

Сайт этюда: <https://www.ptcusercommunity.com/thread/127826>

Многие задачи по математике, физике, химии, механике, термодинамике и другим школьным и вузовским дисциплинам сводятся к решению уравнений и систем уравнений [41]. Поэтому полезно будет узнать, какие инструменты для решения такого рода задач есть у пакета Mathcad, очень популярного у школьников, студентов инженеров и ученых. Эти инструменты объединены в группу "Решение уравнений" встроенных функций Mathcad, которые используют различные *численные методы*. В среде Mathcad 15 названия этих методов можно узнать, если на имени некоторых функции, их реализующих, нажать правую кнопку мыши. Через это же нажатие можно изменить некоторые установки численных методов.

В группе "Решение уравнений" традиционно находятся *семь* функций (см. второе название этюда). Есть такой классический вестерн "Великолепная семерка", голливудская адаптация философской драмы Акиры Куросавы "Семь самураев". В американском фильме "главным" в семерке ковбоев, защищавших крестьян от бандитов, был Крис Адамс, которого сыграл Юл Бриннер. Семерка — это некое сакральное число не только в культуре и истории (семь древних мудрецов, семь чудес света, семь дней недели, семь нот в гамме и т.д.), но и в естествознании — семь цветов радуги, семь базовых единиц измерения международной системы СИ и т.д.

Есть еще в среде Mathcad и оператор **solve** для символьного (аналитического) решения уравнений и их систем. Описание этих инструментов будет сделано на несложных школьных "водных" примерах.

Задача 1. Моторная лодка прошла по реке в одну сторону ($L = 10$ km), а потом вернулась в исходную точку, затратив на этот "круиз" 1 час 45 минут (t). Спрашивается, какова скорость течения воды в реке (неизвестная x), если собственная скорость лодки (v — скорость в стоячей воде) равна 12 км/ч (kph — мы, следуя современному тренду, будем стараться использовать международное написание единиц измерения).

Раньше подобные "школьные" задачи решались в несколько действий. Но не всякая задача может быть решена пошагово. Поэтому-то люди и придумали алгебру. В древние времена, например, пока не была выведена формула корней квадратного уравнения, не каждое такое уравнение можно было решить пошагово, причем решения были очень хитроумными. Кстати, нашу задачу о моторной лодке тоже сходу нельзя решить пошагово. Читатель, найди, если сможешь, пошаговое решение этой задачи и сравни найденное решение с тем, какое приведено ниже. Первый шаг такого решения может быть таким: $2 \cdot 10 \text{ km} / 12 \text{ kph} = 1 \text{ hr} + 40 \text{ min}$: движение в текущей, а не в стоячей воде увеличило

Этюд 4

время пути на 5 минут. Многие студенты, подлаживаясь под несколько устаревшие требования преподавателей, проводят пошаговые вычисления на компьютере в среде того же пакета Mathcad, и переписывают ответ в расчетную записку, имитируя ручной счет.

Сейчас в связи с широким использованием компьютеров в образовательной сфере принято составлять, а затем решать уравнения, выбирая их подходящие корни, и об этом мы рассуждали в этюде 1. Пойдем и мы по этому пути, но, составив уравнение, попробуем решить его не на бумаге, а на компьютере в среде математической программы Mathcad.

В нашей задаче о моторной лодке время в пути t — это суммарное время, затраченное на поездку в одну сторону $L / (v + x)$ (условно будем считать, что это движение по течению реки), и в обратную сторону (против течения) $L / (v - x)$. Поэтому наше уравнение будет иметь вид:

$$(L / (v + x)) + (L / (v - x)) = t$$

0. solve

Начнем с решения полученного уравнения средствами *символьной математики* Mathcad. Формальное, более правильное и более длинное название символьной математики — *компьютерные аналитические преобразования*, но у нас прижилась калька с английского — *symbolic math*. Это название мы и будем использовать далее.

Если *численная математика* (которая, повторяем, тоже есть в среде Mathcad и составляет его основу) оперирует числами, хранящимися в переменных, то *символьная математика* работает сначала с самими переменными-символами, а потом уж — с числами, какие в них хранятся.

На рисунке 4.1 показано решение уравнения прямого и обратного движения моторной лодки по реке с помощью оператора **solve** *символьной математики* Mathcad (на этом и некоторых других рисунках будут показаны позиции меню и панели инструментов Mathcad Prime и Mathcad 15 для решения описываемых задач).

$$\frac{L}{v+x} + \frac{L}{v-x} = t \xrightarrow{\text{solve, } x} \begin{bmatrix} \frac{\sqrt{t \cdot v \cdot (t \cdot v - 2 L)}}{t} \\ -\frac{\sqrt{t \cdot v \cdot (t \cdot v - 2 L)}}{t} \end{bmatrix}$$

Рис. 4.1. Аналитическое решение задачи о движении моторной лодки

Из полученного общего аналитического решения (из вектора с двумя элементами-формулами — см. рис. 4.1) можно скопировать один элемент, подставить в него исходные значения переменных L , t и v (см. рис. 2) и получить численный ответ — скорость течения воды в реке. Ответ будет выдан в метрах, деленных на секунду (Mathcad по умолчанию ориентирован на СИ — на международную систему исчислений), и подправлен пользователем на более уместные тут километры в час (kph). Mathcad, повторяем, — это не просто математический, а физико-математический пакет [4]: переменные Mathcad хранят не числа, а физические величины (длину, время, силу, массу и т.д.), что очень полезно при расчетах в задачах с физическим смыслом. Это существенно ускоряет и упрощает расчеты, позволяет избежать ошибок в них, а также автоматизирует соответствующие преобразования единиц измерения.

$$L := 10 \text{ km} \quad t := 1 \text{ hr} + 45 \text{ min} \quad v := 12 \text{ kph}$$

$$\frac{\sqrt{t \cdot v \cdot (t \cdot v - 2L)}}{t} = 2.619 \text{ kph}$$

Рис. 4.2. Решение задачи о моторной лодке по найденной на рис. 4.1 формуле

Спрашивается, для чего же тогда в пакете Mathcad есть и численная математика, если задачу можно просто и красиво решить с помощью символьной математики? Дело в том, что символьная математика, нацеленная на выдачу всех решений в виде формул (абсолютная точность!), часто не справляется с более-менее сложной задачей, и это показано на рисунках 4.3 и 4.4.

$$\frac{L}{v+x} + \frac{L}{v-x^2} = t \xrightarrow{\text{solve, } x} ?$$

Полученный результат этой символьной операции слишком длинный для отображения, но он может использоваться в последующих расчетах, если будет присвоен функции или переменной.

Рис. 4.3. Поиск корня уравнения: очень объемный скрытый ответ

На рисунке 4.3 в уравнении движения моторной лодки один из иксов был возведен в квадрат. Физический смысл уравнения пропал (складывается скорость с квадратом скорости).

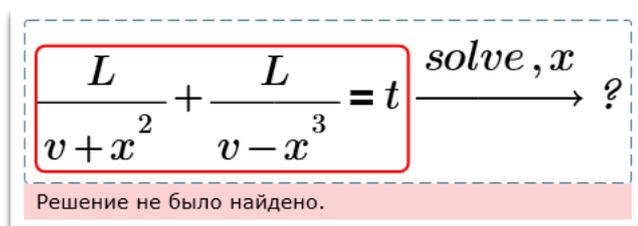
Примечание

Таковыми "нефизическими" формулами заполнены все современные учебники и задачки по математике. И это не очень хорошо, вернее, совсем плохо. Хорошо тогда, когда за формулой скрывается какая-нибудь физическая реальность. Такое направление математики условно называют "натуральная математика". Проблема размерностей исчезнет сразу же, как только мы вспомним, что еще во времена Франсуа Виета господствовал принцип однородности, который требовал в подобных случаях умножать икс на некую единицу, получая везде квадраты скорости. Но, как водится, у современных математиков, умножение на единицу не производится, а размерности игнорируются. Типичная задача, часто предлагаемая школьникам и студентам для решения в рамках курса математического анализа, о которой мы уже писали в этюде 3 — это сравнение самой

Этюд 4

функции и ее производной. Но сравнение функции с ее первой производной равносильно физически некорректному сравнению пути (длины) и скорости — разных физических величин.

Но сейчас главное не физический смысл уравнения, показанного на рис. 4.3. Важно то, что пакет Mathcad, решив это чуть усложненное уравнение, не смог вывести на дисплей ответ — настолько тот оказался громоздким. Но это еще полбеда, которую можно исправить, прочитав комментарий на рис. 4.3. В этюде 8 на рис. 8.13 показано, как вытаскивается такой объемный ответ символической математики. Настоящая "беда" показана на рис. 4.4 для еще более усложненного уравнения. Если, например, один x возвести в квадрат, а другой в куб, то символическая математика Mathcad "поднимет руки вверх" и скажет: "Сдаюсь!".


$$\frac{L}{v+x^2} + \frac{L}{v-x^3} = t \quad \text{solve, } x \quad ?$$

Решение не было найдено.

Рис. 4.4. Поиск корня уравнения: решение не найдено

Если в константы этого "нефизического" уравнения подставить безразмерные численные значения известных величин, то хотя бы один действительный корень у этого уравнения зафиксировать удастся — см. рис. 4.5, где данная задача решена графически.

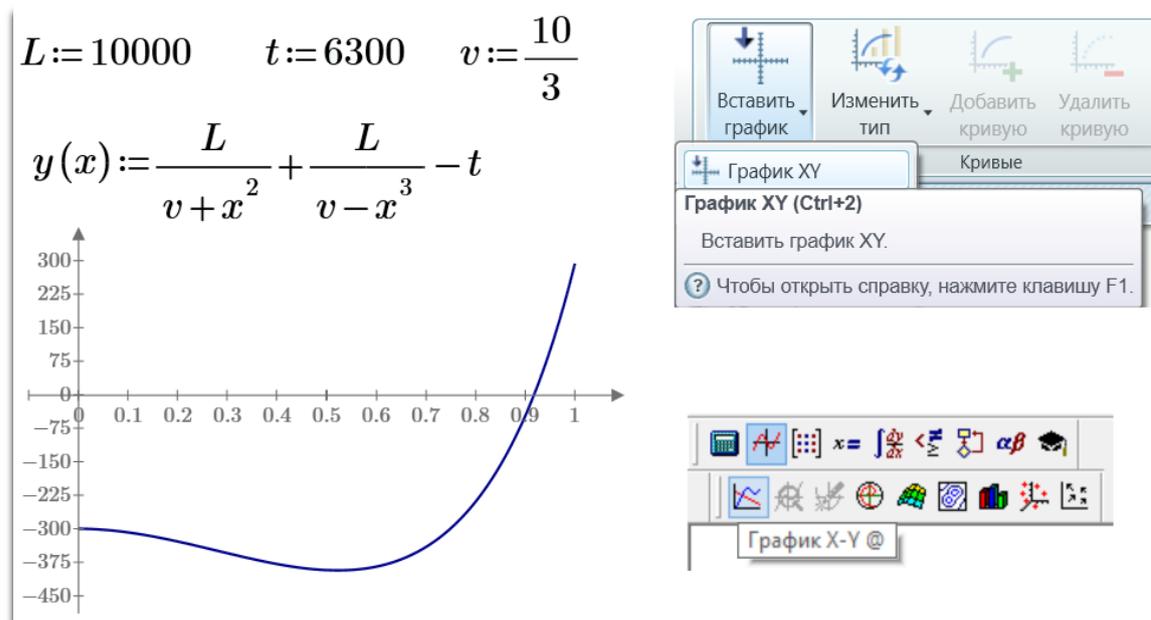


Рис. 4.5. Графический поиск нуля функции

1. root+root

Из рисунка 4.5 видно, что у нашего уравнения, превращенного в функцию пользователя переносом переменной t в левую часть, есть как минимум один действительный корень в районе числа 0.9. Уточнить численное значение этого корня поможет встроенная в Mathcad функция `root` — см. рис. 4.6 и 4.7.

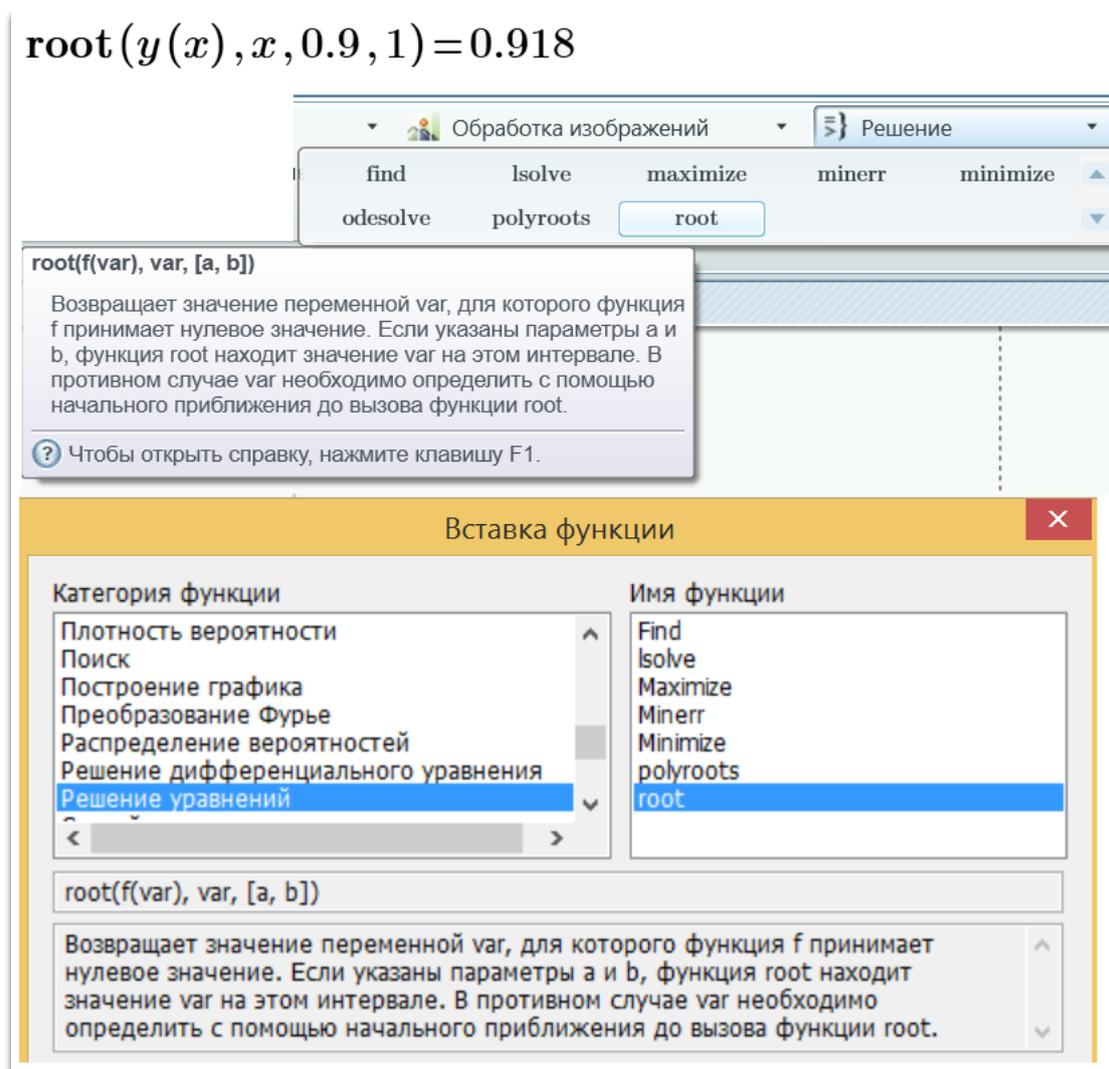


Рис. 4.6. Работа в среде Mathcad встроенной функции `root` с четырьмя аргументами

На рисунке 4.6 показан вызов функции `root` с четырьмя аргументами, а на рис. 4.7 с двумя. В обоих случаях ответ выведен с тремя знаками после десятичной точки. Но можно вывести и большее число знаков — до 15. В первом случае (рис. 4.6) нуль функции $y(x)$ ищется методом деления пополам на интервале, заданном третьим и четвертым аргументами функции `root` (см. авторскую анимацию этого метода на сайте <http://communities.ptc.com/videos/1468>). Во втором случае (рис. 4.7) нуль функции рассчитывается методом секущих с опорой на первое предположение $x := 1$ (анимация —

<http://communities.ptc.com/videos/1466>). В среде Mathcad для вычисления нуля функции пользователя фактически есть две одинаковые по имени, но разные по своей сути встроенные функции **root**.

$$x := 1 \quad \text{root}(y(x), x) = 0.918$$

Рис. 4.7. Работа в среде Mathcad встроенной функции `root` с двумя аргументами

На рисунке 4.8 показана работа функции `root` на более сложном примере — с функцией пользователя $\sin(x)/x$, которая имеет бесконечное число нулей. На отрезке от 2 до 7 функция $y(x)$ имеет два нуля (π и 2π), но четырехаргументная функция `root` ответа не выдала, так как функция $y(x)$ имеет одинаковые знаки на концах этого отрезка и функция `root` считает, что там не может быть корней уравнения. На отрезке от 1 до 17 нулей уже пять, один из которых (9.425) найден четырехаргументной функцией `root`. На концах отрезка от 1 до 17 функция $y(x)$ имеет разные знаки. При первом приближении, равном 0.01, двухаргументная функция `root` выдала не ближайший нуль (3.14), а "очень-очень дальний": 298.451. Понять эти особенности применения функции `root` можно только после детального рассмотрения численных методов, заложенных в эту функцию — метода деления отрезка пополам и метода секущих.

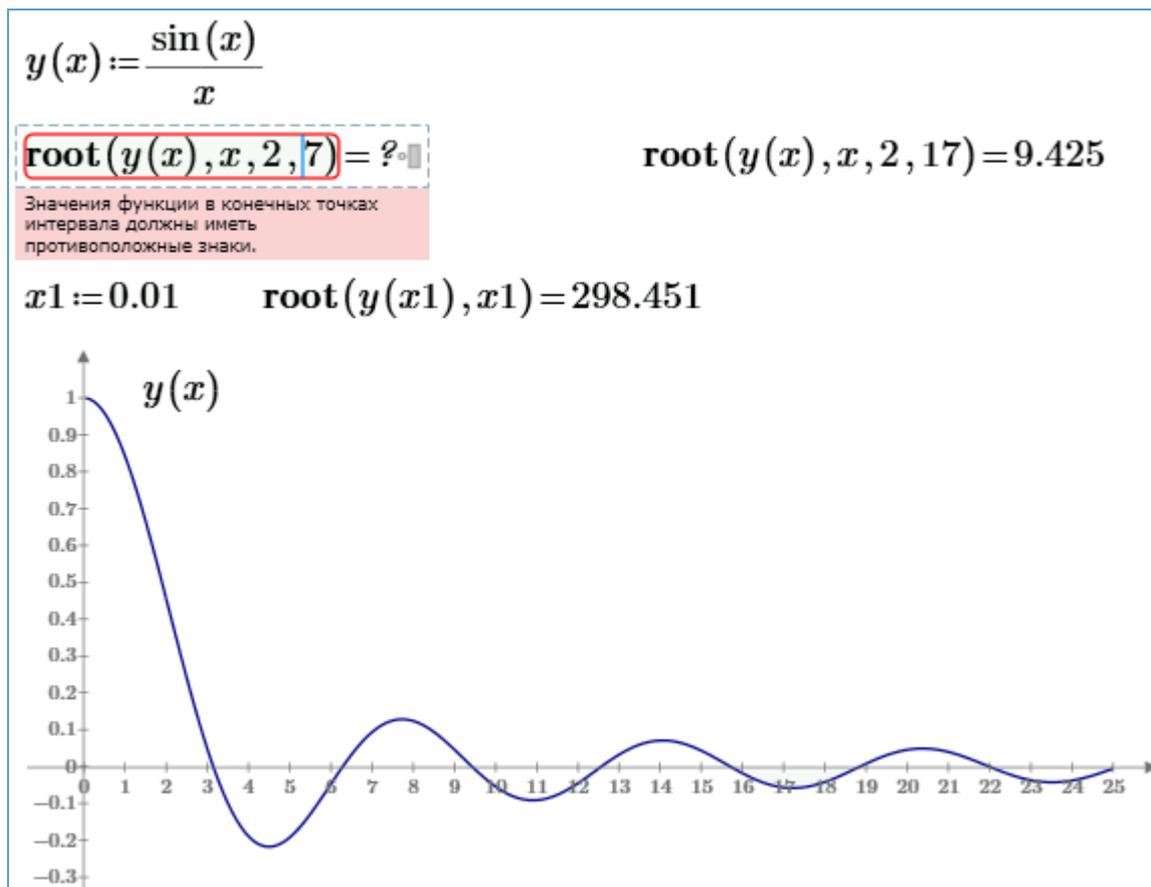


Рис. 4.8. Особенности работы двухаргументной функции `root`

Ранее мы отметили, что символьная математика Mathcad оперирует не числами, а символами — самими переменными, хранящими или не хранящими числа. Но это не совсем так.

Если какая-либо переменная выражения хранит численное значение, то символьная математика будет работать не с самой переменной (с символом), а с числом, хранящимся в этой переменной. На рисунке 4.3 была показана осечка символьной математики Mathcad при решении довольно простого уравнения. Но если всем переменным этого уравнения, кроме переменной x , задать численные значения, то символьный оператор `solve` успешно справится с задачей — см. рис. 4.9.

$$\frac{L}{v+x^2} + \frac{L}{v-x^3} = t \xrightarrow{\text{solve}, x} \begin{bmatrix} 0.91753907432064754652 \\ -0.10072745491075931273 - 0.504473077723681i \\ -0.10072745491075931273 + 0.50447307772368065275i \\ -0.35804208224956446053 - 1.43206766221807i \\ -0.35804208224956446053 + 1.4320676622180746717i \end{bmatrix}$$

Рис. 4.9. Численный ответ символьного оператора

На рисунке 4.9 показано, что "символьный" оператор `solve` в отличие от "численной" функции `root` выдал все пять корней уравнения (один действительный и четыре комплексных с мнимой частью) без установки интервала (рис. 4.6) или первого предположения (рис. 4.7). Кроме того, если численная математика при выводе ответа "на печать", как мы уже отметили, по умолчанию ограничивается тремя знаками после десятичной точки, то символьный оператор `solve` в этом случае выдал численные решения с двадцатью знаками после запятой. При "численном" ответе количество значащих цифр можно увеличить до 15, а при символьном до 250.

Примечание. Лишить переменную ее численного значения для последующих аналитических преобразований можно операторами: `clearsym(a)` (Mathcad Prime) или `a := a` (Mathcad 15).

Если наше уравнение с числовыми константами (см. рис. 4.9) и дальше усложнять, то на каком-то этапе оператор `solve` не сможет найти корни. Функция же `root` по-прежнему будет выдавать корень, правда, лишь один из многих и с опорой на заданный интервал поиска (рис. 4.6) или на первое приближение (рис. 4.7). При этом задавать интервал поиска придется, исходя из уверенности, что корень на этом интервале имеется. Метод секущих же при неправильном первом приближении вообще не выдаст нужного результата. Это такой своеобразный компромисс. Отсюда общее правило: поставленную математическую задачу нужно стараться сначала решить аналитически в общем виде, не придавая переменным конкретных числовых значений (рис. 1) или придавая отдельным или всем переменным числовые значения (рис. 4.9). Если же ответа найти не получается, то придется переходить к поиску частных решений численными методами, дополняя их анализом графиков.

На рисунке 4.10 показано использование графика и функции `root` в двух ее вариантах для решения нашей задачи о моторной лодке. Интересный факт. Двухаргументная функция `root` при первом приближении, равном нулю, выдала не ожидаемый положительный, а неожиданный отрицательный корень. Этот нюанс можно понять, если опять же учесть особенности метода секущих при поиске нулей функции и после построения графика не на отрезке от -3 до 3 км/ч, а на отрезке -13 до 13 км/ч, охватывающем точки разрыва, что мы сделаем ниже. Дело в том, что функция `root` с двумя аргументами работает так. Пользователь задает одну опорную точку поиска (первое предположение, но это далеко не всегда первое приближение — см. пример на рисунке 4.8). Далее пакет Mathcad **правее** этой точки на расстоянии `STOL` (по умолчанию это 0.001 в нашем случае метров, т.е. один миллиметр) фиксирует вторую точку и проводит через две эти точки секущую, которая будет близка к касательной). Эта секущая где-то

пересекает ось X — это будет третьей, очередной точкой итерационного поиска корня. Далее реализуется классический метод секущих. Правая точка разрыва нашей анализируемой функции "перекидывает" поиск в область отрицательных значений. Можно сказать, что в функцию `root` с двумя аргументами заложен гибрид метода Ньютона (касательных) и метода секущих. На авторском сайте <http://communities.ptc.com/videos/1411> можно видеть анимацию метода Ньютона для одиночного уравнения, а на сайте <http://communities.ptc.com/videos/1472> для системы двух уравнений.

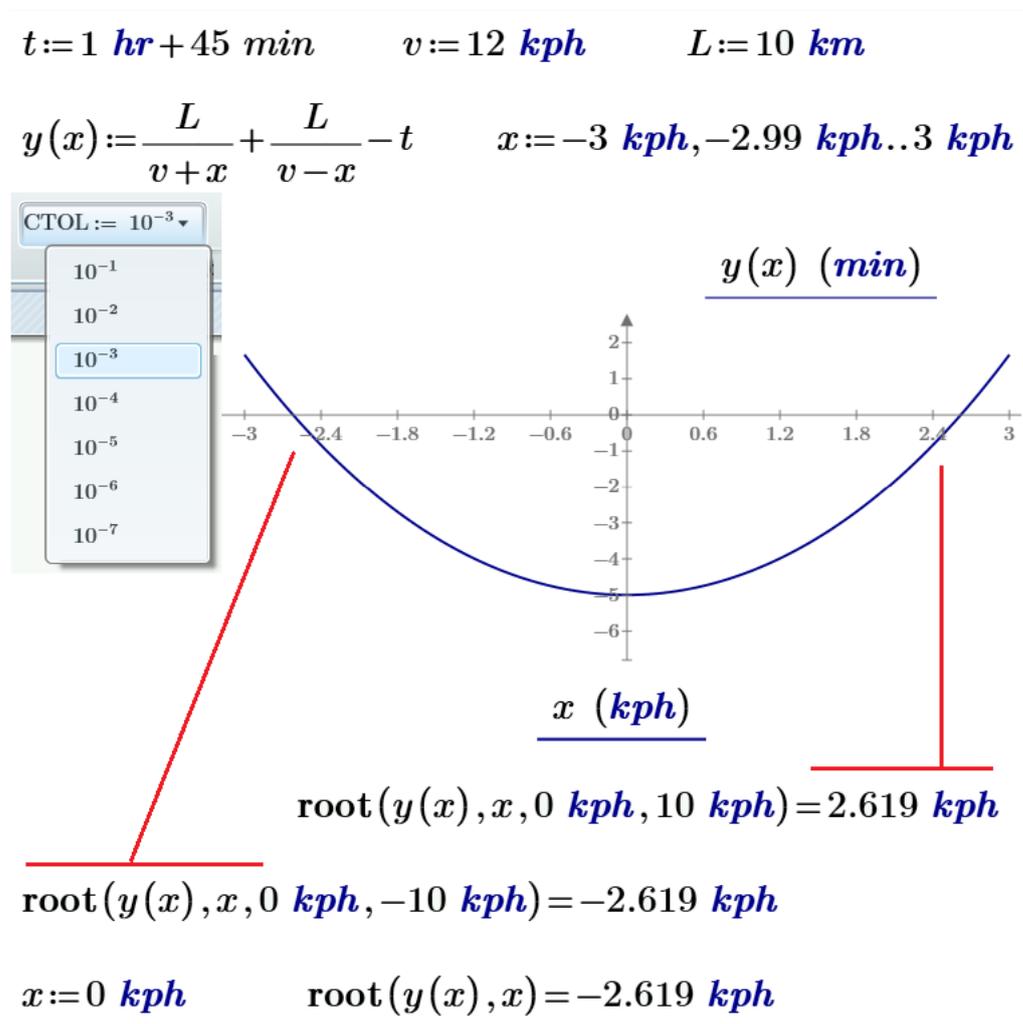


Рис. 4.10. Графическое и численное (через функцию `root`) решение задачи о моторной лодке

Примечание. Для повышения точности поиска нуля функции с помощью встроенной в Mathcad функции `root` можно не уменьшать значение системной переменной `STOL` (см. список левее графика на рис 4.10), а умножить анализируемую функцию на 10^3 , 10^4 и т.д. Выбор точности вычислений — это отдельная задача. С одной стороны,

повышенная точность никогда не будет лишней, а с другой стороны, она приводит к замедлению расчетов и срыву их в ряде случаев. Во всем нужна мера!

Уравнение движения моторной лодки на рисунке 4.1 можно преобразовать в квадратное. К такому приему обычно прибегают в школах, т.к. школьников, как правило, учат решать аналитически только квадратные уравнения — см. текст у рис. 5.13 в этюде 5. Как такое преобразование можно сделать в среде Mathcad, показано на рис. 4.11.

The screenshot shows the Mathcad interface with the following content:

$$\frac{L}{v+x} + \frac{L}{v-x} = t \xrightarrow{\text{simplify}} \frac{2 \cdot L \cdot v}{v^2 - x^2} = t$$

$$2 \cdot L \cdot v - t \cdot (v^2 - x^2) \xrightarrow{\text{coeffs}, x} \begin{bmatrix} 2 \cdot L \cdot v - t \cdot v^2 \\ 0 \\ t \end{bmatrix}$$

$$t \cdot x^2 + 0 \cdot x + (2 \cdot L \cdot v - t \cdot v^2) = 0 \quad a \cdot x^2 + b \cdot x + c = 0$$

The interface includes a top toolbar with icons for 'Удалить область', 'Операторы', 'Символы', 'Программирование', 'Константы', and 'Символьные операции'. A dropdown menu is open under 'Операторы', showing 'Ключевые слова' such as 'assume', 'cauchy', 'coeffs', 'collect', 'combine', 'factor', 'float', 'fourier', 'invlaplace', 'invztrans', 'laplace', 'rewrite', and 'simplify'. The 'coeffs' operator is highlighted, with a tooltip that reads: 'Вернуть коэффициенты полинома. Чтобы открыть справку, нажмите клавишу F1.' Below this, a 'Символьные' (Symbolic) window is open, showing a grid of operators including 'float', 'solve', 'factor', 'collect', 'rectangular', 'simplify', 'expand', 'series', and 'Modifiers'. The 'coeffs' operator is selected in this window, and a tooltip for it reads: 'Полиномиальные коэффициенты'.

Рис. 4.11. Определение коэффициентов полинома

Оператор символической математики `simplify` (упростить) приводит левую часть исходного выражения к общему знаменателю, умножает обе части уравнения на

полученный знаменатель и переносит все слагаемые в левую часть уравнения (рис. 4.11). Таким способом выделяется функция, которая приравнена к нулю. Оператор `coeffs` находит коэффициенты этой функции-полинома (в данном случае квадратного). Это квадратное уравнение можно решить оператором `solve`, но... см. ниже.

Примечание. Квадратичная функция, полученная после преобразования исходного уравнения движения моторной лодки, не равна исходной функции, а только имеет с ней два общих корня. В этом можно убедиться, взглянув на графики, показанные на рис. 4.12. Поскольку по условию задачи $v = 12$ km/h , а уравнение движения не имеет смысла при $x = \pm v$, то функция, стоящая в левой части уравнения, имеет разрывы при x , равном 12 и минус 12 km/h . Квадратичная же функция таких разрывов, естественно, не имеет.

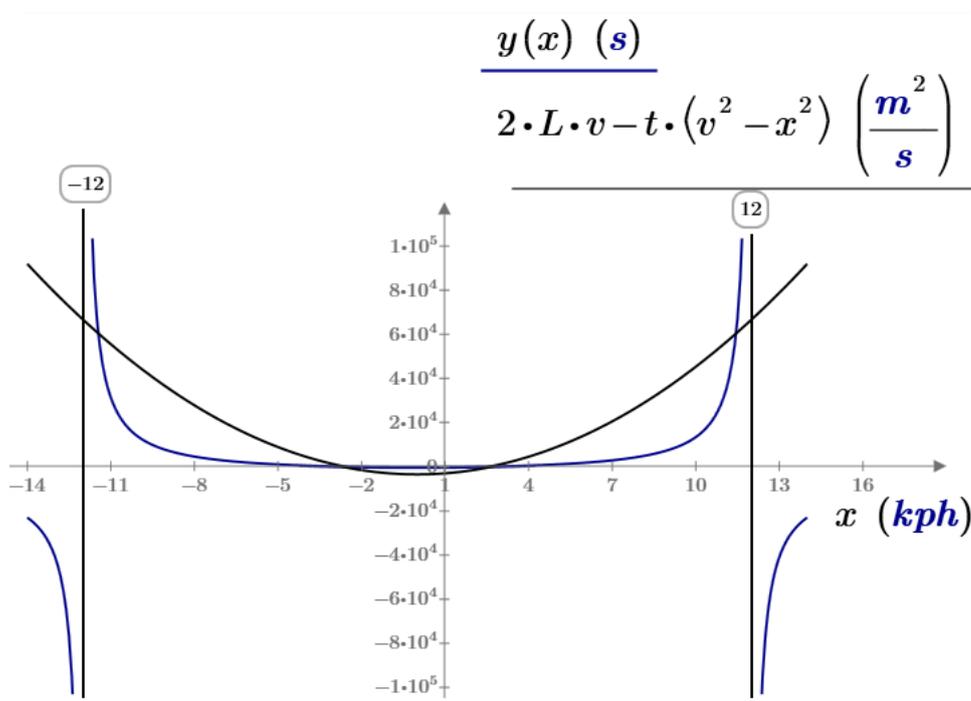


Рис. 4.12. Исходное и квадратное уравнение движения моторной лодки

2. polyroots

Если выражение представляет собой полином (квадратный, например, см. выше), то можно найти все его нули, используя еще одну функцию из "великолепной семерки Mathcad" — функцию `polyroots`. Она имеет в качестве аргумента вектор коэффициентов полинома и возвращает его нули (вектор, который на один элемент короче вектора-аргумента), т.е. решение нашей задачи — см. рис. 4.13.

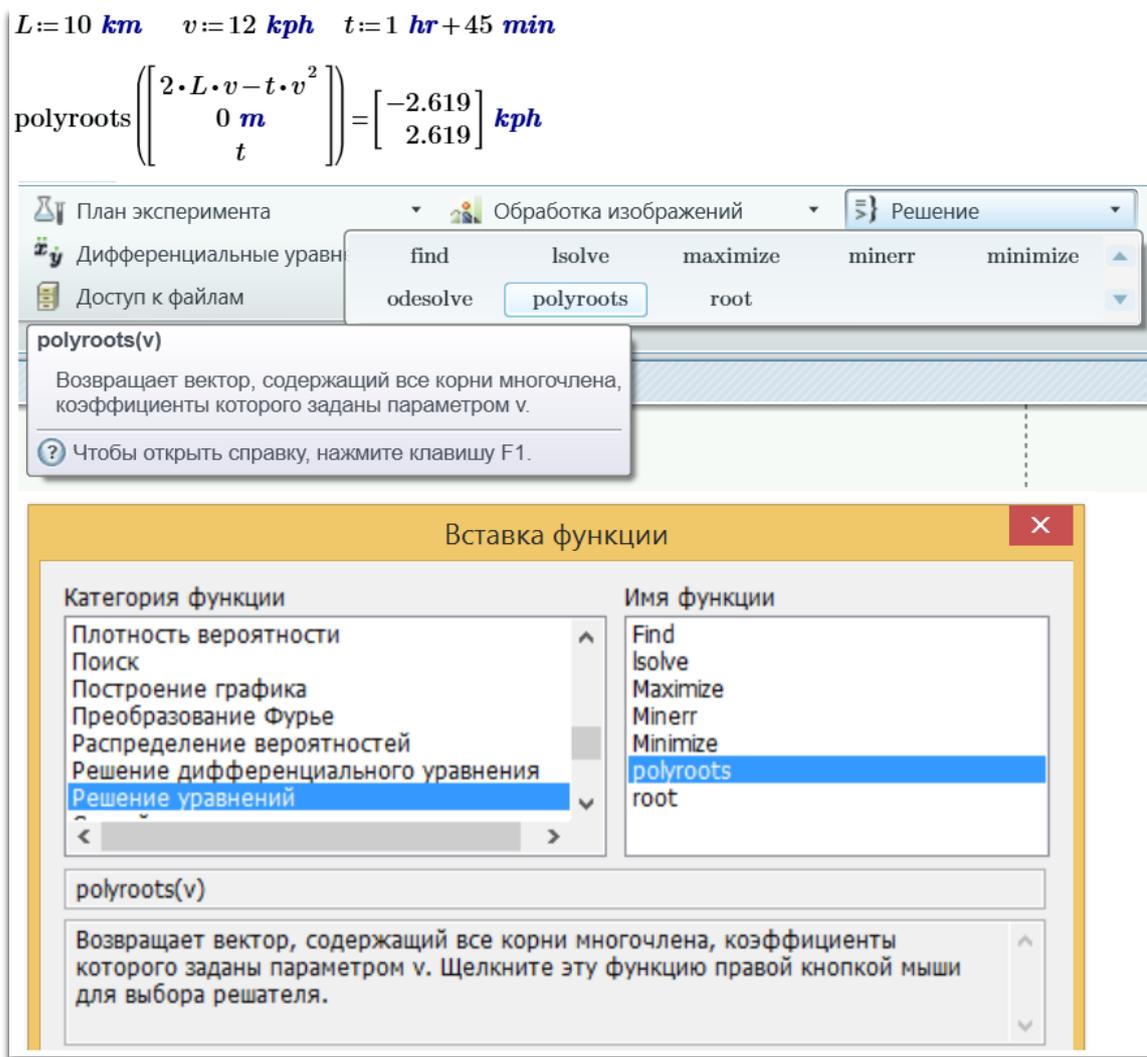


Рис. 4.13. Поиск нулей полинома в среде Mathcad

В нашей задаче о движении моторной лодки полином оказался квадратным и его корни, повторяем, можно было найти через оператор символической математики `solve` (см. рис. 4.1). Но в случае полиномов высокой степени оператор `solve` не работает. Вот тут-то и пригодится численная встроенная функция `polyroots`.

3. Find

Показать работу еще одной функции из "великолепной семерки Mathcad" — функции `Find` — поможет нам еще одна дополнительная моторная лодка.

Задача 2. От двух пристаней на прямолинейном участке реки одновременно отходят две моторные лодки и плывут навстречу друг другу. Они встречаются в точке, делящей этот участок реки в соотношении золотого сечения. Найти скорость второй лодки v_2 и скорость течения воды в реке v , если известна скорость первой лодки v_1 , расстояние между

Этюд 4

пристанями L и время t движения лодок до встречи. (Шуточный вариант задачи: от двух станций по однопутной дороге вышли навстречу друг другу два поезда. И не столкнулись! Почему? Ответ: не судьба!).

Мы в этой задаче используем знаменитое "Золотое сечение", т.е. такое деление отрезка на две неравные части, при котором длина меньшей части отрезка так относится к длине большей части, как длина большей части относится к длине всего отрезка (см. рис. 4.14). Это свойство золотого сечения помнят многие, чего не скажешь о формуле золотого сечения. На сайте <http://communities.ptc.com/videos/1521> показана авторская анимация метода золотого сечения при численном поиске на заданном отрезке минимума функции одного аргумента.

Золотое сечение в задачу вставлено неслучайно. Можно поискать в своей памяти или в справочниках (бумажных или интернетовских) формулу золотого сечения. Но можно поступить иначе [32] — вспомним основную идею этюда 1: написать в среде Mathcad само уравнение золотого сечения применительно к нашей задаче о моторных лодках и решить его аналитически, получив нужную формулу — см. рис. 4.14.

$$\frac{a}{L-a} = \frac{L-a}{L} \xrightarrow{\text{solve, } a} \begin{bmatrix} \frac{L \cdot (\sqrt{5} + 3)}{2} \\ -\frac{L \cdot (\sqrt{5} - 3)}{2} \end{bmatrix} \quad L := 10 \text{ km} \quad \begin{bmatrix} \frac{L \cdot (\sqrt{5} + 3)}{2} \\ \frac{L \cdot (3 - \sqrt{5})}{2} \end{bmatrix} = \begin{bmatrix} 26.18 \\ 3.82 \end{bmatrix} \text{ km}$$

Рис. 4.14. Решение уравнения золотого сечения в среде Mathcad

На рисунке 4.14 оператор `solve` выдал два решения, из которых нам подходит только второе — 3.82 km. Первое же решение (26.18 km) лежит вне рассматриваемого отрезка. Символьная математика, повторяем, выдает все ответы, из которых нужно еще уметь выбрать подходящее. Или уметь заставить оператор `solve` выдать только один нужный ответ.

На рисунке 4.15 показано решение в среде Mathcad Prime и Mathcad 15 задачи о двух моторных лодках, сводящееся к решению системы двух уравнений с двумя неизвестными. Решение найдено с помощью функции `Find`, требующей начального приближения к решению.

L	v_1	t
(km)	(kph)	(min)
10	12	30

Решить

Начальные приближения

$$v_2 := 7 \text{ kph} \quad v := 1 \text{ kph}$$

Ограничения

$$t \cdot (v_1 - v) = \frac{L \cdot (3 - \sqrt{5})}{2}$$

$$t \cdot (v_2 + v) = L - \frac{L \cdot (3 - \sqrt{5})}{2}$$

Решатель

$$\text{Find}(v_2, v) = \begin{bmatrix} 8 \\ 4.361 \end{bmatrix} \text{ kph}$$

$L := 10 \text{ km} \quad v_1 := 12 \text{ kph} \quad t := 30 \text{ min}$

Given

$$v_2 := 7 \text{ kph} \quad v := 1 \text{ kph}$$

$$t \cdot (v_1 - v) = \frac{L \cdot (3 - \sqrt{5})}{2}$$

$$t \cdot (v_2 + v) = L - \frac{L \cdot (3 - \sqrt{5})}{2}$$

$$\text{Find}(v_2, v) = \begin{bmatrix} 8 \\ 4.361 \end{bmatrix} \text{ kph}$$

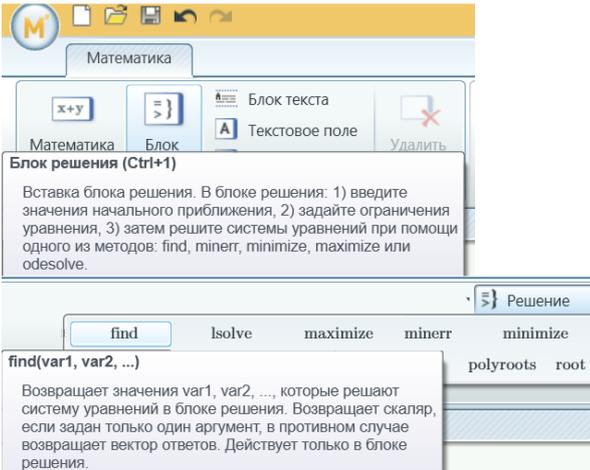


Рис. 4.15. Решение систем уравнений с помощью функции Find

Встроенная функция Find меняет значение своих аргументов, начиная от начального приближения/предположения так, чтобы уравнения системы превратились в тождества. Вернее, почти в тождества. Дело в том, что и обе функции root (рис. 4.7 и 4.9) и функция Find (рис. 4.7) возвращают значения, отличающиеся от точных решений на величину, не превышающую по модулю значения, хранящегося в системной переменной `STOL`. Ведь что такое корень уравнения?! Корень — это значение переменной, при котором уравнение превращается в тождество. Но при численном (приближенном!) решении найти точный корень не всегда удастся. Да и время тратить на это не целесообразно. Подстановка приближенного значения корня в уравнение приводит к тому, что правые и левые части уравнения отличаются друг от друга на значение, хранимое в переменной `STOL`, которое по умолчанию равно 0.001. Это значение можно менять, решая конкретную задачу. На сайте с авторской анимацией <http://communities.ptc.com/videos/1472> можно видеть особенности поиска четырех корней системы двух нелинейных уравнений: уравнения эллипса и уравнения лемнискаты Бернулли (см. также рис. 13.14 из этюда 13). На сайте

<http://communities.ptc.com/videos/2418> можно увидеть анимацию, показывающую, как задание первого приближения влияет выбор конкретного корня из нескольких существующих. Более подробно о методах решения, заложенных в функцию Find, можно почитать в работе [42].

Примечание. В среде Mathcad Prime по сравнению с Mathcad 15 существенно изменилась технология решения уравнений с помощью функции Find. В среде Mathcad Prime не нужно больше вводить ключевое слово Given. Достаточно ввести область Решить с тремя подобластями. От ключевого слова Given отказались в том числе и потому, что многие пользователи после этого слова нажимали клавишу пробела, превращали тем самым это ключевое слово в комментарий и... не понимали, почему Mathcad отказывается решать систему уравнений. На рис. 4.15 показаны для сравнения решения в обеих версиях Mathcad.

Примечание. Переменная L , которой на рис. 4.15 присваивается начальное значение 10 km ($L := 10 \text{ km}$), подчеркнута волнистой чертой, указывающей на некую ненормальную расчетную ситуацию в среде Mathcad 15. Переменная L по умолчанию в среде Mathcad 15 хранит значение одного литра (единица вместимости) и это значение пользователь переопределяет. В среде Mathcad Prime эта недоработка (неудобство) исправлена — там можно иметь две независимые переменные L : единицу вместимости и отдельную пользовательскую переменную, хранящую, как в нашем случае, расстояние.

4. Isolve

Можно понять, что система двух алгебраических уравнений движения двух моторных лодок навстречу друг другу, показанная на рис. 4.15, *линейна*, и применить к ней еще одну функцию из "великолепной семерки Mathcad" — функцию Isolve, предназначенную для решения (solve) именно систем линейных (l) алгебраических уравнений (СЛАУ) — см. рис. 4.16.

L	v_1	t
(km)	(kph)	(min)
10	12	30

$$0 \cdot v_2 - t \cdot v = \frac{L \cdot (3 - \sqrt{5})}{2} - t \cdot v_1$$

$$t \cdot v_2 + t \cdot v = L - \frac{L \cdot (3 - \sqrt{5})}{2}$$

$$M := \begin{bmatrix} 0 & -t \\ t & t \end{bmatrix} \quad V := \begin{bmatrix} \frac{L \cdot (3 - \sqrt{5})}{2} - t \cdot v_1 \\ L - \frac{L \cdot (3 - \sqrt{5})}{2} \end{bmatrix}$$

$$\begin{bmatrix} v_2 \\ v \end{bmatrix} := \text{Isolve}(M, V) = \begin{bmatrix} 8 \\ 4.361 \end{bmatrix} \text{ kph}$$

Рис. 4.16. Решение СЛАУ в среде Mathcad

На рисунке 4.16 система уравнений, идентичная системе с рис. 4.15, с помощью тождественных алгебраических преобразований приведена к виду классической линейной системы: слева неизвестные v_2 и v со своими коэффициентами, справа — свободные члены. Функция `lsolve` имеет два аргумента: матрицу коэффициентов при неизвестных СЛАУ (у нас это M) и вектор свободных членов v . Возвращает функция `lsolve` вектор найденных значений неизвестных. При решении СЛАУ с помощью функции `lsolve` (рис. 4.16) начальные приближения (см. рис. 4.15) вводить не надо, т.к. у этой системы либо есть, причем единственное решение, либо решений совсем нет, либо решений бесконечно много. В [32] дан графический анализ этой особенности с привлечением понятия ранга матрицы.

Примечание. Вторым аргументом функции `lsolve` может быть не только квадратная (классический случай СЛАУ), но и прямоугольная матрица, отображающая недоопределенную или переопределенную системы.