times 10^{-10} \quad F2(X_1, X_2) = 1.831 \times 10^{-4}$$

Решим теперь эту систему графически. Для этого в уравнениях системы (5.9) выразим x_2 через x_1 и на одном чертеже построим графики

функций $x_2 = \frac{2x_1^2 - 5x_1 + 1}{x_1}$ и $x_2 = \pm\sqrt{x_1 + 3\lg(x_1)}$. Координаты точек пересечения графиков функций и будут искомыми решениями(см.рис.4.2)

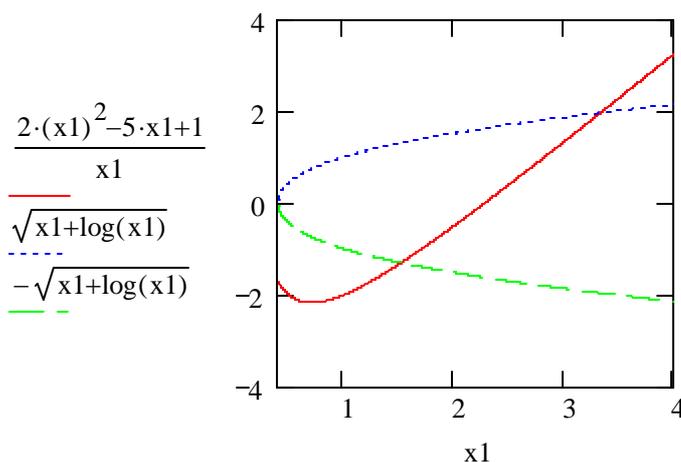


Рис.4.2 Графическое решение системы уравнений в Mathcad.

4.3 Решение системы нелинейных уравнений с помощью встроенных функций Mathcad

Для решения систем нелинейных уравнений следует использовать вычислительный блок Given. Необходимо выполнить следующие действия

1. Задать начальное приближение.
2. Ввести ключевое слово Given
3. Записать систему уравнений. Знак равенства набирается комбинацией клавиш Ctrl+=.
4. Задать ограничения на поиск решения, если они есть.
5. Ввести функцию Find или Minerr с неизвестными в качестве параметров.

Функция Find (x_1, x_2, \dots, x_n) возвращает вектор решения системы.

Функция Minerr (x_1, x_2, \dots, x_n) возвращает значения аргументов, при которых невязка будет минимальной (для случая двух уравнений минимальным будет расхождение между двумя кривыми). Если точное решение существует, то функция Minerr позволяет найти его также как и функция Find. Если же точного решения нет, то функция Find указывает на ошибку, а функция Minerr возвращает минимум невязки.

Пример 4.2:

Найти решение нелинейной системы уравнений.

$$\begin{cases} x + 3 \cdot \log(x) - y^2 = 0 \\ 2x^2 - xy - 5x + 1 = 0 \end{cases}$$

Решение :

$$x := 3.4 \quad y := 2.2$$

Given

$$x + 3 \cdot \log(x) - y^2 = 0$$

$$2x^2 - xy - 5x + 1 = 0$$

$$\text{Minerr}(x, y) = \begin{pmatrix} 3.487 \\ 2.262 \end{pmatrix}$$

5 ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

5.1 Постановка задачи

Пусть функция $y(x)$ задана множеством своих значений в точках $x = x_0, x_1, \dots, x_n$, (т.е. таблицей)

x	x_0	x_1	\dots	x_n
$y(x)$	y_0	y_1	\dots	y_n

Точки x_0, x_1, \dots, x_n называются узлами интерполяции.

Требуется найти значения функции $y(x)$ в точках $x \neq x_i, i = 0, 1, \dots, n$.

Поставленная проблема решается путем приближенной замены функции $y(x)$ другой функцией $F(x)$, заданной аналитическим выражением, которую можно вычислить при любом значении x из заданного интервала.

Приближение функции $y(x)$ более простой функцией $F(x)$ называется аппроксимацией.

Если значения аппроксимирующей функции в узлах интерполяции совпадают с табличными значениями заданной функции $y(x)$, то такой способ введения аппроксимирующей функции $F(x)$ называется интерполяцией.

Таким образом, интерполяцией называется построение аппроксимирующей функции $F(x)$, удовлетворяющей условиям

$$F(x_0) = y_0, F(x_1) = y_1, \dots, F(x_n) = y_n. \quad (5.1)$$

Условия (5.1) называются условиями Лагранжа.

Интерполяционной функцией называется аппроксимирующая функция $F(x)$, удовлетворяющая условиям Лагранжа.

Задача интерполяции в нахождении приближенных значений табличной функции $y(x)$ при аргументах x , не совпадающих с узлами интерполяции, путем вычисления интерполяционной функции $F(x)$. Если значение аргумента расположено внутри интервала $[x_0, x_n]$, то нахождение приближенного значения табличной функции $y(x)$ называется интерполяцией, если требуется найти значение аппроксимирующей функции вне интервала $[x_0, x_n]$, то этот процесс называется экстраполяцией. Происхождение этих терминов связано с латинскими словами: inter-между, extra-вне, pole-узел.

Геометрически задача интерполяции состоит в построении кривой $y = F(x)$, которая проходит через заданное множество точек $M_i(x_i, y_i), i = 0, 1, \dots, n$. На рис.5.1. показано построение интерполяционной кривой. Рассмотрим теперь методы построения интерполяционной функции.

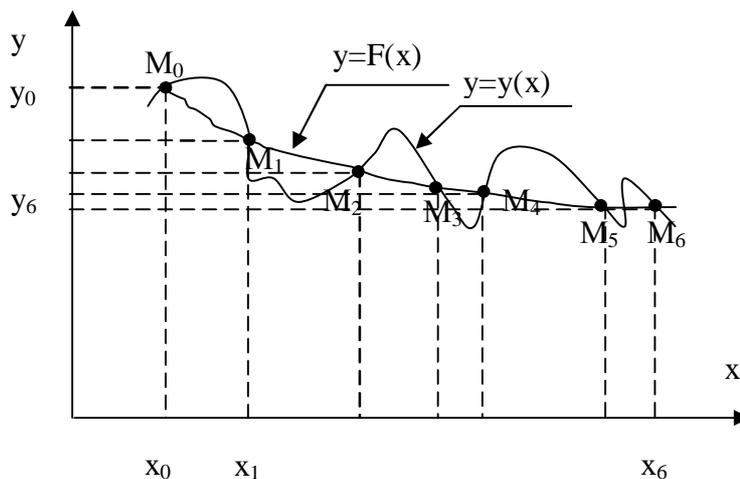


Рис.5.1. Построение интерполяционной кривой

5.2 Интерполяция каноническим полиномом

Будем искать интерполирующую функцию $F(x)$ в виде канонического полинома степени n :

$$F(x) = P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (5.2)$$

Коэффициенты полинома a_i определяются из условий Лагранжа $P_n(x_i) = y_i, i = 0, 1, \dots, n$, что с учетом выражения (5.2) дает систему линейных уравнений с $n + 1$ неизвестными:

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0 \\ a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1 \\ \dots \\ a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n \end{cases} \quad (5.3)$$

Систему уравнений (5.3) можно записать в виде

$$\sum_{i=1}^n a_i x^i = y_i, \quad i = 0, 1, \dots, n$$

или в матричной форме

$$C \cdot a = y \quad (5.4)$$

где a - вектор, содержащий неизвестные коэффициенты полинома a_i , y - вектор, содержащий значения функции в узлах интерполяции, y_i , а матрица C имеет вид

$$C = \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}$$

Система линейных алгебраических уравнений имеет единственное решение, если определитель матрицы C отличен от нуля. Этот определитель, называемый определителем Вандермонда, отличен от нуля, если среди узлов x_i нет совпадающих.

Решение системы (5.3) в матричном виде имеет вид

$$a = C^{-1} \cdot y$$

где C^{-1} - обратная матрица ($C^{-1} \cdot C = E$, E - единичная матрица).

Рассмотрим пример реализации полиномиальной интерполяции в Mathcad.

Пример 5.1

По заданным значениям функции в узлах интерполяции построить канонический полином. Построить график полученного интерполяционного полинома.

Решение

1. Определяем координаты узлов интерполяции
- 2.

$$X := \begin{pmatrix} 2.3 \\ 3.2 \\ 4.8 \\ 7.1 \end{pmatrix} \quad Y := \begin{pmatrix} 6.5 \\ 16.3 \\ 8.4 \\ 7.1 \end{pmatrix}$$

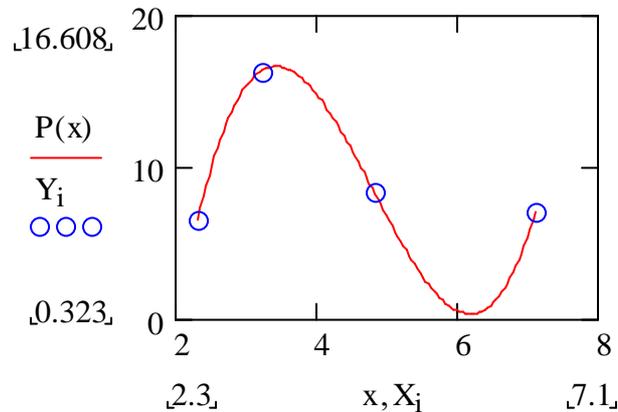
3. Определяем коэффициенты интерполяционного полинома

$$i := 0..3 \quad j := 0..3$$

$$C_{i,j} := (X_j)^i \quad a := C^{-1} \cdot Y \quad P(x) := \sum_{k=0}^3 a_k \cdot x^k$$

4. Строим график интерполяционной функции

$$x := 2.3, 2.34 .. 7.1$$



5.4 Многоинтервальная интерполяция

Рассмотренная выше полиномиальная интерполяция не всегда дает удовлетворительные результаты. Несмотря на выполнение условий Лагранжа в узлах, интерполяционная функция может иметь значительное отклонение от аппроксимируемой кривой между узлами.

С увеличением количества узлов возрастает и степень интерполяционного полинома, что

приводит к резкому увеличению погрешности в результате возникновения так называемого явления волнистости. Для того чтобы избежать высокой степени полинома отрезок интерполяции разбивают на несколько частей и на каждом частичном интервале строят самостоятельный полином невысокой степени. Рассмотрим наиболее часто используемые виды многоинтервальной интерполяции, а также способы их реализации в Mathcad.e.

5.4.1 Линейная интерполяция

Кусочно-линейная интерполяция является простейшим видом многоинтервальной интерполяции, при которой исходная функция на каждом частичном интервале $[x_i, x_{i+1}]$ аппроксимируется отрезком прямой, соединяющей точки (x_i, y_i) и (x_{i+1}, y_{i+1}) . Запишем уравнение этой прямой.

Уравнение прямой на отрезке $[x_i, x_{i+1}]$ имеет вид $r_i(x) = a_i + b_i x$. Для определения коэффициентов a и b воспользуемся условиями $r_i(x_i) = a_i + b_i x_i = y_i$ и $r_i(x_{i+1}) = a_i + b_i x_{i+1} = y_{i+1}$. Решая эти уравнения, найдем коэффициенты a_i и b_i .

$$a_i = y_{i-1} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \cdot x_i,$$

$$b_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

Таким образом, в случае линейной интерполяции интерполяционная функция на каждом из отрезков $[x_i, x_{i+1}]$ определяется по формуле

$$F(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \cdot (x - x_i)$$

В Mathcad'е имеется встроенная функция `linterp`, осуществляющая линейную интерполяцию. Обращение к этой функции имеет вид

`linterp(vx,vy,x)`.

Аргументами этой функции являются два вектора **`vx`** и **`vy`**, содержащие координаты узлов интерполяции и независимая переменная **`x`**.

Рассмотрим пример построения линейной интерполяции в Mathcad.

Пример 5.2.

Для точек (1.1;7.7), (2.8;9.8), (3.7;6.3), (4.1;1.6) построить линейную интерполяцию. По полученной функции найти прогноз в точке $x=3.5$.

Решение:

1. Задаем координаты узлов интерполяции

$$X := \begin{pmatrix} 1.1 \\ 2.8 \\ 3.7 \\ 4.1 \end{pmatrix} \quad Y := \begin{pmatrix} 7.7 \\ 9.8 \\ 6.3 \\ 1.6 \end{pmatrix}$$

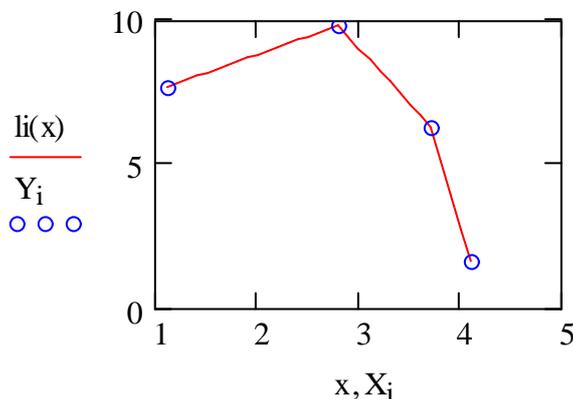
2. Определяем функцию **`li(x)`**, осуществляющую линейную интерполяцию.

$$li(x) := linterp(X, Y, x)$$

3. Строим график линейной интерполяционной функции.

`i := 0..3`

`x := 1.1, 1.2.. 4.1`



4. Находим прогноз в точке $x=3.5$

$$li(3.5) = 7.078 \blacksquare$$

5.4.2. Сплайн-интерполяция

Существенным недостатком линейной интерполяции является то, что в точках стыка разных интерполяционных полиномов оказывается разрывной их первая производная. Этот недостаток устраняется при использовании особого вида много интервальной интерполяции. интерполяции сплайнами (англ. *spline* . рейка, линейка).

Сплайн . это функция, которая на каждом частичном интервале представляется полиномом некоторой степени, а на всем заданном отрезке непрерывна вместе с несколькими своими производными. На практике широкое применение получили сплайны третьей степени (кубические сплайны).

На интервале $[x_{i-1}, x_i]$ кубический сплайн можно представить в виде

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \quad (5.5)$$

где a_i, b_i, c_i, d_i - коэффициенты сплайнов; $i = 1, 2, \dots, n$ - номер сплайна(интервала).

Коэффициенты сплайна определяются из следующих условий:

1) условия Лагранжа

$$s_i(x_i) = y_i, \quad s_i(x_{i-1}) = y_{i-1} \quad (5.6)$$

2) условия непрерывности первой и второй производных сплайнов в узлах

$$s_i'(x_i) = s_{i+1}'(x_i), \quad s_i''(x_i) = s_{i+1}''(x_i) \quad (5.7)$$

Кроме этого необходимо задать дополнительные условия на концах интервала, т.е. в точках x_0 и x_n . Если потребовать нулевой кривизны сплайна в этих точках, то дополнительными условиями будут являться равенства нулю вторых производных сплайнов на концах интервала интерполяции:

$$s_1'(x_0) = 0, \quad s_n''(x_n) = 0 \quad (5.8)$$

Дополнительные условия могут быть и иными, их выбор зависит от конкретной задачи.

Подставив выражение (5.5) в условия (5.6), (5.7), (5.8), получим систему из $4n$ уравнений относительно коэффициентов сплайнов a_i, b_i, c_i и d_i . Решив эту систему, определим коэффициенты сплайна.

Интерполяция сплайнами имеет очень простую и наглядную физико-механическую аналогию. Если попытаться совместить упругую металли-

ческую линейку с узловыми точками, то форма, которую примет в этом случае линейка будет совпадать с графиком кубического сплайна (сплошная линия на рис.1). Вне узловых точек, где линейка свободна, она описывается уравнением прямой. Соответствующее поведение сплайна обеспечивается условием (5.8)

Для кубической сплайн-интерполяции в Mathcad используется встроенная функция `interp`. Обращение к этой функции имеет вид:

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

Функция `interp` имеет следующие аргументы:

- `vs` – вектор вторых производных, созданный функцией `lspline(X,Y)`, `pspline(X,Y)` или `cspline(X,Y)`;
- `X` – вектор опытных значений аргумента, расположенных в порядке возрастания;
- `Y` – вектор опытных значений функции;
- `x` – значение аргумента, при котором вычисляется интерполирующее значение

Перед вызовом функции `interp` необходимо определить первый из ее аргументов – вектор `vs`. Делается это при помощи одной из трех встроенных функций тех же аргументов `X` и `Y`:

- `lspline(X,Y)` – создает вектор коэффициентов кривой, которая приближается к прямой линии в граничных точках;
- `pspline(X,Y)` – создает вектор коэффициентов кривой, которая приближается к квадратичной параболе в граничных точках;
- `cspline(X,Y)` – создает вектор коэффициентов кривой, которая приближается к кубической параболе в граничных точках;

Выбор конкретной функции сплайн-коэффициентов влияет на интерполяцию вблизи конечных точек интервала.

Пример 5.3

Для точек (1.1;7.7), (2.8;9.8), (3.7;6.3), (4.1;1.6) построить кубическую сплайн-интерполяцию. По полученной функции найти прогноз в точке $x=3.5$.

Решение:

1. Определяем координаты узлов интерполяции

$$X := \begin{pmatrix} 1.1 \\ 2.8 \\ 3.7 \\ 4.1 \end{pmatrix} \quad Y := \begin{pmatrix} 7.7 \\ 9.8 \\ 6.3 \\ 1.6 \end{pmatrix}$$

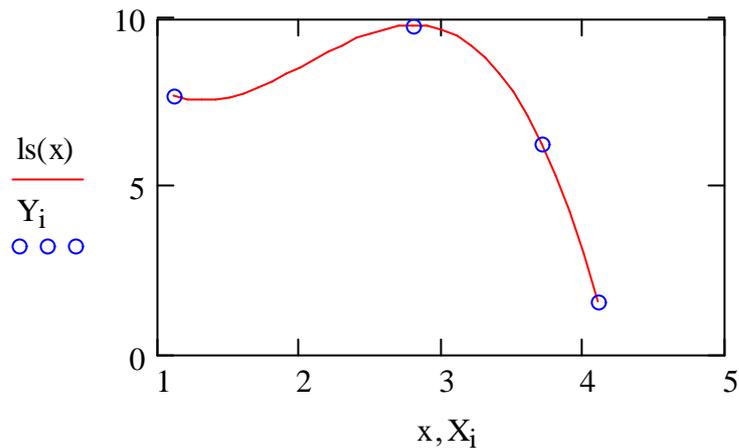
- 2 Определяем вектор коэффициентов сплайна

vs := cspline(X, Y)

3 Строим график кубической сплайн интерполяции

i := 0..3

x := 1.1, 1.2.. 4.1



4 Находим прогноз в точке x=3.5

$$ls(3.5) = 7.81$$

6. ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ И ИНТЕГРИРОВАНИЕ

6.1 Дифференцирование функций, заданных аналитически

По определению, производная функции $f(x)$:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (6.1)$$

Переходя от бесконечно малых к конечным разностям, получим приближенную формулу численного дифференцирования:

$$f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (6.2)$$

Из формулы (6.2) следует простой вычислительный алгоритм:

1. Задать значение точки, в которой вычисляется производная
2. Задать значение приращения Δx

3. Вычислить производную в соответствии с формулой (6.2)

Замена бесконечно малых приращений конечными – причина возникновения ошибки. Для её оценки, разложим функцию $f(x)$ в точке $x + \Delta x$ в ряд Тэйлора:

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!} \Delta x + \frac{f''(x)}{2!} (\Delta x)^2 + \dots + \frac{f^{(n)}(x)}{n!} (\Delta x)^n \dots \quad (6.3)$$

Подставив (6.3) в (6.2), после приведения подобных членов получим:

$$f'(x) \approx f'(x) + \frac{f''(x)}{2!} \Delta x + \dots \quad (6.4)$$

Отсюда видно, что все члены, начиная со второго, определяют отличие значения производной от её точного значения.

Основной член погрешности $f''(x) \cdot \Delta x / 2!$

Пример 1.

Рассмотрим решение задачи о численном вычислении производных в Mathcad.

Решение:

1. Задание функции и ее производная:

$$f(x) := \sin(x) \quad df(x) := \frac{d}{dx} f(x)$$

2. Задание пространственной сетки:

$$\begin{aligned} X_{\min} &:= 0 & X_{\max} &= \frac{\pi}{2} \\ N &:= 2000 & i &:= 0..N \end{aligned}$$

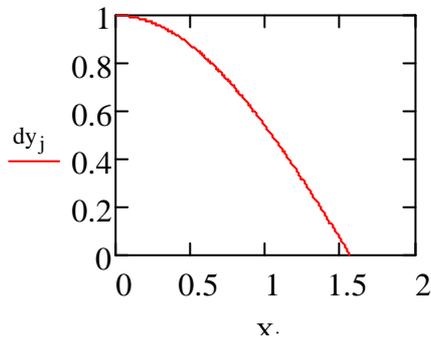
$$\Delta x = \frac{X_{\max} - X_{\min}}{N} \quad X_i := X + \Delta x \cdot i$$

3. Вычисление производной функции в соответствии с (6.2):

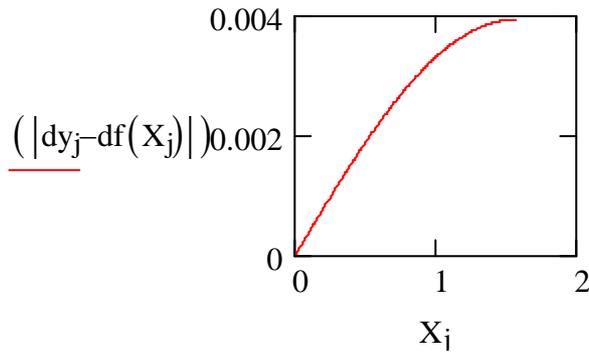
$$j := 0..N-1$$

$$dy_j = \frac{f(x_{j+1}) - f(x_j)}{\Delta x}$$

4. Строим график приближенного значения производной



5. Строим график модуля разности между точным и приближенным значением производной



6.2 Интегрирование функций, заданных аналитически

С геометрической точки зрения, определенный интеграл

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

есть площадь фигуры, ограниченная графиком функции $f(x)$ и прямыми $x=a$ $x=b$ (рис.6.1)

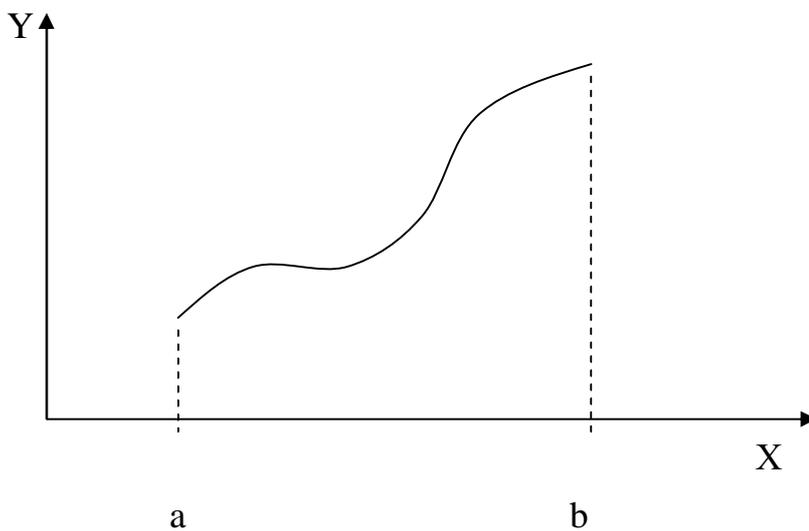


Рис 6.1 Геометрический смысл определенного интеграла

Разделим отрезок $[a; b]$ на n равных отрезков длиной ΔX , где

$$\Delta X = \frac{b-a}{n}$$

Тогда координата правого конца i -го отрезка определяется по формуле:

$$X_i = X_0 + i\Delta X$$

где $x_0 = a$, $i = 0, 1, \dots, n$

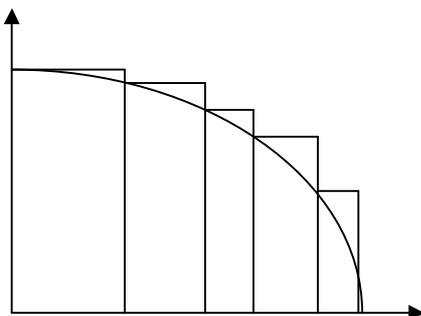


Рис 6.2 Метод левых прямоугольников

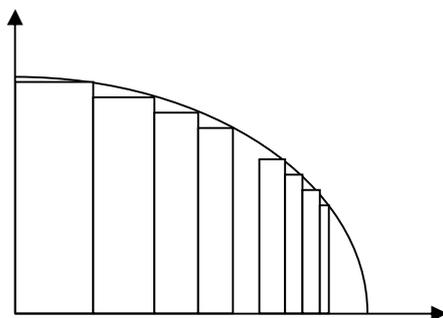


Рис 6.3. Метод правых прямоугольников

Тогда значение определенного интеграла вычисляется по формулам:

$$F_L = \sum_{i=0}^{N-1} f(x_i)\Delta x \text{ - метод левых прямоугольников(рис. 6.2);}$$

$$F_R = \sum_{i=1}^N f(x_i)\Delta x \text{ - метод правых прямоугольников(рис. 6.3).}$$

Пример 2

Рассмотрим решение задачи о вычислении интеграла $\int_0^{\frac{\pi}{2}} \sin(x) dx$ в пакете Mathcad. Соответствующий документ состоит из блоков.

1 Задание подынтегральной функции:

$$f(x) := \sin(x);$$

2 Задание пространственной сетки:

$$\begin{aligned} x_{\min} &:= 0 & x_{\max} &:= \frac{\pi}{2} \\ N &:= 2000 & i &:= 0 \dots N \\ \Delta x &:= \frac{x_{\max} - x_{\min}}{N} & x_i &:= x_{\min} + \Delta x \cdot i \end{aligned}$$

3 Вычисление значения интеграла по формуле левых и правых прямоугольников:

$$\begin{aligned} F_R &:= \left(\sum_{i=1}^N Y_i \right) \Delta x \\ F_L &:= \left(\sum_{i=0}^{N-1} Y_i \right) \Delta x \\ F_R &= 1.0003926 & F_L &= 0.9996072 \end{aligned}$$

Пример 3

Вычислить приближенное значение интеграла

$$\int_0^1 \frac{8}{(3x+4)^2} dx,$$

используя формулы левых и правых прямоугольников, при $n=1000$

Решение:

- 1 Задаём функцию $f(x)$, отрезок $[a, b]$ и функцию нахождения дифференциалов n -го порядка
- 2 Находим значение интеграла функции для использования его в дальнейшем решении для сравнения

$$\begin{aligned} f(x) &:= \frac{8}{(3x+4)^2} \\ a &:= 0 & b &:= 1 & n &:= 100 \end{aligned}$$

$$dif_n(x,n) := \frac{d^n}{dx^n} f(x)$$

$$\int_a^b f(x)dx = 0.285714286$$

3 Составим функцию **Integr** с входными параметрами:

a, b - левая и правая граница интеграла;

n - количество разбиений;

$char$ - если имеет значение "left", то идет подсчет по формуле левых прямоугольников; любое другое - по формуле правых прямоугольников.

$$Integr(a,b,n,char) := \left[\begin{array}{l} hf \leftarrow \frac{b-a}{n} \\ for_i \in 0..n \\ \left[\begin{array}{l} xf_i \leftarrow a + hf \cdot i \\ dif_i \leftarrow |dif_n(xf_i,1)| \end{array} \right] \\ \left[\begin{array}{l} if \left[char = "left", \left(hf \cdot \sum_{t=0}^{n-1} f(xf_t) \right), \left(hf \cdot \sum_{t=1}^n f(xf_t) \right) \right] \\ \left[\frac{(b-a) \cdot hf}{2} \right] \cdot \max(dif) \end{array} \right] \end{array} \right]$$

Эта функция возвращает значение интеграла и погрешность найденного значения

Результаты вычислений:

$$Integr(a,b,n,"left") = \begin{pmatrix} 0.287 \\ 0.004 \end{pmatrix}$$

$$Integr(a,b,n,"right") = \begin{pmatrix} 0.284 \\ 0.004 \end{pmatrix}$$

Погрешность показывает, что полученное значение интеграла верно до третьего знака после запятой.

6.3 Метод Монте-Карло

Пусть необходимо вычислить определенный интеграл

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

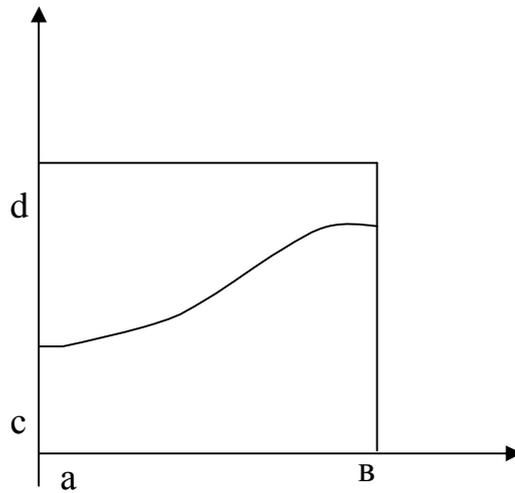


Рис 6.1. К иллюстрации метода Монте-Карло.

Сгенерируем N пар случайных чисел в прямоугольнике $a \leq x_i \leq b$, $c \leq y_i \leq d$.

Тогда доля точек (x_i, y_i) , с условием $y_i \leq f(x_i)$ является оценкой отношения интеграла от $S(x)$ к площади рассматриваемого прямоугольника.

Оценка интеграла может быть получена по формуле:

$$F_N = A \frac{n_s}{N}$$

где n_s - число точек, удовлетворяющих условию $y_i \leq f(x_i)$, N - полное кол-во точек, A - площадь прямоугольника

7. РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ И СИСТЕМ.

7.1 Основные определения

Обыкновенным дифференциальным уравнением n –го порядка называется соотношение вида

$$H(t, x, x', x'', \dots, x^{(n)}) = 0 \quad (7.1)$$

между независимым переменным t , его функцией $x(t)$ и производными $x'(t), x''(t), \dots, x^{(n)}(t)$. Решением этого дифференциального уравнения называется функция $x(t)$, которая после подстановки в уравнение обращает его в тождество.

Системой дифференциальных уравнений 1-го порядка называется система вида:

$$x' = f(x, t) \quad (7.4)$$

и начальному условию при $t = t_0$:

$$x(t_0) = x_0 \quad (7.5)$$

На примере этой задачи изложим идею метода Эйлера. Пусть требуется найти решение задачи (1.4), (1.5) на интервале $[t_0, t_n]$. На этом интервале построим систему равноотстоящих точек с достаточно малым шагом h :

$$h = \frac{t_n - t_0}{n}, t_i = t_0 + h, i = 0, 1, 2, \dots, n$$

Для вычисления значения функции в точке t_1 разложим функцию $x(t)$ в окрестности точки t_0 в ряд Тейлора:

$$x(t_1) = x(t_0) + h \cdot x'(t_0) + \frac{h^2}{2} \cdot x''(t_0) + \dots \quad (7.6)$$

При достаточно малом h мы можем пренебречь членами выше 2-го порядка, и с учетом соотношения $x'(t_0) = f(x_0, t_0)$ получим формулу для вычисления приближенного значения функции $x(t)$ в точке t_1 :

$$x(t_1) \approx x_1 = x_0 + h \cdot f(x_0, t_0) \quad (7.7)$$

Рассматриваем найденную точку (x_1, t_1) как начальное условие задачи Коши, найдем аналогично значение решения $x(t)$ в точке t_2 :

$$x_2 = x_1 + h \cdot f(x_1, t_1) \quad (7.8)$$

Аналогичным образом находим значения решения в последующих точках:

$$x_{i+1} = x_i + h \cdot f(x_i, t_i), i = 0, 1, \dots, n-1 \quad (7.9)$$

Правило (1.9) носит название метода Эйлера. Иногда его называют методом ломаных или методом касательных.

Как видно из формулы (1.6), метод Эйлера имеет погрешность на каждом шаге (локальную погрешность) пропорциональную h^2 . Суммарная погрешность метода Эйлера после n шагов пропорциональна $n \cdot h^2$. Поскольку $n = 1/h$, то суммарная погрешность метода Эйлера пропорциональна h , т.е. метод Эйлера имеет точность первого порядка по h . Метод Эйлера имеет простую геометрическую интерпретацию (см. рис. 7.1)

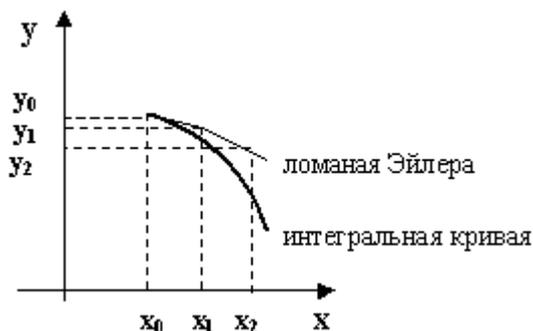


Рис 7.1 Нахождение решения методом Эйлера.

Пример 1

Решить задачу Коши для дифференциального уравнения

$$\frac{dy}{dx} = x^2 + y, \quad y(0) = 5$$

1 Определяем функцию, стоящую в правой части уравнения

$$f(x, y) := x^2 + y$$

2 Задаем начальные условия

$$x_0 := 0$$

$$y_0 := 5$$

3 Определяем функцию, которая находит решение дифференциального уравнения методом Эйлера. Она имеет следующие аргументы:

- x_0, x_N -левый и правый концы интервала интегрирования
- y_0 -значение решения в точке x_0
- N -число точек разбиения отрезка $[x_0, x_N]$
- f -имя функции, определяющей правую часть дифференциального уравнения

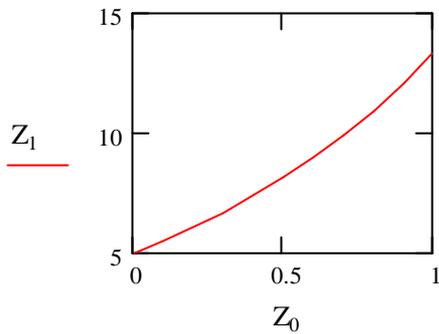
$$\text{Euler}(x_0, x_N, y_0, N, f) := \left| \begin{array}{l} h \leftarrow \frac{(x_N - x_0)}{N} \\ X_0 \leftarrow x_0 \\ Y_0 \leftarrow y_0 \\ \text{for } i \in 1..N \\ \quad \left| \begin{array}{l} X_i \leftarrow X_{i-1} + h \\ Y_i \leftarrow Y_{i-1} + h \cdot f(X_{i-1}, Y_{i-1}) \end{array} \right. \\ \left(\begin{array}{l} X \\ Y \end{array} \right) \end{array} \right.$$

4 Находим решение дифференциального уравнения на отрезке $[0;3]$ методом Эйлера

$$x_N := 1 \quad N := 10$$

$$Z := \text{Euler}(x_0, x_N, y_0, N, f)$$

5. Строим график полученного решения



Встроенные функции Mathcad для решения обыкновенных дифференциальных уравнений и систем.

Рассмотрим встроенные функции Mathcad, предназначенные для решения обыкновенных дифференциальных уравнений.

➤ Функция Odesolve

Применение функции Odesolve требует задания вычислительного блока, состоящего из трех частей:

- 1) Ключевого слова Given
- 2) Дифференциального уравнения или системы обыкновенных дифференциальных уравнений и соответствующих граничных или начальных условий
- 3) Функции Odesolve ([vector],t,b,[n]), где vector-необязательный параметр(только для систем дифференциальных уравнений), содержащий имена неизвестных функций, t-имя независимой переменной, b-конечная точка интервала интегрирования, n-необязательный параметр, определяющий число шагов интегрирования, на которых вычисляется решение дифференциального уравнения или системы.

Результатом выполнения функции Odesolve является функция, являющаяся решением обыкновенного дифференциального уравнения(или вектор функции для системы ОДУ).

Пример:

Решить задачу Коши

$$x''' + x' + x = e^{-t} + \cos(t),$$

$$x(0) = 1, \quad x'(0) = 0, \quad x''(0) = 0.4$$

1 Задаем уравнение и начальные условия

Given

$$\frac{d^3}{dt^3}x(t) + \frac{d}{dt}x(t) + x(t) = \exp(-t) + \cos(t)$$

$$x(0) = 1$$

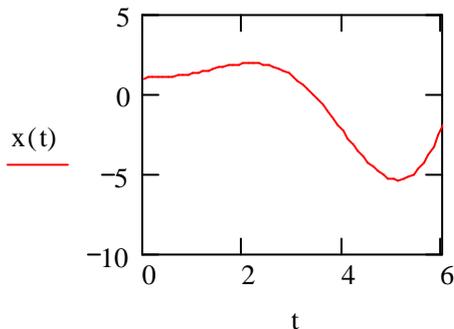
$$x'(0) = 0$$

$$x''(0) = 0.4$$

2 Вызываем функцию Odesolve

```
x := Odesolve(t,6)
```

```
t := 0,0.1..6
```



$x(t) =$

1
1.002
1.009
1.022
1.041
1.067
1.1
1.139

➤ Функция rkfixed

Функция `rkfixed` находит решение дифференциального уравнения или системы обыкновенных дифференциальных уравнений 1-го порядка. Обращение к функции имеет вид:

`Rkfixed(init,t1,t2,npoints,D)`.

Аргументы функции:

`Init`-вектор начальных условий

`t1,t2`- начало и конец интервала интегрирования

`npoints`- количество точек разбиений интервала $[t1,t2]$

`D`-вектор правых частей дифференциального уравнения или системы

Пример

Решить задачу Коши с использованием функции `rkfixed`

$$\begin{aligned}x' + x + x^3 &= \cos(t) \\ x(0) &= 2\end{aligned}$$

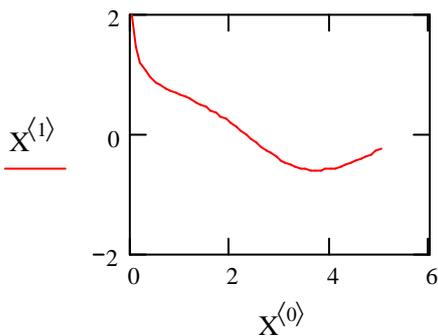
1. Задаем правую часть дифференциального уравнения

$$D(t, x) := -x - x^3 + \cos(t)$$

2. Вызываем функцию `rkfixed`

$$X := \text{rkfixed}(2, 0, 5, 50, D)$$

3. Строим график решения



Помимо функции `rkfixed` в Mathcad используется функция `rkadapt`. Вызов функции `rkadapt` аналогичен вызову функции `rkfixed`. Функция `rkfixed` ищет решение с постоянным шагом. Функция `rkadapt` проверяет, как быстро изменяется решение и соответственно изменяет шаг интегрирования.

Рассмотренные выше функции позволяют находить решения и систем обыкновенных дифференциальных уравнений.

Пример:

Решить задачу Коши для системы обыкновенных дифференциальных уравнений

$$\begin{aligned}x' &= \cos(x \cdot y) \\y' &= \sin(x + t \cdot y), \\x(0) &= 0, \quad y(0) = 0\end{aligned}$$

на интервале $[0, 10]$.

Решение:

1 Определяем границы интервала интегрирования

`a := 0`
`b := 10`

2 Задаем уравнения и начальные условия

`Giver`
`x'(t) = cos(x(t)·y(t))`
`y'(t) = sin(x(t) + t·y(t))`

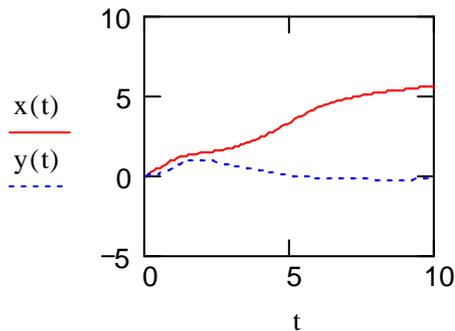
$$x(a) = 0$$

$$y(a) = 0$$

$$\begin{pmatrix} x \\ y \end{pmatrix} := \text{Odesolve} \left[\begin{pmatrix} x \\ y \end{pmatrix}, t, b \right]$$

3 Строим график решения системы

$t := 0, 0.1.. 10$



8 ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

8.1 Примеры уравнений

Определение 1: Уравнением с частными производными (УЧП) n -го порядка с t независимыми переменными x_1, x_2, \dots, x_m называется соотношение между независимыми переменными x_1, x_2, \dots, x_m , неизвестной функцией $u(x_1, x_2, \dots, x_m)$ и ее частными производными до n – го порядка включительно:

$$F(x_1, \dots, x_m, u, u_{x_1}, \dots, u_{x_m}, \dots, u_{x_m x_m \dots x_m}) = 0$$

Для упрощения записи мы используем следующие обозначения

$$u_{x_1} = \frac{\partial u}{\partial x_1}, u_{x_1 x_1} = \frac{\partial^2 u}{\partial x_1^2}, u_{x_1 x_2} = \frac{\partial^2 u}{\partial x_1 \partial x_2}, \dots$$

и т.д.

Примеры УЧП:

$$u_t = u_{xx} \text{ (одномерное уравнение теплопроводности)}$$

$$u_t = u_{xx} + u_{yy} \text{ (двумерное уравнение теплопроводности)}$$

$$u_{tt} = u_{xx} + u_{yy} + u_{zz} \text{ (трехмерное волновое уравнение)}$$

$$u_{tt} = u_{xx} + \alpha u_t + \beta u \text{ (телеграфное уравнение)}$$

Существует множество методов решения уравнений в частных производных. Перечислим некоторые из них

- Метод разделения переменных
- Метод интегральных преобразований
- Метод преобразования координат
- Численные методы (метод конечных разностей, метод конечных элементов и др.)
- Методы теории возмущений
- Метод функций Грина
- Вариационные методы

и др.

1. Основные типы линейных уравнений в частных производных

Линейным уравнением в частных производных 2-го порядка с двумя независимыми переменными называется уравнение вида:

$$Au_{xx} + Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G, \text{ где}$$

A, B, C, D, E, F - константы или функции независимых переменных x и y .

Пример:

- $u_{tt} = e^{-t} \cdot u_{xx} + \sin t$ (линейное уравнение)
- $u \cdot u_{xx} + u_t = 0$ (нелинейное уравнение)

В зависимости от знака величины $B^2 - 4 \cdot A \cdot C$ уравнения можно отнести гиперболическому, эллиптическому или эллиптическому типу.

Уравнение в некоторой точке будем называть уравнением

гиперболического типа, если в этой точке $B^2 - 4 \cdot A \cdot C < 0$,

эллиптического типа, если в этой точке $B^2 - 4 \cdot A \cdot C > 0$,

параболического типа, если в этой точке $B^2 - 4 \cdot A \cdot C = 0$,

Уравнения параболического типа описывают процессы теплопроводности и диффузии. К параболическому типу относится уравнение $u_t = u_{xx}$.

Уравнения гиперболического типа описывают волновые процессы. Примером гиперболического уравнения является уравнение $u_{tt} = u_{xx}$

Уравнения эллиптического типа описывают стационарные процессы (т.е. процессы, не зависящие от времени). Уравнением эллиптического типа является уравнение

$$u_{xx} + u_{yy} = 0.$$

Если коэффициенты уравнения не являются постоянными, то оно может иметь разный тип в разных областях. Примером является уравнение

$$y \cdot u_{xx} + u_{yy} = 0.$$

Для этого уравнения $B^2 - 4 \cdot A \cdot C = -4 \cdot y$, поэтому уравнение является эллиптическим при $y > 0$, параболическим при $y = 0$, гиперболическим при $y < 0$.

8.2 Использование встроенных функций Mathcad для решения уравнений в частных производных.

Решение параболических уравнений.

Задача 1 Найти распределение температуры в стержне длиной $L (0 \leq x \leq L)$, если его начальная температура задается произвольной функцией $f(x)$. Температуры концов стержня равны $u(0,t) = u_1 = const$, $u(L,t) = u_2 = const$. На боковой поверхности стержня происходит теплообмен по закону Ньютона со средой, температура которой равна u_0 .

Начально-краевая задача, описывающая распределение температуры стержня, имеет вид:

$$\frac{\partial u}{\partial t} = a^2 \cdot \frac{\partial^2 u}{\partial x^2} - h \cdot (u - u_0), 0 \leq x \leq L, 0 < t < \infty$$

$$u(0,t) = u_1, \quad u(L,t) = u_2, \quad 0 < t < \infty,$$

$$u(x,0) = \varphi(x), \quad 0 < x < L.$$

В Mathcad 11 имеются встроенные функции для решения одномерных уравнений в частных производных `pdesolve` и `numol`.

Для решения этой задачи в Mathcad необходимо выполнить следующие действия:

- 1 Ввести служебное слово `Given`.
- 2 Ввести уравнения. Производные вводятся с помощью символа «.» как нижние индексы (например, u_t или u_{xx}). Для ввода знака равенства используется комбинация клавиш `Cntrl+”=”`. В левой части уравнения записывается первая производная по t , все остальное - в правой части уравнения. При вводе уравнения после имени неизвестной функции и ее производных должен присутствовать список аргументов, например $u(x,t), u_{xx}(x,t), u_t(x,t)$.
- 3 Ввести начальные и граничные условия, где производные вводятся как нижние индексы, а знак равенства через `Cntrl+”=”`.
- 4 Завершить решающий блок функцией

$$\text{pdesolve}(u, x, xrange, t, trange, xpts, tpts),$$

которая возвращает функцию двух переменных, являющуюся решением задачи.

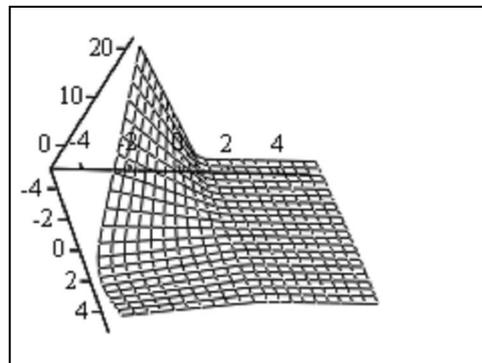
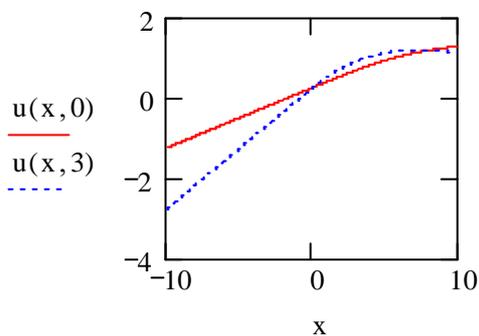
Функция `pdesolve` имеет следующие аргументы

- u – имя функции (без аргументов);
- x – имя пространственной переменной;

- $xrange$ – массив из двух элементов, указывает границы изменения пространственной переменной x ;
- t – имя временной переменной;
- $trange$ – массив, содержащий границы изменения переменной t .

Ниже приведен пример решения задачи в Mathcad.

```
f(x) := 0.25 + sin(0.15·x)
u0 := 2 L := 8 u1 := 0.25 u2 := 1.182 T := 3
h := 1 a := 5
Giver
u_t(x,t) = a^2 · u_xx(x,t) - h · (u(x,t) - u0)
u(0,t) = u1
u(L,t) = u2
u(x,0) = f(x)
u := Pdesolve[u, x, (0, L), t, (0, T), 30, 30]
uz := CreateMesh(u, 0, L, 0, T)
```



u

8.3 Решение гиперболических уравнений

Функция `pdesolve` позволяет решать дифференциальные уравнения и системы, в которые не входят уравнения, содержащие производную по t выше первой степени. Однако в любых гиперболических уравнениях есть вторая производная по t . Поэтому, чтобы решить гиперболическое уравнение, нужно его преобразовать в систему дифференциальных уравнений в частных производных, введя дополнительную функцию $v = \frac{\partial u}{\partial t}$.

Задача 2 Найти решение начально-краевой задачи для гиперболического уравнения:

$$\frac{\partial^2 \omega}{\partial t^2} = a^2 \cdot \frac{\partial^2 \omega}{\partial x^2} + \sin(xt), 0 < x < L, t > 0$$

$$\omega(x,0) = \varphi(x) \quad \omega_t(x,0) = \psi(x)$$

$$\omega(0,t) = \mu(t) \quad \omega(L,t) = \xi(t)$$

Решение: Введем дополнительную функцию

$$v(x,t) = \frac{\partial \omega(x,t)}{\partial t}$$

и таким образом сведем гиперболическое уравнение к системе дифференциальных уравнений в частных производных

$$v = \frac{\partial \omega}{\partial t}$$

$$\frac{\partial v}{\partial t} = a^2 \cdot \frac{\partial^2 \omega}{\partial x^2} + \sin(xt),$$

$$\omega(x,0) = \varphi(x) \quad v(x,0) = \psi(x)$$

$$\omega(0,t) = \mu(t) \quad \omega(L,t) = \xi(t).$$

Будем решать эту систему с помощью блока Given-Pdesolve.

Первым параметром в функции pdesolve будет массив имен функций, т.е.

$\begin{pmatrix} \omega \\ v \end{pmatrix}$. Результатом выполнения функции pdesolve будет вектор функция $\begin{pmatrix} \omega \\ v \end{pmatrix}$, представляющая решение системы.

Ниже приведен пример решения задачи для гиперболического уравнения в Mathcad.

Определяем функции $\Phi(x)$, $\psi(x)$, $\xi(t)$, $\mu(t)$

$$L := 2 \cdot \pi$$

$$\Phi(x) := \sin\left(\frac{x \cdot \pi}{L}\right)$$

$$\xi(t) := 0$$

$$a := 4$$

$$\psi(x) := \cos(x)$$

Задаем уравнения и граничные условия

Given

$$V_t(x,t) = a^2 \cdot \omega_{xx}(x,t) + \sin(x \cdot t)$$

$$\omega_t(x,t) = V(x,t)$$

$$V(x,0) = \psi(x)$$

$$\omega(x,0) = \Phi(x)$$

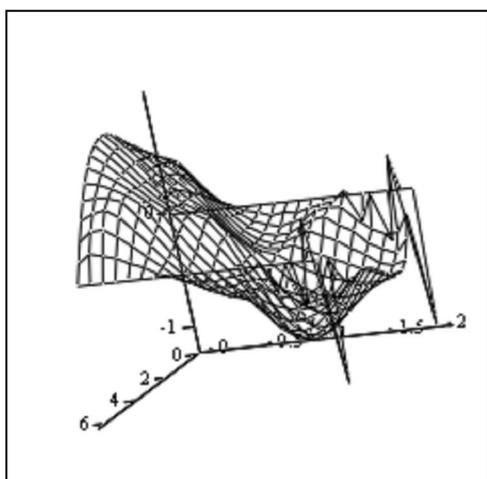
$$\omega(0,t) = \xi(t)$$

$$\omega(L,t) = \mu(t)$$

$$\begin{pmatrix} \omega \\ v \end{pmatrix} := \text{Pdesolve} \left[\begin{pmatrix} \omega \\ v \end{pmatrix}, x, \begin{pmatrix} 0 \\ L \end{pmatrix}, t, \begin{pmatrix} 0 \\ T \end{pmatrix}, 100, 100 \right]$$

Строим график решения

$M := \text{CreateMesh}(\omega, 0, L, 0, T)$



M

Литература

1. Конспект лекций по дисциплине «Информатика». Основы работы с пакетом прикладных программ MatchCAD по разделу «Численные методы решения прикладных задач с помощью пакета MatchCAD: для студентов технических специальностей / Сост.: А.А. Гончаров, Л.В. Васильева.- Краматорск: ДГМА, 2003.-44с.
2. Методические указания и контрольные задания по дисциплине «Информатика» для студентов заочного факультета инженерного направления обучения. Часть 3. / Сост. Л. В. Шевцова. – Краматорск: ДГМА, 2003 г. – 28 с.
3. **Макаров, Е. Г.** Инженерные расчеты в Mathcad: учебный курс/Макаров Е.Г. — СПб: Питер. 2005 — 448 с.: ил.

