<u>Обязательный базовый раздел</u>:
С 23 сентября по 6 октября – 2 недели
 15 минут на прохождение (с любого IP, не обязательно Политехского)
2 попытки
70%-ный порог прохождения теста

<u>Тест №2:</u>
С 23 сентября по 6 октября – 2 недели
 90 минут на прохождение (с IP Политеха, под руководством преподавателя!!!)
1 попытка

Работа с матрицами и решение систем линейных алгебраических уравнений (СЛАУ)

Работа с матрицами

Действия с матрицами

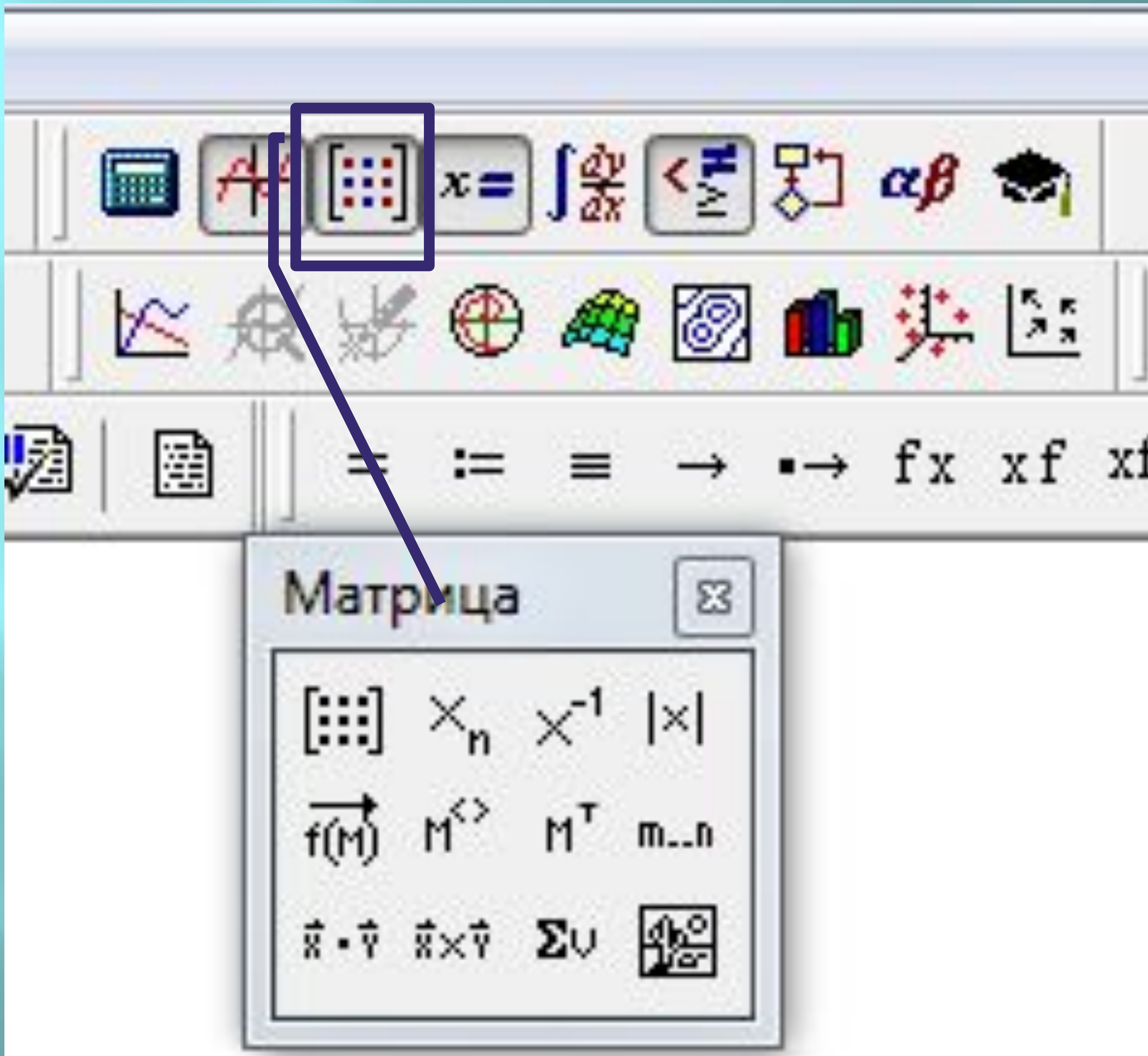
Поэлементные

$$x \cdot y = x_1 \cdot y_1 + x_2 \cdot y_2$$

Векторные

$$\left(x_1 \cdot y_1;\ x_2 \cdot y_2\right)$$

$$\left(x \cdot y\right)$$


Матрица

$$A := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \boxed{1} \end{pmatrix}$$

Строка

Элемент

Столбец

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 5 \\ 6 & 7 & 1 \end{pmatrix}$$

Индексация элементов начинается с 0!!!

Индексы:
Строка (1), столбец (0)

$$A_{\boxed{1,0}} = 4$$

$$A_{0,1} = 2$$

ORIGIN:=1

1 – вставка матрицы заданного размера
2 – индекс, обращение к элементу матрицы
3 – обратная матрица
4 – определитель
5 – столбец матрицы
6 – транспонированная матрица
7 – диапазон изменения переменной
8 – сума компонентов вектора

Основные числовые характеристики матриц

$\det(M)$	определитель
$tr(M)$	след
$rang(M)$ $rg(M)$	Ранг
$\mu(M)$	Число обусловленности
$\lVert M \rVert_{mun}$	Норма

Определитель матрицы **(детерминант)**

равен алгебраической сумме всевозможных произведений элементов данной матрицы, взятых по одному из каждой строки и из каждого столбца.

Для вычисления определителя матрицы используется знак модуля:

$$|A|.$$

След –

сумма всех диагональных элементов матрицы. Вычисляется с помощью функции tr(A).

Ранг -

число линейно-независимых строк матрицы. Вычисляется с помощью функции

rank(A).

Нормы матриц

могут быть самые разнообразные, в зависимости от того, в каком пространстве происходит работа с матрицами. Наиболее распространены следующие нормы:

Неопределенная норма: $\|\alpha\|_m = \max_i \sum_j |\alpha_{ij}|$

$L1$ норма: $\|\alpha\|_l = \max_j \sum_i |\alpha_{ij}|$

Евклидову норма: $\|\alpha\|_e = \sqrt{\sum_j |\alpha_{ij}|^2}$

Для вычисления **норм матриц** используют функции :

normi(*A*) - Возвращает неопределенную норму

norm1(*A*) - Возвращает $L1$ норму

norme(*A*) - возвращает евклидову норму

Число обусловленности – произведение двух норм A и A^{-1} . Вычисляется с помощью функций:

condi(A),
cond1(A),
conde(A)

для каждого типа нормы, соответственно.

$$A := \begin{pmatrix} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{pmatrix}$$

$|A| = 5$ определитель матрицы

$tr(A) = 8$ след матрицы

$rank(A) = 3$ ранг матрицы

$norm1(A) = 7$ норма L_1

$normi(A) = 7$ неопределенная норма

$norme(A) = 6.083$ евклидова норма

$cond1(A) = 23.8$ число обусловленности L_1

«Собственные» характеристики матрицы

Собственные числа- решения характеристического уравнения:

$$\det(A-\lambda E)=0$$

Собственные вектора- вектора, удовлетворяющие условию:

$$A\cdot z=\lambda\cdot z$$

$$\det(A - \lambda E) = 0$$

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad \lambda E := \lambda \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\left| \begin{pmatrix} 1-\lambda & 2 \\ 3 & 4-\lambda \end{pmatrix} \right| \rightarrow \lambda^2 - 5\cdot\lambda - 2$$

$$\text{polyroots}\left(\begin{pmatrix} -2 \\ -5 \\ 1 \end{pmatrix} \right) = \begin{pmatrix} -0.372 \\ 5.372 \end{pmatrix}$$

Вычисление «собственных» характеристик матрицы

Собственные числа- **eigenvals** (A)

Собственные вектора- **eigenvecs** (A) найдет все собственные вектора матрицы A.

Вычисление

«собственных» характеристик матрицы

eigenvec (A, $\lambda_к$) найдет собственный вектор, соответствующий собственному числу $\lambda_к$

Собственные вектора для данного собственного числа не уникальны!

Они могут отличаться на константу, *k*, поскольку в этом случае условие:

$$A\cdot\left(k\cdot z\right)=\lambda\cdot\left(k\cdot z\right)$$

все равно будет выполняться.

$$A := \begin{pmatrix} 3 & 5 & 1 \\ 2 & 1 & 1 \\ 2 & -1 & 4 \end{pmatrix}$$

$$z := \mathrm{eigenvals}(A) \qquad z = \begin{pmatrix} -1.072 \\ 6.224 \\ 2.848 \end{pmatrix}$$

$$v := \mathrm{eigenvecs}(A) \qquad v = \begin{pmatrix} 0.748 & -0.765 & -0.548 \\ -0.53 & -0.391 & -0.148 \\ -0.399 & -0.512 & 0.823 \end{pmatrix}$$

$$v1 := \mathrm{eigenvec}(A, z_2) \qquad v1 = \begin{pmatrix} -0.548 \\ -0.148 \\ 0.823 \end{pmatrix}$$

проверка полученного результата

$$A \cdot v1 = \begin{pmatrix} -1.561 \\ -0.421 \\ 2.344 \end{pmatrix} \qquad z_2 \cdot v1 = \begin{pmatrix} -1.561 \\ -0.421 \\ 2.344 \end{pmatrix} \qquad z_2 \cdot v^{\langle 2 \rangle} = \begin{pmatrix} -1.561 \\ -0.421 \\ 2.344 \end{pmatrix}$$

QR-разложение

$$A=Q\bullet R:$$

QQ^T=E(единичная матрица)

R-верхнетреугольная матрица.

Функция для этого разложения:

M=**qr**(A)

Матрица M, является объединением матриц Q и R. Чтобы получить сами матрицы необходимо воспользоваться функцией

submatrix(M, *i_нач*, *i_кон*, *j_нач*, *j_кон*).

Эта функция возвращает субматрицу(подматрицу), состоящую из всех элементов в строках с ***i_нач*** по ***i_кон*** и столбцах с ***j_нач*** по ***j_кон***.

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 5 \\ 6 & 7 & 1 \end{pmatrix}$$

$$M := \mathrm{lu}(A) = \begin{array}{cc} \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} & \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 6 & 7 & 1 \\ 0 & 1 & 0 & 0.667 & 1 & 0 & 0 & -3.667 & 4.333 \\ 1 & 0 & 0 & 0.167 & -0.227 & 1 & 0 & 0 & 3.818 \end{pmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \end{array}$$

$$L := \mathrm{submatrix}(M, 0, 2, 3, 5) = \begin{pmatrix} 1 & 0 & 0 \\ 0.667 & 1 & 0 \\ 0.167 & -0.227 & 1 \end{pmatrix}$$

$$U := \mathrm{submatrix}(M, 0, 2, 6, 8) = \begin{pmatrix} 6 & 7 & 1 \\ 0 & -3.667 & 4.333 \\ 0 & 0 & 3.818 \end{pmatrix}$$

LU-разложение

LU-разложением матрицы A, или треугольным разложением, называется матричное разложение вида

$$PA=LU,$$

где L и U – нижняя и верхняя треугольные матрицы соответственно, <u>при этом матрица L на диагонали имеет 1</u>; а P –матрица перестановок.

Функция для этого разложения:

M=**lu**(A)

возвращает матрицу M, являющуюся объединением матриц P, L и U.

Чтобы получить сами матрицы необходимо воспользоваться функцией

submatrix(M, *i_нач, i_кон, j_нач, j_кон*).

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 1 & 5 \\ 6 & 7 & 1 \end{pmatrix}$$

$$M := qr(A) = \begin{array}{cc} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} & \\ \begin{pmatrix} 0.137 & 0.337 & 0.931 & 7.28 & 6.593 & 3.983 \\ 0.549 & -0.808 & 0.212 & 0 & 3.245 & -2.547 \\ 0.824 & 0.483 & -0.296 & 0 & 0 & 3.556 \end{pmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \end{array}$$

$$Q := submatrix(M,0,2,0,2) = \begin{pmatrix} 0.137 & 0.337 & 0.931 \\ 0.549 & -0.808 & 0.212 \\ 0.824 & 0.483 & -0.296 \end{pmatrix}$$

$$R := submatrix(M,0,2,3,5) = \begin{pmatrix} 7.28 & 6.593 & 3.983 \\ 0 & 3.245 & -2.547 \\ 0 & 0 & 3.556 \end{pmatrix}$$

Разложение Холецкого

LU-разложение симметричной матрицы:

$$A = L\,L^T.$$

Функция для этого разложения:

L = **cholesky** (A)

При этом L - нижнетреугольная матрица.

Разложение Холецкого

Матрица А должна быть симметричной, положительно определенной, т.е. все ее собственные числа должны быть действительными и положительными.

$$A := \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

$$\mathrm{eigenvals}(A) = \begin{pmatrix} 0.586 \\ 2 \\ 3.414 \end{pmatrix}$$

$$M := \mathrm{cholesky}(A) = \begin{pmatrix} 1.414 & 0 & 0 \\ -0.707 & 1.225 & 0 \\ 0 & -0.816 & 1.155 \end{pmatrix}$$

$$M \cdot M^{T} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

Системы линейных алгебраических уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n \end{cases}$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{12} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{pmatrix}$$

Решение СЛАУ

точные методы - конечные алгоритмы для вычисления корней системы (решение систем с помощью обратной матрицы, правило Крамера, метод Гаусса и др.)

итерационные методы - позволяют получить решение системы с заданной точностью путем сходящихся итерационных процессов (метод итерации, метод Зейделя, метод градиентного спуска и др.)

Решение СЛАУ

Вследствие неизбежных округлений результаты даже *точных методов* являются приближенными

Эффективное применение *итерационных методов* существенно зависит от удачного выбора начального приближения и быстроты сходимости процесса

Системы линейных уравнений вида Ax=b можно решать «напрямую»

$$x=A^{-1}\cdot b$$

Можно решать с помощью функции

lsolve(*A*, *b*).

Функция возвращает вектор решения *x* такой, что *Ax = b.*

Функция lsolve использует LU-разложение.

Перед решением СЛАУ желательно вычислить определитель матрицы коэффициентов, чтобы убедиться в существовании и единственности решения.

Еще одним условием существования единственного решения является условие равенства ранга матрицы числу ее столбцов (rank(M) =cols (M)), выполнение этого условия гарантирует линейную независимость уравнений системы.

Mathcad Professional - [Решение матричных уравнений.mcd]

Файл Правка Вид Вставка Формат Математика Символы Окно ?

Матрица системы:

$$A := \begin{pmatrix} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{pmatrix}$$

Матрица правой части:

$$b := \begin{pmatrix} 5 \\ 0 \\ 15 \end{pmatrix}$$

Примечание: образцы матрицы и вектора соответствуют линейной системе

$$3x_1 - x_2 = 5$$

$$-2 \cdot x_1 + x_2 + x_3 = 0$$

$$2 \cdot x_1 - x_2 + 4 \cdot x_3 = 15$$

Вычисление определителя

$$|A| = 5$$

Определитель отличен от нуля,
система имеет единственное решение

Матрицы

Вычисление определителя |

Вычисление решения системы

$$x := A^{-1} \cdot b \qquad x = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

Решение системы с помощью функции lsolve

$$x := \mathrm{lsolve}(A, b) \qquad x = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

Проверка правильности решения

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

Метод Гаусса

Исходную систему приводят последовательным исключением неизвестных к эквивалентной системе с треугольной матрицей

$$\begin{cases} x_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \ldots + \alpha_{1n}x_n = \beta_1 \\ \qquad\qquad x_2 + \alpha_{23}x_3 + \ldots + \alpha_{2n}x_n = \beta_2 \\ \ldots \\ \qquad\qquad\qquad\qquad\qquad\qquad x_n = \beta_n \end{cases}$$

$$x_n = \beta_n$$

$$x_i = \beta_i - \sum_{j=i+1}^{n} \alpha_{ij}x_j \qquad i = n-1, n-2, \ldots 1$$

В Mathcad прямой и обратный ходы метода Гаусса выполняет функция

__rref__(*A*).

Которая возвращает ступенчатую форму матрицы *A*. Для использование этой функции необходимо сформировать единую матрицу, соединив вместе массивы ***A*** и ***b*** с помощью функции

augment(*A*, *B*)

-Которая возвращается массив, сформированный расположением ***A*** и ***B*** бок о бок.

Массивы ***A*** *и* ***B*** *должны иметь одинаковое число строк.*

Mathcad Professional - [Метод Гаусса.mcd]

Файл Правка Вид Вставка Формат Математика Символы Окно ?

Матрица системы:

$$A := \begin{pmatrix} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{pmatrix}$$

Матрица правой части:

$$b := \begin{pmatrix} 5 \\ 0 \\ 15 \end{pmatrix}$$

Примечание: образцы матрицы и вектора соответствуют линейной системе

$$3x_1 - x_2 = 5$$

$$-2 \cdot x_1 + x_2 + x_3 = 0$$

$$2 \cdot x_1 - x_2 + 4 \cdot x_3 = 15$$

$\mathrm{ORIGIN} := 1$

Формирование расширенной матрицы системы:

$$Ab := \mathrm{augment}(A, b) \qquad Ab = \begin{pmatrix} 3 & -1 & 0 & 5 \\ -2 & 1 & 1 & 0 \\ 2 & -1 & 4 & 15 \end{pmatrix}$$

Приведение расширенной матрицы системы к ступенчатому виду (прямой и обратный ходы метода Гаусса):

$$Ag := \mathrm{rref}(Ab) \qquad Ag = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

Формирование столбца решения системы:

$$x := \mathrm{submatrix}(Ag, 1, 3, 4, 4) \qquad x = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

Проверка правильности решения:

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

Нелинейные системы.

Для решения систем нелинейных уравнений (или неравенств), используют так называемый *блок решения,* который начинается с ключевого слова Given *(дано)* и заканчивается вызовом функции Find *(найти).* Между ними располагают «логические утверждения», задающие ограничения на значения искомых величин, т.е., уравнения или неравенства.

Всем переменным, используемым для обозначения неизвестных величин, должны быть заранее присвоены начальные значения.

При использовании блока Given/Find Mathcad применяет для решения систем уравнений и неравенств итерационный **метод Левенберга-Маркардта**, который является квазиньютоновским методом (разновидностью градиентного метода). **Алгоритм Левенберга – Маркардта** – метод оптимизации, направленный на решение задач о наименьших квадратах.

Алгоритм рассчитан на то, чтобы найти нули или, в наихудшем случае, минимизировать сумму квадратов компонент вектора значений нескольких функций, зависящих от нескольких переменных при заданных неравенствами ограничениях. Определение невязок таково:

Для уравнений:

error = left_side - right_side

Для неравенств <, >, , , или :

error = 0

если неравенство истинно, иначе

error = left_side - right_side

Когда решение ищется в поле комплексных чисел, Mathcad рассматривает вещественные и мнимые части как отдельные переменные в алгоритме и создает из комплексного уравнения два вещественных, отделяя вещественную и мнимую части исходного уравнения.

Метод Левенберга-Маркардта неприменим в случае, когда неизвестных меньше, чем ограничений. В этих случаях Mathcad возвращает ошибку *"слишком мало ограничений"* .

Расчет по методу Левенберга-Маркардта заканчивается, если:

- больше невозможно заметно уменьшить норму вектора невязки. В этом контексте "заметно" означает больше, чем больший из TOL и TOL· |*error*|. Этот критерий останавливает процесс решения, когда невязки не могут быть уменьшены далее.
- Когда норма минимизируемой функции становится относительно малой. В этом методе "относительно малый" означает норму меньшую, чем больший из TOL и TOL |v|. Этот критерий останавливает процесс решения, когда не находится нового определенного приближения.

Когда достигается одного из условий окончания расчета, и величина вектора невязки меньше или равна TOL , то Mathcad возвращает в качестве ответа значения неизвестных.

Если система решается функцией *Find,* и величина вектора невязки оказывается больше, чем TOL, то Mathcad отмечает систему уравнений сообщением об ошибке "***решение не найдено***", если при этом выполнено одно из следующих условий:

- Достигнута точка, из которой не может быть получено более точное приближение к решению.
- Достигнута точка, из которой невозможно выбрать подходящее направление спуска — направление вдоль которого ищется следующее приближение. В связи с этим продолжать итерации невозможно.
- Достигнут предел точности вычислений. Дальнейшие вычисления не увеличивают точность найденного решения вследствие влияния ошибок округления. Это часто случается, если установлено значение встроенной переменной TOL меньшее, чем 10^{-15}.

Причины появления сообщения об <u>ошибке</u> «*решение не найдено*» :

- Поставленная задача может не иметь решения.
- Для уравнения, которое не имеет вещественных решений, в качестве начального приближения взято вещественное число. Если решение задачи комплексное, то оно не будет найдено, если только в качестве начального приближения не взято также комплексное число.
- В процессе поиска решения последовательность приближений попала в точку локального минимума невязки. Метод поиска решения, который используется в Mathcad, не позволяет в этом случае построить следующее приближение, которое бы уменьшало невязку. Для поиска искомого решения пробуйте использовать различные начальные приближения или добавьте ограничения на переменные в виде неравенств, чтобы обойти точку локального минимума.
- В процессе поиска решения получена точка, которая не является точкой локального минимума, но из которой метод минимизации не может определить дальнейшее направление движения. Метод преодоления этой проблемы — такой же, как для точки локального минимума: измените начальное приближение или добавьте ограничения в виде неравенств, чтобы миновать нежелательную точку остановки.
- Возможно, поставленная задача не может быть решена с заданной точностью. Если значение встроенной переменной TOL слишком мало, то Mathcad может достигнуть точки, находящейся достаточно близко к решению задачи, но уравнения и ограничения при этом не будут выполнены с точностью, задаваемой переменной TOL. Попробуйте увеличить значение TOL где-нибудь выше блока решения уравнений.

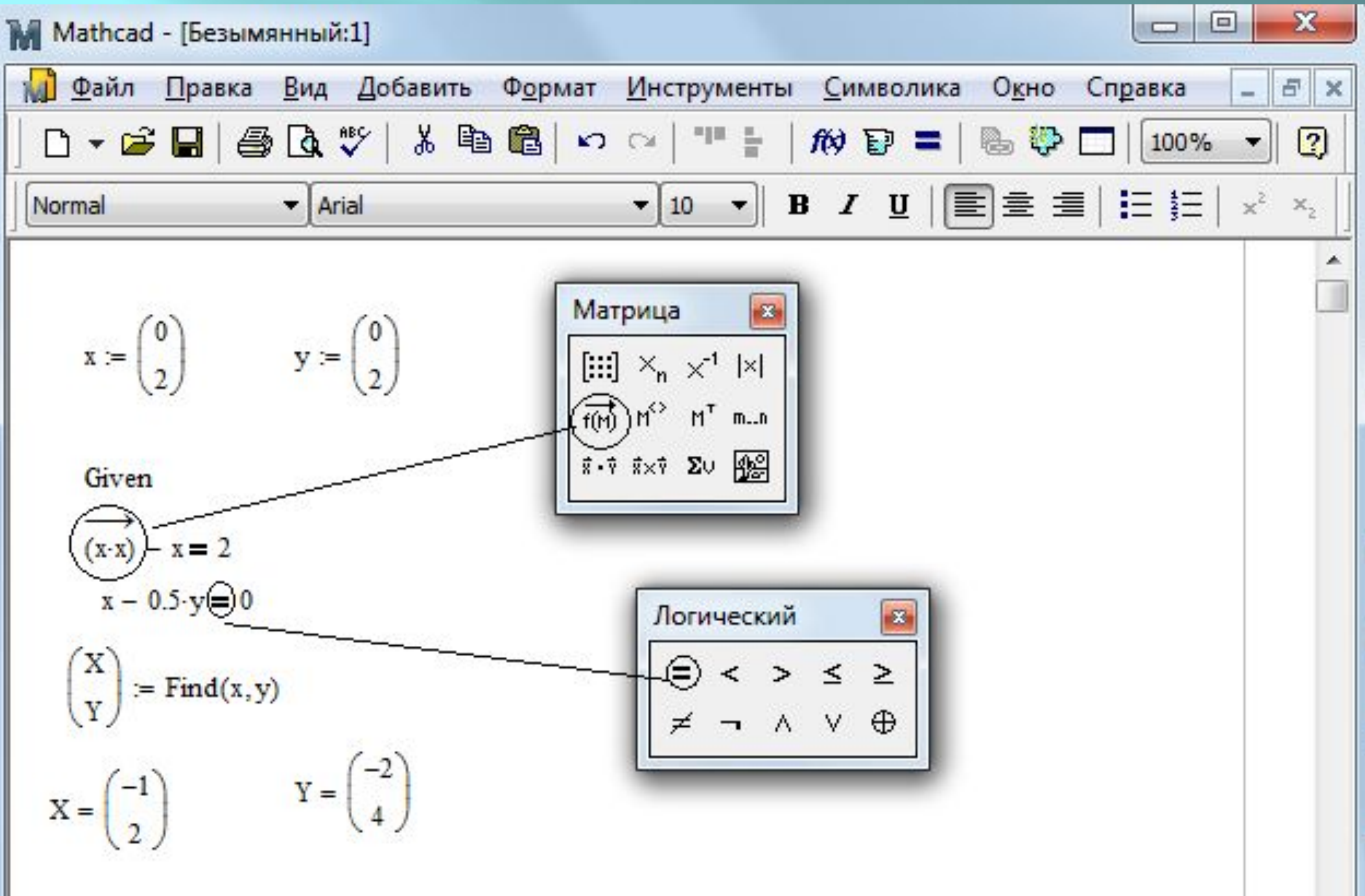
Если система решается функцией *Minerr*, то Mathcad возвращает решение в любом случае, даже если вектор невязки не близок к нулю.

Если процесс поиска решения заканчивается вследствие исчерпания заданного числа итераций, а ответ еще не найден, то состояния *Find* или *Minerr* отмечаются сообщением об ошибке

“отсутствует сходимость”.

<u>Одновременное нахождение несколько решений.</u>

<u>Система уравнений и начальные приближения должны быть переписаны в векторной форме.</u> Каждая переменная будет вектором, содержащим столько компонент, сколько решений находится. В системе изменения коснутся преимущественно членов с перемножением переменных, поскольку в данном случае необходим результат поэлементного перемножения векторов.


Mathcad - [Безымянный:1]
Файл Правка Вид Добавить Формат Инструменты Символика Окно Справка
100%
Normal
Arial
10
Матрица
Логический
x := (0, 2)
y := (0, 2)
Given
(x·x) − x = 2
x − 0.5·y = 0
(X, Y) := Find(x,y)
X = (−1, 2)
Y = (−2, 4)

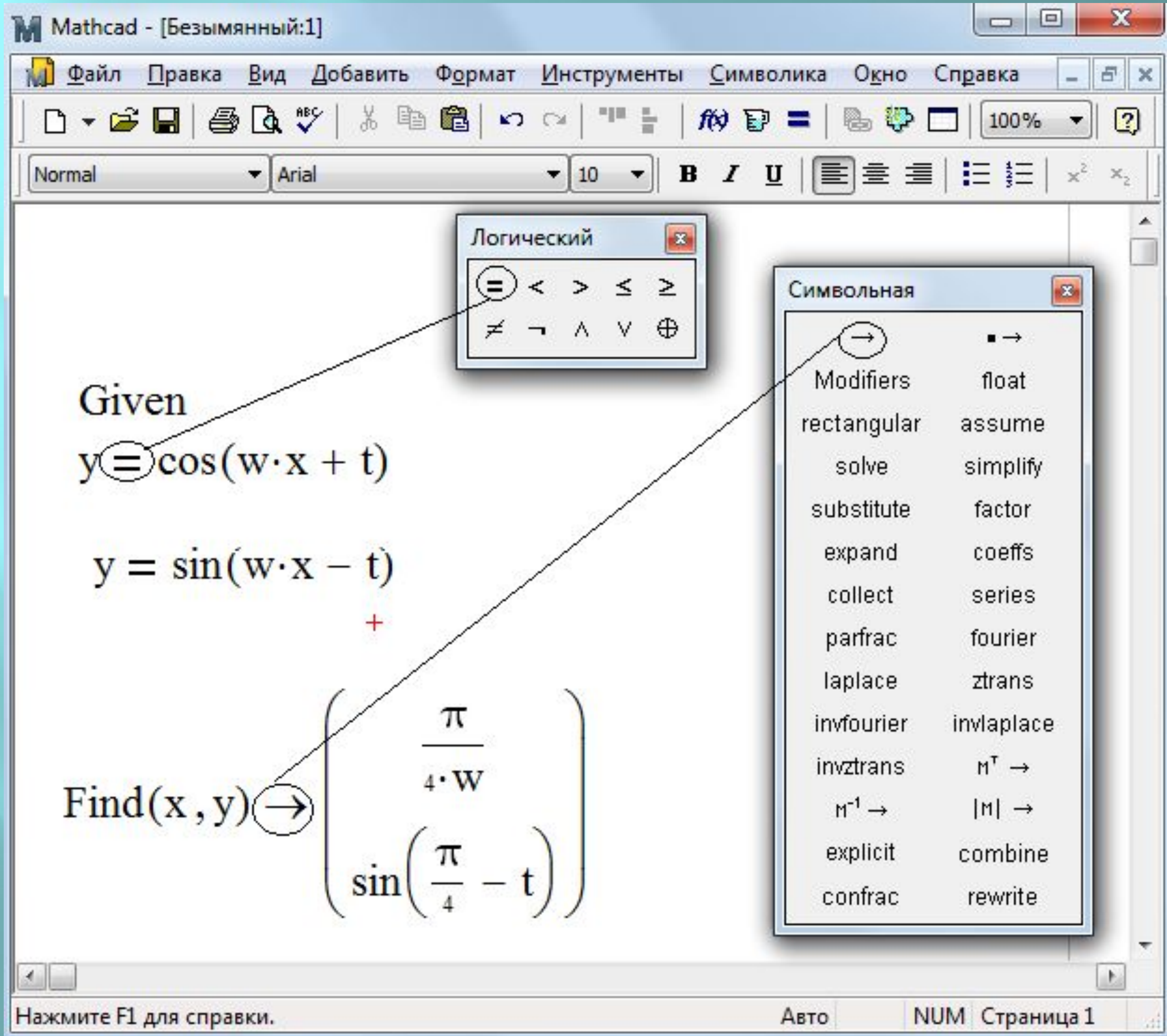
Аналитическое решение линейных и нелинейных систем уравнений

(в последних версиях используется ядро математической системы MuPAD, а в версиях MAthcad - до 13 включительно - Maple)

Блок Given-Find позволяет найти не только численное, но и аналитическое решение. Обычно при этом система уравнений записывается только с использованием буквенных обозначений переменных, без конкретных чисел.

Для получения аналитического решения используется оператор аналитического вычисления «→».

Следует обратить внимание, что здесь уже нет необходимости указывать начальные приближения, поскольку решение идет не численными, а символьными методами.


Mathcad - [Безымянный:1]
Файл Правка Вид Добавить Формат Инструменты Символика Окно Справка
100%
Normal
Arial
10
Логический
Символьная
Modifiers
float
rectangular
assume
solve
simplify
substitute
factor
expand
coeffs
collect
series
parfrac
fourier
laplace
ztrans
invfourier
invlaplace
invztrans
explicit
combine
confrac
rewrite
Given
y = cos(w·x + t)
y = sin(w·x − t)
Find(x, y) →
Нажмите F1 для справки.
Авто
NUM
Страница 1

Все рассмотренные методы решения относятся к задачам с квадратной матрицей, которые обычно и возникают при реализации численных методов. Однако, часто физические задачи приводят к системе прямоугольных матриц.

Конечно, в редких случаях система с прямоугольной матрицей может оказаться совместной (если выбран соответствующий вектор b) и система будет иметь решение.

Любопытно, что итерационный алгоритм блока Given/Find справляется с такой задачей, а алгоритм исключения, заложенный в функции lsolve — нет.