Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Амурский государственный университет
(ФГБОУ ВО «АмГУ»)

ПРОГРАММНЫЕ КОМПЛЕКСЫ ОБЩЕГО НАЗНАЧЕНИЯ

сборник учебно-методических материалов

для направления подготовки 18.03.01 – Химическая технология

Благовещенск 2018 г.

Печатается по решению
редакционно-издательского совета
факультета математики и информатики
Амурского государственного
университета

Составитель: Рыженко А.В.

Программные комплексы общего назначения: сборник учебно-методических материалов для направления подготовки 18.03.01. – Благовещенск: Амурский гос. ун-т, 2018.

Рассмотрен на заседании кафедры математического анализа и моделирования «___» _______________ 201 ___ г., протокол № __________.

Методические материалы по дисциплине

Методические указания для выполнения практических заданий
Лабораторной работы 1 «Калькулятор»-Mathcad

Задание 1.1. Найдите пределы последовательностей $\{a_n\}$, $\{b_n\}$, $\{c_n\}$. Для указанных значений $\varepsilon=10^{-k}$ найдите такие $N(\varepsilon)$, чтобы все элементы последовательности с номерами $n>N(\varepsilon)$ совпадали с предельным значением до k-го знака после запятой. Для заданных значений M укажите такие значения $N(M)$, чтобы для всех членов бесконечно большой последовательности $\{c_n\}$ с номерами $n>N$ выполнялось неравенство $|c_n|>M$. Изобразите графически сходящиеся последовательности и их пределы. Изобразите графически бесконечно большую последовательность.

Порядок выполнения задания

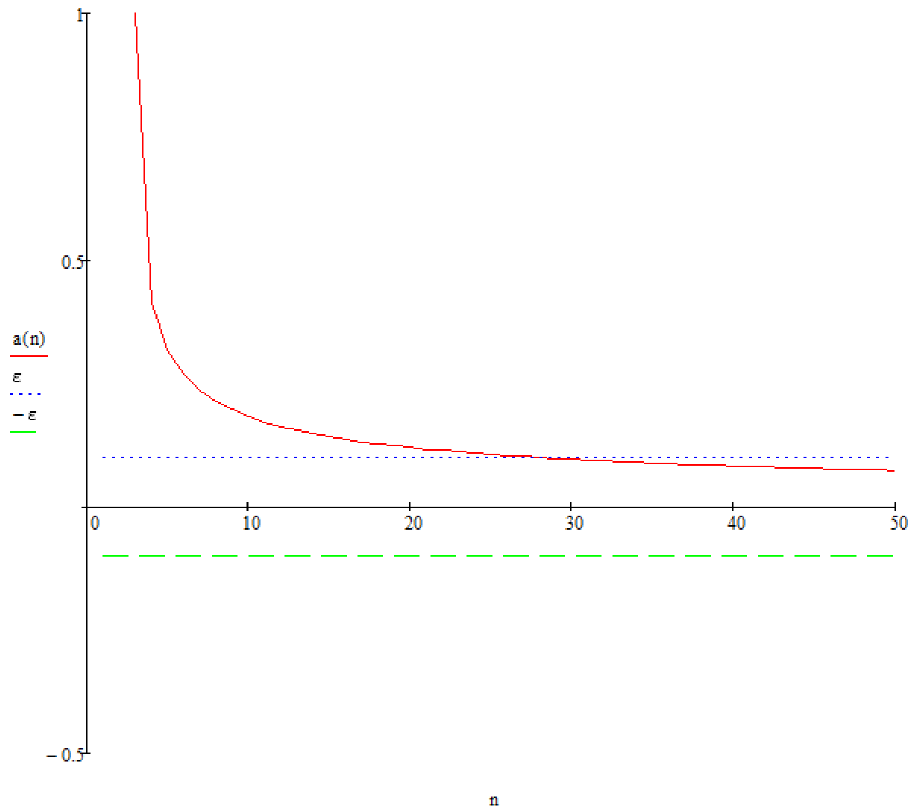
1. Убедитесь, что в меню установлен режим автоматических вычислений.
2. Введите элемент последовательности $\{a_n\}$ как функцию переменной n.
3. Используя символьную математику пакета, найдите аналитически предел a последовательности $\{a_n\}$.
4. Запишите уравнение для определения $N(\varepsilon)$ и решите его, используя символьную математику.
5. Для каждого заданного значения ε: вычислите по полученной формуле значение $N(\varepsilon)$.
6. Для любого $n>N(\varepsilon)$, например для $n=[N(\varepsilon)]+1$, вычислите $|a_n-a|$ и сравните его с ε.
7. Выбрав достаточно большое значение N, вычислите значения элементов последовательности для $n=1, 2, ..., N$ и изобразите на графике элементы последовательности как функции n.
8. Изобразите на том же графике горизонтальные прямые $y=a\pm\varepsilon$. Найдите на графике значение $N(\varepsilon)$.
9. Выполните аналогичные вычисления (пп. 1-7) для последовательности $\{b_n\}$.
10. Аналогично исследуйте бесконечно большую последовательность $\{c_n\}$.

Пример. Исследуйте последовательность $a_n=\sqrt{n-2}-\sqrt{n-3}$. Для ε, равного 0.1, 0.05, 0.01, укажите соответствующее значение $N(\varepsilon)$. Изобразите графически процесс сходимости.
Ниже приведен фрагмент рабочего документа с соответствующими вычислениями.

$a(n) := \sqrt{n-2} - \sqrt{n-3}$ $\qquad \lim\limits_{n \to \infty} a(n) \to 0$

Находим N(ε) $\qquad \left(\sqrt{n-2} - \sqrt{n-3}\right) - \varepsilon \qquad N(\varepsilon) := \dfrac{\varepsilon^4 + 10 \cdot \varepsilon^2 + 1}{4 \cdot \varepsilon^2} \qquad N(0.1) = 27.502 \qquad N(0.05) = 102.501 \qquad N(0.01) = 2502.5$

$a(28) = 0.099 \qquad a(103) = 0.05 \qquad a(2504) = 0.01 \qquad \varepsilon := 0.1 \qquad n := 1..50$



Задание 1.2. Найдите предел функции $f(x)$ в точке $x=a$. Для значения $\varepsilon=10^{-k}$ найдите $\delta(\varepsilon)$ такое, чтобы значения функции в точках $0<|x-a|<\delta$ совпадали с предельным значением до k-го знака после запятой. Изобразите графически поведение функции в окрестности точки $x=a$.

Порядок выполнения задания

1. Убедитесь, что в меню установлен режим автоматических вычислений.
2. Определите функцию $f(x)$.
3. Используя символьную математику пакета, найдите $\lim\limits_{x \to a} f(x)$.
4. Запишите уравнение для определения $\delta(\varepsilon)$ и решите его, используя символьную математику.
5. Для каждого заданного значения ε вычислите по полученной формуле значение $\delta(\varepsilon)$.
6. Вычислите $\max\limits_{0<|x-a|<\delta} |f(x)-A|$ и сравните его с ε.
7. Постройте график функции в окрестности точки $x=a$.
8. Изобразите на том же графике горизонтальные прямые $y=A\pm\varepsilon$. Найдите на графике значение $\delta(\varepsilon)$.

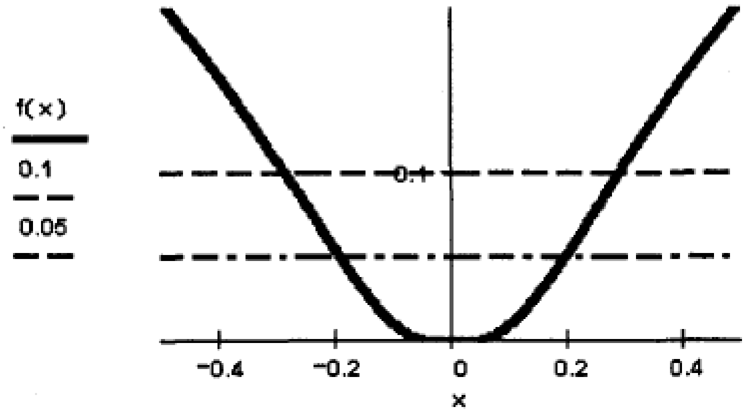
Пример

$$f(x) := \exp\left(\frac{-1}{\sqrt[3]{x^2}}\right) \qquad \lim_{x \to 0} f(x) \to 0$$

$$\exp\left(\frac{-1}{\sqrt[3]{x^2}}\right) - \varepsilon \qquad \begin{bmatrix} \left(\frac{-1}{\ln(\varepsilon)}\right)^{\left(\frac{3}{2}\right)} \\ -\left(\frac{-1}{\ln(\varepsilon)}\right)^{\left(\frac{3}{2}\right)} \end{bmatrix} \qquad \delta(\varepsilon) := \left(\frac{-1}{\ln(\varepsilon)}\right)^{\left(\frac{3}{2}\right)}$$

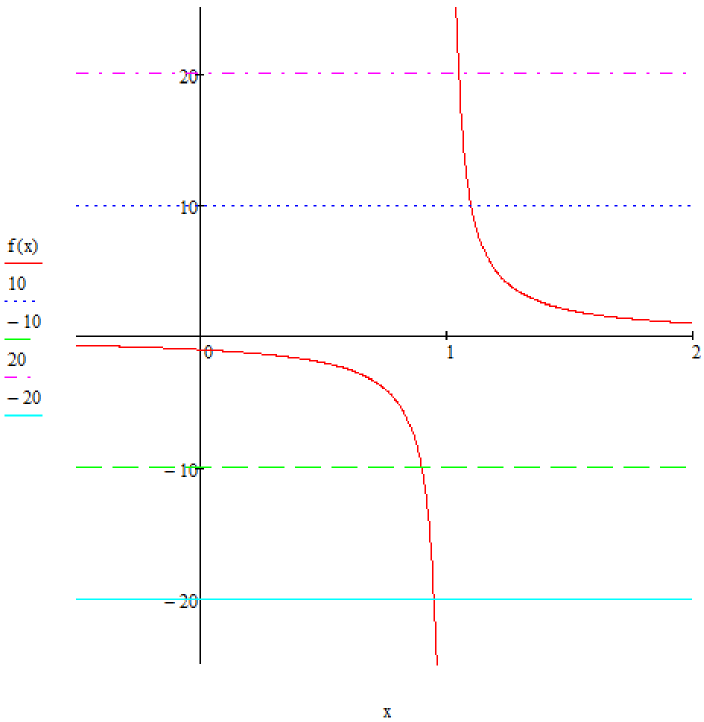
$\delta(0.1) = 0.286 \qquad \delta(0.05) = 0.193$

$f(0.286) = 0.1 \qquad f(0.193) = 0.05$



$f(x) := \frac{1}{x-1}$ $\lim_{x \to 1} f(x) \to \text{undefined}$ $\frac{1}{x-1} - M$ $\delta(M) := \frac{1}{M} + 1$

$\delta(10) = 1.1$ $\delta(20) = 1.05$ $f(1.1) = 10$ $f(1.05) = 20$



Задание 1.3. Сравните бесконечно малые в нуле функции. Постройте графики функций в окрестности предельной точки.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Определите все заданные функции.
3. Вычислите аналитически пределы функций в нуле, чтобы убедиться в том, что все заданные функции – бесконечно малые в указанной точке.
4. Сравните заданные бесконечно малые функции, вычислив пределы их отношений в нуле.
5. Сформулируйте и запишите в рабочем документе вывод.
6. Постройте графики всех заданных функций, подтверждающие сходство и различие поведения бесконечно малых в окрестности предельной точки.

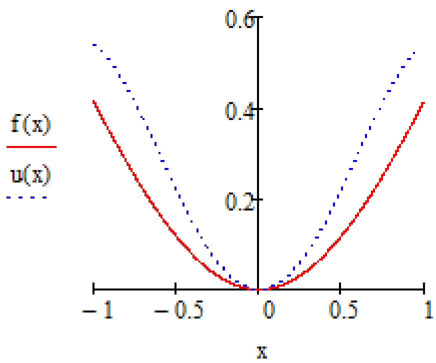
Пример

$f(x) := \sqrt{1 + x^2} - 1$

$\lim_{x \to 0} f(x) \to 0$

$u(x) := x^2 \cos(x)$

$\lim_{x \to 0} u(x) \to 0$



$$p(x) := e^{x^2} - 1 \qquad \lim_{x \to 0} p(x) \to 0$$

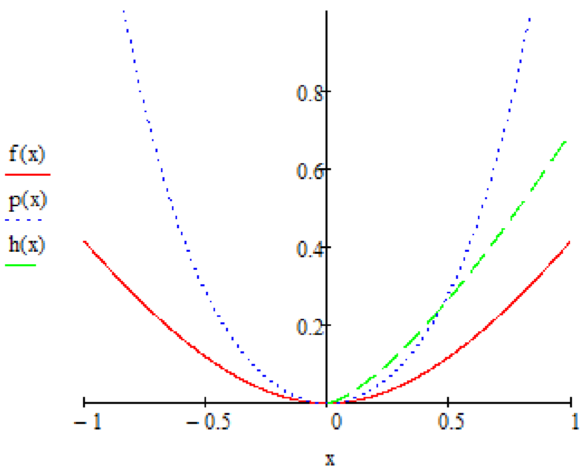
$$h(x) := x\ln\left(1 + \sqrt{x}\right) \qquad \lim_{x \to 0} h(x) \to 0$$

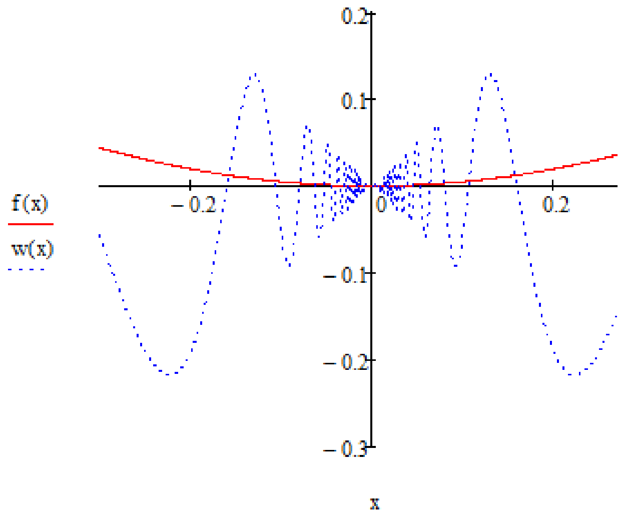
$$v(x) := 1 - \cos\left(x^3\right) \qquad \lim_{x \to 0} v(x) \to 0$$

$$w(x) := x\sin\left(\frac{1}{x}\right) \qquad \lim_{x \to 0} w(x) \to 0$$

$$\lim_{x \to 0} \frac{f(x)}{p(x)} \to \frac{1}{2} \qquad \lim_{x \to 0} \frac{f(x)}{h(x)} \to 0 \qquad \lim_{x \to 0} \frac{f(x)}{w(x)} \to \text{undefined}$$

Вывод: все данные функции бесконечно малые. Причем $f(x)=O(p(x))$, то есть функции $f(x)$ и $p(x)$ одного порядка малости; $f(x)=o(h(x))$, то есть $f(x)$ более высокого порядка малости, чем $h(x)$; функции $f(x)$ и $w(x)$ несравнимы.





Задание 1.4. Найдите точки разрыва заданных функций и определите их тип.

<u>Порядок выполнения задания</u>

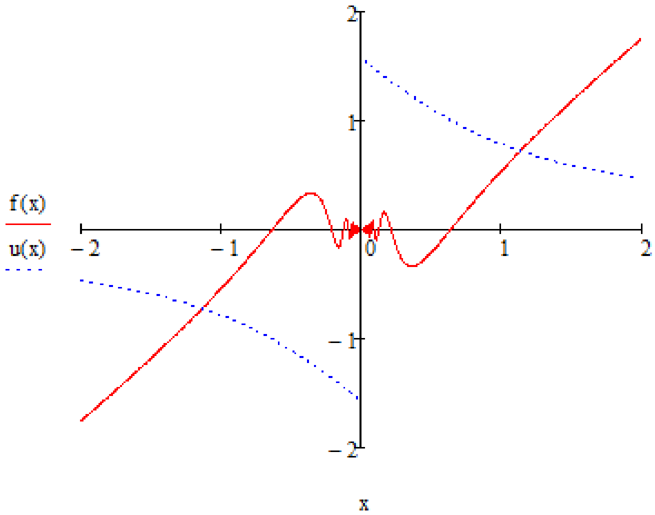
1. Установите автоматический режим вычислений и режим отображения результатов символьных вычислений по горизонтали.
2. Определите выражение для функции.
3. Вычислите предел функции в точке разрыва.
4. Вычислите односторонние пределы функции в точке разрыва.
5. Постройте график функции в окрестности точки разрыва.
6. Сформулируйте вывод.
7. Выполните вычисления пп. 2-6 для всех функций задания.

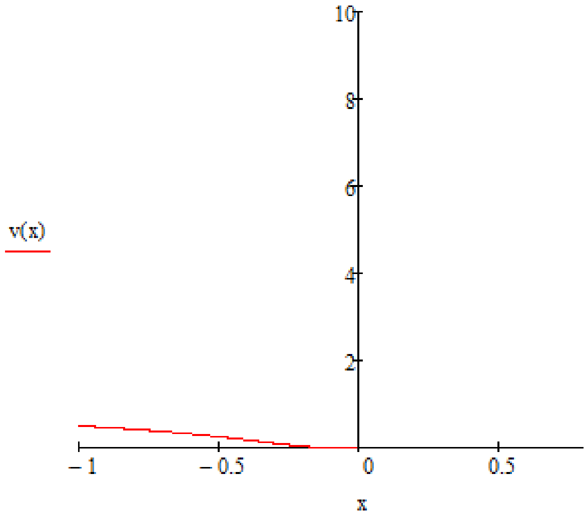
<u>Пример</u>. Вариант исследования разрывов функций $f(x)=x\cos(1/x)$, $u(x)=\mathrm{arctg}(1/x)$, $v(x)=2^{1/x}$ приведен выше. Все три функции имеют разрыв в точке $x=0$. Функция $f(x)=x\cos(1/x)$ имеет в нуле устранимый разрыв. Положив $f(0)=0$, получим непрерывную функцию. Функция $u(x)=\mathrm{arctg}(1/x)$ имеет в нуле скачок. Функция $v(x)=2^{1/x}$ имеет в нуле разрыв второго рода, поскольку бесконечен предел в нуле справа.

$f(x) := x\cos\left(\frac{1}{x}\right)$ $\quad \lim_{x \to 0^+} f(x) \to 0$ $\quad \lim_{x \to 0^-} f(x) \to 0$ $\quad \lim_{x \to 0} f(x) \to 0$

$u(x) := \operatorname{atan}\left(\frac{1}{x}\right)$ $\quad \lim_{x \to 0^+} u(x) \to \frac{\pi}{2}$ $\quad \lim_{x \to 0^-} u(x) \to -\frac{\pi}{2}$ $\quad \lim_{x \to 0} u(x) \to \text{undefined}$

$v(x) := 2^{\frac{1}{x}}$ $\quad \lim_{x \to 0^+} v(x) \to \infty$ $\quad \lim_{x \to 0^-} v(x) \to 0$ $\quad \lim_{x \to 0} v(x) \to \text{undefined}$





Задание 1.5. Найдите (аналитически и графически) точки, в которых достигаются наибольшее и наименьшее значения заданной на отрезке непрерывной функции. Найдите нуль функции на заданном отрезке. Решите уравнение $f(x)=0$.

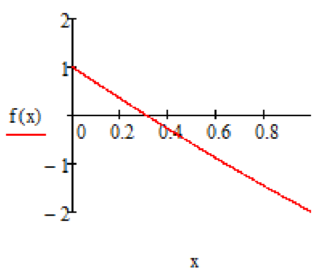
Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов символьных вычислений по горизонтали.
2. Определите выражение для функции.
3. Постройте график функции.
4. Найдите наибольшее и наименьшее значения функции на отрезке.
5. Решите уравнение $f(x)=0$, используя функцию root, выбрав в качестве нулевого приближения сначала левый, а потом правый конец заданного отрезка.

Пример. Найдите наибольшее и наименьшее значения функции $f(x)=2^x-4x$ на отрезке [0,1]; решите на этом отрезке уравнение $2^x-4x=0$.

Ниже приведен фрагмент рабочего документа, содержащий соответствующие графики и вычисления.

$f(x) := 2^x - 4x$

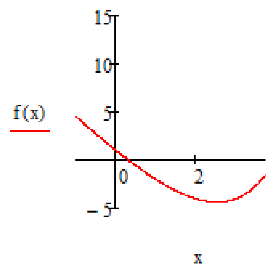


$m := f(1) \qquad M := f(0) \qquad m = -2 \qquad M = 1$

$x := 0 \qquad \operatorname{root}(f(x), x) = 0.31$

$x := 1 \qquad \operatorname{root}(f(x), x) = 0.309907$

$f(0.3099069) = 0.0000001$



Задание 1.6. Найдите по определению производную функции $f(x)$. Вычислите значение производной в указанной точке.

Порядок выполнения задания

1. Установите режим автоматических вычислений и режим отображения результатов вычислений по горизонтали.

2. Определите функцию.
3. Определите приращение функции в указанной точке.
4. Вычислите предел отношения приращения функции к приращению аргумента при стремлении приращения аргумента к нулю.
5. Вычислите производную аналитически.

Пример. Найдите в точке x=0 производную функции $f(x)=\begin{cases}\dfrac{\cos x-\cos 3x}{x}, & x\neq 0;\\ 0, & x=0.\end{cases}$

Поскольку $f(x)$=0, то приращение функции $\Delta f(x)$ в точке x=0 равно $\Delta f(x)=f(0+\Delta x)-f(0)=f(\Delta x)-f(0)=f(\Delta x)$.

$$f(x) := \frac{\cos(x)-\cos(3x)}{x} \qquad \lim_{\Delta x\to 0}\frac{f(\Delta x)}{\Delta x} \to 4$$

Производная функции $f(x)$ в точке x=0 существует и равна 4, $f'(x)$=4.

Задание 1.7. Вычислите производную функции $f(x)$ по определению. Найдите значение производной функции в указанных точках. Вычислите по определению односторонние производные функции $g(x)=|f(x)|$ в указанных точках. Постройте графики обеих функций и объясните различие поведения функций в окрестности указанной точки.

Порядок выполнения задания

1. Установите режим автоматических вычислений и режим отображения результатов вычислений по горизонтали.
2. Найдите производную от функции $f(x)$ как предел отношения приращения функции к приращению аргумента.
3. Вычислите значения производной в указанных точках.
4. Попытайтесь так же найти производную функции $g(x)$.
5. Вычислите по определению производную и односторонние производные функции $g(x)$ в указанной точке. Сравните полученные значения.
6. Постройте графики функций $f(x)$ и $g(x)$.

Пример. Вычислите производную функции $f(x)=x^2-4$ по определению. Найдите значение производной функции в точках $x_0=\pm 2$. Вычислите в указанных точках по определению односторонние производные функции $g(x)=|x^2-4|$. Постройте графики обеих функций и объясните различие поведения функций в окрестности указанной точки. Фрагмент рабочего документа с соответствующими вычислениями приведен ниже.

$$\lim_{\Delta x\to 0}\frac{(x+\Delta x)^2-4-\left(x^2-4\right)}{\Delta x} \to 2\cdot x \qquad \lim_{\Delta x\to 0}\frac{\left|(x+\Delta x)^2-4\right|-\left|x^2-4\right|}{\Delta x} \to \text{undefined}$$

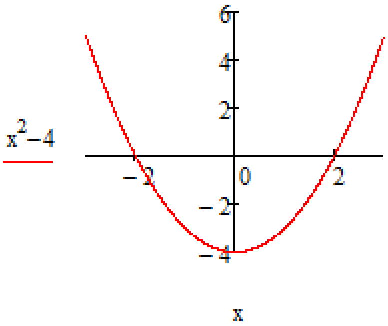
Производная функции $f(x)$ равна $2x$, $f'(1)$=2, $f'(2)$=4, производная функции $g(x)$ в произвольной точке не существует.

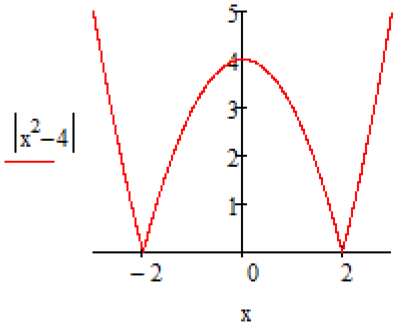
$$\lim_{\Delta x\to 0}\frac{\left|(1+\Delta x)^2-4\right|-\left|1^2-4\right|}{\Delta x} \to -2 \qquad \lim_{\Delta x\to 0^+}\frac{\left|(1+\Delta x)^2-4\right|-\left|1^2-4\right|}{\Delta x} \to -2 \qquad \lim_{\Delta x\to 0^-}\frac{\left|(1+\Delta x)^2-4\right|-\left|1^2-4\right|}{\Delta x}$$

В точке x=1 производная равна минус 2, правая и левая производная также минус 2.

$$\lim_{\Delta x\to 0}\frac{\left|(2+\Delta x)^2-4\right|-\left|2^2-4\right|}{\Delta x} \to \text{undefined} \qquad \lim_{\Delta x\to 0^+}\frac{\left|(2+\Delta x)^2-4\right|-\left|2^2-4\right|}{\Delta x} \to 4 \qquad \lim_{\Delta x\to 0^-}\frac{\left|(2+\Delta x)^2-4\right|-\left|2^2-4\right|}{\Delta x}$$

В точке x=2 производная не существует, правая производная имеет значение 4, левая – минус 4.





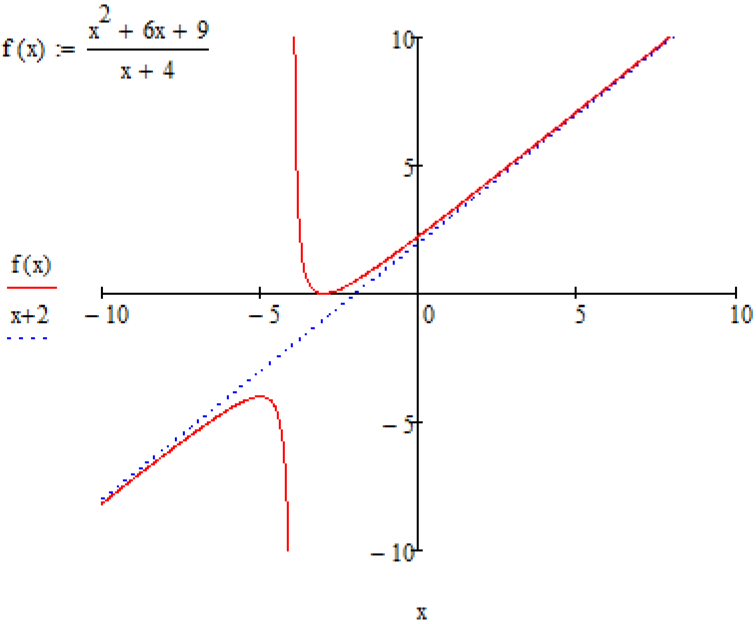
Задание 1.8. Изобразите график заданной функции и подтвердите построение аналитическим исследованием: найдите координаты точек пересечения с координатными осями, найдите и постройте наклонные асимптоты. Запишите уравнения вертикальных асимптот.

<u>Порядок выполнения задания</u>

1. Установите режим автоматических вычислений и режим отображения результатов вычислений по горизонтали.
2. Определите функцию $f(x)$ и постройте ее график.
3. Найдите точку пересечения с осью ординат, вычислив $f(0)$. Сравните с данными на графике.
4. Найдите точку пересечения с осью абсцисс, решив уравнение $f(x)=0$. Сравните с данными на графике.
5. Найдите точки разрыва функции, вычислите соответствующие односторонние пределы, запишите уравнения вертикальных асимптот. Сравните с данными на графике.
6. Вычислите пределы $k=\lim\limits_{x\to\pm\infty}\dfrac{f(x)}{x}$ и $b=\lim\limits_{x\to\pm\infty}\left(f(x)-kx\right)$; изобразите прямую $y=kx+b$.
7. Покажите, что равен нулю предел на бесконечности расстояния d от точки на кривой до наклонной асимптоты ($d\le|f(x)-(kx+b|)$).

<u>Пример</u>. Постройте график и проверьте построение аналитическим исследованием для функции $f(x)=\dfrac{x^2+6x+9}{x+4}$.

$$f(x) := \frac{x^2+6x+9}{x+4}$$



$$k := \lim_{x\to\infty}\frac{f(x)}{x} \to 1$$

$$b := \lim_{x\to\infty}(f(x)-k\,x) \to 2$$

$$\lim_{x\to -4^+} f(x) \to \infty \qquad \lim_{x\to -4^-} f(x) \to -\infty$$

$$f(0) = 2.25$$

$$\frac{x^2+6x+9}{x+4}\ \text{factor} \to \frac{(x+3)^2}{x+4}$$

Задание 1.9. Изобразите график заданной функции и подтвердите построение аналитическим исследованием: постройте график производной, найдите нули производной. Найдите координаты точек экстремума.

<u>Порядок выполнения задания</u>

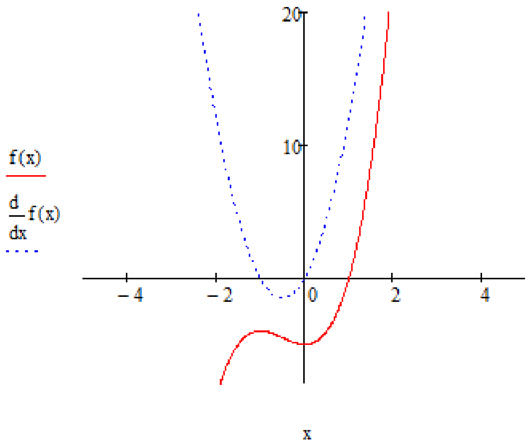
1. Установите режим автоматических вычислений и режим отображения результатов вычислений по горизонтали.
2. Определите функцию $f(x)$ и постройте ее график.
3. Постройте график производной.
4. Найдите нули производной, решив уравнение $f(x)=0$.
5. Вычислите и запишите координаты точек экстремума, Укажите их тип (максимум/минимум).

Пример. Постройте график и подтвердите построение аналитическим исследованием для функции $f(x)=2x^3+3x^2-5$.

$f(x) := 2x^3 + 3x^2 - 5 \qquad f1(x) := \frac{d}{dx}f(x) \rightarrow 6 \cdot x^2 + 6 \cdot x$

$2x^3 + 3x^2 - 5 \qquad 6 \cdot x^2 + 6 \cdot x \qquad \begin{pmatrix} 0 \\ -1 \end{pmatrix} \qquad f1(-1) = 0 \qquad f(-1) = -4$

$f1(0) = 0 \qquad f(0) = -5$



Точка максимума (-1, -4), минимума (0,-5).

Задание 1.10. Изобразите график заданной функции и подтвердите построение аналитическим исследованием: постройте график второй производной, найдите нули второй производной. Найдите координаты точек перегиба.

Порядок выполнения задания
1. Установите режим автоматических вычислений.
2. Определите функцию $f(x)$ и постройте ее график.
3. Постройте график второй производной.
4. Найдите нули второй производной, решив уравнение $f''(x)=0$.
5. Вычислите и запишите координаты точек перегиба.
6. Опишите интервалы выпуклости и вогнутости функции.

Пример. Постройте график и подтвердите построение аналитическим исследованием для функции $f(x) = -\left(\frac{x}{x+2}\right)^2$

$$f(x) := -\left(\frac{x}{x+2}\right)^2$$

$$-\left(\frac{x}{x+2}\right)^2 \qquad \mathbf{f(x)}$$

$$-2\cdot\frac{x}{(x+2)^2} + 2\cdot\frac{x^2}{(x+2)^3} \qquad \mathbf{f'(x)}$$

$$\frac{-2}{(x+2)^2} + 8\cdot\frac{x}{(x+2)^3} - 6\cdot\frac{x^2}{(x+2)^4} \qquad \mathbf{f''(x)}$$

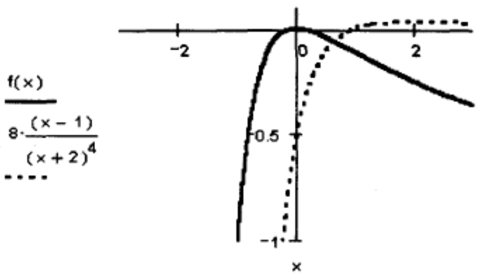
$$8\cdot\frac{(x-1)}{(x+2)^4}$$ **Упрощенное выражение f''(x)**

Упрощенное выражение f'(x) **f'(x)=0** **Точка максимума**

$$x\cdot\frac{(x+2)}{\left(x^2\cdot(x+3)\right)^{\left(\frac{2}{3}\right)}} \qquad -2 \qquad (-2,\ 1.587)$$

$$f(-2) = 1.587$$

$f(1) = -0.111$ **Точка перегиба (1, -0.111)**



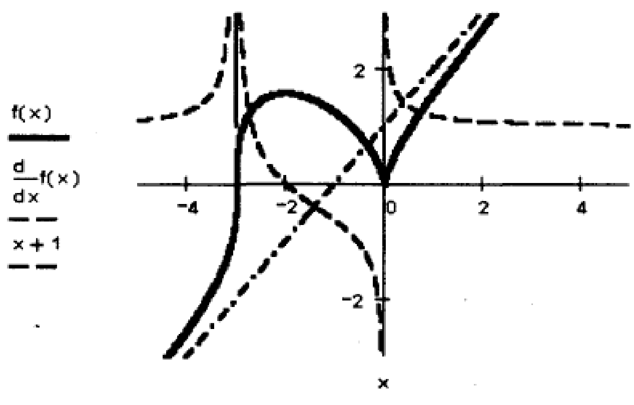
Задание 1.11. Постройте график функции и подтвердите изображение аналитическим исследованием.

Порядок выполнения задания

1. Установите автоматический режим вычислений.
2. Определите функцию и постройте ее график.
3. Найдите наклонные асимптоты и изобразите их.
4. Постройте график первой производной и найдите точки экстремума.
5. Постройте график второй производной и найдите точки перегиба.

Пример. Постройте график функции $f(x) = \sqrt[3]{x^2(x+3)}$ и подтвердите изображение аналитическим исследованием.

$$f(x) := \sqrt[3]{x^2\cdot(x+3)}$$



$k := \lim\limits_{x \to \infty} \frac{f(x)}{x} \to 1 \qquad b := \lim\limits_{x \to \infty} (f(x) - x) \to 1$

$k = 1 \quad b = 1$ **Наклонная асимптота y=x+1**

f(x)	**f'(x)**
$\sqrt[3]{x^2\cdot(x+3)}$	$\frac{1}{\left[3\cdot(x^2\cdot(x+3))^{\left(\frac{2}{3}\right)}\right]}\cdot(2\cdot x\cdot(x+3)+x^2)$

Упрощенное выражение f'(x)	**f'(x)=0**	**Точка максимума**
$x\cdot\frac{(x+2)}{(x^2\cdot(x+3))^{\left(\frac{2}{3}\right)}}$	-2 $f(-2) = 1.587$	(-2, 1.587)

f'(x)	**Упрощенное выражение f''(x)**
$x\cdot\frac{(x+2)}{(x^2\cdot(x+3))^{\left(\frac{2}{3}\right)}}$	$\frac{-2}{\left[(x+3)\cdot(x^2\cdot(x+3))^{\left(\frac{2}{3}\right)}\right]}$

Знаменатель f''(x)	**f''(x)=0 не существует**	**Точка перегиба**
$(x+3)\cdot(x^2\cdot(x+3))^{\left(\frac{2}{3}\right)}$	$\begin{bmatrix} 0 \\ -3 \end{bmatrix}$ $f(-3) = 0$	(-3, 0)

Задание 1.12. Изобразите линии, заданные в декартовых координатах явно уравнением $y=f(x)$ и неявно уравнением $F(x, y)=0$. Запишите уравнения касательной и нормали к каждой кривой в указанных точках и изобразите их на графике.

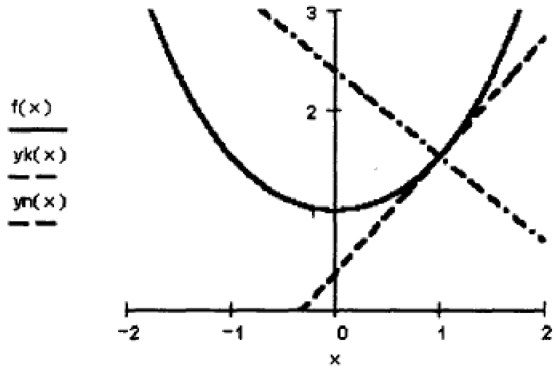
Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов вычислений по горизонтали.
2. Определите функцию $f(x)$.
3. Запишите уравнение касательной, проходящей через заданную точку, в виде, разрешенном относительно переменной y.
4. Запишите уравнение нормали, проходящей через заданную точку, в виде, разрешенном относительно переменной y.
5. Постройте график функции и изобразите на нем касательную и нормаль.
6. Введите левую часть уравнения, задающего линию в неявной форме, и разрешите его относительно переменной y.
7. Определите функции, задающие явно части кривой, определенной в условии неявно.
8. Проверьте, лежат ли заданные точки на кривой и не являются ли они особыми.
9. Запишите уравнения касательной и нормали в каждой из точек в виде, разрешенном относительно переменной y.
10. Постройте графики функций, определенных в п. 7. Изобразите на тех же графиках касательные и нормали к ним.

Пример. Построите цепную линию, уравнение которой $y=\mathrm{ch}\, x$, и касательную и нормаль к ней в точке (1, ch 1). Постройте эллипс $\frac{x^2}{2^2}+\frac{y^2}{3^2}=1$ и касательные и нормали к нему в точках $\left(1, \pm\frac{3\sqrt{3}}{2}\right)$.

$$f(x) := \frac{e^x + e^{-x}}{2} \qquad f1(x) := \frac{d}{dx}f(x)$$

$$k := f1(1) \quad yk(x) := k\cdot(x-1) + f(1) \quad yn(x) := \frac{-1}{k}\cdot(x-1) + f(1)$$



$$F(x,y) = \frac{x^2}{4} + \frac{y^2}{9} - 1 \qquad \begin{bmatrix} \frac{3}{2}\cdot\sqrt{-x^2+4} \\ \frac{-3}{2}\cdot\sqrt{-x^2+4} \end{bmatrix} \qquad y1(x) := \frac{3}{2}\cdot\sqrt{-x^2+4}$$

$$\frac{x^2}{4} + \frac{y^2}{9} - 1 \qquad y2(x) := \frac{-3}{2}\cdot\sqrt{-x^2+4}$$

Проверка принадлежности точек

$\left(1, \frac{3\cdot\sqrt{3}}{2}\right)$, $\left(1, -\frac{3\cdot\sqrt{3}}{2}\right)$ кривой

$$\frac{1}{4} + \frac{(y1(1))^2}{9} - 1 = 0 \qquad y1(1) = 2.598 \qquad y2(1) = -2.598 \qquad \frac{3\cdot\sqrt{3}}{2} = 2.598$$

Вычисление частных производных

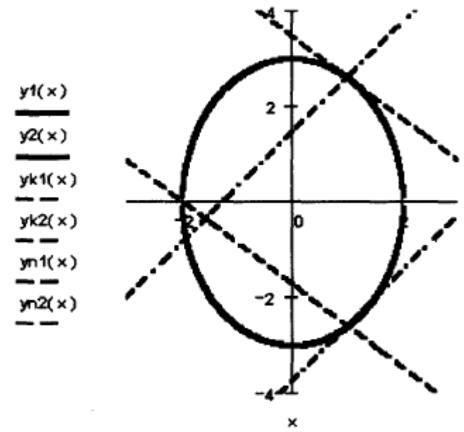
$$\frac{x^2}{4} + \frac{y^2}{9} - 1 \qquad \frac{1}{2}\cdot x \qquad Fx(x,y) := \frac{1}{2}\cdot x \qquad kx := Fx\left(1, \frac{3\cdot\sqrt{3}}{2}\right)$$

$$\frac{x^2}{4} + \frac{y^2}{9} - 1 \qquad \frac{2}{9}\cdot y \qquad Fy(x,y) := \frac{2}{9}\cdot y \qquad ky := Fy\left(1, \frac{3\cdot\sqrt{3}}{2}\right)$$

Уравнения касательных и нормалей

$$yk1(x) := y1(1) - \frac{kx}{ky}\cdot(x-1) \qquad yk2(x) := y2(1) - \frac{kx}{ky}\cdot(x-1)$$

$$yn1(x) := y1(1) + \frac{ky}{kx}\cdot(x-1) \qquad yn2(x) := y2(1) + \frac{ky}{kx}\cdot(x-1)$$



Задание 1.13. Изобразите на плоскости кривую, заданную параметрически, касательную и нормаль к ней в указанной точке.

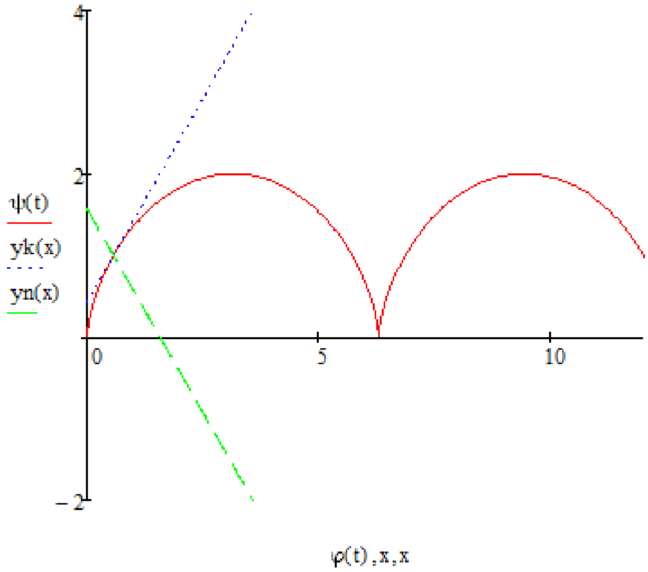
Порядок выполнения задания

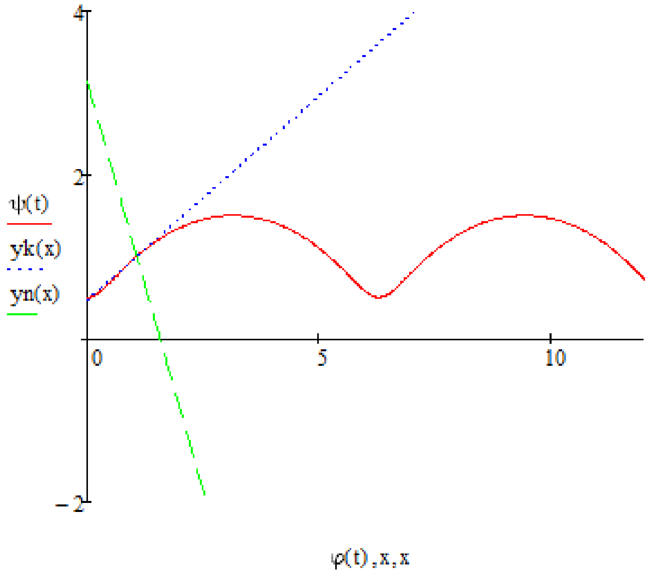
1. Установите автоматический режим вычислений и режим отображения результатов по горизонтали.
2. Определите функции, задающие кривую параметрически.
3. Найдите аналитически (символьно) производные обеих функций.
4. Запишите уравнения касательной и нормали в точках в виде функций переменной *x*.
5. Постройте график функции и изобразите на нем касательную и нормаль.

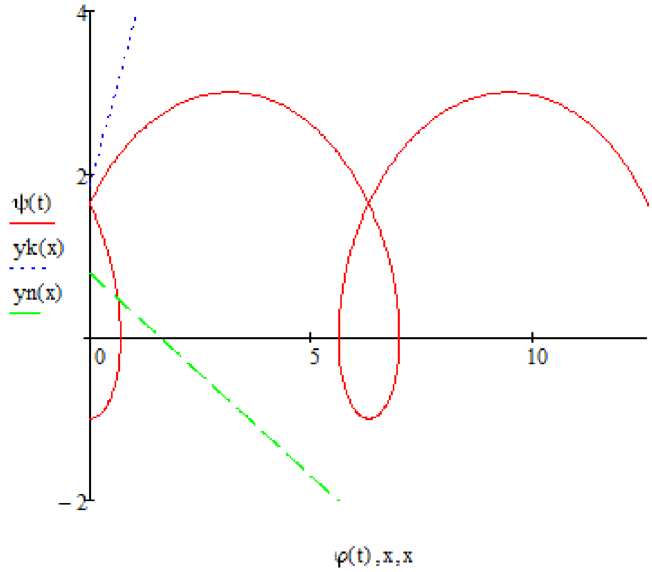
Пример. Для значений λ, равных 1, 0.5, 2, постройте циклоиды $x = t - \lambda \sin t$, $x = 1 - \lambda \cos t$, $t \in (0, 4\pi)$, а также касательные и нормали к ним в точке $\left(\frac{\pi}{2} - 1, \frac{1}{\lambda}\right)$.

$\lambda := 1 \quad \varphi(t) := t - \lambda \sin(t) \quad d\varphi(t) := \frac{d}{dt}\varphi(t) \quad \psi(t) := 1 - \lambda \cos(t) \quad d\psi(t) := \frac{d}{dt}\psi(t)$

$$yk(x) := \frac{d\psi\left(\frac{\pi}{2}\right)}{d\varphi\left(\frac{\pi}{2}\right)}\left(x - \varphi\left(\frac{\pi}{2}\right)\right) + \psi\left(\frac{\pi}{2}\right) \qquad yn(x) := \frac{-d\varphi\left(\frac{\pi}{2}\right)}{d\psi\left(\frac{\pi}{2}\right)}\left(x - \varphi\left(\frac{\pi}{2}\right)\right) + \psi\left(\frac{\pi}{2}\right)$$







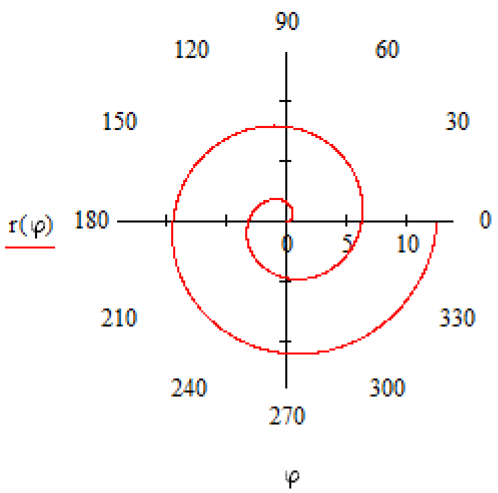
Задание 1.14. Изобразите кривую, заданную в полярных координатах. Запишите уравнения касательной и нормали к ней.

Порядок выполнения задания

1. Установите режим автоматических вычислений и режим отображения результатов вычислений по горизонтали.
2. Определите функцию $r = r(\varphi)$.
3. Изобразите в полярных координатах линию для значений φ из указанного в задании промежутка.
4. Найдите аналитически (символьно) производную $r1 = r'(\varphi)$.
5. Найдите аналитически выражение для углового коэффициента касательной.
6. Запишите в декартовых координатах уравнения касательной и нормали к кривой в указанной точке.
7. Выполните аналогичные построения для других значений параметров, изменив в рабочем документе соответствующие константы.

Пример. Изобразите архимедовы спирали $r = a\varphi$ для a=1, 0.5, 2. Запишите уравнения касательных и нормалей к ним в точках $\varphi = \frac{\pi}{4}, \frac{\pi}{2}, \pi$. Фрагмент рабочего документа с соответствующими вычислениями и графиками приведен для a=1, $\varphi = \frac{\pi}{4}$ ниже.

$a := 1$ $\qquad r(\varphi) := a\,\varphi$



$r1(\varphi) := \frac{d}{d\varphi} r(\varphi) \qquad tg(\varphi) := \frac{r(\varphi)}{r1(\varphi)}$

$\varphi 0 := \frac{\pi}{4} \qquad r0 := r(\varphi 0) \qquad k := tg(\varphi 0)$

$x0 := r0\cos(\varphi 0) \qquad y0 := r0\sin(\varphi 0)$

$yk(x) := k(x - x0) + y0$

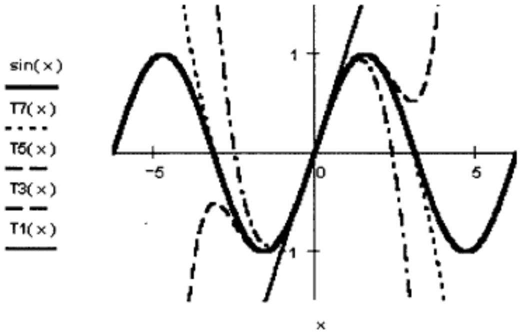
$yn(x) := \frac{-1}{k}(x - x0) + y0$

Задание 1.15. Запишите формулу Тейлора заданной функции. Исследуйте аналитически и графически зависимость погрешности приближенной формулы Тейлора от степени многочлена и от расстояния $|x-x_0|$. Исследуйте поведение остаточного члена формулы Тейлора. Выполните задание для функции $f(x)$ из задания 1.12.

<u>*Порядок выполнения задания*</u>

1. Установите автоматический режим вычислений и режим отображения результатов вычисления по горизонтали.
2. Определите функцию $f(x)$.
3. Запишите разложение функции по формуле Тейлора до указанного порядка k.
4. Определите многочлены Тейлора степени n=1, 2, ... , k.
5. Постройте график функции и многочленов Тейлора степени n=1, 2, ... , k.
6. Изобразите для этих степеней графики остаточных членов.

<u>*Пример*</u>. Исследуйте формулу Тейлора для функции $\sin x$ в окрестности нуля, вычисляя многочлены Тейлора до 7-й степени и соответствующие остаточные члены. Фрагмент рабочего документа с соответствующими вычислениями и графиками приведен ниже.



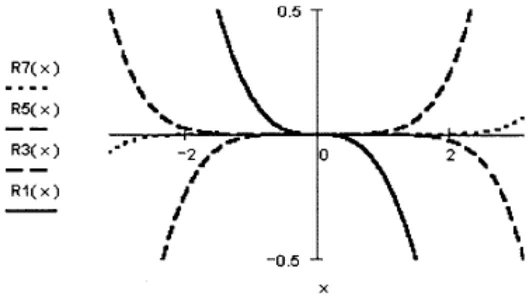
$$\sin(x) \qquad x - \frac{1}{6}\cdot x^3 + \frac{1}{120}\cdot x^5 - \frac{1}{5040}\cdot x^7 + O(x^9)$$

$$T7(x) := x - \frac{1}{6}\cdot x^3 + \frac{1}{120}\cdot x^5 - \frac{1}{5040}\cdot x^7 \qquad R7(x) := \sin(x) - T7(x)$$

$$T5(x) := x - \frac{1}{6}\cdot x^3 + \frac{1}{120}\cdot x^5 \qquad R5(x) := \sin(x) - T5(x)$$

$$T3(x) := x - \frac{1}{6}\cdot x^3 \qquad R3(x) := \sin(x) - T3(x)$$

$$T1(x) := x \qquad R1(x) := \sin(x) - T1(x)$$



Задание 1.16. Запишите для заданной функции формулу Тейлора указанного порядка в окрестности точки x_0. Изобразите график функции и ее многочлена Тейлора.

<u>*Порядок выполнения задания*</u>

1. Установите автоматический режим вычислений и режим отображения результатов вычисления по горизонтали.
2. Определите функцию $f(x)$.
3. Выполните замену переменной $t = x - x_0$ и запишите полученную функцию переменной t.
4. Разложите функцию переменной t по формуле Тейлора в окрестности нуля.
5. Определите функцию переменной t, равную правой части формулы Тейлора, и выполните в ней замену $x=t+x_0$.
6. Постройте графики функции $f(x)$ и ее многочлена Тейлора.

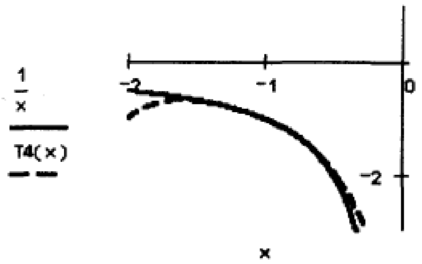
<u>*Пример*</u>. Запишите формулу Тейлора четвертого порядка для функции $f(x) = \frac{1}{x}\sin x$ в окрестности точки x_0=-1. Постройте график функции и ее многочлена Тейлора в окрестности этой точки.

$t = x + 1 \qquad x = t - 1 \qquad \frac{1}{x} \qquad \frac{1}{(t-1)}$

$\frac{1}{(t-1)} \qquad -1 - t - t^2 - t^3 - t^4 + O(t^5)$

$-2 - x - (x+1)^2 - (x+1)^3 - (x+1)^4 + O((x+1)^5)$

$T4(x) := -2 - x - (x+1)^2 - (x+1)^3 - (x+1)^4$



Задание 1.18. Вычислите неопределенный интеграл $\int f(x)dx$ и проверьте правильность вычислений, постройте графики семейства первообразных.

Порядок выполнения задания

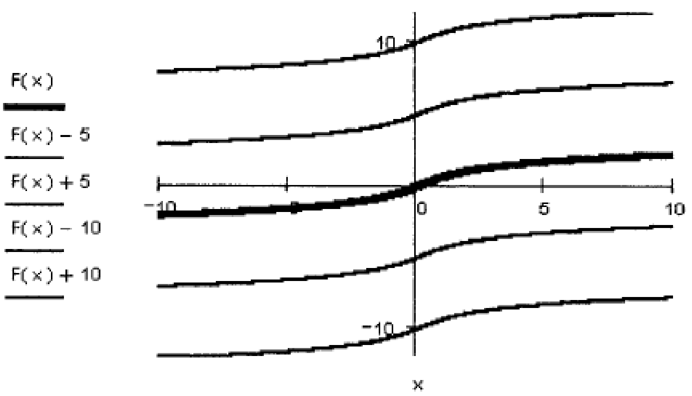
1. Установите автоматический режим вычислений и режим отображения результатов вычислений по горизонтали.
2. Определите подынтегральную функцию как функцию переменной *x*.
3. Найдите первообразную, используя символьную математику пакета.
4. Определите первообразную как функцию переменной.
5. Найдите производную первообразной, используя символьную математику пакета.
6. Упростите производную от первообразной, сравните результате подынтегральной функцией.
7. Постройте на одном графике изображения нескольких первообразных.

Пример

$f(x) := \frac{1}{\sqrt{2 \cdot x^2 - x + 2}} \qquad \int f(x)\,dx \rightarrow \frac{1}{2} \cdot \sqrt{2} \cdot \operatorname{asinh}\left[\frac{4}{15} \cdot \sqrt{15} \cdot \left(x - \frac{1}{4}\right)\right]$

$F(x) := \frac{1}{2} \cdot \sqrt{2} \cdot \operatorname{asinh}\left[\frac{4}{15} \cdot \sqrt{15} \cdot \left(x - \frac{1}{4}\right)\right]$

$\frac{d}{dx}F(x) \rightarrow \frac{2}{15} \cdot \sqrt{2} \cdot \frac{\sqrt{15}}{\sqrt{1 + \frac{16}{15} \cdot \left(x - \frac{1}{4}\right)^2}} \qquad \frac{1}{\sqrt{2 + 2 \cdot x^2 - x}}$



Задание 1.19. Для заданной функции *f*(*x*) исследуйте поведение интегральных сумм на отрезке интегрирования [*a*, *b*], разбивая его на равные части. Вычислите определенный интеграл и сравните его значение со значениями пределов интегральных сумм.

<u>*Порядок выполнения задания*</u>

1. Установите автоматический режим вычислений и режим отображения результатов по горизонтали.
2. Определите подынтегральную функцию как функцию переменной *x* и постройте ее график.
3. Вычислите определенный интеграл.
4. Запишите выражение для интегральной суммы, полученной при разбиении отрезка интегрирования на равные части, когда значение функции вычисляется в левом конце отрезка разбиения. Найдите ее предел при числе отрезков разбиения, стремящемся к бесконечности.
5. Запишите выражение для интегральной суммы, полученной при разбиении отрезка интегрирования на равные части, когда значение функции вычисляется в правом конце отрезка разбиения. Найдите ее предел при числе отрезков разбиения, стремящемся к бесконечности.
6. Запишите выражение для интегральной суммы, полученной при разбиении отрезка интегрирования на равные части, когда значение функции вычисляется в середине отрезка разбиения. Найдите ее предел при числе отрезков разбиения, стремящемся к бесконечности.
7. Сравните полученные значения пределов между собой и со значением интеграла.
8. Постройте графики интегральных сумм как функций числа разбиений отрезка интегрирования.
9. Постройте графики интегральных сумм как функций длины отрезка разбиений.

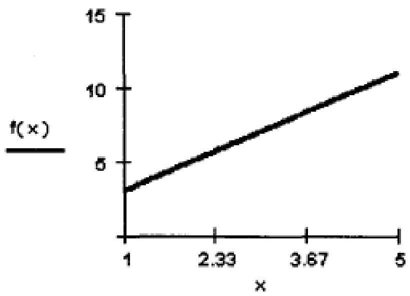
<u>*Пример*</u>

$f(x) := 2x + 1 \qquad a := 1 \qquad b := 5$

$$\int_1^5 f(x)\,dx \to 28$$

$$S := \frac{f(1) + f(5)}{2} \cdot (5 - 1)$$

$S = 28$



Значения подынтегральной функции вычисляются в левых концах отрезков

$$Sl(N) := \sum_{i=0}^{N-1} f\left(a + i \cdot \frac{b-a}{N}\right) \cdot \frac{b-a}{N} \qquad \lim_{N \to \infty} Sl(N) \to 28$$

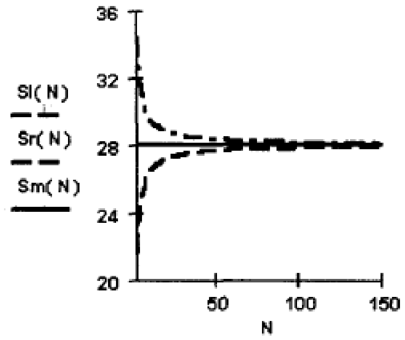
Значения подынтегральной функции вычисляются в правых концах отрезков

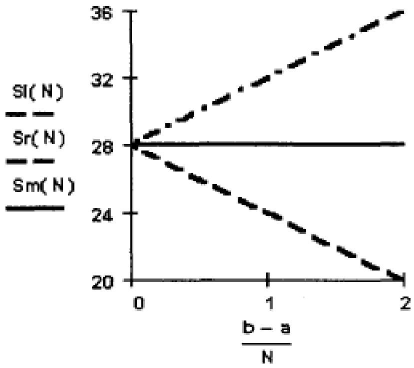
$$Sr(N) := \sum_{i=0}^{N-1} f\left[a + (i+1) \cdot \frac{b-a}{N}\right] \cdot \frac{b-a}{N} \qquad \lim_{N \to \infty} Sr(N) \to 28$$

Значения подынтегральной функции вычисляются в серединах отрезков

$$Sm(N) := \sum_{i=0}^{N-1} f\left[a+\left(i+\frac{1}{2}\right)\cdot\frac{b-a}{N}\right]\cdot\frac{b-a}{N} \qquad \lim_{N\to\infty} Sm(N) \to 28$$

$N := 2 .. 150$





Задание 1.20. Вычислите определенный интеграл $\int_a^b f(x)dx$ непосредственно и с помощью замены переменных.

Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов вычислений по горизонтали.
2. Определите подынтегральную функцию как функцию от x.
3. Найдите определенный интеграл символьно.
4. Введите выражение для новой переменной t как функции от x.
5. Вычислите новые пределы интегрирования по t.
6. Выразите переменную x через t, решив уравнение $\varphi(x)$-t=0.
7. Найдите символьно производную x по t.
8. Скопируйте в буфер обмена выражение для x через t и выполните символьно замену переменной в подынтегральной функции.
9. Упростите подынтегральную функцию.
10. Запишите и упростите новую подынтегральную функцию, которая получается умножением выражения, полученного в предыдущем пункте, на производную x по t.
11. Вычислите определенный интеграл, интегрируя полученную функцию по t по отрезку, найденному в п. 4.

Пример

$$f(x) := \frac{1}{\cos(x)\cdot(1+\cos(x))}$$

Символьное вычисление интеграла

$$\int_0^{\frac{\pi}{4}} f(x)\,dx \to \lim_{x \to \frac{\pi}{4}^-} -\tan\left(\frac{1}{2}\cdot x\right) - \ln\left(\tan\left(\frac{1}{2}\cdot x\right)-1\right) + \ln\left(\tan\left(\frac{1}{2}\cdot x\right)+1\right) + i\cdot\pi$$

Замена переменной t=tg(x) $\quad t(x) := \tan(x)$

Вычисление новых пределов интегрирования

$$t(0) = 0 \qquad t\left(\frac{\pi}{4}\right) = 1$$

Выражение x=x(t) и символьное вычисление производной от x по t

$$\tan(x) - t \qquad \operatorname{atan}(t) \qquad \frac{1}{(1+t^2)}$$

Замена переменной в подынтегральной функции и упрощение полученного выражения

$$\frac{1}{\cos(x)\cdot(1+\cos(x))} \qquad \frac{\sqrt{1+t^2}}{\left(1+\frac{1}{\sqrt{1+t^2}}\right)} \qquad \frac{(1+t^2)}{\left(\sqrt{1+t^2}+1\right)}$$

Вычисление и упрощение подынтегрального выражения

$$\frac{(1+t^2)}{\left(\sqrt{1+t^2}+1\right)}\cdot\frac{1}{(1+t^2)} \qquad \frac{1}{\left(\sqrt{1+t^2}+1\right)}$$

Символьное вычисление определенного интеграла

$$\int_0^1 \frac{1}{\left(\sqrt{1+t^2}+1\right)}\,dt \to 1-\sqrt{2}+\operatorname{asinh}(1)$$

Вычисление числового значения с пятью десятичными знаками

$$1-\sqrt{2}+\operatorname{asinh}(1) \qquad .46717$$

Задание 1.21. Исследуйте функцию, заданную интегралом с переменным верхним пределом $\int_a^x f(t)dt$.

Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов вычислений по горизонтали.
2. Введите выражение интеграла с переменным верхним пределом и вычислите его символьно.
3. Определите функцию, заданную введенным в предыдущем пункте интегралом.
4. Найдите первую и вторую производные определенной в предыдущем пункте функции.
5. Найдите горизонтальные асимптоты, если они существуют, вычислив пределы функции на $\pm\infty$.
6. Постройте график функции, заданной интегралом, и изобразите на нем горизонтальные асимптоты.

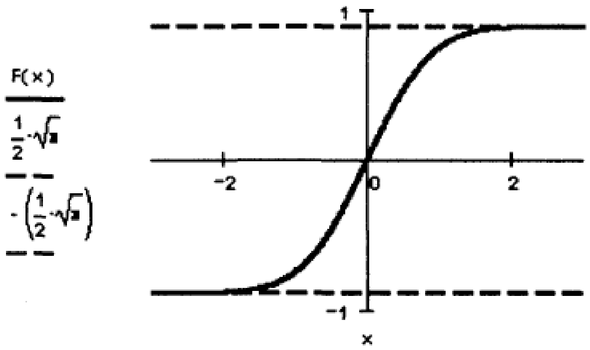
Пример

$$\int_0^x \exp(-t^2)\,dt \to \frac{1}{2}\cdot\sqrt{\pi}\cdot\operatorname{erf}(x)$$

$$F(x) := \int_0^x \exp(-t^2)\,dt \qquad F(0) = 0 \qquad F(1) = 0.747 \qquad F(2) = 0.882$$

$$\frac{d}{dx}F(x) \to \exp(-x^2) \qquad \frac{d^2}{dx^2}F(x) \to -2\cdot x\cdot\exp(-x^2)$$

$$\lim_{x\to\infty} F(x) \to \frac{1}{2}\cdot\sqrt{\pi} \qquad \lim_{x\to-\infty} F(x) \to \frac{-1}{2}\cdot\sqrt{\pi}$$



Задание 1.22. Вычислите несобственный интеграл по неограниченному промежутку непосредственно и через предел. Постройте график допредельной функции, заданной интегралом с переменным пределом.

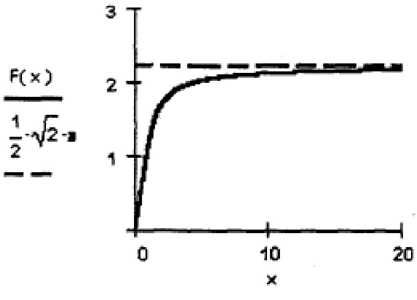
Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов вычислений по горизонтали.
2. Введите выражение подынтегральной функции.
3. Вычислите несобственный интеграл от заданной функции по заданному неограниченному промежутку непосредственно используя операцию вычисления определенного интеграла.
4. Определите функцию, заданную интегралом с переменным пределом.
5. Вычислите соответствующий предел функции, определенной в предыдущем пункте.
6. Сравните полученные значения несобственного интеграла.
7. Постройте график допредельной функции, определенной в п.3, и его горизонтальные асимптоты, вычисленные в п. 5.

Пример

$$f(x) := \frac{x^2+1}{x^4+1} \qquad F(x) := \int_0^x f(t)\,dt$$

$$\int_0^\infty f(t)\,dt \to \frac{1}{2}\cdot\sqrt{2}\cdot\pi \qquad \lim_{x\to\infty} F(x) \to \frac{1}{2}\cdot\sqrt{2}\cdot\pi$$



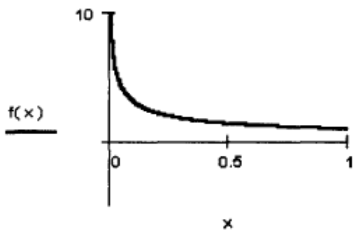
Задание 1.23. Вычислите несобственный интеграл от неограниченной функции по заданному отрезку непосредственно и через предел. Постройте график подынтегральной функции и график функции, заданной интегралом с переменным пределом.

Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов вычислений по горизонтали.
2. Введите выражение подынтегральной функции.
3. Постройте график подынтегральной функции, чтобы убедиться в том, что она является бесконечно большой на одном из концов промежутка интегрирования.
4. Вычислите несобственный интеграл от заданной функции по заданному неограниченному промежутку непосредственно, используя операцию вычисления определенного интеграла.
5. Определите функцию, заданную интегралом с переменным пределом.
6. Вычислите предел функции, определенной в п. 5.
7. Сравните полученные значения несобственного интеграла.
8. Сформируйте таблицу значений переменного предела и таблицу соответствующих значений функции, заданной интегралом.
9. Постройте график табличной функции, вычисленной в предыдущем пункте.
10. Сравните значение функции в нуле на графике с вычисленным значением несобственного интеграла.

Пример

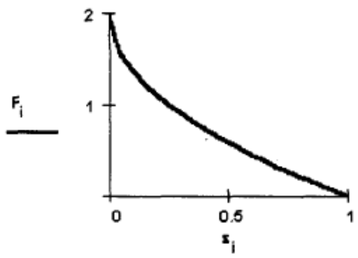
$$f(x) := \frac{1}{\sqrt{x}} \qquad \int_0^1 f(x)\,dx \to 2$$



$$F(\varepsilon) := \int_\varepsilon^1 f(x)\,dx$$

$$\lim_{\varepsilon \to 0} F(\varepsilon) \to 2$$

$$i := 0..20 \qquad \varepsilon_i := 1 - \frac{i}{20} \qquad F_i := F(\varepsilon_i)$$



Задание 1.24. Разложите заданную функцию в ряд Тейлора по степеням и исследуйте поведение частичных сумм ряда Тейлора в окрестности нуля. Исследуйте поведение остатка ряда.

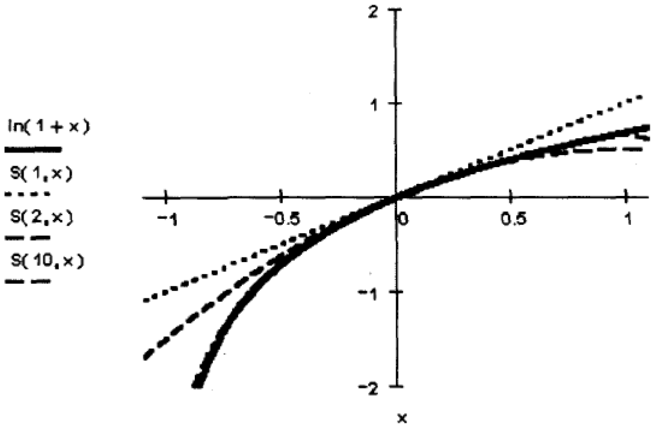
Порядок выполнения задания

1. Установите автоматический режим вычислений.
2. Определите функцию $f(x)$.
3. Используйте разложение функции по формуле Тейлора для записи ряда Тейлора функции.
4. Вычислите символьно сумму полученного ряда.
5. Запишите выражение частичной суммы ряда как функцию числа слагаемых и переменной x.
6. Постройте график функции и графики нескольких частичных сумм ряда.
7. Постройте графики соответствующих остатков ряда.

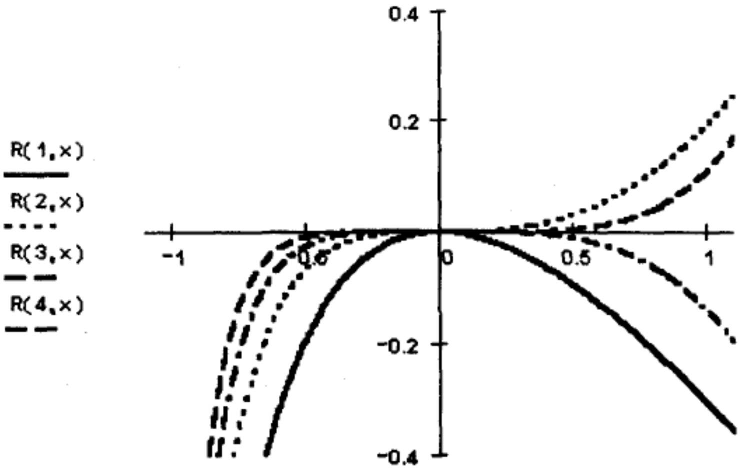
Пример

$$f(x) := \ln(1+x) \qquad c(n) := \frac{(-1)^{n-1}}{n}$$

$$S(n,x) := \sum_{k=1}^{n} c(k)\cdot x^{k} \qquad \sum_{n=1}^{\infty} c(n)\cdot x^{n} \rightarrow \ln(1+x)$$



$$R(n,x) := f(x) - S(n,x)$$



Задание 1.25. Запишите для заданной функции разложение в ряд Тейлора в окрестности точки x_0. Изобразите график функции и графики нескольких частичных сумм ряда Тейлора.

Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов по горизонтали.
2. Определите функцию $f(x)$.
3. Выполните замену $t=x-x_0$ и запишите полученную функцию переменной t.
4. Используйте разложение функции переменной t по формуле Тейлора в окрестности нуля, чтобы записать для нее ряд Тейлора.
5. Определите частичную сумму ряда Тейлора как функцию числа слагаемых и переменной x.
6. Постройте графики функции $f(x)$ и частичных сумм ее ряда Тейлора.

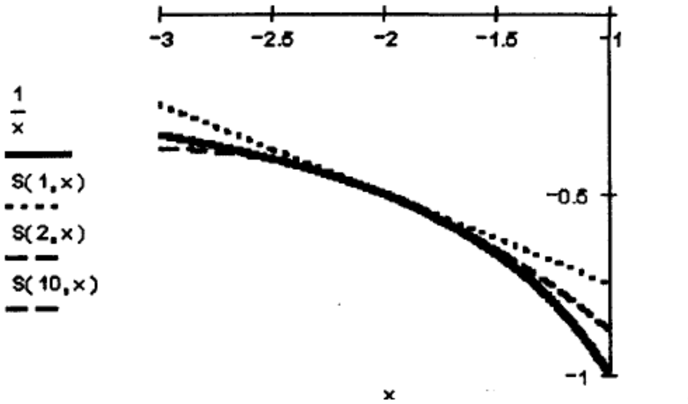
Пример Запишите ряд Тейлора функции $1/x$ в окрестности точки x_0=-2. Постройте график функции и частичных сумм ее ряда Тейлора в окрестности этой точки. Фрагмент рабочего документа с соответствующими вычислениями и графиками приведен ниже.

$t = x + 2 \qquad x = t - 2 \qquad \frac{1}{x} \qquad \frac{1}{(t-2)}$

$$\frac{1}{(t-2)} \qquad \frac{-1}{2} - \frac{1}{4}\cdot t - \frac{1}{8}\cdot t^2 - \frac{1}{16}\cdot t^3 - \frac{1}{32}\cdot t^4 - \frac{1}{64}\cdot t^5 + O(t^6)$$

$$c(n) := -\frac{1}{2^{n+1}} \qquad S(n,x) := \sum_{k=0}^{n} c(k)\cdot(x+2)^k$$

$$\sum_{n=0}^{\infty} -\frac{1}{2^{n+1}}\cdot(x+2)^n \to \frac{1}{x}$$



Задание 1.26. Изобразите график и линии уровня функции $z=f(x, y)$ в указанной прямоугольной области. Опишите поведение функции (укажите приближенно координаты локальных экстремумов и седловых точек, если они есть) в заданной области. Рассмотрите функцию $z = x^k y^m \exp\left(-\frac{x^2}{A} - \frac{y^2}{B}\right)$ в квадрате $-a<x<a$, $-a<y<a$.

Порядок выполнения задания

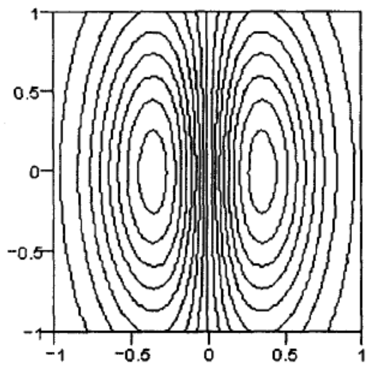
1. Установите автоматический режим вычислений.
2. Введите определение функции.
3. Определите в плоскости переменных (x, y) прямоугольную сетку $x_i=a+ih$, $i=1, 2, \ldots, N$, $y_j=a+ih$, $j=1, 2, \ldots, N$, в области $a<x<b$, $c<y<d$.
4. Сформируйте матрицу z_{ij} значений функции $z=f(x, y)$ на сетке: $z_{ij}=f(x_i, y_j)$.
5. Постройте линии уровня функции.
6. Проанализируйте картину линий уровня, найдите точки экстремума и седловые точки, если они есть, и укажите приближенно их координаты.

Пример

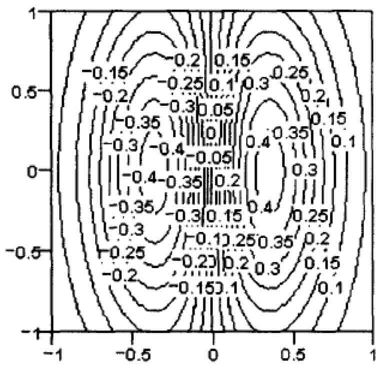
$$f(x,y) := x \cdot \exp\left(-x^2 - \frac{y^2}{4}\right)$$

$N := 40 \qquad i := 0..N \qquad j := 0..N$

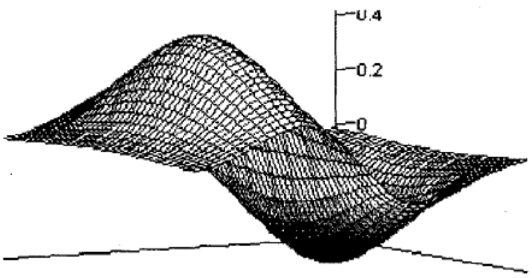
$x_i := -2 + 0.1 \cdot i \qquad y_j := -2 + 0.1 \cdot j \qquad z_{i,j} := f(x_i, y_j)$



z



z



z

Задание 1.27. Найдите частные производные и градиент функции и $u=f(x, y, z)$. Вычислите в заданной точке градиент функции и производную по направлению из этой точки в начало координат. Найдите $\frac{\partial^2 f}{\partial x \partial y}$, $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial x \partial z}$, $\frac{\partial^2 f}{\partial y^2}$, $\frac{\partial^2 f}{\partial z^2}$.

<u>*Порядок выполнения задания*</u>

1. Установите автоматический режим вычислений и режим отображения результатов по горизонтали.
2. Введите определение функции.
3. Используйте меню символьной математики для вычисления частных производных по всем переменным.
4. Определите частные производные как функции переменных x, y, z.
5. Определите вектор grad f как вектор, координаты которого – определенные в предыдущем пункте функции.
6. Вычислите grad f и | grad f | в указанной точке.
7. Определите вектор направления.
8. Вычислите производную по направлению в указанной точке.
9. Найдите производные $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial y^2}$, $\frac{\partial^2 f}{\partial z^2}$.
10. Найдите смешанные производные.

<u>*Пример*</u>

Найдите частные производные и градиент функции $u=xyze^{x+y+z}$. Вычислите в точке $M(0, 1, -1)$ производную по направлению $l=\left(\frac{1}{2},\frac{\sqrt{3}}{2},0\right)$ и градиент функции в этой точке. Найдите $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial x \partial y}$. Ниже приведен фрагмент рабочего документа с соответствующими вычислениями.

$$f(x, y, z) := x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$\frac{d}{dx} x \cdot y \cdot z \cdot \exp(x + y + z) \rightarrow y \cdot z \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$\frac{d}{dy} x \cdot y \cdot z \cdot \exp(x + y + z) \rightarrow x \cdot z \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$\frac{d}{dz} x \cdot y \cdot z \cdot \exp(x + y + z) \rightarrow x \cdot y \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$dfx(x, y, z) := \frac{d}{dx} f(x, y, z)$$

$$dfx(x, y, z) := y \cdot z \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$dfx(x, y, z) := y \cdot z \cdot \exp(x + y + z) \cdot (1 + x)$$

$$dfy(x, y, z) := \frac{d}{dy} f(x, y, z)$$

$$dfy(x, y, z) := x \cdot z \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$dfy(x, y, z) := z \cdot x \cdot \exp(x + y + z) \cdot (1 + y)$$

$$dfz(x, y, z) := \frac{d}{dz} f(x, y, z)$$

$$dfz(x, y, z) := x \cdot y \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$dfz(x, y, z) := x \cdot y \cdot \exp(x + y + z) \cdot (1 + z)$$

$$gradf(x, y, z) := \begin{bmatrix} dfx(x, y, z) \\ dfy(x, y, z) \\ dfz(x, y, z) \end{bmatrix} \qquad l := \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ 0 \end{bmatrix}$$

Вторая производная по x

$$\frac{d^2}{dx^2} f(x, y, z) \rightarrow 2 \cdot y \cdot z \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$2 \cdot y \cdot z \cdot \exp(x + y + z) + x \cdot y \cdot z \cdot \exp(x + y + z)$$

$$y \cdot z \cdot \exp(x + y + z) \cdot (2 + x)$$

$$\frac{d}{dx} \frac{d}{dy} f(x, y, z) \rightarrow z \cdot \exp(x + y + z) \cdot (1 + x) \cdot (1 + y)$$

Задание 1.28. Запишите для заданной функции $z=f(x, y)$ многочлены Тейлора нулевого, первого и второго порядка в окрестности указанной точки. Вычислите и сравните значения многочленов и функции в указанных точках. Приведите вычисленные значения в виде таблицы. Исследуйте зависимость погрешности аппроксимации функции многочленом Тейлора от расстояния между точками и от степени многочлена. Выполните вычисления для функции из задания 1.26. Выполните вычисления в окрестности любой точки экстремума функции.

Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов по горизонтали.
2. Введите определение функции.
3. Найдите частные производные первого порядка.
4. Найдите все частные производные второго порядка.
5. Вычислите значения производных в заданной точке.
6. Запишите выражения для многочленов Тейлора нулевой, первой и второй степени как функций переменных (x, y).
7. Вычислите расстояния от заданных точек до центра окрестности, значения функции и многочленов Тейлора в заданных точках; присвойте вычисленные значения элементам матрицы и выведите их в виде таблицы (матрицы).

Пример. Выполните задание для функции $f(x, y)=\sin(x+\sin y)$ в окрестности точки $M(1, 2)$. Вычислите значения функции и ее многочленов Тейлора T_0, T_1, T_2 в точках $M_0(1.4, 2.8)$, $M_1(1.2, 2.4)$, $M_2(0.99, 1.98)$. Фрагмент рабочего документа с соответствующими вычислениями приведен ниже.

$$f(x,y) := \sin(x+\sin(y))$$

$$\frac{d}{dx}\sin(x+\sin(y)) \qquad \cos(x+\sin(y))$$

$$fx(x,y) := \cos(x+\sin(y)) \qquad f(1,2) = 0.94325$$

$$\frac{d}{dy}\sin(x+\sin(y)) \qquad \cos(x+\sin(y))\cdot\cos(y)$$

$$fy(x,y) := \cos(x+\sin(y))\cdot\cos(y)$$

$$\frac{d}{dx}\cos(x+\sin(y)) \qquad -\sin(x+\sin(y))$$

$$fxx(x,y) := -\sin(x+\sin(y))$$

$$\frac{d}{dy}\cos(x+\sin(y))\cdot\cos(y)$$

$$-\sin(x+\sin(y))\cdot\cos(y)^2 - \cos(x+\sin(y))\cdot\sin(y)$$

$$fyy(x,y) := -\sin(x+\sin(y))\cdot\cos(y)^2 - \cos(x+\sin(y))\cdot\sin(y)$$

$$\frac{d}{dy}\cos(x+\sin(y)) \qquad -\sin(x+\sin(y))\cdot\cos(y)$$

$$fxy(x,y) := -\sin(x+\sin(y))\cdot\cos(y)$$

$$a := fx(1,2) \qquad b := fy(1,2)$$

$$a = -0.33207 \qquad b = 0.13819$$

$$A := fxx(1,2) \qquad B := fxy(1,2) \qquad C := fyy(1,2)$$

$$A = -0.94325 \qquad B = 0.39253 \qquad C = 0.1386$$

$$T0(x,y) := f(1,2)$$

$$T1(x,y) := T0(x,y) + a\cdot(x-1) + b\cdot(y-2)$$

$$T2(x,y) := T1(x,y) + \frac{1}{2!}\cdot\left(A\cdot(x-1)^2 + 2\cdot B\cdot(x-1)\cdot(y-2) + C\cdot(y-2)^2\right)$$

$r(x,y) := \sqrt{(x-1)^2 + (y-2)^2}$

ORIGIN := 1 $\quad$ $i := 1..3$

$x_1 := 1.4 \quad x_2 := 1.2 \quad x_3 := 0.99$

$y_1 := 2.8 \quad y_2 := 2.4 \quad y_3 := 1.98$

$F_{1,i} := r(x_i, y_i) \quad F_{2,i} := T0(x_i, y_i) \quad F_{3,i} := T1(x_i, y_i)$

$F_{4,i} := T2(x_i, y_i) \quad F_{5,i} := f(x_i, y_i)$

$R_{1,i} := r(x_i, y_i) \quad R_{2,i} := F_{5,i} - T0(x_i, y_i) \quad R_{3,i} := F_{5,i} - T1(x_i, y_i)$

$R_{4,i} := F_{5,i} - T2(x_i, y_i) \quad R_{5,i} := f(x_i, y_i)$

Задание 1.29. Запишите для заданной функции $z=f(x, y)$ формулу Тейлора третьего порядка в окрестности указанной точки. Найдите стационарные точки функции. Исследуйте каждую из них на экстремум. Изобразите график и линии уровня функции в окрестности стационарных точек. Выполните вычисления для функции из задания 1.26. Выполните вычисления в окрестности любой точки экстремума функции.

Порядок выполнения задания

1. Установите автоматический режим вычислений и режим отображения результатов по горизонтали.
2. Введите определение функции.
3. Найдите частные производные функции.
4. Постройте линии уровня заданной функции.
5. Найдите стационарные точки.
6. Вычислите $A = \dfrac{\partial^2 f(x_0, y_0)}{\partial x^2}$, $B = \dfrac{\partial^2 f(x_0, y_0)}{\partial x \partial y}$, $C = \dfrac{\partial^2 f(x_0, y_0)}{\partial y^2}$ в каждой стационарной точке (x_0, y_0) и исследуйте ее на экстремум.
7. Изобразите поверхность $z=A(x-x_0)^2+2B(x-x_0)(y-y_0)+C(y-y_0)^2$ и график функции $z=f(x, y)$ в окрестности каждой стационарной точки (x_0, y_0).
8. Изобразите линии уровня функции $z=A(x-x_0)^2+2B(x-x_0)(y-y_0)+C(y-y_0)^2$ и график функции $z=f(x, y)$ в окрестности каждой стационарной точки (x_0, y_0).

Пример. Запишите формулу Тейлора четвертого порядка для функции $z=\sin xy \exp(-x^2-y^2)$ в окрестности точки (1, 0). Найдите стационарные точки функции. Исследуйте каждую из них на экстремум. Изобразите график и линии уровня функции в окрестности стационарных точек. Фрагмент рабочего документа с соответствующими вычислениями и графиками приведен ниже.

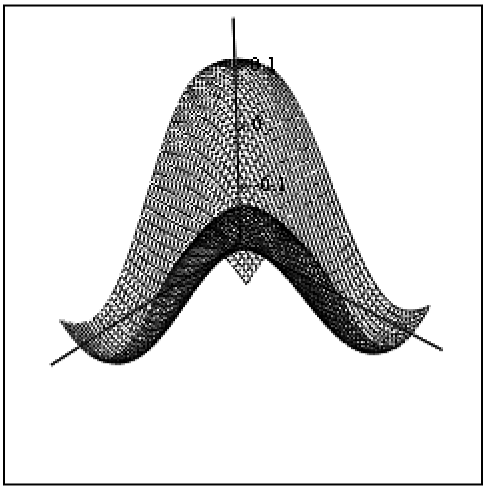
$$f(x,y) := \sin(x\,y)\,e^{-\left(x^2+y^2\right)}$$

$$f(x,y)\ \text{series}, x = 1, y = 0, 3\ \rightarrow y\cdot e^{-1} - \frac{7\cdot y^3\cdot e^{-1}}{6} - y\cdot e^{-1}\cdot(x-1) - y\cdot e^{-1}\cdot(x-1)^2$$

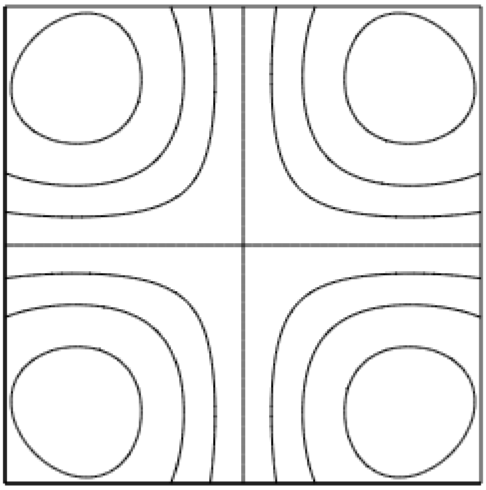
$$dfx(x,y) := \frac{d}{dx}f(x,y)\ \text{simplify}\ \rightarrow e^{-x^2-y^2}\cdot(y\cdot\cos(x\cdot y) - 2\cdot x\cdot\sin(x\cdot y))$$

$$dfy(x,y) := \frac{d}{dy}f(x,y)\ \text{simplify}\ \rightarrow e^{-x^2-y^2}\cdot(x\cdot\cos(x\cdot y) - 2\cdot y\cdot\sin(x\cdot y))$$

$N := 50 \qquad i := 0..N \qquad j := 0..N \qquad x_i := -1 + \frac{1}{25}i \qquad y_j := -1 + \frac{1}{25}j \qquad z_{i,j} := f\left(x_i, y_j\right)$



z



z

Given

$$\exp(-x^2 - y^2)\cdot(\cos(x\cdot y)\cdot y - 2\cdot x\cdot\sin(x\cdot y)) = 0$$

$$-\exp(-x^2 - y^2)\cdot(-\cos(x\cdot y)\cdot x + 2\cdot\sin(x\cdot y)\cdot y) = 0$$

$$\begin{bmatrix} x0 \\ y0 \end{bmatrix} := \text{Find}(x,y) \qquad x0 = -0.681 \qquad y0 = 0.681$$

$$A := \frac{d^2}{dx^2}f(x,y) \qquad B := \frac{d}{dx}\frac{d}{dy}f(x,y) \qquad C := \frac{d^2}{dy^2}f(x,y)$$

$$A = 0.775 \qquad B = 0.113 \qquad C = 0.775$$

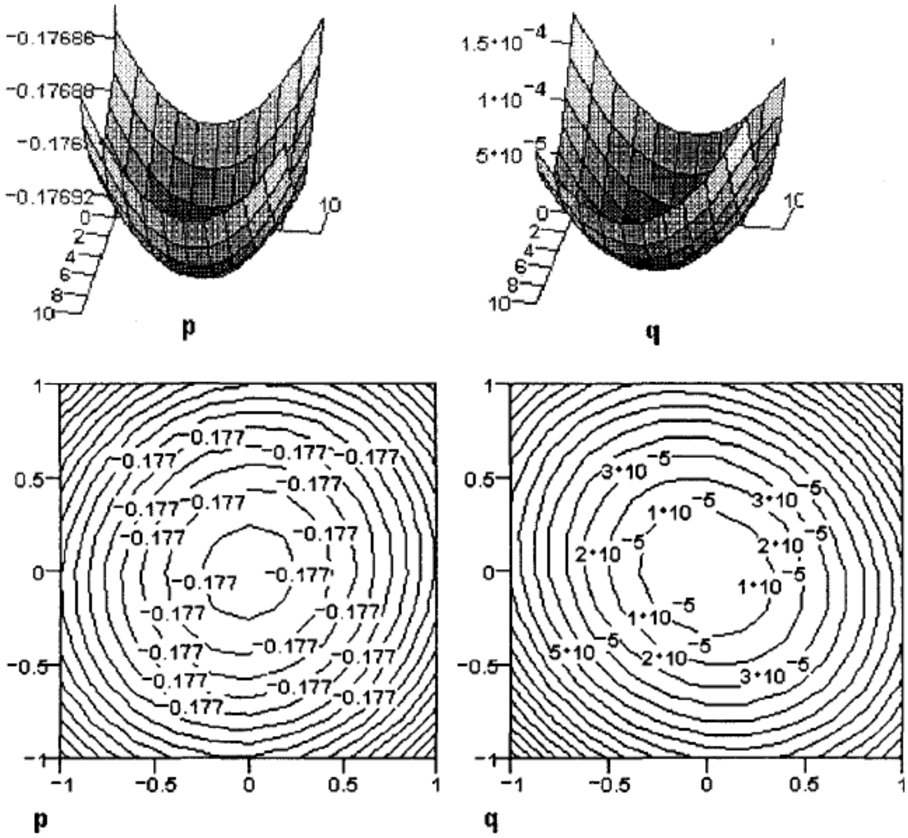
$$A\cdot C - B^2 = 0.588$$

$$g(x,y) := A\cdot(x - x0)^2 + (2\cdot B\cdot(x - x0)\cdot(y - y0)) + C\cdot(y - y0)^2$$

$i := 0 .. 10 \qquad j := 0 .. 10$

$x_i := x0 - 0.01 + 0.002 \cdot i \qquad y_j := y0 - 0.01 + 0.002 \cdot j$

$p_{i,j} := f(x_i, y_j) \qquad q_{i,j} := g(x_i, y_j)$



Методические указания для выполнения практических заданий Лабораторной работы 2 «Матричные операции»

Задание 2.1. Умножая на матрицы специального вида, сформируйте матрицу-столбец и матрицу-строку, соответственно равные столбцу и строке матрицы *A* с заданными в условии номерами. Вычислите суммы элементов столбца и строки, номера которых указаны в задании. Переставьте указанные в задании строки и столбцы матрицы.

<u>*Порядок выполнения задания*</u>

1. Установите режим автоматического выполнения вычислений.
2. Определите и введите матрицу *A*.
3. Введите матрицу, умножение на которую выделяет столбец и строку матрицы с указанным номером. Выполните умножение.
4. Введите матрицу, умножение на которую суммирует элементы указанных столбца и строки. Выполните умножение.
5. Введите матрицу, умножение на которую переставляет указанные столбцы и строки. Выполните умножение.

<u>*Пример*</u>

$$D := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Суммирование элементов по столбцам

$Crow := (1 \ \ 1 \ \ 1)$

$Crow \cdot D = [\,12 \ \ 15 \ \ 18\,]$

Выделение одного столбца

$$C3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad D \cdot C3 = \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix}$$

Суммирование элементов по строкам

$$Ccol := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad D \cdot Ccol = \begin{bmatrix} 6 \\ 15 \\ 24 \end{bmatrix}$$

Выделение одной строки

$C1 := (1 \ \ 0 \ \ 0) \quad C1 \cdot D = [\,1 \ \ 2 \ \ 3\,]$

$C2 := (0 \ \ 1 \ \ 0) \quad C2 \cdot D = [\,4 \ \ 5 \ \ 6\,]$

Умножение на единичную матрицу

$$E := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E \cdot D = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad D \cdot E = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Умножение на число, умножение на скалярную матрицу

$$2 \cdot D = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix} \qquad E2 := \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \qquad E2 \cdot D = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$$

Перестановка двух строк

$$C12 := \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad C12 \cdot D = \begin{bmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \\ 7 & 8 & 9 \end{bmatrix} \quad C23 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad C23 \cdot D = \begin{bmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \end{bmatrix}$$

Перестановка двух столбцов

$$D \cdot C12 = \begin{bmatrix} 2 & 1 & 3 \\ 5 & 4 & 6 \\ 8 & 7 & 9 \end{bmatrix} \qquad D \cdot C23 = \begin{bmatrix} 1 & 3 & 2 \\ 4 & 6 & 5 \\ 7 & 9 & 8 \end{bmatrix}$$

Задание 2.2. Докажите, что матрица $H = E - 2\dfrac{vv^T}{|v|^2}$ (v – вектор-столбец) – ортогональная матрица. Проверьте для нее свойства ортогональной матрицы. В качестве v возьмите первый столбец матрицы A из задания 2.1.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите матрицу-столбец V и единичную матрицу E соответствующей размерности.
4. Вычислите матрицу H.
5. Вычислите произведения $H^T H$ и HH^T.
6. Вычислите H^{-1}. Сравните H^{-1} и H^T.
7. Покажите, что векторы-столбцы матрицы H имеют единичную длину и попарно ортогональны. Убедитесь, что выполняется равенство $|H| = 1$.

Пример

$$V := \begin{bmatrix}1\\0\\1\\1\end{bmatrix} \qquad E := \text{identity}(4) \qquad E = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

Вычисление матрицы H $\quad H := E - \dfrac{2}{(|V|)^2}\cdot V\cdot V^T$

$$H = \begin{bmatrix}0.333&0&-0.667&-0.667\\0&1&0&0\\-0.667&0&0.333&-0.667\\-0.667&0&-0.667&0.333\end{bmatrix} \qquad H^T = \begin{bmatrix}0.333&0&-0.667&-0.667\\0&1&0&0\\-0.667&0&0.333&-0.667\\-0.667&0&-0.667&0.333\end{bmatrix}$$

Доказательство ортогональности матрицы H

$$H\cdot H^T = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \qquad H^T\cdot H = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

Вычисление обратной матрицы

$$H^{-1} = \begin{bmatrix}0.333&0&-0.667&-0.667\\0&1&0&0\\-0.667&0&0.333&-0.667\\-0.667&0&-0.667&0.333\end{bmatrix} \qquad H^{-1} - H^T = \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

Проверка свойств ортогональной матрицы

ORIGIN := 1

$H^{\langle 1\rangle}\cdot H^{\langle 1\rangle} = 1 \qquad H^{\langle 1\rangle}\cdot H^{\langle 2\rangle} = 0 \qquad H^{\langle 2\rangle}\cdot H^{\langle 3\rangle} = 0$

$H^{\langle 2\rangle}\cdot H^{\langle 2\rangle} = 1 \qquad H^{\langle 1\rangle}\cdot H^{\langle 3\rangle} = 0 \qquad H^{\langle 2\rangle}\cdot H^{\langle 4\rangle} = 0$

$H^{\langle 3\rangle}\cdot H^{\langle 3\rangle} = 1 \qquad H^{\langle 1\rangle}\cdot H^{\langle 4\rangle} = 0 \qquad H^{\langle 3\rangle}\cdot H^{\langle 4\rangle} = 0$

$H^{\langle 4\rangle}\cdot H^{\langle 4\rangle} = 1 \qquad |H| = -1$

Задание 2.3. Докажите, что матрица P идемпотентна и вычислите ее определитель. Покажите, что матрица I=2P-E инволютивна. Вычислите ее определитель и обратную матрицу.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Введите матрицу P.
3. Вычислите P^2 и P^2-P.
4. Вычислите det P и P^{-1}.
5. Введите единичную матрицу E той же размерности, что и матрица P.
6. Вычислите матрицу I=2P-E.
7. Вычислите матрицу I^2.
8. Вычислите det I и I^{-1}.

Пример

$P := \begin{pmatrix} -26 & -18 & -27 \\ 21 & 15 & 21 \\ 12 & 8 & 13 \end{pmatrix}$ $\quad P^2 = \begin{pmatrix} -26 & -18 & -27 \\ 21 & 15 & 21 \\ 12 & 8 & 13 \end{pmatrix}$ $\quad P^2 - P = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ матрица идемпотентна

$\left|\begin{pmatrix} -26 & -18 & -27 \\ 21 & 15 & 21 \\ 12 & 8 & 13 \end{pmatrix}\right| = 0$ обратная матрица не существует $P^{-1} =$

$E := \mathrm{identity}(3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\quad I := 2P - E = \begin{pmatrix} -53 & -36 & -54 \\ 42 & 29 & 42 \\ 24 & 16 & 25 \end{pmatrix}$ $\quad I^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ матрица I инволютивна

$\left|\begin{pmatrix} -53 & -36 & -54 \\ 42 & 29 & 42 \\ 24 & 16 & 25 \end{pmatrix}\right| = -1$ $\quad I^{-1} = \begin{pmatrix} -53 & -36 & -54 \\ 42 & 29 & 42 \\ 24 & 16 & 25 \end{pmatrix}$

Задание 2.4. Вычислите разложением по указанной строке (столбцу) определитель матрицы $A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$. Выполните вычисления для матрицы H, построенной в задании 2.2.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Присвойте переменной ORIGIN значение, равное 1.
3. Введите матрицу.
4. Запишите в тетради выражение для вычисления определителя матрицы разложением по указанной в задании строке (столбцу).
5. Сформируйте матрицы, полученные вычеркиванием соответствующих строк и столбцов заданной матрицы, и выведите их на экран.
6. Введите выражение для вычисления определителя, вычислите и выведите на экран его значение.
7. Вычислите определитель матрицы, используя функцию Mathcad. Сравните результаты.

Пример

$A := \begin{pmatrix} 1 & -2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ -7 & 5 & 6 & 7 \\ 3 & 10 & 12 & 13 \end{pmatrix}$

M11 := submatrix(A,2,4,2,4) M12 := augment(submatrix(A,2,4,1,1),submatrix(A,2,4,

M13 := augment(submatrix(A,2,4,1,2),submatrix(A,2,4,4,4)) M14 := submatrix(A,2

$M11 = \begin{pmatrix} 3 & 0 & 1 \\ 5 & 6 & 7 \\ 10 & 12 & 13 \end{pmatrix}$ $\quad M12 = \begin{pmatrix} 2 & 0 & 1 \\ -7 & 6 & 7 \\ 3 & 12 & 13 \end{pmatrix}$ $\quad M13 = \begin{pmatrix} 2 & 3 & 1 \\ -7 & 5 & 7 \\ 3 & 10 & 13 \end{pmatrix}$ $\quad M14 = \begin{pmatrix} 2 & 3 & 0 \\ -7 & 5 & 6 \\ 3 & 10 & 12 \end{pmatrix}$

$detA := 1\,|M11| - (-2)\,|M12| + 3\,|M13|$ $\quad detA = 477$ $\quad |A| = 477$

Задание 2.5. Исследуйте и, если решение существует, найдите по формулам Крамера

решение системы $\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n. \end{cases}$

Порядок выполнения задания

1. Установите режим автоматического выполнения вычислений и режим отображения результатов вычислений по горизонтали.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите матрицу системы и столбец правых частей.
4. Вычислите определитель матрицы системы. Система имеет единственное решение, если определитель отличен от нуля.
5. Вычислите определители матриц, полученных заменой соответствующего столбца столбцом правых частей.
6. Найдите решение системы по формулам Крамера.

Пример

ORIGIN := 1

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ -1 & 2 & -3 & 4 \\ 0 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad B := \begin{bmatrix} 30 \\ 10 \\ 3 \\ 10 \end{bmatrix} \qquad \Delta := |A| \qquad \Delta = -4$$

Определитель отличен от нуля, система имеет единственное решение

$$\Delta 1 := \left| \begin{bmatrix} 30 & 2 & 3 & 4 \\ 10 & 2 & -3 & 4 \\ 3 & 1 & -1 & 1 \\ 10 & 1 & 1 & 1 \end{bmatrix} \right| \quad \Delta 1 = -4 \qquad \Delta 2 := \left| \begin{bmatrix} 1 & 30 & 3 & 4 \\ -1 & 10 & -3 & 4 \\ 0 & 3 & -1 & 1 \\ 1 & 10 & 1 & 1 \end{bmatrix} \right| \quad \Delta 2 = -8$$

$$\Delta 3 := \left| \begin{bmatrix} 1 & 2 & 30 & 4 \\ -1 & 2 & 10 & 4 \\ 0 & 1 & 3 & 1 \\ 1 & 1 & 10 & 1 \end{bmatrix} \right| \quad \Delta 3 = -12 \qquad \Delta 4 := \left| \begin{bmatrix} 1 & 2 & 3 & 4 \\ -1 & 2 & -3 & 4 \\ 0 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \right| \quad \Delta 4 = -4$$

$$x1 := \frac{\Delta 1}{\Delta} \qquad x1 = 1 \qquad x2 := \frac{\Delta 2}{\Delta} \qquad x2 = 2$$

$$x3 := \frac{\Delta 3}{\Delta} \qquad x3 = 3 \qquad x4 := \frac{\Delta 4}{\Delta} \qquad x4 = 1$$

Задание 2.6. Решите как матричное уравнение $A\boldsymbol{x}=\boldsymbol{b}$ систему линейных алгебраических уравнений из задания 2.5.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Введите матрицу системы и матрицу-столбец правых частей.
3. Вычислите решение системы по формуле $\boldsymbol{x}=A^{-1}\boldsymbol{b}$.
4. Проверьте правильность решения умножением матрицы системы на вектор-столбец решения.
5. Найдите решение системы с помощью функции lsolve и сравните результаты вычислений.

Пример

$$x + 2\cdot y + 3\cdot z = 7 \qquad x - 3\cdot y + 2\cdot z = 5 \qquad x + y + z = 3$$

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \\ 1 & 1 & 1 \end{bmatrix} \qquad b := \begin{bmatrix} 7 \\ 5 \\ 3 \end{bmatrix}$$

$$x := A^{-1}\cdot b \qquad x = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \qquad A\cdot x - b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$x := \mathrm{lsolve}(A, b) \qquad x = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

Задание 2.7. Найдите методом Гаусса решение системы линейных алгебраических уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$ из задания 2.5.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите матрицу системы и матрицу-столбец правых частей.
4. Сформируйте расширенную матрицу системы.
5. Приведите расширенную матрицу системы к ступенчатому виду.
6. Сформируйте столбец решения системы.
7. Проверьте правильность решения умножением матрицы системы на вектор-столбец решения.

Пример

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 1 & -3 & 2 \\ 1 & 1 & 1 \end{bmatrix} \qquad b := \begin{bmatrix} 7 \\ 5 \\ 3 \end{bmatrix}$$

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

$$Ar := \mathrm{augment}(A, b) \qquad Ar = \begin{bmatrix} 1 & 2 & 3 & 7 \\ 1 & -3 & 2 & 5 \\ 1 & 1 & 1 & 3 \end{bmatrix}$$

$$Ag := \mathrm{rref}(Ar) \qquad Ag = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

$$x := \mathrm{submatrix}(Ag, 1, 3, 4, 4) \qquad x = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \qquad A\cdot x - b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Задание 2.8. Найдите методом простых итераций приближенное решение линейной системы

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \vdots \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n. \end{cases}$$

<u>Порядок выполнения задания</u>

1. Установите режим автоматических вычислений.
2. Преобразуйте исходную систему $C\boldsymbol{x}=\boldsymbol{d}$ к виду $\boldsymbol{x}=\boldsymbol{b}+A\boldsymbol{x}$.
3. Введите матрицы A и $\boldsymbol{b}$.
4. Проверьте достаточное условие сходимости.
5. Определите нулевое (начальное) приближение решения.
6. Задайте количество итераций.
7. Введите формулу вычисления последовательных приближений решения и вычислите их.
8. Выведите на экран матрицу приближенных решений.
9. Вычислите погрешность найденного приближения.

<u>Пример</u>

$$b := \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} \quad A := \begin{bmatrix} 0 & -0.06 & 0.02 \\ -0.03 & 0 & 0.05 \\ -0.01 & -0.02 & 0 \end{bmatrix}$$

norm1(A) = 0.08 norm2(A) = 0.070975

norme(A) = 0.088882 normi(A) = 0.08

$x^{\langle 0 \rangle} := b \quad k := 1..10 \qquad x^{\langle k \rangle} := b + A \cdot x^{\langle k-1 \rangle}$

x =

	6	7	8	9
0	1.907024	1.907024	1.907024	1.907024
1	3.188647	3.188647	3.188647	3.188647
2	4.917157	4.917157	4.917157	4.917157

$$\varepsilon := \frac{\left| x^{\langle 10 \rangle} - x^{\langle 9 \rangle} \right|}{\left| x^{\langle 9 \rangle} \right|} \qquad \varepsilon = 8.662467 \cdot 10^{-15}$$

Задание 2.9. Исследуйте однородную систему линейных алгебраических уравнений $A\boldsymbol{x}=\boldsymbol{0}$.

<u>Порядок выполнения задания</u>

1. Установите режим автоматических вычислений.
2. Введите матрицу системы.
3. Вычислите ранг матрицы системы.
4. Приведите матрицу системы к ступенчатому виду.
5. Определите базисные и свободные переменные.
6. Запишите полученную эквивалентную систему.
7. Используя символьные вычисления, решите полученную систему относительно базовых переменных.
8. Запишите общее решение системы.
9. Найдите фундаментальную систему решений.

<u>Пример</u>

$$A := \begin{bmatrix} 1 & 4 & 2 & 0 & -3 \\ 2 & 9 & 5 & 2 & 1 \\ 1 & 3 & 1 & -2 & -9 \\ 3 & 12 & 6 & 0 & -8 \\ 2 & 10 & 6 & 4 & 7 \end{bmatrix}$$

rank(A) = 3 **Система нетривиально совместна.**

Приведем матрицу системы к ступенчатому виду:

$$Ag := \mathrm{rref}(A) \qquad Ag = \begin{bmatrix} 1 & 0 & -2 & -8 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Свободные переменные x3, x4, базисные переменные x1, x2, x5,

Запишем и решим эквивалентную систему:

$$\text{Given} \quad \begin{aligned} &x1 - 2\cdot x3 - 8\cdot x4 = 0 \\ &x2 + x3 + 2\cdot x4 = 0 \\ &x5 = 0 \end{aligned} \qquad \mathrm{Find}(x1, x2, x5) \rightarrow \begin{bmatrix} 2\cdot x3 + 8\cdot x4 \\ -x3 - 2\cdot x4 \\ x5 \end{bmatrix}$$

Запишем общее решение системы:

$$X(x3, x4) := \begin{bmatrix} 2\cdot x3 + 8\cdot x4 \\ -x3 - 2\cdot x4 \\ x3 \\ x4 \\ 0 \end{bmatrix}$$

Найдем фундаментальную систему решений:

$$X(1,0) = \begin{bmatrix} 2 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad X(0,1) = \begin{bmatrix} 8 \\ -2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Задание 2.10. Исследуйте неоднородную систему линейных алгебраических уравнений $A\boldsymbol{x}=\boldsymbol{b}$ для двух различных правых частей $\boldsymbol{b}=\boldsymbol{b}_1$ и $\boldsymbol{b}=\boldsymbol{b}_2$.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Введите матрицу системы и расширенные матрицы системы для обеих правых частей.
3. Вычислите ранги основной матрицы и ранги расширенных матриц обеих систем.
4. Сформулируйте и запишите в рабочем документе соответствующий вывод.
5. Приведите расширенную матрицу совместной системы к ступенчатому виду.
6. Определите базисные и свободные переменные.
7. Запишите эквивалентную систему и разрешите ее относительно базисных переменных.
8. Запишите общее решение системы.
9. Найдите два различных частных решения системы.
10. Проверьте правильность найденных решений.

Пример

$$A := \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 2 & 3 & 2 & 1 & 1 \\ 3 & 3 & 2 & 0 & 0 \end{bmatrix} \quad b1 := \begin{bmatrix} 1 \\ 4 \\ 2 \\ 7 \\ 10 \end{bmatrix} \quad b2 := \begin{bmatrix} 1 \\ 4 \\ 2 \\ 7 \\ 9 \end{bmatrix}$$

$Ar1 := augment(A, b1)$ $\qquad$ $Ar2 := augment(A, b2)$

$$Ar1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 4 \\ 0 & 1 & 1 & 1 & 1 & 2 \\ 2 & 3 & 2 & 1 & 1 & 7 \\ 3 & 3 & 2 & 0 & 0 & 10 \end{bmatrix} \quad Ar2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 4 \\ 0 & 1 & 1 & 1 & 1 & 2 \\ 2 & 3 & 2 & 1 & 1 & 7 \\ 3 & 3 & 2 & 0 & 0 & 9 \end{bmatrix}$$

$rank(A) = 3 \quad rank(Ar1) = 4 \quad rank(Ar2) = 3$

rg(A)=3, rg(Ar1)=4, rg(A) отличен от rg(Ar1), система Ax=b1 несовместна; rg(A)=rg(Ar2)=3, система Ax=b2 совместна.

Приведем матрицу системы к ступенчатой форме

$$rref(Ar2) = \begin{bmatrix} 1 & 0 & 0 & -1 & -1 & 2 \\ 0 & 1 & 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Свободные переменные x4, x5, базисные переменные x1,x2,x3.
Запишем и решим эквивалентную систему

Given $\quad x1 - x4 - x5 = 2$

$x2 + x4 + x5 = -1 \qquad Find(x1, x2, x3) \rightarrow \begin{bmatrix} x4 + x5 + 2 \\ -x4 - x5 - 1 \\ x3 \end{bmatrix}$

$x3 = 3$

Общее решение: $\qquad$ **Частные решения:**

$$X(x4, x5) := \begin{bmatrix} x4 + x5 + 2 \\ -x4 - x5 - 1 \\ 3 \\ x4 \\ x5 \end{bmatrix} \quad X(1, 0) = \begin{bmatrix} 3 \\ -2 \\ 3 \\ 1 \\ 0 \end{bmatrix} \quad X(0, 1) = \begin{bmatrix} 3 \\ -2 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

Задание 2.11. Докажите, что векторы $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4$ образуют базис в пространстве L_4 , и найдите координаты вектора $\boldsymbol{x}$ в этом базисе.

Порядок выполнения задания

1. Установите автоматический режим вычислений.
2. Введите матрицу C со столбцами $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4$.
3. Покажите, что определитель матрицы C отличен от нуля.
4. Вычислите матрицу, обратную к матрице C.
5. Запишите выражение и вычислите координаты вектора в базисе $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4$.
6. Проверьте вычисления, выполнив обратный переход.

Пример

Матрица перехода **Координаты вектора x в старом базисе**

$$C := \begin{bmatrix} 1 & 2 & 5 & 1 \\ -2 & -1 & -2 & -1 \\ 1 & 1 & -3 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \qquad |C| = 12 \qquad xe := \begin{bmatrix} 1 \\ -1 \\ 3 \\ -2 \end{bmatrix}$$

Координаты x вектора в новом базисе **Проверка**

$$xf := C^{-1} \cdot xe \qquad xf = \begin{bmatrix} 0.083 \\ 1.333 \\ -0.417 \\ 0.333 \end{bmatrix} \qquad C \cdot xf = \begin{bmatrix} 1 \\ -1 \\ 3 \\ -2 \end{bmatrix}$$

Задание 2.12. Исследуйте на линейную независимость системы векторов $\boldsymbol{f}_1$, $\boldsymbol{f}_2$, $\boldsymbol{f}_3$, $\boldsymbol{f}_4$ и $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$. Выделите в линейно зависимой системе независимую подсистему. Найдите линейные выражения всех векторов линейно зависимой системы через векторы линейно независимой подсистемы.

<u>*Порядок выполнения задания*</u>

1. Установите автоматический режим вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите матрицы C и D, столбцами которых являются векторы $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}$ и $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$ соответственно.
4. Найдите ранги матриц C и D.
5. Сформулируйте вывод о линейной зависимости исследуемых систем векторов.
6. Приведите матрицу линейно зависимой системы к ступенчатому виду и выделите линейно независимую подсистему.
7. Запишите выражения для всех векторов линейно зависимой системы через линейно независимую подсистему и проверьте их.

<u>*Пример*</u>

ORIGIN := 1

$$C := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 4 & 3 \end{bmatrix} \quad \begin{matrix} f1 \equiv C^{\langle 1 \rangle} \\ f2 \equiv C^{\langle 2 \rangle} \\ f3 \equiv C^{\langle 3 \rangle} \\ f4 \equiv C^{\langle 4 \rangle} \end{matrix} \qquad D := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{bmatrix} \quad \begin{matrix} g1 \equiv D^{\langle 1 \rangle} \\ g2 \equiv D^{\langle 2 \rangle} \\ g3 \equiv D^{\langle 3 \rangle} \\ g4 \equiv D^{\langle 4 \rangle} \end{matrix}$$

$\mathrm{rank}(C) = 3$ **Векторы-столбцы матрицы C линейно зависимы**

$\mathrm{rank}(D) = 4$ **Векторы-столбцы матрицы D линейно независимы**

$$\mathrm{rref}(C) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Первые три столбца матрицы С линейно независимы

Выражение векторов линейно-зависимой системы f1, f2, f3, f4 через векторы линейно независимой подсистемы f1, f2, f3

f1=f1

f2=f2

f3=f3

f4=f1 – f2 + f3

Проверка

$$f4 := C^{\langle 1 \rangle} - C^{\langle 2 \rangle} + C^{\langle 3 \rangle} \quad f4 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \end{bmatrix}$$

Задание 2.13. Заданы векторы $\boldsymbol{x}$, $\boldsymbol{y}$ и $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$ евклидова пространства E^4. Докажите, что векторы $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$ образуют ортонормированный базис в E^4. Вычислите $x \cdot y$, $|\boldsymbol{x}|$, $|\boldsymbol{y}|$ и $\cos\varphi$. Найдите координаты векторов $\boldsymbol{x}$, $\boldsymbol{y}$ в базисе $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$ и вычислите $x \cdot y$, $|\boldsymbol{x}|$, $|\boldsymbol{y}|$ и $\cos\varphi$. Сравните полученные величины. Докажите, что матрица перехода к базису $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$ – ортогональная матрица.

Порядок выполнения задания

1. Установите автоматический режим вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите векторы-столбцы $\boldsymbol{x}$, $\boldsymbol{y}$.
4. Определите матрицу U, столбцами которой являются векторы $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$, и вычислите ее ранг.
5. Вычислите длины и попарные скалярные произведения векторов $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$, вычислите U^{-1} и U^T. Сформулируйте вывод об ортогональности матрицы U.
6. Вычислите $x \cdot y$, $|\boldsymbol{x}|$, $|\boldsymbol{y}|$ и $\cos\varphi$.
7. Вычислите координаты векторов $\boldsymbol{x}$, $\boldsymbol{y}$ в базисе $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$.
8. Вычислите $x \cdot y$, $|\boldsymbol{x}|$, $|\boldsymbol{y}|$ и $\cos\varphi$ в базисе $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{g}_3$, $\boldsymbol{g}_4$.

Пример

ORIGIN := 1

$$x := \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} \quad y := \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix} \quad U := \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} \\ \frac{1}{2} & \frac{-1}{2} & \frac{-1}{2} & \frac{-1}{2} \\ \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$g1=U^{\langle 1 \rangle}$

$g2=U^{\langle 2 \rangle}$

$g3=U^{\langle 3 \rangle}$

$g4=U^{\langle 4 \rangle}$

Столбцы матрицы U единичной длины и попарно ортогональны

$U^{\langle 1\rangle}\cdot U^{\langle 1\rangle} = 1$ $U^{\langle 2\rangle}\cdot U^{\langle 2\rangle} = 1$ $U^{\langle 3\rangle}\cdot U^{\langle 3\rangle} = 1$ $U^{\langle 4\rangle}\cdot U^{\langle 4\rangle} = 1$

$U^{\langle 1\rangle}\cdot U^{\langle 2\rangle} = 0$ $U^{\langle 1\rangle}\cdot U^{\langle 3\rangle} = 0$ $U^{\langle 1\rangle}\cdot U^{\langle 4\rangle} = 0$

$U^{\langle 2\rangle}\cdot U^{\langle 3\rangle} = 0$ $U^{\langle 2\rangle}\cdot U^{\langle 4\rangle} = 0$ $U^{\langle 3\rangle}\cdot U^{\langle 4\rangle} = 0$

Транспонированная матрица U   Обратная матрица

$$U^T = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix} \qquad U^{-1} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ -0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}$$

$$x\cdot y = 8 \qquad \sqrt{x\cdot x} = 2.646 \qquad \sqrt{y\cdot y} = 3.162 \qquad \frac{x\cdot y}{\sqrt{x\cdot x}\cdot\sqrt{y\cdot y}} = 0.956$$

Координаты векторов x и y в базисе g1, g2, g3, g4

$$xg := U^T\cdot x \quad xg = \begin{bmatrix} 2.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \qquad yg := U^T\cdot y \quad yg = \begin{bmatrix} 3 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

$$xg\cdot yg = 8 \quad \sqrt{xg\cdot xg} = 2.646 \quad \sqrt{yg\cdot yg} = 3.162 \quad \frac{xg\cdot yg}{\sqrt{xg\cdot xg}\cdot\sqrt{yg\cdot yg}} = 0.956$$

Задание 2.14. Оператор A, действующий в линейном пространстве X_n, задан своей матрицей в некотором базисе. Найдите координаты образа $\boldsymbol{y}=A\boldsymbol{x}$ вектора $\boldsymbol{x}\in X_n$. В линейном пространстве X_n введен новый базис. Найдите координаты вектора $\boldsymbol{x}\in X_n$, координаты образа $\boldsymbol{y}=A\boldsymbol{x}$ и матрицу оператора в новом базисе.

<u>Порядок выполнения задания</u>

1. Установите автоматический режим вычислений.
2. Введите заданную матрицу оператора и вектор $\boldsymbol{x}$.
3. Вычислите координаты образа $A\boldsymbol{x}$ вектора $\boldsymbol{x}$.
4. Введите матрицу P перехода к новому базису в пространстве X_n и найдите обратную к ней.
5. Найдите координаты вектора $\boldsymbol{x}$ в новом базисе.
6. Найдите матрицу оператора в новом базисе.
7. Найдите координаты образа $A\boldsymbol{x}$ в новом базисе.

<u>Пример</u>

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -3 & 2 & 0 \\ -4 & 5 & 11 & 0 \\ 5 & 4 & 3 & 2 \end{bmatrix} \quad x := \begin{bmatrix} 1 \\ -1 \\ 2 \\ -2 \end{bmatrix} \quad y := A \cdot x \quad y = \begin{bmatrix} -3 \\ 7 \\ 13 \\ 3 \end{bmatrix}$$

$$P := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad xnew := P^{-1} \cdot x \quad xnew = \begin{bmatrix} 2 \\ -3 \\ 4 \\ -2 \end{bmatrix}$$

$$ynew := P^{-1} \cdot y \quad ynew = \begin{bmatrix} -10 \\ -6 \\ 10 \\ 3 \end{bmatrix}$$

$$Anew := P^{-1} \cdot A \cdot P \quad Anew = \begin{bmatrix} 1 & 6 & 7 & 11 \\ 4 & -4 & -13 & -13 \\ -9 & -8 & 0 & -2 \\ 5 & 9 & 12 & 14 \end{bmatrix}$$

Проверка:

$$Anew \cdot xnew = \begin{bmatrix} -10 \\ -6 \\ 10 \\ 3 \end{bmatrix} \quad P \cdot Anew \cdot P^{-1} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -3 & 2 & 0 \\ -4 & 5 & 11 & 0 \\ 5 & 4 & 3 & 2 \end{bmatrix}$$

Задание 2.15. Опишите структуру образа и ядра линейного оператора, действующего в линейном пространстве X, заданного своей матрицей в некотором базисе. Проверьте, принадлежат ли векторы $\boldsymbol{x}$, $\boldsymbol{y}$ ядру оператора, а векторы $\boldsymbol{u}$, $\boldsymbol{v}$ – его образу.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите матрицу оператора и векторы-столбцы $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{u}$, $\boldsymbol{v}$.
4. Вычислите ранг матрицы оператора. Сформулируйте вывод о ранге и дефекте оператора.
5. Приведите матрицу оператора к ступенчатому виду.
6. Укажите базис в образе оператора.
7. Запишите линейную однородную систему, которая: описывает ядро оператора, и разрешите ее относительно базисных переменных.
8. Найдите базис в ядре оператора.
9. Проверьте принадлежность векторов $\boldsymbol{x}$ и $\boldsymbol{y}$ ядру оператора.
10. Проверьте принадлежность векторов $\boldsymbol{u}$ и $\boldsymbol{v}$ образу оператора.

Пример

ORIGIN := 1

Матрица оператора

$$A := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 1 & 2 & 2 \\ 2 & 2 & 2 & 3 \end{bmatrix} \quad x := \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \end{bmatrix} \quad y := \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad u := \begin{bmatrix} 4 \\ 5 \\ 6 \\ 9 \end{bmatrix} \quad v := \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

rank(A) = 3 Ранг оператора r=3, дефект оператора d=n-r=4-3=1

$$rref(A) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

В базисный минор входят 2-й, 3-й и 4-й столбцы матрицы, они образуют базис в образе оператора

Базис в образе оператора образуют векторы

$$A^{\langle 2 \rangle} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} \quad A^{\langle 3 \rangle} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix} \quad A^{\langle 4 \rangle} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \end{bmatrix}$$

Базисные переменные x2, x3, x4, свободная переменная x1

Given x1 + x2=0

x3=0

x4=0

$$Find(x2, x3, x4) \rightarrow \begin{bmatrix} -x1 \\ 0 \\ x4 \end{bmatrix}$$

Базис в ядре оператора - вектор $E1 := \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$

$$A \cdot x = \begin{bmatrix} 7 \\ 10 \\ 12 \\ 17 \end{bmatrix} \quad A \cdot y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Вектор Ax отличен от нуля, x не принадлежит Ker(A), Ay=0, y принадлежит Ker(A)

rank(augment(submatrix(A, 1, 4, 2, 4), u)) = 3

Базисные столбцы матрицы оператора и вектор u образуют линейно зависимую систему; вектор u принадлежит Im(A)

rank(augment(submatrix(A, 1, 4, 2, 4), v)) = 4

Базисные столбцы матрицы оператора и вектор v образуют линейно независимую систему; вектор v не принадлежит Im(A)

Задание 2.16. Найдите собственные значения и собственные векторы оператора, заданного матрицей. Запишите матрицу оператора в базисе из собственных векторов, если таковой существует. Запишите матрицу перехода к собственному базису.

Порядок выполнения задания

1. Установите режим автоматических вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите матрицу оператора А.
4. Запишите характеристический многочлен оператора и найдите его корни.
5. Для каждого собственного значения λ найдите собственные векторы – фундаментальную систему решений однородной системы $(A-\lambda E)\boldsymbol{x}=0$.
6. Проверьте подстановкой правильность вычислений.
7. Запишите матрицу оператора в базисе из собственных векторов.
8. Найдите собственные значения и собственные векторы оператора, используя встроенные функции Mathcad.

9. Сравните результаты вычислений.

Пример

$\mathbf{ORIGIN} := 1$

$$A := \begin{bmatrix} 1 & 1 & 2 & 3 \\ 0 & 2 & 2 & 4 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad E := \mathbf{identity}(4) \quad E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1-\lambda & 1 & 2 & 3 \\ 0 & 2-\lambda & 2 & 4 \\ 0 & 0 & 1-\lambda & -2 \\ 0 & 0 & 0 & 2-\lambda \end{bmatrix} \quad (1-\lambda)^2\cdot(2-\lambda)^2 \quad \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix}$$

$$\mathbf{rref}(A - 1\cdot E) = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad E1 := \begin{bmatrix} 0 \\ 2 \\ -1 \\ 0 \end{bmatrix} \quad E2 := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{rref}(A - 2\cdot E) = \begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad E3 := \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad E4 := \begin{bmatrix} 0 \\ 1 \\ -2 \\ 1 \end{bmatrix}$$

$$A\cdot E1 = \begin{bmatrix} 0 \\ 2 \\ -1 \\ 0 \end{bmatrix} \quad A\cdot E2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad A\cdot E3 = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \end{bmatrix} \quad A\cdot E4 = \begin{bmatrix} 0 \\ 2 \\ -4 \\ 2 \end{bmatrix}$$

$$As := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad C := \mathbf{eigenvals}(A) \quad C = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

$$EC := \mathbf{eigenvecs}(A) \quad EC = \begin{bmatrix} 1 & 0.707 & 0 & 0.408 \\ 0 & 0.707 & 0.894 & 0 \\ 0 & 0 & -0.447 & 0.816 \\ 0 & 0 & 0 & -0.408 \end{bmatrix}$$

$$A\cdot EC^{\langle 1\rangle} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad A\cdot EC^{\langle 3\rangle} = \begin{bmatrix} 0 \\ 0.894 \\ -0.447 \\ 0 \end{bmatrix}$$

$$A\cdot EC^{\langle 2\rangle} = \begin{bmatrix} 1.414 \\ 1.414 \\ 0 \\ 0 \end{bmatrix} \quad A\cdot EC^{\langle 4\rangle} = \begin{bmatrix} 0.816 \\ 0 \\ 1.633 \\ -0.816 \end{bmatrix}$$

Задание 2.17. Найдите минимум целевой функции $f(x, y)=ax+by$ при указанных ограничениях.

Порядок выполнения задания

1. Установите режим автоматических вычислений.

2. Запишите в виде $y=kx+b$ уравнения прямых, ограничивающих область допустимых значений переменных.
3. Изобразите на графике соответствующие прямые и определите область допустимых значений переменных.
4. Постройте для одного или нескольких значений C линии уровня целевой функции $f(x, y)=C$ (столько, сколько понадобится, чтобы понять, имеет ли задача решение и где достигается искомый экстремум).
5. Если задача имеет единственное решение, найдите вершину, в которой достигается искомое экстремальное значение (максимум или минимум) целевой функции, и укажите ее координаты.
6. Вычислите значение целевой функции в найденной точке.
7. Если задача имеет бесконечное множество решений, экстремум достигается на отрезке, на полупрямой или на прямой, то вычислите соответствующее экстремальное значение целевой функции и опишите множество решений.
8. Сформулируйте и запишите в рабочем документе вывод.

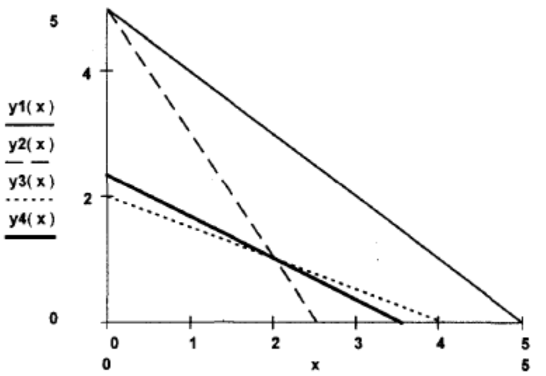
<u>*Пример*</u>

$x + y=5 \quad -x+5 \quad y1(x) := -x+5$

$x + 2\cdot y=4 \quad \frac{-1}{2}\cdot x+2 \quad y2(x) := \frac{-1}{2}\cdot x+2$

$2\cdot x + y=5 \quad -2\cdot x+5 \quad y3(x) := -2\cdot x+5$

$2\cdot x + 3\cdot y=c \quad \frac{-2}{3}\cdot x+\frac{1}{3}\cdot c \quad c := 7 \quad y4(x) := \frac{-2}{3}\cdot x+\frac{1}{3}\cdot c$



Задача имеет единственное решение. Минимум целевой функции достигается в точке пересечения прямых x+2=4, 2x+y=5.

Given $\quad x + 2\cdot y=4$

$\quad 2\cdot x + y=5 \quad \text{Find}(x,y) \rightarrow \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

$f(x,y) := 2\cdot x + 3\cdot y \quad \text{fmin} := f(2,1) \quad \text{fmin} = 7$

Минимальное значение целевой функции равно 7, оно достигается в точке x=2, y=1. fmin=f(2,1)=7.

Задание 2.18. Найдите нормальное обобщенное решение линейной системы $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ для двух вариантов правых частей $\boldsymbol{b}=\boldsymbol{b}_1$ и $\boldsymbol{b}=\boldsymbol{b}_2$. Найдите классическое решение системы, если оно существует, и сравните его с обобщенным.

<u>*Порядок выполнения задания*</u>

1. Установите автоматический режим вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите матрицу системы и обе правые части.
4. Найдите матрицу и правые части нормальной системы метода наименьших квадратов.
5. Найдите нормальные обобщенные решения системы для обеих правых частей.
6. Найдите значения невязок, достигнутые на нормальных решениях.

7. Сформулируйте и запишите в рабочем документе вывод о совместности систем.
8. Найдите классическое решение совместной системы. Сравните его с обобщенным решением.
9. Найдите невязку для классического решения.
10. Сформулируйте вывод и запишите его в рабочем документе.

Пример

ORIGIN := 1

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 4 \\ 2 & 5 & 7 \\ 2 & 4 & 7 \end{bmatrix} \quad b1 := \begin{bmatrix} 3 \\ 3 \\ 7 \\ 6 \end{bmatrix} \quad b2 := \begin{bmatrix} 3 \\ 3 \\ 7 \\ 7 \end{bmatrix}$$

$$AP := A^T \cdot A \quad bp1 := A^T \cdot b1 \quad bp2 := A^T \cdot b2 \quad AP = \begin{bmatrix} 10 & 22 & 35 \\ 22 & 49 & 77 \\ 35 & 77 & 123 \end{bmatrix}$$

$$xn1 := AP^{-1} \cdot bp1 \quad xn1 = \begin{bmatrix} 1 \\ 1 \\ -9.76996 \cdot 10^{-15} \end{bmatrix}$$

$$xn2 := AP^{-1} \cdot bp2 \quad xn2 = \begin{bmatrix} 2.66667 \\ 0.33333 \\ -8.43769 \cdot 10^{-15} \end{bmatrix}$$

$$r1 := |A \cdot xn1 - b1| \quad r2 := |A \cdot xn2 - b2|$$

$$r1 = 2.70127 \cdot 10^{-13} \quad r2 = 0.57735$$

Минимум невязки для системы Ax=b1 с точностью до погрешностей округления равен нулю. Следовательно, система Ax=b1 совместна. Минимум невязки для системы Ax=b2 равен - 0.57735, отличен от нуля, следовательно, система Ax=b2 несовместна

$$rref(augment(A, b1)) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad x := \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad \begin{matrix} r := |A \cdot x - b1| \\ r = 0 \end{matrix}$$

Классическое решение x совместной системы Ax=b1 совпадает с нормальным обобщенным решением xn1, минимум невязки для точного решения равен нулю

Задание 2.19. Найдите методом наименьших квадратов значения коэффициентов линейной зависимости $y=ax+b$ по заданным эмпирическим данным. Используя найденную линейную зависимость, найдите значение y в указанной точке x.

Порядок выполнения задания

1. Установите автоматический режим вычислений.
2. Присвойте переменной ORIGIN значение, равное единице.
3. Введите векторы $\boldsymbol{x}$, $\boldsymbol{y}$, элементы которых – заданные эмпирические данные.
4. Определите матрицу A соответствующей линейной системы, первый столбец которой $A^{<1>}=\boldsymbol{x}$ а элементы второго единицы.

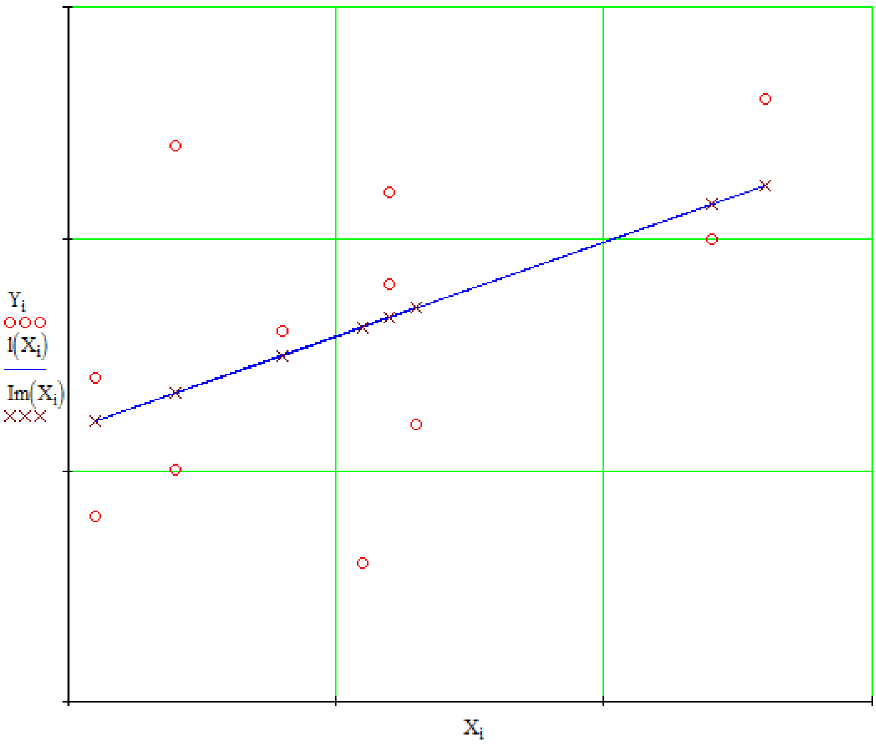
5. Найдите решение нормальной системы метода наименьших квадратов, используя функцию lsolve.
6. Вычислите аппроксимирующую прямую, используя функции intersept(x, y) и slope (x, y), которые вычисляют по заданным векторам экспериментальных данных $\boldsymbol{x}$, $\boldsymbol{y}$ коэффициенты b, a.
7. Изобразите графики полученных линейных функций и заданные экспериментальные точки.
8. Найдите значение $y=ax+b$ в указанной точке x.

Пример

$$X := \begin{pmatrix} 0.871 \\ 0.861 \\ 0.864 \\ 0.873 \\ 0.861 \\ 0.868 \\ 0.872 \\ 0.884 \\ 0.872 \\ 0.864 \\ 0.886 \end{pmatrix} \quad Y := \begin{pmatrix} 0.988 \\ 0.989 \\ 0.990 \\ 0.991 \\ 0.992 \\ 0.993 \\ 0.994 \\ 0.995 \\ 0.996 \\ 0.997 \\ 0.998 \end{pmatrix} \quad A^{\langle 1 \rangle} := X \quad A^{\langle 2 \rangle} := \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} a \\ b \end{pmatrix} := \mathrm{lsolve}\left(A^T \cdot A, A^T \cdot Y\right) \qquad l(x) := a\,x + b$$

$$lm(x) := \mathrm{intercept}(X, Y) + \mathrm{slope}(X, Y)\,x \qquad i := 1$$



$l(0.870) = 0.993$ $\qquad lm(0.870) = 0.99$

$a = 0.204$ $\qquad b = 0.815$

$\mathrm{slope}(X, Y) = 0.204$ $\qquad \mathrm{intercept}(X, Y)$

Методические указания для выполнения практических заданий
***Лабораторной работы 3 «*Основные приемы работы с пакетом MATLAB»**

1.1. Работа в командном окне

1.1.1. Вход в систему MATLAB. Доступ к справочной информации

При работе в операционных системах Windows 9x, Windows 2000, Windows XP вход в пакет MATLAB осуществляется простым кликом левой кнопки мыши по соответствующей иконке. В версиях пакета 5.0 и выше имеется собственный встроенный редактор текста. Командное окно пакета находится в одном окне, а редактор текста – в другом.

Существуют следующие способы получить информацию о функциях пакета MATLAB в процессе работы. Найти дополнительную информацию о командах, опциях и примерах можно, воспользовавшись справкой в диалоговом режиме, литературой и прилагаемой к пакету документацией.

1. Команда **`help`**

Команда **`help <имя_группы>`** выведет на экран список функций, размещенных в этой группе с краткими пояснениями. Например, команда

```
>> help elmat
```

выведет список функций, предназначенных для создания и работы с матрицами специального вида. Ввод команды с именем определенной функции выдаст на экран описание этой функции.

2. Команда **`lookfor`**

Эта команда позволяет выполнить поиск m-функции по ключевому слову, при этом анализируется первая строка комментария и она же выводится на экран, если в ней встретилось ключевое слово. Например, в системе MATLAB нет m-функции с именем `inverse`, и поэтому на команду

```
>> help inverse
```

ответом будет

`inverse.m not found`. Однако команда `lookfor inverse` покажет все совпадения, найденные в справочных файлах.

3. Меню **Help.**
4. Обращение к Web-серверу фирмы The Mathworks, Inc.

1.1.2. Операции строчного редактирования. Сессия MATLAB

При работе с MATLAB в командном режиме действует простейший строчный редактор: стрелками вверх ↑, вниз ↓ можно перелистывать предыдущие команды. Они используются для подстановки после маркера строки ввода **>>** ранее введенных строк, – например, для их исправления, дублирования или дополнения. Такая возможность существует благодаря организации программного стека, хранящего строки с исполненными ранее командами. Клавиша ESC служит для очистки строки ввода.

Кратко приведем некоторые команды управления окном:

`clc` – очистка экрана;

`echo [file_name] on|off` – включает | выключает режим вывода на экран текста файла-сценария;

`echo [file_name]` – меняет режим вывода на противоположный;

`echo on|off all` – включает | отключает режим вывода на экран текста всех m-файлов;

`more on|off` – включает | отключает режим постраничного вывода (для просмотра больших m-файлов).

Сеанс работы с MATLAB называется сессией. Она содержит строки ввода, вывода и сообщений об ошибках. Входящие в сессию определения переменных и функций, расположенные в рабочей области памяти, можно записать на диск (в бинарный файл с расширением .mat), используя команду **`save`**:

`save fn` – запись рабочей области всех переменных в файл `fn.mat`;

`save fn X Y Z` – записываются значения переменных `X`, `Y` и `Z` в файл `fn.mat`.

Команда **`load`** позволяет считать с диска данные рабочей области. Если команда используется в ходе проведения сессии, то произойдет замена значений текущих переменных теми значениями, которые были сохранены в считываемом mat-файле. Для задания имен загружаемых файлов может использоваться знак *, означающий загрузку всех файлов с определенными признаками.

Для очистки рабочего пространства используется функция **`clear`**:

`clear` – уничтожение определений всех переменных;

`clear x` – уничтожение определения переменной `x`;

Уничтоженная (стертая в рабочем пространстве) переменная становится неопределенной. Использовать такие переменные нельзя, такие попытки будут сопровождаться выдачей сообщений об ошибке.

Во избежание непроизводительных потерь памяти при работе с объемными данными (векторы, матрицы и массивы) следует использовать команду **`pack`**, осуществляющую дефрагментацию рабочей области. Эта команда переписывает все определения рабочей области на диск, очищает ее и затем записывает все определения без «дыр» и «мусора» в рабочую память.

Просмотр рабочей области возможен как с помощью браузера **Workspace Browser**, так и в командном режиме: **`who`** выводит список определенных переменных, а команда **`whos`** – список переменных с указанием их размера и объема занимаемой памяти.

Так как сессия является результатом проб и ошибок, ее текст, наряду с правильными определениями, содержит сообщения об ошибках, переопределения функций и переменных и т.д. Поэтому фрагменты сессии можно оформить в виде *дневника* с помощью команды **`diary`**:

`diary file_name` – ведет запись на диск в виде текстового файла с указанным именем всех команд в строках ввода и полученных результатов;

`diary off | on` – приостанавливает / начинает запись в файл.

Таким образом, чередуя команды `diary off` и `diary on`, можно сохранять нужные фрагменты сессии в их формальном виде. Например:

```
>> diary myfile1.txt
>> 1+2;
>> diary myfile2.txt
>> 2+3;
>> diary off
>> sin(l)
ans =
0.8415
>> diary on
```

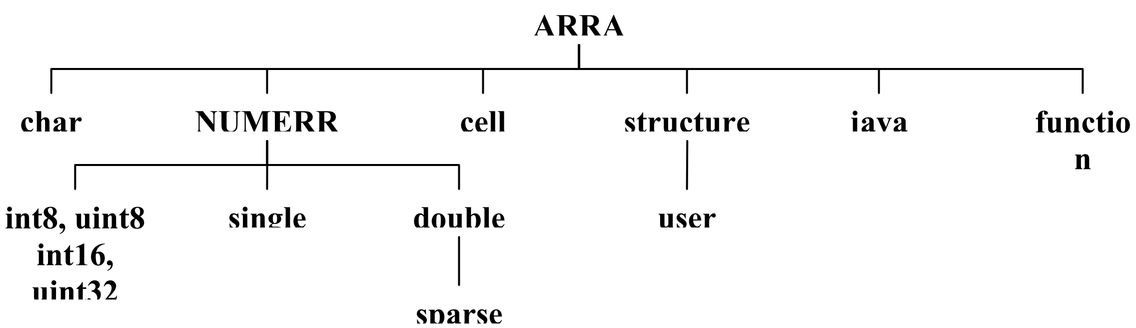
С помощью команды **`type`** возможно просмотреть содержимое текста такого файла со всеми записанными действиями.

<u>Перенос строки</u>

Если математическое выражение выходит за размер экрана монитора, то целесообразно перенести его часть на следующую строку. Для этого используется символ многоточие `...` – три и более точки. В командном режиме число возможных символов в одной строке – 4096, в m-файле – не ограничено, но с такими длинными строками работать неудобно. Поэтому применение в файлах-сценариях символа переноса строки улучшает наглядность программ.

<u>Типы данных</u>

Структура типов данных MATLAB можно представить в виде схемы:



Типы данных array и numerric являются виртуальными, так как к ним нельзя отнести какие-либо переменные. Они служат для определения и комплектования некоторых типов данных. Таким образом, в MATLAB определены типы данных, представляющие собой многомерные массивы:

1. **`single`** – числовые массивы с числами одинарной точности;
2. **`double`** – числовые массивы с числами удвоенной точности;
3. **`char`** – строчные массивы с элементами-символами;
4. **`sparse`** – разреженные матрицы с элементами-числами двойной точности;
5. **`cell`** – массивы ячеек;
6. **`structure`** – массивы структур с полями;
7. **`function handle`** – дескрипторы функций;
8. **`int8`**, ..., **`uint32`** – массивы 8-, 16-, 32-разрядных целых чисел со знаками и без знаков.

1.1.3. Арифметические операции. Встроенные скалярные функции. Оператор присваивания. Форматы вывода данных. Логические операторы и операторы отношения

MATLAB можно использовать как обыкновенный калькулятор, который выполняет основные арифметические операции: + (сложение), – (вычитание), * (умножение), / (деление), ^ (возведение в степень) и содержит набор констант (ans, pi, e, i, eps, realmax, realmin, inf, NaN, end).

Пример.

```
>> (2+3*pi)/2
ans =
     5.7124
```

Для более сложных вычислений в пакете предусмотрены дополнительные операции. Кратко перечислим основные функции, имеющиеся в пакете. Описание других можно найти, используя справку в диалоговом режиме.

Функция	Описание	Функция	Описание
`abs()`	Абсолютное значение числа	**`cos()`**	cos(x) – косинус
`sin()`	sin(x) – синус	**`tan()`**	tg(x) – тангенс
`exp()`	e^x – экспонента	**`log()`**	ln(x) – натуральный логарифм
`acos()`	arccos(x) – арккосинус	**`sqrt()`**	$\sqrt{x}$ – извлечение кв. корня
`atan()`	arctg(x) – арктангенс	**`round()`**	Округление числа до ближайшего целого
`ceil()`	Наибольшее целое для данного действительного числа	**`floor()`**	Наименьшее целое число для данного действительного числа
`rem()`	Остаток от деления двух чисел	**`sign()`**	Знак числа (1,если знак + и -1, если знак –)
`real()`	Вещественная часть комплексного числа	**`imag()`**	Мнимая часть комплексного числа
`angle()`	Аргумент комплексного числа	**`conj()`**	Комплексно сопряженное число

MATLAB является интерпретирующим языком непосредственных вычислений. Операторы пакета обычно имеют форму:

`>> имя_переменной=выражение`

или просто

`>> выражение`

Выражение формируется из операторов, функций и имен переменных. Если имя переменной и знак равенства в левой части отсутствуют, автоматически генерируется переменная **ans**, которой присваивается результат вычислений. MATLAB различает строчные и прописные буквы в именах команд, функций и переменных!

Например, для присвоения выражениям различных имен используется знак равенства.

```
>> s=3-floor(exp(3.3))
s  =
   -24
```

Точка с запятой позволяет задать несколько выражений и вывод значений результата после нажатия клавиши <Enter> не производится:

```
>> d=sin(5.6);              Задание d.
>> a=cos(6.5);              Задание a.
>> a+d                      Вывод на экран a+d.
ans =
    0.3453
```

В пакете MATLAB используются следующие операторы отношений и логические операторы:

==	Равно	~	НЕ
~=	Не равно	&	И
<	Меньше	|	ИЛИ
>	Больше	xor	ИЛИ НЕ
<=	Меньше или равно		
>=	Больше или равно		

Поскольку все вычисления в MATLAB выполняются с двоичной точностью, формат вывода может изменяться с помощью следующих команд:

Название формата	Описание формата представления числа
format short	С фиксированной точкой и 4 знаками после точки (по умолчанию)
format long	С фиксированной точкой и 14 знаками после точки
format short e	Научная нотация с 4 десятичными знаками
format long e	Научная нотация с 15 десятичными знаками
format short g	С фиксированной точкой и 3 знаками после точки
format long g	Научная нотация с 13 десятичными знаками
format hex	Шестнадцатеричный формат
format bank	Денежный формат (два знака после точки)
format rat	Числа представляются отношением двух целых чисел
format compact	Удаление вывода пустых строк между строками результата
format loose	Вывод пустых строк между строками результата

Например:

```
>> format long
>> 3*cos(1.2)
ans =
    1.08707326343002
```

1.1.4. Создание матриц. Операции с матрицами и массивами. Векторные и матричные функции

MATLAB работает с одним видом объектов – числовыми прямоугольными матрицами, элементами которых могут быть в общем случае комплексные числа. Все переменные представляют собой матрицы, матрицы 1×1 интерпретируются как скаляры, матрицы с одной строкой или одним столбцом – как векторы. В системе MATLAB матрицы могут быть созданы разными способами:

1) введены явно с помощью списка элементов;
2) сгенерированы встроенными операторами или функциями;
3) созданы в m-файлах;
4) загружены из внешнего файла данных.

Например, в результате выполнения оператора

```
>> A=[1 2 3;3 2 1;4 5 6]
```

или

```
>> A=[1 2 3
        3 2 1
        4 5 6]
```

MATLAB создает матрицу 3×3 и присваивает ее значение переменной A.

```
A =
     1   2   3
     3   2   1
     4   5   6
```

В пакет MATLAB встроен ряд функций, позволяющих создавать матрицы специального вида. Приведем некоторые из них:

1) функция **rand(n)** создает матрицу размера n×n (функция **rand(m,n)** – матрицу размера m×n), каждый элемент которой – случайное число с равномерным законом распределения в диапазоне [0,1];

2) функция **magic(n)** создает матрицу размера n×n, которая является магическим квадратом;

3) функция **zeros(m,n)** создает нулевую матрицу размера m ×n;

4) функция **ones(m,n)** создает матрицу размера m×n, каждый элемент которой равен единице;

5) функция **diag(x)** создает матрицу, у которой на главной диагонали стоят элементы вектора x;

6) функция **diag(A)** создает матрицу, у которой на главной диагонали стоят диагональные элементы матрицы A.

7) функция **eye(m,n)** создает единичную матрицу размером (m×n), т. е. с единицами по главной диагонали и остальными нулевыми элементами;

8) функция **fliplr(A**) формирует матрицу, переставляя столбцы известной матрицы A относительно вертикальной оси;

9) функция **flipud(A)** переставляет строки заданной матрицы A относительно горизонтальной оси;

10) функция **rot90(A)** формирует матрицу путем "поворота" заданной матрицы A на 90 градусов против часовой стрелки;

11) функция **reshape(A,m,n)** образует матрицу размером (m×n) путем выборки элементов заданной матрицы A по столбцам и последующему распределению этих элементов по n столбцам, каждый из которых содержит m элементов; при этом число элементов матрицы A должно равняться m×n;

12) **tril(A)** образует нижнюю треугольную матрицу на основе матрицы A путем обнуления ее элементов выше главной диагонали;

13) **triu(A)** образует верхнюю треугольную матрицу на основе матрицы A путем обнуления ее элементов ниже главной диагонали;

14) **rref(A)** осуществляет приведение матрицы к верхней треугольной форме.

Ссылки на отдельные элементы матриц и векторов осуществляются с помощью индексов в круглых скобках.

Приведем несколько примеров задания матриц, выделения элементов и подматриц.

`>> Z=zeros(4,5);` *Создание нулевой матрицы размера 4×5.*

```
>> O=ones(3,2);                   Создание единичной матрицы размера 3×2.
>> X=0:0.5:2                            Вывод на экран вектора X размера 1×5.
X =
  0  0.5000  1.0000  1.5000  2.0000
>> Y=sin(X);                            Создание вектора Y размера 1×5, каждый
                                        элемент которого является sin от X.
>> A(2,3)
ans =
    1                           Выделение элемента (2,3) из матрицы A.
>> A(8)                                 Альтернативное выделение элемента (2,3)
ans =                                   как 8-го по счету, если записать матрицу в
    1                           один вектор-столбец.
>> A(1:2,1:3)                           Выделение подматриц A.
ans =
    1  2  3
    3  2  1
>> A([1 2],[1 3])
ans =
    1  3
    3  1
>> A(2,2)=round(tan(8))                 Замена элемента матрицы A.
ans =
    1  2  3
    3 -7  1
    4  5  6
>> A(:,2)                               Выделение 2-го столбца.
ans =
    2
   -7
    5
>> A(2,:)                               Выделение 2-й строки.
ans =
    3 -7  1
```

В пакете MATLAB доступны следующие матричные операции: + (сложение), – (вычитание), * (умножение), ^ (возведение в степень, применима только для квадратных матриц!), ’ (транспонирование), / (правое деление), \ (левое деление). Если размерность матриц не соответствует выполняемой операции, то система генерирует сообщение об ошибке. Приведенные операции могут стать поэлементными, если перед ними поставить точку. Приведем примеры использования матричных операций.

```
>> D=[1 2;4 8];
>> C=D’
C =
   1  4
   2  8
>> 5*(D*C)^2
ans =
    2125  8500
    8500  3400
>> C^2                          Произведение матриц C*C.
ans =
    9  36
   18  72
>> C.^2                         Квадрат каждого элемента матрицы C.
ans =
```

```
    1  16
    4  64
>> sin(D./2)
ans =
    0.4794  0.8415
    0.9093  0.7568
```

Часто возникает необходимость использования векторных или матричных функций.

Аргументами векторных функций являются векторы (строки или столбцы). Если в качестве аргумента указана матрица размера m×n, то данная функция действует постолбцово, т.е. результатом действия является вектор-столбец, каждый элемент которого – результат действия этой функции на соответствующий столбец. Например, максимальный элемент прямоугольной матрицы находится с помощью команды **max(max(A))**. Наиболее употребительными матричными функциями являются:

inv(X) – обратная матрица,

det(X) – определитель матрицы,

size(X) – размерность матрицы,

norm(X) – норма вектора или матрицы,

rank(X) – ранг матрицы,

cond(X) – число обусловленности, …

1.2. Программирование. Циклы. Условные операторы

Операторы управления MATLAB и операторы отношения при использовании работают так же, как и в большинстве языков программирования.

Оператор цикла **for** в общем виде записывается как:

```
for <переменная_цикла>=<выражение_цикла>
  <Выполняемые_операторы>
end
```

Оператор цикла **while** записывается следующим образом:

```
while <условие>
  <Выполняемые_операторы>
end
```

Условный оператор **if** имеет следующий синтаксис:

```
if <условие>
  <Выполняемые_операторы1>
else
  <Выполняемые_операторы2>
end
```

В пакете MATLAB возможно также множественное ветвление.

Оператор варианта **switch**, каждая ветвь которого определяется оператором **case**; переход в нее выполняется тогда, когда переменная оператора **switch** принимает значение, указанное после case, или одно из значений списка case. После выполнения какой-либо из ветвей происходит выход из switch, при этом значения, заданные в других ветвях case, уже не проверяются. Если подходящих значений не нашлось, то выполняется ветвь оператора переключения, соответствующая otherwise. Например:

```
switch a
  case -1
    disp('a = -1')
  case 0
    disp('a = 0')
  case 1
    disp('a = 1')
  case {2, 3, 4}
    disp('a равно 2 или 3 или 4 ')
```

```
  otherwise
    disp('а не равно -1, 0, 1, 2, 3 и 4')
end
```

Для обработки исключительных ситуаций используется конструкция **`try...catch`**:

```
try
  % операторы, выполнение которых может привести к ошибке
catch
  % операторы, которые следует выполнить при возникновении
  % ошибки в блоке между try и catch
end
```

Например:

```
try
  A = load('my.dat');
  pie(A)
catch
  disp(' не могу найти файл my.dat ')
end
X = [1; 2; -1; -2];
X = X.^2
```

Приведем несколько примеров использования циклов и условий.

Пример. Найдем значение факториал пяти (5!).

```
>> a=1;
>> for i=1:5
a=a*i;
end
>> a
a =
  120
```

Пример. Для различных значений *x*, полученных делением на 3 чисел от 0 до 10, выведем значения квадратов и кубов этих чисел.

```
>> n=10;
>> k=0;
>> while k<=n
x=k/3;
disp([x x^2 x^3])
k=k+1;
end
```

Пример. В зависимости от *s* присвоим переменной *a* разные значения.

```
>> s=10;
>> if (s<0)|(s>10)
a=s^2+5;
else
a=0;
end
>> a
a =
  0
```

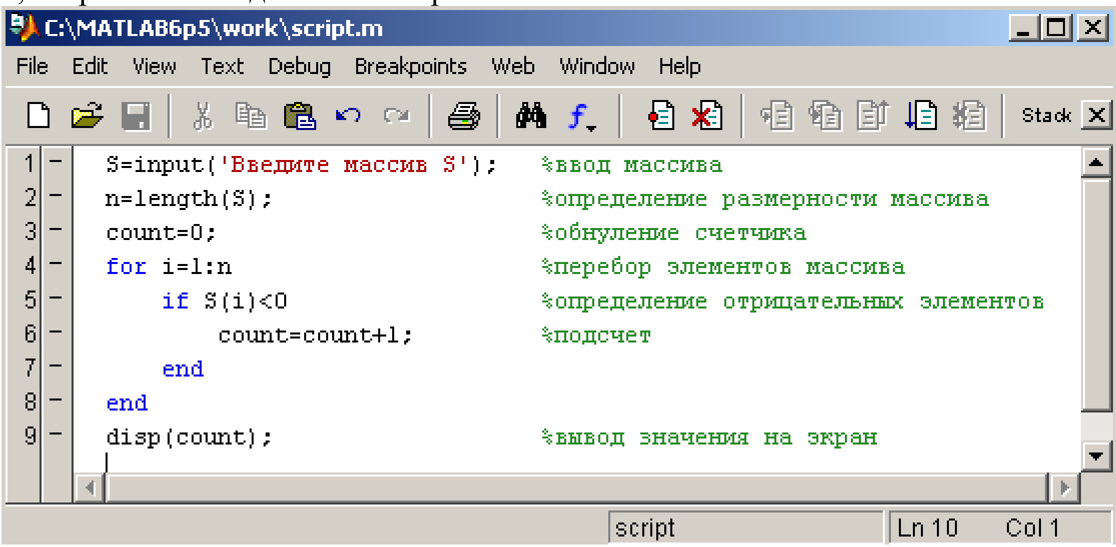
1.3. М-файлы

MATLAB может последовательно выполнять последовательность операторов, записанных в файл на диске. Имена таких файлов имеют вид <имя>.m и такие файлы называются m-файлами. Большая часть работы в MATLAB заключается в создании, редактировании и выполнении подобных m-файлов. Существуют два типа m-файлов: файлы-программы (сценарии) и файлы-функции.

1.3.1. Файлы-программы

Файлы-программы состоят из последовательности обычных операторов пакета MATLAB. Если m-файл с таким сценарием имеет имя, – например, start.m, то команда `start`, введенная в командной строке, вызовет выполнение последовательности операторов этого файла. Переменные в программе являются глобальными и могут изменить значения переменных с теми же именами в командном окне. В таких файлах легко исправить ошибки, внутри файла можно ссылаться на другие m-файлы и рекурсивно на этот же файл.

Приведем пример программы, которая подсчитывает число отрицательных элементов массива, сохраненной под именем script.m.



Вызов в командной строке

```
>> script
```

приведет к выполнению операторов программы:

```
>> script
Введите массив S   [2 -7 4 -8 2 -56]
     3
```

1.3.2. Файлы-функции

Файлы-функции дают возможность расширять MATLAB, поскольку определенные пользователем функции имеют тот же статус, что и другие функции MATLAB. Переменные по умолчанию являются локальными и не влияют на имена и значения переменных в текущей рабочей области MATLAB.

Общий вид задания:

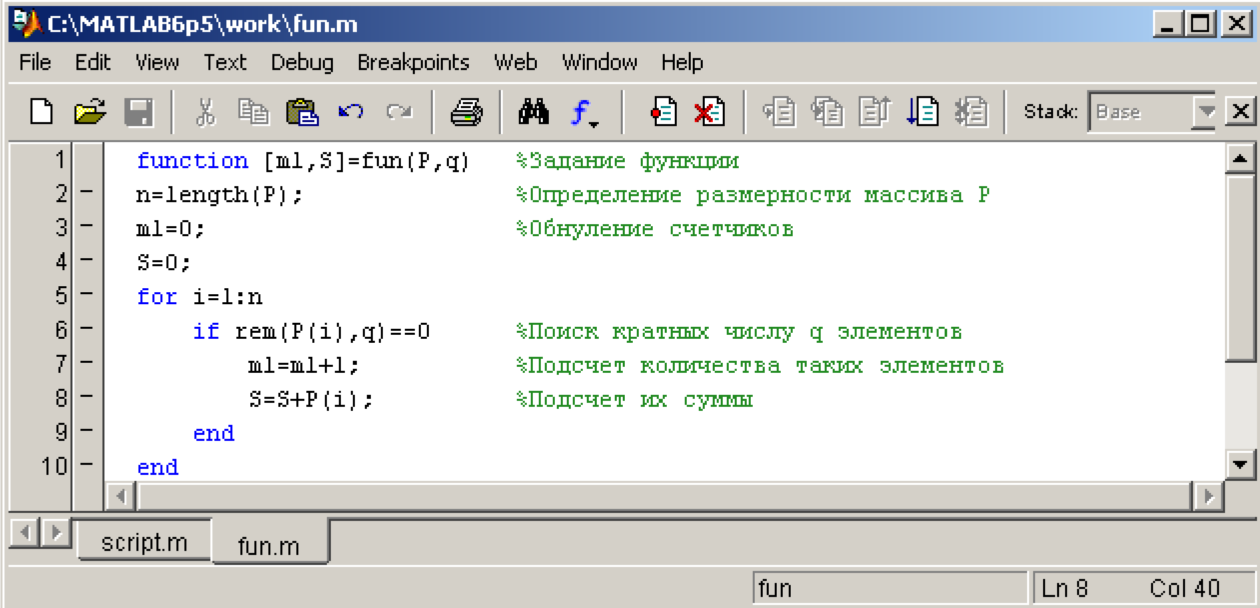
```
function [x1,x2,…]=name(y1,y2,…)
<тело функции>
```

Первая строка функции содержит объявление выходных аргументов, имени функции и входные аргументов, без этой строки файл является программой, но не функцией.

Совпадение имени функции и имени файла является обязательным условием в MATLAB.

Символ % указывает на то, что вся строка после него является комментарием и игнорируется при выполнении программы.

Приведем пример функции, которая производит подсчет количества элементов массива, кратных заданному числу, и вычисляет сумму таких элементов.



Вызов в командной строке (после задания массива P):

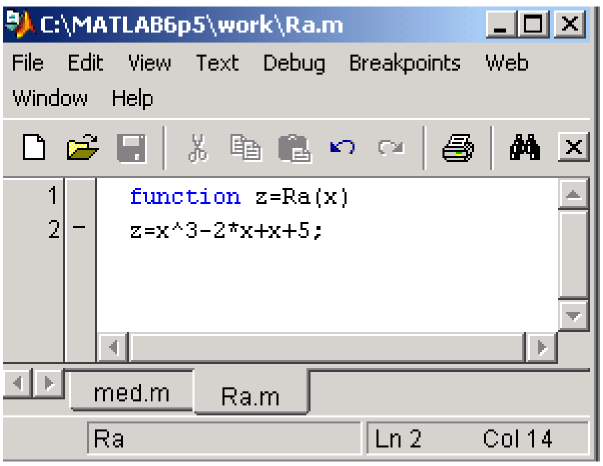
```
>> [m,S]=fun(P,3)
m =
     2
S =
    12
```

присвоит переменным *m* и *S* значения выходных параметров.

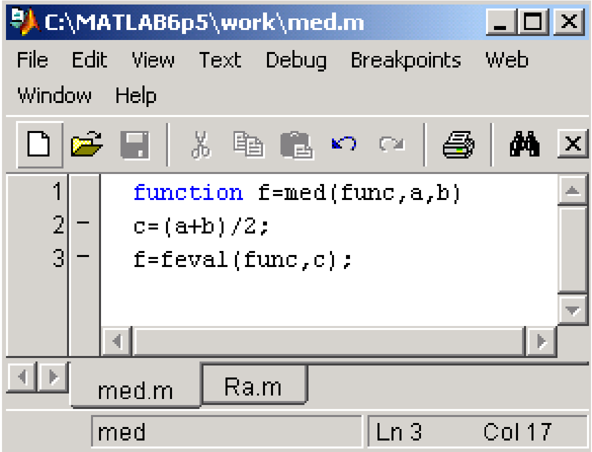
Для вычисления значения функции, имя которой передается в функцию через строковую переменную, часто используется функция **`feval`**, аргументами которой являются имя функции и координата точки, в которой вычисляется значение данной функции.

Приведем пример использования функции **`feval`**.

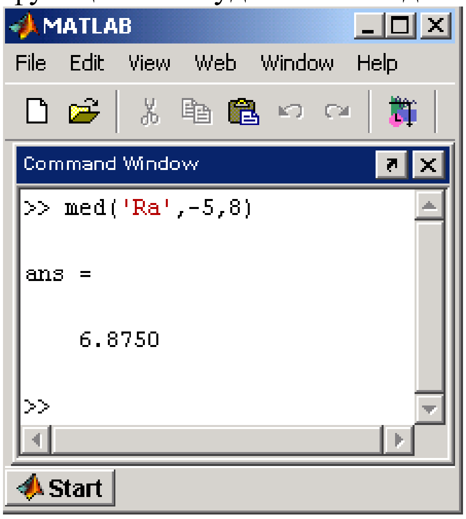
Зададим свою функцию `Ra` изменений аргумента.



И пусть, например, требуется обратиться к этой функции `med` в другом файле-функции, вычисляющем значение функции в середине указанного отрезка:



Результат обращения к функции med будет иметь вид:



1.3.3. Функции, задаваемые пользователем

Рассмотрим два способа задания функций, задаваемых пользователем в программе MATLAB. Первый способ использует команду inline, а второй использует оператор @, чтобы создать так называемую «анонимную функцию». Второй метод является новым в программе MATLAB 7, и в настоящее время этому методу отдается предпочтение. Например:

```
>> f=@(x) x^2
f =
    @(x) x^2
>> f1=inline('x^2','x')
f1 =
    Inline function:
    f1(x) = x^2
```

Когда функция задана, вы можете ее вычислить, например:

```
>> f(4)
ans =
     16
>> f1(4)
ans =
     16
```

Чтобы получить векторизованную версию функции:

```
>> f=@(x) x.^2
>> f1=inline('x.^2', 'x')
```

Теперь мы можем вычислить любую функцию для вектора, например:

```
>> f(1:5)
ans =
     1  4  9  16  25
```

По аналогии, можно задавать функцию и большего числа переменных:

```
>> g=@(x,y) x^2+y^2; g(1,2);
>> g1=inline('x^2+y^2', 'x', 'y'); g1(1,2)
ans =
     5
```

Для того чтобы вычислить векторы – значения функции в точках (1, 3) и (2, 4), функцию нужно задать следующим образом:

```
>> g=@(x,y) x.^2+y.^2;
>> g([1 2], [3 4])
ans =
     10  20
```

1.3.4. Текстовые строки. Сообщения об ошибках

Текстовые строки вводятся в MATLAB в виде текста в одинарных кавычках: m=`Mistake`. Вывод текстовой строки осуществляется при помощи оператора **`disp`**. Оператор `disp(m)` выведет на экран сообщение:

```
Mistake
```

Сообщение об ошибках лучше выводить при помощи функции **`error`**, так как после обращения к данной функции выполнение m-файла будет прекращено. Для выхода из цикла (и из программы) можно воспользоваться командой **`break`**.

В m-файле запрос на ввод данных можно организовать при помощи оператора **`input`**. При вводе оператора `m=input('введите размерность массива')` на экран выводится запрос и выполнение работы файла приостанавливается до ввода требуемых данных с клавиатуры. После нажатия клавиши <Enter> данные присваиваются переменной m.

*Методические указания для выполнения практических заданий Лабораторной работы 4 «***Работа с графикой в MATLAB»**

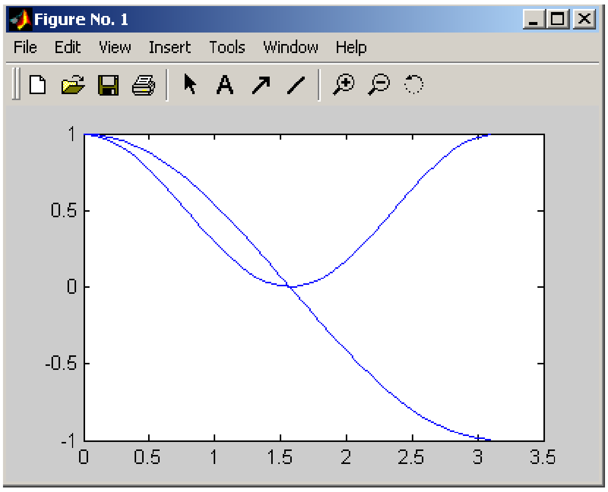
Пакет MATLAB позволяет строить двух- и трехмерные графики кривых и поверхностей. Дополнительные возможности и описание графиков можно найти, используя справку в диалоговом режиме.

2.1. Построение двумерных графиков

Команду `plot` используют для построения графика двухмерной функции. Следующий пример иллюстрирует, как построить графики функций $y = \cos(x)$ и $y = \cos^2(x)$ на интервале [0,π].

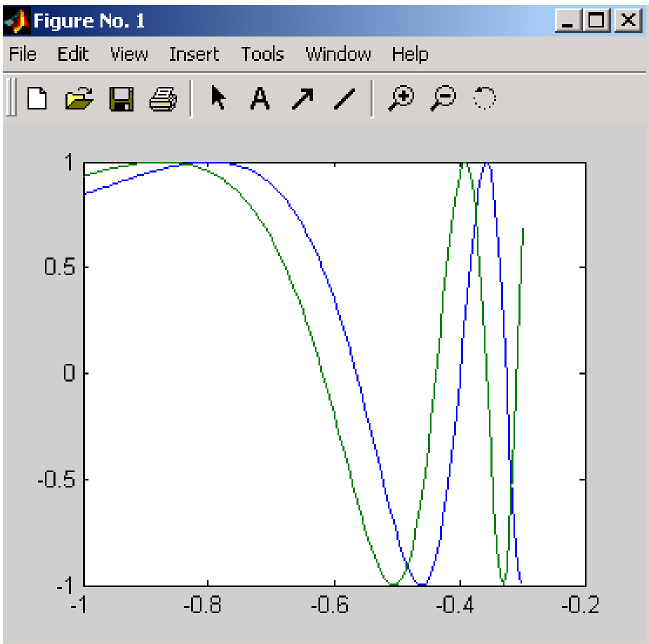
```
>> x=0:0.1:pi;
>> y=cos(x);
>> z=cos(x).^2;
>> plot(x,y,x,z,'b')
```

В первой строке программы задается область с шагом 0.1. В следующих двух строках задаются две функции. Первые три строки заканчиваются точкой с запятой – это предотвращает вывод матриц x, y, z на экран. Команда plot строит графики сплошной линией синего цвета.



Сравнение нескольких функций удобно производить, отобразив их графики на одних осях. Например, построим на отрезке $[-1,-0.3]$ графики функций $f(x)=\sin\left(\frac{1}{x^2}\right)$, $g(x)=\sin\left(\frac{1.2}{x^2}\right)$.

```
>> x=-1:0.005:-0.3;
>> f=sin(x.^-2);
>> g=sin(1.2*x.^-2);
>> plot(x,f,x,g)
```



Команда `plot` позволяет легко задать стиль (тип маркера и тип линий) и цвет линий, например:

```
>> plot(x,f,'k-',x,g,'k:')
```

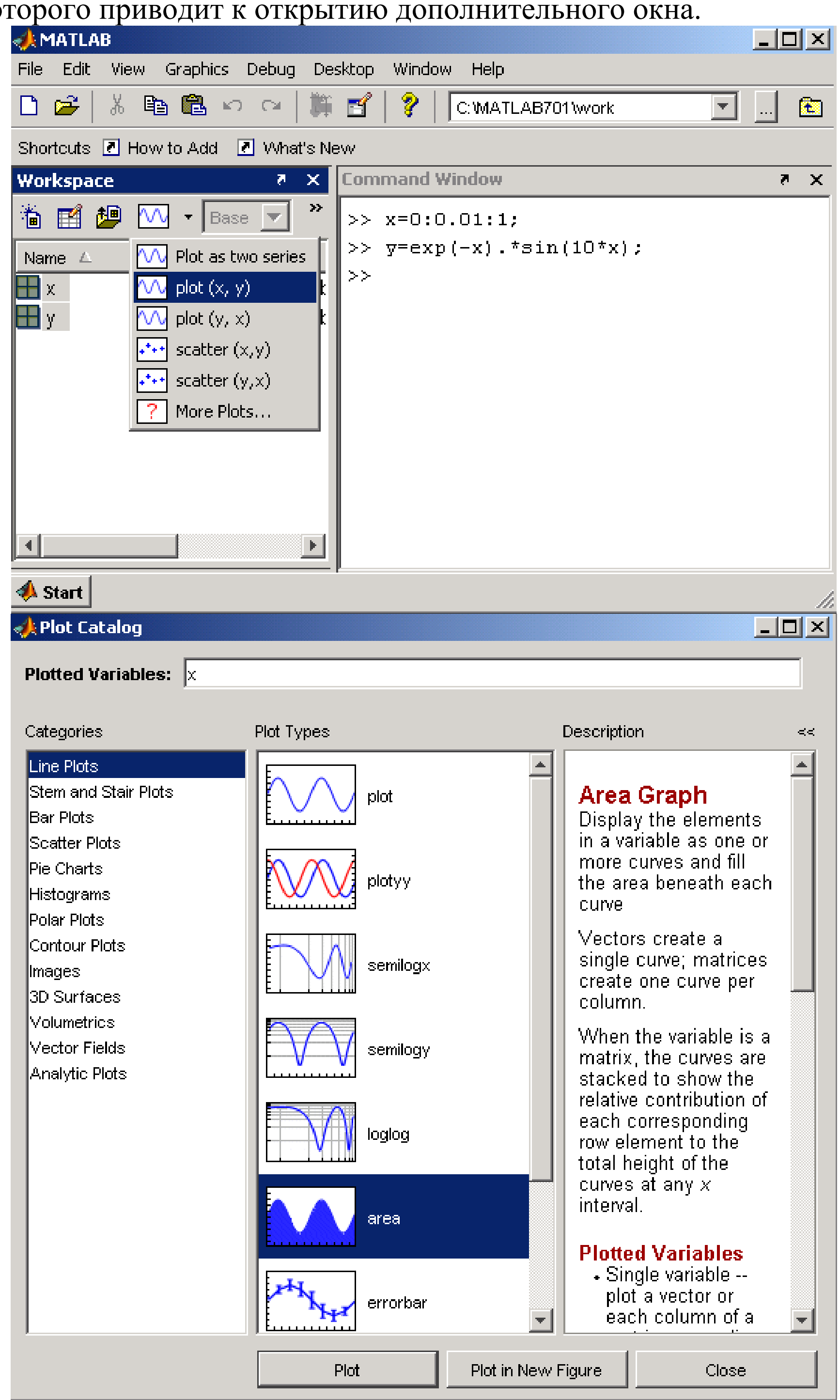
осуществляет построение первого графика сплошной черной линией, а второго – черной пунктирной.

В таблице приведены возможные значения аргументов с указанием результата.

Цвет		Тип маркера		Тип линии	
y	желтый	.	точка	-	сплошная
m	розовый	o	кружок	:	пунктирная
c	голубой	×	крестик	-.	штрихпунктирная
r	красный	+	знак «плюс»	--	штриховая
g	зеленый	*	звездочка		

b	синий	s	квадрат		
w	белый	d	ромб		
k	черный	v,∧,<,>,∨	треугольники		
		p,h	звезды		

Создание различных типов графиков можно осуществить с помощью средств графического интерфейса окна **Workspace**. Вместо вызова команды `plot` выделим два массива x и y в области **Workspace** и выберем команду `plot(x,y)` из панели инструментов окна. Другие типы графиков отображены в этом же контекстном меню и в пункте **More Plots,** выбор которого приводит к открытию дополнительного окна.

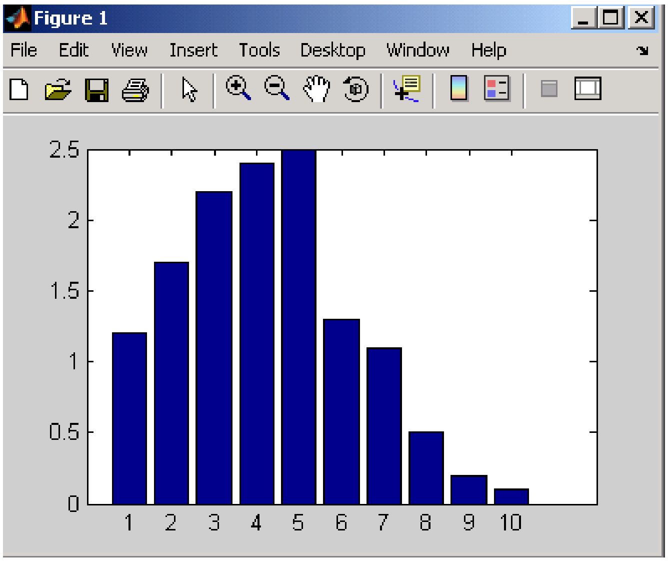


2.2. Диаграммы и гистограммы

Наглядным способом представления векторных и матричных данных являются диаграммы и гистограммы. Значение элемента вектора пропорционально высоте столбика диаграммы или площади сектора диаграммы. Гистограммы используются для получения информации о распределении данных по заданным интервалам.

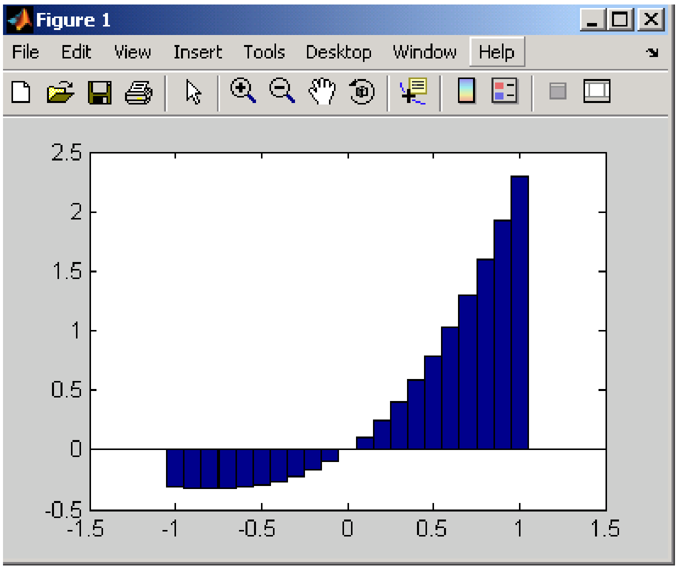
Отображение вектора в виде столбчатой диаграммы осуществляется функцией `bar`.

```
>> data=[1.2 1.7 2.2 2.4 2.5 1.3 1.1 0.5 0.2 0.1];
>> bar(data)
```



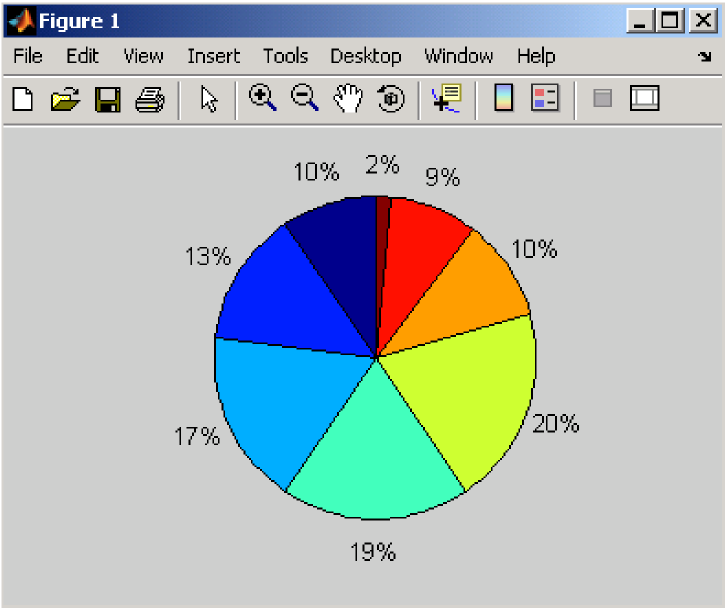
Разметку горизонтальной оси можно задать вектором с возрастающими значениями, что учитывается в первом аргументе функции `bar`. Выбор ширины столбцов осуществляется заданием третьего дополнительного аргумента. Например:

```
>> t=-1:0.1:1;
>> x=sin(t).*exp(t);
>> bar(t,x,1.0)
```



Для учета вклада каждого из элементов вектора в общую сумму его элементов оказываются полезными круговые диаграммы:

```
>> data=[1.2 1.7 2.2 2.4 2.5 1.3 1.1 0.2];
>> pie(data)
```

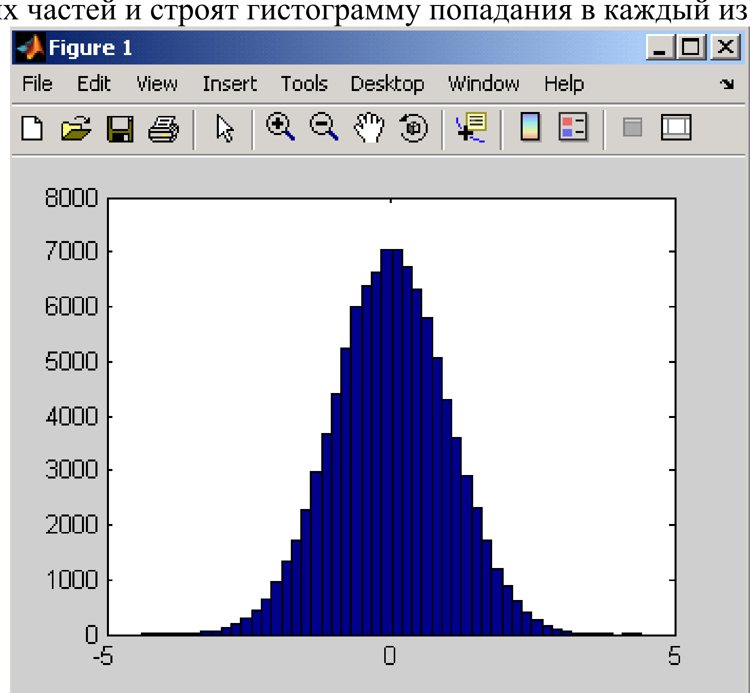


Визуализация векторных данных может быть осуществлена при помощи `pie3` и `bar3`, которые строят трехмерные круговые и столбчатые диаграммы.

Для получения наглядного представления о распределении данных служит функция `hist`.

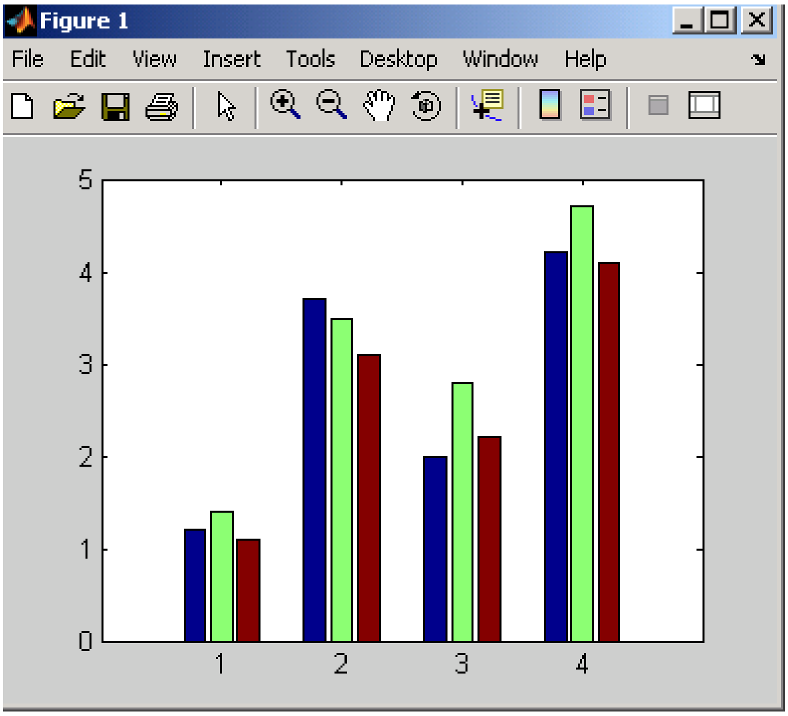
```
>> data=randn(100000,1);
>> hist(data,50)
```

Данная последовательность команд заполняет вектор `data` числами, распределенными по нормальному закону, разбивает интервал, которому они принадлежат, на пятьдесят равных частей и строят гистограмму попадания в каждый из интервалов.



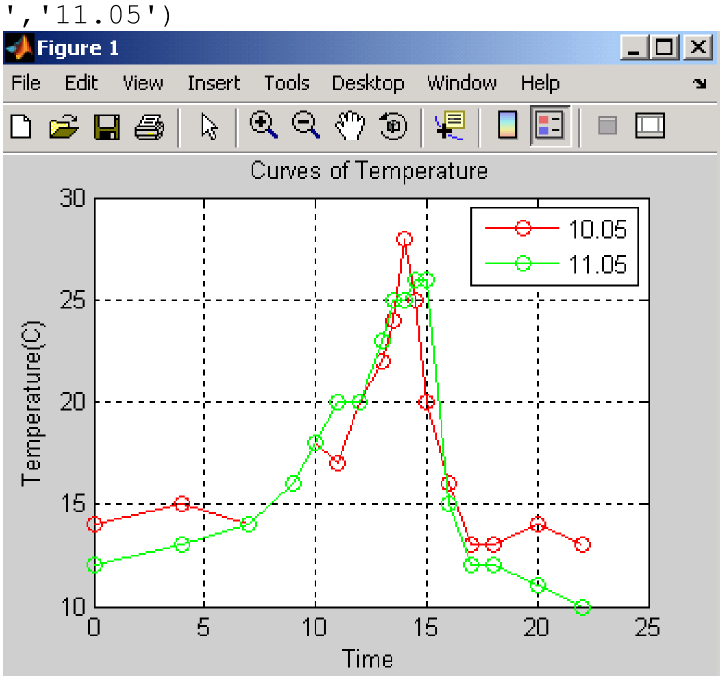
Предположим, в матрице DATA, состоящей из четырех строк и трех столбцов, содержатся результаты измерений трех величин за четыре момента времени. Для построения столбчатой диаграммы:

```
>> DATA=[1.2 1.4 1.1;3.7 3.5 3.1;2 2.8 2.2;4.2 4.7 4.1];
>> bar(DATA)
```



Удобство оформления графиков зависит от дополнительных элементов оформления: координатной сетки, подписей к осям, заголовка и легенды. Сетка наносится командой `grid on`, функции `xlabel`, `ylabel` служат для размещения подписей к осям, `title` – для заголовка, `legend` – сопровождение легендой. Приведем пример графиков изменения суточной температуры.

```
>> time = [0 4 7 9 10 11 12 13 13.5 14 14.5 15 16 17 18 20 22];
>> temp1=[14 15 14 16 18 17 20 22 24 28 25 20 16 13 13 14 13];
>> temp2=[12 13 14 16 18 20 20 23 25 25 26 26 15 12 12 11 10];
>> plot (time,temp1,'ro-',time,temp2,'go-')
>> grid on;
>> title('Curves of Temperature');
>> ylabel('Temperature(C)');
>> xlabel('Time');
>> legend('10.05','11.05')
```



2.3. Графики функций двух переменных

Для отображения функции двух переменных следует:

1. Сгенерировать матрицы с координатами узлов сетки на прямоугольной области определения функции.
2. Вычислить функцию в узлах сетки и записать полученные значения в матрицу.
3. Использовать одну из встроенных функций пакета.

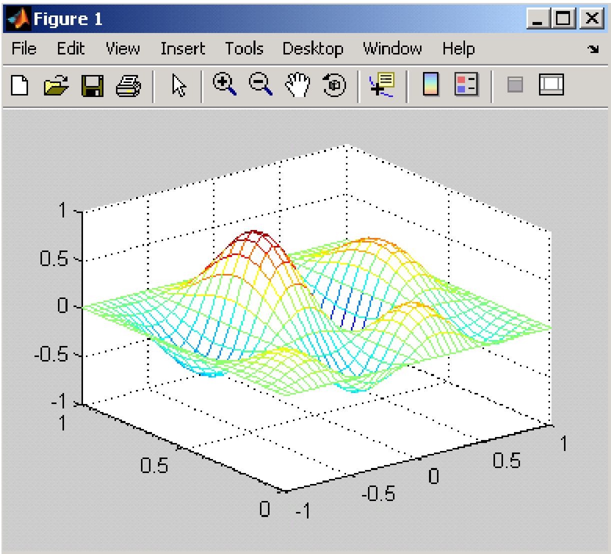
Сетка генерируется функцией `meshgrid`, вызываемой с двумя входными аргументами:

```
>> [X,Y]=meshgrid(-1:0.05:1, 0:0.05:1);
>> Z=4*sin(2*pi*X).*cos(1.5*pi*Y).*(1-X.^2).*Y.*(1-Y);
>> mesh(X,Y,Z)
```

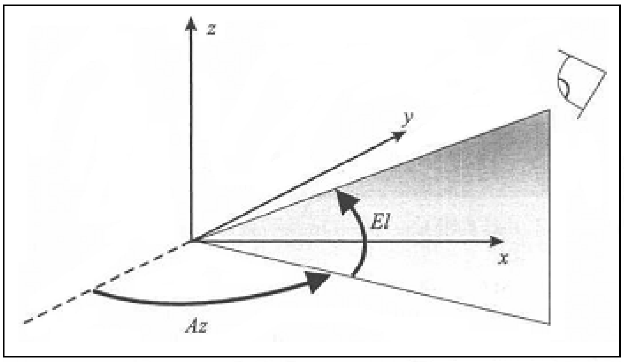
Функция `surf` строит каркасную поверхность графика функции и заливает каждую клетку определенным цветом, зависящим от значения функции в точках, соответствующим углам клетки.

```
>> surf(X,Y,Z)
```

Команда `shading flat` позволяет убрать каркасные линии, а команда `shading interp` позволяет получить поверхность плавно залитую цветом. Команда `colorbar` выводит рядом с графиком цветовую шкалу. Линии уровня можно построить с помощью команд `meshc, surfc`, которые имеют те же входные аргументы, что и `mesh, surf`.



Графические возможности пакета позволяют менять положение трехмерного графика. Положение наблюдателя характеризуется двумя углами: азимутом и углом возвышения. Изменение положения наблюдателя относительно графика осуществляет функция `view`. В примере указаны значения углов по умолчанию.

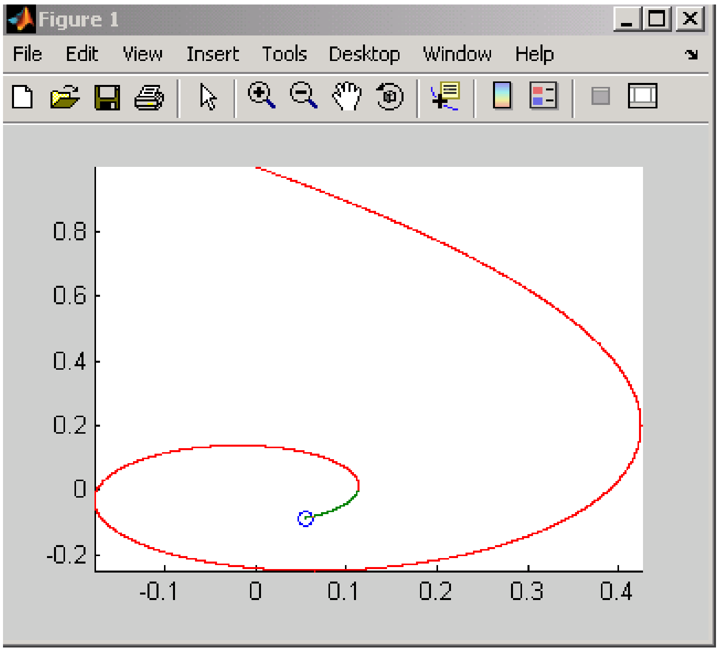



```
>> [Az,El]=view
Az =
  -37.5000
El =
    30
```

2.4. Анимированные графики

При изучении движения точки на плоскости или в пространстве полезно не только построить траекторию точки, но и следить за ее движением по этой траектории. Для построения анимированных графиков применяются функции `comet` и `comet3`.

```
>> t=[0:0.001:10];
>> x=sin(t)./(t+1);
>> y=cos(t)./(t+1);
>> comet(x,y)
```



2.5. Работа с несколькими графиками

Обычно графики выводятся в специальное графическое окно с заголовком **Figure 1**. При следующем построении графика новый график выводится в то же самое окно. MATLAB представляет возможность вывода каждого графика в свое окно, нескольких графиков в одно окно, в пределах одного окна нескольких графиков, каждого в своих осях.

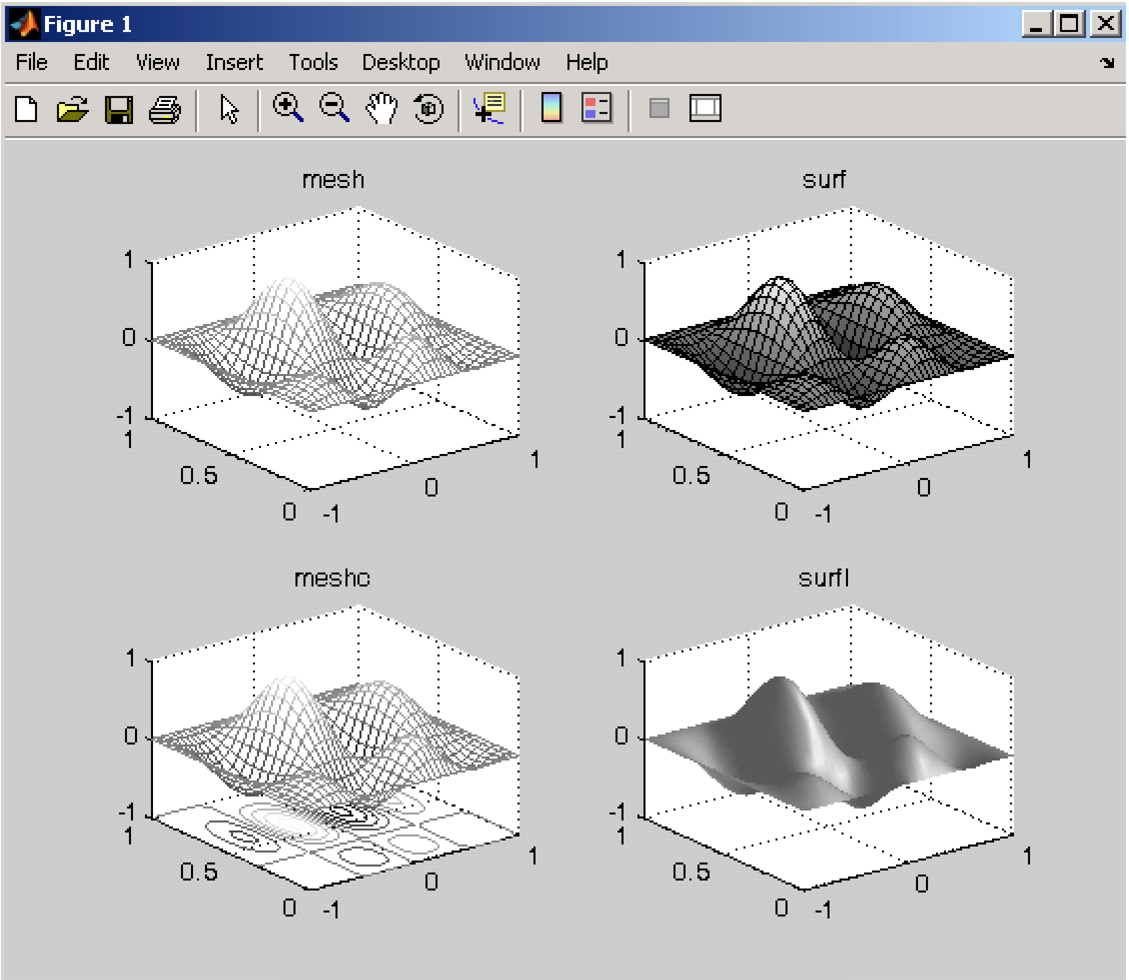
Следующая последовательность команд построит графики функций в отдельных окнах **Figure1** и **Figure2**, причем последнее окно будет текущим и все последующие изменения будут соответственно отражаться именно на этом окне. Чтобы сделать текущим другое окно, следует дважды щелкнуть по нему мышкой и вернуться к продолжению команд.

```
>> [X,Y]=meshgrid(-1:0.05:1,0:0.05:1);
>> Z=4*sin(2*pi*X).*cos(1.5*Y*pi).*(1-X.^2).*Y.*(1-Y);
>> figure
>> mesh(X, Y, Z)
>> figure
>> surfl(X, Y, Z)
```

MATLAB позволяет разместить в графическом окне несколько осей и вывести на них различные графики. Для это можно воспользоваться функцией subplot, которая располагает оси в виде матрицы с тремя параметрами: число подграфиков по вертикали, горизонтали, номер подграфика. Следующий пример иллюстрирует использование этой функции.

```
>> [X,Y]=meshgrid(-1:0.05:1,0:0.05:1);
>> Z=4*sin(2*pi*X).*cos(1.5*Y*pi).*(1-X.^2).*Y.*(1-Y);
>> subplot(2,2,1)
>> mesh(X, Y, Z)
>> title('mesh')
>> subplot(2,2,2)
```

```
>> surf(X, Y, Z)
>> title('surf')
>> subplot(2,2,3)
>> meshc(X, Y, Z)
>> title('meshc')
>> subplot(2,2,4)
>> surfl(X, Y, Z)
>> shading interp
>> title('surfl')
>> colormap (gray)
```



Методические указания для выполнения практических заданий
Лабораторной работы 5 «*Принципы создания приложений с графическим интерфейсом пользователя*»

Приложения MATLAB с графическим интерфейсом являются графическими окнами, содержащими элементы управления (кнопки, списки, переключатели, флаги, полосы скроллинга, области ввода, меню), а также оси и текстовые области для вывода результатов работы. Создание приложений включает следующие основные этапы – расположение нужных элементов интерфейса в пределах графического окна и программирование событий, которые возникают при обращении пользователя к данным объектам, – например, при нажатии кнопки. Процесс работы над приложением допускает постепенное добавление элементов в графическое окно, запуск, тестирование приложения и возврат в режим редактирования. Конечным результатом является программа с графическим интерфейсом пользователя (GUI), содержащаяся в одном или нескольких файлах, запуск которой производится указанием ее имени в командной строке или в другом приложении MATLAB. Элементы управления являются графическими объектами – потомками графического окна в иерархии объектов. Они могут быть созданы при помощи специальных функций низкоуровневой графики. Однако гораздо эффективнее воспользоваться визуальной средой GUIDE, которая позволяет создать заготовку окна приложения, разместить на ней элементы управления, запрограммировать события и их взаимосвязь.

3.1. Среда GUIDE

Перейдите в визуальную среду GUIDE, выполнив команду guide в командной строке. Появляется окно GUIDE Quick Start, изображенное на рис. 1, которое помогает настроить визуальную среду на создание приложения нужного типа или открыть существующее для продолжения работы. Шаблон для нового приложения выбирается в списке GUIDE templates на вкладке Create New GUI. Возможен один из следующих вариантов:
1) Blank GUI (Default) – пустое окно приложения без элементов управления, осей и меню;
2) GUI with Uicontrols – окно приложения со строками ввода, переключателями и кнопками;
3) GUI with Axes and Menu – окно приложения, содержащее оси, раскрывающийся список, кнопки и меню;
4) Modal Question Dialog – модальное диалоговое окно с кнопками Yes и No.

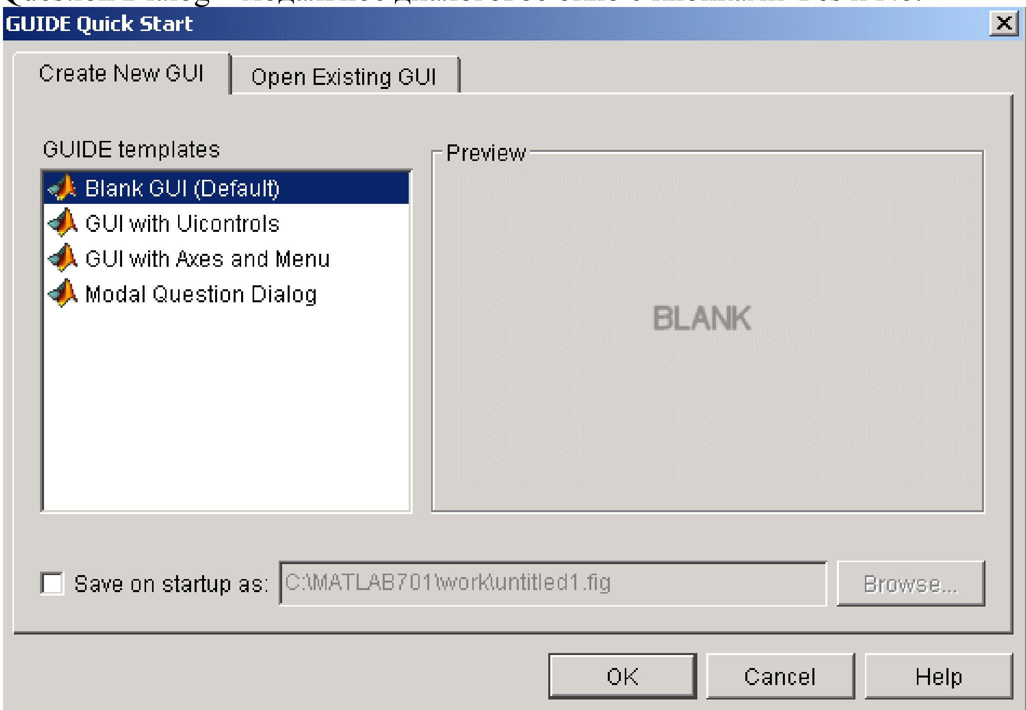


Рис. 1. Окно старта GUIDE.

Установка флага Save on startup as позволяет задать имя файла, в котором будет содержаться окно приложения.
Выберите опцию Blank GUI и нажмите кнопку OK. Появляется редактор среды GUIDE, заголовок которого untitled.fig означает, что в нем открыт новый файл, увеличьте немного размеры окна так, как изображено на рис. 2. Редактор среды GUIDE содержит: строку меню; панель инструментов управления приложением; заготовку окна приложения с нанесенной сеткой; панель инструментов для добавления элементов интерфейса на окно приложения.
Редактор позволяет разместить различные элементы интерфейса (рис. 3). Для этого требуется нажать соответствующую кнопку на панели инструментов и поместить выбранный объект щелчком мыши на заготовку окна приложения.

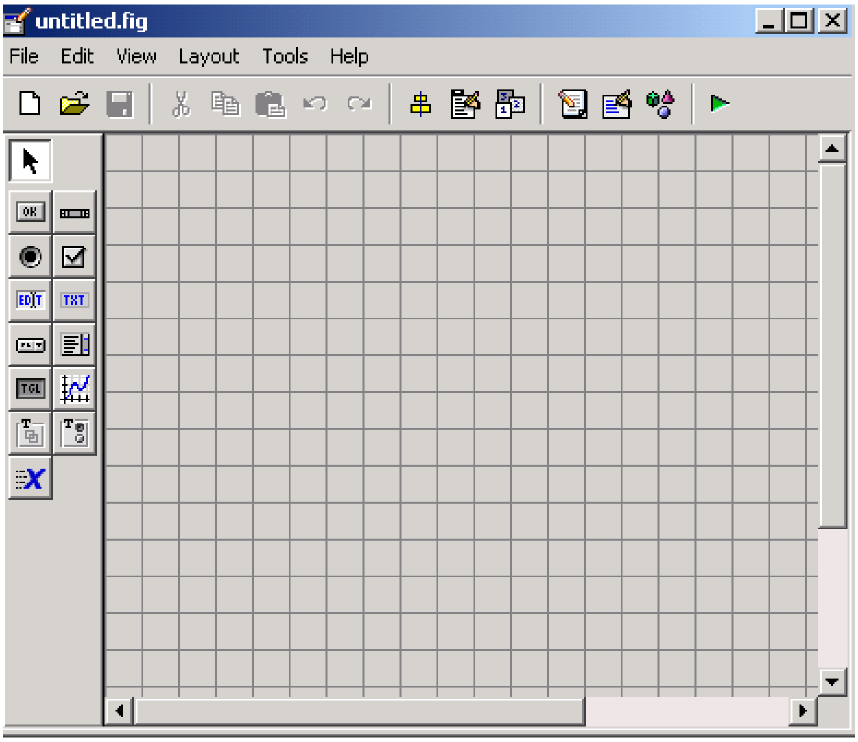


Рис. 2. Редактор среды GUIDE.

Другой способ состоит в задании прямоугольной области объекта перемещением мыши по области заготовки окна с удержанием левой кнопки. Размер и положение добавленных объектов изменяются при помощи мыши. Перед изменением размера следует выбрать режим выделения объектов и сделать объект текущим, щелкнув по нему кнопкой мыши. Названия элементов управления появляются на всплывающих подсказках при наведении курсора мыши на кнопки панели инструментов.

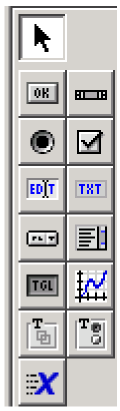


Рис. 3. Панель инструментов.

Приложение в данный момент находится в режиме редактирования. Любой объект можно удалить с окна, выделив его и нажав <Delete>. Запуск приложения производится при помощи кнопки Run. При нажатии на нее появляется диалоговое окно GUIDE, которое сообщает о том, что запуск приложения приведет к его сохранению. Приложение запускается в отдельном окне с заголовком mygui0. Пользователь может нажимать на кнопки, устанавливать флаги, переключатели, обращаться к спискам, – разумеется, при этом ничего полезного не происходит. Недостаточно разместить элементы интерфейса в окне приложения, следует позаботиться о том, чтобы каждый элемент выполнял нужные функции при обращении к нему пользователя.

3.2. Программирование событий

Приведем объяснения принципа программирования событий в среде GUIDE. Приложение в MATLAB хранится (по умолчанию) в двух файлах с расширением fig и m, первый из них содержит информацию о размещенных в окне приложения объектах, а второй является M-файлом с основной функцией и подфункциями. Добавление элемента интерфейса из редактора приложения приводит к автоматическому созданию соответствующей подфункции. Данную подфункцию следует наполнить содержимым – операторами, которые выполняют обработку события, возникающего при обращении пользователя к элементу интерфейса.

Приведем пример создания простого приложения, окно которого содержит оси и две кнопки, предназначенные для построения графика и очистки осей. Перейдите в среду GUIDE командой guide с пустым окном приложения так, как было описано в предыдущем разделе. Выберите в меню Tools редактора приложений пункт GUI Options..., появляется диалоговое окно GUI Options. Данное окно позволяет устанавливать некоторые общие свойства создаваемого приложения. Убедитесь, что в раскрывающемся списке Commandline accessibility текущей строкой является Callback (GUI becomes Current Figure within Callbacks), включен переключатель Generate FIG-file and Mfile и установлены все флаги, входящие в группу с ним.

Расположите на форме оси и кнопку так, как показано на рис. 4. На кнопке автоматически размещается надпись Push Button, а на осях – axes 1.

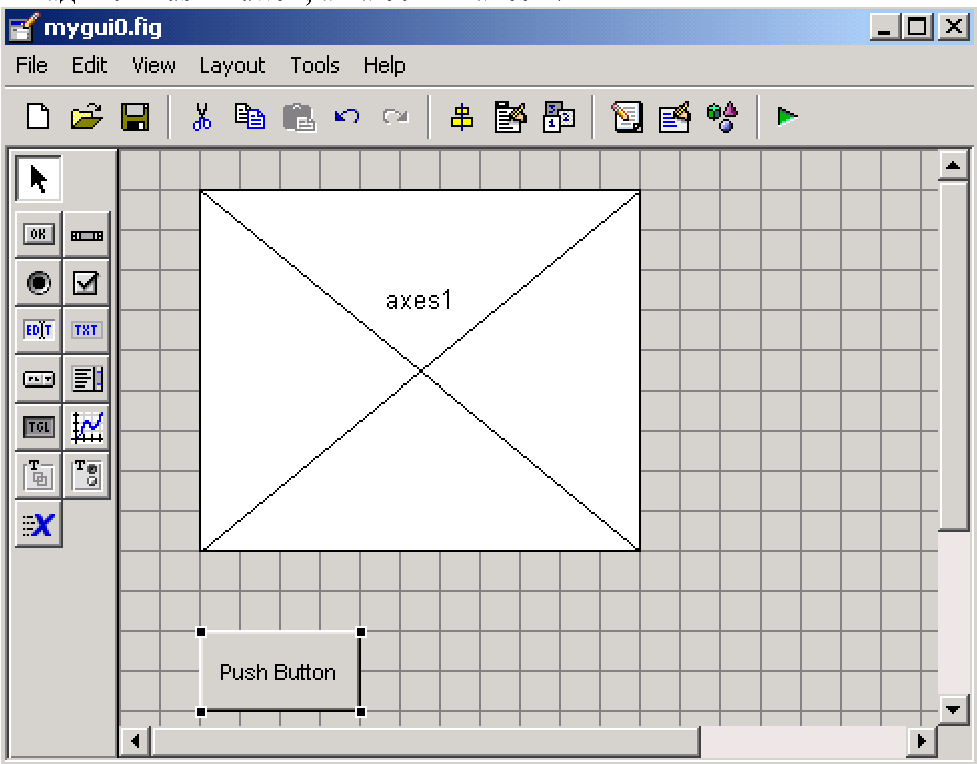


Рис. 4. Расположение кнопки и осей.

Следующий этап очень важный. Кнопка и оси являются элементами интерфейса, им следует дать имена, которые уникальным образом идентифицировали бы их среди всех объектов окна приложения и свидетельствовали об их назначении. Щелчком мыши выделите кнопку Push Button и вызовите инспектор свойств Property Inspector при помощи панели инструментов управления приложением (либо используя контекстное меню кнопки) (рис. 5).

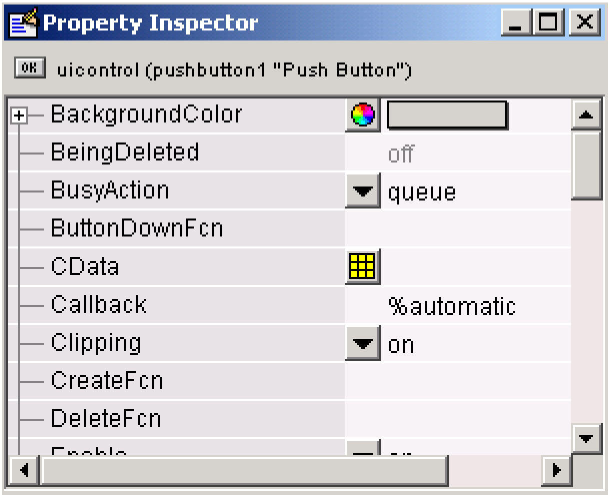


Рис. 5. Редактор свойств.

Появляется окно инспектора свойств, в котором содержится таблица названий свойств кнопки и их значений. Напомним, что элементы управления являются объектами UI Objects, которые принадлежат графическому окну согласно иерархии графических объектов MATLAB. Важнейшим свойством в данный момент является Tag, которое содержит тэг или имя объекта. При добавлении элементов управления им автоматически присваиваются имена, причем по умолчанию они состоят их типа объекта и номера (в нашем случае pushbutton1). Дадим кнопке информативное имя, которые говорит как о типе элемента, так и о его назначении. Эта кнопка будет служить для построения графика, что следует отразить в ее имени. Установите свойство Tag в значение btnPlot, для чего щелкните мышью по строке справа от названия свойства, наберите требуемое значение и нажмите <Enter>. Аналогичным образом установите тег осей в значение axMain. Следует различать имя объекта и надпись на нем. Сейчас кнопка с именем btnPlot содержит текст Push Button. Измените его, установив в инспекторе свойств Property Inspector свойство string в значении «построить». Теперь надпись на кнопке соответствует ее назначению.

При создании приложений с графическим интерфейсом вам придется часто прибегать к инспектору свойств Property Inspector для установки свойств объектов в подходящие значения. Не обязательно каждый раз делать объект текущим, а затем открывать окно инспектора свойств Property Inspector при помощи соответствующей кнопки на панели инструментов среды GUIDE.

Сохраните приложение, для чего выберите в меню File среды GUIDE пункт Save as, создайте папку MyFirstGui (например, в текущем каталоге) и дайте имя mygui0.fig окну приложения. Обратите внимание, что открылся редактор М-файлов, содержащий сопутствующий файл mygui0.m.

Основная функция mygui0 предназначена для инициализации приложения, поэтому редактировать ее не следует. Обратимся к функции btnPlotCallback. Приложение mygui0 содержит одну кнопку «Построить» с именем btnPlot. Когда пользователь нажимает на нее в работающем приложении, то генерируется событие Callback данного элемента управления. При этом вызывается подфункция btnPlotCallback. Заметьте, что имя подфункции образовано именем (тегом) кнопки и названием события. Данная подфункция не содержит операторов, и при нажатии на кнопку ничего не происходит. Завершающий этап состоит в задании действий, которые выполняются при нажатии пользователем на кнопку «Построить». Запрограммируйте подфункцию btnPlot_Callback обработки события Callback кнопки «Построить» с именем btnPlot в соответствии с листингом (автоматически созданные комментарии убирать не обязательно).

```
function btnPlot_Callback(hObject, eventdata, handles)
% hObject    handle to btnPlot (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
x=-2:0.2:2;
y=exp(-x.^2);
plot(x,y)
```

В дальнейшем мы будем говорить, что требуется запрограммировать событие Callback некоторого объекта без указания имени подфункции и подробного описания самого объекта. Перейдем теперь к запуску приложения mygui0. Нажмите кнопку Run на панели инструментов среды GUIDE. Поскольку приложение сохранено в отдельном каталоге MyFirstGui, то появляющееся диалоговое окно предлагает сделать этот каталог текущим. Убедитесь, что выбран переключатель Change MATLAB current directory, и нажмите OK. Теперь каталог MyFirstGui стал текущим, и при последующих запусках приложения это окно выводиться не будет (при условии, если вы не измените текущий каталог из рабочей среды). Перед запуском может появиться еще одно диалоговое окно, в котором следует подтвердить сохранение файлов с расширениями fig и m, содержащих приложение. После выполнения описанных действий на экране отображается окно приложения mygui0. Нажатие на кнопку «Построить» приводит к отображению графика функции на осях. Следующим шагом будет добавление кнопки, предназначенной для очистки осей.

Завершите приложение при помощи кнопки закрытия окна в правом верхнем углу и продолжите работу над mygui0 в среде GIUDE. Добавьте кнопку, задайте в редакторе свойств имя btnClear и надпись «Очистить». Осталось запрограммировать событие Callback этой кнопки. Для быстрого перехода к соответствующей подфункции btnClearCallback воспользуйтесь всплывающим меню. Становится активным окно редактора М-файлов, причем заголовок нужной подфункции выделен. Разместите единственный оператор очистки осей cla в подфункции:

```
function btnClear_Callback(hObject, eventdata, handles)
% hObject    handle to btnClear (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla
```

Запустите приложение и убедитесь, что названия кнопок соответствуют выполняемым действиям.

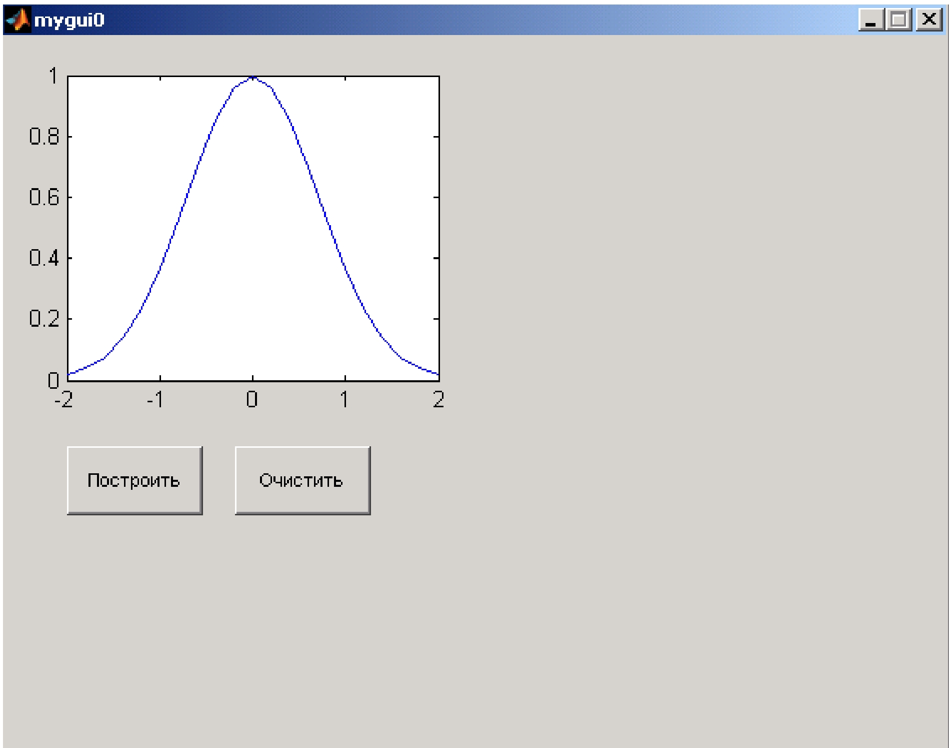


Рис. 6. Вид формы окна после запуска приложения.

3.3. Управление свойствами объектов

Разработка приложения связана с изменением свойств объектов, которые они получают по умолчанию при размещении их на заготовке окна. Элементы управления являются графическими объектами Uicontrol, и полная информация о назначении их свойств доступна в браузере свойств графических объектов справочной системы MATLAB.

Некоторые из свойств, – например, размер кнопки, надпись на ней и свойства шрифта, устанавливаются при создании объекта в режиме редактирования и изменять их, как правило, не требуется. Ряд свойств объектов необходимо устанавливать программно, прямо в ходе работы приложения для обеспечения согласованного поведения элементов управления.

Приступим к усовершенствованию приложения mygui0. Пусть необходимо, чтобы при запуске доступной являлась только кнопка «Построить», а при нажатии на «Построить» выводился график и она становилась недоступной, зато появлялась бы возможность нажать кнопку «Очистить» для удаления графика и начать все сначала. Итак, всегда доступна только одна из кнопок в зависимости от состояния осей. Решение поставленной задачи требует привлечения свойства объекта Enable, которое отвечает за возможность доступа к нему пользователем, значение ‘on’ разрешает доступ, а ‘off’, соответственно, запрещает.

Поскольку элементы интерфейса являются графическими объектами, то для изменения их свойств следует прибегнуть к функции set. В данном случае изменяется значение только одного свойства, поэтому функция set должна быть вызвана с тремя входными аргументами – указателем на объект, строкой с названием свойства и его значением. Заметьте, что свойства одного объекта должны изменяться в подфункции обработки события Callback другого объекта. Следовательно, надо иметь возможность доступа к указателю на любой существующий объект. Требуемые указатели содержат входные аргументы hObject и handles подфункций, которые обрабатывают события элементов управления. В hObject хранится указатель на тот объект, событие которого обрабатывается в данный момент, а handles является структурой с указателями на все объекты. Имена полей структуры совпадают со значениями свойств Tag существующих элементов интерфейса. Например, handles.btnPlot является указателем на кнопку «Построить» с именем btnPlot. Доступ к кнопке «Очистить» должен быть запрещен в начале работы приложения, пока пользователь не нажмет «Построить». Установите в инспекторе свойств для кнопки «Очистить» Enable в off, используйте кнопку со стрелкой в строке со значением свойства. Остальные изменения значения Enable кнопок должны происходить в ходе работы приложения. Для разрешения и запрещения доступа к кнопкам нужно внести дополнения в обработку их событий Callback.

В подфункцию обработки события Callback кнопки «Построить» добавьте:

1) установку свойства Enable кнопки «Очистить» в ‘on’ (после вывода графика следует разрешить доступ к ней);

2) установку свойства Enable кнопки «Построить» в ‘off’ (после вывода графика она должна стать недоступной).

Аналогичные изменения произведите в обработке события Callback кнопки «Очистить», а именно:

1) установку свойства Enable кнопки «Построить» в ‘on’ (после очистки осей следует разрешить доступ к ней);

2) установку свойства Enable кнопки «Очистить» в ‘off’ (после очистки осей она должна стать недоступной).

Подфункции btnPlot_Callback и btnClear_Callback должны быть запрограммированы следующим образом:

```
function btnPlot_Callback(hObject, eventdata, handles)
x=-2:0.2:2;
y=exp(-x.^2);
plot(x,y)
set(hObject,'Enable','off')
set(handles.btnClear,'Enable','on')
```

```
function btnClear_Callback(hObject, eventdata, handles)
cla
set(hObject,'Enable','off')
set(handles.btnPlot,'Enable','on')
```

Сохраните изменения в редакторе М-файлов. Запустите приложение mygui0 и убедитесь, что всегда доступной является только одна из кнопок «Построить» или «Очистить» – это является хорошей подсказкой для пользователя о возможных действиях. Закройте окно приложения и редактор приложений.

3.4. Запуск приложения и его редактирование

Запуск приложения осуществляется не только из редактора приложений. Возникает вопрос, как работать с уже созданным приложением с графическим интерфейсом и, возможно, вносить в него требуемые изменения. Для запуска приложения достаточно в качестве команды задать его имя в командной строке >> mygui0

Появляется окно приложения; обращение к элементам интерфейса окна приводит к соответствующим действиям. В любой момент можно продолжить редактирование. Перейдите в режим редактирования приложения mygui0. Измените название окна приложения на «Визуализация функций». Заголовок окна задается свойством Name графического окна. Установите в инспекторе свойств Property Inspector свойство Name окна приложения в «Визуализация функций». Запустите mygui0 и убедитесь, что приложение имеет нужный заголовок.

Желательно располагать элементы интерфейса в порядке, обеспечивающем удобную работу пользователя с приложением. Приложение хорошо выглядит, если однотипные элементы, – например, кнопки, флаги и т.д., – определенным образом выровнены в окне приложения. Среда GUIDE предоставляет разработчику приложений несколько способов выравнивания добавляемых объектов – сетку, линейку и специальные инструменты.

3.5. Размеры объектов и их выравнивание

Редактор приложений содержит заготовку окна приложения, размер которого изменяется при помощи мыши. Для задания точных размеров и положения окна запущенного приложения выберите в инспекторе свойств Property Inspector подходящие единицы измерения (пикселы, сантиметры и т.д.), обратившись к свойству Unit, и установите свойство Position в нужное значение.

На окно нанесена сетка с достаточно крупным шагом (по умолчанию). Аккуратное расположение большого числа элементов управления требует точного задания их положения. Поскольку элементы управления Uicontrol являются графическими объектами (потомками графических окон), то естественно ожидать, что среди их свойств есть предназначенные для задания их размеров и положения в графическом окне. Эти свойства называются Position и Unit, причем значением Position должен быть вектор из четырех чисел (расстояние от левого и нижнего края окна, ширина и высота) в единицах измерения, установленных в Unit. Элементы управления и оси могут быть сгруппированы и размещены на некоторой панели – графическом объекте Uipanel. В этом случае они являются потомками объекта Uipanel и их положение задается относительно этой панели. Сама панель есть потомок графического окна, поэтому значение свойства Position объекта Uipanel определяет ее положение в пределах окна. Предком для элементов управления может быть еще одна панель Uibuttongroup, которая предназначена для обеспечения согласованного поведения группы переключателей или кнопок-переключателей – когда может быть включен только один элемент управления из всей группы. Для добавления панелей Uipanel и Uibuttongroup в окно приложения служат, соответственно, кнопки Panel и Button Group редактора среды GUIDE. Используем эти панели для размещения переключателей и флагов в окне Приложения mygui0. Значения свойств Position и Units могут быть установлены в инспекторе свойств Property Inspector при размещении элементов управления и панелей в окне приложения или в подфункциях обработки событий, если работающее приложение должно изменить вид окна. Для аккуратного расположения элементов управления и панелей на этапе проектирования интерфейса можно избрать и более простой способ, чем установка свойству Position подходящего значения. Он состоит в использовании вспомогательных средств среды GUIDE: сетки, линеек и инструментов выравнивания. Для отображения линеек и доступа к свойствам сетки следует в меню Tools выбрать пункт Grid and Rules**.** Появляется диалоговое окно Grid and Rulers, изображенное на рис. 7.

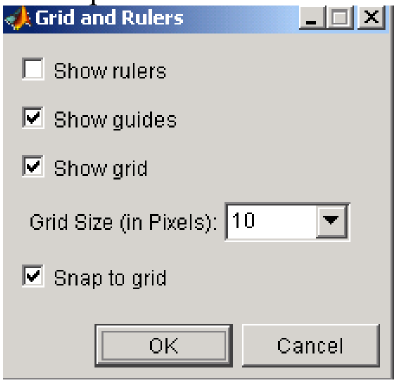


Рис. 7. Диалоговое окно.

3.6. Программирование элементов интерфейса

3.6.1. Флаги, рамки

Флаги позволяют произвести одну или несколько установок, определяющих ход работы приложения. Продолжите работу над mygui0, предоставьте пользователю возможность наносить линии сетки на график. Окно приложения должно содержать два флага с названиями «сетка по x» и «сетка по y». Если пользователь нажимает кнопку «Построить», то не только строится график функции, но и на оси наносится сетка по выбранным координатам. Нажатие на «Очистить» должно приводить к исчезновению графика функции и скрытию сетки.

Обычно несколько элементов управления со схожим назначением группируются и помещаются на некоторой панели. Зачастую применяются панели двух типов: Uipanel и Uibuttongroup. В данном случае подойдет обычная панель Uipanel, поскольку флаги являются независимыми и могут быть включены или сброшены по отдельности.

Измените размеры осей, освободив справа место для рамки. Нанесите панель на окно приложения при помощи соответствующей кнопки. По умолчанию имя панели uipanel 1, а ее заголовок Panel. В инспекторе свойств измените заголовок на «Сетка», воспользовавшись свойством Title объекта Uipanel. За положение имени панели отвечает свойство TitiePosition, которое по умолчанию имеет значение ‘lefttop’, т.е. в левом верхнем углу панели. Изучите остальные возможные варианты самостоятельно, обратившись к инспектору свойств Property Inspector или браузеру свойств графических объектов в справочной системе MATLAB.

На панель добавьте два флага так, как показано на рис. 8.

Разместите поясняющие подписи рядом с флагами и дайте им имена. Задайте свойству Tag верхнего флага значение chbxGridX, а свойству string, отвечающему за подпись флага, – значение «сетка по x». Аналогичным образом определите свойства нижнего флага, установите свойство Tag в chbxGridY и string в – «сетка по y». Если текст не помещается рядом с флагом, увеличьте ширину области флага при помощи мыши.

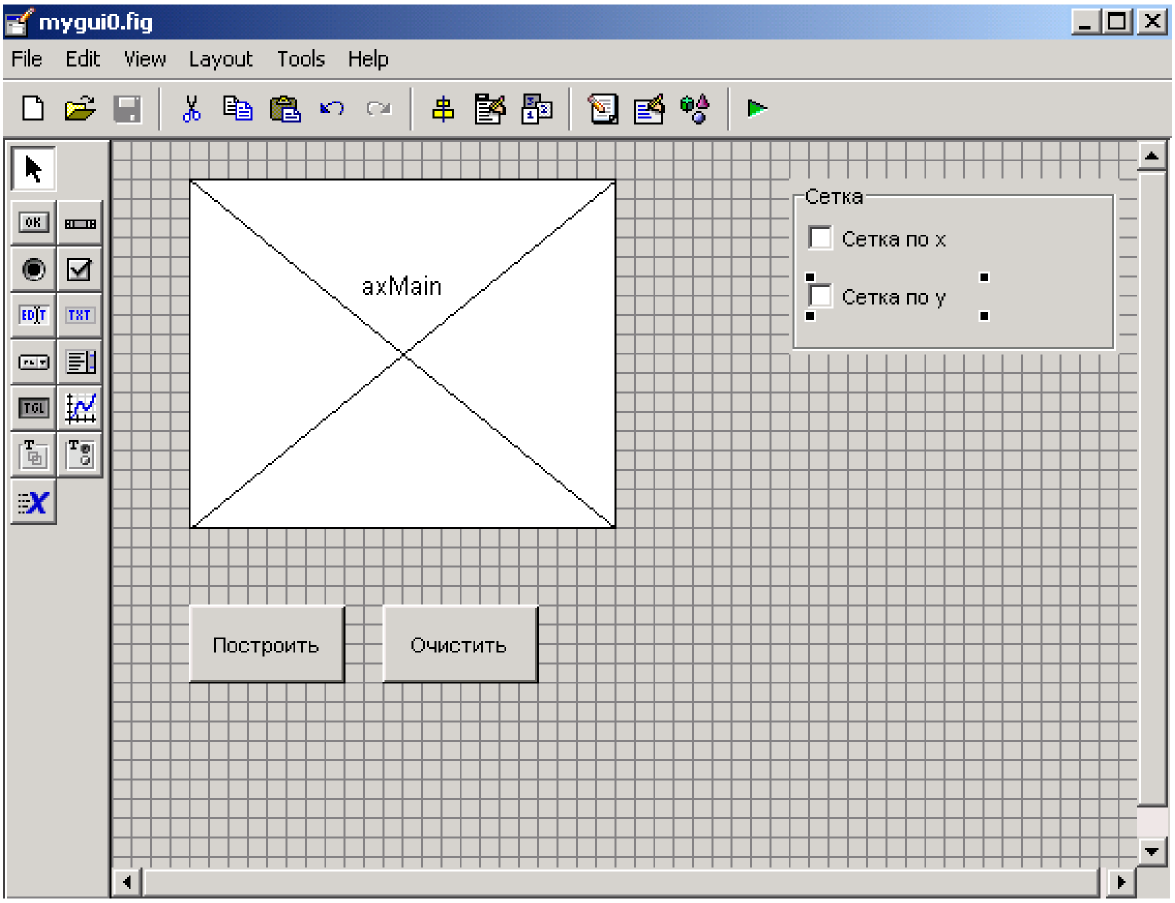


Рис. 8. Форма приложения.

Осталось сделать так, чтобы при нажатии пользователем кнопки «Построить» происходило отображение линий сетки в зависимости от установленных флагов, а нажатие на «Очистить» приводило к скрытию сетки. Блок обработки события Callback кнопки «Построить» следует дополнить проверкой состояния флагов. Свойство флага value принимает значение 1 при установке флага пользователем и, соответственно, равно нулю, если флаг сброшен.

Значение свойства графического объекта позволяет получить функция get, первым входным аргументом которой должен быть указатель на объект, а вторым – название свойства. Как и в случае согласования кнопок «Построить» и «Очистить», в подфункции обработки события Callback одного объекта (кнопки) мы должны получить указатель на другой объект (флаг). Для этого следует обратиться к полям chbxGridX и chbxGridY Структуры handles и в зависимости от их значений установить свойства осей X Grid и Y Grid в ‘on’ и ‘off’.

Произведите необходимые изменения в подфункции обработки события Callback кнопки «Построить»:

```
function btnPlot_Callback(hObject, eventdata, handles)
%Построение графика функции
x=-2:0.2:2;
y=exp(-x.^2);
plot(x,y)
%Проверка флага сетка по x
if get (handles.chbxGridX,'Value')
    %Флаг включен, добавить линии сетки
    set(gca,'XGrid','on')
else
     %Флаг выключен, убрать линии сетки
      set(gca,'XGrid','off')
end
%Проверка флага сетка по y
```

```
if get (handles.chbxGridX,'Value')
    %Флаг включен, добавить линии сетки
    set(gca,'YGrid','on')
else
     %Флаг выключен, убрать линии сетки
      set(gca,'YGrid','off')
end
%Кнопка "Построить" должна стать недоступной после вывода графика
set(hObject,'Enable','off')
%Кнопка "Очистить" должна стать доступной
set(handles.btnClear,'Enable','on')
```

Запустите приложение mygui0 и убедитесь, что установка флагов влияет на отображение сетки при нажатии на кнопку «Построить»**.** Смена состояния флагов сетки не приводит к немедленным изменениям на графике. Пользователь должен перестроить график, нажимая последовательно кнопки «Очистить» и «Построить». Для немедленного реагирования приложения на состояние флагов следует определить их события Callback. Программирование данных событий заключается в проверке состояния флага и отображении или скрытии соответствующих линий сетки. Сделайте текущим флаг «сетка по x» в редакторе приложений и перейдите к подфункции chbxGridX_Callback при помощи всплывающего меню данного объекта. Запрограммируйте событие Callback флага. Используйте аргумент hObject соответствующих подфункций, содержащий указатель на объект, событие которого обрабатывается в текущий момент времени. Аналогичным образом обработайте событие Callback второго флага «сетка по y»:

```
function chbxGridY_Callback(hObject, eventdata, handles)
if get (hObject,'Value')
    %Флаг включен, добавить линии сетки
    set(gca,'YGrid','on')
else
     %Флаг выключен, убрать линии сетки
      set(gca,'YGrid','off')
end

function chbxGridX_Callback(hObject, eventdata, handles)
if get (hObject,'Value')
    %Флаг включен, добавить линии сетки
    set(gca,'XGrid','on')
else
     %Флаг выключен, убрать линии сетки
      set(gca,'XGrid','off')
end
```

Модифицированные подфункции позволяют пользователю наносить и убирать сетку по каждой координате при помощи флагов без перестроения графика функции.
Самостоятельно внесите дополнения в подфункцию обработки события Callback кнопки «Очистить», обеспечивающие удаление не только графика функции, но и линий сетки.
Флаги предоставляют пользователю возможность выбора одной или сразу нескольких опций. Одновременный выбор *только одной* опции осуществляется при помощи переключателей.

3.6.2. Переключатели

Модернизируйте интерфейс приложения mygui0, предоставьте пользователю возможность выбирать тип маркера (кружок, квадрат или отсутствие маркера) при помощи трех переключателей. Переключатели обычно группируются по их предназначению, и пользователь может выбрать только одну опцию, т. е. всегда установлен единственный переключатель из группы. Для обеспечения такой согласованной работы переключателей служит панель Uibuttongroup. Добавьте в окно приложения панель Uibuttongroup, воспользовавшись кнопкой Button Group среды GUIDE. Задайте имя «Тип маркера» для этой панели аналогично тому, как вы определяли имя обычной панели при размещении флагов. Нанесите на нее три переключателя. Установите их свойствам Tag Значения rbMarkCirc, rbMarkSq, rbMarkNone, а String – круги, квадраты, без маркеров соответственно (рис. 9).

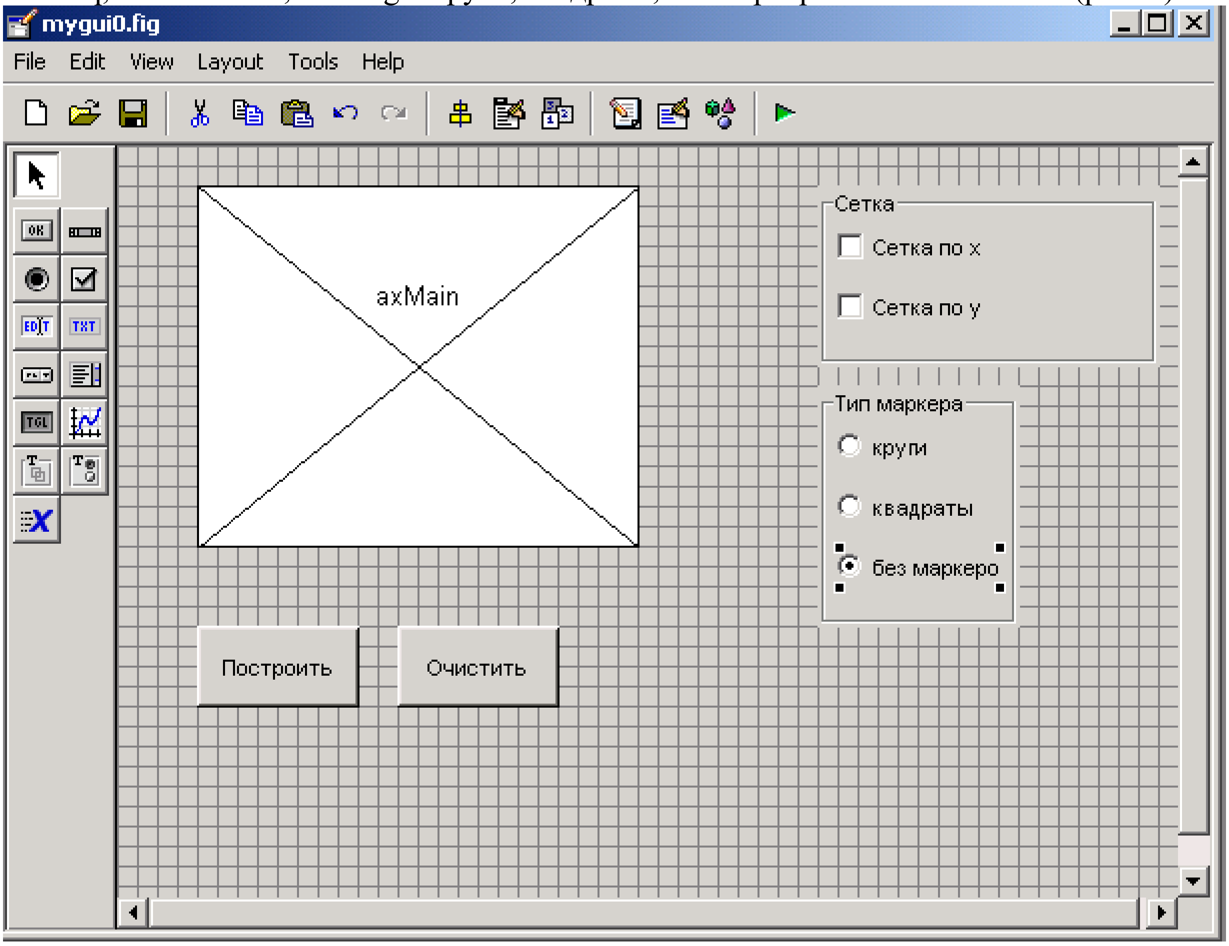


Рис. 9. Форма приложения.

Состояние переключателя, так же как и флага, определяется его свойствами value, Max и Min, причем по умолчанию Max установлено в 1, а Min – в 0. Для включенного переключателя совпадают значения value и max, а для выключенного – Value и Min. Задайте в инспекторе свойств Property Inspector значение 1 свойству value переключателя с надписью «без маркеров», он будет включен при запуске программы. Значение свойства value устанавливается следующим образом.

Выделите переключатель и отобразите его свойства в Property Inspector, нажмите кнопку в строке с value. Появляется окно Value, изображенное на рис. 10. Выделите при помощи мыши строку со значением 0.0 и перейдите в режим редактирования значения двойным щелчком мыши. Измените 0.0 на 1 и нажмите OK.

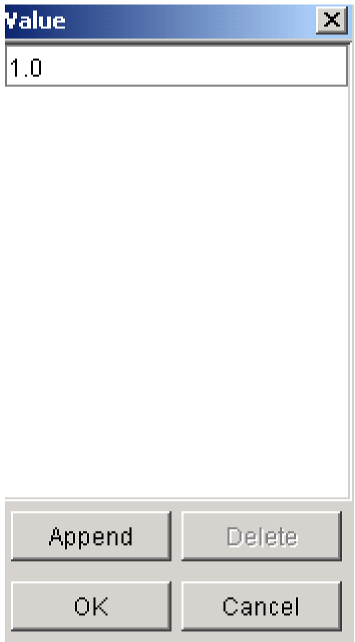


Рис. 10. Окно Value.

Обратите внимание, что в инспекторе свойств Property Inspector значение value изменилось на единицу и включился переключатель «без маркеров» на заготовке окна приложения. Остальные переключатели панели «Тип маркера» выключены. Предположим, что пользователь установил один из переключателей. Происходит обращение к соответствующей подфункции обработки события Callback переключателя, которая должна изменить тип маркеров линии. Обсудим программирование этого события, поскольку оно связано с решением общей проблемы – обменом данными между подфункциями.

Изменение типа маркеров линии не представляет труда – если известен указатель на линию, то достаточно обратиться к свойству линии Marker при помощи функции set и установить его в нужное значение. Указатель на линию возвращает функция plot в выходном аргументе, его следует записать в некоторую переменную, – например, Line. Использовать указатель на линию придется в других подфункциях, обрабатывающих событие Callback переключателей. Очень важно понять, что переменная Line инициализируется при вызове подфункции btnPlot_Callback и после ее завершения становится недоступной, поскольку все переменные, определенные в подфункции, являются локальными и по окончании работы подфункции *не определены*. Обмен данных между подфункциями проще всего осуществить при помощи структуры handles. Подфункция, передающая данные, должна содержать запись данных в новое поле и сохранение структуры функцией guidata. Тогда входной аргумент – структура handles всех подфункций будет содержать добавленное поле, в которое занесено соответствующее значение. В нашем случае данные, подлежащие сохранению, являются указателем на линию, созданную в результате работы подфункции btnPlot_Callback. Внесите необходимые изменения в подфункцию btnPlotCallback и запрограммируйте обработку событий callback переключателей в подфункциях rbMarkCirc_Callback, rbMarkSq_Callback, rbMarkNone_Callback:

```
function btnPlot_Callback(hObject, eventdata, handles)
% hObject    handle to btnPlot (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Построение графика функции
x=-2:0.2:2;
y=exp(-x.^2);
%Запись указателя в поле line структуры handles
handles.Line=plot(x,y);
%Сохранение структуры handles для использования в других подфункциях
```

```
guidata(gcbo,handles);

function varargout=rbMarkCirc_Callback(hObject, eventdata, handles)
% Выбран переключатель маркеры-круги
set(handles.Line,'Marker','o'); % Размещение маркеров-кругов на линии

function varargout=rbMarkSq_Callback(hObject, eventdata, handles)
% Выбран переключатель маркеры-квадраты
set(handles.Line,'Marker','s'); % Размещение маркеров-квадратов на линии

function varargout=rbMarkNone_Callback(hObject, eventdata, handles)
% Выбран переключатель без маркеров
set(handles.Line,'Marker','none'); % удаление маркеров с линии
```

Запустите приложение mygui, отобразите график функции, нажав «Построить», и убедитесь в том, что возможна установка только одного из переключателей и она приводит к появлению соответствующих маркеров на графике функции или их удалению.

3.6.3. Списки

Модернизируйте интерфейс приложения mygui0, предоставьте пользователю возможность выбора цвета линии графика из раскрывающегося списка (синий, красный, зеленый). Перейдите в режим редактирования и добавьте при помощи панели управления раскрывающийся список. Установите в инспекторе свойств Property Inspector Tag в значение pmcolor.

Элементами раскрывающегося списка являются строки, которые вводятся в инспекторе свойств Property Inspector. Нажмите кнопку в строке со свойством string раскрывающегося списка – появляется окно String. В области ввода текста замените текст "Pop-up Menu" на строки «синий», «красный», «зеленый» (без кавычек), разделяя их при помощи <Enter> и нажмите ОК. Запустите mygui и убедитесь, что раскрывающийся список содержит требуемые строки.

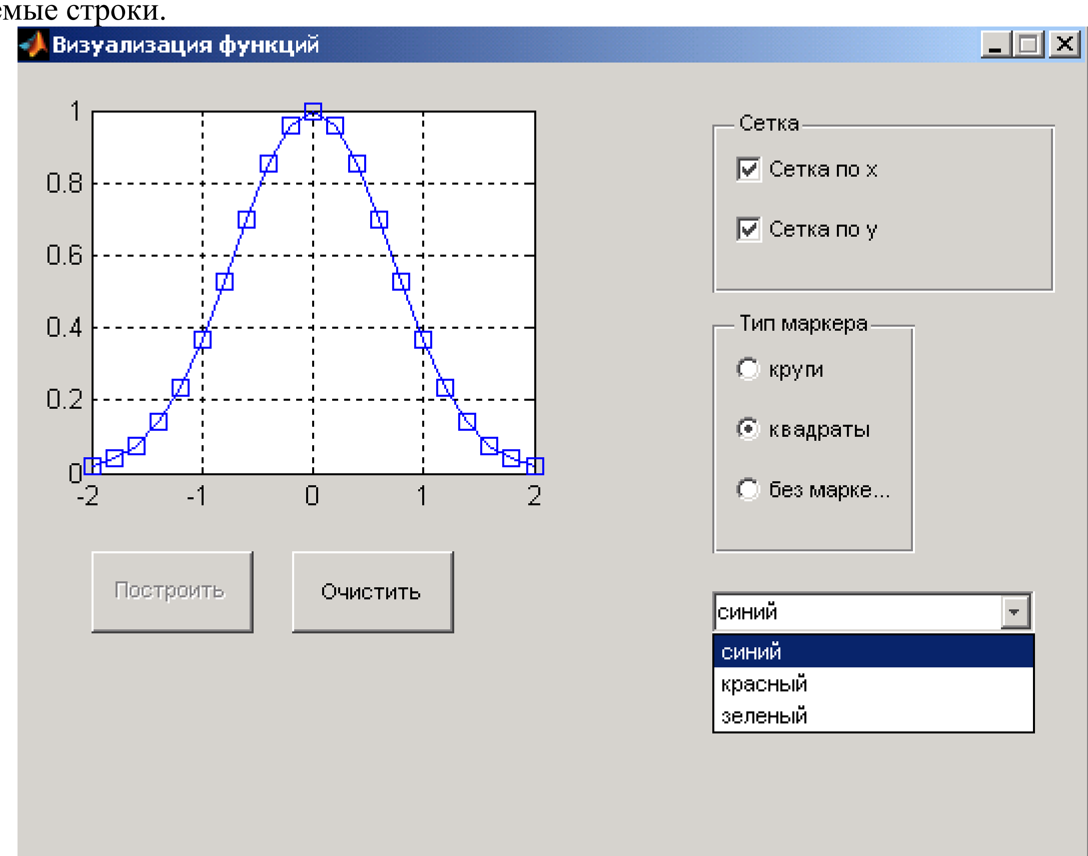


Рис. 11. Запуск приложения.

Выбор различных строк пока не приводит к изменению цвета линии – требуется запрограммировать событие callback раскрывающегося списка. Обработка события Callback раскрывающегося списка состоит в определении выбора пользователя и соответствующем изменении цвета линии. Свойство списка value содержит номер выбранной строки, которые нумеруются с единицы. Перейдите к подфункции pmColor_Callback и запрограммируйте обработку выбора пользователя. Примените оператор switch для установки цвета линии в зависимости от номера выбранной строки списка.

```
function pmColor_Callback(hObject, eventdata, handles)
%Определение номера выбранной строки
Num=get(hObject,'Value');
switch Num
    case 1
        %Выбран синий цвет линии
        set(handles.Line,'Color','b');
    case 2
         %Выбран красный цвет линии
        set(handles.Line,'Color','r');
    case 3
         %Выбран зеленый цвет линии
        set(handles.Line,'Color','g');
end
```

Запустите приложение, постройте график, нажав на «Построить», и убедитесь в том, что раскрывающийся список позволяет изменять цвет линии графика функции.

3.6.4. Полосы скроллинга

Усовершенствуйте интерфейс приложения mygui0, предоставив пользователю возможность устанавливать ширину линии при помощи полосы скроллинга. Добавьте полосу скроллинга (Slider) в окно приложения и задайте ей имя scrWidth в свойстве Tag. Снабдите полосу скроллинга текстовым пояснением «Толщина линии», как и раскрывающийся список.

Теперь следует определить соответствие между положением бегунка полосы и числовым значением ее свойства value.

Проделайте следующие установки из редактора свойств.

1. В мах занесите десять, а в Min – единицу. Свойства мах и Min полосы скроллинга отвечают за границы значений, записываемых в value, при перемещении бегунка.
2. Определите начальное положение, записав в value единицу. Нажмите кнопку в строке с названием «Свойства», и в появившемся окне Value измените значение на единицу.
3. Обратитесь к свойству SliderStep. Его значением является вектор из двух компонент, первая из которых определяет относительное изменение value при нажатии на кнопки со стрелками полосы скроллинга, а вторая – при перетаскивании бегунка мышью.

Для того чтобы нажатие на кнопки полосы изменяло value на десять процентов, а щелчок мыши справа или слева от бегунка –на двадцать, следует установить значение [0.1 0.2] свойства sliderstep. Раскройте строку sliderstep щелчком мыши по знаку плюс слева от названия свойства и в появившихся строках x и y введите 0.1 и 0.2. Осталось запрограммировать событие Callback полосы скроллинга с именем scrWidth, которое состоит в задании ширины линии, равной округленному значению value. Перейдите к подфункции scrWidth_callback и добавьте в ней оператор установки ширины линии:

```
function scrWidth_Callback(hObject, eventdata, handles)
% Получение ширины линии в зависимости от положения бегунка
%           на полосе скроллинга
width=get(hObject,'Value');
% установка толщины линии
set(handles.Line,'LineWidth',round(width))
```

Запустите mygui0 и убедитесь, что полоса скроллинга позволяет легко изменять толщину линии построенного графика (рис. 12).

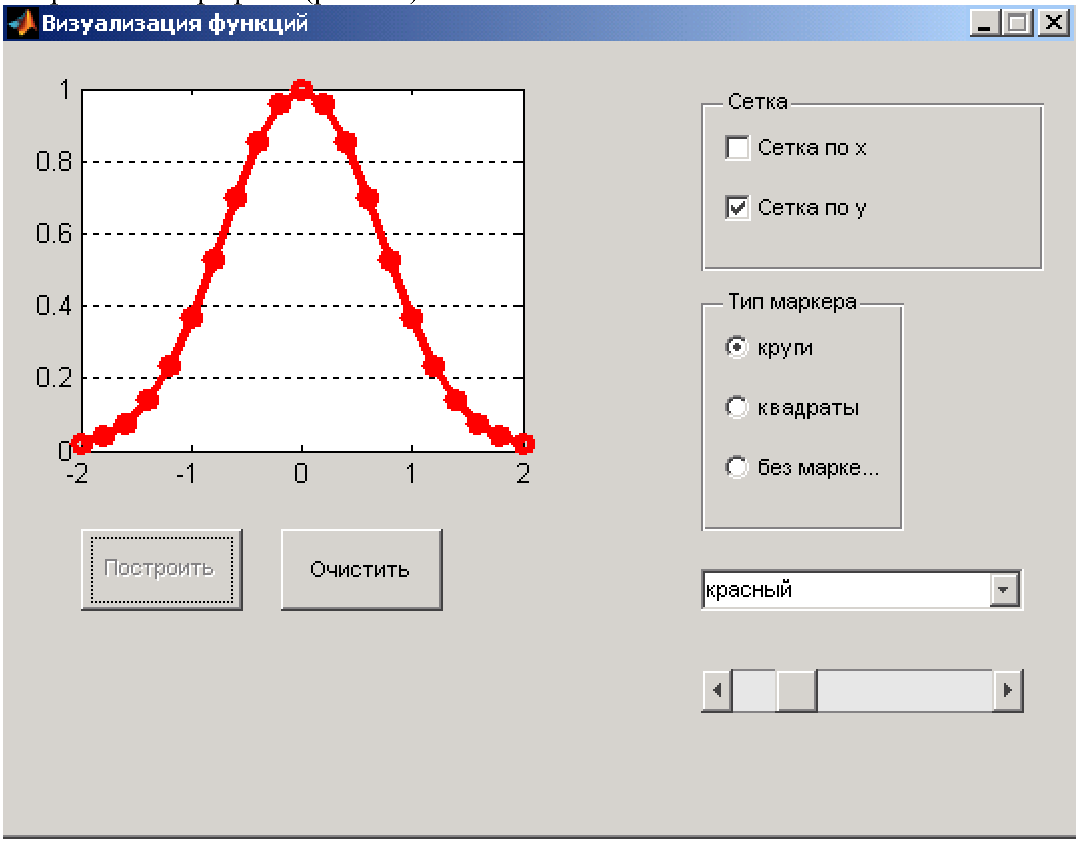


Рис. 12. Запуск приложения.

3.6.5. Область ввода текста

Обычные текстовые области (Static Text) позволяют лишь вывести некоторый текст в окно приложения. Обмен текстовой информацией между пользователем и приложением осуществляется при помощи областей ввода текста. Предоставьте пользователю возможность ввести выражение для функции, график которой требуется построить.

Добавьте в окно приложения область ввода текста, установите ее свойство Tag в значение editFun и снабдите ее пояснением в текстовой области, расположенной выше – как показано на рис. 13. При запуске приложения mygui строка для ввода функции должна быть пустой, поэтому в редакторе свойств удалите из String строку Edit Text.

Набранная строка в области ввода должна содержать выражение для функции с учетом правил записи поэлементных операций. Содержимое этой строки является значением свойства string области ввода, его надо будет преобразовать в inline-функцию при программировании события кнопки «Построить» и вычислить вектор значений функции.

С другой стороны, если курсор находится внутри области ввода, то нажатие <Enter> вызовет возникновение ее события CallBack. Разумно предусмотреть, чтобы в этом случае построился график функции и произошли все события, запрограммированные в подфункции обработки события Callback кнопки «Построить». Для этого вовсе не требуется копировать операторы подфункции btnPlot_Callback в подфункцию editFun_CallBack, можно просто вызвать btnPlot_Callback. Первым ее входным аргументом должен быть указатель на кнопку «Построить», поскольку hObject является указателем на тот объект, событие которого обрабатывается, а в данном случае мы программно моделируем нажатие кнопки пользователем. Листинг содержит операторы для построения графика функции, введенной пользователем:

```
function btnPlot_Callback(hObject, eventdata, handles)
%Построение графика функции
```

```
x=-2:0.2:2;
funstr=get(handles.editFun,'String');
fun=inline(funstr);
y=fun(x);
%y=exp(-x.^2);
%Запись указателя в поле line структуры handles
handles.Line=plot(x,y);
…….
function editFun_Callback(hObject, eventdata, handles)
btnPlot_Callback(handles.btnPlot,eventdata,handles)
```

Запуск mygui0 иллюстрируется на рис. 13.

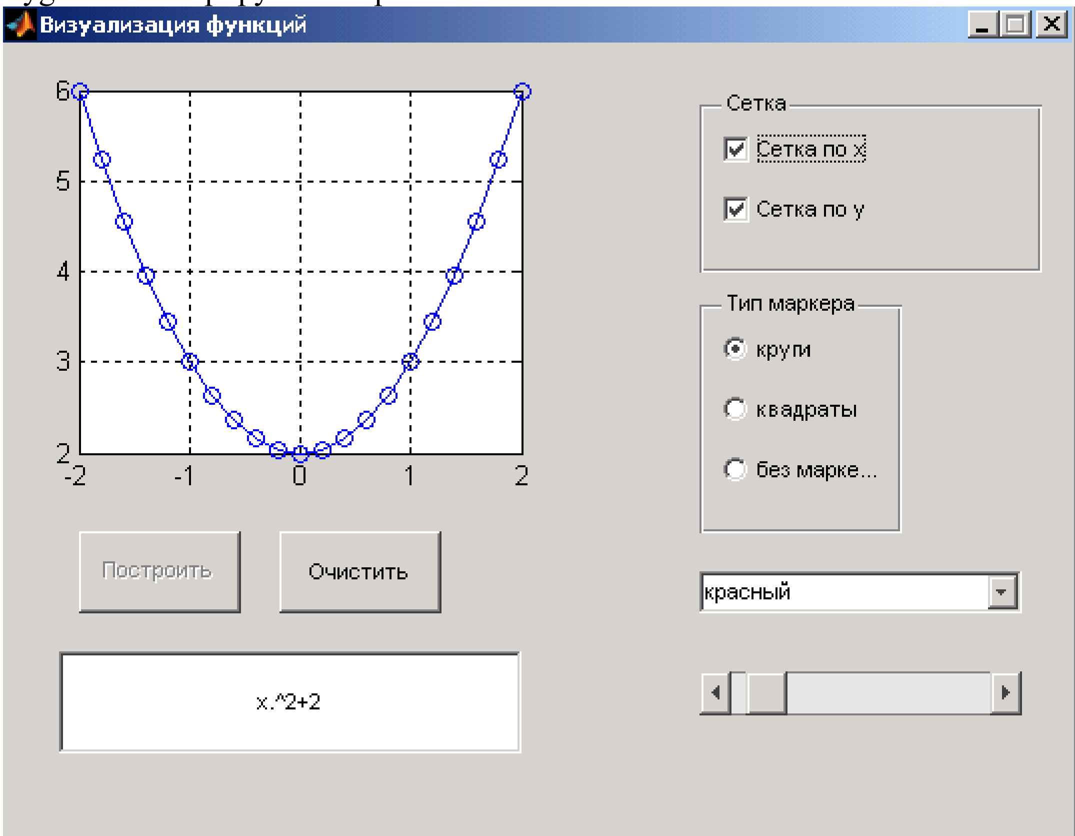


Рис. 13. Запуск приложения.

Дополним наше приложение еще двумя областями ввода текста и двумя кнопками. Допустим, пользователю необходимо ввести некоторую константу в одно окно и вывести ее удвоенное значение в другом окне; кроме того, добавим возможность закрытия окна приложения дополнительной кнопкой «Закрыть». Добавьте области ввода текста, установите свойство Tag в значения editConst и EditResult соответственно. Значения String редактора свойств очистите. Установите свойство Tag в значение PbnConst для первой кнопки и Close – для второй, а также свойство String в значения «Вывести» и «Закрыть». Для программирования необходимо использовать встроенные функции Matlab согласования форматов переменных. Следующий листинг отражает обработку добавленных кнопок:

```
function PbnClose_Callback(hObject, eventdata, handles)
close;

function PbnConst_Callback(hObject, eventdata, handles)
Con=get(handles.editConst,'String');
c=str2num(Con)*2;
Res=num2str(c);
```

```
set(handles.editResult,'String',Res);
```

На рис. 14 показан запуск приложения mygui0.

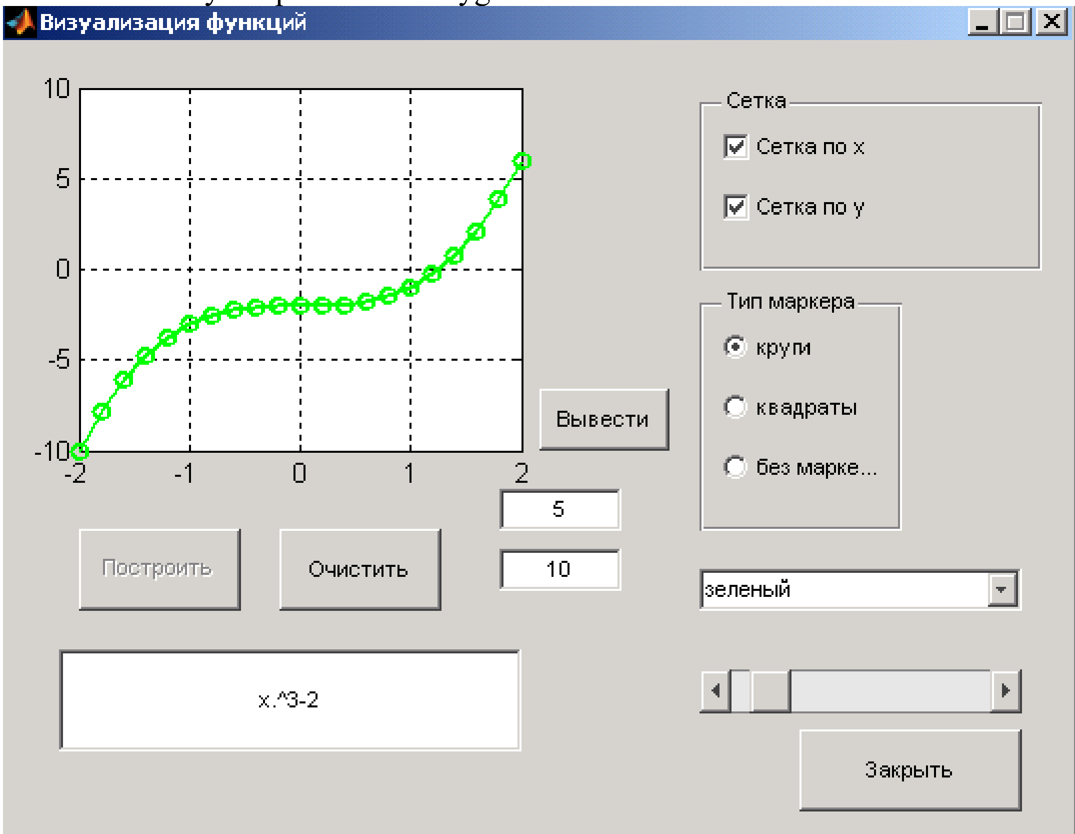


Рис. 14. Запуск приложения.