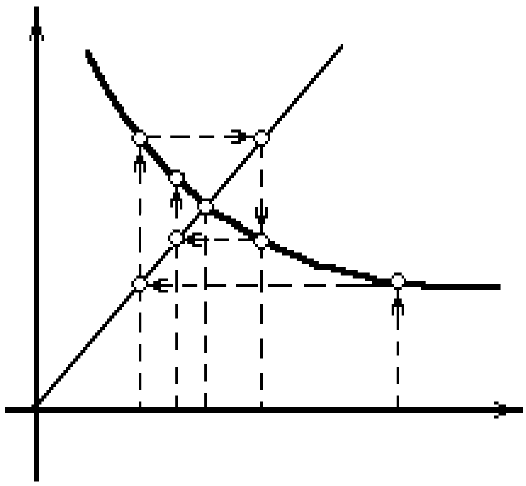
ЧИСЛЕННЫЕ МЕТОДЫ АНАЛИЗА



♦ ИЗДАТЕЛЬСТВО ГОУ ВПО ТГТУ ♦

Министерство образования и науки Российской Федерации

Государственное образовательное учреждение
высшего профессионального образования
«Тамбовский государственный технический университет»

ЧИСЛЕННЫЕ МЕТОДЫ АНАЛИЗА

Методические указания
для студентов специальности 240802

Тамбов
Издательство ГОУ ВПО ТГТУ
2010

УДК 66.011(076.5)
ББК В192.1я73-5
Ч-671

Рецензент

Доктор технических наук, профессор ГОУ ВПО ТГТУ
Е.Н. Туголуков

Составители:

А.Н. Пахомов,
Ю.В. Пахомова

Ч-671 Численные методы анализа : методические указания / сост. : А.Н. Пахомов, Ю.В. Пахомова. – Тамбов : Изд-во ГОУ ВПО ТГТУ, 2010. – 32 с. – 50 экз.

Даны методические указания к лабораторным работам по курсу «Основы математической физики и численные методы анализа»» для студентов специальности 240802.

УДК 66.011(076.5)

ББК В192.1я73-5

Учебное издание

ЧИСЛЕННЫЕ МЕТОДЫ АНАЛИЗА

Методические указания

Составители:

ПАХОМОВ Андрей Николаевич,
ПАХОМОВА Юлия Владимировна

Редактор Е.С. Кузнецова
Инженер по компьютерному макетированию Т.Ю. Зотова

Подписано в печать 02.10.2010
Формат 60 × 84/16. 1,86 усл. печ. л. Тираж 50 экз. Заказ № 529

Издательско-полиграфический центр ГОУ ВПО ТГТУ
392000, Тамбов, ул. Советская, 106, к. 14

ВВЕДЕНИЕ

Современному инженеру в своей практике приходится решать задачи расчёта и моделирования технологического процесса и оборудования. Для успешного решения подобных задач необходимо уметь применять численные и аналитические методы расчёта уравнений и их систем. Выбор метода решения в значительной мере определяет быстроту и точность получаемого решения.

Выбор подходящего метода для решения уравнений зависит от характера рассматриваемой задачи. Задачи, сводящиеся к решению отдельных уравнений и их систем, можно классифицировать по числу уравнений и в зависимости от предлагаемого характера и числа решений. Одно уравнение называется *линейным*, *алгебраическим* или *трансцендентным* в зависимости от того, имеет оно одно решение, n решений или неопределённое число решений. Систему уравнений будем называть *линейной* или *нелинейной* в зависимости от математической природы входящих в неё уравнений [2].

Решение линейного уравнения с одним неизвестным получается достаточно просто (см. школьный курс математики).

РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Нелинейные уравнения можно разделить на два класса – алгебраические и трансцендентные.

Алгебраическими уравнениями называют уравнения, содержащие только алгебраические функции (целые, рациональные, иррациональные): $P(x_1, x_2, \ldots, x_n)$, где P – многочлен с коэффициентами из поля рациональных чисел. В частности, многочлен является целой алгебраической функцией.

Уравнения, содержащие другие функции (тригонометрические, показательные, логарифмические и др.), называются *трансцендентными*, например:

$$\cos x = x, \quad \log x = x - 5, \quad x^3 = \log x + x^5 + 40.$$

Более строгое определение таково: Трансцендентное уравнение – это уравнение вида $f(x) = g(x)$, где функции f и g являются аналитическими функциями, и по крайней мере одна из них не является алгебраической.

Методы решения нелинейных уравнений делятся на две группы:

1) точные методы;
2) итерационные методы.

Точные методы позволяют записать корни в виде некоторого конечного соотношения (формулы). Из школьного курса алгебры известны такие методы для решения тригонометрических, логарифмических, показательных, а также простейших алгебраических уравнений.

Как известно, многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение степени выше четвёртой. Кроме того, в некоторых случаях уравнение содержит коэффициенты, известные лишь приблизительно, и, следовательно, сама задача о точном определении корней уравнения теряет смысл. Для их решения используются *итерационные методы* с заданной степенью точности.

Пусть дано уравнение

$$f(x) = 0, \tag{1}$$

где:

1) функция $f(x)$ непрерывна на отрезке $[a, b]$ вместе со своими производными первого и второго порядка;
2) значения $f(x)$ на концах отрезка имеют разные знаки ($f(a)\cdot f(b) < 0$);
3) первая и вторая производные $f'(x)$ и $f''(x)$ сохраняют определённый знак на всём отрезке.

Условия 1 и 2 гарантируют, что на интервале $[a, b]$ находится хотя бы один корень, а из 3 следует, что $f(x)$ на данном интервале монотонна, и поэтому корень будет единственным.

Решить уравнение (1) *итерационным методом* – значит установить, имеет ли оно корни, сколько корней, и найти значения корней с нужной точностью.

Всякое значение ξ, обращающее функцию $f(x)$ в нуль, т.е. такое, что $f(\xi) = 0$, называется *корнем уравнения* (1) или *нулём* функции $f(x)$.

Задача нахождения корня уравнения $f(x) = 0$ итерационным методом состоит из двух этапов:

1) *определение отрезка локализации корней* – отыскание приближённого значения корня или содержащего его отрезка;
2) *уточнение приближённых корней* – доведение их до заданной степени точности.

Процесс определения отрезка локализации корней начинается с установления знаков функции $f(x)$ в границах $x = a$ и $x = b$ области её существования.

***Пример* 1.** Определить отрезок локализации корней уравнения:

$$x^3 - 7x + 3 = 0. \tag{2}$$

Построим график этой функции (рис. 1).

Следовательно, уравнение (2) имеет три действительных корня, лежащих в интервалах [–3, –1], [0, 1] и [1, 5].

Приближённые значения корней (*начальные приближения*) могут быть также известны из физического смысла задачи, из решения аналогичной задачи при других исходных данных или могут быть найдены графическим способом.

В инженерной практике распространён *графический способ* определения приближённых корней.

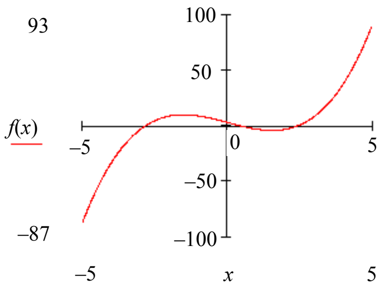


Рис. 1. Определение отрезка локализации

Принимая во внимание, что действительные корни уравнения (1) – это точки пересечения графика функции $f(x)$ с осью абсцисс, достаточно построить график функции $f(x)$ и отметить точки пересечения $f(x)$ с осью Ox или отметить на оси Ox отрезки, содержащие по одному корню. Построение графиков часто удаётся сильно упростить, заменив уравнение (1) *равносильным* ему уравнением:

$$f_1(x) = f_2(x), \tag{3}$$

где функции $f_1(x)$ и $f_2(x)$ более простые, чем функция $f(x)$. Тогда, построив графики функций $y = f_1(x)$ и $y = f_2(x)$, искомые корни получим как абсциссы точек пересечения этих графиков.

***Пример* 2.** Графически отделить корни уравнения (рис. 2):

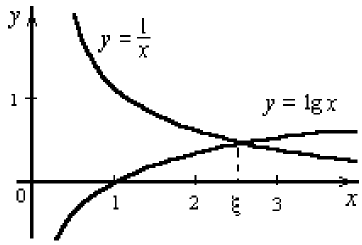


Рис. 2. Графическое определение корня

$$x \lg x = 1. \tag{4}$$

Уравнение (4) удобно переписать в виде равенства:

$$\lg x = 1/x$$

Отсюда ясно, что корни уравнения (4) могут быть найдены как абсциссы точек пересечения логарифмической кривой $y = \lg x$ и гиперболы $y = 1/x$. Построив эти кривые, приближённо найдём единственный корень $\xi = 2,5$ уравнения (4) или определим его содержащий отрезок [2, 3].

Итерационный процесс состоит в последовательном уточнении начального приближения x_0. Каждый такой шаг называется *итерацией*. В результате итераций находится последовательность приближённых значений корня $x_1, x_2, ..., x_n$. Если эти значения с увеличением числа итераций n приближаются к истинному значению корня, то говорят, что итерационный процесс *сходится*.

Метод половинного деления

Для нахождения корня уравнения (1), принадлежащего отрезку $[a, b]$, делим этот отрезок пополам. Если $f\left(\frac{a+b}{2}\right) = 0$, то $\xi\left(\frac{a+b}{2}\right) = 0$ является корнем уравнения. Если $f\left(\frac{a+b}{2}\right) \neq 0$, то выбираем ту из половин $\left[a, \frac{a+b}{2}\right]$ или $\left[\frac{a+b}{2}, b\right]$, на концах которой функция $f(x)$ имеет противоположные знаки. Новый суженный отрезок $[a_1, b_1]$ снова делим пополам и производим те же самые действия.

Метод половинного деления практически удобно применять для грубого нахождения корня данного уравнения, метод прост и надёжен, всегда сходится.

***Пример* 3.** Методом половинного деления уточнить корень уравнения $f(x) \equiv x^4 + 2x^3 - x - 1 = 0$, лежащий на отрезке [0, 1].

Последовательно имеем:

$f(0) = -1$; $f(1) = 1$; $f(0,5) = 0,06 + 0,25 - 0,5 - 1 = -1,19$;
$f(0,75) = 0,32 + 0,84 - 0,75 - 1 = -0,59$;

$f(0{,}875) = 0{,}59 + 1{,}34 - 0{,}88 - 1 = +\,0{,}05$;
$f(0{,}8125) = 0{,}436 + 1{,}072 - 0{,}812 - 1 = -\,0{,}304$;
$f(0{,}8438) = 0{,}507 + 1{,}202 - 0{,}844 - 1 = -\,0{,}135$;
$f(0{,}8594) = 0{,}546 + 1{,}270 - 0{,}859 - 1 = -\,0{,}043$ и т.д.

Можно принять

$$\xi = \frac{1}{2}(0{,}859 + 0{,}875) = 0{,}867.$$

Метод хорд

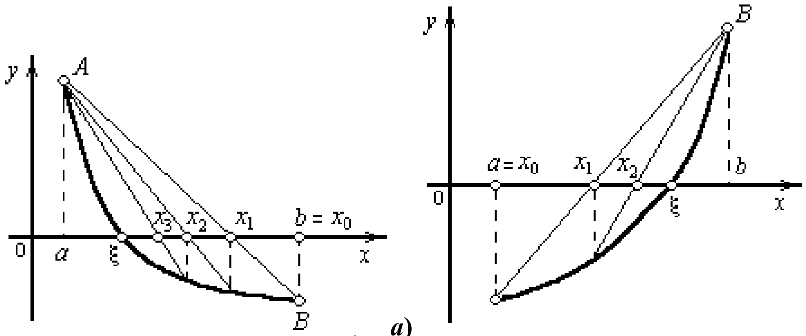
В данном методе процесс итераций состоит в том, что в качестве приближений к корню уравнения (1) принимаются значения $x_1, x_2, ..., x_n$ точек пересечения хорды AB с осью абсцисс (рис. 3). Сначала запишем уравнение хорды AB:

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

Для точки пересечения хорды AB с осью абсцисс ($x = x_1$, $y = 0$) получим уравнение:

$$x_1 = \frac{f(a)}{f(b) - f(a)}\left(b - a\right).$$

Пусть для определённости $f''(x) > 0$ при $a \le x \le b$ (случай $f''(x) < 0$ водится к нашему, если записать уравнение в виде $f(x) = 0$). Тогда кривая $= f(x)$ будет выпукла вниз и, следовательно, расположена ниже своей хорды AB. Возможны два случая: 1) $f(a) > 0$ (рис. 3, *а*) и 2) $f(b) < 0$ (рис. 3, *б*).



а) *б*) **а) б)**

Рис. 3. Метод хорд

В первом случае конец a неподвижен и последовательные приближения:

$$x_0 = b;$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f(x_i) - f(a)}(x_i - a),\ (i = 0,\ 1,\ 2,\ ...) \qquad (5)$$

образуют ограниченную монотонно убывающую последовательность, причём

$$a < \xi < ... < x_{i+1} < x_i < ... < x_1 < x_0 .$$

Во втором случае неподвижен конец b, а последовательные приближения:

$$x_0 = a;$$

$$x_{i+1} = x_i - \frac{f(x_i)}{f(b) - f(x_i)}(b - x_i) \qquad (6)$$

образуют ограниченную монотонно возрастающую последовательность, причём

$$x_0 < x_1 < ... < x_i < x_{i+1} < ... < \xi < b .$$

Обобщая эти результаты, заключаем:

1) неподвижен тот конец, для которого знак функции $f(x)$ совпадает со знаком её второй производной $f''(x)$;

2) последовательные приближения x_n лежат по ту сторону корня ξ, где функция $f(x)$ имеет знак, противоположный знаку её второй производной $f''(x)$.

Итерационный процесс продолжается до тех пор, пока не будет обнаружено, что $|x_i - x_{i-1}| < \varepsilon$, где ε – заданная предельная абсолютная погрешность.

***Пример* 4.** Найти положительный корень уравнения

$$f(x) \equiv x^3 - 0{,}2x^2 - 0{,}2x - 1{,}2 = 0$$

с точностью $\varepsilon = 0{,}01$.

Прежде всего определяем отрезок локализации корня.

Так как $f(1) = –0,6 < 0$ и $f(2) = 5,6 > 0$, то искомый корень ξ лежит в интервале [1, 2]. Полученный интервал велик, поэтому разделим его пополам.

Так как $f(1,5) = 1,425 > 0$, то $1< \xi < 1,5$.

Так как $f''(x) = 6x – 0,4 > 0$ при $1 < x < 1,5$ и $f(1,5) > 0$, то воспользуемся формулой (5) для решения поставленной задачи:

$$x_1 = 1 + \frac{0,6}{1,425 + 0,6}(1,5 - 1) = 1,15\,; \quad |x_1 - x_0| = 0,15 > \varepsilon\,,$$

следовательно, продолжаем вычисления;

$f(x_1) = –0,173;$

$$x_2 = 1,15 + \frac{0,173}{1,425 + 0,173}(1,5 - 1,15) = 1,190\,; \quad |x_2 - x_1| = 0,04 > \varepsilon\,,$$

$f(x_2) = –0,036;$

$$x_3 = 1,190 + \frac{0,036}{1,425 + 0,036}(1,5 - 1,190) = 1,198\,; \quad |x_3 - x_2| = 0,008 < \varepsilon\,.$$

Таким образом, можно принять $\xi = 1,198$ с точностью $\varepsilon = 0,01$.

Точный корень уравнения $\xi = 1,2$.

Метод Ньютона

Отличие этого итерационного метода от предыдущего состоит в том, что вместо хорды на каждом шаге проводится касательная к кривой $y = f(x)$ при $= x_i$ и ищется точка пересечения касательной с осью абсцисс (рис. 4). При этом не обязательно задавать отрезок $[a, b]$, содержащий корень уравнения (1), достаточно найти лишь некоторое начальное приближение корня $x = x_0$.

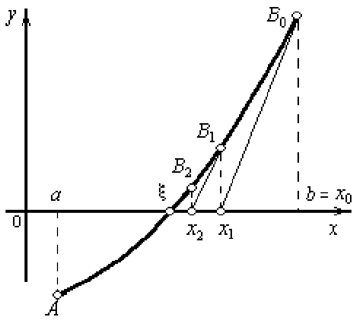


Рис. 4. Метод Ньютона

Применяя метод Ньютона, следует руководствоваться следующим правилом: *в качестве исходной точки x_0 выбирается тот конец интервала* $[a, b]$, *которому отвечает ордината того же знака*, *что и знак* $f''(x)$.

Уравнение касательной, проведённой к кривой $y = f(x)$ через точку B_0 с координатами x_0 и $f(x_0)$, имеет вид

$$y - f(x_0) = f'(x_0)(x - x_0)\,.$$

Отсюда найдём следующее приближение корня x_1 как абсциссу точки пересечения касательной с осью Ox ($y = 0$)

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}\,.$$

Аналогично могут быть найдены и следующие приближения как точки пересечения с осью абсцисс касательных, проведённых в точках B_1, B_2 т.д. Формула для $i + 1$ приближения имеет вид

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}\,. \tag{7}$$

Для окончания итерационного процесса может быть использовано или условие $|f(x_i)| < \varepsilon$, или условие близости двух последовательных приближений $|x_i - x_{i-1}| < \varepsilon$.

Итерационный процесс сходится, если

$$f(x_0)f''(x_0) > 0.$$

Метод последовательных приближений

Для использования метода последовательных приближений исходное нелинейное уравнение $f(x) = 0$ заменяется равносильным уравнением

$$x = \varphi(x). \tag{8}$$

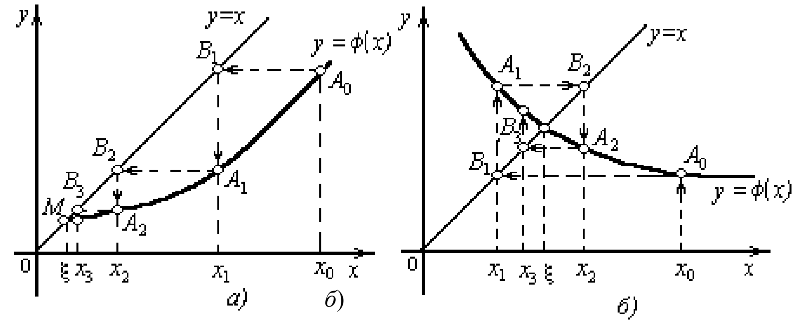
Пусть известно начальное приближение корня $x = x_0$. Подставляя это значение в правую часть уравнения (8), получим новое приближение:

$$x_1 = \varphi(x_0).$$

Далее, подставляя каждый раз новое значение корня в (8), получаем последовательность значений:

$$x_{i+1} = \varphi(x_i), \quad (i = 0, 1, \ldots, n). \tag{9}$$

Геометрически метод итерации может быть пояснён следующим образом. Построим на плоскости xOy графики функций $y = x$ и $y = \varphi(x)$. Каждый действительный корень ξ уравнения (8) является абсциссой точки пересечения M кривой $y = \varphi(x)$ с прямой $y = x$ (рис. 5, *а*).



а)

Рис. 5. Сходящиеся итерационные процессы

Отправляясь от некоторой точки A_0 $[x_0, \varphi(x_0)]$, строим ломаную $A_0B_1A_1B_2A_2$ … («лестница»), звенья которой попеременно параллельны оси Ox и оси Oy, вершины $A_0, A_1, A_2, \ldots$ лежат на кривой $y = \varphi(x)$, а вершины $B_1, B_2, B_3, \ldots,$ – на прямой $y = x$. Общие абсциссы точек A_1 и B_1, A_2 и B_2, …, очевидно, представляют собой соответственно последовательные приближения x_1, x_2, … корня ξ.

Возможен также другой вид ломаной $A_0B_1A_1B_2A_2$ … – «спираль» рис. 5, *б*). Решение в виде «лестницы» получается, если производная $\varphi'(x)$ положительна, а решение в виде «спирали», если $\varphi'(x)$ отрицательна.

На рисунке 5, *а*, *б* кривая $y = \varphi(x)$ в окрестности корня ξ пологая, т.е. $|\varphi'(x)| < 1$, и процесс итерации сходится. Однако, если рассмотреть случай, где $|\varphi'(x)| > 1$, то процесс итерации может быть расходящимся (рис. 6). оэтому для практического применения метода итерации нужно выполнение достаточного условия сходимости итерационного процесса.

Пусть функция $\varphi(x)$ определена и дифференцируема на отрезке $[a, b]$, причём все её значения $\varphi(x) \in [a, b]$.

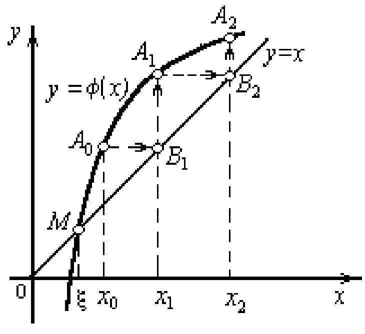


Рис. 6. Расходящийся итерационный процесс

Тогда, если существует правильная дробь q такая, что $|\varphi'(x)| \leq q < 1$ при $a < x < b$, то:

1) процесс итерации $x_{i+1} = \varphi(x_i)$, $(i = 0, 1, \ldots, n)$ сходится независимо от начального значения $x_0 \in [a, b]$;
2) предельное значение $\xi = \lim\limits_{n\to\infty} x_n$ является единственным корнем уравнения $x = \varphi(x)$ на отрезке $[a, b]$.

***Пример* 5.** Уравнение

$$f(x) \equiv x^3 - x - 1 = 0 \tag{10}$$

имеет корень $\xi \in [1, 2]$, так как $f(1) = -1 < 0$ и $f(2) = 5 > 0$.

Уравнение (10) можно записать в виде

$$x = x^3 - 1. \tag{11}$$

Здесь

$$\varphi(x) = x^3 - 1 \quad \text{и} \quad \varphi'(x) = 3x^2;$$

поэтому

$$\varphi'(x) \ge 3 \quad \text{при} \quad 1 \le x \le 2$$

и, следовательно, условия сходимости процесса итерации не выполнены.

Если записать уравнение (10) в виде

$$x = \sqrt[3]{x+1}\,, \tag{12}$$

то будем иметь:

$$\psi(x) = \sqrt[3]{x+1} \quad \text{и} \quad \psi'(x) = \frac{1}{3\sqrt[3]{(x+1)^2}}\,.$$

Отсюда $0 < \psi'(x) < \frac{1}{3\sqrt[3]{4}} < \frac{1}{4}$ при $1 \le x \le 2$ и, значит, процесс итерации для уравнения (12) быстро сойдётся.

Найдём корень ξ уравнения (10) с точностью до 10^{-2}. Вычисляем последовательные приближения x_n с одним запасным знаком по формуле

$$x_i = \sqrt[3]{x_i + 1}\,,\ \left(i = 0,\ 1,\ 2,\ \ldots,\ n\right).$$

Найденные значения помещены в табл. 1.

1. Значения последовательных приближений x_i

i	0	1	2	3	4
x_i	1	1,260	1,312	1,322	1,3243

С точностью до 10^{-2} можно положить $\xi = 1{,}324$.

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

Способы решения систем линейных уравнений делятся на две группы:

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

2) *итерационные методы*, позволяющие получить решение системы с заданной точностью путём сходящихся итерационных процессов (метод итерации, метод Зейделя и др.).

Вследствие неизбежных округлений результаты даже точных методов являются приближёнными. При использовании итерационных методов, сверх того, добавляется погрешность метода.

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

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных $x_1, x_2, \ldots, x_n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1\,; \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2\,; \\ \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n\,. \end{cases} \tag{13}$$

В соответствии с правилом умножения матриц рассмотренная система линейных уравнений может быть записана в матричном виде

$$Ax = b, \tag{14}$$

где:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}; \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}; \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}. \tag{15}$$

Матрица A, столбцами которой являются коэффициенты при соответствующих неизвестных, а строками – коэффициенты при неизвестных в соответствующем уравнении, называется *матрицей системы*; матрица-столбец b, элементами которой являются правые части уравнений системы, называется *матрицей правой части* или просто *правой частью системы*. Матрица-столбец x, элементы которой – искомые неизвестные, называется *решением системы*.

Если матрица A – неособенная, т.е. $\det A \ne 0$, то система (13) или эквивалентное ей матричное уравнение (14) имеет единственное решение.

В самом деле, при условии det $A \neq 0$ существует обратная матрица A^{-1}. Умножая обе части уравнения (14) на матрицу A^{-1}, получим:

$$A^{-1}Ax = A^{-1}b,$$
$$x = A^{-1}b. \quad (16)$$

Формула (16) даёт решение уравнения (14), и оно единственно.

Для решения системы линейных уравнений в MathCAD применяется функция *lsolve*.

Формат: **lsolve(*A*, *b*) –** возвращает вектор решения x такой, что $Ax = b$.

***Аргументы*:**

A – квадратная, несингулярная матрица.

b – вектор, имеющий столько же рядов, сколько рядов в матрице ***A***.

Метод итерации

Пусть дана линейная система (13). Введя в рассмотрение матрицы (15), систему (13) коротко можно записать в виде матричного уравнения (14). Предполагая, что диагональные коэффициенты

$$a_{ij} \neq 0 \ (i = 1, 2, \ldots, n),$$

разрешим первое уравнение системы (13) относительно x_1, второе – относительно x_2 и т.д. Тогда получим эквивалентную систему

$$\begin{cases} x_1 = \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \ldots + \alpha_{1n}x_n\,; \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_2 + \ldots + \alpha_{2n}x_n\,; \\ \ldots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + \ldots + \alpha_{n,n-1}x_{n-1}\,, \end{cases} \quad (17)$$

где $\beta_i = \dfrac{b_i}{a_{ii}}$; $\alpha_{ij} = -\dfrac{a_{ij}}{a_{ii}}$ при $i \neq j$ и $\alpha_{ij} = 0$ при $i = j$ $(i, j = 1, 2, \ldots, n)$.

Введя матрицы $\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_{n1} & \alpha_{n2} & \ldots & \alpha_{nn} \end{bmatrix}$ и $\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \ldots \\ \beta_n \end{bmatrix}$, систему (18) можно записать в матричной форме

$$x = \beta + \alpha x,$$

а любое $(k + 1)$ приближение вычисляется по формуле

$$x^{(k+1)} = \beta + \alpha x^{(k)}. \quad (18)$$

Напишем формулы приближений в развёрнутом виде:

$$\begin{cases} x_i^{(0)} = \beta_i\,; \\ x_i^{(k+1)} = \beta_i + \sum\limits_{j=1}^{n} \alpha_{ij} x_i^{(k)}; \\ (\alpha_{ii} = 0;\ i = 1, \ldots, n;\ k = 0, 1, 2, \ldots, n). \end{cases} \quad (18')$$

Процесс итерации для приведённой линейной системы (18) сходится к единственному её решению, если какая-нибудь каноническая норма матрицы α меньше единицы, т.е. для итерационного процесса (19) достаточное условие есть

$$\| \alpha \| < 1. \quad (19)$$

Таким образом, процесс итерации для системы (17) сходится, если:

1) $\| \alpha \|_m = \max\limits_i \sum\limits_j |\alpha_{ij}| < 1$ (m – норма или неопределённая норма),

или

2) $\| \alpha \|_l = \max\limits_j \sum\limits_i |\alpha_{ij}| < 1$ (l – норма или норма $L1$),

или

3) $\| \alpha \|_k = \sqrt{\sum\limits_{i,j} |\alpha_{ij}|^2} < 1$ (k – норма или Евклидова норма).

Также для системы (13) процесс итерации сходится, если выполнены неравенства:

1) $\left|a_{ii}\right| > \sum_{j=1}^{n}{}' \left|a_{ij}\right| \quad (i = 1, 2, \ldots, n)$ (20)

или

2) $\left|a_{jj}\right| > \sum_{i=1}^{n}{}' \left|a_{ij}\right| \quad (j = 1, 2, \ldots, n)$, (21)

где штрих у знака суммы означает, что при суммировании пропускаются значения $i = j$, т.е. сходимость имеет место, если модули диагональных элементов матрицы A системы (13) или для каждой строки превышают сумму модулей недиагональных элементов этой строки, или же для каждого столбца превышают сумму модулей недиагональных элементов этого столбца.

***Пример* 6.** Пусть $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$. Имеем:

$\| A \|_m = \max\,(1 + 2 + 3,\ 4 + 5 + 6,\ 7 + 8 + 9) = \max\,(6, 15, 24) = 24;$

$\| A \|_l = \max\,(1 + 4 + 7,\ 2 + 5 + 8,\ 3 + 6 + 9) = \max\,(12, 15, 18) = 18;$

$\| A \|_k = \sqrt{1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 8^2 + 9^2} = \sqrt{285} \approx 16{,}9$.

В Mathcad существуют специальные функции для вычисления норм матриц:

normi(*A*) – возвращает неопределённую норму матрицы A;

norm1(*A*) – возвращает $L1$, норму матрицы A;

norme(*A*) – возвращает Евклидову норму матрицы A.

В качестве *условия окончания* итерационного процесса можно взять условие

$$\frac{\left\| x^{(k+1)} - x^{(k)} \right\|}{\left\| x^{(k+1)} \right\|} \le \varepsilon\,, \tag{22}$$

где ε – заданная погрешность приближённого решения $x \approx x^{(k+1)}$.

РЕШЕНИЕ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

В отличие от систем линейных уравнений для систем нелинейных уравнений не существует прямых методов решения. Лишь в отдельных случаях систему можно решить аналитически. Например, для системы из двух уравнений иногда удаётся выразить одно неизвестное через другое и таким образом свести задачу к решению одного нелинейного уравнения относительно одного неизвестного. Поэтому итерационные методы для нелинейных систем приобретают особую важность.

Метод Ньютона

Рассмотрим нелинейную систему уравнений

$$\begin{cases} f_1(x_1,\ x_2,\ \ldots,\ x_n) = 0; \\ f_2(x_1,\ x_2,\ \ldots,\ x_n) = 0; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_n(x_1,\ x_2,\ \ldots,\ x_n) = 0 \end{cases} \tag{23}$$

или в векторной форме

$$f(x) = 0, \tag{23′}$$

где $f = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_n \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}.$

Для решения системы (23′) будем пользоваться методом последовательных приближений.

Предположим, известно k-е приближение $x^{(k)} = \left(x_1^{(k)},\ x_2^{(k)},\ \ldots,\ x_n^{(k)}\right)$ одного из изолированных корней $x = (x_1,\ x_2,\ \ldots,\ x_n)$ векторного уравнения (23′). Тогда точный корень уравнения (23′) можно представить в виде

$$x = x^{(k)} + \Delta x^{(k)}, \tag{24}$$

где $\Delta x^{(k)} = \left(\Delta x_1^{(k)},\ \Delta x_2^{(k)},\ \ldots,\ \Delta x_n^{(k)}\right)$ – поправка (погрешность корня).

Подставляя выражение (24) в (23′), будем иметь

$$f\left(x^{(k)} + \Delta x^{(k)}\right) = 0\,. \tag{25}$$

Предполагая, что функция $f(x)$ непрерывно дифференцируема в некоторой выпуклой области, содержащей x и $x^{(k)}$, разложим левую часть уравнения (25) по степеням малого вектора $\Delta x^{(k)}$, ограничиваясь линейными членами:

$$f\left(x^{(k)}+\Delta x^{(k)}\right)=f\left(x^{(k)}\right)+f'\left(x^{(k)}+\Delta x^{(k)}\right)=0 \qquad (26)$$

или в развёрнутом виде:

$$\begin{cases} f_1\left(x_1^{(k)}+\Delta x_1^{(k)},\ \ldots, x_n^{(k)}+\Delta x_n^{(k)}\right)=f_1\left(x_1^{(k)},\ \ldots, x_n^{(k)}\right)+\Delta x_1^{(k)}\dfrac{\partial f_1}{\partial x_1}+\ldots+\Delta x_n^{(k)}\dfrac{\partial f_1}{\partial x_n}=0; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_n\left(x_1^{(k)}+\Delta x_1^{(k)},\ \ldots, x_n^{(k)}+\Delta x_n^{(k)}\right)=f_n\left(x_1^{(k)},\ \ldots, x_n^{(k)}\right)+\Delta x_1^{(k)}\dfrac{\partial f_n}{\partial x_1}+\ldots+\Delta x_n^{(k)}\dfrac{\partial f_n}{\partial x_n}=0. \end{cases} \qquad (26')$$

Из формул (26) и (26′) вытекает, что под производной $f'(x)$ следует понимать *матрицу Якоби* системы функций $f_1, f_2, \ldots, f_n$ относительно переменных $x_1, x_2, \ldots, x_n$, т.е.

$$f'(x)=W(x)=\begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \cdots & \dfrac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial f_n}{\partial x_1} & \dfrac{\partial f_n}{\partial x_2} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{bmatrix}$$

или в краткой записи

$$f'(x)=W(x)=\left[\frac{\partial f_i}{\partial x_j}\right] \quad (i,\ j=1,\ 2,\ \ldots,\ n).$$

Поэтому формула (26) может быть записана в следующем виде:

$$f(x^{(k)})+W(x^{(k)})\,\Delta x^{(k)}=0.$$

Если $\det W(x)=\det\left[\dfrac{\partial f_i}{\partial x_j}\right]\neq 0$ det, то $\Delta x^{(k)}=-W^{-1}(x^{(k)})\,f(x^{(k)})$.

Отсюда видно, что метод Ньютона решения системы (23) состоит в построении итерационной последовательности:

$$x^{(k+1)}=x^{(k)}-W^{-1}(x^{(k)})\,f(x^{(k)}) \ (k=0,\ 1,\ 2,\ \ldots,\ n). \qquad (27)$$

Если все поправки становятся достаточно малыми, счёт прекращается. Иначе новые значения x_i используются как приближённые значения корней, и процесс повторяется до тех пор, пока не будет найдено решение или не станет ясно, что получить его не удастся.

Пример 7. Методом Ньютона приближённо найти положительное решение системы уравнений

$$\begin{cases} f_1(x,y,z)=x^2+y^2+z^2-1; \\ f_1(x,y,z)=2x^2+y^2-4z; \\ f_1(x,y,z)=3x^2-4y+z^2. \end{cases}$$

исходя из начального приближения $x_0=y_0=z_0=0{,}5$.

Полагая:

$$x^{(0)}=\begin{bmatrix}0{,}5\\0{,}5\\0{,}5\end{bmatrix},\quad f(x)=\begin{bmatrix}f_1(x,y,z)\\f_2(x,y,z)\\f_3(x,y,z)\end{bmatrix},$$

имеем

$$f(x)=\begin{bmatrix}x^2+y^2+z^2-1\\2x^2+y^2-4z\\3x^2-4y+z^2\end{bmatrix}.$$

Отсюда

$$f\left(x^{(0)}\right)=\begin{bmatrix}0{,}25+0{,}25+0{,}25-1\\0{,}50+0{,}25-2{,}00\\0{,}75-2{,}00+0{,}25\end{bmatrix}=\begin{bmatrix}-0{,}25\\-1{,}25\\-1{,}00\end{bmatrix}.$$

Составим матрицу Якоби

$$W(x)=\begin{bmatrix}\frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z}\\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z}\\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z}\end{bmatrix}=\begin{bmatrix}2x & 2y & 2z\\ 4x & 2y & -4\\ 6x & -4 & 2z\end{bmatrix}.$$

Имеем

$$W\left(x^{(0)}\right)=\begin{bmatrix}1 & 1 & 1\\ 2 & 1 & -4\\ 3 & -4 & 1\end{bmatrix}, \text{ причём } \Delta=\det W\left(x^{(0)}\right)=\begin{vmatrix}1 & 1 & 1\\ 2 & 1 & -4\\ 3 & -4 & 1\end{vmatrix}=-40 .$$

Следовательно, матрица $W\left(x^{(0)}\right)$ – неособенная. Составим обратную ей матрицу

$$W^{-1}\left(x^{(0)}\right)=-\frac{1}{40}\begin{bmatrix}-15 & -5 & -5\\ -14 & -2 & 6\\ -11 & 7 & -1\end{bmatrix}=\begin{bmatrix}\frac{3}{8} & \frac{1}{8} & \frac{1}{8}\\ \frac{7}{20} & \frac{1}{20} & -\frac{3}{20}\\ \frac{11}{40} & -\frac{7}{40} & \frac{1}{40}\end{bmatrix}.$$

По формуле (27) получаем первое приближение

$$x^{(1)}=x^{(0)}-W^{-1}\left(x^{(0)}\right)f\left(x^{(0)}\right)=\begin{bmatrix}0{,}5\\ 0{,}5\\ 0{,}5\end{bmatrix}-\begin{bmatrix}\frac{3}{8} & \frac{1}{8} & \frac{1}{8}\\ \frac{7}{20} & \frac{1}{20} & -\frac{3}{20}\\ \frac{11}{40} & -\frac{7}{40} & \frac{1}{40}\end{bmatrix}\begin{bmatrix}-0{,}25\\ -1{,}25\\ -1{,}00\end{bmatrix}=$$

$$=\begin{bmatrix}0{,}5\\ 0{,}5\\ 0{,}5\end{bmatrix}+\begin{bmatrix}0{,}375\\ 0\\ -0{,}125\end{bmatrix}=\begin{bmatrix}0{,}875\\ 0{,}500\\ 0{,}375\end{bmatrix}.$$

Аналогично находятся дальнейшие приближения. Результаты вычислений приведены в табл. 2.

2. Последовательные приближения корней

i	x	y	z
0	0,5	0,5	0,5
1	0,875	0,5	0,375
2	0,78981	0,49662	0,36993
3	0,78521	0,49662	0,36992

Останавливаясь на приближении $x^{(3)}$, будем иметь:

$$x = 0{,}7852;\quad y = 0{,}4966;\quad z = 0{,}3699.$$

РЕШЕНИЕ УРАВНЕНИЙ СРЕДСТВАМИ MATHCAD

Решение одного уравнения

Для простейших уравнений вида $f(x) = 0$ решение в Mathcad находится с помощью функции *root*.

Root(*f* (*x*1, *x*2, …), *x*1, *a*, *b*): возвращает значение $x1$, принадлежащее отрезку [a, b], при котором выражение или функция $f(x)$ обращается в нуль. Оба аргумента этой функции должны быть скалярами. Функция возвращает скаляр.

***Аргументы*:**

***f* (*x*1, *x*2, …) –** функция, определённая где-либо в рабочем документе, или выражение. Выражение должно возвращать скалярные значения.

***x*1** – имя переменной, которая используется в выражении. Этой переменной перед использованием функции *root* необходимо присвоить числовое значение. Mathcad использует его как начальное приближение при поиске корня.

***a*, *b* –** необязательны; если используются, то должны быть вещественными числами, причём ***a* < *b*.**

Если после многих итераций Mathcad не находит подходящего приближения, то появится сообщение (отсутствует сходимость). Эта ошибка может быть в Can't converge to a solution. инами.

- Уравнение не имеет корней.
- Корни уравнения расположены далеко от начального приближения.

- Выражение имеет локальные max и min между начальным приближением и корнями.
- Выражение имеет разрывы между начальными приближениями и корнями.
- Выражение имеет комплексный корень, но начальное приближение было вещественным.

Чтобы установить причину ошибки, исследуйте график $f(x)$. Он поможет выяснить наличие корней уравнения $f(x) = 0$ и, если они есть, определить приблизительно их значения. Чем точнее выбрано начальное приближение корня, тем быстрее будет *root* сходиться.

Рекомендации по использованию функции *root*.

- Для изменения точности, с которой функция *root* ищет корень, нужно изменить значение системной переменной TOL. Если значение TOL увеличивается, функция *root* будет сходиться быстрее, но ответ будет менее точен. Если значение TOL уменьшается, то функция *root* будет сходиться медленнее, но ответ будет более точен. Чтобы изменить значение TOL определённой точке рабочего документа, используйте определение вида $TOL := 0.01$. Чтобы изменить значение TOL для всего рабочего документа, выберите команду **Математика ⇒ Параметры… ⇒ Переменные ⇒ Допуск сходимости (TOL)**.
- Если два корня расположены близко друг от друга, следует уменьшить TOL, чтобы различить их.
- Если функция $f(x)$ имеет малый наклон около искомого корня, функция $root(f(x), x)$ может *сходиться* к значению r, отстоящему от корня достаточно далеко. В таких случаях для нахождения более точного значения корня необходимо уменьшить значение TOL. Другой вариант заключается в замене уравнения $f(x) = 0$ на $g(x) = 0$:

$$g(x) = \frac{f(x)}{\frac{d}{dx} f(x)} .$$

- Для выражения $f(x)$ с известным корнем a нахождение дополнительных корней $f(x)$ эквивалентно поиску корней уравнения $h(x) = f(x)/(x - a)$. одобный приём полезен для нахождения корней, расположенных близко друг к другу. Проще искать корень выражения $h(x)$, чем пробовать искать другой корень уравнения $f(x) = 0$, выбирая различные начальные приближения.

Нахождение корней полинома. Для нахождения корней выражения, имеющего вид

$$v_n x^n + \ldots + v_2 x^2 + v_1 x + v_0,$$

лучше использовать функцию *polyroots*, нежели *root*. В отличие от функции *root*, функция *polyroots* не требует начального приближения и возвращает сразу все корни, как вещественные, так и комплексные.

Polyroots(*v*): возвращает корни полинома степени n. Коэффициенты полинома находятся в векторе v длины $n + 1$. Возвращает вектор длины n, состоящий из корней полинома.

***Аргументы*:**

v – вектор, содержащий коэффициенты полинома.

Решение систем уравнений

MathCAD даёт возможность решать также и системы уравнений. Максимальное число уравнений и переменных равно 50. Результатом решения системы будет численное значение искомого корня.

Для решения системы уравнений необходимо выполнить следующее.

- Задать начальное приближение для всех неизвестных, входящих в систему уравнений. Mathcad решает систему с помощью итерационных методов.
- Напечатать ключевое слово *Given*. Оно указывает Mathcad, что далее следует система уравнений.
- Введите уравнения и неравенства в любом порядке. Используйте **[Ctrl]=** для печати символа =. Между левыми и правыми частями неравенств может стоять любой из символов <, >, ≥ и ≤.
- Введите любое выражение, которое включает функцию *Find*, например: $a := Find(x, y)$.

***Find*(*z*1, *z*2, ...)**: возвращает точное решение системы уравнений. исло аргументов должно быть равно числу неизвестных.

Ключевое слово *Given*, уравнения и неравенства, которые следуют за ним, и какое–либо выражение, содержащее функцию *Find*, называют *блоком решения уравнений*.

Следующие выражения недопустимы внутри блока решения.

- Ограничения со знаком ≠.
- Дискретный аргумент или выражения, содержащие дискретный аргумент в любой форме.
- Неравенства вида $a < b < c$.

Блоки решения уравнений не могут быть вложены друг в друга, каждый блок может иметь только одно ключевое слово *Given* и имя функции *Find*.

Функция, которая завершает блок решения уравнений, может быть использована аналогично любой другой функции. Можно произвести с ней следующие три действия.

- Вывести найденное решение, напечатав выражение вида

$Find(var1, var2, \ldots) = .$

- Определить переменную с помощью функции *Find*:

$a := Find(x)$ – скаляр;

$var := Find(var1, var2, \ldots)$ – вектор.

Это удобно сделать, если требуется использовать решение системы уравнений в другом месте рабочего документа.

- Определить другую функцию с помощью *Find*:

$f(a, b, c, \ldots) := Find(x, y, z, \ldots)$.

Эта конструкция удобна для многократного решения системы уравнений для различных значений некоторых параметров *a, b, c,…,* непосредственно входящих в систему уравнений.

Сообщение об ошибке `No solution was found. Try changing the guess value or the value of TOL or CTOL.` (решение не найдено) при решении уравнений появляется, когда:

- Поставленная задача может не иметь решения.
- Для уравнения, которое не имеет вещественных решений, в качестве начального приближения взято вещественное число и наоборот.
- В процессе поиска решения последовательность приближений попала в точку локального минимума невязки. Для поиска искомого решения нужно задать различные начальные приближения.
- Возможно, поставленная задача не может быть решена с заданной точностью. Попробуйте увеличить значение TOL.

Приближённые решения. Функция *Minerr* очень похожа на функцию *Find* (использует тот же алгоритм). Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, *Minerr* возвращает это приближение. Функция *Find* в этом случае возвращает сообщение об ошибке. Правила использования функции *inerr* такие же, как и функции *Find*.

Minerr(*z*1, *z*2, ...): Возвращает приближённое решение системы уравнений. Число аргументов должно быть равно числу неизвестных.

Если *Minerr* используется в блоке решения уравнений, необходимо всегда включать дополнительную проверку достоверности результатов.

Символьное решение уравнений

В Mathcad можно быстро и точно найти численное значение корня с помощью функции *root*. Но имеются некоторые задачи, для которых возможности Mathcad позволяют находить решения в символьном (аналитическом) виде.

Решение уравнений в символьном виде позволяет найти точные или приближённые корни уравнения.

- Если решаемое уравнение имеет параметр, то решение в символьном виде может выразить искомый корень непосредственно через параметр. Поэтому вместо того, чтобы решать уравнение для каждого нового значения параметра, можно просто заменять его значение в найденном символьном решении.
- Если нужно найти все комплексные корни полинома со степенью меньше или равной 4, символьное решение даст их точные значения в одном векторе или в аналитическом или цифровом виде.

Команда **Символы ⇒ Переменные ⇒ Вычислить** позволяет решить уравнение относительно некоторой переменной и выразить его корни через остальные параметры уравнения. Чтобы решить уравнение символьно, необходимо:

- напечатать выражение (для ввода знака равенства используйте комбинацию клавиш **[Ctrl]=**);
- выделить переменную, относительно которой нужно решить уравнение, щёлкнув на ней мышью;
- выбрать пункт меню **Символы ⇒ Переменные ⇒ Вычислить**.

Нет необходимости приравнивать выражение нулю. Если Mathcad не находит знака равенства, он предполагает, что требуется приравнять выражение нулю.

Чтобы решить систему уравнений в символьном виде, необходимо выполнить следующее.

- Напечатать ключевое слово *Given*.
- Напечатать уравнения в любом порядке ниже слова *Given*. Удостоверьтесь, что для ввода знака = используется **[Ctrl]=**.
- Напечатать функцию *Find*, соответствующую системе уравнений.
- Нажать **[Ctrl].** (клавиша CTRL, сопровождаемая точкой). Mathcad отобразит символьный знак равенства →.
- Щёлкнуть мышью на функции *Find*.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

***Задание* 1.** Найти *методом половинного деления* отличный от нуля корень уравнения:

1) $x^2 - 5\sin x = 0$;
2) $\sin x - 1/x = 0$;
3) $\lg x - \cos x = 0$.

Корни отделить графически.

***Задание* 2.** Найти, используя *метод хорд*, действительный корень ξ уравнения:

1) $x^3 - 2x^2 + x - 3 = 0$;
2) $x^3 - 2x^2 + 3x - 5 = 0$;
3) $x^4 - 5x^3 + 2x^2 - 10x + 1 = 0$

с точностью $\varepsilon = 10^{-3}$.

***Задание* 3.** Найти, используя *метод Ньютона*, действительный корень ξ уравнения:

1) $x^3 - 2x^2 + x - 3 = 0$;
2) $x^3 - 2x^2 + 3x - 5 = 0$;
3) $x^4 - 5x^3 + 2x^2 - 10x + 1 = 0$

с точностью $\varepsilon = 10^{-3}$.

***Задание* 4.** Найти наибольший положительный корень ξ уравнения:

1) $x^3 + x = 1000$;
2) $4x - 5\ln x = 5$;
3) $e^x - 10x = 0$

с точностью $\varepsilon = 10^{-3}$, используя *метод итераций*.

***Задание* 5.** Систему

$$\begin{cases} 2x_1 + 3x_2 - 4x_3 + x_4 - 3 = 0; \\ x_1 - 2x_2 - 5x_3 + x_4 - 2 = 0; \\ 5x_1 - 3x_2 + x_3 - 4x_4 - 1 = 0; \\ 10x_1 + 2x_2 - x_3 + 2x_4 + 4 = 0 \end{cases}$$

привести к виду, годному для применения *метода итерации*.

***Задание* 6.** Решить системы *методом итерации*.

1) $\begin{cases} 4x_1 + 0{,}24x_2 - 0{,}08x_3 = 8; \\ 0{,}09x_1 + 3x_2 - 0{,}15x_3 = 9; \\ 0{,}04x_1 - 0{,}08x_2 + 4x_3 = 20; \end{cases}$

2) $\begin{cases} 3x_1 + 2x_2 + x_3 = 4; \\ x_1 + x_2 - x_3 = 1; \\ x_1 - 2x_2 + x_3 = 3. \end{cases}$

***Задание* 7.** Решить систему уравнений *методом хорд*:

$$\begin{cases} 2x_1 - x_2 + x_3 = -3; \\ 3x_1 + 5x_2 - 2x_3 = 1; \\ x_1 - 4x_2 + 10x_3 = 0. \end{cases}$$

***Задание* 8.** Приближённо найти положительные решения системы нелинейных уравнений *методом Ньютона*:

$$\begin{cases} f_1(x_1, x_2) \equiv x_1 + 3\lg x_1 - x_2^2 = 0; \\ f_2(x_1, x_2) \equiv 2x_1^2 - x_1x_2 - 5x_1 + 1 = 0. \end{cases}$$

УПРАЖНЕНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

***Упражнение* 1.** Построить график функции $f(x)$ (табл. 3) и приблизительно определить один из корней уравнения. Решить уравнение $f(x) = 0$ с точностью $\varepsilon = 10^{-4}$:

1) с помощью встроенной функции Mathcad *root*;
2) методом Ньютона (касательных), используя функцию *until*;
3) методом итерации, используя функцию *until*.

Определить число итераций в каждом методе с помощью функции *last*.

3. Варианты упражнения 1

№ варианта	$f(x)$	№ варианта	$f(x)$
1	$e^{x-1} - x^3 - x$ $x \in [0,1]$	9	$0{,}25x^3 + x - 2$ $x \in [0,2]$
2	$x - \dfrac{1}{3 + \sin(3{,}6x)}$ $x \in [0,1]$	10	$\arccos\dfrac{1 - x^2}{1 + x^2} - x$ $x \in [2,3]$
3	$\arccos x - \sqrt{1 - 0{,}3x^3}$ $x \in [0,1]$	11	$3x - 4\ln x - 5$ $x \in [2,4]$
4	$\sqrt{1 - 0{,}4x^2} - \arcsin x$ $x \in [0,1]$	12	$e^x - e^{-x} - 2$ $x \in [0,1]$
5	$3x - 14 + e^x - e^{-x}$ $x \in [1,3]$	13	$\sqrt{1 - x} - \mathrm{tg}x$ $x \in [0,1]$

6	$\sqrt{2x^2+1{,}2-\cos x}-1$ $x\in[0,1]$	14	$1-x+\sin x-\ln(1+x)$ $x\in[0,2]$
7	$\cos\left(\frac{2}{x}\right)-2\sin\left(\frac{1}{x}\right)+\frac{1}{x}$ $x\in[1,2]$	15	$x^5-x-0{,}2$ $x\in[1,2]$
8	$0{,}1x^2-x\ln x$ $x\in[1,2]$		

***Упражнение* 2.** Для полинома $g(x)$ (табл. 4) выполнить следующие действия:

1) с помощью команды **Символы ⇒ Коэффициенты полинома** создать вектор V, содержащий коэффициенты полинома;
2) решить уравнение $g(x) = 0$ с помощью функции *polyroots*;
3) решить уравнение символьно, используя команду **Символы ⇒ Переменные ⇒ Вычислить**;
4) разложить на множители, используя **Символы ⇒ Фактор.**

4. Варианты упражнения 2

№ варианта	$g(x)$	№ варианта	$g(x)$
1	$x^4-2x^3+x^2-12x+20$	9	$x^4+x^3-17x^2-45x-100$
2	$x^4+6x^3+x^2-4x-60$	10	$x^4-5x^3+x^2-15x+50$
3	$x^4-14x^2-40x-75$	11	$x^4-4x^3-2x^2-20x+25$
4	$x^4-x^3+x^2-11x+10$	12	$x^4+5x^3+7x^2+7x-20$
5	$x^4-x^3-29x^2-71x-140$	13	$x^4-7x^3+7x^2-5x+100$
6	$x^4+7x^3+9x^2+13x-30$	14	$x^4+10x^3+36x^2+70x+75$
7	$x^4+3x^3-23x^2-55x-150$	15	$x^4+9x^3+31x^2+59x+60$
8	$x^4-6x^3+4x^2+10x+75$		

***Упражнение* 3.** Решить систему линейных уравнений (табл. 5):

1) используя функции *Find*;
2) матричным способом и используя функцию *lsolve*;
3) методом Гаусса;
4) методом итерации.

Оценить погрешность решения методом итерации.

5. Варианты упражнения 3

№ варианта	Система линейных уравнений	№ варианта	Система линейных уравнений
1	$\begin{cases}2x_1+x_2+2x_3+3x_4=8\\3x_1+3x_3=6\\2x_1-x_2+3x_4=4\\x_1+2x_2-x_3+2x_4=4\end{cases}$	5	$\begin{cases}x_1-2x_2+6x_3+x_4=88\\5x_1+2x_3-3x_4=88\\7x_1-3x_2+7x_3+2x_4=181\\3x_1-7x_2+5x_3+2x_4=99\end{cases}$
2	$\begin{cases}x_1+2x_2+3x_3+4x_4=22\\2x_1+3x_2+x_3+2x_4=17\\x_1+x_2+x_3-x_4=8\\x_1-2x_3-3x_4=-7\end{cases}$	6	$\begin{cases}x_1-2x_2-8x_4=-7\\x_1+4x_2-7x_3+6x_4=-8\\x_1+x_2-5x_3+x_4=-10\\2x_1-x_2+2x_4=7\end{cases}$
3	$\begin{cases}9x_1+10x_2-7x_3-x_4=23\\7x_1-x_3-5x_4=37\\5x_1-2x_3+x_4=22\\4x_1+x_2+2x_3+3x_4=26\end{cases}$	7	$\begin{cases}2x_1+2x_2+6x_3+x_4=15\\-x_2+2x_3+x_4=18\\4x_1-3x_2+x_3-5x_4=37\\3x_1-5x_2+x_3-x_4=30\end{cases}$
4	$\begin{cases}6x_1-x_2+10x_3-x_4=158\\2x_1+x_2+10x_3+7x_4=128\\3x_1-2x_2-2x_3-x_4=7\\x_1-12x_2+2x_3-x_4=17\end{cases}$	8	$\begin{cases}4x_1-5x_2+7x_3+5x_4=165\\2x_1+x_2-3x_3-x_4=-15\\9x_1+4x_3-x_4=194\\x_1-x_2-2x_3-3x_4=-19\end{cases}$

№ варианта	Система линейных уравнений	№ варианта	Система линейных уравнений
9	$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = -4 \\ x_1 - 3x_2 - 6x_4 = -7 \\ 2x_2 - x_3 + 2x_4 = 2 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -2 \end{cases}$	13	$\begin{cases} 2x_1 - 3x_3 - 2x_4 = -16 \\ 2x_1 - x_2 + 13x_3 + 4x_4 = 213 \\ 3x_1 + x_2 + 2x_3 + x_4 = 72 \\ x_1 - 12x_3 - 5x_4 = -159 \end{cases}$
10	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$	14	$\begin{cases} 7x_1 + 7x_2 - 7x_3 - 2x_4 = 5 \\ 3x_1 + 4x_2 + 5x_3 + 8x_4 = 60 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 27 \\ 2x_1 - 2x_3 - x_4 = -1 \end{cases}$
11	$\begin{cases} 2x_1 - 8x_2 - 3x_3 - 2x_4 = -18 \\ x_1 - 2x_2 + 3x_3 - 2x_4 = 28 \\ x_2 + x_3 + x_4 = 10 \\ 11x_2 + x_3 + 2x_4 = 21 \end{cases}$	15	$\begin{cases} 6x_1 - 9x_2 + 5x_3 + x_4 = 124 \\ 7x_2 - 5x_3 - x_4 = -54 \\ 5x_1 - 5x_2 + 2x_3 + 4x_4 = 83 \\ 3x_1 - 9x_2 + x_3 + 6x_4 = 45 \end{cases}$
12	$\begin{cases} 2x_1 - x_2 + 4x_3 + x_4 = 66 \\ 2x_2 - 6x_3 + x_4 = -63 \\ 8x_1 - 3x_2 + 6x_3 - 5x_4 = 146 \\ 2x_1 - 7x_2 + 6x_3 - x_4 = 80 \end{cases}$		

***Упражнение* 4.** Преобразовать нелинейные уравнения системы из табл. 6 к виду $f_1(x) = y$ и $f_2(y) = x$. Построить их графики и определить начальное приближение решения. Решить систему нелинейных уравнений

1) с помощью функции *Minerr*;
2) методом Ньютона.

6. Варианты упражнения 4

№ варианта	Система нелинейных уравнений	№ варианта	Система нелинейных уравнений
1	$\begin{cases} \sin x + 2y = 2, \\ \cos(y-1) + x = 0{,}7. \end{cases}$	4	$\begin{cases} \cos(x+0{,}5) + y = 0{,}8, \\ \sin y - 2x = 1{,}6. \end{cases}$
2	$\begin{cases} \sin(x+0{,}5) - y = 1, \\ \cos(y-2) + x = 0. \end{cases}$	5	$\begin{cases} \sin(x-1) = 1.3 - y, \\ x - \sin(y+1) = 0.8. \end{cases}$
3	$\begin{cases} \cos x + y = 1{,}5, \\ 2x - \sin(y-0{,}5) = 1. \end{cases}$	6	$\begin{cases} \cos(x+0{,}5) + y = 1, \\ \sin y - 2x = 2. \end{cases}$
7	$\begin{cases} -\sin(x+1) + y = 0{,}8, \\ \sin(y-1) + x = 1{,}3. \end{cases}$	12	$\begin{cases} \cos(x-2) + y = 0, \\ \sin(y+0{,}5) - x = 1. \end{cases}$
8	$\begin{cases} \sin(x) - 2y = 1, \\ \sin(y-1) + x = 1{,}3. \end{cases}$	13	$\begin{cases} \cos(x+0{,}5) + y = 1, \\ \sin(y+0{,}5) - x = 1. \end{cases}$
9	$\begin{cases} \sin y + x = -0{,}4, \\ 2y - \cos(x+1) = 0. \end{cases}$	14	$\begin{cases} \sin(x) - 2y = 1, \\ \cos(y+0{,}5) - x = 2. \end{cases}$
10	$\begin{cases} \sin(x+2) - y = 1{,}5, \\ \cos(y-2) + x = 0{,}5. \end{cases}$	15	$\begin{cases} 2y - \sin(x-0{,}5) = 1, \\ \cos(y) + x = 1{,}5. \end{cases}$
11	$\begin{cases} \cos(x+0{,}5) - y = 2, \\ \sin y - 2x = 1. \end{cases}$		

***Упражнение* 5.** Символьно решить системы уравнений:

$$\begin{cases} 3x + 4\pi y = a; \\ 2x + y = b; \end{cases}$$

$$\begin{cases} 2y + \pi z = a; \\ \pi z - z = b; \\ 3y + x = c. \end{cases}$$

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие методы решения нелинейных уравнений вам известны?
2. В каких случаях необходимо использовать итерационные методы?
3. Каким условиям должна соответствовать функция $f(x)$ и что они гарантируют?
4. Что значит решить уравнение итерационным методом?
5. Из каких этапов состоит задача нахождения нуля функции $f(x)$ итерационным методом?
6. Назовите способы отделения корней.
7. В чём состоит итерационный процесс?
8. В чём сущность метода половинного деления?
9. В чём сущность метода хорд?
10. Какой из концов отрезка $[a, b]$ в методе хорд считается неподвижным?
11. Сформулируйте условие окончания итерационного процесса в методе хорд.
12. В чём сущность метода Ньютона?
13. Как выбрать начальное приближение для метода Ньютона?
14. Как в Mathcad организовать итерационный процесс?
15. Что влияет на скорость сходимости итерационного процесса?
16. В чём сущность метода итерации, как ещё называют этот метод?
17. Какие виды итерационных процессов вам известны?
18. Сформулируйте достаточные условия сходимости метода итерации.
19. Назовите точные методы решения систем линейных уравнений.
20. Какие функции Mathcad используются для их реализации?
21. Сформулируйте достаточные условия сходимости метода итерации для систем линейных уравнений.
22. Какие виды норм матриц вам известны и как их вычислять?
23. Назовите особенности метода Зейделя.
24. В чём сущность метода Ньютона для решения систем нелинейных уравнений?
25. Какой должна быть матрица Якоби?
26. Когда можно прекратить вычисления по методу Ньютона?
27. Какие функции для решения одного уравнения в Mathcad вы знаете?
28. В каких случаях Mathcad не может найти корень уравнения?
29. Как изменить точность, с которой функция *root* ищет корень?
30. Назовите функции для решения систем уравнений в Mathcad и особенности их применения.
31. Дайте сравнительную характеристику функциям *Find* и *Minerr*.
32. Как символьно решить уравнение или систему уравнений в Mathcad?
33. Назовите особенности использования символьного решения уравнений.

СПИСОК ЛИТЕРАТУРЫ

1. Mathcad 6.0 Plus. Финансовые, инженерные и научные расчёты в среде Windows 95 / пер. с англ. – М. : Информационно-издательский дом «Филинъ», 1996. – 712 с.
2. Амосов, А.А. Вычислительные методы для инженеров / А.А. Амосов и др. – М. : Высшая школа, 1994.
3. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М. : Наука, 1987.
4. Боглаев, Ю.П. Вычислительная математика и программирование / Ю.П. Боглаев. – М. : Высшая школа, 1990.
5. Воробьева, Г.Н. Практикум по вычислительной математике / .Н. Воробьева, А.Н. Данилова. – М. : Высшая школа, 1990. – 207 с.
6. Демидович, Б.П. Основы вычислительной математики / Б.П. Демидович, И.А. Марон. – М. : Наука, 1970. – 664 с.
7. Дьяконов, В.П. Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ / В.П. Дьяконов. – М. : Наука, 1987. – 240 с.
8. Дьяконов, В.П. Справочник по MathCAD PLUS 6.0 PRO / В.П. Дьяконов. – М. : СК Пресс, 1997. – 336 с.
9. Очков, В.Ф. Mathcad 7 Pro для студентов и инженеров / В.Ф. Очков. – М. : КомпьютерПресс, 1998. – 384 с.
10. Плис, А.И. Лабораторный практикум по высшей математике / А.И. Плис, Н.А. Сливина. – М. : Высшая школа, 1994. – 416 с.
11. Турчак, Л.И. Основы численных методов / Л.И. Турчак. – М. : Наука, 1987. – 320 с.
12. Шуп Терри, Е. Прикладные численные методы в физике и технике / Е. Шуп Терри. – М. : Высшая школа, 1990. – 254 с.