5 и 6. Minimize&Maximize

Об очередной функции "великолепной семерки" — о функции `Minimize`, будет рассказано на примере *задачи оптимизации*, связанной также с "водным транспортом".

Задача 3. Определить крейсерскую скорость судна — скорость при которой затраты на его эксплуатацию будут минимальны.

Задача предельно упрощена — затраты на эксплуатацию судна состоят из двух частей: почасовой зарплаты экипажа, пропорциональной времени движения судна (обратно пропорциональной скорости судна), и затрат на горючее, пропорциональных квадрату скорости судна (коэффициенты пропорциональности — `a` и `b`). Увеличивая скорость судна, мы экономим на зарплате экипажа, но при этом приходится больше тратить денег на горючее. Попробуем найти тут оптимальное решение!

На рисунке 4.17 показано решение этой типичной задачи оптимизации с помощью встроенной функции `Minimize` с графической иллюстрацией решения.

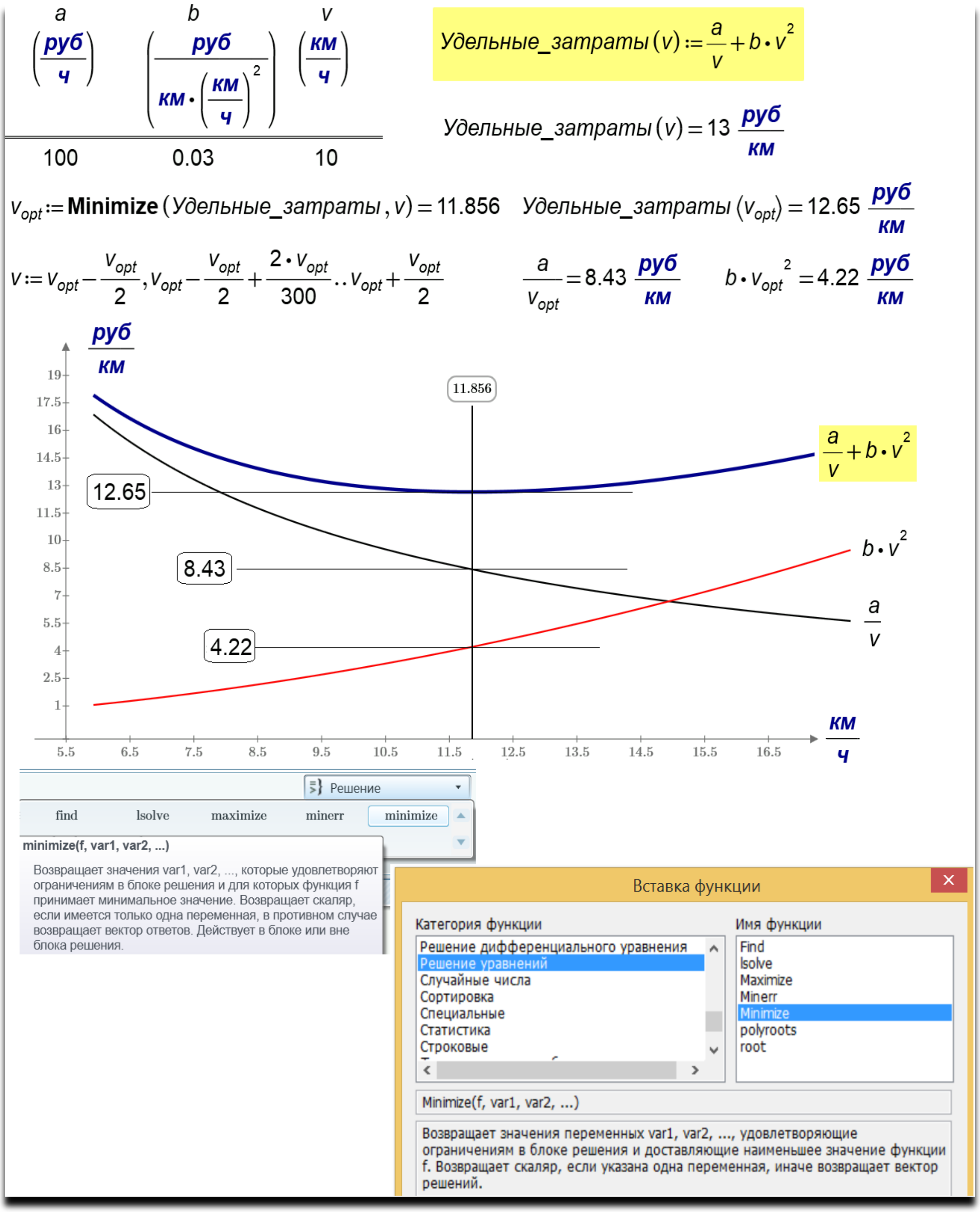

Рис. 4.17. Нахождение крейсерской скорости судна численной математикой Mathcad

Функция `Minimize` меняет значение своего второго аргумента, начиная от заданного предполагаемого значения (у нас это 10 км/ч) так, чтобы значение первого аргумента (целевой функции `Удельные_затраты`) приняло минимальное значение. Если бы мы не минимизировали затраты, а максимизировали, например, прибыль владельца судна, то нужно было бы при решении такой задачи функцию `Minimize` заменить на функцию `Maximize`. В оптимизационных задачах часто присутствуют ограничения —

скорость судна, например, не может превышать максимально допустимую. В этом случае функции `Minimize` или `Maximize` нужно будет поместить в область `Ограничения` блока `Решить`, показанного на рис. 4.15.

Найти минимум нашей целевой функции `Удельные_затраты` можно и средствами символьной математики Mathcad, что показано на рис. 4.18.

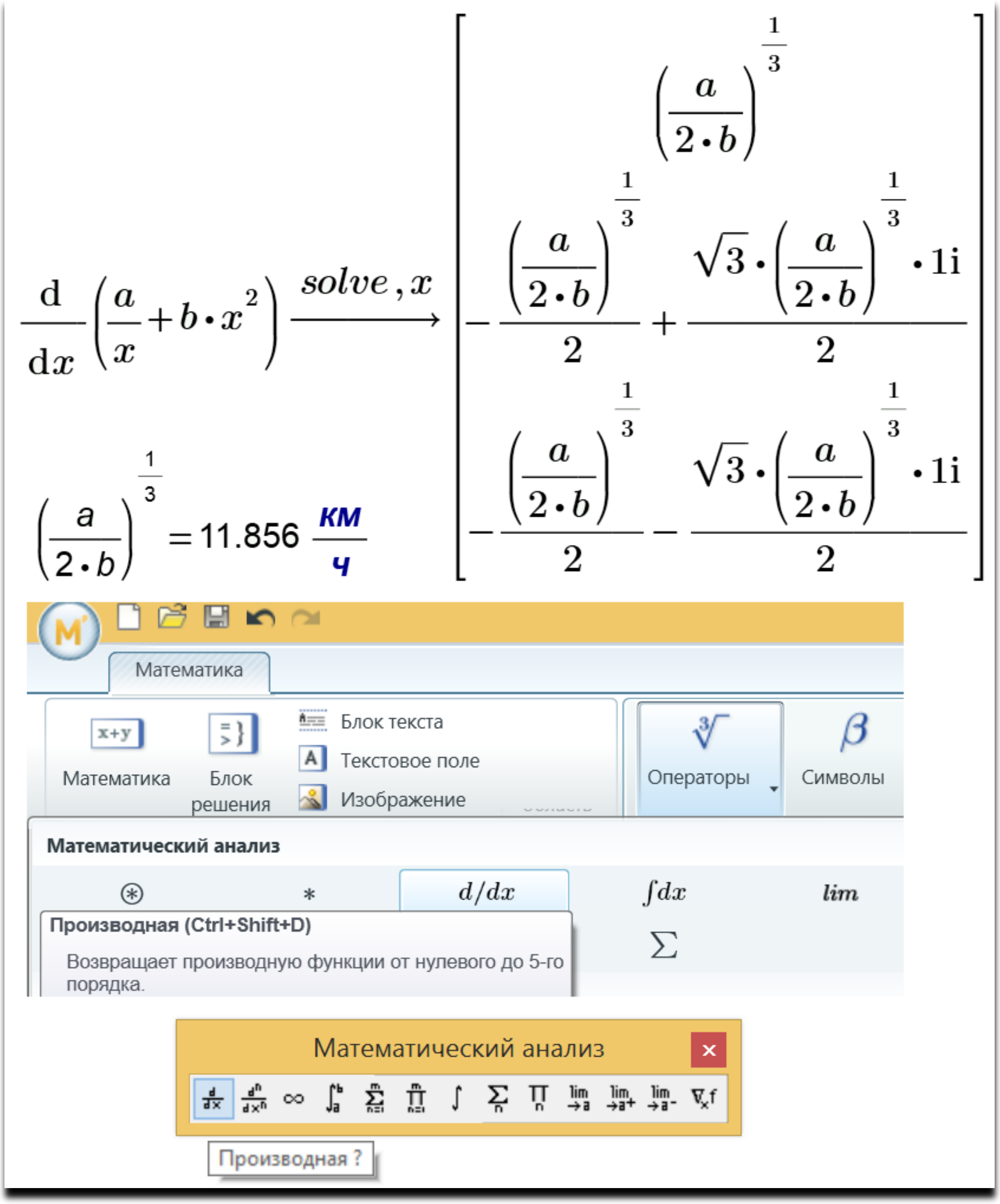

Рис. 4.18. Нахождение крейсерской скорости судна символьной математикой Mathcad

На рисунке 4.18 ведется поиск нулей первой производной функции по удельным затратам на километр пути судна. Но если затраты на топливо будут зависеть от скорости судна, взятой не во второй степени, а в степени `n` (этот коэффициент, близкий к двойке, уточняют экспериментально) то символьная математика уже не справится с такой усложненной задачей (рис. 4.19), и придется вернуться к численным методам решения задач (рис. 4.17).

$$\boxed{\frac{d}{dx}\left(\frac{a}{x}+b\cdot x^{n}\right)} \xrightarrow{solve\,,x} ?$$

Символьный результат не найден.

Рис. 4.19. Осечка при работе с символьной математикой Mathcad

7. Minerr

Последняя[1] функция "великолепной семерки Mathcad" — это функция `Minerr` (Minimal Error — минимальная ошибка). Если функция `Find` (см. рис. 4.15) не находит решения системы уравнений, то она возвращает сообщение об ошибке. Функция же `Minerr` в такой ситуации возвратит не сообщение об ошибке, а значения своих аргументов (невязку системы), при которых система уравнений будет максимально приближена к системе тождеств — точку последнего приближения к решению. В старых версиях Mathcad не было функций `Minimize` и `Maximize`, и задачи оптимизации приходилось решать именно через функцию `Minerr`. Попробуем и мы поступить так. На рисунке 4.20 показано, как эта функция решает задачу крейсерской скорости судна: при оптимальном движении затраты на эксплуатацию судна будут максимально приближены к нулю (мечта всех судовладельцев).

[1] Последняя в списке, но не последняя по важности – смотрим далее.

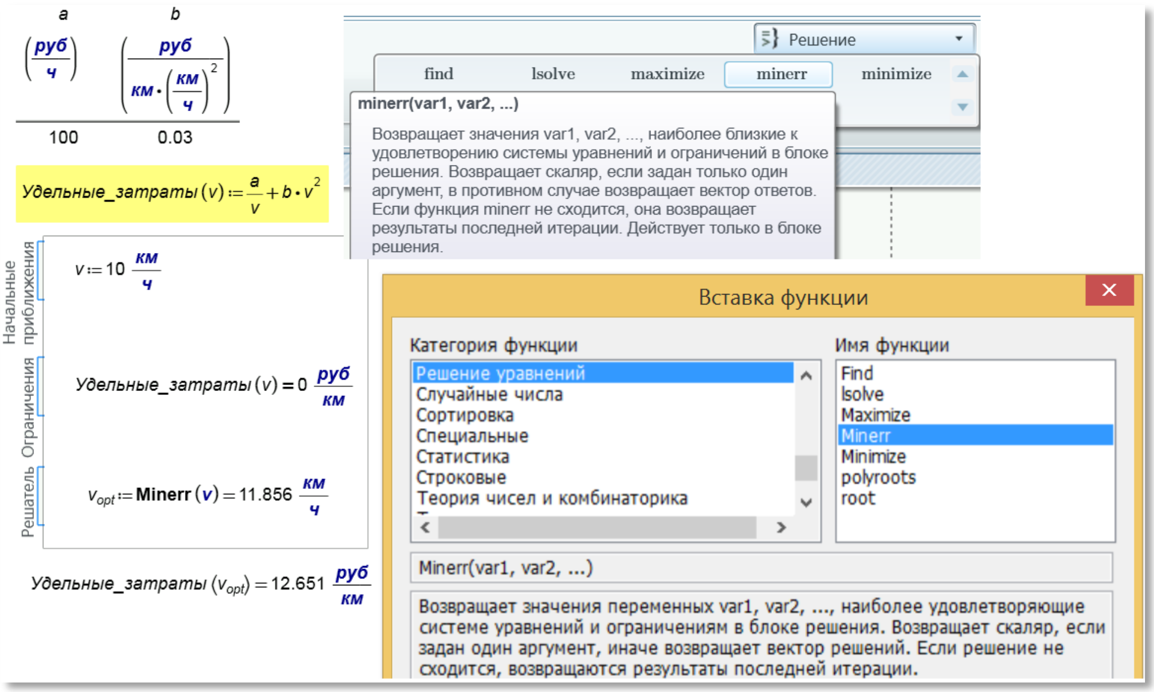

Рис. 4.20. Решение задачи оптимизации с помощью функции `Minerr`

Функция `Minerr` поможет нам в этюде 8 сделать анимацию правила Чебышева (см. рис. 8.8 и 8.9) при укороченном одном стержне.

Функцию `Minerr` можно считать ***главной*** в "великолепной семерке Mathcad", т.к. ею можно заменить и функцию `Find`, и функцию `root` (в двух ее вариантах), и функцию `polyroots`, и функцию `lsolve`, и в ряде случаев функции `Minimize` и `Maximize`. При использовании функции `Minerr` надо обязательно предусматривать проверку решений. Нередки случаи, когда решения могут оказаться ошибочными, чаще всего из-за того, что из нескольких корней находится нереальный (или не представляющий интереса) корень. Дело в том, что функция `Minerr` пытается найти максимальное приближение к искомому числу путем минимизации среднеквадратической погрешности решения. Следует заранее убедиться в том, что решение существует, и как можно точнее указать начальное приближение к решению.

С другой стороны, главной (незаменимой) функцией "великолепной семерки Mathcad" можно считать функцию `root`, т.к. она одна остается активированной при окончании месячного пробного использования Mathcad Prime и при переходе к Mathcad Express.

Компьютерная математика с универсальными и скрытыми от пользователей методами аналитических и численных решений заставляет нас забывать о типах уравнений. Вспомним о них!