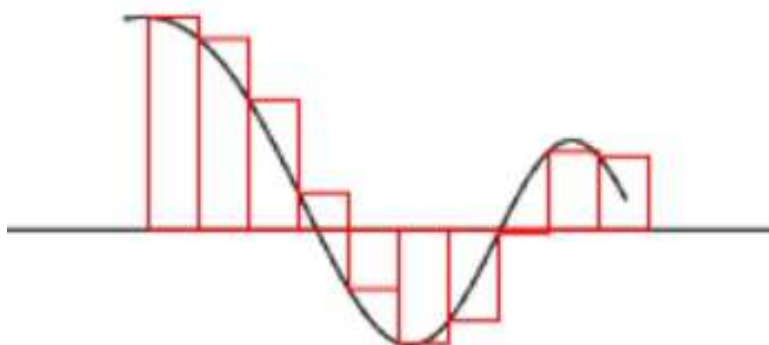


Владимирский государственный университет

Н. А. МАЛОВА

**ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ
ИНЖЕНЕРНО-ТЕХНИЧЕСКИХ
ЗАДАЧ В СТРОИТЕЛЬСТВЕ**

Учебно-практическое пособие



Владимир 2022

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Н. А. МАЛОВА

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ИНЖЕНЕРНО-ТЕХНИЧЕСКИХ ЗАДАЧ В СТРОИТЕЛЬСТВЕ

Учебно-практическое пособие

Электронное издание



Владимир 2022

ISBN 978-5-9984-1487-9

© ВлГУ, 2022

УДК 519.6
ББК 22.19

Рецензенты:

Кандидат технических наук, доцент
доцент кафедры строительных конструкций
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых
Т. Н. Яшкова

Кандидат технических наук
начальник проектно-сметного отдела ООО «Климат-Сервис»
А. А. Суцинин

Издается по решению редакционно-издательского совета ВлГУ

Малова, Н. А.

Численные методы решения инженерно-технических задач в строительстве [Электронный ресурс] : учеб.-практ. пособие / Н. А. Малова ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2022. – 128 с. – ISBN 978-5-9984-1487-9. – Электрон. дан. (5,41 Мб). – 1 электрон. опт. диск (CD-ROM). – Систем. требования: Intel от 1,3 ГГц ; Windows XP/7/8/10 ; Adobe Reader ; диск-код CD-ROM. – Загл. с титул. экрана.

Представлена методика решения типовых задач в инженерных расчетах в программном комплексе MathCAD. Рассмотрены основные функции программы, необходимые для решения задач. Приведены примеры решений.

Предназначено для магистрантов направления подготовки 08.04.01 – Строительство заочной формы обучения с применением дистанционных технологий.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Ил. 98. Табл. 3. Библиогр.: 10 назв.

ISBN 978-5-9984-1487-9

© ВлГУ, 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
Глава 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	7
1.1. Основные понятия численных методов	7
1.2. Численное решение нелинейных уравнений	11
1.3. Операции с матрицами. Численное решение системы линейных уравнений	15
1.4. Численное решение обыкновенных дифференциальных уравнений	23
1.5. Метод конечных элементов (МКЭ): основные понятия....	28
Глава 2. ЗНАКОМСТВО С ПРОГРАММОЙ MATHCAD	32
2.1. Назначение программы	32
2.2. Интерфейс рабочей области	32
2.3. Основные соглашения при вводе данных и выполнении математических операций	33
2.4. Символьные вычисления	36
2.5. Построение двумерных графиков.....	39
2.6. Блок решения	44
2.7. Элементы программирования	45
Глава 3. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ, РЕАЛИЗОВАННЫЕ В ПРОГРАММЕ MATHCAD.....	48
3.1. Численное решение нелинейных уравнений в MathCAD	48
3.2. Операции с матрицами	51
3.3. Решение систем линейных уравнений.....	53

3.4. Решение обыкновенных дифференциальных уравнений.....	54
3.5. Реализация решения МКЭ в системе MathCAD	57
ПРАКТИЧЕСКИЕ ЗАДАНИЯ	60
ПРИМЕРЫ РЕШЕНИЙ С ПОМОЩЬЮ МАТНСАД 14.....	69
ПРИМЕРЫ РЕШЕНИЙ С ПОМОЩЬЮ РТС МАТНСАД	87
ЗАКЛЮЧЕНИЕ.....	126
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	127

ВВЕДЕНИЕ

Важный этап при строительстве инженерных сооружений – их проектирование. При этом выполняется большой объем расчетных работ: арифметические вычисления, определение интегралов, решение систем алгебраических уравнений, решение дифференциальных уравнений и др. Цель освоения дисциплины «Численные методы решения инженерно-технических задач в строительстве» – приобретение навыков применения методов численного решения типовых задач с привлечением программного комплекса MathCAD.

В пособии рассмотрены теоретические основы численных методов решения популярных задач, представлена их практическая реализация в среде MathCAD. Приведены практические задания, способствующие освоению начал численных методов, приобретению навыков работы в программе MathCAD и самостоятельного решения ряда инженерных задач. Подробно рассмотрены примеры численного решения предложенных заданий.

Самостоятельная работа по исследованию функции помогает разобраться в задачах технической и строительной механики, сопротивления материалов, требующих анализа эпюр и нахождения экстремальных значений усилий и перемещений. Нелинейные уравнения и методы итераций для их решения позволяют вычислять объемные напряженные состояния в конструкциях, решать задачи на устойчивость и др. Решение систем линейных уравнений методом сил и методом перемещений используется в задачах строительной механики. Задание по вычислению перемещений от действия продольной нагрузки заключительное. Оно позволяет закрепить полученные навыки программирования, составления матриц, построения графиков нескольких функций в одной системе координат и их форматирования, а также знакомит с методом конечных элементов в решении задачи.

В главе 1 «Теоретические сведения» рассмотрены основные положения численных методов, необходимые для выполнения заданий.

В главе 2 «Знакомство с программой MathCAD» описано функционирование современных пакетов РТС MathCAD: принципы ввода

данных, функций, символов и операторов, принципы обработки информации и тому подобное, а также рассмотрены отличия в работе от более ранних версий на примере MathCAD 14.

В главе 3 «Численные методы решения задач, реализованные в программе MathCAD» приведены приемы численного решения задач и представлена их реализация в MathCAD.

Даны практические задания, требования к выполнению заданий, варианты заданий и методические указания к их решению. Выполненные работы представляются преподавателю в электронном виде.

Рассмотрены примеры решения заданий, реализованные в программе MathCAD 14, а также в программе PTC MathCAD Express.

Глава 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Основные понятия численных методов

Численные методы – мощное математическое средство для решения научно-технических и инженерных задач. Там, где невозможно точное аналитическое решение, применяются численные методы для реализации математических вычислений в решении сложных физико-технических, экономических и других прикладных задач. Что же такое численное решение?

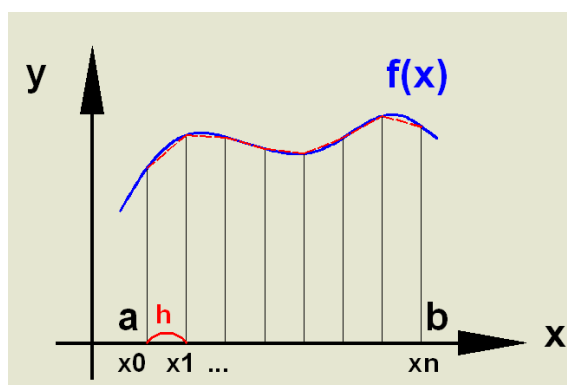


Рис. 1.1

Общим для всех численных методов является переход от функции непрерывного аргумента к функции дискретного аргумента (рис. 1.1). В численном решении производные заменяются их разностными аналогами, интегралы – суммами, функции – многочленами. Например, производная заменяется разностью функций, в которой Δx – шаг аргумента:

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

Обычно в численном решении шаг аргумента обозначается символом h .

Сравним нахождение производной при помощи формулы (аналитическое решение) с численным методом. Пусть дана функция

$$f(x) = x^3 + 2x$$

Найдем значение производной при $x = 2$ аналитически:

$$f(x) = 3x^2 + 2 \quad f(2) = 14$$

Точность результата в численном решении зависит в данном случае от величины шага h :

$$\text{– при } h = 0.1 \quad f(2) = \frac{f(x + 0.1) - f(x)}{0.1} = 14.61$$

$$- \text{при } h = 0.01 \quad f(2) = \frac{f(x + 0.01) - f(x)}{0.01} = 14.06$$

$$- \text{при } h = 0.001 \quad f(2) = \frac{f(x + 0.001) - f(x)}{0.001} = 14.006$$

Поскольку численный метод не предполагает абсолютно точного решения (хотя результат может быть и точным), существует решение в первом (грубом) приближении, затем во втором (или третьем), более точном.

Численное решение задачи сводится к построению **математической модели** (математической формулировке), которая описывает основные законы физического процесса. Решение задачи записывается в виде последовательности элементарных арифметических и логических операций, называемой **алгоритмом**. Логические операции контролируют ход решения: позволяют сравнивать полученные данные при помощи операторов (больше, меньше, равно, или), т. е. задают определенное условие, которому следует ход решения. Алгоритм изображается в виде **блок-схемы** (рис. 1.2, слева). Для решения алгоритма на компьютере составляется **программа** на машинном языке (**программирование**).

На первом этапе составления программы происходит ее **отладка** (проверка наличия ошибок) на тестовых задачах с уже известным решением для получения достоверности результатов. Как правило, вычисления производят в определенном интервале аргумента с нарастанием или убыванием шага, при этом для вычислений используется одна и та же формула, но с приращением аргумента. Такое вычисление на каждом шаге аргумента называется **итерацией**. Последовательность итераций составляет **цикл**. Он заканчивается на основании некоторого условия (логической операции) (рис. 1.2, справа).

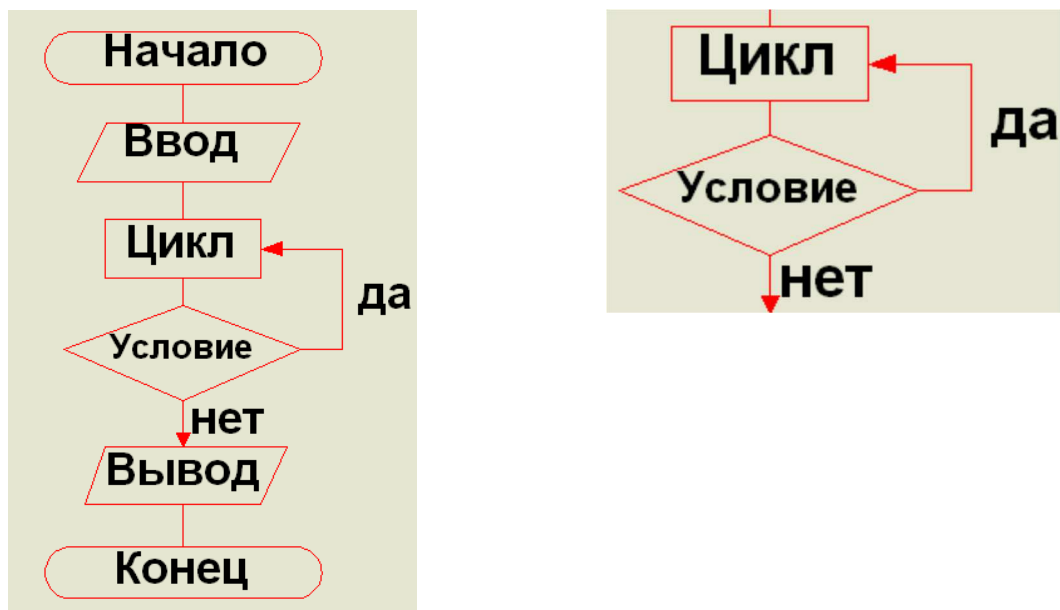


Рис. 1.2

Рассмотрим на примере популярной и довольно простой задачи, как формируется цикл программы. Суть задачи такова. Участники совещания заходили в зал по одному, и каждый пожимал руку всем, кто там уже находился. Всего получилось 105 рукопожатий. Вопрос: сколько было участников совещания?

С чего начинается составление любой программы? С ввода начальных данных, определения неизвестных и переменных, которые меняются (нарастают или убывают) в процессе счета. В нашей задаче в качестве неизвестного выступает количество участников. Примем это количество за N (целое число). Но мы знаем, что вначале $N = 1$ (первый вошедший в зал участник), количество рукопожатий $R = 0$, так как первому вошедшему некому пожать руку. Далее формируем цикл, который состоит из двух итераций (простейших арифметических операций). Известно, что участники заходят по одному, значит, переменная N изменяется на единицу: $N = N + 1$. Количество рукопожатий с каждым прибывшим увеличивается на количество участников за вычетом единицы (исключаем самого первого), тогда $R = R + (N - 1)$. Счет рукопожатий идет до тех пор, пока не сравняется с заданным конечным значением (в нашем примере это 105). Тогда цикл завершится, а значение N в последней итерации и будет ответом. Если просчитать все итерации, получим 15 участников. Алгоритм задачи представлен на рис. 1.3.

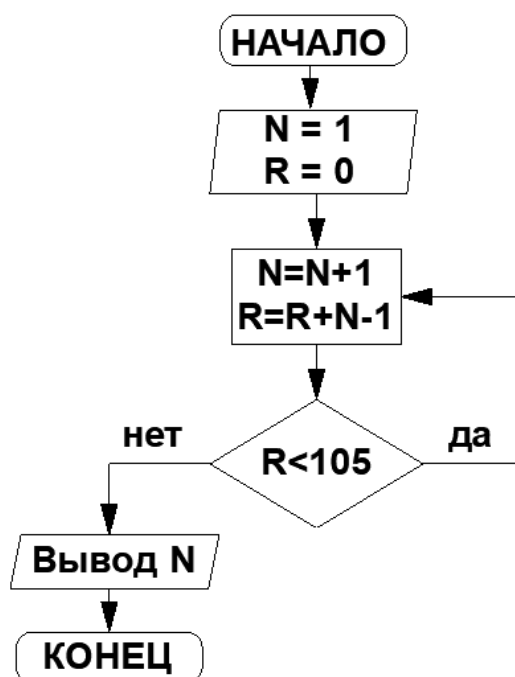


Рис. 1.3

Как правило, одна и та же задача может иметь несколько значений начальных данных (в нашем примере количество рукопожатий может быть любым), тогда условием цикла будет не число 105, а некое X , объявленное в алгоритме внутренней переменной.

В более сложных расчетах при замене функции на дискретную возникает погрешность, называемая *погрешностью метода*. Таким образом, основное требование, предъявляемое к вычислительному алгоритму, – точность. На практике величину ε (вычисление с заданной погрешностью) выбирают, учитывая возможность вычислительной машины.

При любых вычислениях в результате округлений ошибки округления могут либо нарастать, либо затухать. Если в процессе вычислений ошибки округления неограниченно возрастают, такой *алгоритм* называется *неустойчивым*, если ошибки округления не накапливаются – алгоритм *устойчивый*.

Корректность алгоритма означает, что задача разрешима при любых входных данных (например, в результате производимых вычислений исключено деление на ноль или извлечение квадратного корня из отрицательного числа и т. п.). Задача поставлена некорректно, если ее решение неустойчиво относительно входных данных (т. е. малому их изменению могут соответствовать большие изменения решения).

Сходимость метода – близость получаемого численного решения к истинному результату. При дискретных методах под сходимостью понимается стремление значений численного решения к соответствующим значениям исходной задачи при стремлении к нулю шага интегрирования.

Таким образом, для получения решения задачи с необходимой точностью ее постановка должна быть корректной, а используемый численный метод должен обладать устойчивостью и сходимостью.

1.2. Численное решение нелинейных уравнений

Нелинейные уравнения вида $F(x) = 0$ делятся на два класса: алгебраические и трансцендентные. **Алгебраические уравнения** содержат только алгебраические функции (например, многочлен или полиномиальную функцию): $5x^4 + 3x^2 - 9 = 0$. **Трансцендентные уравнения** содержат тригонометрические, показательные, логарифмические функции: $\sin(x) + 2\cos(x^2) = 0$.

Методы решения нелинейных уравнений могут быть прямыми и итерационными. **Прямые методы** позволяют записать корни в виде некоторого конечного соотношения (формулы). Например, по известной формуле вычисляются корни квадратного уравнения. Однако очень часто на практике прямыми методами решить уравнения не удается. В этом случае используют **итерационные методы**, т. е. методы последовательных приближений.

Алгоритм решения итерационным методом состоит из двух этапов:

- отделения корня;
- уточнения корня.

Под отделением корня понимают определение промежутка, содержащего один и только один корень уравнения. Одна из точек этого промежутка принимается за начальное приближение корня. Способ уточнения корня до заданной степени точности зависит от метода решения.

Для отделения корня наиболее нагляден графический способ. Пример показан на рис. 1.4.

$$f(x) = x^3 - 3x^2 + 2$$

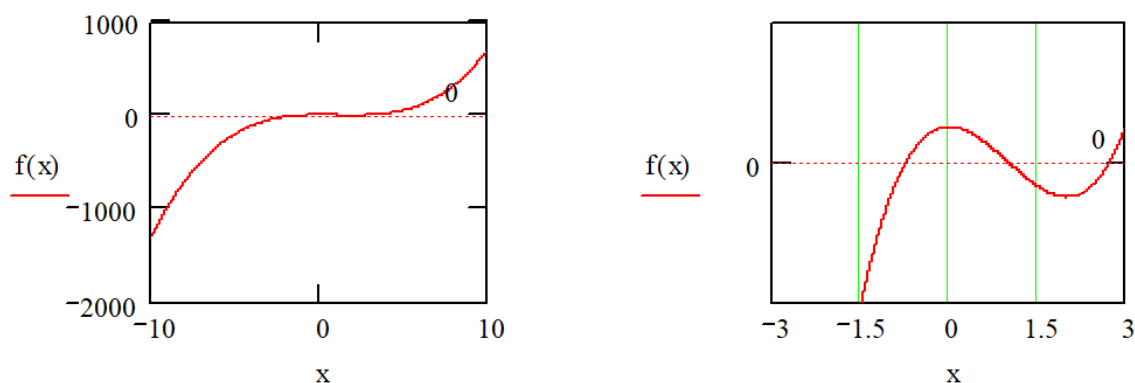


Рис. 1.4

На графике справа (см. рис. 1.4) легко определить, что данная функция имеет три корня в промежутках $[-1.5, 0]$; $[0, 1.5]$; $[1.5, 3]$. В ряде случаев приближенное значение корня может быть определено из физических соображений, если решение связано с конкретной прикладной задачей.

Уточнение корня предполагает вычисление значений корней уравнения в каждом из интервалов с заданной погрешностью.

Многие методы опираются на теорему, известную из математического анализа.

Если функция $f(x)$, определяющая уравнение $f(x) = 0$, на концах отрезка $[a, b]$ принимает значения разных знаков, т. е. $f(a) \times f(b) < 0$, то на этом отрезке содержится по крайней мере один корень уравнения. Если же функция $f(x)$ непрерывна и дифференцируема и ее производная сохраняет знак внутри отрезка $[a, b]$, то на этом отрезке находится только один корень уравнения. В случае, если на концах интервала функция имеет одинаковые знаки, корни либо отсутствуют, либо их четное число.

Рассмотрим некоторые численные методы решения нелинейных уравнений.

Метод половинного деления. Пусть на интервале $[a, b]$ функция $f(x)$ имеет единственный корень. Будем сужать этот интервал, так чтобы внутри всегда находился искомый корень. Разобьем интервал пополам: $c = \frac{a+b}{2}$. В каждом подынтервале вычислим произведения конечных значений функции: $f(a) \cdot f(c)$ и $f(c) \cdot f(b)$. Оставим ту половину интервала, в которой произведение значений функции отрицательно (рис. 1.5).

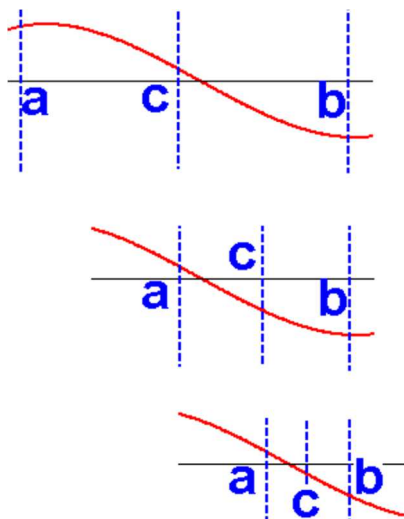


Рис. 1.5

Затем оставшийся интервал вновь делим пополам и так далее, пока длина интервала не станет меньше заданной точности вычисления: $|a_i - b_i| < \varepsilon$.

Метод довольно прост и надежен, всегда сходится, хотя и медленно, устойчив к ошибкам округления. Приведем алгоритм метода (рис. 1.6).

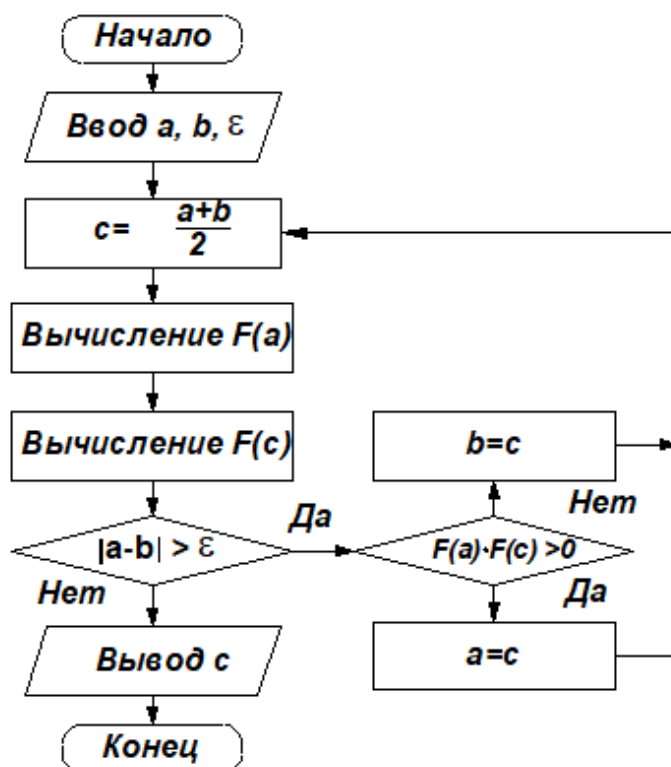


Рис. 1.6

Метод касательных (Ньютона). Пусть имеется функция $f(x)$, меняющая знак на некотором интервале $[a, b]$ (т. е. имеющая корень в данном интервале). Если на этом интервале функция не меняет кривизны, то очевидно, что первая и вторая производные на этом отрезке одного знака. Если провести касательную к кривой в любой точке интервала (рис. 1.7), то точка пересечения касательной x_0 пройдет через данный интервал и будет первым приближением искомого корня x .

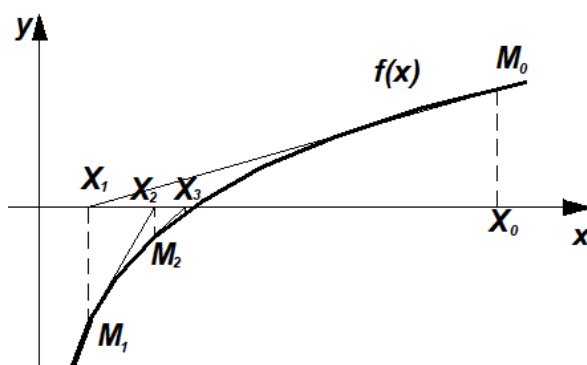


Рис. 1.7

Согласно уравнению касательной

$$y - f(x_0) = df(x_0) \cdot (x - x_0)$$

получим первое приближение корня x_1 :

$$x_1 = x_0 - \frac{f(x_0)}{df(x_0)}$$

Затем, строя касательную в точке x_1 , ищем второе приближение x_2 и т. д. Точность нахождения корня x определяется заранее заданной погрешностью. Если касательную провести на промежутке, где первая и вторая производные имеют противоположные знаки, то $x^{(k)}$ может выйти за пределы интервала $[a, b]$ и ухудшить приближения. Для окончания итерационного процесса может быть использовано условие $|x^{(k+1)} - x^{(k)}| < \varepsilon$, где погрешность ε задана заранее.

Таким образом, на каждой итерации объем вычислений больше, чем в предыдущем методе, однако скорость сходимости значительно выше. В этом методе нет необходимости задавать интервал поиска, достаточно указать некоторое начальное значение x_0 . Недостаток метода состоит в том, что он расходится в тех областях, где $f'(x) \approx 0$. Алгоритм метода Ньютона показан на рис. 1.8.

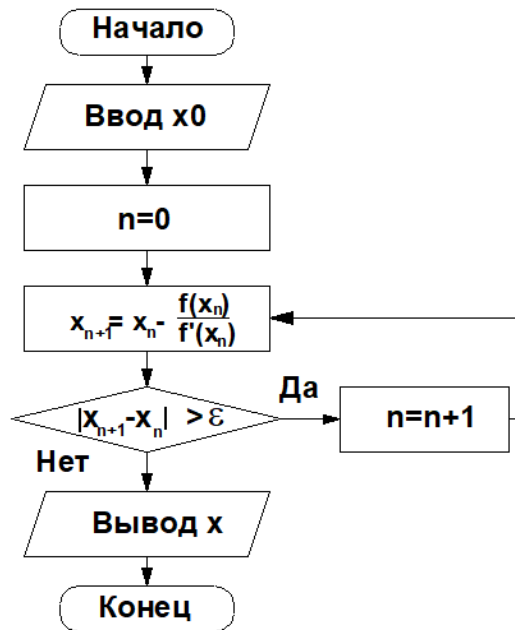


Рис. 1.8

Особенность вычисления корней многочленов (полиномов) вида $a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0 = 0$ – заведомо известное их количество, которое равно степени полинома n . Корни могут быть как действительные, так и комплексные. В отличие от трансцендентных уравнений, полином более чувствителен к погрешностям.

1.3. Операции с матрицами. Численное решение системы линейных уравнений

Матрица – упорядоченный массив, каждый элемент этого массива имеет свой индекс. Матрицы подразделяются:

– на матрицу-вектор (один столбец и n строк):

$$V = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

– матрицу-скаляр (однострочную матрицу):

$$A = (1 \ 2 \ 3)$$

– двумерную матрицу-массив, состоящую из n строк и m столбцов:

$$M = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Если число строк n равно числу столбцов m , **матрица** называется **квадратной**, а число n – **порядком матрицы**. Нижние индексы элемента матрицы означают ее порядковый номер. Первый индекс означает номер строки, второй – номер столбца. Как правило, нумерация ячеек в матрице начинается с единицы, счет строк – сверху вниз, столбцов – слева направо. Для квадратной матрицы M , показанной выше, порядковые номера элементов задаются следующим образом:

$$M_{1,1} = 1, M_{1,2} = 2, M_{2,1} = 3, M_{2,2} = 4$$

Главной диагональю матрицы называется диагональ, проведённая из левого верхнего угла матрицы в правый нижний. Побочной диагональю матрицы считается диагональ, проведённая из левого нижнего угла матрицы в правый верхний. Для квадратной матрицы это означает, что $M_{1,1}$ и $M_{2,2}$ расположены на главной диагонали, а $M_{1,2}$ и $M_{2,1}$ – на побочной.

Для элемента матрицы-вектора индекс записывается только один (номер строки), индекс строчной матрицы – двойной, также как в матрице-массиве, только первый индекс всегда равен нулю.

Среди квадратных матриц важную роль занимают так называемые диагональные матрицы, т. е. такие, у которых отличны от нуля лишь элементы, стоящие вдоль диагонали:

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

Если все числа, стоящие вдоль диагонали, равны между собой, **матрица** называется **скалярной**; в случае, если они равны единице, – **единичной**:

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Матрица, все элементы которой равны нулю, называется **нулевой**.

Транспонированная матрица – матрица, в которой переставлены строки и столбцы.

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad M^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

Матрица-вектор транспонированная является скалярной, поэтому для ее элементов следует задавать двойной индекс. Квадратная матрица равна транспонированной, если она симметрична, т. е. $a_{ij} = a_{ji}$:

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{pmatrix} \quad M^T = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{pmatrix}$$

Определитель матрицы. У каждой квадратной матрицы есть число, называемое определителем (детерминантом). Обозначается $|M|$, или $\det M$, или Δ . Неквадратная матрица определителя не имеет. Матрица, определитель которой равен нулю, называется **вырожденной**, или **сингулярной**. Порядок квадратной матрицы – максимальное количество её строк/столбцов. Определитель матрицы второго порядка вычисляется по формуле

$$M = \begin{pmatrix} a1 & b1 \\ a2 & b2 \end{pmatrix}, \det M = a1 \cdot b2 - a2 \cdot b1$$

Определитель матрицы третьего порядка:

$$M = \begin{pmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{pmatrix}, \det M = a1 \cdot \begin{vmatrix} b2 & c2 \\ b3 & c3 \end{vmatrix} - b1 \cdot \begin{vmatrix} a2 & c2 \\ a3 & c3 \end{vmatrix} + c1 \cdot \begin{vmatrix} a2 & b2 \\ a3 & b3 \end{vmatrix}$$

Здесь элементы определителя умножаются на алгебраические дополнения (миноры элементов $a1, b1, c1$). Алгебраические дополнения элемента матрицы получаются при вычеркивании ее строки и столбца.

Аналогично для матриц высших порядков определитель получается путем произведения элементов $a, b, c, \dots$ какой-либо строки или какого-либо столбца на их алгебраические дополнения:

$$\det = a1 \cdot A1 - b1 \cdot B1 + c1 \cdot C1 - d1 \cdot D1 \dots$$

Сложение матриц. Суммой двух прямоугольных матриц A и B , имеющих одинаковое число строк и столбцов, называется матрица C ,

элементы которой равны суммам соответствующих элементов матриц А и В:

$$A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2m} + b_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \dots & a_{nm} + b_{nm} \end{pmatrix} \quad A + B = B + A$$

Например:

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

Произведение матриц. Произведением матрицы А на число α называется матрица С, все элементы которой получены из элементов матрицы А, умноженных на число α :

$$A = \begin{pmatrix} 1 & -2 & 2 \\ 2 & -1 & 3 \\ 3 & 2 & 1 \end{pmatrix} \quad 3 \cdot A = \begin{pmatrix} 3 & -6 & 6 \\ 6 & -3 & 9 \\ 9 & 6 & 3 \end{pmatrix}$$

Произведение матриц А и В возможно при условии, что число столбцов матрицы А равно числу строк матрицы В. Новая матрица С получится следующим образом: элемент i -й строки j -го столбца равен сумме произведений элементов i -й строки матрицы А на соответствующие элементы j -го столбца матрицы В. В матрице С число строк определяется по числу строк первой матрицы, а число столбцов – по числу столбцов второй матрицы (таким образом, переместительный закон при перемножении матриц не соблюдается):

$$A = \begin{pmatrix} 1 & -2 & 2 \\ 2 & -1 & 3 \\ 3 & 2 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 4 & 3 \\ 2 & 1 \\ 1 & 5 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 2 & 11 \\ 9 & 20 \\ 17 & 16 \end{pmatrix}$$

Первый элемент (1-я строка, 1-й столбец) матрицы С получим, умножив последовательно элементы 1-й строки матрицы А на элементы 1-го столбца матрицы В:

$$1 \cdot 4 - 2 \cdot 2 + 2 \cdot 1 = 2$$

Элемент 1-й строки 2-го столбца получим, умножив 1-ю строку матрицы А на 2-й столбец матрицы В:

$$1 \cdot 3 - 2 \cdot 1 + 2 \cdot 5 = 11$$

Аналогично формируем 2-ю строку матрицы С, последовательно умножая 2-ю строку матрицы А на столбцы матрицы В:

$$2 \cdot 4 - 1 \cdot 2 + 3 \cdot 1 = 9; \quad 2 \cdot 3 - 1 \cdot 1 + 3 \cdot 5 = 20$$

Третья строка:

$$3 \cdot 4 + 2 \cdot 2 + 1 \cdot 1 = 17; \quad 3 \cdot 3 + 2 \cdot 1 + 1 \cdot 5 = 16$$

Попытка перемножить матрицы В и А не приведет ни к какому результату, поскольку число столбцов матрицы В не равно числу строк матрицы А. Если обе матрицы квадратные, с равным количеством строк и столбцов, результат перемножения А и В и В и А будет различен:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 4 \\ 2 & 3 \end{pmatrix} \quad A \cdot B = \begin{pmatrix} 7 & 10 \\ 17 & 24 \end{pmatrix} \quad B \cdot A = \begin{pmatrix} 15 & 22 \\ 11 & 16 \end{pmatrix}$$

Обратная матрица. Всякая невырожденная квадратная матрица ($|A| \neq 0$) имеет обратную матрицу М, если $A \cdot M = E$ (Е – единичная матрица). Матрица, обратная матрице А, обозначается A^{-1} . Для матрицы 2-го порядка элементы, стоящие на главной диагонали, меняем местами, а у элементов побочной диагонали меняем знак на противоположный, затем каждый элемент делим на определитель и получаем обратную матрицу:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad A^{-1} = \begin{pmatrix} \frac{a_{22}}{\Delta} & \frac{-a_{12}}{\Delta} \\ \frac{-a_{21}}{\Delta} & \frac{a_{11}}{\Delta} \end{pmatrix}$$

Пример:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad \Delta = 1 \cdot 4 - 2 \cdot 3 = -2$$

$$A^{-1} = \begin{pmatrix} \frac{4}{-2} & \frac{-2}{-2} \\ \frac{-3}{-2} & \frac{1}{-2} \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1.5 & -0.5 \end{pmatrix}$$

Решение системы линейных уравнений. Система линейных уравнений вида

$$a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n = b_1$$

$$a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n = b_2$$

.....

$$a_{n1} \cdot x_1 + a_{n2} \cdot x_2 + \dots + a_{nn} \cdot x_n = b_n$$

записывается в матричном формате так:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \quad \rightarrow \quad \vec{B} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix},$$

где A – матрица системы (или матрица коэффициентов), B – матрица-вектор правых частей системы. Правая часть не содержит неизвестных.

Решение системы уравнений – вектор неизвестных X , подстановка которых в уравнения преобразует их в тождество.

$$\vec{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

Матричная запись системы уравнений следующая: $A \cdot \vec{X} = \vec{B}$.

Системы линейных уравнений решаются точными и численными (приблизительными) методами. Решение может быть единственным, если определитель матрицы системы $\det A \neq 0$. В противном слу-

чае система называется вырожденной и либо имеет множество решений (при $B = 0$), либо не имеет решения ($|B| \neq 0$).

К точным решениям следует отнести **метод Крамера**:

$$x_j = \frac{\det A_j}{\det A}, \quad j = 1..n,$$

где $\det A_j$ – определитель матрицы, получаемый заменой j -го столбца матрицы A столбцом правых частей. Такому решению требуется $n \cdot n!$ арифметических операций (умножений).

Метод Гаусса также относится к точным методам и называется методом исключения. Однако при решении методом Гаусса неизбежны округления значений при вычислении, которые приводят к погрешностям. Таким образом, к точным методам его относят условно. Метод Гаусса следует интерпретировать как метод, в котором первоначально матрица приводится к верхней треугольной форме (прямой ход), а далее – к единичной (обратный ход).

Приведем матрицу системы A к верхней треугольной форме (т. е. к такой, в которой ниже главной диагонали все элементы матрицы равны нулю). Для этого вычтем из второго уравнения системы первое, умноженное на такое число, при котором коэффициент при x_1 обратится в нуль. То же проделываем с остальными уравнениями. В результате все коэффициенты 1-го столбца, лежащие ниже главной диагонали, обратятся в нуль. Затем, используя второе уравнение, обратим в нуль соответствующие коэффициенты 2-го столбца. Последовательно продолжая этот процесс, приведем матрицу системы к верхней треугольной форме. Процесс приведения матрицы к верхней треугольной форме называется **прямым ходом метода Гаусса**. Вычисление неизвестных – **обратный ход** метода.

Пусть матрица системы имеет верхнюю треугольную форму, поэтому $a_{ij} = 0$ при $i > j$, т. е. все элементы ниже главной диагонали равны нулю. Тогда из последнего уравнения сразу определяем X_n . Подставляя X_n в предпоследнее уравнение, находим x_{n-1} и т. д.

Общие формулы имеют следующий вид:

$$x_k = \frac{b_k}{a_{kk}} \quad \text{при } k = n$$

$$x_k = \frac{1}{a_{kk}} \cdot \left[b_k - \sum_{l=k+1}^n (a_{kl} \cdot x_l) \right] \quad \text{при } k = n-1, n-2, \dots$$

Рассмотрим пример решения системы трех уравнений с тремя неизвестными.

$$x_1 - 2 \cdot x_2 + 3 \cdot x_3 = 1$$

$$2 \cdot x_1 + 3 \cdot x_2 + 4 \cdot x_3 = 2$$

$$5 \cdot x_1 - 4 \cdot x_2 + 3 \cdot x_3 = 3$$

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

Последовательно из 2-го и 3-го уравнений вычтем 1-е:

$$2 \cdot x_1 + 3 \cdot x_2 + 4 \cdot x_3 = 2$$

$$- 2 \cdot (x_1 - 2 \cdot x_2 + 3 \cdot x_3 = 1)$$

$$= 0 \cdot x_1 + 7 \cdot x_2 - 2 \cdot x_3 = 0$$

$$5 \cdot x_1 - 4 \cdot x_2 + 3 \cdot x_3 = 3$$

$$- 5 \cdot (x_1 - 2 \cdot x_2 + 3 \cdot x_3 = 1)$$

$$= 0 \cdot x_1 + 6 \cdot x_2 - 12 \cdot x_3 = -2$$

Далее в полученных уравнениях обнуляем коэффициенты 2-го столбца (опускаем процедуру вычитания):

$$0 \cdot x_1 + 0 \cdot x_2 - 72 \cdot x_3 = -14,$$

и находим x_3 :

$$x_3 = \frac{14}{72} = 0.194$$

Недостаток метода Гаусса состоит в его трудоемкости и неустойчивости к погрешности. Это означает, что малые погрешности вначале могут привести к большим погрешностям результата.

К численным методам следует отнести **метод прогонки** (относится к числу прямых методов решения и используется в случае

большого количества нулевых коэффициентов) и **итерационные методы**. Рассмотрим метод простых итераций.

Система $\vec{A} \cdot \vec{x} = \vec{b}$ преобразуется в эквивалентную систему вида $\vec{x} = \vec{C} \cdot \vec{x} + \vec{f}$:

$$\begin{aligned} x_1 &= \frac{-a_{12}}{a_{11}} \cdot x_2 - \frac{a_{13}}{a_{11}} \cdot x_3 - \dots - \frac{a_{1n}}{a_{11}} \cdot x_n + \frac{b_1}{a_{11}} \\ x_2 &= \frac{-a_{21}}{a_{22}} \cdot x_1 - \frac{a_{23}}{a_{22}} \cdot x_3 - \dots - \frac{a_{2n}}{a_{22}} \cdot x_n + \frac{b_2}{a_{22}} \\ &\dots \\ x_n &= \frac{-a_{n2}}{a_{nn}} \cdot x_1 - \frac{a_{n2}}{a_{nn}} \cdot x_2 - \dots + \frac{b_n}{a_{nn}} \end{aligned} \quad \vec{f} = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \dots \\ \frac{b_n}{a_{nn}} \end{pmatrix}$$

Преобразованная матрица C будет иметь нулевую главную диагональ.

$$\vec{x}^{(k+1)} = \vec{C} \cdot \vec{x}^{(k)} + \vec{f} \quad k = 0, 1, 2, \dots$$

$X^{(0)}$ – начальное приближение для вектора решения, может быть любым. Решение системы C ищут как предел последовательности векторов при $k \rightarrow \infty$ (k – номер итерации). Достаточным условием сходимости может служить выполнение одной из следующих групп неравенств:

$$\sum_{j=1}^n |\tilde{N}_{ij}| < 1 \quad i = 1, 2, \dots, n \quad \sum_{i=1}^n |\tilde{N}_{ij}| < 1 \quad j = 1, 2, \dots, n$$

Это означает, что сумма всех строк по модулю меньше единицы либо сумма всех столбцов по модулю меньше единицы.

1.4. Численное решение обыкновенных дифференциальных уравнений

Многие технические задачи формулируются в виде дифференциальных уравнений, решение которых – одна из важнейших математических задач.

По числу независимых переменных дифференциальные уравнения делятся на две различные категории: обыкновенные дифференциальные уравнения (ОДУ), содержащие одну независимую переменную, и уравнения с частными производными, содержащие несколько независимых переменных. Например, уравнение вида

$$\frac{d}{dx}y + a \cdot y = b$$

относится к ОДУ, а уравнение теплопроводности Фурье

$$\frac{d}{dt}u = a^2 \cdot \frac{d^2}{dx^2}$$

считается уравнением с частными производными. В общем виде ОДУ записывается так:

$$F(x, y, y', y'', \dots, y^n) = 0,$$

где x – независимая переменная.

Наивысший порядок n входящей в уравнение производной называется **порядком дифференциального уравнения**. Уравнение 1-го порядка имеет вид $F(x, y, y') = 0$, уравнение 2-го порядка – $F(x, y, y', y'') = 0$.

Линейным называется **дифференциальное уравнение**, линейное относительно искомой функции и её производных. Например, линейное уравнение 1-го порядка будет иметь вид

$$y' - x^3 \cdot y = \cos(x)$$

Решением дифференциального уравнения называется всякая функция $y = \varphi(x)$, которая после ее подстановки превращает уравнение в тождество. График решения $y = \varphi(x)$ – интегральная кривая.

Общее решение ОДУ n -го порядка содержит n произвольных постоянных $C_1, C_2, \dots, C_n$: $y = \varphi(x, C_1, C_2, \dots, C_n)$.

Теорема Коши. Если правая часть $f(x, y)$ уравнения $y' = f(x, y)$ и ее частная производная $f_y'(x, y)$ определены и непрерывны в некоторой области G изменения переменных x, y , то для всякой внутренней точки (x_0, y_0) этой области данное уравнение имеет единственное решение, принимающее заданное значение $y = y_0$ при $x = x_0$.

Через каждую точку в области G при $n > 1$ проходит не одна интегральная кривая. Для выделения некоторого частного решения уравнения 1-го порядка достаточно задать координаты (x_0, y_0) произвольной точки на данной кривой; для уравнений высших порядков этого недостаточно. Для выделения частного решения из общего нужно задавать столько дополнительных условий, сколько произвольных постоянных в общем решении, т. е. исходя из порядка уравнения. Для уравнения 2-го порядка следует задать два дополнительных условия, благодаря которым можно найти значения двух произвольных постоянных.

В зависимости от способа задания дополнительных условий для получения частного решения дифференциального уравнения существуют два различных типа задач: **задача Коши** и **краевая задача**. Если условия задаются в одной точке, то это задача Коши. Дополнительные условия в задаче Коши называются начальными условиями, а точка $x = x_0$ — начальной точкой. Если дополнительные условия задаются более чем в одной точке, то такая задача называется краевой. Сами дополнительные условия называются краевыми, или граничными.

Пример задачи Коши

$$y'' = y - \sin(3x)$$

$$y(0) = 0$$

$$y'(0) = 1$$

Пример краевой задачи

$$y'' = 2y - x$$

$$y(0) = 0$$

$$y'(1) = 0$$

Задача Коши для дифференциального уравнения состоит в том, чтобы найти решение уравнения $y' = f(x, y)$, удовлетворяющее начальному условию $y(x = x_0) = y_0$.

Например, частным решением задачи $y = y'$ с начальным условием $y(x_0 = 0) = 1$ является функция $f(x) = e^x$, проходящая через точку $(0, 1)$, удовлетворяющую начальным условиям (рис. 1.9).

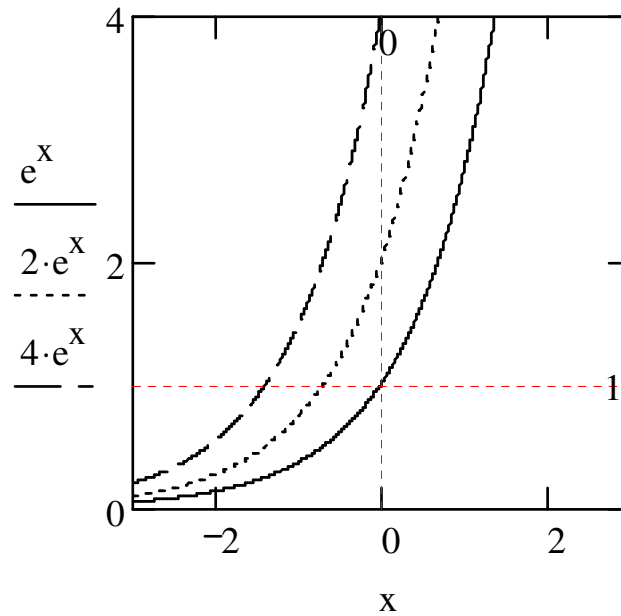


Рис. 1.9

Метод Эйлера основан на разложении искомой функции $Y(x)$ в ряд Тейлора. В начальной точке $P_0(x_0, y_0)$ (рис. 1.10) касательная $y'_0 = f(x_0, y_0)$. Согласно уравнению касательной $y - y_0 = y' \cdot (x - x_0)$ можно найти ординату y_1 , соответствующую абсциссе $x_1 = x_0 + h$ (точка $P_1(x_1, y_1)$).

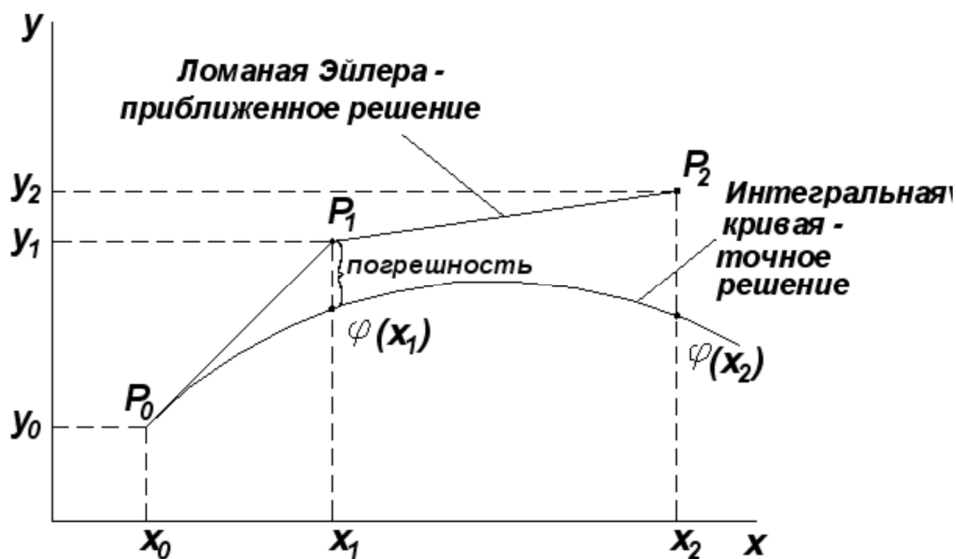


Рис. 1.10

Заменив x на x_1 и y' на $f(x_0, y_0)$, получим выражение

$$y_1 = y_0 + h \cdot f(x_0, y_0)$$

На следующем шаге ($x_2 = x_1 + h$) получим точку $P_2(x_2, y_2)$:

$$y_2 = y_1 + h \cdot f(x_1, y_1)$$

Алгоритм метода следующий.

$$x_i = x_{i-1} + h$$

$$y_i = y_{i-1} + h \cdot f(x_{i-1}, y_{i-1}) \quad i = 1, 2 \dots m$$

Графическая иллюстрация приближенного решения – ломаная, соединяющая последовательно точки $P_0, P_1, P_2, \dots, P_m$.

Погрешность на каждом шаге составит h^2 . С уменьшением шага h локальная погрешность метода Эйлера снизится, однако при этом возрастет число узлов, что неблагоприятно повлияет на точность результатов.

Метод Рунге – Кутты – одношаговый, решаемый по схемам различных порядков точности (1-й порядок точности и есть метод Эйлера). Наиболее распространена схема 4-го порядка точности:

$$y_{i+1} = y_i + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4)$$

Коэффициенты k_1, k_2, k_3, k_4 определяются следующими зависимостями:

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} \cdot k_1\right)$$

$$k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} \cdot k_2\right)$$

$$k_4 = f(x_i + h, y_i + h \cdot k_3)$$

Метод требует большого объема вычислений, поскольку на каждом шаге необходимо четырехкратное вычисление правой части уравнения $f(x, y)$, однако это окупается повышенной точностью, что дает возможность проводить вычисления с большим шагом.

1.5. Метод конечных элементов (МКЭ): основные понятия

Любую непрерывную величину (температуру, давление, перемещение) можно аппроксимировать дискретной моделью, строящейся на множестве кусочно-непрерывных функций, определенных на конечном числе подобластей (элементов).

Аппроксимация (лат. *approximo* – приближаюсь) – замена одних математических объектов, функций другими, более простыми, близкими к исходным.

Дискретный (лат. *discretus* – прерывистый) – прерывистый, состоящий из отдельных частей.

Дискретная модель строится на предположении, что числовые значения искомой величины в каждой внутренней точке области известны. При построении дискретной модели поступают следующим образом:

- в рассматриваемой области фиксируется конечное число точек (узлов);
- непрерывная величина в каждом узле считается переменной, которая должна быть определена;
- область разбивается на конечное число элементов, которые имеют общие узловые точки и в совокупности аппроксимируют форму области.

Непрерывная величина аппроксимируется в каждом элементе полиномом, который определяется с помощью узловых значений этой величины. Полиномы (функции элемента) подбираются таким образом, чтобы сохранялась непрерывность значений по границам элемента.

Типы конечных элементов следующие:

- одномерные (отрезок), имеют два узла по концам;
- двумерные (треугольники и четырехугольники), могут иметь как прямолинейные, так и криволинейные стороны;
- трехмерные (тетраэдры и параллелепипеды).

Узлы одномерного элемента (прямолинейного отрезка) длиной L (рис. 1.11) обозначаются индексами i и j . Значения искомых функций в узлах определим как Φ_i и Φ_j .

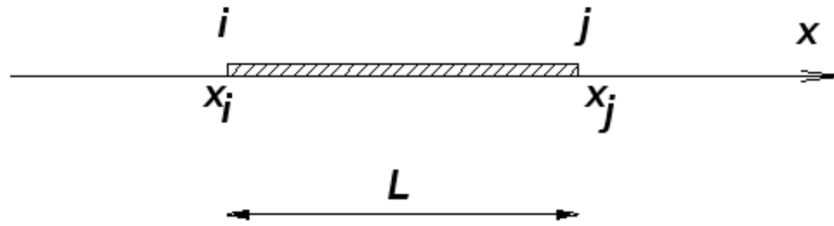


Рис. 1.11

Полиномиальная функция скалярной величины $f = \alpha_1 + \alpha_2 \cdot x$. В узловых точках $f = \Phi_i$ при $x = x_i$ и $f = \Phi_j$ при $x = x_j$. Получим выражения для значений функции в узлах конечного элемента (КЭ):

$$\Phi_i = \alpha_1 + \alpha_2 \cdot x_i; \quad \Phi_j = \alpha_1 + \alpha_2 \cdot x_j$$

Решая совместно предыдущие выражения, получим:

$$\alpha_1 = \frac{\Phi_i \cdot x_j}{L} - \frac{\Phi_j \cdot x_i}{L}; \quad \alpha_2 = \frac{\Phi_i - \Phi_j}{x_i - x_j} = \frac{\Phi_i - \Phi_j}{L}$$

Линейные функции от x называются **функциями формы** и обозначаются через N :

$$N_i = \frac{x_j - x}{L} \quad N_j = \frac{x - x_i}{L}$$

$$\phi = N_i \Phi_i + N_j \Phi_j = [N] \cdot \{\Phi\},$$

где $[N] = (N_i \ N_j)$ – матрица-строка;

$$\{\Phi\} = \begin{pmatrix} \Phi_i \\ \Phi_j \end{pmatrix} \text{ – вектор-столбец.}$$

$$\text{Функция } N_i = \frac{x_j - x}{L} = 1 \text{ в узле } i \text{ и } N_i = \frac{x_j - x}{L} = 0 \text{ в узле } j.$$

$$\text{Аналогично для } N_j = \frac{x - x_i}{L} = 1 \text{ в узле } j \text{ и } N_j = \frac{x - x_i}{L} = 0 \text{ в узле } i.$$

Эти значения характерны для функций формы (единица в одном определенном узле и нуль во всех остальных узлах).

Пример воздействия продольных сил на прямолинейный стержень представлен на рис. 1.12.

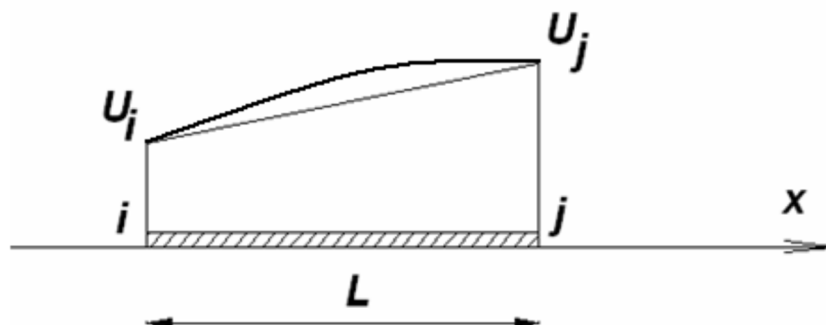


Рис. 1.12

Известна зависимость для абсолютного перемещения рассматриваемого участка (КЭ):

$$\Delta l = \int \frac{N_x}{E \cdot A} dx$$

Обозначим $\Delta l = U$.

$$N_x = \frac{d}{dx} U \cdot E \cdot A$$

Рассмотрим элемент, в котором в качестве искомой функции принимается функция перемещения U (рис. 1.13).

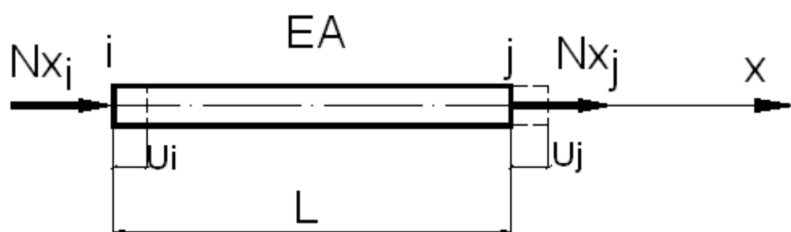


Рис. 1.13

Именно функция перемещения имеет постоянные значения в узлах (в отличие от продольных усилий, напряжений и относительной деформации). Запишем зависимость перемещения на конечном элементе через функцию формы:

$$U = U_i \cdot N_i + U_j \cdot N_j$$

Свяжем внутренние продольные усилия, возникающие в узлах i и j , с узловыми перемещениями.

Поскольку МКЭ строго основан на правилах математики, все векторные величины (усилия и перемещения) считаются положительными, если их векторы совпадают с положительным направлением оси координат. Таким образом, усилия, создающие положительные с точки зрения направления оси X перемещения, будут совпадать с положительным направлением оси X. Однако с точки зрения правил сопротивления материалов усилие N_{x_i} – отрицательная величина, поскольку вызывает в узле i сжатие. Учитывая для левого узла i противоречие в правилах знаков, получим следующие выражения для усилий:

$$N_{x_i} = -\left(\frac{dU}{dx} \cdot E \cdot A\right) = -E \cdot A \cdot \left[\frac{d}{dx}(U_i \cdot N_i) + \frac{d}{dx}(U_j \cdot N_j)\right]$$

$$N_{x_j} = \left(\frac{dU}{dx} \cdot E \cdot A\right) = E \cdot A \cdot \left[\frac{d}{dx}(U_i \cdot N_i) + \frac{d}{dx}(U_j \cdot N_j)\right]$$

Поскольку U_i и U_j – узловые величины, их можно вывести за дифференциал как постоянные значения.

$$\frac{d}{dx}N_i = \frac{d}{dx}\left(\frac{x_j - x}{L}\right) = \frac{-1}{L} \cdot \frac{d}{dx}N_j = \frac{d}{dx}\left(\frac{x - x_i}{L}\right) = \frac{1}{L}$$

Окончательно получим следующее:

$$N_{x_i} = \frac{E \cdot A}{L} \cdot [U_i \cdot 1 + U_j \cdot (-1)]$$

$$N_{x_j} = \frac{E \cdot A}{L} \cdot [U_i \cdot (-1) + U_j \cdot 1]$$

В матричной записи:

$$\begin{pmatrix} N_{x_i} \\ N_{x_j} \end{pmatrix} = \frac{E \cdot A}{L} \cdot \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} U_i \\ U_j \end{pmatrix}$$

$$\frac{E \cdot A}{L} \cdot \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = [K_\epsilon],$$

где $[K_\epsilon]$ – матрица жесткости КЭ.

Таким образом, для КЭ узловые усилия и перемещения связаны зависимостью, которую коротко можно записать так:

$$\{N_x\} = [K_e] \cdot \{U\},$$

где фигурные скобки $\{ \}$ обозначают матрицу-вектор.

Глава 2. ЗНАКОМСТВО С ПРОГРАММОЙ MATHCAD

2.1. Назначение программы

MathCAD – мощный современный математический программный комплекс, разработанный компанией PTC (США). Он позволяет выполнять различные вычисления, реализовывать алгоритмы, обрабатывать статистические данные, формировать результаты вычислений в табличной и графической формах, решать любые поставленные задачи легко, наглядно и точно. Современный продукт MathCAD (PTC MathCAD Prime) – это отраслевой стандарт программного обеспечения для инженерных расчетов, помогающий решить самые сложные проблемы. В этой главе речь пойдет о работе в современных версиях PTC MathCAD, но также будут приводиться отдельные комментарии, касающиеся работы в предыдущих версиях (на примере MathCAD 14), если описываемые приемы существенно разнятся.

2.2. Интерфейс рабочей области

MathCAD производит как символьные (точные), так и численные расчеты. Основная зона программы – рабочий лист, в который заносятся все необходимые вычисления и текстовые комментарии. Все математические операции выполняются с привлечением встроенных функций, операторов, вычислительных блоков, специальных символов, загрузка которых осуществляется из меню или специальных панелей. Ввод числовых значений, операторов и символов возможен частично напрямую с клавиатуры либо с помощью панелей операторов, символов, констант и тому подобного, расположенных на ленте *Математика* (рис. 2.1).

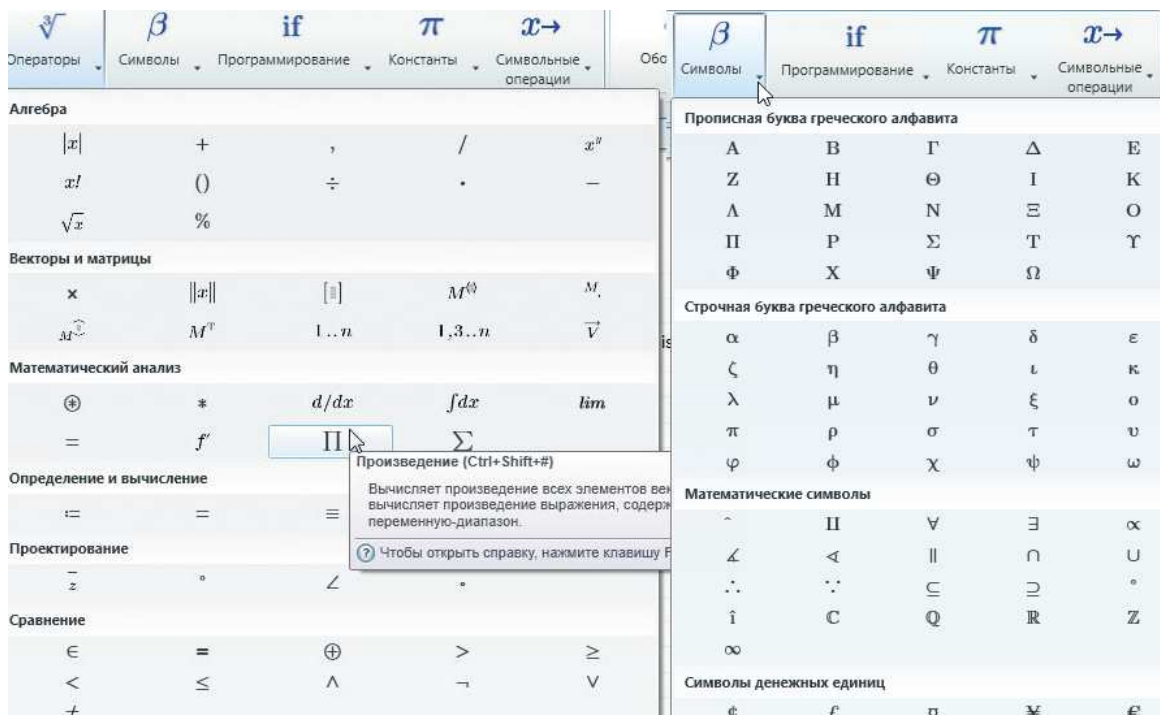


Рис. 2.1

2.3. Основные соглашения при вводе данных и выполнении математических операций



Для ввода данных и управления ими в зоне рабочего листа находится курсор. Курсор ввода имеет вид прямого креста и устанавливается мышью. При вводе данных курсор изменяет свой вид на синий вертикальный отрезок (в MathCAD 14 – на уголок) и позволяет выделять один или несколько символов либо выражение в целом. Для выделения фрагмента или всего выражения используется клавиша «Пробел», позволяющая регулировать ширину этого захвата. Перемещение по выражению осуществляется стрелками клавиатуры. На рис. 2.2, слева показано выделение одного символа и целиком выражения в PTC MathCAD Prime, на рис. 2.2 справа – в MathCAD 14.

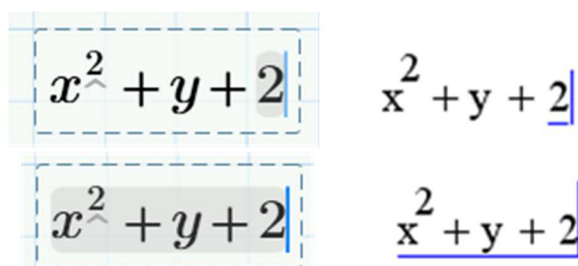


Рис. 2.2

Любое записанное выражение заключается в рамку, которая исчезает при выходе из него (щелкнуть вне записи). Для редактирования достаточно щелкнуть на выражении и внести необходимые изменения, используя текущее положение курсора.

Перемещение выделенного выражения по рабочему листу осуществляется при помощи курсора в виде стрелочек (рис. 2.3, слева), который появится, если приблизиться указателем к краю рамки. В MathCAD 14 курсор перемещения имеет вид черной руки (рис. 2.3, справа).

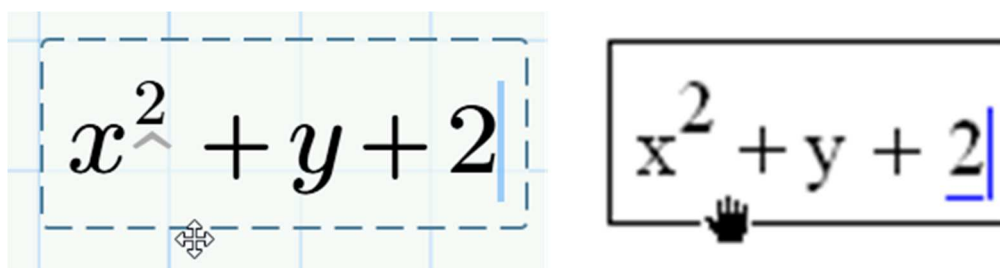


Рис. 2.3

Прежде всего следует знать, что формат ввода чисел предполагает отделение целой части от дробной при помощи точки. Ввод запятой в качестве разделителя будет считаться ошибочным. Символы, записанные в виде прописной и строчной букв (X и x), являются для MathCAD двумя разными переменными. То же относится и к одинаковым символам русского и английского алфавитов (например, русская «а» и английская «a» различаются программой).



Математические символы, которые вводятся в рабочем листе при помощи панелей, содержат пустые прямоугольники – шаблоны. Они упрощают ввод числовых значений или переменных, а также позволяют избежать лишних ошибок.

Вычисления происходят автоматически (как правило, функция автоматического вычисления включена). Кнопка автоматического расчета расположена в меню *Расчет* (в версии MathCAD 14 – сообщение «AUTO» в строке состояния).

Вводимые символы и функции для дальнейших вычислений задаются при помощи оператора присвоения вида «:=» (например, $x:=5.5$ или $f(x):=2x-1$).

Запись ведется сверху вниз и слева направо, поскольку именно в таком направлении все выражения считываются программой.

Для простых арифметических вычислений или вычислений с определенными записью выше переменными достаточно записать выражение и ввести знак «=». Например, если переменная x определена выше как $x:=2$, то выражение, включающее в себя данную переменную, можно запустить на счет:

$$x^2 - 3x = -2$$

Если в уравнение или неравенство входит переменная, не определенная записью выше, обычный знак «=» следует заменить жирным, который вводится комбинацией клавиш Ctrl и «=».

$$x^2 - 3 \cdot x = 5$$

Этот оператор также можно найти на панели операторов в разделе *Сравнение*. В MathCAD 14 оператор выбирается из панели *Boolean*.

Форматом отображения результата управляет панель *Форматирование формул* (рис. 2.4). На точность вычисления заданный формат не оказывает влияния.

В MathCAD 14 формат вывода задается функцией *Floating Point* (меню *Symbolics – Evaluate*). По умолчанию ее значение равно 20.

Диалоговое окно доступно только при выделенном выражении (результате).

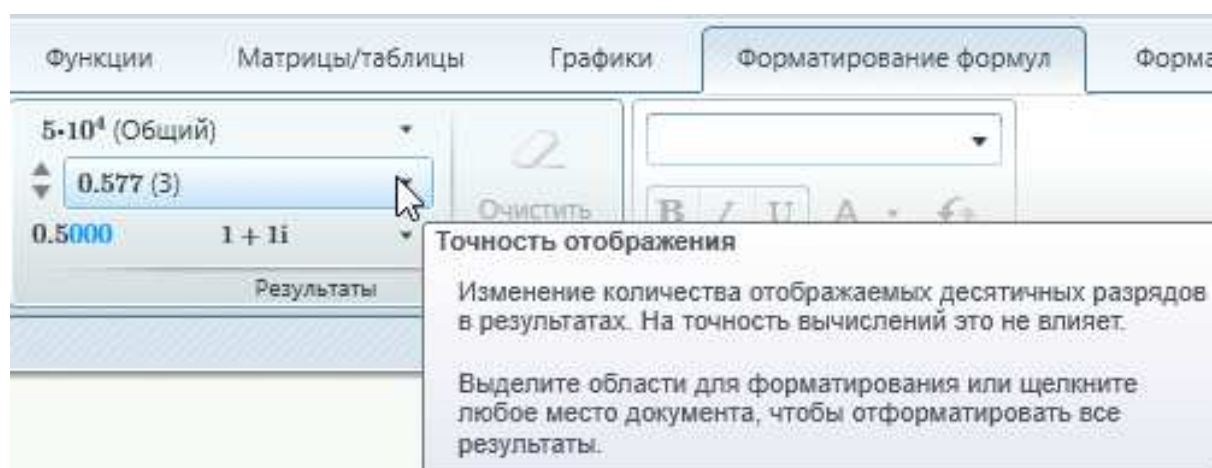


Рис. 2.4

2.4. Символьные вычисления

Для простых символьных вычислений и преобразований следует использовать функцию *solve* (решить):

$$x^2 - 2x = 8 \xrightarrow{\text{solve}} \begin{bmatrix} 4 \\ -2 \end{bmatrix}$$



Функция вызывается из раздела *Символьные операции*, расположенного на ленте *Математика*. В MathCAD 14 функция вызывается из меню *Symbolics* или выбирается на аналогичной панели. Например, для нахождения корня квадратного уравнения от переменной *x* достаточно применить функцию *solve*. Если же уравнение содержит несколько неизвестных, после ввода функции *solve* следует ввести запятую и уточнить поиск для решения уравнения.

$$x^2 - 2x + y = 8 \xrightarrow{\text{solve}, y} -x^2 + 2 \cdot x + 8$$

В том случае, если вывод решения представляет собой арифметическое выражение, можно дополнительно применить вывод в десятичном формате при помощи функции *float* (вывод в десятичном формате с плавающей точкой). Количество знаков вывода следует указать после функции *float*.

$$x^2 - 2x = 5 \xrightarrow[\text{float}, 3]{\text{solve}} \begin{bmatrix} 3.45 \\ -1.45 \end{bmatrix}$$

Обе функции расположены на панели *Символьные операции*, в группе *Ключевые слова* (рис. 2.5).

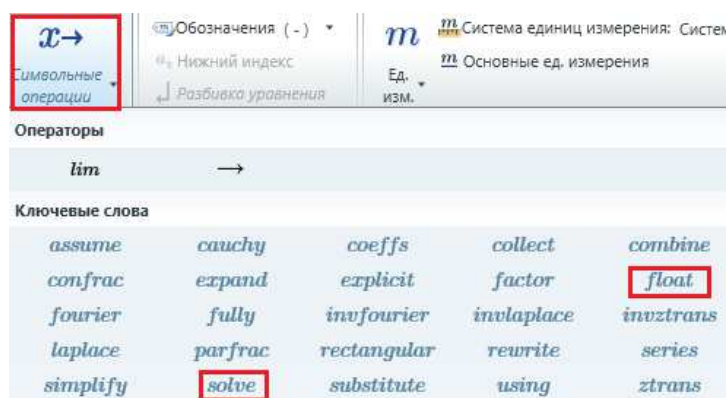


Рис. 2.5

Символьные функции предназначены как для выполнения операций с переменными, так и для записи выражения в целом. Например, необходимо разложить на множители левую часть уравнения. После выделения выражения выбираем функцию *factor* (разложить на множители).

$$x^2 - 5x + 6 \xrightarrow{\text{factor}} (x - 3) \cdot (x - 2)$$

Приведем ещё несколько примеров символьных операций. Функция *simplify* (упростить) позволяет записать выражение в более удобном виде.

$$\frac{(x^2 - 2x + 1) \cdot (x^2 - 9)}{x - 1} \xrightarrow{\text{simplify}} (x - 3) \cdot (x - 1) \cdot (x + 3)$$

Функция *coeffs* выдает вектор коэффициентов полинома в строгом порядке сверху вниз, начиная с коэффициента при x^0 . Отсутствующие коэффициенты записываются равными нулю. Для получения вектора надо выделить не всё выражение, а только переменную x , коэффициенты при которой записываются в вектор.

$$5x^4 - 3x^3 + 2.5x - x^2 \xrightarrow{\text{coeffs}} \begin{bmatrix} 0.0 \\ 2.5 \\ -1.0 \\ -3.0 \\ 5.0 \end{bmatrix}$$

Функция *substitute* (подстановка) часто используется при замене конкретной переменной заданным значением.

$$5nx^4 - 3nx^3 + 2.5nx \xrightarrow{\text{substitute}, n=3.2} 16.0 \cdot x \cdot (x^3 - 0.6 \cdot x^2 + 0.5)$$



Для вычисления интегралов, пределов, дифференциалов и тому подобного применяются операторы панели *Операторы* (в версии MathCAD 14 – панель *Calculus*). Сначала вводится необходимый шаблон и заполняется. Для получения результата вычисления символьных выражений используется стрелка $\rightarrow$, расположенная в группе *Определение и вычисление*. Иначе стрелку можно вставить комбинацией клавиш Ctrl и «.» (точка). В MathCAD 14 стрелка расположена на панели *Evaluation*.

$$\frac{d}{dx} (5x^4 - 3x^3 + 2.5x - x^2) \rightarrow 20.0 \cdot x^3 - 9.0 \cdot x^2 - 2.0 \cdot x + 2.5$$

Часто в функциях применяются переменные, которые должны принимать ряд фиксированных значений. Такие переменные называются ранжированными. Формат задания значений ранжированным переменным следующий:

$$a[a + h]..b,$$

где a , b – область значений переменной; h – шаг. Если задается ранжированная переменная вида $x..10$, то это означает, что x принимает значения от 1 до 10 с шагом 1. Если шаг не равен единице, то после начального значения переменной через запятую пишется второе значение ряда, разница между первым и вторым значениями и будет шагом ранжированной переменной. Например, интервал для x от 1 до 3 с шагом 0.25 надо записать в таком порядке:

$$x := 1, 1.25..3 \quad \begin{bmatrix} 1 \\ 1.25 \\ 1.5 \\ 1.75 \\ 2 \\ 2.25 \\ 2.5 \\ 2.75 \\ 3 \end{bmatrix}$$

Не следует забывать, что при убывающем ранге ($b < a$) шаг должен быть отрицательным, т. е. второе записываемое значение будет меньше первого. Формат записи появится в виде шаблона автоматически, как только после первого присвоенного значения ранжированной переменной будет введена запятая.

$$x := 1, ..$$

Для шага, равного единице, можно воспользоваться шаблоном $1..n$ (расположен в группе *Векторы и матрицы*). Полный шаблон ранжированной переменной имеет вид $1,3..n$ (расположен там же).

2.5. Построение двухмерных графиков

При построении графиков используют шаблон-график (рис. 2.6) из ленты *Графики* – *Вставить график* (в версии MathCAD 14 для вставки графика применяется панель *Graph* или комбинация клавиш Shift и «2»).

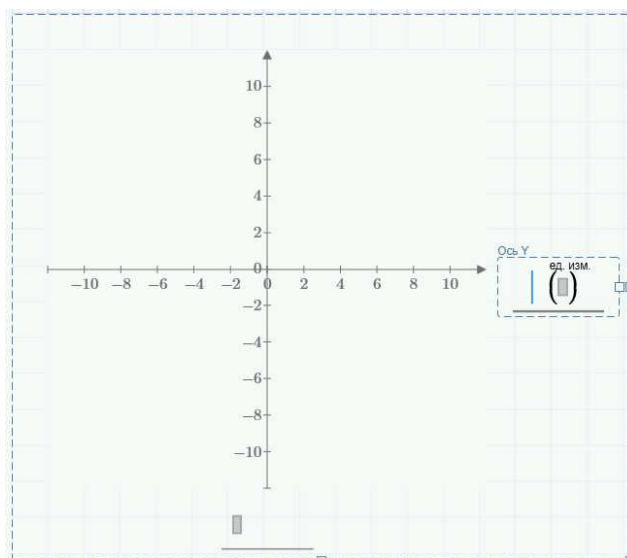


Рис. 2.6

В правой стороне графика вместо шаблона вводится имя функции, определенной записью выше (рис. 2.7, слева), либо её выражение (рис. 2.7, справа). Внизу графика вводится аргумент. При необходимости выражению и аргументу задаются единицы измерения.

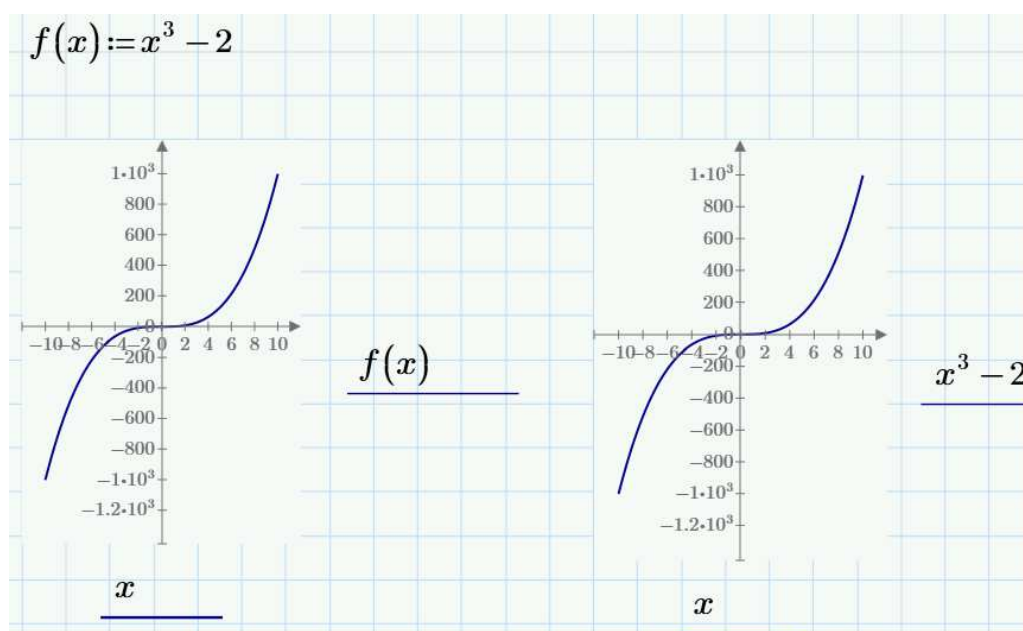


Рис. 2.7

После ввода аргумента и функции строится график, а оси абсцисс и ординат разбиваются автоматически. Часто построенный график требует редактирования: укрупнения фрагмента, изменения пределов на осях. Для изменения предела выполняется щелчок в области его значения, и последнее редактируется с помощью клавиатуры (рис. 2.8). Аналогичным образом укрупняется нужный фрагмент (рис. 2.9).

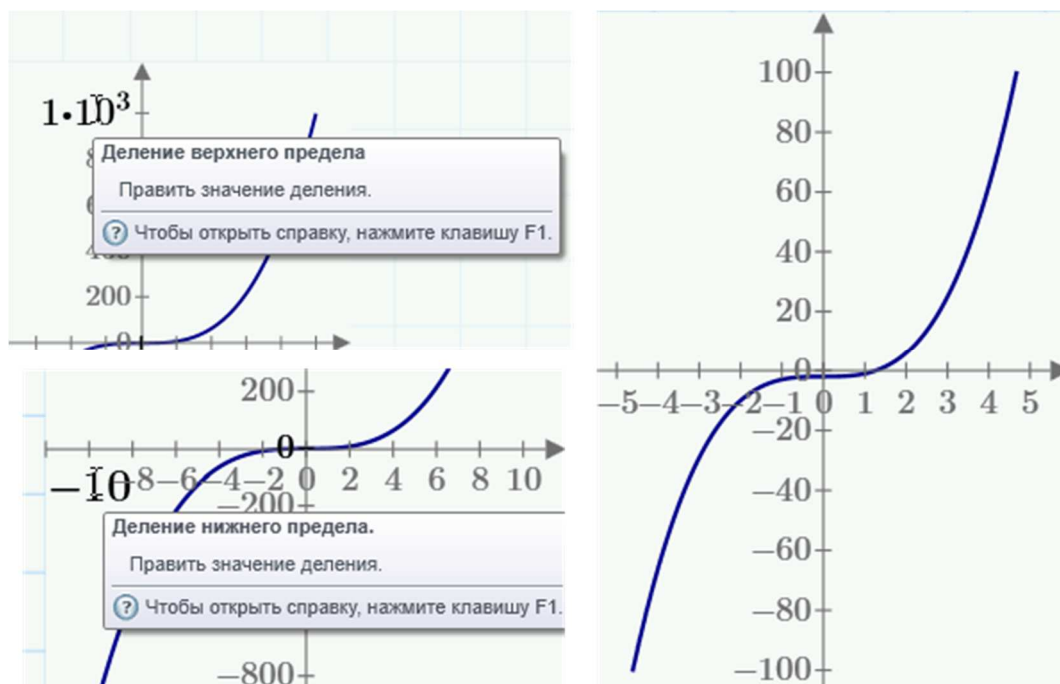


Рис. 2.8

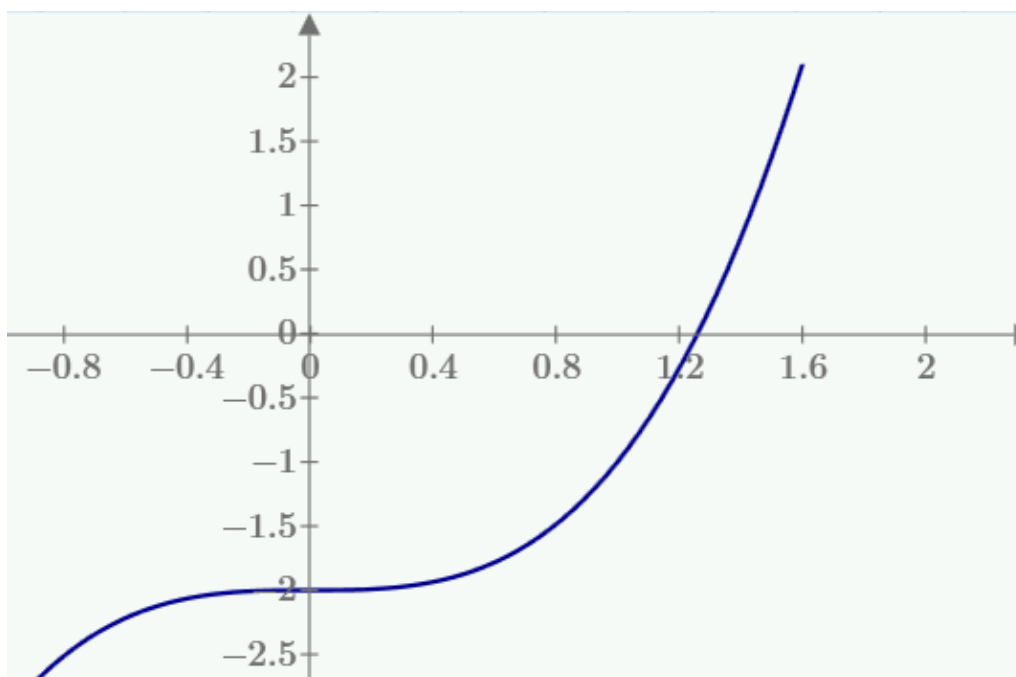


Рис. 2.9

В версии MathCAD 14 построенный график дополнительно редактируется при помощи диалогового окна *Format*, которое можно вызвать из контекстного меню, щелкнув правой кнопкой мыши в области графика, или двойным щелчком левой кнопкой. Диалоговое окно имеет несколько вкладок, в том числе управляющих настройкой системы координат (*X-Y Axes*), отображением графиков (*Traces*), подписями (*Labels*) и др. Увеличением изображения управляет диалоговое окно *Zoom* (вызывается из контекстного меню графика или панели *Graph*). Чтобы укрупнить фрагмент графика, надо выделить мышью область укрупнения и щелкнуть в диалоговом окне *Zoom* по кнопке в виде лупы с плюсом.

Значения числовой оси графика задаются с определенной точностью в меню *Форматирование значений графика*. Кроме назначенной точности отображения можно изменить формат единиц.

Часто требуется в области одного графика создать несколько кривых. Чтобы добавить к существующей кривой новую зависимость, применяется команда *Добавить кривую*, расположенная на ленте *Графики*. Для добавления новой зависимости должна быть выделена функция (рис. 2.10), для добавления нового аргумента – выделен аргумент. В результате выполнения команды появится новый шаблон, в который следует вставить имя новой функции (выражения) или имя нового аргумента.

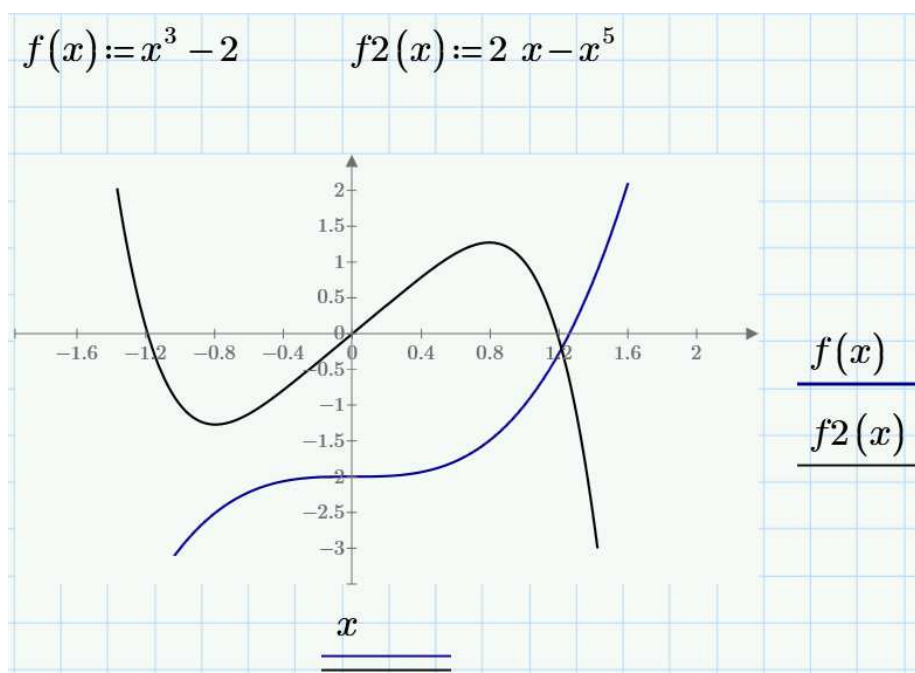


Рис. 2.10

В MathCAD 14 для добавления новой кривой в области графика выделяется имя функции, вводится запятая, в появившийся шаблон записывается новая функция или выражение. Аналогичным образом (через запятую) добавляется новый аргумент.

Если независимой переменной, являющейся аргументом функции, присвоено конкретное значение, график не будет построен. В этом случае следует отменить присвоение, либо построить график выше присвоения, либо задать другой аргумент.

Часто для нескольких графиков функций в одной системе координат требуется аргумент, принимающий значение в разных областях (интервалах) числовой оси, например при построении эпюр по участкам одного стержня с разными функциональными зависимостями. Для этого используют ранжированные переменные, определенные заранее.

Допустим, первая функция имеет зависимость $f1(x) = 2 \cdot x + 6$ ($0 < x < 2$), а вторая функция имеет зависимость $f2(x) = 10 - (x - 2)$ ($2 < x < 5$).

Чтобы получить обе зависимости в одной системе координат, заменим аргументы на $x1$ и $x2$ и зададим их как ранжированные переменные с шагом 0.1. График двух функций показан на рис. 2.11.

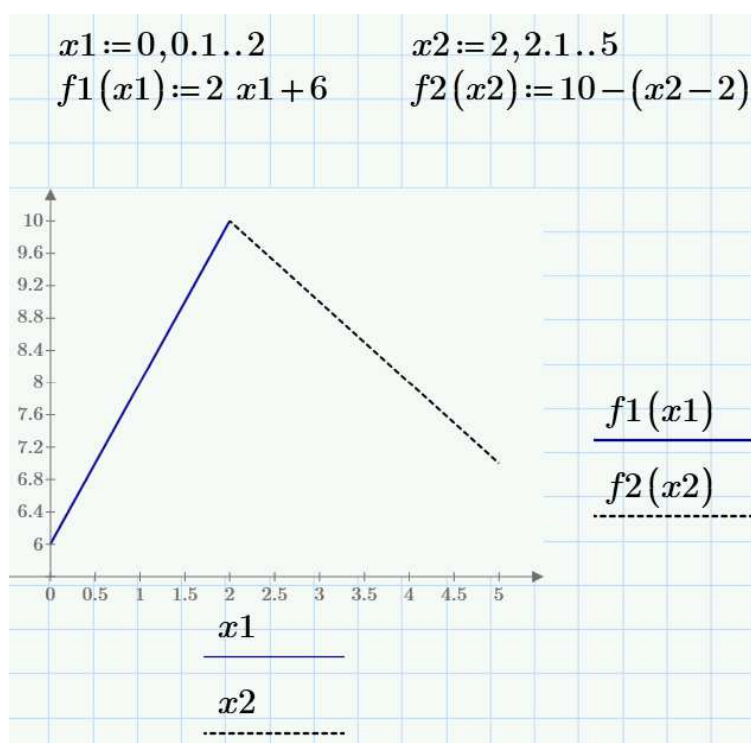


Рис. 2.11

График функции может быть представлен разными типами, которые задаются в меню *Изменить тип* ленты *Графики* (рис. 2.12). Некоторые представления графиков требуют определенного задания функции и аргумента (например, в виде массивов).

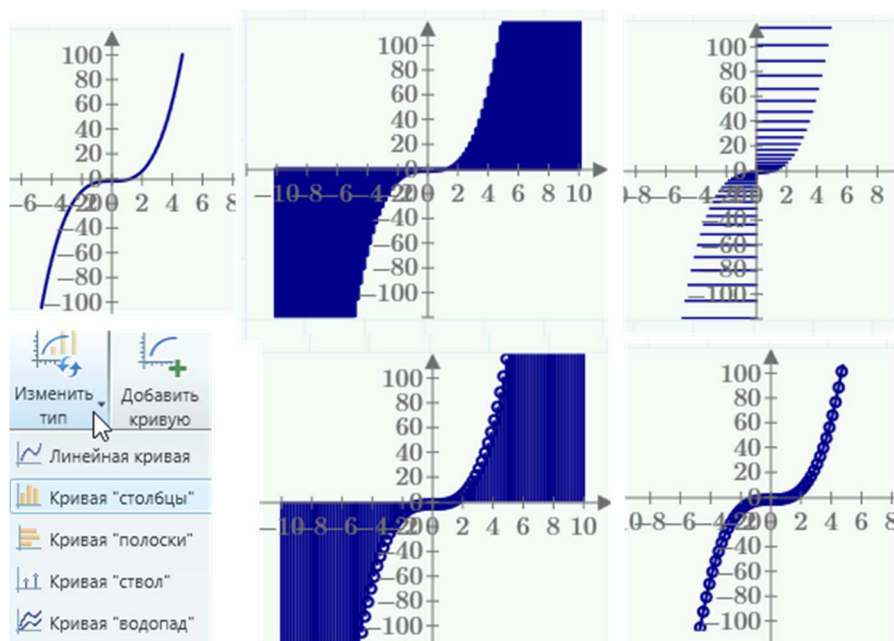


Рис. 2.12

Для отметки на графике конкретных значений функции или аргумента (рис. 2.13) применяются команды *Добавить вертикальный маркер*, *Добавить горизонтальный маркер*, расположенные на ленте *Графики*. Около добавленного маркера появится шаблон, в который вводят необходимое значение.

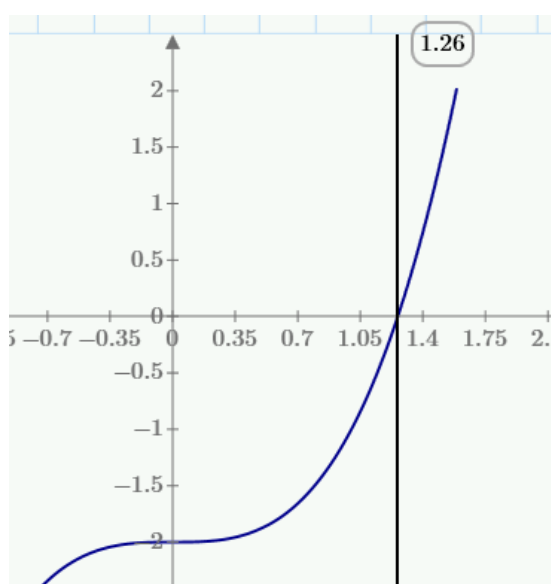


Рис. 2.13

Кривые графика могут быть представлены линиями разной толщины, разного типа и цвета, с символами, количество которых назначается числом точек. На рис. 2.14 кривой назначен пунктирный тип линии красного цвета с отметкой символами в количестве 100 точек.

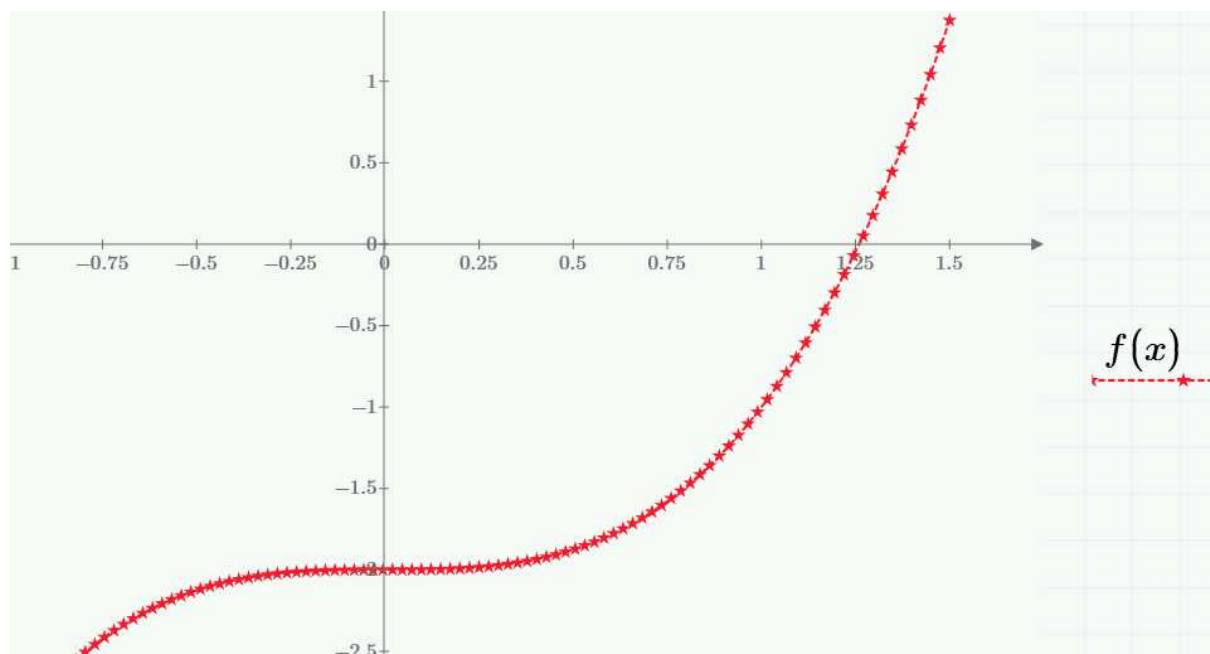


Рис. 2.14

2.6. Блок решения

Блок решения (вычислительный блок) позволяет решать различные уравнения и системы уравнений в отдельной области, представлять решение в удобной и наглядной форме. Вставка блока осуществляется соответствующей кнопкой ленты *Математика*. В теле блока задаются прежде всего начальные условия (или начальные приближения), затем записывается уравнение (или система уравнений) и решение при помощи одного из ключевых слов: *Find*, *Minerr*, *Minimize*, *Maximize*, *Odesolve*, которые можно набрать вручную или вызвать из групп функций *Решение*, *Дифференциальные уравнения* ленты *Функции*. Условно блок делится на три раздела (они выделены слева заголовками): начальные приближения, ограничения, решатель. Если вычисляется несколько переменных, они выводятся в матричном (векторном) виде. Пример решения квадратного уравнения показан на рис. 2.15. Здесь приведено довольно простое решение, которому ограничения не требуются. Для каждого искомого корня уравнения задано своё начальное приближение.

Решение уравнения	$x := 0$	Решение уравнения	$x := -2$
	$3(x^2 - 5) + 2x = 0$		$3(x^2 - 5) + 2x = 0$
	$\text{Find}(x) = 1.927$		$\text{Find}(x) = -2.594$

Рис. 2.15

В MathCAD 14 решение в вычислительном блоке предваряется ключевым словом *Given*. Блок охватывает всё горизонтальное пространство рабочего листа, поэтому любое написание параллельно блоку посторонних выражений будет считаться встроенным в блок и повлечет за собой ошибки.

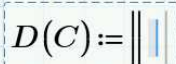
2.7. Элементы программирования

Наряду с использованием встроенных функций, MathCAD позволяет также создавать собственные функции, в которых применяются операторы программирования, расположенные на панели *Программирование* ленты *Математика* (рис. 2.16).



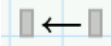
Рис. 2.16

Обычно запись программы начинается с ввода её имени и перечня в скобках внутренних переменных (иногда их может и не быть). Программа определяется знаком присвоения так же, как и любая другая функция. Программный блок отделяется от имени программы вертикальной чертой, которая создается кнопкой в виде двойной вертикальной черты, называемой *Программа* (вставка новой программной структуры). Ниже приведен пример создания программы вычисления количества участников конференции по известному количеству рукопожатий (задача была рассмотрена в гл. 1).



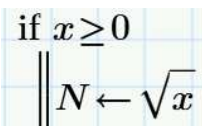
Здесь D – имя программы, C – внутренняя переменная. Правее знака присвоения вставлена вертикальная черта для создания программной структуры. В зависимости от сложности программы (количества циклов, условий и т. п.) вертикальных разделителей может быть несколько.

Операторы программирования имеют свою конструкцию, которая появляется в шаблонном виде при их вводе (не рекомендуется вводить операторы самостоятельно с клавиатуры во избежание ошибок).

 – знак присвоения для переменной или функции. Аналогичен записи $x :=$.

 – блок ввода условия (если...).

Например, вычисление квадратного корня для положительного значения аргумента в блоке условия *if* запишется так (N будет равно квадратному корню из x при условии, что x не меньше нуля):



else – означает *иначе* и работает совместно с оператором *if* в качестве альтернативы. Предыдущий пример будет более корректен, если учесть любое значение аргумента. Полностью функция вычисления квадратного корня будет записана так:


```

F(x) := || if x ≥ 0
        || || N ← √x
        || else
        || || return “нет решения”

F(9) = 3

F(-9) = “нет решения”

```

Оператор *return* осуществляет возврат программы и применяется для вывода результата или сообщений по различным условиям.

else if, also if – дополнения (уточнения) к блоку *if*.

```

while ||
||

```

– цикл по условию (предпроверке). Итерации цикла будут выполняться, пока заданное в строке *while* условие соблюдается. В качестве примера приведем полностью программу, составленную для задачи вычисления участников конференции.

```

D(C) := || N ← 1
        || R ← 0
        || while R < C
        || || N ← N + 1
        || || R ← R + N - 1
        || N

D(105) = 15   D(120) = 16

```

В теле цикла *while* выполняется счет людей (N) и рукопожатий (R) до тех пор, пока количество рукопожатий меньше заданного значения C. Как только R станет больше заданного C, цикл *while* остановит счет и выдаст программе величину N из последней итерации.

```

for || ∈ ||
||

```

– цикл с параметром. В строке *for* (для) записываются границы значений параметра (как правило, в виде ранжированной переменной), ниже шаблон предназначен для ввода тела цикла. Пример применения цикла *for* – вычисление суммы целых чисел от 1 до n.

Ниже для проверки результата приведена аналитическая формула расчета суммы целых чисел от 1 до n.

$$SUM(n) := \begin{cases} k \leftarrow 0 \\ \text{for } i \in 1..n \\ \quad k \leftarrow k + i \end{cases}$$

$$SUM(10) = 55$$

$$\frac{10 \cdot (10 + 1)}{2} = 55$$

break – досрочный выход из программы или цикла.

continue – продолжение цикла.

try (on error) – инструкция по обработке ошибок.

Глава 3. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ, РЕАЛИЗОВАННЫЕ В ПРОГРАММЕ MATHCAD

3.1. Численное решение нелинейных уравнений в MathCAD

Многие уравнения не решаются аналитически, поэтому для их решения применяются численные методы. Это означает, что ищется приближенное решение $f(x) = 0$ с заданной погрешностью. Наиболее распространены итерационные методы (т. е. методы последовательных приближений):

- простейший итерационный метод по начальному приближению корня x_0 ;
- метод половинного деления;
- метод секущих, или метод линейной интерполяции;
- метод касательных, или метод Ньютона.

Рассмотрим подробнее метод Ньютона. Пусть на отрезке $[a, b]$ содержится единственный корень $x = \alpha$ уравнения $f(x) = 0$, а функция имеет непрерывные производные $f'(x)$ и $f''(x)$, которые не обращаются в нуль на данном отрезке. Проведем касательную к функции в некоторой точке x_1, y_1 внутри интервала $[a, b]$. Уравнение касательной:

$$y - y_1 = f'(x_1) \cdot (x - x_1)$$

Поскольку $y_1 = f(x_1)$, получим для точки пересечения касательной с осью x ($y = 0$):

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

Это означает, что существует такая окрестность точки $x = \alpha$, что последовательность

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \quad n = 1, 2, \dots$$

будет сходиться к $x = \alpha$.

Программа MathCAD предусматривает несколько способов и методов решения нелинейных уравнений. Поскольку все численные функции выдают результат с некоторой погрешностью, необходимо заранее задать ее величину. Погрешность регулируется системной переменной TOL, которая по умолчанию равна 0.001. Ее значение можно переопределить в рабочем листе либо задать в меню *Расчет* (в MathCAD 14 назначается в диалоговом окне математических опций, меню *Tools*, пункт *Worksheet Options*).

Функция *root* ($f(x), x, [a, b]$), где $f(x)$ – заданная функция; x – аргумент; $[a, b]$ – интервал, в котором ищется корень функции $f(x)$, предполагает предварительное построение графика и определение интервала, внутри которого имеется одно пересечение с осью абсцисс.

Пример решения уравнения функцией *root*:

$$\text{root}(e^x - x^3, x, 1, 2) = 1.857$$

С заданной записью выше функцией:

$$f(x) := e^x - x^3$$

$$\text{root}(f(x), x, 1, 2) = 1.857$$

Функция *polyroots*(v) (находится в группе функций *Решение ленты Функции*) используется для нахождения корней полинома. Решается относительно вектора v коэффициентов полинома (первый элемент вектора – коэффициент при x^0 , далее – по возрастанию порядка степени аргумента).

$$\begin{array}{l}
 x^5 + 3(x^3 - 2x^2) + x = 0 \\
 x^5 + 3(x^3 - 2x^2) + x \xrightarrow{\text{coeffs}} \begin{bmatrix} 0 \\ 1 \\ -6 \\ 3 \\ 0 \\ 1 \end{bmatrix} \quad v := \begin{bmatrix} 0 \\ 1 \\ -6 \\ 3 \\ 0 \\ 1 \end{bmatrix} \\
 \text{polyroots}(v) = \begin{bmatrix} -0.68 + 2.042i \\ -0.68 - 2.042i \\ 0 \\ 0.184 \\ 1.175 \end{bmatrix}
 \end{array}$$

Блок решения со следующей структурой:

- начальные условия (задается начальное значение искомой переменной – некоторая опорная точка);
- запись уравнения в виде приближенного равенства (используется знак « $\approx$ » жирного начертания);
- выражения с ключевым словом *Find* или *Minerr*.

Пример блока:

$$\begin{array}{l}
 x := 1 \\
 e^x - x^3 \approx 0 \\
 \text{Find}(x) = 1.857
 \end{array}$$

Итерационный метод можно записать в виде программного блока. Функция вычисления корня нелинейного уравнения по методу Ньютона может быть записана следующим образом.

$$\text{Newton}(f, x) := \left| \begin{array}{l} x1 \leftarrow x + 2 \cdot \text{TOL} \\ \text{while } |x1 - x| > \text{TOL} \\ \quad \left| \begin{array}{l} x \leftarrow x1 \\ x1 \leftarrow x - \frac{f(x)}{\frac{d}{dx}f(x)} \end{array} \right. \end{array} \right.$$

Для нахождения корня необходимо построить график функции, определить интервал поиска, задать выражение $f(x)$, после чего ввести

стартовое значение переменной в качестве параметра x функции *Newton*.

3.2. Операции с матрицами

Матрица – упорядоченный массив, каждый элемент этого массива имеет свой индекс. Нумерацией индексов управляет функция

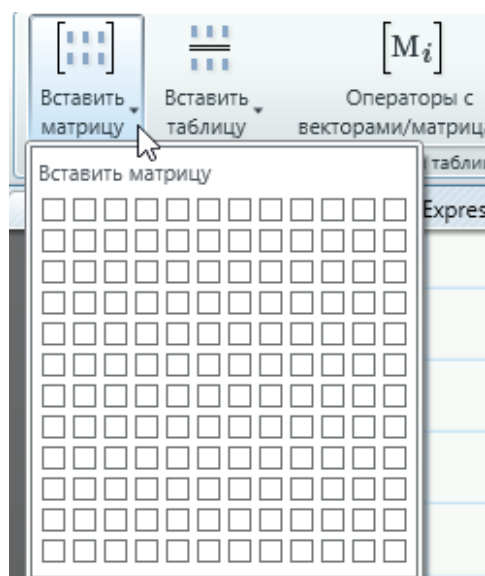


Рис. 3.1

ORIGIN (меню *Расчет*), которая имеет значение, соответствующее первому порядковому номеру элемента матрицы. По умолчанию это значение равно нулю, т. е. первый элемент матрицы нулевой.

Среди матриц выделяют матрицу-вектор (один столбец и m строк), матрицу-скаляр (однострочная матрица) и двумерную матрицу-массив, состоящую из m строк и n столбцов. Один из способов создания матрицы – вставка шаблона (расположен в ленте *Матрицы/таблицы*) (рис. 3.1).

Другой способ задания матрицы – поэлементный. Например, для создания квадратной матрицы 2-го порядка следует задать четыре элемента.

$$K_{0,0} := 1 \quad K_{0,1} := 2 \quad K_{1,0} := 3 \quad K_{1,1} := 1$$

$$K = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$

Элемент матрицы записывается с нижними индексами, которые указывают на номер строки (первый индекс) и номер столбца (второй индекс). Элемент матрицы-вектора записывается с одним индексом. Ввод индекса осуществляется при помощи шаблона (панель *Операторы с векторами/матрицами* ленты *Матрицы/таблицы*) либо клавишей «[».

На панели *Операторы с векторами/матрицами* также имеются шаблоны для ввода вектора, символа векторного произведения, об-

ратной матрицы, вывода столбца или строки матрицы, вычисления транспонированной матрицы и определителя.

Некоторые функции характеристик матриц:

$cols(A)$ – возвращает число столбцов матрицы A ;

$rows(A)$ – возвращает число строк матрицы A ;

$mean(A)$ – вычисляет среднее значение элементов матрицы A .

Функции формирования матриц следующие:

$identity(n)$ – формирует единичную матрицу с размерами $n \times n$:

$$identity(3) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$stack(A,B,C,...)$ – объединяет матрицы с равным количеством столбцов сверху вниз:

$$A1 := \begin{pmatrix} 1 & 4 & 8 & 0 \\ 3 & 6 & 5 & 9 \end{pmatrix} \quad A2 := (-2 \ 0 \ 11 \ 4) \quad stack(A1, A2) = \begin{pmatrix} 1 & 4 & 8 & 0 \\ 3 & 6 & 5 & 9 \\ -2 & 0 & 11 & 4 \end{pmatrix}$$

$augment(A,B,C,...)$ – объединяет матрицы с равным количеством строк бок о бок (слева направо):

$$B1 := \begin{pmatrix} 1 & 0 \\ 2 & 4 \end{pmatrix} \quad B2 := \begin{pmatrix} 3 & 7 & 8 \\ 0 & 5 & 6 \end{pmatrix} \quad augment(B1, B2) = \begin{pmatrix} 1 & 0 & 3 & 7 & 8 \\ 2 & 4 & 0 & 5 & 6 \end{pmatrix}$$

Выделение части матрицы с указанием количества строк и столбцов осуществляется функцией $submatrix(A,ir,jr,ic,jc)$, где A – основная матрица; ir, jr (записывается числом) – порядковые номера строк выделенной матрицы (естественно, $jr > ir$); ic, jc – порядковые номера столбцов.

Функции, реализующие численные алгоритмы решения задач линейной алгебры, следующие:

$rref(A)$ – выполняет элементарные операции со строками (приводит матрицу к ступенчатому виду с единичным базисным минором);

$eigenvals(A)$, $eigenvecs(A)$ – вычисление соответственно собственных значений и собственных векторов квадратной матрицы A ;

$lsolve(A,B)$ – решение системы линейных алгебраических уравнений $A \cdot x = B$.

3.3. Решение систем линейных уравнений

Подробнее рассмотрим матричный метод решения систем алгебраических уравнений на примере четырех уравнений с четырьмя неизвестными:

$$a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 + a_{14} \cdot x_4 = b_1$$

$$a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 + a_{24} \cdot x_4 = b_2$$

$$a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 + a_{34} \cdot x_4 = b_3$$

$$a_{41} \cdot x_1 + a_{42} \cdot x_2 + a_{43} \cdot x_3 + a_{44} \cdot x_4 = b_4$$

Система в свернутом виде может быть записана так:

$$\sum_{i=1}^4 a_{i,j} \cdot x_i = b_j \quad j=1,2,3,4$$

В соответствии с правилом умножения матриц (если $C = A \cdot B$, то $c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{in} \cdot b_{nj}$) рассмотренная система уравнений может быть записана в матричном виде: $A \cdot x = B$, где A – матрица системы, столбцами которой являются коэффициенты при соответствующих неизвестных x_i , а строками – коэффициенты при неизвестных в соответствующем уравнении; B – матрица-столбец, элементами которой являются правые части уравнений (матрица правой части); x – матрица-столбец, элементами которой являются искомые неизвестные (решение системы).

Если матрица системы невырожденная (определитель матрицы отличен от нуля), то система уравнений имеет единственное решение.

К точным решениям следует отнести метод Крамера:

$$x_j = \frac{\det A_j}{\det A} \quad j=1,2..n,$$

где $\det A_j$ – определитель матрицы, получаемый заменой j -го столбца матрицы A столбцом правых частей.

В программе MathCAD решение систем линейных уравнений матричным способом реализовано следующим образом.

Функция $lsolve(A,B)$ возвращает решение системы в векторном виде. Для применения функции должна быть сформирована матрица системы A и вектор правой части системы B . В явном виде решение также можно записать так:

$$x := A^{-1} \cdot B$$

Решение системы уравнений методом Гаусса осуществляется при помощи функции $rref(C)$. Для того чтобы функция нашла решение, необходимо объединить матрицу системы A и матрицу правой части системы B функцией $augment$. Запись решения следующая:

$C := augment(A,B)$

$x := rref(C)^{<n>}$

Последний столбец матрицы C (его порядковый номер n) и есть решение системы.

3.4. Решение обыкновенных дифференциальных уравнений

К обыкновенным дифференциальным уравнениям относят уравнения, содержащие одну независимую переменную. В общем виде уравнения могут быть записаны как $F(x, y, y', y'', \dots, y^n) = 0$, где x – независимая переменная. Наивысший порядок n входящей в уравнение производной называется порядком дифференциального уравнения.

Решение ОДУ – всякая функция $y = \varphi(x)$, которая после ее подстановки в уравнение превращает его в тождество.

Решение в общем виде:

$$y = \varphi(x, C_1, C_2, \dots, C_n)$$

Частное решение получается из общего, если произвольным постоянным $C_1, C_2, \dots, C_n$ придать определенные значения. Для выделения частного решения из общего следует задавать столько дополнительных условий, сколько произвольных постоянных в общем решении, т. е. исходя из порядка уравнения. Если эти условия задаются в одной точке, то такая задача называется задачей Коши. Дополнительные условия в задаче Коши называются начальными условиями, а точка $x = x_0$, в которой они задаются, – начальной точкой.

Один из наиболее точных методов численного решения ОДУ – метод Рунге – Кутты, одношаговый, решаемый по схемам различных

порядков точности. Порядок точности определяется по шагу h на сетке. Схема Рунге – Кутта 4-го порядка точности:

$$y_{i+1} = y_i + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4),$$

где k – коэффициенты (подробно формулы их вычисления приведены в п. 1.4).

Программа вычисления пошаговых значений y , реализованная в MathCAD, для ОДУ 1-го порядка представлена ниже.

```
rk4(y0, a, b, n, f) :=
| y1 ← y0
| x1 ← a
| h ← (b - a) / n
| for i ∈ 2 .. n + 1
|   | k1 ← f(x_{i-1}, y_{i-1})
|   | k2 ← f(x_{i-1} + h/2, y_{i-1} + h/2 · k1)
|   | k3 ← f(x_{i-1} + h/2, y_{i-1} + h/2 · k2)
|   | k4 ← f(x_{i-1} + h, y_{i-1} + h · k3)
|   | y_i ← y_{i-1} + h/6 · (k1 + 2 · k2 + 2 · k3 + k4)
|   | x_i ← x_{i-1} + h
| Y ← augment(x, y)
```

Здесь y_0 – начальное значение функции при $x_0 = a$; $[a, b]$ – интервал интегрирования; n – заданное количество шагов на интервале; $f(x, y)$ – уравнение первой производной.

Обыкновенные дифференциальные уравнения высших порядков для численного решения функциями MathCAD должны быть представлены задачей Коши в векторной форме. Само решение функциями выдается в табличном и графическом видах.

Рассмотрим решение на примере дифференциального уравнения 2-го порядка при помощи функции *rkfixed* (решение Рунге – Кутта 4-го порядка точности с фиксированным шагом).

Математическая постановка задачи сводится к представлению вектора начальных условий и вектора системы ОДУ. Предварительно уравнение должно быть преобразовано в систему уравнений 1-го порядка:

$$y_1 = y(x)$$

$$y_2 = y'(x)$$

$$y_2' = f(x, y_1, y_2) = y(x)''$$

Определен вектор начальных условий y :

$$y = \begin{bmatrix} y(x_0) \\ y'(x_0) \end{bmatrix}$$

и векторная функция правых частей уравнения $D(x, y)$:

$$D(x, y) = \begin{bmatrix} y_1 \\ y'(x, y_1, y_2) \end{bmatrix}$$

Формат функции *rkfixed*($y, x_0, x_k, Npoints, D$) включает в себя определенный выше вектор начальных условий y , начальное x_0 и конечное x_k значения интервала интегрирования, количество расчетных точек $Npoints$ и векторную функцию правых частей $D(x, y)$. Численное решение уравнения получаем в виде матрицы, которая содержит в первом столбце координаты N узлов равномерной сетки на отрезке $[x_0, x_k]$, во втором столбце – приближенные значения решения в этих узлах функции, в третьем столбце – значения первой производной.

Функция *rkadapt* решает уравнение методом Рунге – Кутты с автоматическим подбором шага.

Численные методы Булирша – Штёра и алгоритм Розенброка для жестких систем представлены рядом функций (*bulstoer*, *stiff*, *stiffb*), также использующих векторное представление всех параметров.

Исключением векторного представления уравнения в MathCAD является функция *odesolve*, позволяющая записать уравнение в символьном виде. Функция вместе с дифференциальным уравнением по-

мещается внутри вычислительного блока, предварительно вводятся начальные условия.

Формат функции $odesolve(x, b[step])$ требует указать конечное значение интервала интегрирования b . Шаг ($step$) необязателен, но при его указании функция решает уравнение методом Рунге – Кутты с фиксированным шагом.

3.5. Реализация решения МКЭ в системе MathCAD

Рассмотрим метод конечных элементов на примере расчета стержня на продольную нагрузку.

В качестве неизвестной (искомой) функции принимается перемещение стержня. Стержень разбивается на n конечных элементов (рис. 3.2). Каждый элемент должен иметь постоянные физические и геометрические характеристики, а приложенная нагрузка на элементе – одну функциональную зависимость (для продольной нагрузки это означает, что либо внутри стержня может быть распределенная нагрузка одной интенсивности, либо отсутствует всякая нагрузка). Все узлы и КЭ нумеруются слева направо.

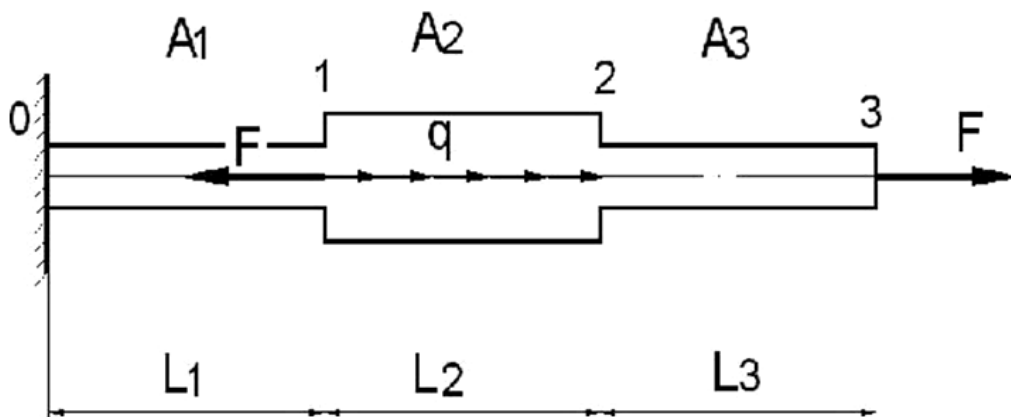


Рис. 3.2

На рис. 3.3 показано, как разбита схема (см. рис. 3.2) на три конечных элемента (при реализации задачи в программе MathCAD первый порядковый номер узла зависит от назначенной системной переменной $ORIGIN$, управляющей нумерацией ячеек в матрице). Узлы 1 и 2 общие для 1-го и 2-го КЭ, а следовательно, приведенные к узлам внешние силы также суммируются.

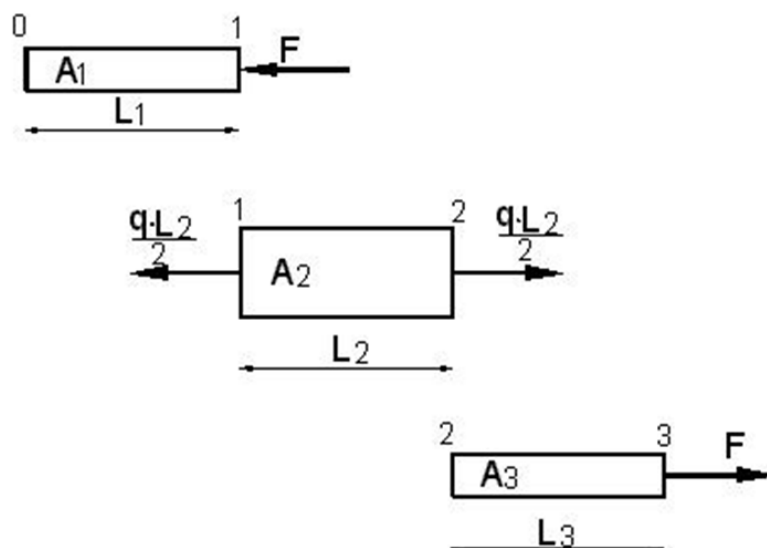


Рис. 3.3

Матрица жесткости одного элемента e имеет вид

$$\frac{E \cdot A_e}{L_e} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

где E – модуль упругости элемента; A – площадь поперечного сечения элемента; L – длина элемента.

Условие равновесия всей системы:

$$\{F\} = [K] \cdot \{U\},$$

где $\{F\}$ – матрица-вектор узловых внешних сил; $[K]$ – глобальная матрица жесткости; $\{U\}$ – матрица-вектор узловых перемещений.

Задача сводится к тому, чтобы, зная матрицы жесткости каждого элемента, на первом этапе получить матрицу жесткости всей системы (глобальную матрицу жесткости), а затем, задав граничные условия, определить с ее помощью неизвестные перемещения и усилия в узлах заданного стержня.

Глобальная матрица системы, разбитой на три КЭ, имеет вид

$$\begin{vmatrix} K^{(1)} & 0 & 0 \\ 0 & K^{(2)} & 0 \\ 0 & 0 & K^{(3)} \end{vmatrix}$$

Каждая матрица КЭ по отношению к предыдущей сдвигается на одну ячейку вправо и одну вниз, суммируя тем самым перемещения одного и того же узла.

Граничные условия для глобальной матрицы задаются исходя из условий закрепления стержня. На схеме (см. рис. 3.2), согласно закреплению нулевого узла, его перемещения должны быть равны нулю.

Матрица-вектор внешних нагрузок составляется в соответствии с приложенными внешними силами (опорные реакции в данном случае как величины неизвестные внешними силами не считаются). Распределенная нагрузка на элементе заменяется равнодействующей и прикладывается в узлы КЭ (см. 2-й КЭ на рис. 3.3).

Вектор перемещений определяется из условия равновесия системы (предварительно в глобальной матрице задаются граничные условия системы):

$$U := \text{lsolve}(K, F)$$

После вычисления перемещений в узлах находят все остальные неизвестные величины (относительные деформации, напряжения, внутренние продольные усилия).

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Самостоятельная работа «Исследование функции»

Цель работы: научиться выполнять в программе MathCAD несложные математические вычисления (нахождение корней функции, производных), строить и оформлять двухмерные графики на примере исследования функции.

К этой работе нет индивидуального задания, её можно выполнить по примерам, приведенным далее.

Методические указания к выполнению задания:

- присвоить выражение функции $f(x)$;
- найти область определения функции;
- найти точки пересечения функции с осью абсцисс (корни функции) и осью ординат $f(0)$. Для нахождения корней функции использовать команду *solve*;
- исследовать функцию на экстремальные значения при помощи производной 1-го порядка. Найти критические точки первой производной: корни производной и возможные точки разрыва. Исследовать найденные точки на максимум и минимум, определить интервалы возрастания/убывания функции. Показать на графике функции экстремальные значения;
- исследовать функцию на выпуклость, вогнутость и точки перегиба. Для этого вычислить производную 2-го порядка и найти ее критические точки (корни второй производной и возможные точки разрыва);
- построить все необходимые графики функции. На графике должны быть показаны корни функции, точки минимума и максимума, точки перегиба. Для наглядности следует построить несколько графиков в разных масштабах с указанием тех или иных точек;
- ввести все необходимые текстовые пояснения.

Задание 1. Решение нелинейных уравнений

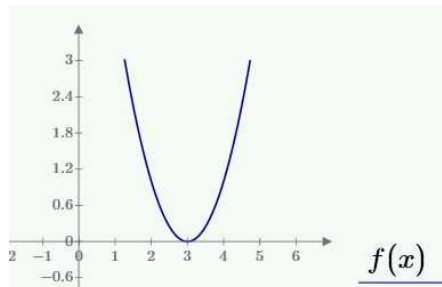
Цель работы: овладеть навыками численного решения нелинейных уравнений с применением функций и методов программирования.

Индивидуальное задание взять из таблицы ниже согласно номеру варианта, выданному преподавателем.

Вариант	Уравнение	Вариант	Уравнение
1	$x^3 + 3x - 2e^{-x} = 0$	16	$6x^5 - 1.25x^2 + 9x^4 = 1$
2	$2x^2 + \cos(x) = 3$	17	$(2x+3)^3 + (x-1)^4 = 16$
3	$2x^2 + \ln(x+5) = 5$	18	$\sin(2 \cdot x) + 1 = x^2$
4	$3x^2 - \cos(\pi \cdot x)^2$	19	$\log(0.01+x) + \frac{0.1}{x^2} = 0.3$
5	$2x^2 - 1.5^x = 1$	20	$e^x + x^2 = 1$
6	$x^2 - \ln(x) - 1$	21	$\cos\left(\pi \cdot x + \frac{\pi}{4}\right) = \frac{\pi}{4} \cdot x$
7	$\sqrt[3]{x} - e^x + 2$	22	$(x-2)^x + 12 = x^3$
8	$4x^3 - (x-2)^4 - 3x^5$	23	$\sqrt[3]{x^2 - 4} = 2$
9	$x^3 \cdot (x-3)^2 = 4x^7$	24	$x^{2x-7} = 13$
10	$3x^3 - \sqrt{(x+4)} + 2$	25	$2x^4 - \sqrt[3]{\cos(0.5 \cdot x)} = 0$
11	$4^x = 6x$	26	$e^{x-1} = x+1$
12	$2x^2 + \cos(x) = 7$	27	$3x^5 - 2.5 \cdot x^3 + 3x^2 - 4x = 6$
13	$(x-1)^4 - \frac{1}{x} = 7$	28	$(3 \cdot x - 5)^4 - x^3 + 3x^2 = 4$
14	$\frac{x^2}{2} - \ln(x) = 2$	29	$x^{x-2} - \ln(x) = 4$
15	$4x^5 - \sqrt[3]{x-3} = 12x$	30	$6^{x-2} - \ln(x+1) = 0$

Методические указания к выполнению задания:

- определить, к какому типу относится заданное уравнение (алгебраическому или трансцендентному);
- преобразовать заданное уравнение в выражение для функции и присвоить это выражение функции $f(x)$;
- построить график функции $f(x)$;
- проанализировать график, выполнив отделение корней. Для этого создать одну или несколько копий графика, укрупнив фрагмент с интервалом, в котором четко виден корень функции. Записать интервалы для поиска корней функции, ссылаясь на график функции (фрагмент графика);
- уточнить корни нелинейного уравнения, используя несколько способов решения:
 - функцию *root*;
 - вычислительный блок с применением функции *Find*;
 - функцию *polyroots* (только для алгебраических уравнений);
 - программирование: составить функции-программы для решения методом половинного деления и методом касательных (методом Ньютона).



Метод половинного деления не может быть применен, если график функции не пересекает ось абсцисс, а только касается её. В этом случае достаточно составить функцию-программу методом касательных;

- показать на графике найденные корни уравнения;
- сравнить полученные результаты.

Задание 2. Решение системы линейных уравнений

Цель работы: на примере системы линейных уравнений с четырьмя неизвестными научиться средствами MathCAD формировать в матричном виде систему линейных уравнений и применять различные методы для ее решения.

Индивидуальное задание получить преобразованием общей системы уравнений, подставив вместо N номер варианта, выданный преподавателем.

$$\begin{aligned}
2.26 \cdot N \cdot x_1 + 4.76 \cdot N \cdot x_2 + 2.78 \cdot N \cdot x_4 + 2.96 \cdot N \cdot x_3 &= 0.9 + 0.5 \cdot N \\
1.68 \cdot N \cdot x_4 - 2 \cdot N \cdot x_3 - 2.68 \cdot N \cdot x_1 + 2.2 \cdot N \cdot x_2 &= -0.41 + 0.15 \cdot N \\
-0.46 \cdot N \cdot x_1 + 1.7 \cdot N \cdot x_2 - 0.96 \cdot N \cdot x_4 &= 0.65 + 0.2 \cdot N \\
0.96 \cdot N \cdot x_3 - 1.3 \cdot N \cdot x_4 + 0.22 \cdot N \cdot x_1 - 2.76 \cdot N \cdot x_2 &= -0.75 + 0.1 \cdot N
\end{aligned}$$

Методические указания к выполнению задания:

- назначить системную переменную $\text{ORIGIN} = 1$;
- переписать систему уравнений в рабочую область MathCAD в общем виде. Выполнить подстановку в каждой строке уравнения, используя функцию *substitute*, например:

$$2.26 \cdot N \cdot x_1 + 4.76 \cdot N \cdot x_2 + 2.78 \cdot N \cdot x_4 + 2.96 \cdot N \cdot x_3 = 0.91 \cdot N \text{ substitute, } N = 41 \rightarrow 92.66 \cdot x_1 + 195.16 \cdot x_2 + 121.36 \cdot x_3 + 113.98 \cdot x_4 = 37.31$$

- скопировать результаты подстановки в отдельные строки, сформировав систему из четырех линейных уравнений;

- сформировать матрицу коэффициентов при неизвестных x_1, x_2, x_3, x_4 , строго соблюдая порядок вставки коэффициентов в ячейки. Можно вставлять в ячейки множитель коэффициента N , предварительно задав его в рабочей области (тогда матрица сама пересчитает значения коэффициентов). Убедиться, что определитель матрицы коэффициентов не равен нулю. Это будет означать, что система имеет единственное решение;

- сформировать вектор свободных членов уравнения;
- решить систему несколькими методами:
 - с помощью матричного уравнения;
 - с помощью функции *lsolve*;
 - составив функцию-программу для решения системы по методу Крамера;
 - с помощью вычислительного блока, используя функцию *Find*;
- для проверки матричного решения использовать вычислительный блок, в котором система уравнений записывается в исходном виде;
- сравнить полученные результаты.

Задание 3. Решение обыкновенных дифференциальных уравнений 2-го порядка

Цель работы: в среде MathCAD применить численные методы для решения ОДУ 2-го порядка.

Найти численное решение линейного однородного уравнения 2-го порядка¹

$$y'' + a_1 \cdot y' + a_2 \cdot y = 0$$

с начальными условиями (задача Коши).

Изобразить график решения и табличный вывод для x , y , y' .

Индивидуальное задание взять из таблицы ниже согласно номеру варианта, выданному преподавателем.

Вариант	a1	a2	y(x ₀)	y'(x ₀)	x ₀	x _k
1	2	0	1	1	0	5
2	-4	4	0	1	-0.75	5
3	2	1	5	0	2	7
4	0	1	3	0	-2	3
5	2	5	8	0	1	6
6	-3	4	1	1	0.3	6
7	6	13	-1	1	0.25	5
8	-8	1	4	1	3	8
9	-2	5	6	2	-4	1
10	-4	8	0	2	1.5	6
11	2	3	0	2	0.2	5
12	-3	4	-1	0.5	2.5	8
13	3	1	1	1	6	11
14	5	5	4	-0.5	-3	2
15	6	10	5	2	0.8	6
16	-4	6	0	0.5	0.75	6
17	2	-3	1	1	-1.2	5
18	-4	12	-1	0	4	9
19	6	9	-2	2	1.75	7
20	4	1	8	0.5	0.35	6

¹ При записи выражений с производными 1-го и 2-го порядков используется символ «штрих». Для его ввода в MathCAD 14 применяется комбинация клавиш Ctrl и «F7», в PTC MathCAD – комбинация клавиш Ctrl и «апостроф».

Окончание таблицы

Вариант	a1	a2	$y(x_0)$	$y'(x_0)$	x_0	x_k
21	0	-1	1	0	-2.5	3
22	-5	4	1	1	0.35	6
23	5	-1	2	0	0.6	6
24	-1	1	1	0	0.5	6
25	7	2	2	0	3.2	9
26	-4	3	6	10	0.1	5
27	4	29	0	15	10	15
28	2	8	1	-2	8	13
29	4	-3	2	4	1	6
30	1	0.25	2	0	9	14

Методические указания к выполнению задания:

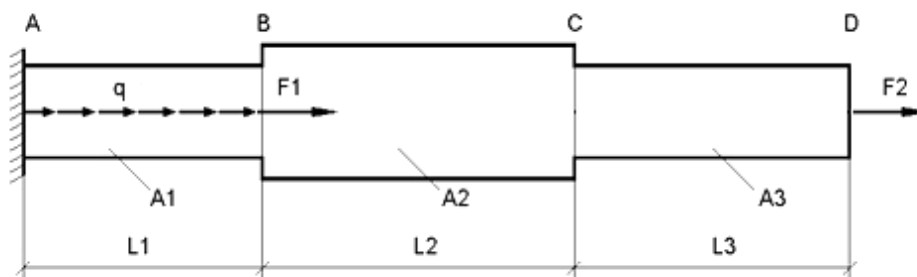
- задать системную переменную **ORIGIN = 1**;
- записать дифференциальное уравнение в общем виде. Задать необходимые начальные данные (присвоить коэффициентам a1 и a2 значения согласно варианту, задать данные интервала интегрирования x_0 , x_k , количество точек (шагов) интегрирования N, шаг интегрирования h и значения аргумента x в пределах интервала интегрирования);
- решить уравнение двумя способами:
 - с помощью функции *odesolve*; результат решения вывести в табличной и графической формах;
 - с помощью функции *rkfixed*; результат решения вывести в табличной и графической формах;
- сравнить полученные результаты (сопоставить табличный и графический выводы).

Задание 4. Расчет методом конечных элементов перемещений и усилий в стержневой конструкции от действия продольной нагрузки

Цель работы: научиться составлять конечно-элементную модель заданной конструкции, матрицы жесткости системы, задавать граничные условия, вычислять усилия в конструкции, используя МКЭ.

Индивидуальное задание взять из таблицы и рисунка ниже (схема нагрузки стержня) согласно номеру варианта, выданному преподавателем.

Ва- ри- ант	F1		F2		q		A1	A2	A3	L1	L2	L3
	кН	сеч	кН	сеч	кН/м	уч-к	10^{-4}м^2	10^{-4}м^2	10^{-4}м^2	м	м	м
1	7	B	-10	D	12	BC	2.0	4.0	2.0	0.25	0.36	0.95
2	-8	C	12	B	-10	CD	2.5	2.5	5.0	0.75	0.62	0.26
3	9	D	-14	C	8	AB	2.1	4.2	2.1	0.90	0.32	0.65
4	-10	B	16	D	-6	BC	2.2	4.4	2.2	0.80	0.42	0.95
5	11	C	-18	B	6	CD	2.3	4.6	2.3	0.60	0.52	1.10
6	-12	D	20	C	-8	AB	2.4	4.8	2.4	0.55	0.75	0.98
7	13	B	-8	D	10	BC	2.6	5.2	2.6	0.45	0.80	1.04
8	-14	C	10	B	-12	CD	2.7	5.4	2.7	0.35	0.70	1.22
9	15	D	-12	C	6	AB	2.8	5.6	2.8	0.40	0.92	0.58
10	-16	B	14	D	-8	BC	2.9	5.8	2.9	0.70	0.45	0.85
11	17	C	-16	B	10	CD	3.0	6.0	3.0	0.26	0.68	0.94
12	-18	D	18	C	-12	CD	2.1	2.1	4.2	0.34	0.85	0.68
13	19	B	-20	C	-6	CB	2.2	2.2	4.4	0.68	0.36	0.90
14	-20	C	10	D	8	AB	2.3	2.3	4.6	0.72	0.44	0.66
15	-7	D	12	B	-10	BC	2.4	2.4	4.8	0.48	0.85	0.35
16	8	B	-14	C	12	CD	2.5	2.5	5.0	0.56	0.74	0.94
17	-9	C	16	D	-12	AB	2.6	2.6	5.2	0.66	0.45	0.82
18	10	D	-18	B	10	BC	2.7	2.7	5.4	0.70	0.38	0.46
19	-11	B	20	D	-8	CD	2.8	2.8	5.6	0.44	0.76	0.84
20	12	C	-10	B	6	AB	2.9	2.9	5.8	0.92	0.35	0.55
21	-13	D	12	C	6	BC	3.0	3.0	6.0	0.75	0.46	0.98
22	14	B	-14	D	-8	CD	6.0	3.0	3.0	0.58	0.70	1.10
23	-15	C	16	B	10	AB	5.8	2.9	2.9	0.48	0.65	1.00
24	16	D	-18	C	-12	BC	5.6	2.8	2.8	0.60	0.42	1.20
25	-17	B	20	C	12	CD	5.4	2.7	2.7	0.38	0.85	1.05
26	18	C	-10	D	-10	AB	5.2	2.6	2.6	0.46	0.90	0.82
27	-19	D	12	B	8	BC	5.0	2.5	2.5	0.64	0.55	1.30
28	20	B	-14	C	-6	CD	4.8	2.4	2.4	0.38	0.88	1.10
29	-15	C	16	D	10	AB	4.6	2.3	2.3	0.80	0.65	1.20
30	6	D	-18	B	8	BC	4.4	2.2	2.2	0.55	0.72	1.25



Методические указания к выполнению задания: используя МКЭ, определить в стальном ступенчатом стержне круглого сечения перемещения, относительные деформации, внутренние усилия и напряжения от действия продольной нагрузки; построить эпюры перемещений, внутренних усилий и напряжений; оценить прочность стержня; принять расчетное сопротивление $R = 200$ МПа, модуль упругости $E = 2 \cdot 10^5$ МПа.

Решение находят численным (МКЭ) и аналитическим методами, полученные результаты сравниваются.

1. Численное решение

Системной переменной ORIGIN задать значение единицы.

Разбить стержень на конечные элементы, каждому КЭ и узлам (границам КЭ) присвоить номера слева направо, начиная с единицы.

Длину и площадь КЭ задать в матричном виде.

Составить для каждого i -го КЭ матрицу жесткости K_i , предварительно задав геометрические и физические характеристики:

$$K_e := \frac{E \cdot A_e}{L_e} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Составить глобальную матрицу системы K . Для объединения матриц жесткости КЭ используйте базовую нулевую матрицу $K0$ с размерами $n \times n$ (n – количество узлов системы) и программу для вставки подматрицы.

$$\text{Rep}(A, B, r, c) := \left| \begin{array}{l} R \leftarrow A \\ \text{for } i \in 1.. \text{rows}(B) \\ \quad \text{for } j \in 1.. \text{cols}(B) \\ \qquad R_{i+r-1, j+c-1} \leftarrow B_{i, j} \\ R \end{array} \right|$$

r, c – номер строки и столбца, с которых начинается вставка подматрицы B (K_e) в матрицу A ($K0$).

Задать граничные условия элементам глобальной матрицы жесткости исходя из условий закрепления. Во избежание получения сингулярной (вырожденной) матрицы необходимо задать вместо нуля первому обнуляемому элементу значение высокого порядка (например, 10^{20}).

Задать вектор узловых внешних нагрузок. Нагрузка в жесткой заделке считается неизвестной. Распределенная нагрузка заменяется двумя сосредоточенными силами, приложенными пропорционально к узлам КЭ. Учитывать знак внешней нагрузки, считая ее положительной, если ее вектор совпадает с положительным направлением оси X КЭ.

Вычислить узловые перемещения:

$U := \text{Isolve}(K, F)$

Определить относительную деформацию, напряжения и усилия в узлах КЭ.

Построить эпюры абсолютных и относительных деформаций, напряжений и продольных усилий по полученным узловым значениям.

Оценить прочность конструкции по максимальному напряжению.

2. Аналитическое решение

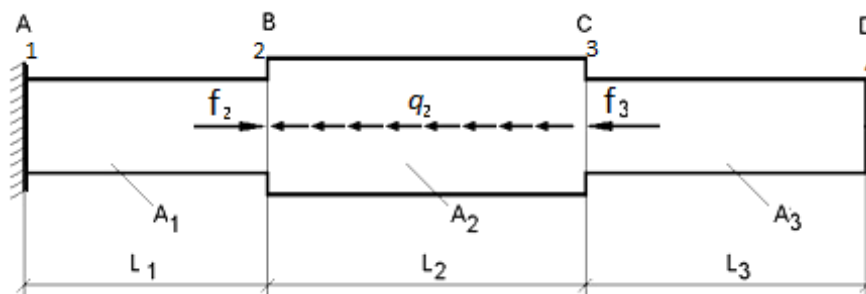
Вычислить опорные реакции, внутренние усилия, напряжения и перемещения в каждом участке и характерном сечении, совпадающем с узлом КЭ.

Сравнить перемещения в сечениях, полученные аналитическим методом, с перемещениями в узлах, найденными численно.

Пример ввода исходных данных согласно варианту представлен ниже.

Вариант	F1		F2		q		A1	A2	A3	L1	L2	L3
	кН	сеч	кН	сеч	кН/м	уч-к	10^{-4}м^2	10^{-4}м^2	10^{-4}м^2	м	м	м
**	5	В	-6	С	-11	ВС	2.0	4.0	2.0	0.3	0.5	0.4

Преобразовать схему нагрузки стержня, представленную ранее, в схему, соответствующую варианту.



Задать вектор сосредоточенных сил.

Узел 1 (A): $f_1 = 0$ (это узел заделки, реакция нам неизвестна, поэтому нуль).

Узел 2 (B): $f_2 = 5$ (в таблице это сила F1, приложенная в сечении B, а это узел 2).

Узел 3 (C): $f_3 = -6$ (в таблице это сила F2, приложенная в сечении C, а это узел 3. Направление силы отрицательное, её значение должно быть со знаком «-»).

Узел 4 (D): $f_4 = 0$ (в сечении D сила не задана).

Задать вектор распределенных нагрузок на участках AB, BC, CD (конечных элементах 1, 2 и 3 соответственно).

КЭ 1: $q_1 = 0$ (на длине L1 распределенная нагрузка не приложена).

КЭ 2: $q_2 = -11$ (на длине L2 распределенная нагрузка задана, учитываем её направление и ставим знак «-»).

КЭ 3: $q_3 = 0$ (на длине L3 распределенная нагрузка не приложена).

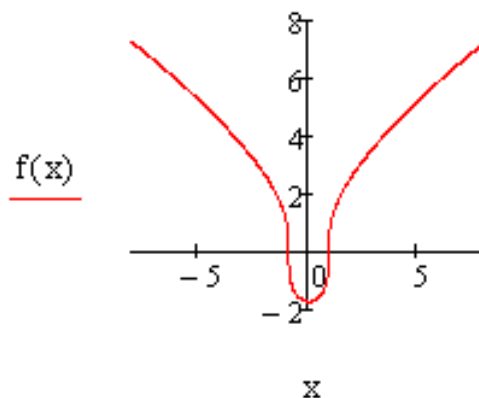
ПРИМЕРЫ РЕШЕНИЙ С ПОМОЩЬЮ MATHCAD 14

Самостоятельная работа «Исследование функции»

Дана функция

$$f(x) := \sqrt[3]{6x^2 - x - 5}$$

График функции построен в общем виде.



Область определения функции:

$$-\infty < x < \infty$$

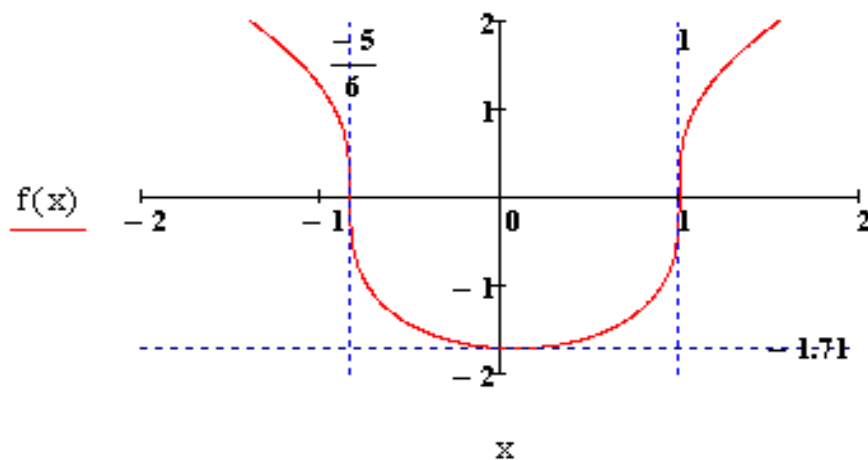
Функция определена на всей числовой оси x и разрывов не имеет.

Корни функции:

$$f(x) = 0 \text{ solve } \rightarrow \begin{pmatrix} 1 \\ -\frac{5}{6} \end{pmatrix}$$

$$f(0) = -1.71$$

Корни на графике отмечены маркерами.



Исследуем функцию на экстремальные значения.

1. Найдем производную 1-го порядка:

$$\frac{d}{dx}f(x) \rightarrow \frac{12 \cdot x - 1}{3 \cdot (6 \cdot x^2 - x - 5)^{\frac{2}{3}}}$$

2. Введем новую функцию для обозначения производной:

$$f1(x) := \frac{12 \cdot x - 1}{3 \cdot \sqrt[3]{(6 \cdot x^2 - x - 5)^2}}$$

Примечание. Во избежание ошибочных решений при подстановке конкретных значений аргумента (могут появиться комплексные числа) замените в знаменателе дробную степень радикалом.

3. Найдем критические точки производной (для удобства переведем результат в десятичный формат):

$$f1(x) = 0 \quad \left| \begin{array}{l} \text{solve} \\ \text{float}, 3 \end{array} \right. \rightarrow 0.0833 \quad 3 \cdot \sqrt[3]{(6 \cdot x^2 - x - 5)^2} = 0 \quad \left| \begin{array}{l} \text{solve} \\ \text{float}, 3 \end{array} \right. \rightarrow \begin{pmatrix} 1.0 \\ -0.833 \end{pmatrix}$$

x1 = -0.833 (точка разрыва производной)

x2 = 0.0833 (корень производной)

x3 = 1.0 (точка разрыва производной)

Расположив все критические точки по порядку на числовой оси X, исследуем их на максимум/минимум функции:

f1(-1) = -2.73
x1 не является экстремумом

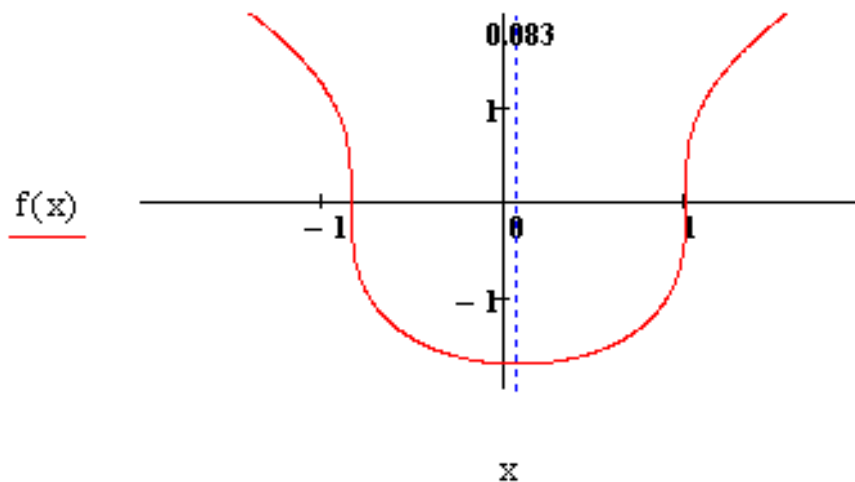
f1(-0.5) = -1.122
x2 является минимумом

f1(0.5) = 0.661
x3 не является экстремумом

f1(2) = 1.16

В окрестности корня x2 = 0.0833 производная меняет знак с «-» на «+», следовательно, в точке x = 0.0833 функция имеет минимум.

Таким образом, функция имеет два участка возрастания/убывания: на интервале $-\infty < x < 0.0833$ производная меньше нуля, функция убывает; на интервале $0.0833 < x < \infty$ производная больше нуля, функция возрастает.



Исследуем функцию на выпуклость/вогнутость и точки перегиба.

1. Найдем производную 2-го порядка:

$$\frac{d^2}{dx^2} f(x) \text{ simplify } \rightarrow -\frac{2 \cdot (36 \cdot x^2 - 6 \cdot x + 91)}{9 \cdot (6 \cdot x^2 - x - 5)^{\frac{5}{3}}}$$

2. Введем новую функцию для обозначения производной:

$$f2(x) := -\frac{2 \cdot (36 \cdot x^2 - 6 \cdot x + 91)}{9 \cdot \sqrt[3]{(6 \cdot x^2 - x - 5)^5}}$$

3. Найдем критические точки производной 2-го порядка:

$$f2(x) = 0 \quad \left| \begin{array}{l} \text{solve} \\ \text{float}, 3 \end{array} \right. \rightarrow \begin{pmatrix} 0.0833 + 1.59i \\ 0.0833 - 1.59i \end{pmatrix} \quad 9 \cdot \sqrt[3]{(6 \cdot x^2 - x - 5)^5} = 0 \quad \left| \begin{array}{l} \text{solve} \\ \text{float}, 3 \end{array} \right. \rightarrow \begin{pmatrix} 1.0 \\ -0.833 \end{pmatrix}$$

Производная не имеет корней, однако не определена (имеет разрыв) в точках: $x_1 = -0.833$, $x_2 = 1.0$.

Исследуем x_1 и x_2 на возможные точки перегиба функции:

$$f2(-1) = -9.309$$

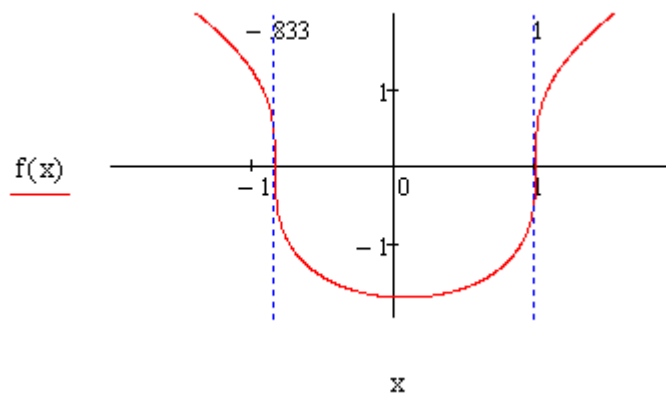
x_1 является точкой перегиба

$$f2(0) = 1.383$$

x_2 является точкой перегиба

$$f2(2) = -0.441$$

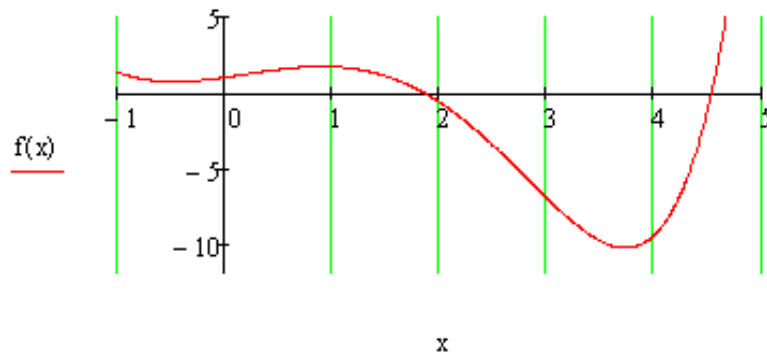
Получим три интервала выпуклости/вогнутости функции: функция выпукла при $x < -0.833$ и при $x > 1$; функция вогнута при $-0.833 < x < 1$.



Задание 1. Решение нелинейных уравнений

Дано уравнение $e^x = x^3$.

Задаем функцию $f(x) := e^x - x^3$.



Решение функцией *root*: на графике выше определяем интервалы аргумента, внутри которых функция обращается в нуль.

$$\text{root}(f(x), x, 1, 2) = 1.857$$

$$\text{root}(f(x), x, 4, 5) = 4.536$$

Корни уравнения: $x_1 = 1.857$, $x_2 = 4.536$.

Решение вычислительным блоком или функцией *polyroots*:

$$x := 1$$

Given

$$e^x - x^3 = 0$$

$$x0 := \text{Find}(x)$$

$$x0 = 1.857$$

Уравнения-полиномы решаются в этом задании функцией *polyroots*. Необходимо составить вектор коэффициентов от низшего a_0 при x^0 до высшего a^n при x^n при помощи команды *coeffs* (панель *Symbolic*). Отсутствующие показатели степеней будут в векторе коэффициентов равны нулю. Для вывода коэффициентов командой *coeffs* необходимо записать только левую часть уравнения. Пример нахождения корней для уравнения $2x - (2x)^5 + 3x^3 - 7x^2 = 0$.

$$2x - (2x)^5 + 3x^3 - 7x^2 \text{ coeffs} \rightarrow \begin{pmatrix} 0 \\ 2 \\ -7 \\ 3 \\ 0 \\ -32 \end{pmatrix} \quad v := \begin{pmatrix} 0 \\ 2 \\ -7 \\ 3 \\ 0 \\ -32 \end{pmatrix} \quad \text{polyroots}(v) = \begin{pmatrix} -0.72 \\ 0 \\ 0.215 - 0.504i \\ 0.215 + 0.504i \\ 0.29 \end{pmatrix}$$

Решение методом половинного деления: составим программу (функция $f(x)$ задается вне программы).

$$D(f, a, b) := \left| \begin{array}{l} \text{while } \left| \frac{a - b}{2} \right| > \text{TOL} \\ \quad \left| \begin{array}{l} x1 \leftarrow \frac{a + b}{2} \\ b \leftarrow x1 \text{ if } f(x1) \cdot f(a) < 0 \\ a \leftarrow x1 \text{ otherwise} \end{array} \right. \\ x \leftarrow x1 \end{array} \right.$$

$$D(f, 0, 2) = 1.857$$

$$D(f, 3, 6) = 4.537$$

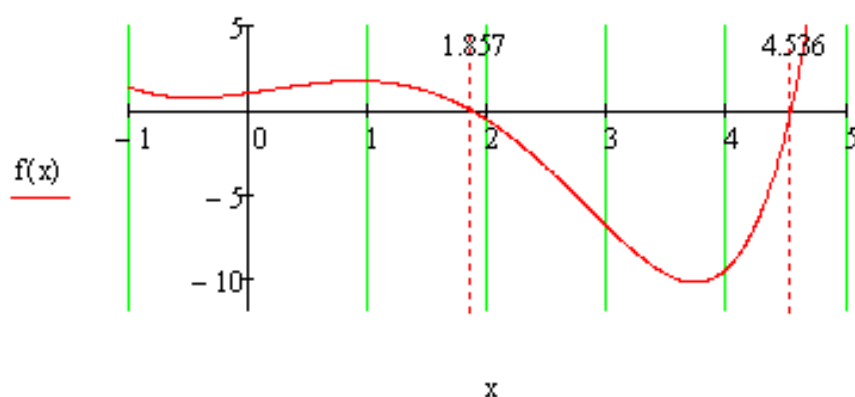
Решение методом Ньютона (функция $f(x)$ задается вне программы):

```

Newton(f, x) :=
  x1 ← x + 2 · TOL
  while |x1 - x| > TOL
    x ← x1
    x1 ← x -  $\frac{f(x)}{\frac{d}{dx}f(x)}$ 

```

На графике искомой функции необходимо показать найденные корни.



Задание 2. Решение системы линейных уравнений

Дана система из четырех уравнений с четырьмя неизвестными x_1, x_2, x_3, x_4 .

$$2 \cdot x_1 + 0.5 \cdot x_2 - 2.7 \cdot x_3 + 9.2 \cdot x_4 = 6$$

$$x_1 - 1.6x_2 + 0.6x_3 - 0.7x_4 = 2$$

$$3x_1 + 1.5x_2 + 8.5x_3 + 4.2x_4 = -3$$

$$-5.2x_1 + 6x_2 + 3.4x_3 - 5x_4 = 2.5$$

Формируем матрицу системы A и матрицу правой части B :

$$A := \begin{pmatrix} 2 & 0.5 & -2.7 & 9.2 \\ 1 & -1.6 & 0.6 & -0.7 \\ 3 & 1.5 & 8.5 & 4.2 \\ -5.2 & 6 & 3.4 & -5 \end{pmatrix} \quad B := \begin{pmatrix} 6 \\ 2 \\ -3 \\ 2.5 \end{pmatrix} \quad |A| = -78.723$$

Задаем системную переменную $\text{ORIGIN}:=1$.

Находим неизвестные по формуле матричной записи системы.

$$x := A^{-1} \cdot B \quad x = \begin{pmatrix} -31.075 \\ -22.227 \\ 8.978 \\ 11.25 \end{pmatrix}$$

Находим неизвестные по методу Крамера. Для вычисления неизвестных составим алгоритм.

```

Kram(A,B) := if cols(A) = rows(A)
               for i ∈ 1..cols(A)
                   C ← A
                   C(i) ← B
                   if |A| ≠ 0
                       xi ←  $\frac{|C|}{|A|}$ 
                       i ← i + 1
                   return "Нет корней: матрица сингулярная" otherwise
               x
           return "Нет корней: матрица неквадратная" otherwise
    
```

$$\text{Kram}(A,B) = \begin{pmatrix} -31.075 \\ -22.227 \\ 8.978 \\ 11.25 \end{pmatrix}$$

Находим неизвестные функцией *lsolve*.

$$x := \text{lsolve}(A,B) \quad x = \begin{pmatrix} -31.075 \\ -22.227 \\ 8.978 \\ 11.25 \end{pmatrix}$$

Проверяем решение вычислительным блоком.

$$2 \cdot x_1 + 0.5 \cdot x_2 - 2.7 \cdot x_3 + 9.2 \cdot x_4 = 6$$

$$x_1 - 1.6x_2 + 0.6x_3 - 0.7x_4 = 2$$

$$3x_1 + 1.5x_2 + 8.5x_3 + 4.2x_4 = -3$$

$$-5.2x_1 + 6x_2 + 3.4x_3 - 5x_4 = 2.5$$

$$\text{Find}(x_1, x_2, x_3, x_4) \rightarrow \begin{pmatrix} -31.075 \\ -22.227 \\ 8.9781 \\ 11.250 \end{pmatrix}$$

Задание 3. Решение обыкновенных дифференциальных уравнений 2-го порядка

Дано уравнение $y''(x) + a_1 \cdot y'(x) + a_2 \cdot y(x) = 0$. Начальные условия следующие: $x_0 := 1$, $y_0(x_0) := 6$, $y_0'(x_0) := 2$, интервал интегрирования (x_0, x_k) , где $x_k := 6$, количество точек интегрирования $N := 100$, шаг интегрирования $h := \frac{x_k - x_0}{N}$.

Значения аргумента x задаем ранжированной переменной

$$x := x_0, x_0 + h..x_k$$

Для численного решения используем две функции: *odesolve* и *rkfixed*.

Задаем системную переменную $\text{ORIGIN} := 1$.

Функция *odesolve*:

Given

$$z''(x) + a_1 z'(x) + a_2 z(x) = 0$$

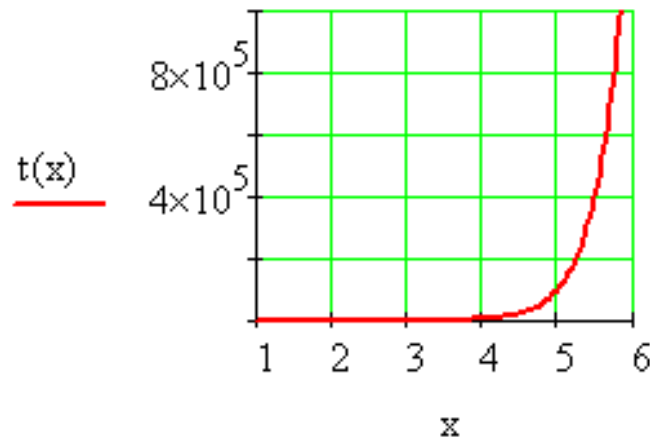
$$z(x_0) = y_0(x_0)$$

$$z'(x_0) = y_0'(x_0)$$

$$t := \text{odesolve}(x, x_k)$$

Для ввода символа «штрих» используйте комбинацию клавиш Ctrl и «F7».

Построим график функции.



Вывод численного решения функцией *odesolve* можно представить в виде матрицы-вектора, если предварительно составить функцию-программу.

```

od(x0,xk,N,f) :=
  x ← x0
  h ← (xk - x0) / N
  for i ∈ 1..N
    odi ← f(x)
    x ← x + h
  od

```

Здесь x_0 , x_k – интервал интегрирования, N – количество точек, f – функция, заданная в вычислительном блоке *odesolve* (в примере функция задана символом t). Для табличного вывода зададим вектор $t1 := od(x_0, x_k, N, t)$

Функция *rkfixed*: для формирования параметров функции необходимо составить вектор начальных условий y и дифференциальное выражение $D(x, y)$.

Вектор начальных условий

$$y := \begin{pmatrix} y_0(x_0) \\ y_0'(x_0) \end{pmatrix}$$

Дифференциальное выражение

$$D(x, y) := \begin{pmatrix} y_2 \\ -a_1 y_2 - a_2 y_1 \end{pmatrix}$$

Функция *rkfixed*:

$t2 := rkfixed(yx0, xk, N, D)$

$t2 =$

	1	2	3
1	1	6	2
2	1.05	6.121	2.847
3	1.1	6.286	3.799
4	1.15	6.503	4.87
5	1.2	6.776	6.08
6	1.25	7.113	7.448
7	1.3	7.524	9
8	1.35	8.017	10.763
9	1.4	8.604	12.768
10	1.45	9.298	15.05

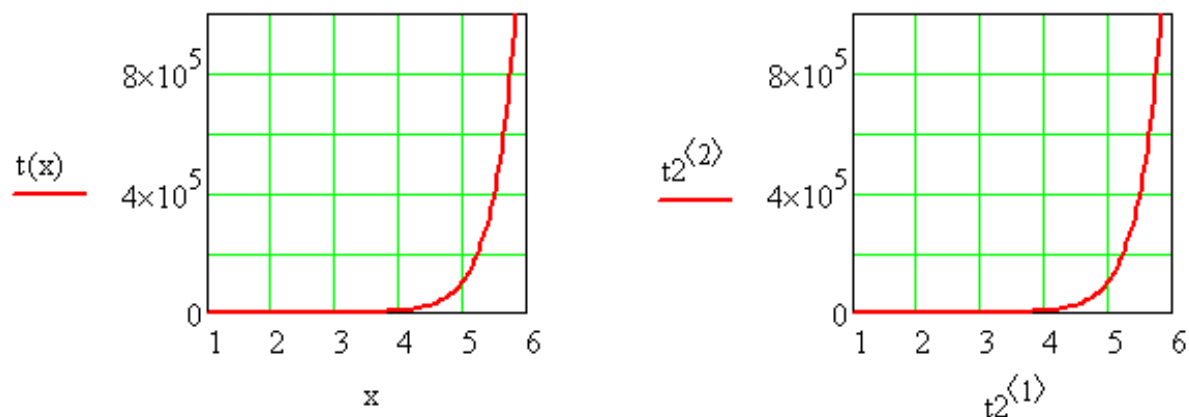
Решение функция выдает в матричной форме, первый столбец – аргумент x , второй – функция $y(x)$, третий – производная 1-го порядка $y'(x)$. Для представления функции в графической форме следует задать в качестве аргумента значения первого столбца $t2^{<1>}$, в качестве функции – значения второго столбца $t2^{<2>}$.

Сравнительный анализ полученных результатов в табличной форме выглядит следующим образом.

$x =$			
1		1	1
1.05		2	1
1.1		3	1
1.15		4	1
1.2		5	1
1.25		6	1
1.3		7	1
1.35	$t1 =$	8	$t2 =$
1.4		9	1
1.45		10	1
1.5		11	1
1.55		12	1
1.6		13	1
1.65		14	1
1.7		15	1

Табличный вывод (первые 15 значений) показывает, что оба способа решения имеют сходные результаты.

Сравнительный анализ полученных результатов в графической форме выглядит следующим образом.



Очевидно, что полученные графики идентичны.

Задание 4. Расчет методом конечных элементов перемещений и усилий в стержне от действия продольной нагрузки

Стержень разбивается на три КЭ в соответствии с переменной площадью поперечных сечений и приложенной нагрузкой. Задаем системную переменную, управляющую нумерацией начального элемента матрицы: ORIGIN:=1.



Нумерация КЭ – слева направо: 1, 2, 3; нумерация узлов – слева направо: 1, 2, 3, 4.

Задаем количество КЭ переменной ne:=3; количество узлов переменной nu:=4. Счетчик КЭ el:=1..ne.

Начальные данные КЭ задаем как элементы матриц.

$$\begin{array}{ll} \text{Площади} & \text{Длина КЭ} \\ \mathbf{A} := \begin{pmatrix} 2 \\ 4 \\ 5 \end{pmatrix} \cdot 10^{-4} \text{ м}^2 & \mathbf{L} := \begin{pmatrix} 0.1 \\ 0.2 \\ 0.1 \end{pmatrix} \text{ м} \end{array}$$

Внешние нагрузки задаем в соответствии с направлением по оси X и нумеруем согласно узлу приложения (сосредоточенные силы, приложенные в узле) или КЭ (распределенные нагрузки). Поскольку реакция заделки неизвестна, считаем ее равной нулю. Если соответствующей нагрузки нет, ее также задаем равной нулю.

$$\mathbf{q} := \begin{pmatrix} 20 \\ 0 \\ 0 \end{pmatrix} \cdot 10^3 \frac{\text{Н}}{\text{м}} \quad \mathbf{f} := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 10 \end{pmatrix} \cdot 10^3 \text{ Н}$$

Модуль упругости $E := 2 \cdot 10^{11} \text{ Н/м}^2$.

Формируем глобальную матрицу жесткости системы. Матрица жесткости КЭ:

$$\mathbf{kk} := \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad \mathbf{K}_{el} := \mathbf{kk} \cdot \frac{E \cdot A_{el}}{L_{el}}$$

Глобальная матрица жесткости системы формируется из суммы матриц жесткости КЭ со сдвигом на одну ячейку вправо и вниз. Предварительно создадим базовую нулевую матрицу с размерами 4×4 :

$$\mathbf{K0}_{nu,nu} := 0,$$

где $\mathbf{K0}$ – базовая нулевая матрица.

Программа-функция *Rep* создается для вставки подматриц КЭ в матрицу $\mathbf{K0}$ со сдвигом на одну ячейку вправо и вниз; r, c – номер соответственно строки и столбца, с которых начинается вставка подматрицы в матрицу; $\mathbf{A}$ – базовая матрица; $\mathbf{B}$ – подматрица.

$$\text{Rep}(\mathbf{A}, \mathbf{B}, r, c) := \left| \begin{array}{l} \mathbf{R} \leftarrow \mathbf{A} \\ \text{for } i \in 1.. \text{rows}(\mathbf{B}) \\ \quad \text{for } j \in 1.. \text{cols}(\mathbf{B}) \\ \qquad \mathbf{R}_{i+r-1, j+c-1} \leftarrow \mathbf{B}_{i, j} \end{array} \right| \mathbf{R}$$

Глобальная матрица системы КТ:

$$KT := \sum_{n=1}^{ne} \text{Rep}(K0, K_n, n, n)$$

Теперь глобальная матрица КТ имеет вид

$$KT = \begin{pmatrix} 4 \times 10^8 & -4 \times 10^8 & 0 & 0 \\ -4 \times 10^8 & 8 \times 10^8 & -4 \times 10^8 & 0 \\ 0 & -4 \times 10^8 & 1.4 \times 10^9 & -1 \times 10^9 \\ 0 & 0 & -1 \times 10^9 & 1 \times 10^9 \end{pmatrix}$$

Необходимо задать граничные условия системы. Первые две ячейки первой строки относятся к 1-му узлу (т. е. заделке), в котором перемещения отсутствуют по условию закрепления. Чтобы матрица не стала вырожденной, первой ячейке вместо нуля зададим очень большое число:

$$KT_{1,1} := 10^{20} \quad KT_{1,2} := 0$$

Задаем вектор узловых внешних нагрузок. Для этого распределенную нагрузку заменяем ее равнодействующей, деля ее пополам для каждого узла соответствующего КЭ, и складываем с узловой нагрузкой:

$$\begin{aligned} F_1 &:= f_1 + \frac{q_1 \cdot L_1}{2} \\ F_2 &:= f_2 + \frac{q_1 \cdot L_1}{2} + \frac{q_2 \cdot L_2}{2} \\ F_3 &:= f_3 + \frac{q_2 \cdot L_2}{2} + \frac{q_3 \cdot L_3}{2} \\ F_4 &:= f_4 + \frac{q_3 \cdot L_3}{2} \end{aligned} \quad F = \begin{pmatrix} 1 \times 10^3 \\ 1 \times 10^3 \\ 0 \\ 1 \times 10^4 \end{pmatrix}$$

Иначе вектор узловых нагрузок можно задать функцией

$$F := \begin{cases} \text{for } i \in 2..nu - 1 \\ F_i \leftarrow f_i + \frac{q_{i-1} \cdot L_{i-1}}{2} + \frac{q_i \cdot L_i}{2} \\ F_1 \leftarrow f_1 + \frac{q_1 \cdot L_1}{2} \\ F_4 \leftarrow f_4 + \frac{q_3 \cdot L_3}{2} \\ F \end{cases}$$

Находим перемещения в узлах:

$$U := KT^{-1} \cdot F$$

$$U = \begin{pmatrix} 0 \\ 2.75 \times 10^{-5} \\ 5.25 \times 10^{-5} \\ 6.25 \times 10^{-5} \end{pmatrix}$$

Следует полученный результат сравнить с аналитическим расчетом.

Вычисление продольных усилий основано на условии равновесия КЭ. Создадим вектор перемещений КЭ:

$$UE_{el} := \begin{pmatrix} UE_{el} \\ UE_{el+1} \end{pmatrix}$$

Исключаем из узловых значений внутренних продольных усилий распределенную нагрузку, добавленную при расчете перемещений в вектор узловых сил:

$$N_{el} := K_{el} \cdot UE_{el} - \begin{pmatrix} \frac{q_{el} \cdot L_{el}}{2} \\ \frac{q_{el} \cdot L_{el}}{2} \end{pmatrix}$$

Учтем, что вектор положительной (с точки зрения вызываемой деформации) продольной силы в левом узле КЭ имеет направление, противоположное положительному направлению оси X системы:

$$N_{el} := \left[\overrightarrow{N_{el}} \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right]$$

Применяя известные зависимости сопротивления материалов, получим выражения для напряжения и относительной продольной деформации:

$$\sigma_{el} := \frac{N_{el}}{A_{el}} \quad \epsilon_{el} := \frac{\sigma_{el}}{E}$$

Численные результаты следующие:

$$N_1 = \begin{pmatrix} 1.2 \times 10^4 \\ 1 \times 10^4 \end{pmatrix} \quad N_2 = \begin{pmatrix} 1 \times 10^4 \\ 1 \times 10^4 \end{pmatrix} \quad N_3 = \begin{pmatrix} 1 \times 10^4 \\ 1 \times 10^4 \end{pmatrix}$$

$$\sigma_1 = \begin{pmatrix} 6 \times 10^7 \\ 5 \times 10^7 \end{pmatrix} \quad \sigma_2 = \begin{pmatrix} 2.5 \times 10^7 \\ 2.5 \times 10^7 \end{pmatrix} \quad \sigma_3 = \begin{pmatrix} 2 \times 10^7 \\ 2 \times 10^7 \end{pmatrix}$$

$$\varepsilon_1 = \begin{pmatrix} 3 \times 10^{-4} \\ 2.5 \times 10^{-4} \end{pmatrix} \quad \varepsilon_2 = \begin{pmatrix} 1.25 \times 10^{-4} \\ 1.25 \times 10^{-4} \end{pmatrix} \quad \varepsilon_3 = \begin{pmatrix} 1 \times 10^{-4} \\ 1 \times 10^{-4} \end{pmatrix}$$

Полученные результаты необходимо сравнить с аналитическим решением:

$$N_1(0) = 12\,000, \quad N_1(L_1) = 10\,000, \quad N_2(0) = 10\,000, \quad N_3(0) = 10\,000.$$

Определим максимальное по модулю напряжение. Максимальное напряжение в 1-м элементе (1-м узле) $\sigma_1 = 60$ МПа.

Условие прочности соблюдается, так как $\sigma_{\max} < R$.

Построим эпюры. Создадим вектор координат узлов системы LX при помощи функции-программы.

$$LX := \begin{cases} LX_1 \leftarrow 0 \\ \text{for } i \in 1..ne \\ \quad LX_{i+1} \leftarrow LX_i + L_i \\ LX \end{cases} \quad LX = \begin{pmatrix} 0 \\ 0.1 \\ 0.3 \\ 0.4 \end{pmatrix}$$

Задаем в виде ранжированной переменной длину каждого КЭ (шаг 0.01 позволит построить заштрихованную эпюру):

$$X(el) := LX_{el}, LX_{el} + 0.01 .. LX_{el+1}$$

$$X1 := X(1) \quad X2 := X(2) \quad X3 := X(3)$$

Для правого предела аргумента графиков функции зададим параметр

$$LT := \sum_{el=1}^{ne} L_{el}$$

Запишем функции в общем виде. Учитывая то, что все величины определялись как матричные, введем нижний индекс 1.

Продольные усилия FN и напряжения Fσ:

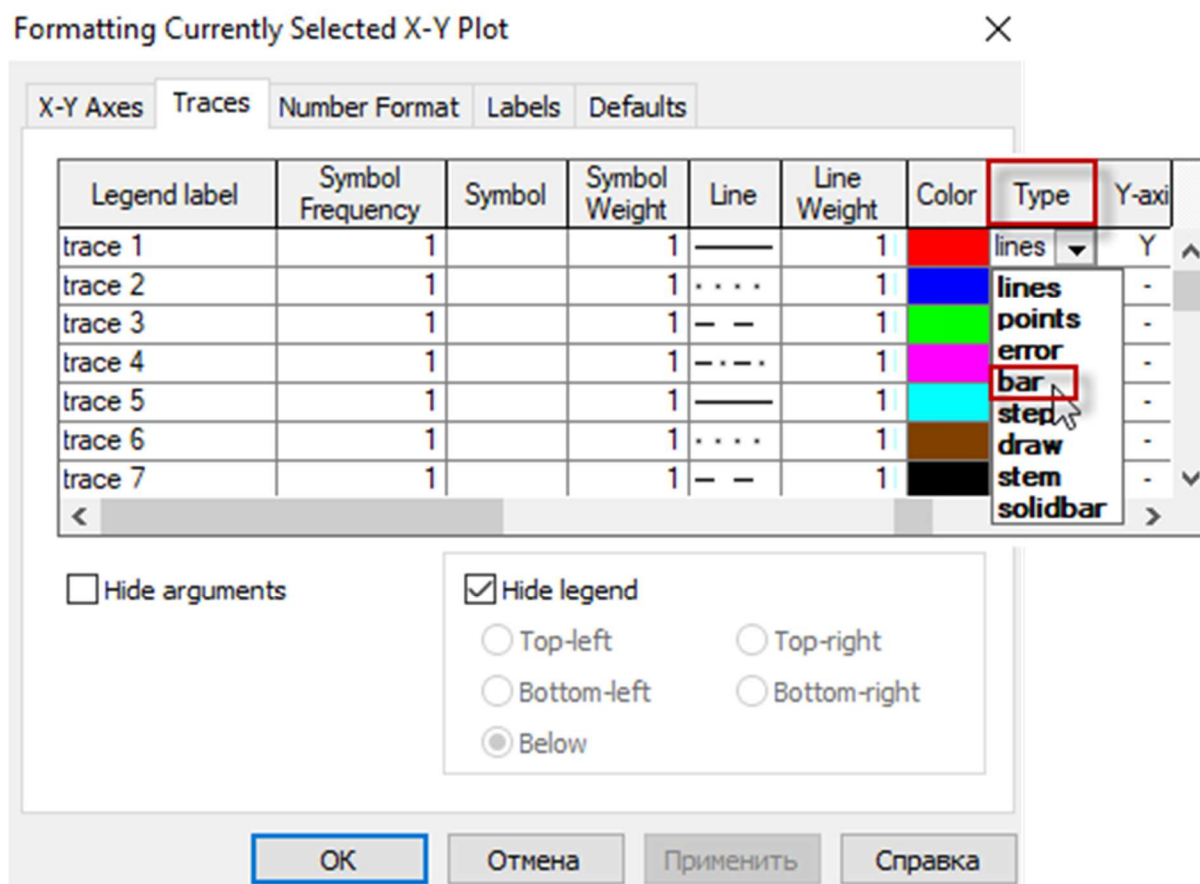
$$FN(x, el) := (N_{el})_1 - q_{el} \cdot (x - LX_{el})$$

$$F\sigma(x, el) := \frac{FN(x, el)}{A_{el}}$$

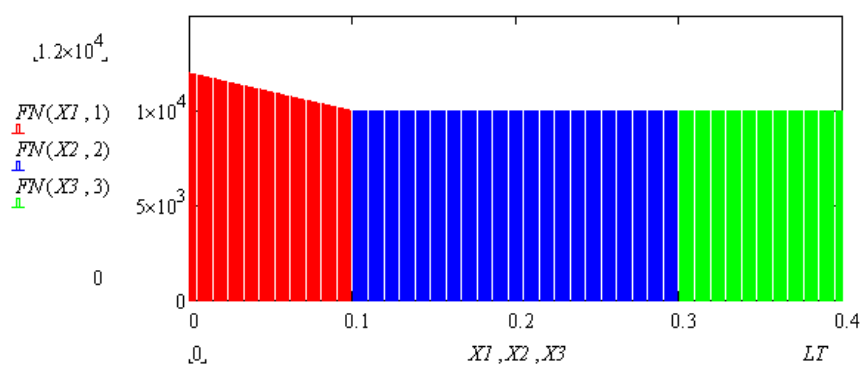
Относительные продольные деформации Fε и абсолютные перемещения FU:

$$F\epsilon(x, el) := \frac{F\sigma(x, el)}{E} \quad FU(x, el) := \int_{LX_{el}}^x F\epsilon(x, el) dx + U_{el}$$

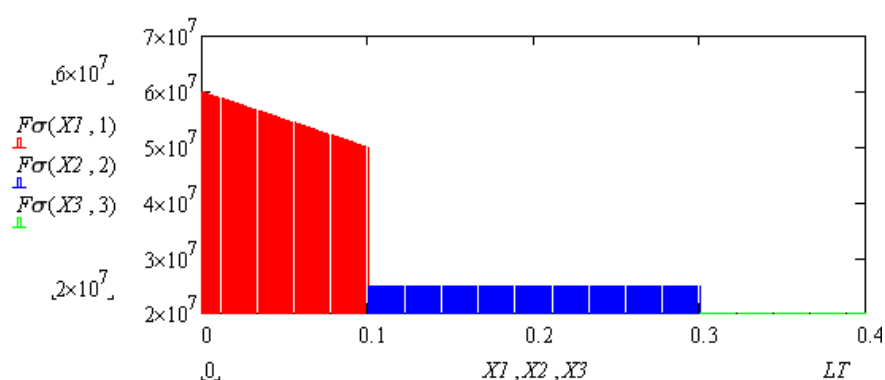
Для того чтобы получилась заштрихованная область на графике, в диалоговом окне *Format* (открывается двойным щелчком по графику или выбирается из контекстного меню графика), во вкладке *Traces* следует заменить линейный тип графика (*lines*) типом *bar*.



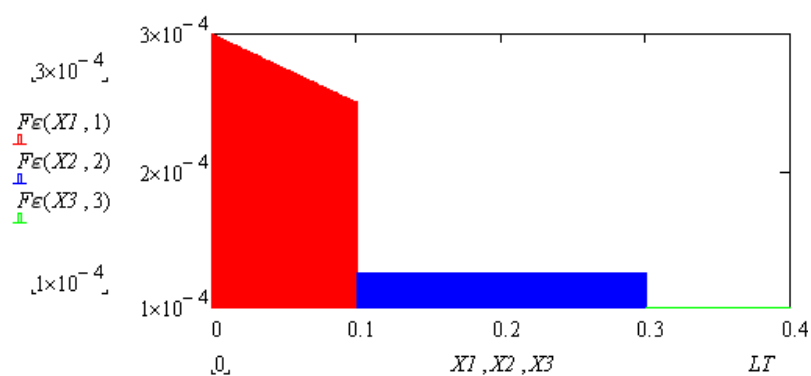
Эпюры продольных сил, Н, представлены ниже.



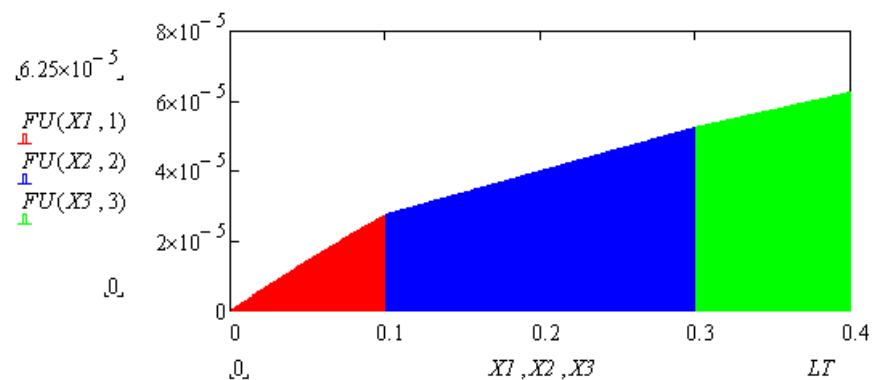
Эпюры напряжения, Па, представлены ниже.



Эпюры относительных деформаций представлены ниже.



Эпюры абсолютных перемещений, м, представлены ниже.



ПРИМЕРЫ РЕШЕНИЙ С ПОМОЩЬЮ PTC MATHCAD

Самостоятельная работа «Исследование функции»

Дана функция

$$f(x) := \sqrt[3]{6x^2 - x - 5}$$

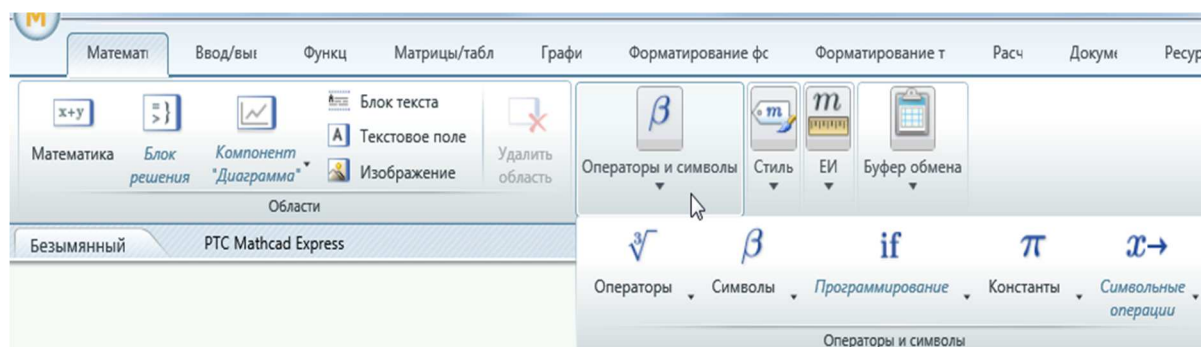
Для записи выражения функции вводится знак присвоения «:=» путем последовательного нажатия клавиш Shift и «:=» (раскладка клавиатуры английская).

Знак радикала вводится клавишей «\».

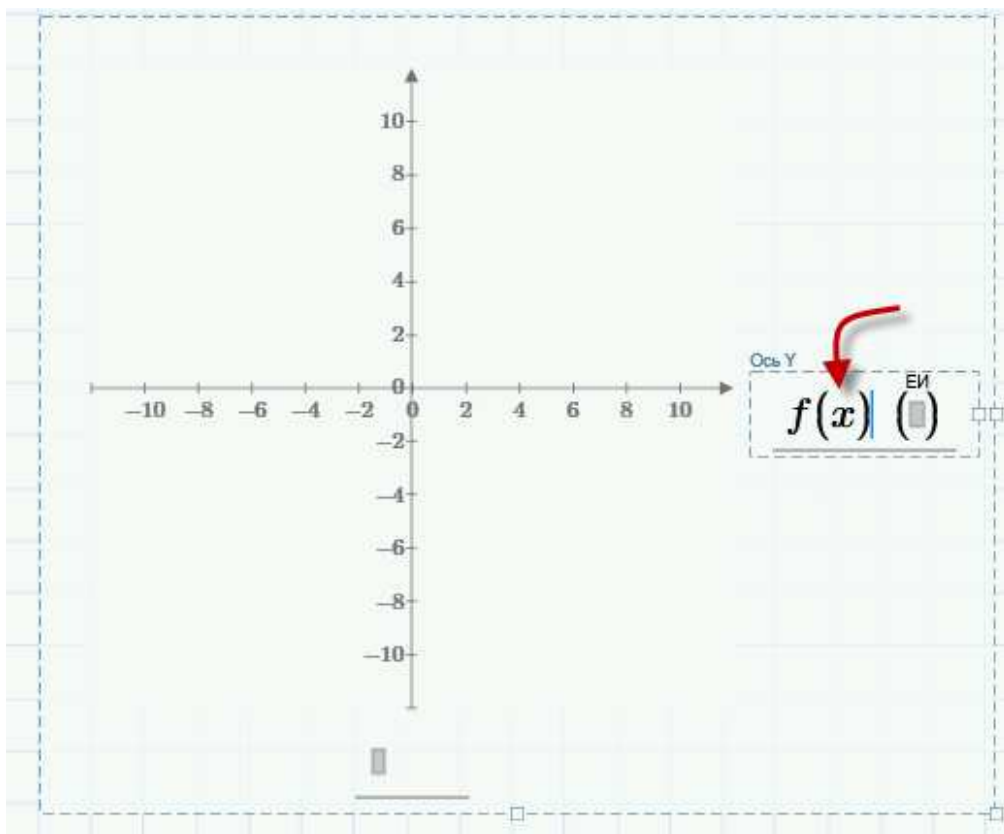
Возведение в степень осуществляется комбинацией клавиш Shift – «^».

Перемещение по выражению происходит с помощью стрелок клавиатуры, для выделения выражения или его части нажимают на клавишу «Пробел».

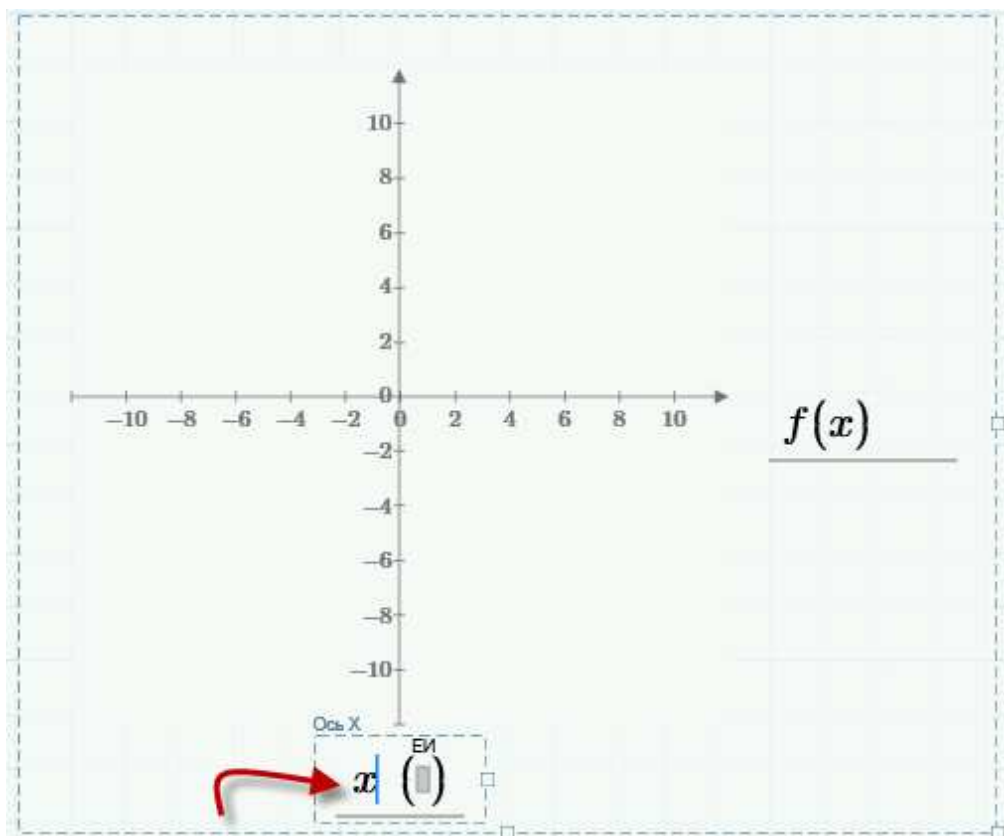
Другой вариант записи символов и выражений – лента *Математика – Операторы и символы*.



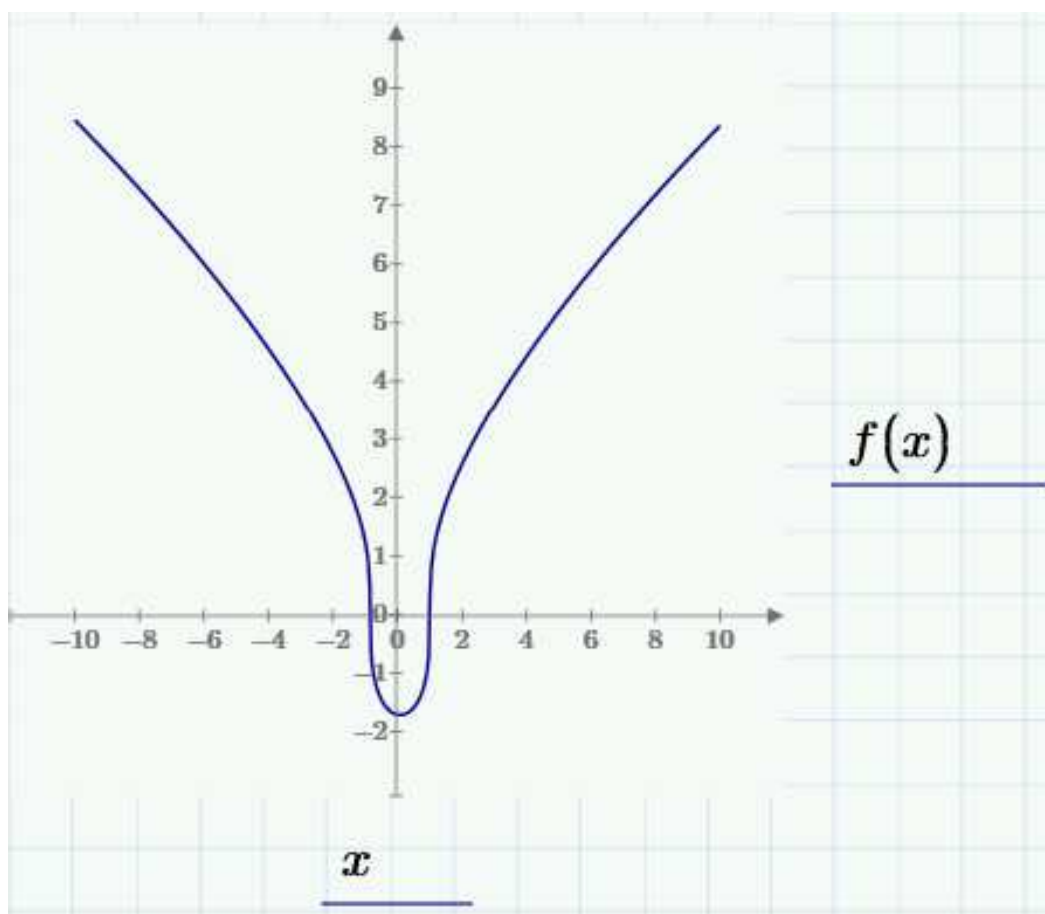
Для вставки графика воспользуйтесь лентой *График – Вставить график – График XY*. В появившемся поле оси Y введите имя функции f(x).



В области аргумента запишите имя переменной x .



Получим график функции заданного выражения.



Функция определена на всей числовой оси X и разрывов не имеет. Её область определения $-\infty < x < \infty$.

Для нахождения корней функции запишем отдельно выражение функции и приравняем это выражение к нулю.

Для вставки оператора «равенство» последовательно нажимают клавиши Ctrl – «=».

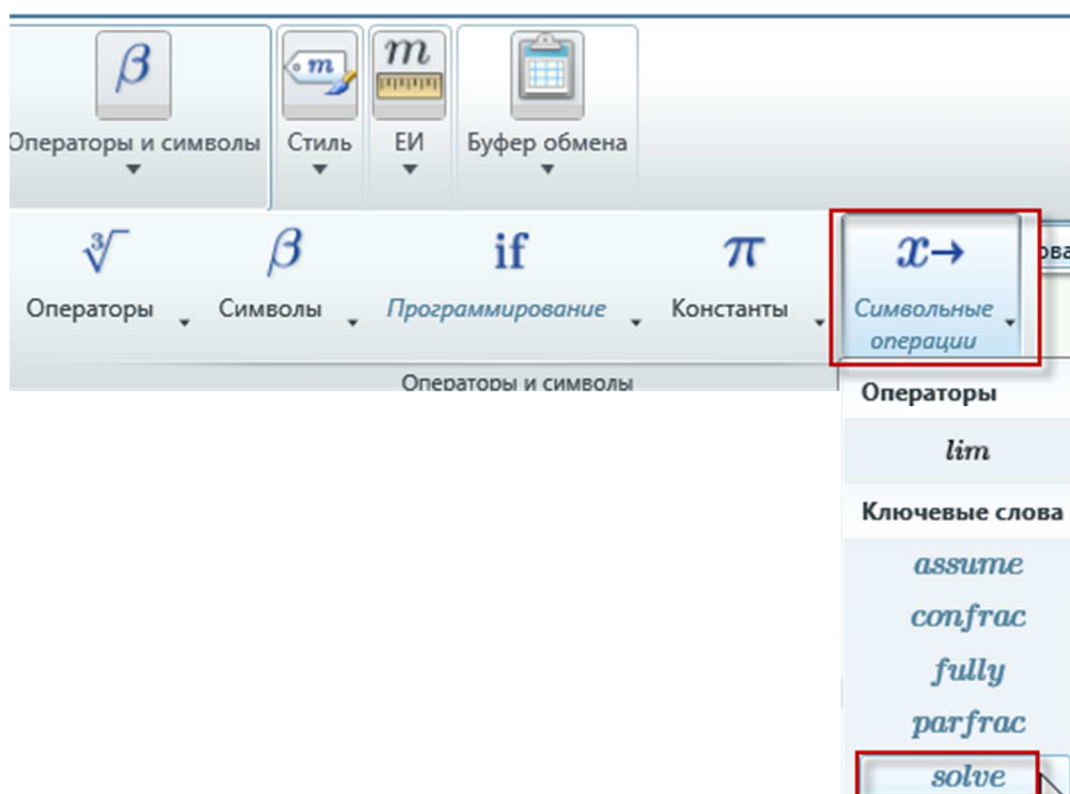
$$f(x) = 0$$

Поскольку в данном примере подрадикальное выражение обращает в нуль всю функцию, достаточно его приравнять к нулю.

Ошибкой не будет и такая запись:

$$6x^2 - x - 5 = 0$$

Поиск корней функции осуществим с помощью функции *solve*.



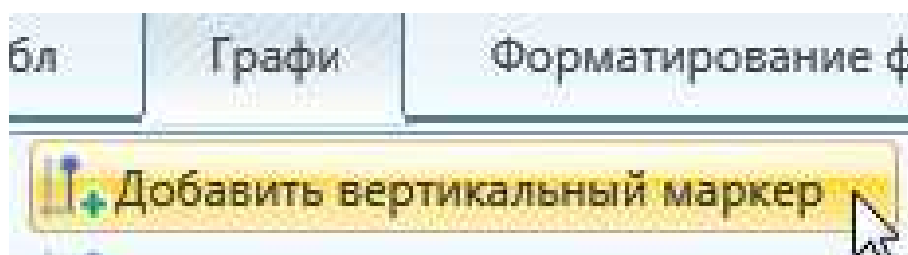
После вставки функции *solve* поставьте запятую и запишите аргумент x (это действие необязательно, если в уравнении только одна неизвестная, как в данном примере):

$$f(x) = 0 \xrightarrow{\text{solve}, x} 1.0$$

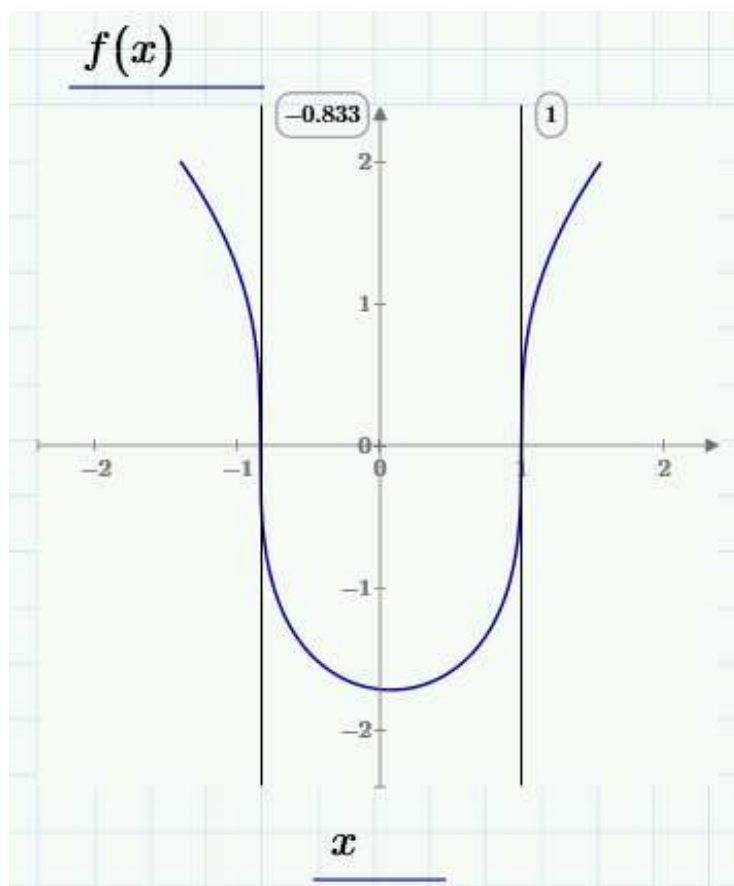
$$6x^2 - x - 5 = 0 \xrightarrow{\text{solve}, x} \begin{bmatrix} 1 \\ 5 \\ -6 \end{bmatrix}$$

Обратите внимание, что аналитическое решение для подрадикального выражения дает все корни.

Скопируйте график и дважды вставьте вертикальный маркер.

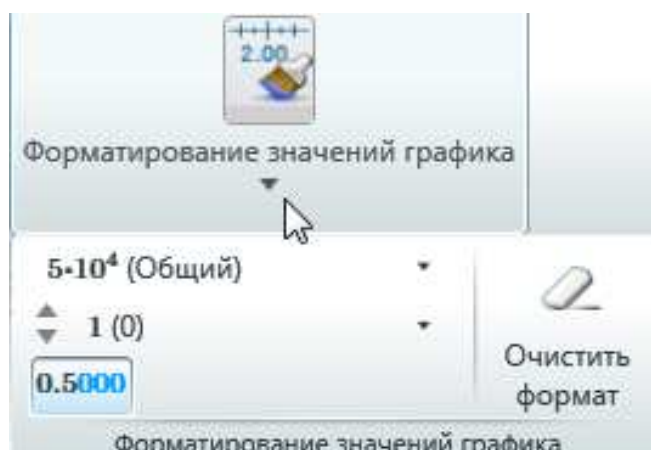


Отметьте на графике найденные значения. Отредактируйте значение аргумента согласно найденным корням.

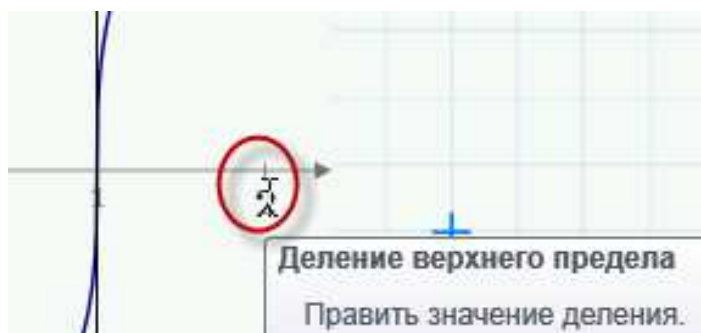


Очевидно, что предельные значения для функции и аргумента в графике можно уменьшить до $[-2, 2]$.

Форматирование графика выполняется в разделе *Форматирование значений графика* ленты *Графики* при условии, что область графика выбрана.



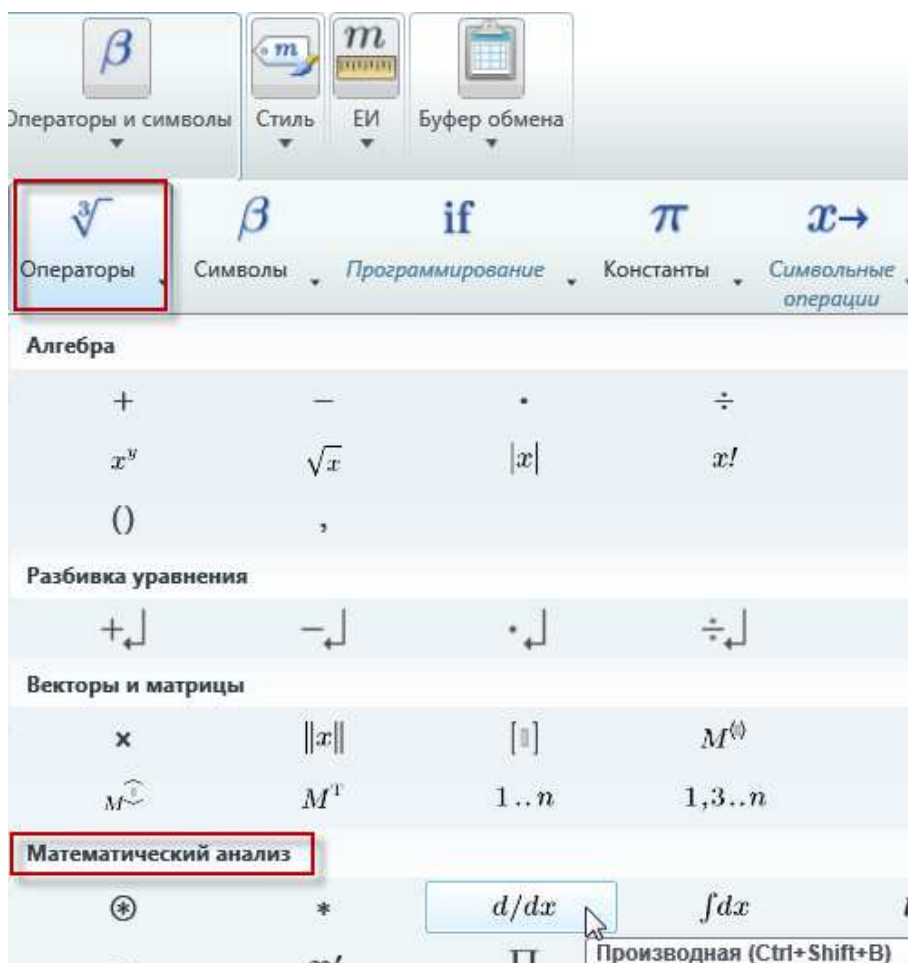
Изменение пределов и деление числовой оси осуществляются непосредственным редактированием её значений.



Исследуем функции на экстремальные значения. Зададим производную 1-го порядка:

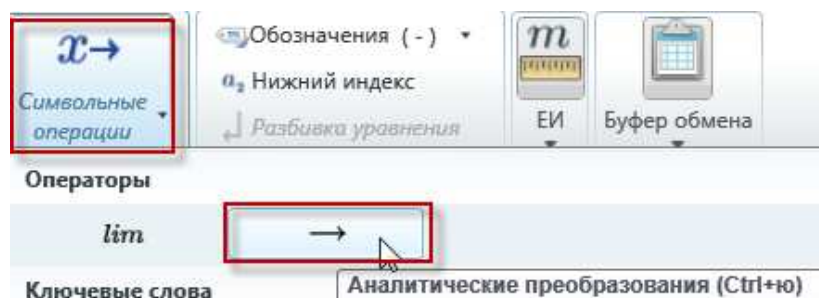
$$f1(x) := \frac{d}{dx} f(x)$$

Знак дифференциала находится в подразделе *Операторы – Математический анализ*.



$$f1(x) \rightarrow -\frac{\sqrt[3]{6 \cdot x^2 - x - 5 \cdot (12 \cdot x - 1)}}{3 \cdot (x - 6 \cdot x^2 + 5)}$$

Чтобы получить выражение производной, воспользуйтесь оператором аналитического преобразования « $\rightarrow$ », который расположен в подразделе *Символьные операции – Операторы* ленты *Математика*. Иначе оператор можно вызвать комбинацией клавиш Ctrl – «.».

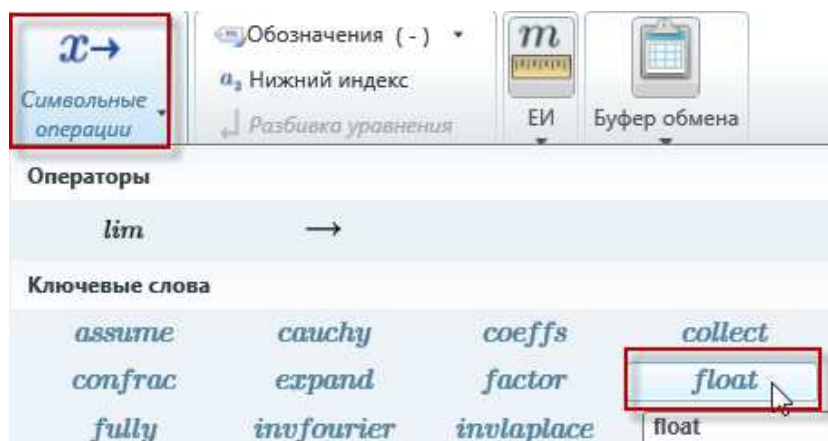


Критические точки первой производной определяются решением уравнения.

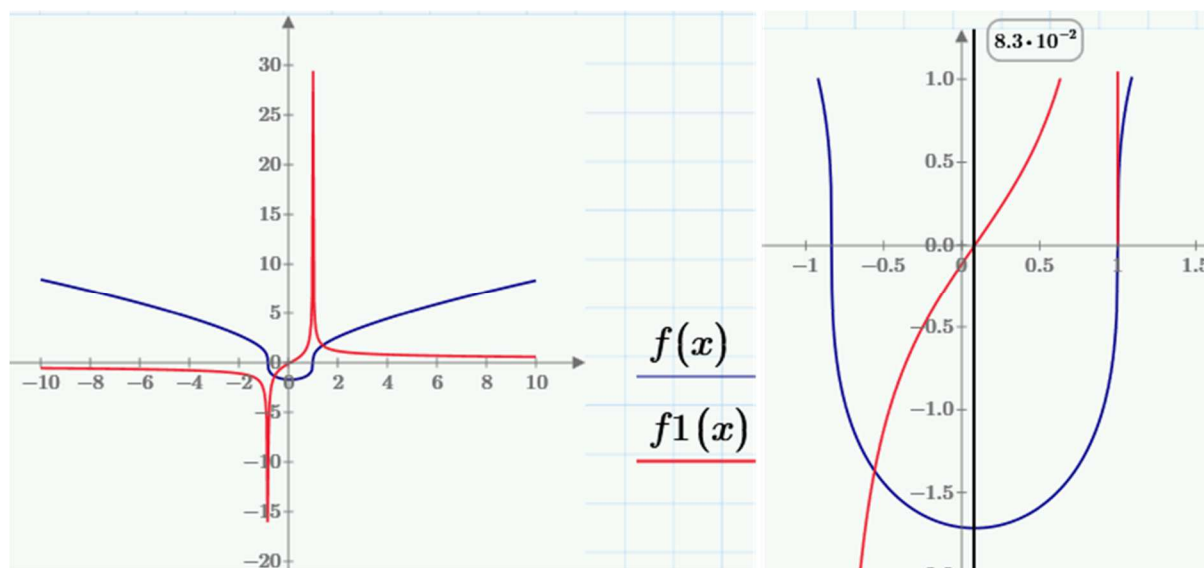
$$f1(x) = 0 \xrightarrow{\text{solve}, x} \begin{bmatrix} 1.0 \\ 1 \\ 12 \end{bmatrix}$$

$$-\frac{\sqrt[3]{6 \cdot x^2 - x - 5 \cdot (12 \cdot x - 1)}}{3 \cdot (x - 6 \cdot x^2 + 5)} = 0 \xrightarrow{\text{solve}, x, \text{float}, 3} \begin{bmatrix} 1.0 \\ 0.0833 \end{bmatrix}$$

Для вывода результата в десятичном формате воспользуйтесь функцией *float*, задав необходимую точность (например, точность до 0.001).



Постройте новый график функции и добавьте второй график первой производной. Для этого воспользуйтесь командой *Добавить кривую* (лента *Графики*).



Исследуем найденные значения, отметив на графике критические точки первой производной. Очевидно, что $x_1 = 1$ – точка разрыва производной. Точка $x_2 = 0.083$ – корень производной. При переходе через точку x_2 производная меняет знак с «–» на «+».

$$f1(0) = -0.114$$

$$f1(0.1) = 0.023$$

Это означает, что точка x_2 – минимум для функции.

Таким образом, функция имеет два участка возрастания/убывания:

- на интервале $-\infty < x < 0.0833$ первая производная меньше нуля, функция убывает;
- на интервале $0.0833 < x < \infty$ первая производная больше нуля, функция возрастает.

Исследуем функции на выпуклость/вогнутость. Зададим производную 2-го порядка:

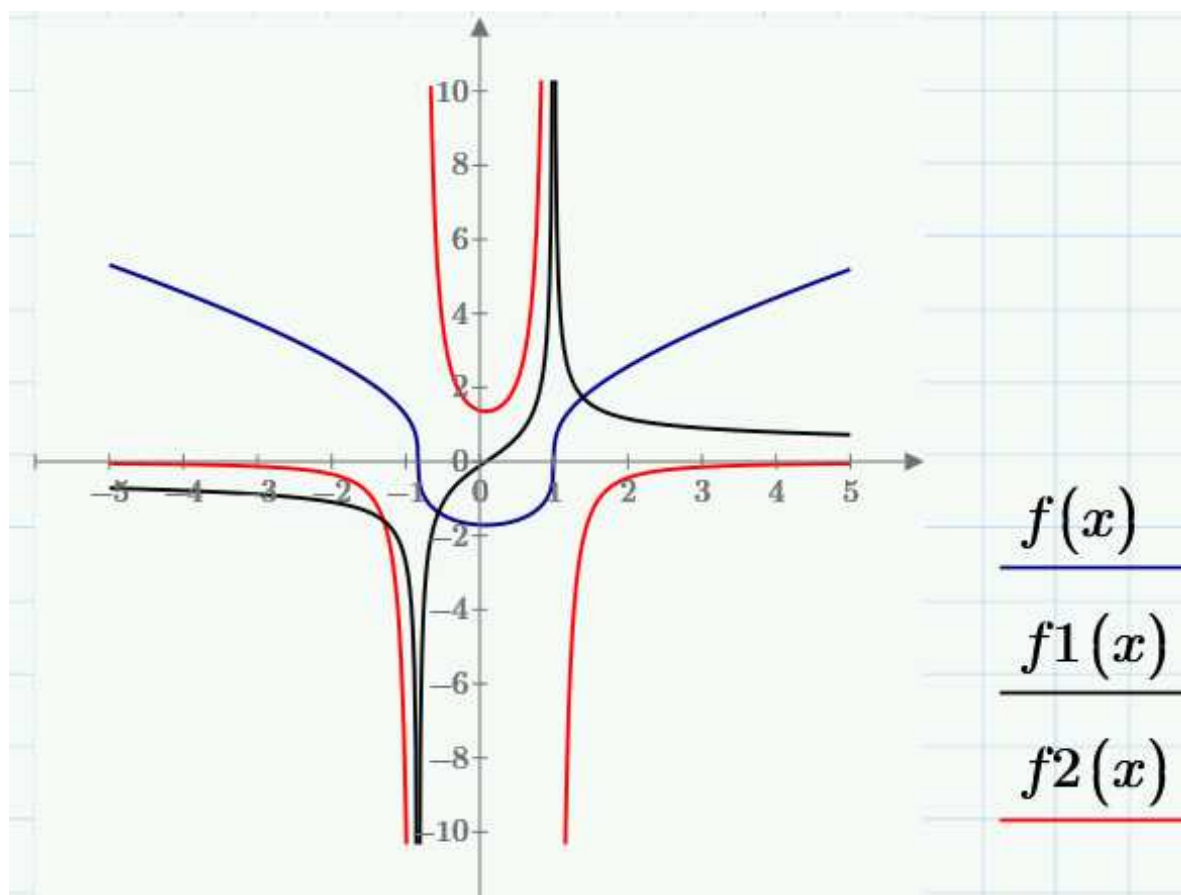
$$f2(x) := \frac{d^2}{dx^2} f(x)$$

Выражение можно упростить с помощью функции *simplify* (находится там же, где используемые ранее функции *solve* и *float*).

$$f2(x) \rightarrow -\frac{4 \cdot \sqrt[3]{6 \cdot x^2 - x - 5}}{x - 6 \cdot x^2 + 5} - \frac{2 \cdot \sqrt[3]{6 \cdot x^2 - x - 5} \cdot (12 \cdot x - 1)^2}{9 \cdot (x - 6 \cdot x^2 + 5)^2}$$

$$f2(x) \xrightarrow{\text{simplify}} -\frac{2 \cdot \sqrt[3]{6 \cdot x^2 - x - 5} \cdot (36 \cdot x^2 - 6 \cdot x + 91)}{9 \cdot (x - 6 \cdot x^2 + 5)^2}$$

Построим график второй производной.



Найдем её критические точки.

$$f_2(x) = 0 \xrightarrow{\text{solve}, x} \begin{bmatrix} 1.0 \\ \frac{1}{12} + \frac{11i \cdot \sqrt{3}}{12} \\ \frac{1}{12} - \frac{11i \cdot \sqrt{3}}{12} \end{bmatrix}$$

$$9 \cdot (x - 6 \cdot x^2 + 5)^2 = 0 \xrightarrow{\text{solve}, x, \text{float}, 3} \begin{bmatrix} 1.0 \\ 1.0 \\ -0.833 \\ -0.833 \end{bmatrix}$$

Вторая производная не имеет корней, однако не определена (имеет разрыв) в точках $x_1 = -0.833$, $x_2 = 1.0$.

Исследуем x_1 и x_2 на возможные точки перегиба:

$$f_2(-1) = -9.309$$

$$f_2(0) = 1.383$$

$$f_2(2) = -0.441$$

Переходя через свои критические точки, вторая производная меняет знак. Таким образом, точки $x_1 = -0.833$ и $x_2 = 1$ – точки перегиба для функции. Согласно знакам второй производной функция выпукла при $x < -0.833$ и при $x > 1$; функция вогнута при $-0.833 < x < 1$.

Задание 1. Решение нелинейного уравнения

Дано нелинейное уравнение трансцендентного типа

$$e^x = x^3$$

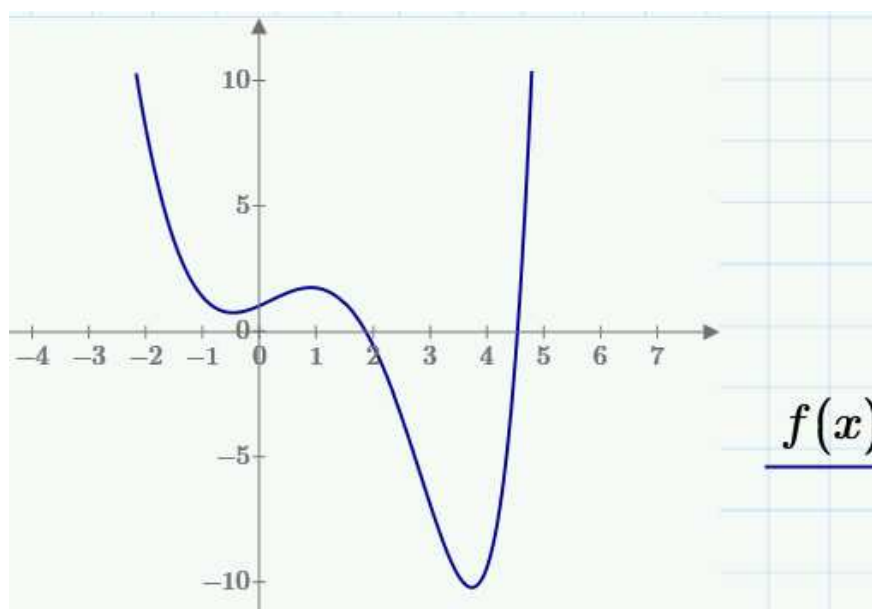
Присвоим данное выражение функции

$$f(x) := e^x - x^3$$

Построим её график.



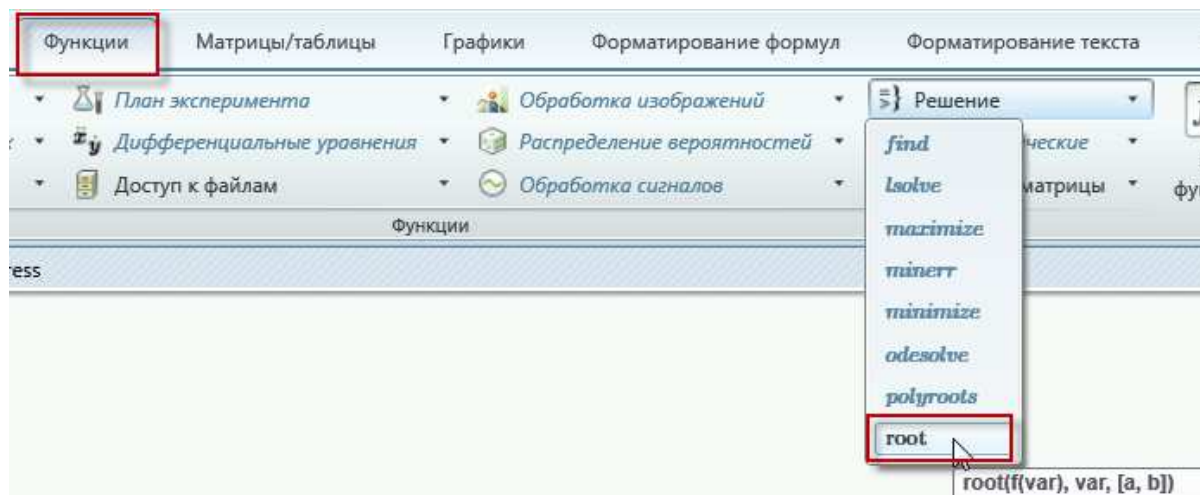
График в таком виде дает очень мало информации, поскольку область функции, в пределах которой выражение обращается в нуль, имеет более узкий интервал. Отредактируем ось ординат, уменьшив её пределы.



На отредактированном графике хорошо видно, что функция имеет два корня в интервалах аргумента x $[0, 3]$ и $[3, 6]$.

Решение 1. Функция *root* позволяет найти только один корень на заданном интервале. Запишем функцию дважды, меняя числовые значения границ интервала поиска корня.

Функция *root* находится в ленте *Функции*, в разделе *Решение*.

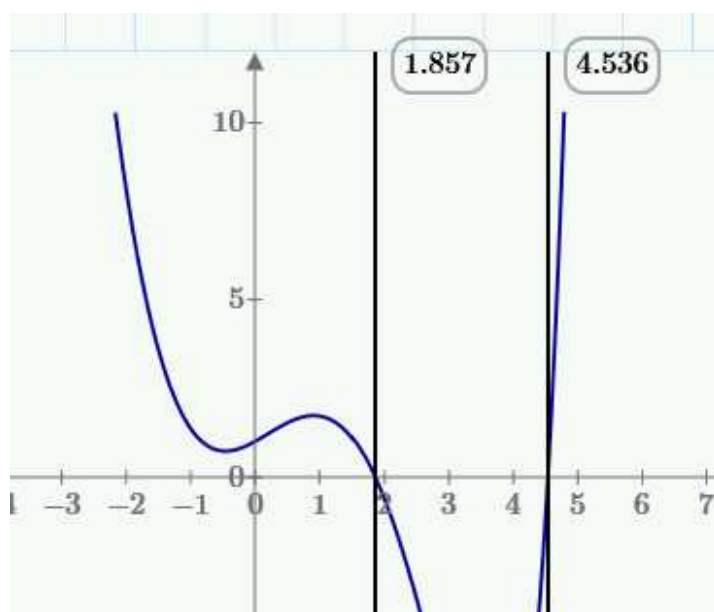


После записи параметров функции *root* вводим обычное «равно»:

$$\text{root}(f(x), x, 0, 3) = 1.857$$

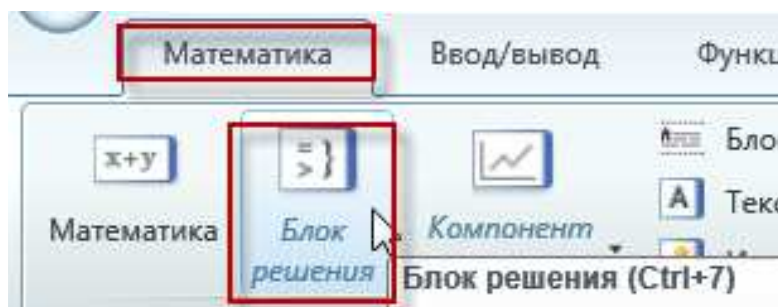
$$\text{root}(f(x), x, 3, 6) = 4.536$$

Отметим найденные корни на графике.

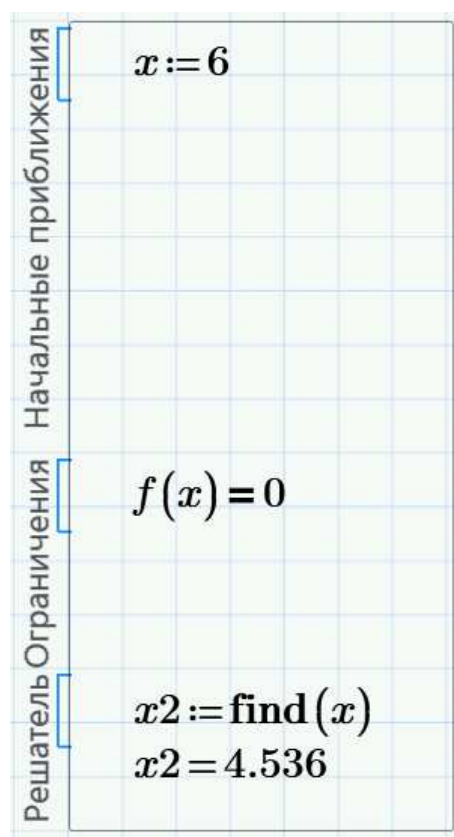
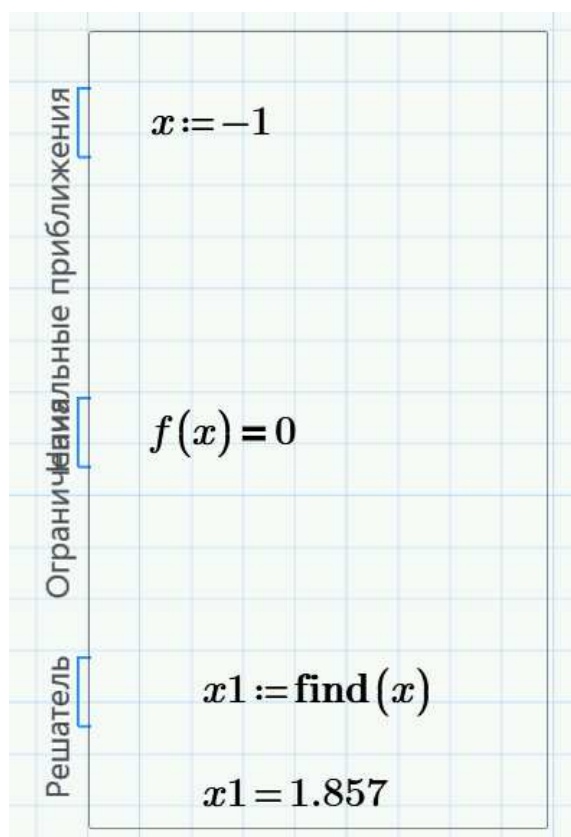


Решение 2. Используем вычислительный блок с функцией *Find*. Для нахождения вычислительным блоком двух корней x_1 и x_2 необходимо задать два начальных приближения для каждого вычислительного блока в отдельности.

Вставка вычислительного блока осуществляется из ленты *Математика – Блок решения* или комбинацией клавиш $\text{Ctrl} - \langle 7 \rangle$.



Задайте внутри первого блока начальное приближение для первого корня (например, -1). Запишите уравнение (или приравняйте функцию к нулю). Присвойте первому искомому корню функцию *Find* (лента *Функции – Решение*) и найдите первый корень x_1 . Повторите действия для нахождения второго корня x_2 .

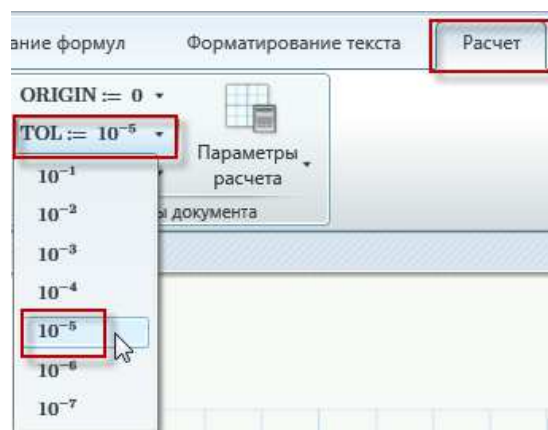
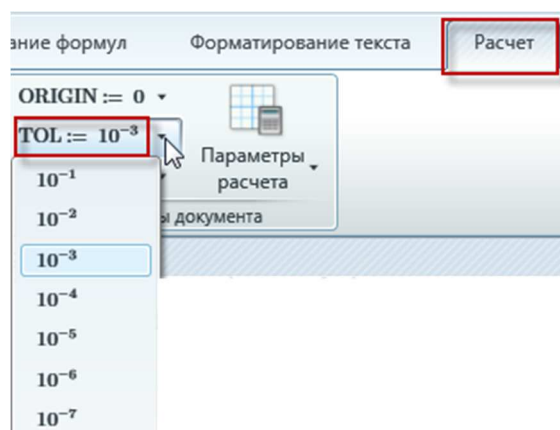


Решение 3 (программирование). Предварительно необходимо составить функцию-программу для нахождения корней.

Все операторы программирования находятся в разделе *Программирование (if)* ленты *Математика*.



Составим программный блок, используя метод Ньютона. Для назначения точности решения понадобится постоянная TOL (допуск сходимости), которая по умолчанию равна 10^{-3} (назначается в ленте *Расчет*), но ей всегда можно задать другое значение точности. Назначьте постоянной значение 10^{-5} .



Составьте с помощью операторов программирования функцию Ньютона (обратите внимание на вертикальные полосы, которые отделяют блок цикла *while* от общего тела программы).

$$NT(f, x) := \left\| \begin{array}{l} x1 \leftarrow x + 2 \cdot TOL \\ \text{while } |x1 - x| > TOL \\ \quad \left\| \begin{array}{l} x \leftarrow x1 \\ x1 \leftarrow x - \frac{f(x)}{\frac{d}{dx} f(x)} \end{array} \right\| \end{array} \right\|$$

Для нахождения корней уравнения запишите ниже программного блока функцию Ньютона с разными значениями начального приближения аргумента.

$$NT(f, 5) = 4.536 \qquad NT(f, 2) = 1.857$$

Если график функции не касается оси абсцисс, а пересекает её (функция, переходя через искомую точку, меняет знак), применяем метод половинного деления. Для этого составляем ещё одну программу.

$$D(f, a, b) := \left\| \begin{array}{l} \text{while } \left\| \frac{a-b}{2} \right\| > TOL \\ \quad \left\| \begin{array}{l} x1 \leftarrow \frac{a+b}{2} \\ \text{if } f(x1) \cdot f(a) < 0 \\ \quad \left\| b \leftarrow x1 \right\| \\ \text{else if } f(x1) \cdot f(a) \geq 0 \\ \quad \left\| a \leftarrow x1 \right\| \end{array} \right\| \\ x \leftarrow x1 \end{array} \right\|$$

Находим корни уравнения:

$$D(f, 0, 2) = 1.857 \qquad D(f, 4, 5) = 4.536$$

Таким образом, все методы дали один и тот же результат.

Решение 4. Функция *polyroots* применяется только для нелинейного алгебраического уравнения с целью нахождения корней полиномов вида $a_0 \cdot x^0 + a_1 \cdot x^1 + \dots + a_n x^n = 0$.

Для нахождения корней функцией $\text{polyroots}(v)$ необходимо предварительно записать вектор коэффициентов уравнения v с помощью функции coeffs . Функция coeffs расположена в разделе *Операторы и символы* ленты *Математика*, в подразделе *Символьные операции*. Функция $\text{polyroots}(v)$ расположена в ленте *Функции – Решение*.

Рассмотрим решение на примере уравнения

$$2x - (2x)^5 + 3x^3 - 7x^2 = 0$$

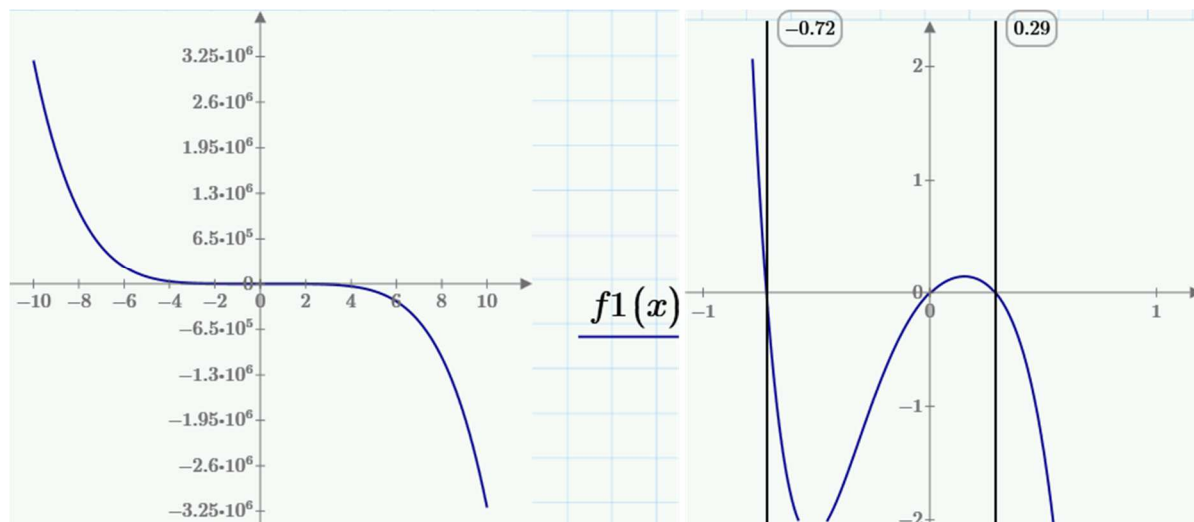
Для нахождения коэффициентов уравнения необходимо записать только его левую часть и вставить функцию coeffs .

$$2x - (2x)^5 + 3x^3 - 7x^2 \xrightarrow{\text{coeffs}} \begin{bmatrix} 0 \\ 2 \\ -7 \\ 3 \\ 0 \\ -32 \end{bmatrix}$$

Присвойте полученный вектор переменной и найдите функцией polyroots все корни уравнения (в том числе и иррациональные).

$$v := \begin{bmatrix} 0 \\ 2 \\ -7 \\ 3 \\ 0 \\ -32 \end{bmatrix} \quad \text{polyroots}(v) = \begin{bmatrix} -0.72 \\ 0 \\ 0.215 - 0.504i \\ 0.215 + 0.504i \\ 0.29 \end{bmatrix}$$

Проверьте результат на графике.



Задание 2. Решение системы линейных уравнений

Дана система линейных уравнений:

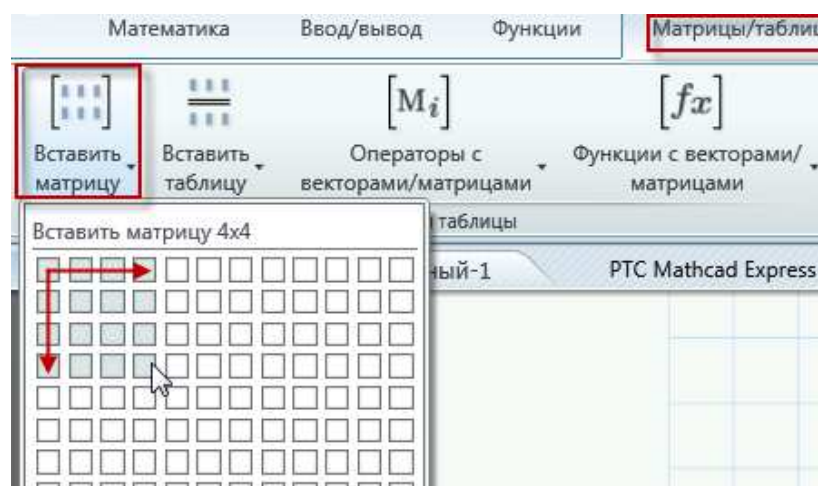
$$2x_1 + 0.5x_2 - 2.7x_3 + 9.2x_4 = 6$$

$$x_1 - 1.6x_2 + 0.6x_3 - 0.7x_4 = 2$$

$$3x_1 + 1.5x_2 + 8.5x_3 + 4.2x_4 = -3$$

$$-5.2x_1 + 6x_2 + 3.4x_3 - 5x_4 = 2.5$$

Решение 1 (матричный метод). Сформируем матрицу коэффициентов A и матрицу правых частей B . Для этого необходимо вставить матрицу. Вставка шаблона матрицы осуществляется из ленты *Матрицы/таблицы* – *Вставить матрицу*.



Для матрицы A задаем размеры 4×4 , для матрицы-вектора B – размеры 4×1 . После присвоения переменным A и B шаблонов матриц заполните их ячейки вручную.

$$A := \begin{bmatrix} 2 & 0.5 & -2.7 & 9.2 \\ 1 & -1.6 & 0.6 & -0.7 \\ 3 & 1.5 & 8.5 & 4.2 \\ -5.2 & 6 & 3.4 & -5 \end{bmatrix}$$

$$B := \begin{bmatrix} \\ \\ \\ \end{bmatrix} \quad B := \begin{bmatrix} 6 \\ 2 \\ -3 \\ 2.5 \end{bmatrix}$$

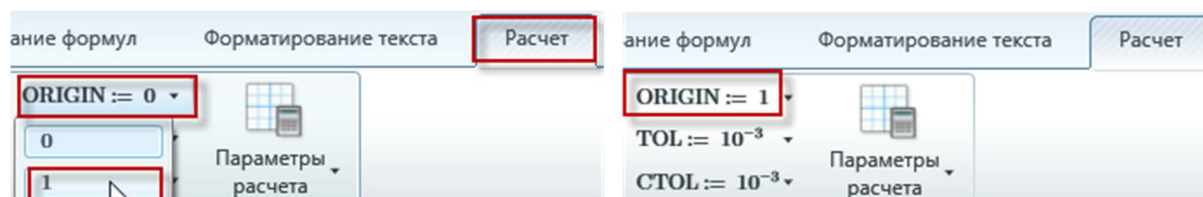
Проверим определитель матрицы. Он должен быть отличен от нуля:

$$|A| = -78.723$$

Символ определителя находится в подразделе *Операторы с векторами/матрицами* ленты *Матрицы/таблицы* или в ленте *Математика – Операторы*.

Отличный от нуля определитель матрицы означает, что система уравнений имеет единственное решение.

Задаем системную переменную $ORIGIN:=1$ (лента *Расчет*).



Матричное уравнение для решения системы линейных уравнений имеет вид

$$x := A^{-1} \cdot B \quad x = \begin{bmatrix} -31.075 \\ -22.227 \\ 8.978 \\ 11.25 \end{bmatrix}$$

Вычисление функцией *lsolve* также основано на матричном уравнении, но матрицы коэффициентов и свободных членов записываются в качестве параметров *lsolve*. Функция *lsolve* расположена в списке *Функции с векторами/матрицами* ленты *Матрицы/таблицы*.

$$x1 := \text{lsolve}(A, B) \quad x1 = \begin{bmatrix} -31.075 \\ -22.227 \\ 8.978 \\ 11.25 \end{bmatrix}$$

Решение 2 (метод Крамера). Для решения системы линейных уравнений по методу Крамера составим функцию с применением элементов программирования.

$$Kr(A, B) := \begin{array}{l} \text{if cols}(A) = \text{rows}(A) \\ \quad \text{for } i \in 1 \dots \text{cols}(A) \\ \quad \quad C \leftarrow A \\ \quad \quad \text{if } |A| \neq 0 \\ \quad \quad \quad x_i \leftarrow \frac{|C|}{|A|} \\ \quad \quad \quad i \leftarrow i + 1 \\ \quad \quad \quad x \end{array}$$

$$Kr(A, B) = \begin{bmatrix} -31.075 \\ -22.227 \\ 8.978 \\ 11.25 \end{bmatrix}$$

Функции *cols*, *rows* расположены в списке *Функции с векторами/матрицами* ленты *Матрицы/таблицы*.

Решение 3. Проверка решения вычислительным блоком. Вставьте блок решения (лента *Математика*).



Задайте в области начальных приближений блока любые начальные значения искомым переменным x_1, x_2, x_3, x_4 (например, 0). Вставьте в блок систему уравнений и запустите на счет функцию *Find* (расположена в разделе *Решение* ленты *Функции*).

Ограничения	$x_1 := 0$	$x_2 := 0$	$x_3 := 0$	$x_4 := 0$
Решатель	$2 x_1 + 0.5 x_2 - 2.7 x_3 + 9.2 x_4 = 6$			
	$x_1 - 1.6 x_2 + 0.6 x_3 - 0.7 x_4 = 2$			
	$3 x_1 + 1.5 x_2 + 8.5 x_3 + 4.2 x_4 = -3$			
	$-5.2 x_1 + 6 x_2 + 3.4 x_3 - 5 x_4 = 2.5$			
	$\text{find}(x_1, x_2, x_3, x_4) = \begin{bmatrix} -31.075 \\ -22.227 \\ 8.978 \\ 11.25 \end{bmatrix}$			

Сравним полученные результаты решения системы матричными методами и методом вычислительного блока – они идентичны.

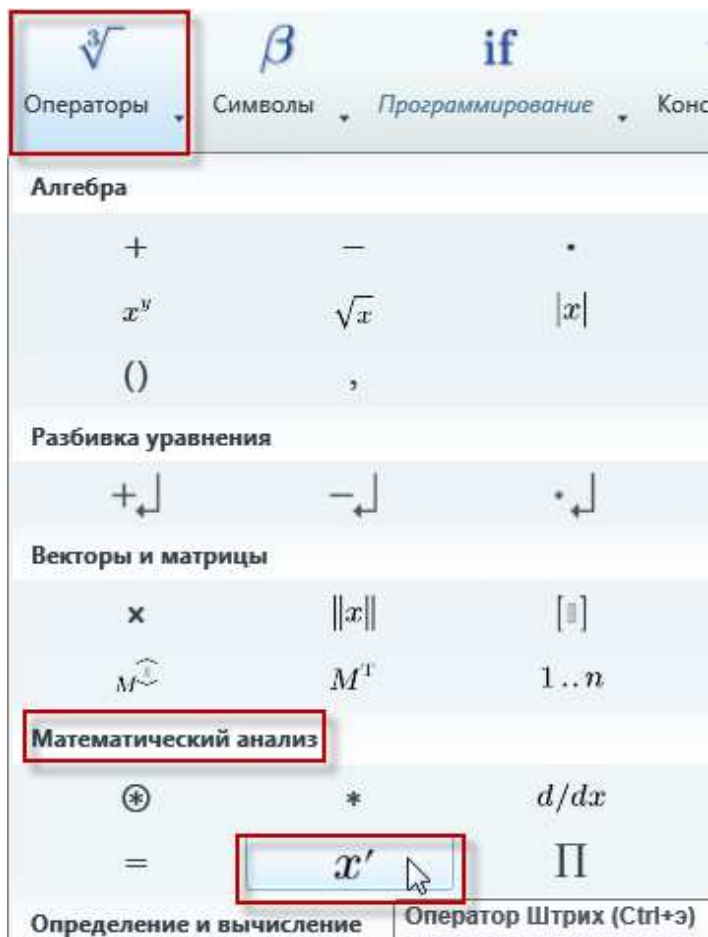
Задание 3. Решение обыкновенных дифференциальных уравнений 2-го порядка

Дано уравнение

$$y''(x) + a_1 \cdot y'(x) + a_2 \cdot y(x) = 0$$

$a_1 := 2 \quad a_2 := 2$

Для ввода символа производной используйте комбинацию клавиш **Ctrl – «'»** или подраздел *Математический анализ* раздела *Операторы* ленты *Математика*. Для второй производной повторите ввод.



Начальные условия следующие:

$$\begin{aligned} x_0 &= 1 & y_0(x_0) &= 6 \\ y_0'(x_0) &= 2 \end{aligned}$$

Интервал интегрирования (x_0, x_k) , где

$$x_0 := 1 \quad x_k := 6$$

Количество точек интегрирования

$$N := 100$$

Шаг интегрирования

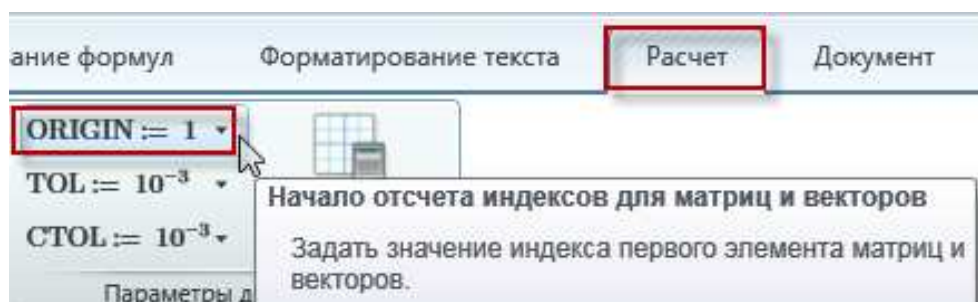
$$h := \frac{x_k - x_0}{N} \quad h = 0.05$$

Значения аргумента x задаем ранжированной переменной

$x := x0, x0 + h \dots xk$

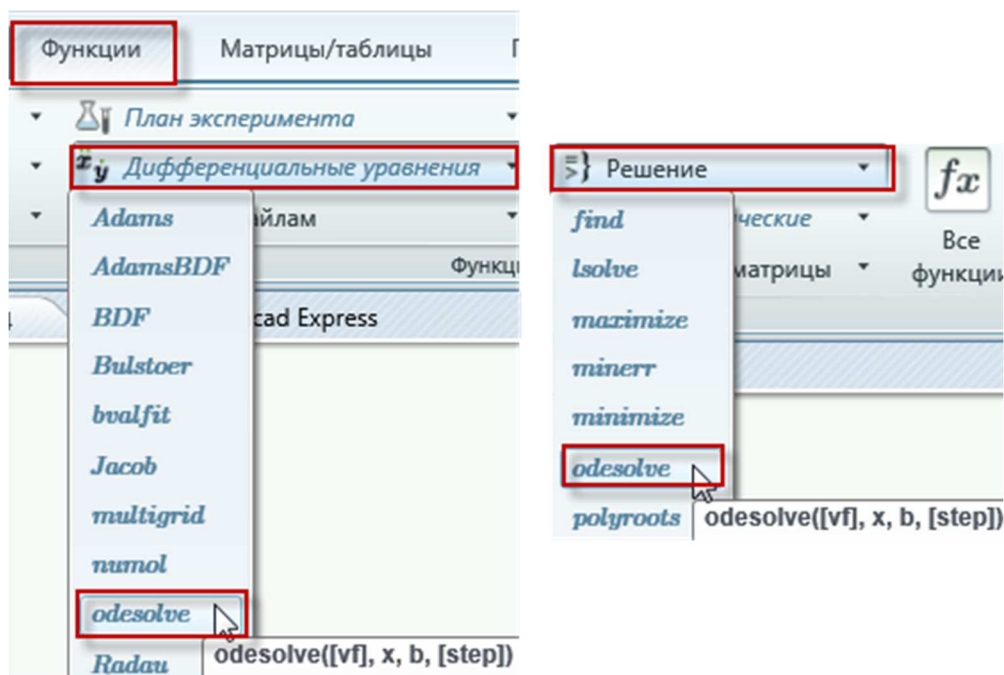
Формат ранжированной переменной с шагом «от..до» появится автоматически после ввода первого значения с последующей запятой. Если шаг не указывается, значит, он равен единице (после начального значения достаточно ввести подряд две точки).

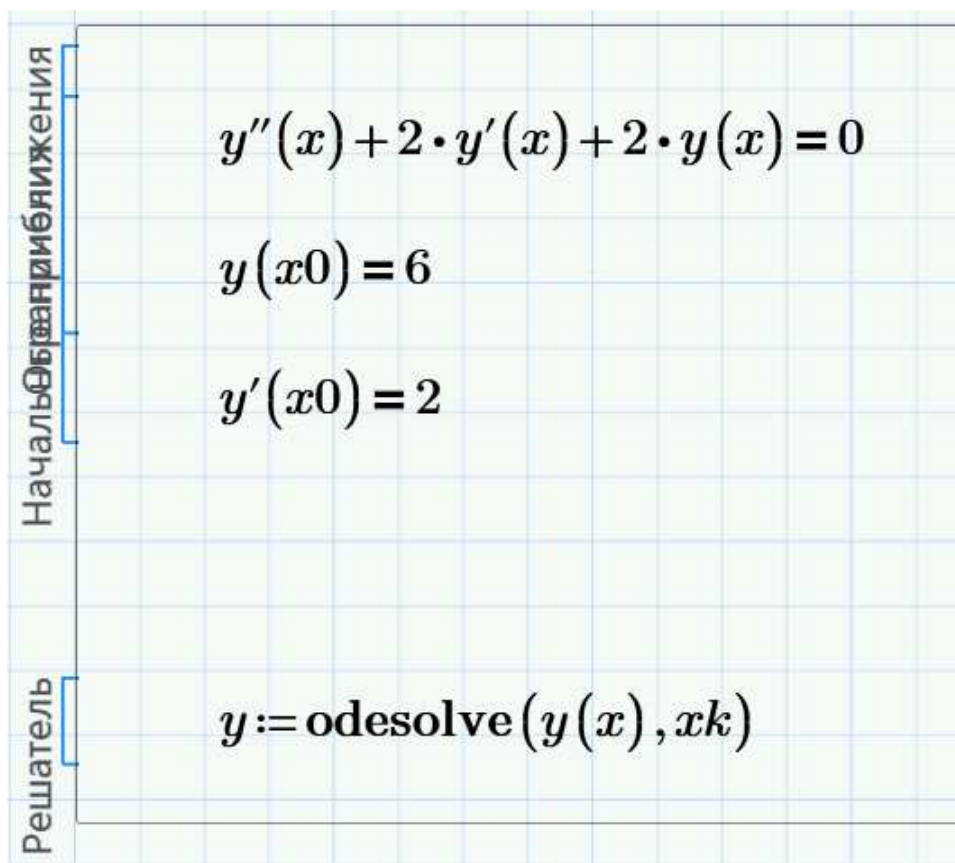
Системную переменную **ORIGIN** задаем равной единице (лента *Расчет*).



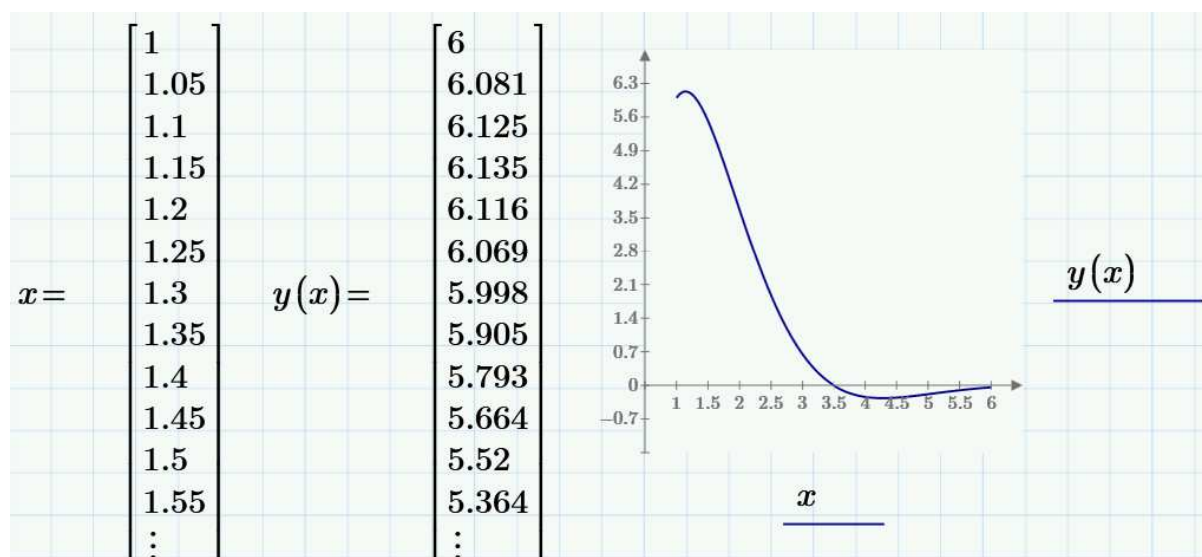
Решение функцией *odesolve*: дифференциальное уравнение решается в вычислительном блоке, внутри блока надо записать само уравнение, его начальные условия и решение функцией *odesolve*.

Функция *odesolve* расположена в ленте *Функции*, в разделах *Дифференциальные уравнения* и *Решение*.





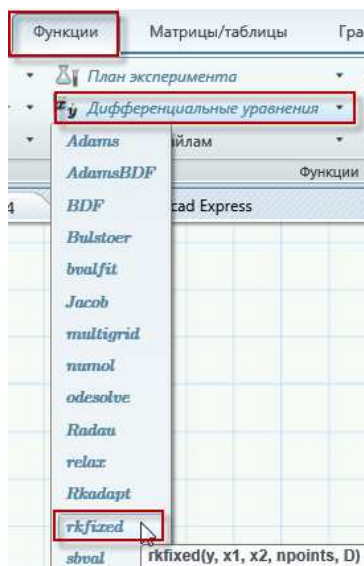
Результат решения в графической и табличной формах приведен ниже.



Решение функцией *rkfixed*: функция *rkfixed* расположена в списке функций раздела *Дифференциальные уравнения* ленты *Функции*.

Для формирования параметров функции необходимо составить вектор начальных условий *y* и дифференциальное выражение *D(x, y)*.

Во избежание путаницы с функцией *y(x)*, решенной ранее в вычислительном блоке, заменим имя функции *y* именем *z*.



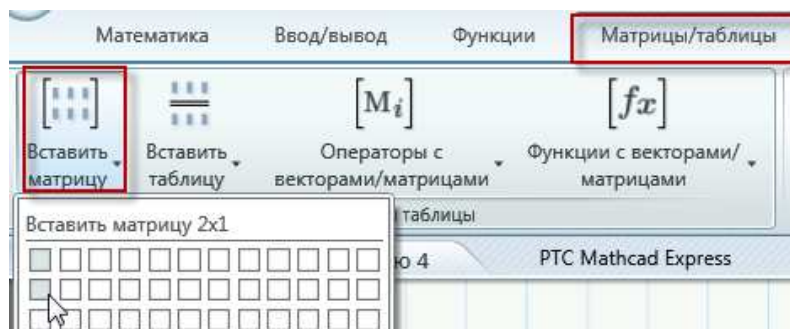
Вектор начальных условий (верхняя строка $y_0(x_0)$, нижняя строка $y'_0(x_0)$):

$$z := \begin{bmatrix} 6 \\ 2 \end{bmatrix}$$

Дифференциальное выражение:

$$D(x, z) := \begin{bmatrix} z_2 \\ -a_1 z_2 - a_2 z_1 \end{bmatrix}$$

Для вставки матрицы воспользуйтесь лентой *Матрицы/таблицы* – *Вставить матрицу*.

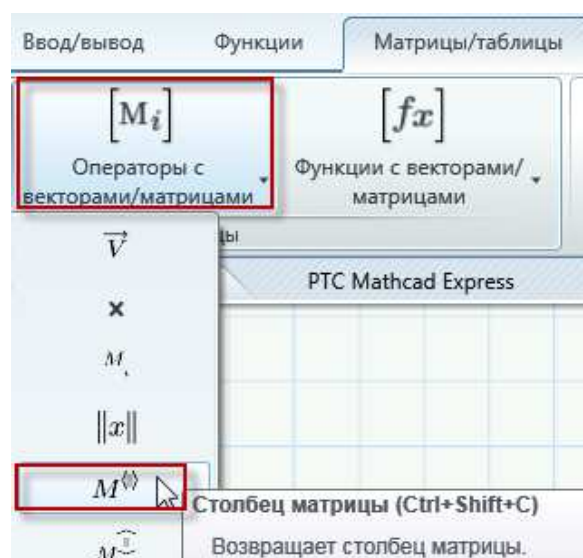


Создадим с помощью функции *rkfixed* массив *t*:

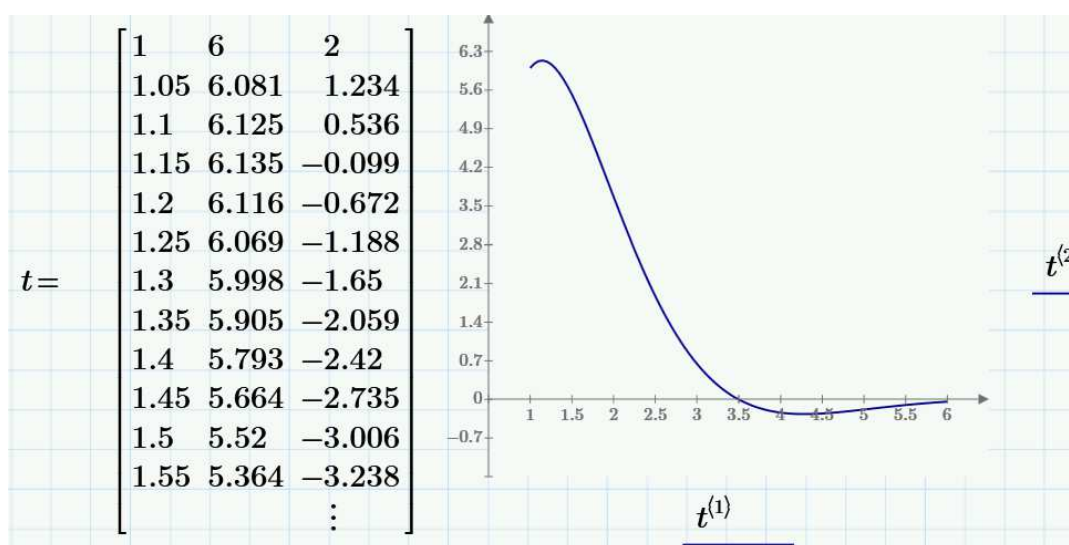
```
t := rkfixed(z, x0, xk, N, D)
```

Он будет содержать в первом столбце аргумент *x*, во втором столбце – функцию *y(x)*, в третьем столбце – функцию *y'(x)*. Для построения графика функции *y(x)* от аргумента *x* надо задать в маркер ординаты графика второй столбец матрицы решения *t*, а в маркер абсцисс графика – первый столбец матрицы решения *t*.

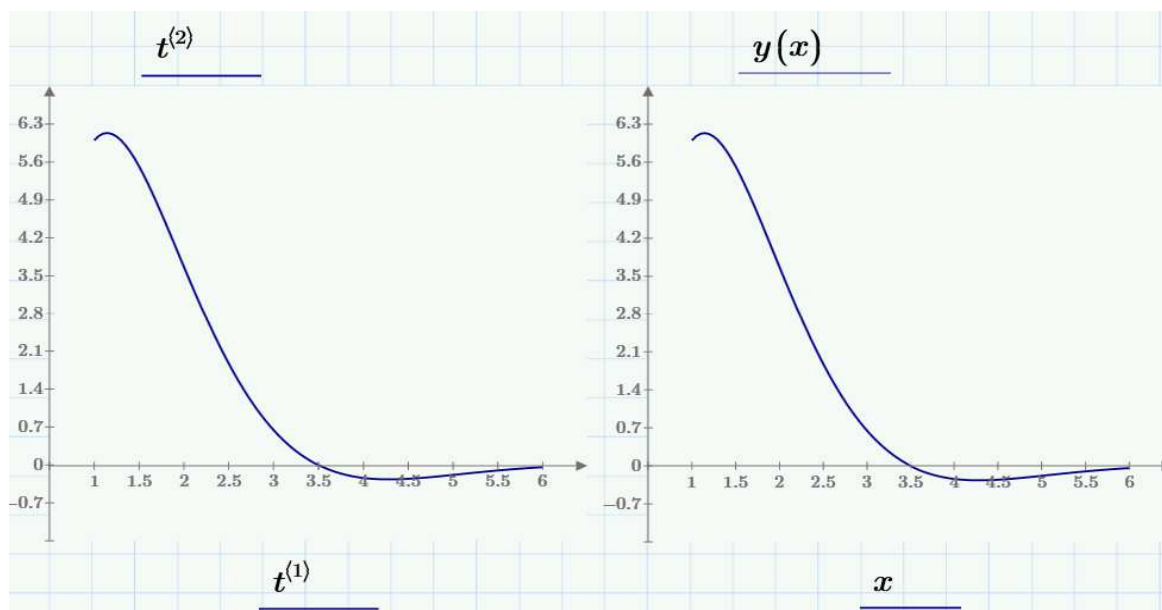
Порядковый номер столбца матрицы *<x>* задается в разделе *Операторы с векторами/матрицами* ленты *Матрицы/таблицы* или комбинацией клавиш Ctrl – Shift – «с».



Табличный и графический выводы решения представлены ниже.



Сравнение результатов первого и второго методов решений в графической и табличной формах представлено ниже.

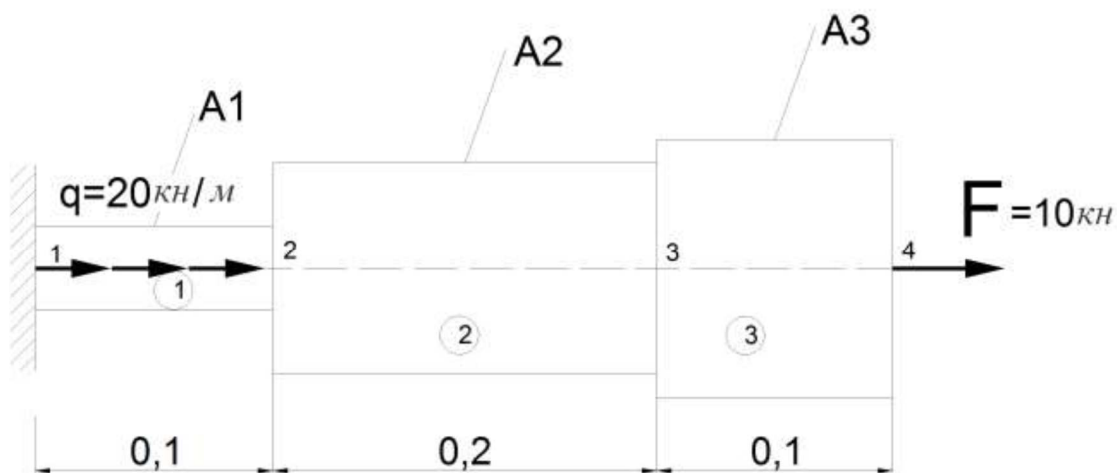


$x =$	$\begin{bmatrix} 1 \\ 1.05 \\ 1.1 \\ 1.15 \\ 1.2 \\ 1.25 \\ 1.3 \\ 1.35 \\ 1.4 \\ 1.45 \\ 1.5 \\ 1.55 \\ \vdots \end{bmatrix}$	$y(x) =$	$\begin{bmatrix} 6 \\ 6.081 \\ 6.125 \\ 6.135 \\ 6.116 \\ 6.069 \\ 5.998 \\ 5.905 \\ 5.793 \\ 5.664 \\ 5.52 \\ 5.364 \\ \vdots \end{bmatrix}$	$t =$	$\begin{bmatrix} 1 & 6 & 2 \\ 1.05 & 6.081 & 1.234 \\ 1.1 & 6.125 & 0.536 \\ 1.15 & 6.135 & -0.099 \\ 1.2 & 6.116 & -0.672 \\ 1.25 & 6.069 & -1.188 \\ 1.3 & 5.998 & -1.65 \\ 1.35 & 5.905 & -2.059 \\ 1.4 & 5.793 & -2.42 \\ 1.45 & 5.664 & -2.735 \\ 1.5 & 5.52 & -3.006 \\ 1.55 & 5.364 & -3.238 \\ & & \vdots \end{bmatrix}$
-------	--	----------	---	-------	--

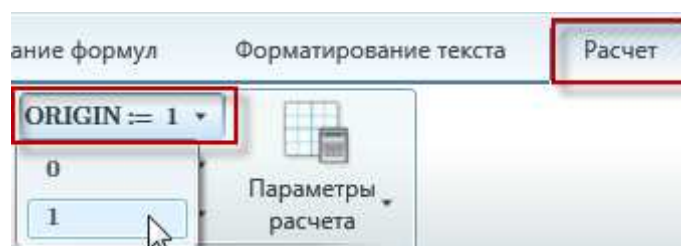
Задание 4. Расчет методом конечных элементов перемещений и усилий в стержне от действия продольной нагрузки

Расчетная схема задачи показана ниже. Нумерация КЭ – слева направо: 1, 2, 3; нумерация узлов – слева направо: 1, 2, 3, 4.

Примем для материала модуль упругости $E = 2 \cdot 10^5$ МПа, расчетное сопротивление $R = 160$ МПа.



Стержень разбивается на три КЭ в соответствии с переменной площадью поперечных сечений и приложенной нагрузкой. Задаем системную переменную, управляющую нумерацией начального элемента матрицы: $ORIGIN:=1$.



Задаем количество КЭ переменной ne , количество узлов – переменной nu :

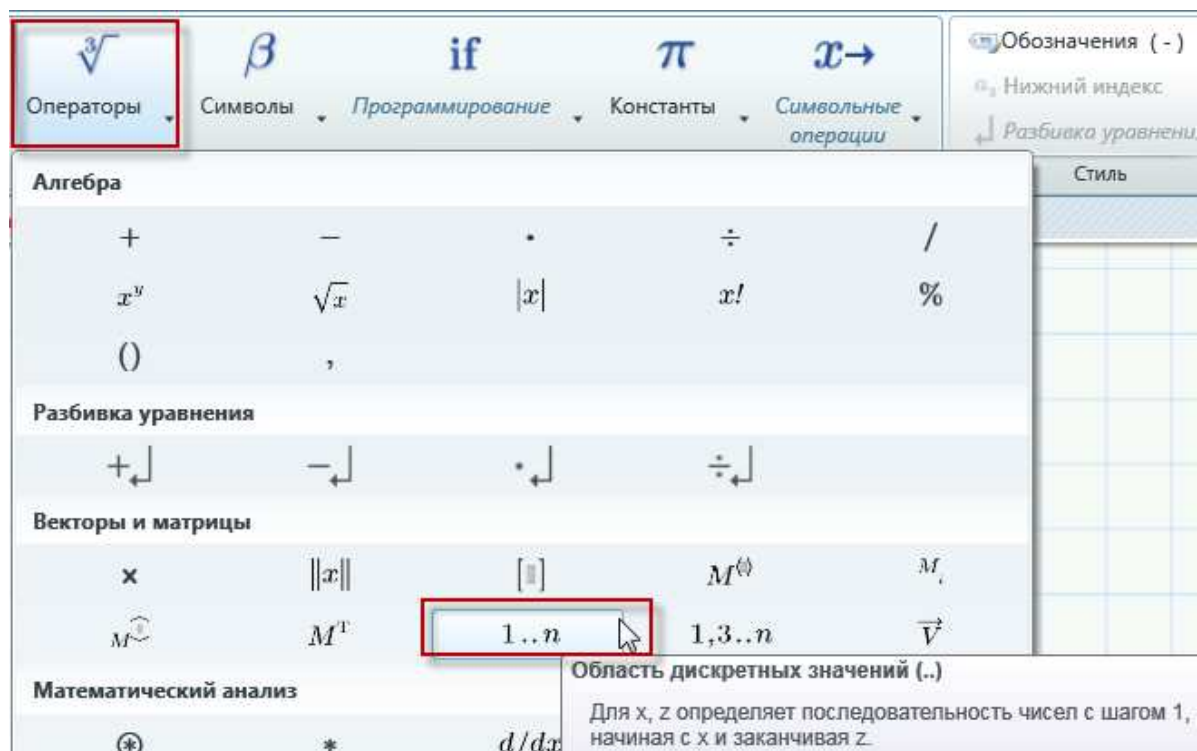
$$ne := 3$$

$$nu := 4$$

Счетчик КЭ задаем ранжированной переменной el :

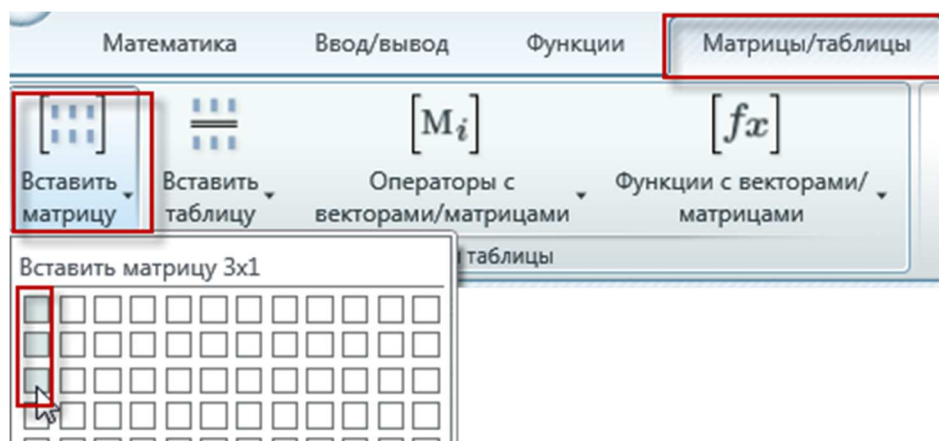
$$el := 1 .. ne \quad el = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Две точки от начального до конечного значений у ранжированной переменной с шагом 1 вводятся обычными точками либо оператором $1..n$ (раздел *Операторы* ленты *Математика*).



$$el := 1 .. ne \quad el = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Начальные данные КЭ (длину L и площадь сечения A) задаем как элементы матрицы.



$$L := \begin{bmatrix} 0.1 \\ 0.2 \\ 0.1 \end{bmatrix} \quad A := \begin{bmatrix} 2 \\ 4 \\ 5 \end{bmatrix} \cdot 10^{-4}$$

Внешние нагрузки задаем в соответствии с направлением по оси X и нумеруем согласно узлу приложения (сосредоточенные силы, приложенные в узле f) или КЭ (распределенные нагрузки q). Поскольку реакция заделки неизвестна, считаем ее равной нулю. Если соответствующей нагрузки нет, ее также задаем равной нулю.

$$q := \begin{bmatrix} 20 \\ 0 \\ 0 \end{bmatrix} \cdot 10^3 \quad f := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 10 \end{bmatrix} \cdot 10^3$$

Расчет проводим в условных единицах, однако имеем в виду, что длина участка L задается в метрах, площадь сечения A – в метрах квадратных, распределенная нагрузка q – в ньютонах на метр, сосредоточенная сила f – в ньютонах, модуль упругости E – в ньютонах на квадратный метр.

$$E := 2 \cdot 10^{11}$$

Формируем глобальную матрицу системы. Матрица жесткости КЭ:

$$kk := \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad K_{el} := kk \cdot \frac{E \cdot A_{el}}{L_{el}}$$

Для ввода нижнего индекса воспользуйтесь клавишей «[». Чтобы вернуться к основанию записи, нажмите клавишу «Пробел».

Глобальная матрица жесткости системы формируется из суммы матриц жесткости КЭ со сдвигом на одну ячейку вправо и вниз. Предварительно создадим базовую нулевую матрицу K0 с размерами 4 × 4:

$$K0_{ni, ni} := 0 \quad K0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Программа-функция *replace* создается для вставки подматриц КЭ в матрицу К0 со сдвигом на одну ячейку вправо и вниз; *r*, *c* – номер соответственно строки и столбца, с которых начинается вставка подматрицы в матрицу; *A* – базовая матрица; *B* – подматрица.

$$replace(A, B, r, c) := \left\| \begin{array}{l} R \leftarrow A \\ \text{for } i \in 1 \dots rows(B) \\ \quad \left\| \begin{array}{l} \text{for } j \in 1 \dots cols(B) \\ \quad R_{i+r-1, j+c-1} \leftarrow B_{i, j} \end{array} \right\| \end{array} \right\| R$$

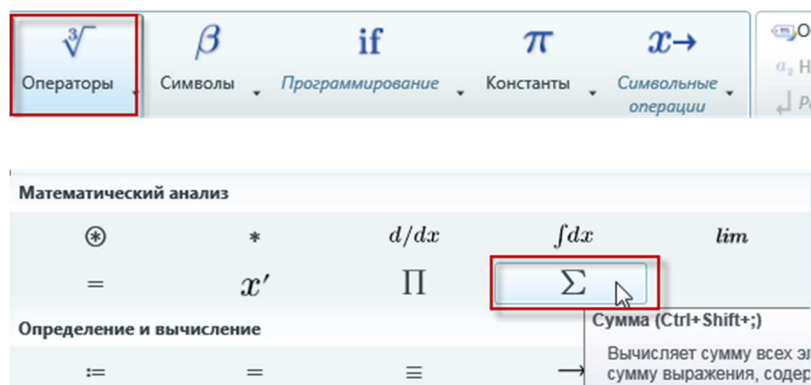
Функции *rows* и *cols* находятся в списке *Функции с векторами/матрицами* ленты *Матрицы/таблицы*.

Глобальная матрица системы КТ формируется на основе функции-программы *replace* (имя функции-программы можно задать любым).

$$KT := \sum_{n=1}^{ne} replace(K0, K_n, n, n)$$

$$KT = \begin{bmatrix} 4 \cdot 10^8 & -4 \cdot 10^8 & 0 & 0 \\ -4 \cdot 10^8 & 8 \cdot 10^8 & -4 \cdot 10^8 & 0 \\ 0 & -4 \cdot 10^8 & 1.4 \cdot 10^9 & -1 \cdot 10^9 \\ 0 & 0 & -1 \cdot 10^9 & 1 \cdot 10^9 \end{bmatrix}$$

Символ суммы Σ расположен в разделе *Операторы* ленты *Математика*.



Необходимо задать граничные условия системы. Первые две ячейки первой строки относятся к первому узлу (т. е. заделке), в котором перемещения отсутствуют по условию закрепления. Чтобы матрица не стала вырожденной, первой ячейке вместо нуля зададим очень большое число.

$$KT_{1,1} := 10^{20} \quad KT_{1,2} := 0$$

$$KT = \begin{bmatrix} 1 \cdot 10^{20} & 0 & 0 & 0 \\ -4 \cdot 10^8 & 8 \cdot 10^8 & -4 \cdot 10^8 & 0 \\ 0 & -4 \cdot 10^8 & 1.4 \cdot 10^9 & -1 \cdot 10^9 \\ 0 & 0 & -1 \cdot 10^9 & 1 \cdot 10^9 \end{bmatrix}$$

Задаем вектор узловых внешних нагрузок. Для этого распределенную нагрузку заменяем ее равнодействующей, деля ее пополам для каждого узла соответствующего КЭ, и складываем с узловой нагрузкой.

$$F_1 := f_1 + \frac{q_1 \cdot L_1}{2}$$

$$F_2 := f_2 + \frac{q_1 \cdot L_1}{2} + \frac{q_2 \cdot L_2}{2}$$

$$F_3 := f_3 + \frac{q_2 \cdot L_2}{2} + \frac{q_3 \cdot L_3}{2}$$

$$F_4 := f_4 + \frac{q_3 \cdot L_3}{2}$$

$$F = \begin{bmatrix} 1 \cdot 10^3 \\ 1 \cdot 10^3 \\ 0 \\ 1 \cdot 10^4 \end{bmatrix}$$

Найдём перемещения в узлах.

$$U := KT^{-1} \cdot F \quad U = \begin{bmatrix} 1 \cdot 10^{-17} \\ 2.75 \cdot 10^{-5} \\ 5.25 \cdot 10^{-5} \\ 6.25 \cdot 10^{-5} \end{bmatrix}$$

Перемещение в первом узле (10^{-17}) можно принять за нуль.

Выполним проверку численного определения перемещения с помощью аналитического расчета.

Реакция в заделке: во избежание ошибок выражение для реакции лучше записать в общем виде, учитывая все возможные нагрузки (отсутствующие нагрузки заданной схемы будут приняты нулевыми):

$$Ra + q_1 \cdot L_1 + q_2 \cdot L_2 + q_3 \cdot L_3 + f_1 + f_2 + f_3 + f_4 = 0$$

или

$$Ra + \sum_{i=1}^{ne} q_i \cdot L_i + \sum_{i=1}^{nu} f_i = 0$$

В уравнении, записанном в общем виде, реакция принимается положительно направленной. Знак, полученный при решении, покажет её истинное направление:

$$Ra := -1.2 \cdot 10^4$$

Усилия на участках следующие:

$$\begin{array}{lll} N1(x) := -Ra - q_1 \cdot x & N1(0) = 1.2 \cdot 10^4 & N1(L_1) = 1 \cdot 10^4 \\ N2 := -Ra - q_1 \cdot L_1 & N2 = 1 \cdot 10^4 & \\ N3 := -Ra - q_1 \cdot L_1 & N3 = 1 \cdot 10^4 & \end{array}$$

Перемещения на границах участков следующие:

$$\Delta 1 := \int_0^{L_1} \frac{N1(x)}{E \cdot A_1} dx \quad \Delta 1 = 2.75 \cdot 10^{-5}$$

$$\Delta 2 := \int_0^{L_2} \frac{N2}{E \cdot A_2} dx + \Delta 1 \quad \Delta 2 = 5.25 \cdot 10^{-5}$$

$$\Delta 3 := \int_0^{L_3} \frac{N3}{E \cdot A_3} dx + \Delta 2 \quad \Delta 3 = 6.25 \cdot 10^{-5}$$

$$U = \begin{bmatrix} 1 \cdot 10^{-17} \\ 2.75 \cdot 10^{-5} \\ 5.25 \cdot 10^{-5} \\ 6.25 \cdot 10^{-5} \end{bmatrix}$$

С правой стороны приведено численное решение перемещений U. Как видно из аналитического расчета, численный метод верен.

Вычисление продольных усилий основано на условии равновесия КЭ. Создадим вектор перемещений КЭ:

$$UE_{el} := \begin{bmatrix} U_{el} \\ U_{el+1} \end{bmatrix}$$

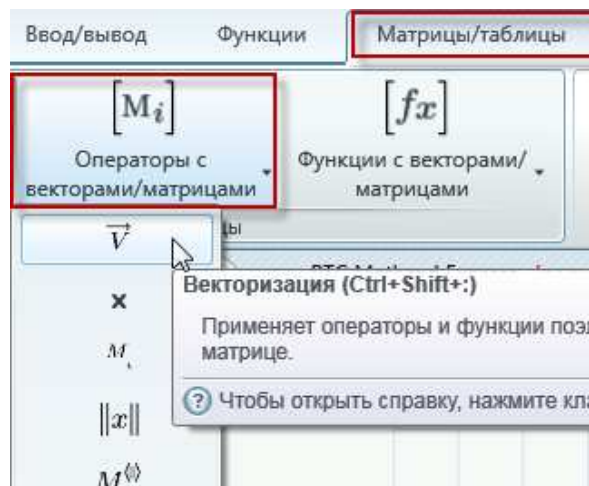
Исключаем из узловых значений внутренних продольных усилий распределенную нагрузку, добавленную при расчете перемещений в вектор узловых сил:

$$N_{el} := K_{el} \cdot UE_{el} - \begin{bmatrix} \frac{q_{el} \cdot L_{el}}{2} \\ \frac{q_{el} \cdot L_{el}}{2} \end{bmatrix}$$

Учтем, что вектор положительной (с точки зрения вызываемой деформации) продольной силы в левом узле КЭ имеет направление, противоположное положительному направлению оси X системы:

$$\overline{N_{el}} := N_{el} \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Символ матрицы-вектора располагается в списке *Операторы с векторами/матрицами* ленты *Матрицы/таблицы*.



Для границ каждого КЭ получим значения продольной силы (сравним с аналитическим решением, полученным ранее):

$$N_1 = \begin{bmatrix} 1.2 \cdot 10^4 \\ 1 \cdot 10^4 \end{bmatrix} \quad N_2 = \begin{bmatrix} 1 \cdot 10^4 \\ 1 \cdot 10^4 \end{bmatrix} \quad N_3 = \begin{bmatrix} 1 \cdot 10^4 \\ 1 \cdot 10^4 \end{bmatrix}$$

$$\sigma_{el} := \frac{N_{el}}{A_{el}} \quad \varepsilon_{el} := \frac{\sigma_{el}}{E}$$

Численные результаты следующие:

$$\sigma_1 = \begin{bmatrix} 6 \cdot 10^7 \\ 5 \cdot 10^7 \end{bmatrix} \quad \sigma_2 = \begin{bmatrix} 2.5 \cdot 10^7 \\ 2.5 \cdot 10^7 \end{bmatrix} \quad \sigma_3 = \begin{bmatrix} 2 \cdot 10^7 \\ 2 \cdot 10^7 \end{bmatrix}$$

$$\varepsilon_1 = \begin{bmatrix} 3 \cdot 10^{-4} \\ 2.5 \cdot 10^{-4} \end{bmatrix} \quad \varepsilon_2 = \begin{bmatrix} 1.25 \cdot 10^{-4} \\ 1.25 \cdot 10^{-4} \end{bmatrix} \quad \varepsilon_3 = \begin{bmatrix} 1 \cdot 10^{-4} \\ 1 \cdot 10^{-4} \end{bmatrix}$$

Определим максимальное по модулю напряжение: максимальное напряжение в первом элементе (первом узле) $\sigma_1 = 60$ МПа. Условие прочности соблюдается, так как $\sigma_{\max} < R$.

Построим эпюры. Создадим вектор координат узлов системы LX при помощи функции-программы.

$$LX := \left\| \begin{array}{l} LX_1 \leftarrow 0 \\ \text{for } i \in 1 \dots ne \\ \quad \left\| \begin{array}{l} LX_{i+1} \leftarrow LX_i + L_i \\ LX \end{array} \right\| \end{array} \right\| \quad LX = \begin{bmatrix} 0 \\ 0.1 \\ 0.3 \\ 0.4 \end{bmatrix}$$

Задаем в виде ранжированной переменной длину каждого КЭ (шаг подбирается к каждой эпюре, шаг от 0.01 позволит построить заштрихованную эпюру). В данном случае зададим шаг 0.001:

$$X(el) := LX_{el}, LX_{el} + 0.001 \dots LX_{el+1}$$

$$X1 := X(1) \quad X2 := X(2) \quad X3 := X(3)$$

Для правого предела аргумента графиков зададим параметр

$$LT := \sum_{el=1}^{ne} L_{el}$$

Запишем функции в общем виде. Учитывая то, что все величины определялись как матричные, введем нижний индекс 1.

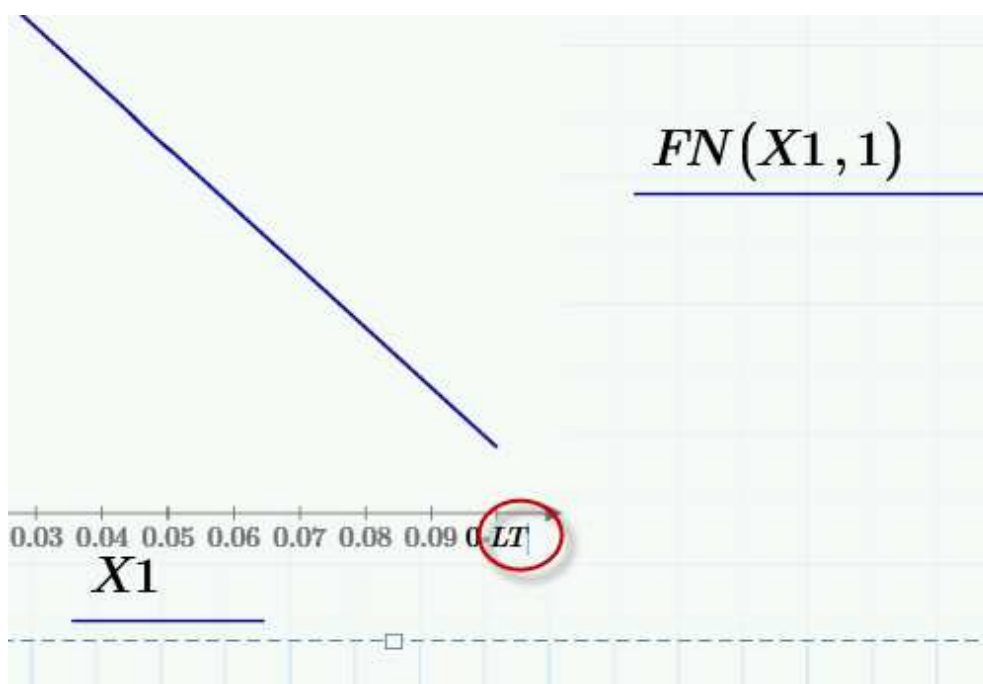
Продольные усилия FN и напряжения Fσ:

$$FN(x, el) := \left(N_{el} \right)_1 - q_{el} \cdot (x - LX_{el}) \quad F\sigma(x, el) := \frac{FN(x, el)}{A_{el}}$$

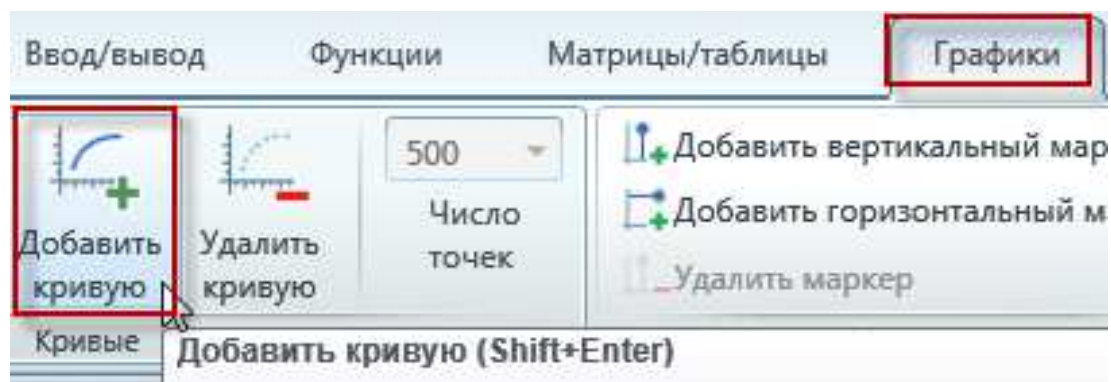
Относительные продольные деформации $F\varepsilon$ и абсолютные перемещения FU :

$$F\varepsilon(x, el) := \frac{F\sigma(x, el)}{E} \quad FU(x, el) := \int_{LX_{el}}^x F\varepsilon(x, el) dx + U_{el}$$

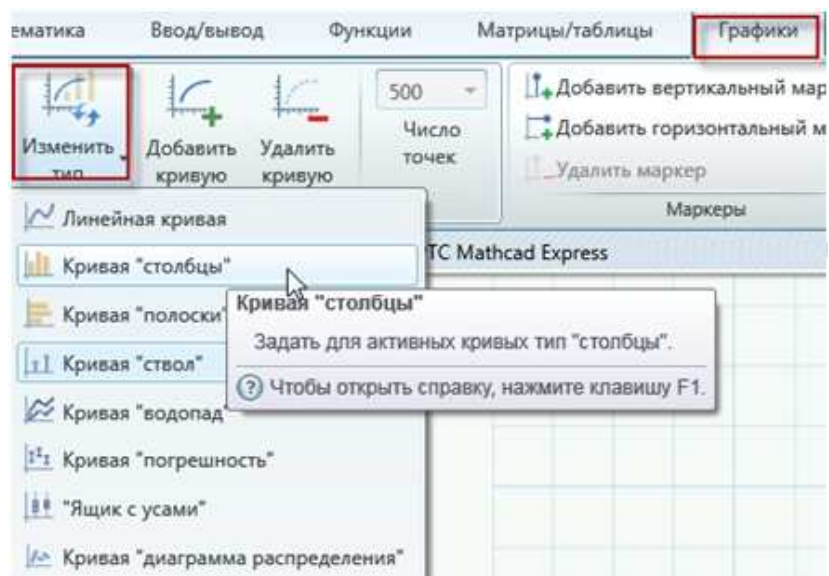
Вставьте график первой функции $FN(X1,1)$. Отредактируйте правый предел числовой оси, задав значение LT .



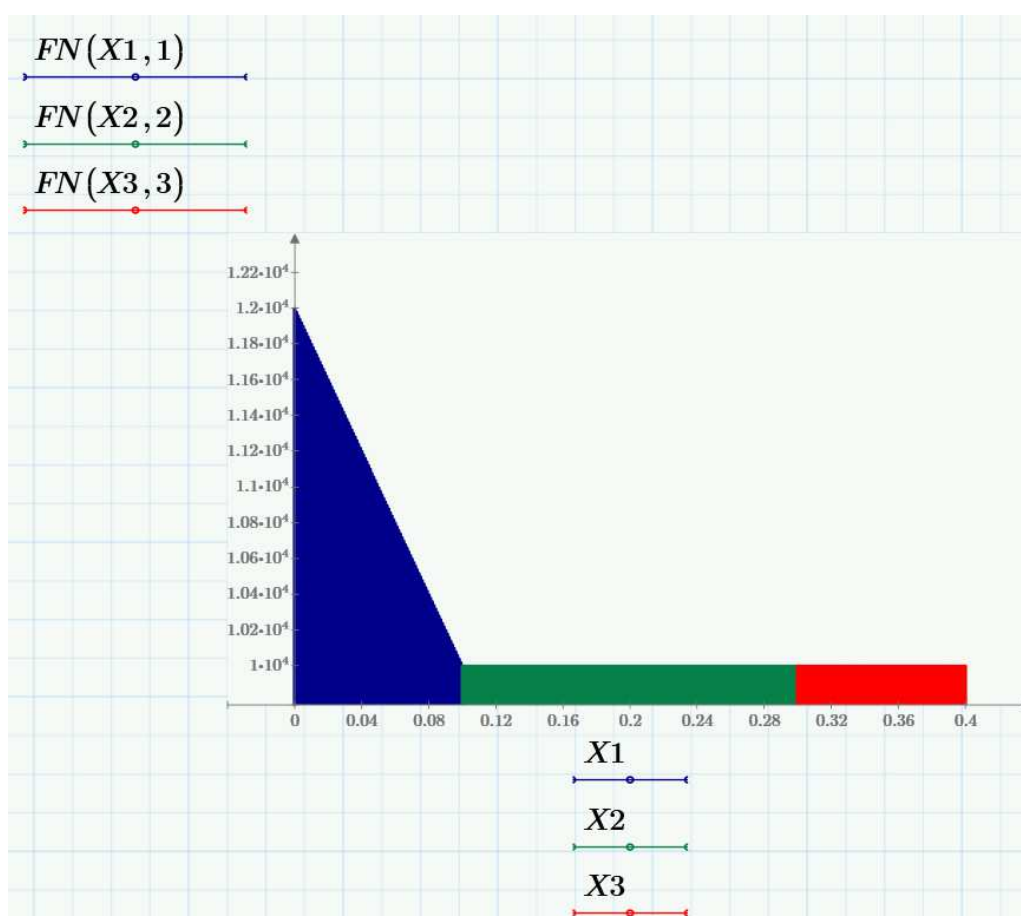
При помощи команды *Добавить кривую* вставьте ещё две функции: $FN(X2,2)$ и $FN(X3,3)$. Каждой функции назначьте свой аргумент ($X1$, $X2$, $X3$ соответственно) также при помощи команды добавления кривой.



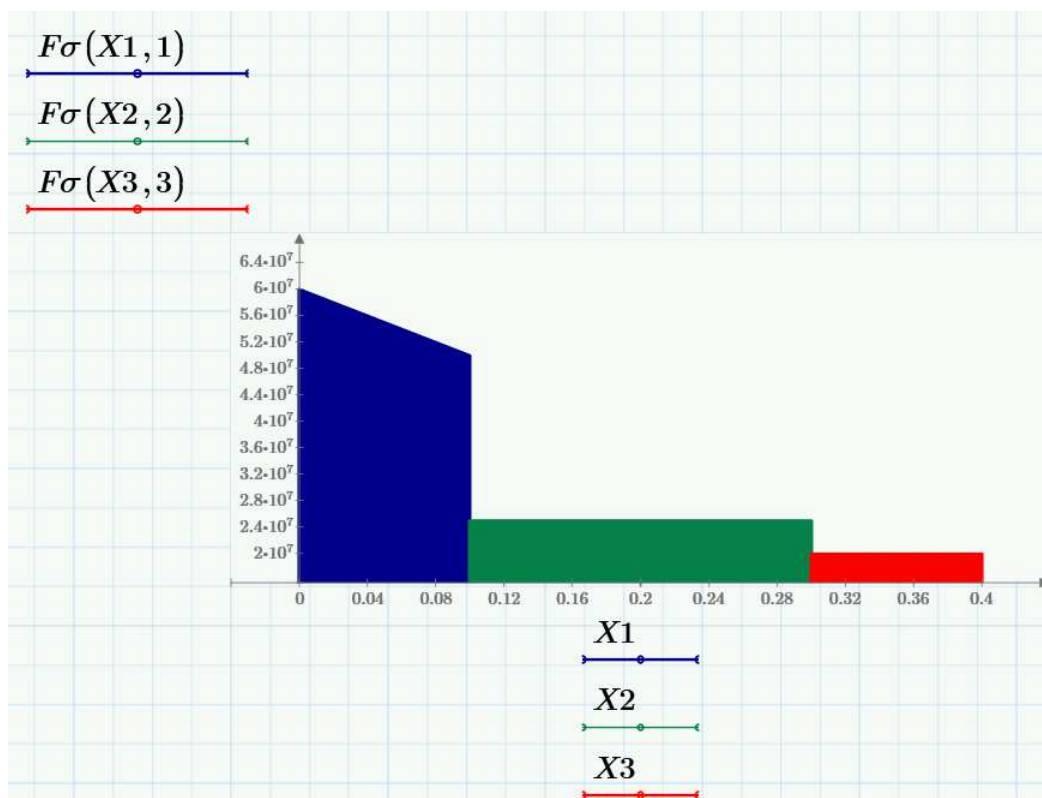
Для того чтобы получилась заштрихованная область на графиках, в ленте *Графики* задайте тип «столбцы».



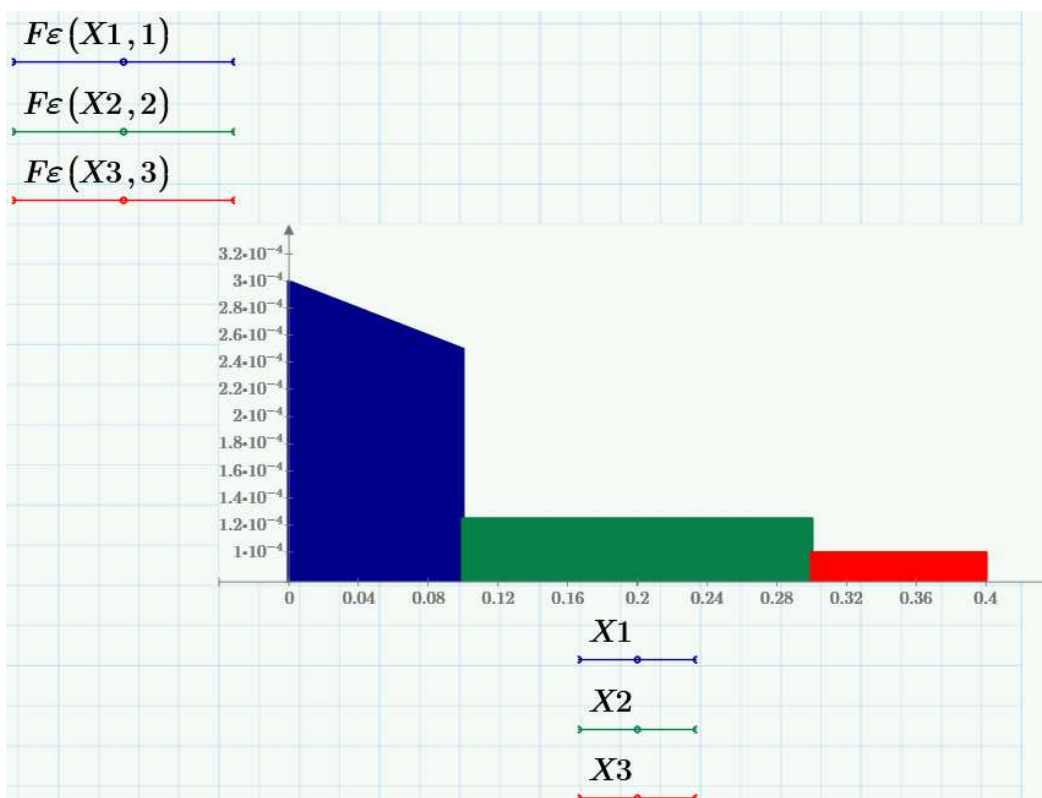
Плотность графиков (шаг для $X(e1)$) подберите самостоятельно. Эпюра продольных сил приведена ниже.



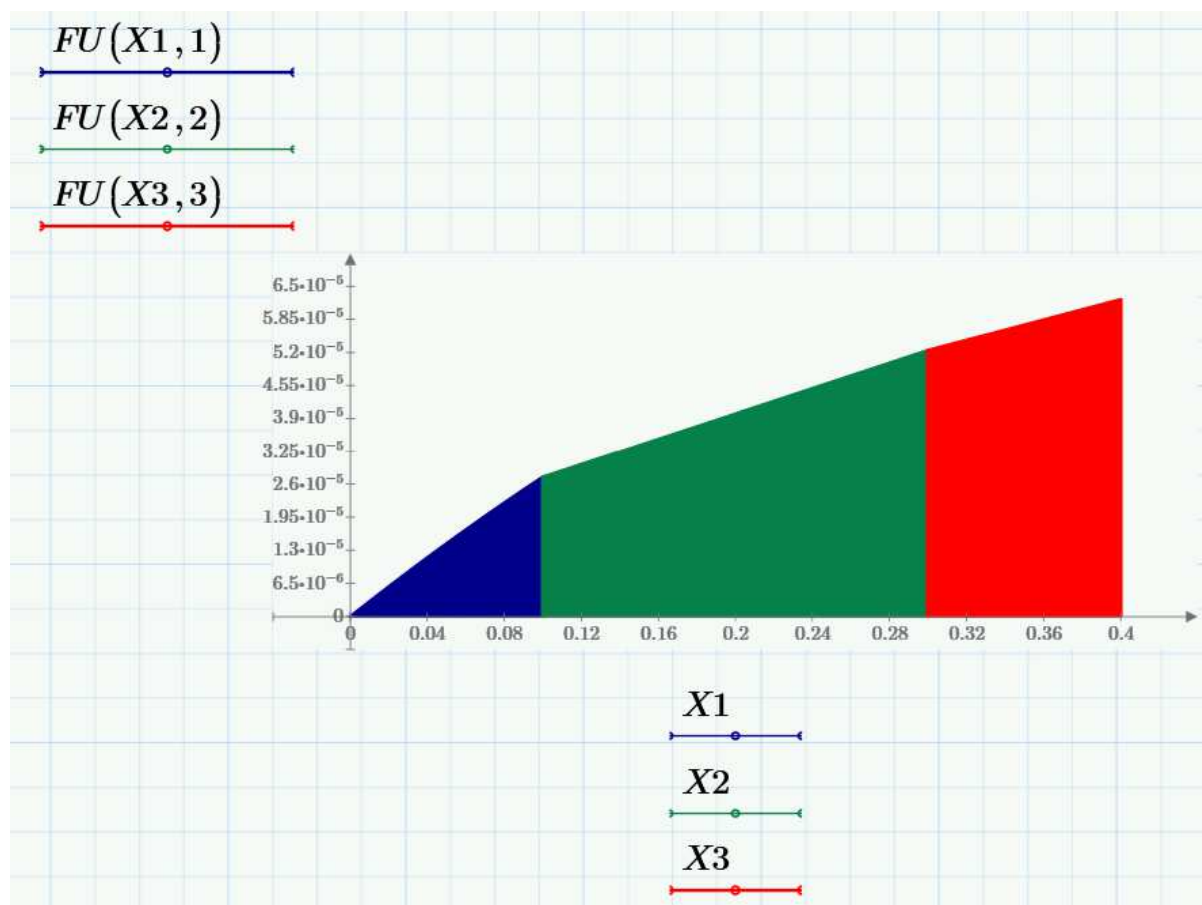
Эпюра напряжения приведена ниже.



Эпюра относительной продольной деформации приведена ниже.



Эпюра абсолютной продольной деформации приведена ниже.



ЗАКЛЮЧЕНИЕ

Учебно-практическое пособие по дисциплине «Численные методы решения инженерно-технических задач в строительстве» дает возможность магистранту самостоятельно изучить алгоритм решения ряда классических инженерных задач численными методами, использовать приобретенный опыт в своей дальнейшей профессиональной деятельности. На примере представленных заданий магистранту предлагается освоить программный комплекс MathCAD. Изученные методики решения задач с применением MathCAD существенно ускорят рутинные вычисления и найдут широкое практическое применение.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Самарский, А. А. Введение в численные методы / А. А. Самарский. – М. : Наука, 1982. – 272 с.
2. Бахвалов, Н. С. Численные методы : учеб. пособие / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – М. : Наука, 1987. – 600 с.
3. Дьяконов, В. MathCAD 2001 : учеб. курс / В. Дьяконов. – СПб. : Питер, 2001. – 621 с. – ISBN 5-318-00367-2.
4. Херхагер, М. MathCAD 2000 : Полное руководство : пер. с нем. / М. Херхагер, Х. Партолль. – Киев : BHV, 2000. – 416 с. – ISBN 3-8273-1343-0.
5. Марчук, Г. И. Методы вычислительной математики : учеб. пособие для вузов по специальности «Прикладная математика» / Г. И. Марчук. – М. : Наука, 1977. – 455 с.
6. Численные методы в строительстве : метод. указания / Владим. гос. ун-т ; сост.: Н. А. Малова, Квофие Р. Охене. – Владимир, 2005. – 44 с.
7. Очков, В. Ф. MathCAD 14 для студентов, инженеров и конструкторов / В. Ф. Очков. – СПб. : БХВ-Петербург, 2007. – 356 с. – ISBN 978-5-9775-0129-3.
8. Кирьянов, Д. В. MathCAD 14 : [наиболее полное руководство] / Д. В. Кирьянов. – СПб. : БХВ-Петербург, 2007. – 682 с. – ISBN 978-5-9775-0106-4.
9. Образовательный математический сайт Exponenta.ru. Раздел MathCAD [Электронный ресурс]. – Режим доступа: <http://old.exponenta.ru/soft/Mathcad/Mathcad.asp> (дата обращения: 12.12.2021).
10. РТС [Электронный ресурс]. – Режим доступа: <https://www.ptc.com> (дата обращения: 12.12.2021).

Учебное электронное издание

МАЛОВА Наталья Анатольевна

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ
ИНЖЕНЕРНО-ТЕХНИЧЕСКИХ ЗАДАЧ В СТРОИТЕЛЬСТВЕ

Учебно-практическое пособие

Редактор Т. В. Евстюничева

Технические редакторы Ш. В. Абдуллаев, Н. В. Пустовойтова

Компьютерная верстка Л. В. Макаровой

Выпускающий редактор А. А. Амирсейидова

Системные требования: Intel от 1,3 ГГц; Windows XP/7/8/10;
Adobe Reader; дисковод CD-ROM.

Тираж 25 экз.

Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых
Изд-во ВлГУ
rio.vlgu@yandex.ru