Оптимизированный фрактал или ФМИ

Валерий Очков

В [1] было отмечено, что в настоящее время в образовательном процессе намечается тенденция одновременного чтения курса информатики и математики. В рамках этой учебной дисциплины (условное название ИнфоМатика или МатИнформика) можно также затрагивать вопросы и учебных курсов Физика и/или Введение в специальность. Тем более в истории есть прецеденты: в тридцатые годы прошлого столетия, например, выдающийся ученый Норберт Винер (отец кибернетики – см. сноску 1) читал в MIT годовой объединенный курс лекций по математике и электротехнике.

На начальном (переходном) этапе можно так кардинально не менять содержание лекций по математике и информатике, объединяя их в одну, а поступить по-иному. Допустимо на первой паре лекционных занятий прочесть студентам классическую математическую лекцию по анализу функции одного или нескольких аргумента, а затем на второй паре – лекцию в рамках занятий по Информатике о том, как Интернет и современные математические программы (системы компьютерной математики) могут на примере решения конкретной практической задачи работать с производными, интегралами, рядами и прочими математическими понятиями, о которых было рассказано на предыдущей лекции. В такую связку можно вклинить и лекцию по физике. Получится очень интересная образовательная технология в рамках дисциплины с условным названием ФизМатИнформика (ФМИ): рассматриваются теоретические аспекты некоторого процесса (математика), далее ставится эксперимент и дается его толкование (физика), а в завершении создается, реализуется (аналитически и/или численно) и анимируется математическая модель процесса.

В [1] было дано конкретное содержание одной такой лекции, связанной с цепной функцией. А вот содержание другой подобной лекции или серии лекций, охватывающей математику и информатику.

Есть такая "народная" задача оптимизации. Народная, в том смысле, что она широко "гуляет" по Интернету, но автор ее неизвестен. В этом несложно убедиться, если в ком-либо поисковике сделать запрос по ключу "Коробка максимального объема". Суть задачи. Берется квадратный лист бумаги (картона, жести и т.д.), в углах которого вырезаются четыре одинаковых квадрата меньшего, естественно, размера – см. первый кадр анимации на рис. 1. Далее из такой крестообразной заготовки складывается коробка загибанием прямоугольных участков вверх – см. остальные три кадра анимации на рисунке 1. Эта анимация превращения квадратной заготовки в коробку сделана в среде Mathcad 15. Ее можно просмотреть, зайдя на сайт статьи по адресу https://www.ptcusercommunity.com/message/430986. Там же лежат все Mathcad-файлы, представленные ниже, а также дополнительные рисунки и файлы.

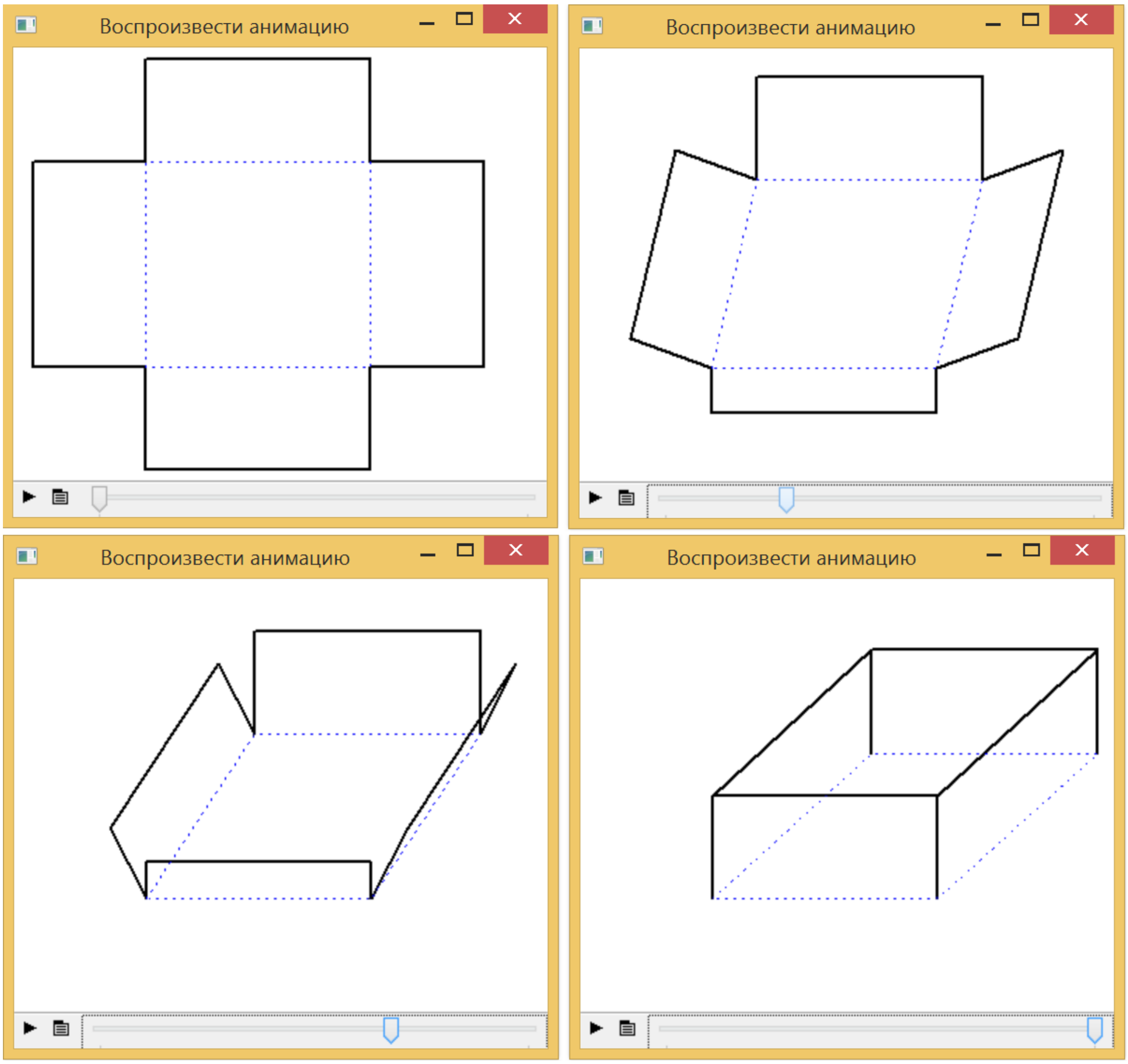


Рис. 1. Кадры анимации изготовления коробки из квадратной заготовки

Анимация, четыре кадра которой показаны на рис. 1, делалась так: в расчет вводились векторы, хранящие координаты вершин двенадцати прямых углов крестообразной заготовки (первый кадр анимации) и этих же точек на свернутой коробке (последний кадр). Затем значения координат вершин углов заготовки плавно (с изменением значения системной переменной `FRAME` от 0 до 999) менялись до значений координат сложенной коробки [2].

Но вернемся к самой задаче. Спрашивается, какой должен быть размер вырезаемых квадратов, чтобы объем полученной коробки был максимальным?

На рисунке 2 показано решение этой задачи с помощью символьной математики пакета Mathcad: создается функция пользователя с именем `V`, с параметром `A` (длина стороны исходной квадратной заготовки) и аргументом `x` (длина стороны вырезаемых четырех квадратов). Функция `V` возвращает объем свернутой коробки – произведение площади ее основания $(A - 2x)^2$ на высоту `x`. От этой функции, представляющей собой кубический полином, берется первая производная по `x`. Получается квадратный полином, у которой ищутся нули Это все можно сделать и в уме, расписывая ход решения на бумаге, но мы все чаще и чаще делаем это на компьютере. Нулей у квадратного полинома два: `A/2` – локальный минимум функции `V` (нулевой объем коробки – коробка вырождается в отрезок прямой) и `A/6` – искомый максимум функции `V`. Это и есть решение нашей "народной" задачи оптимизации раскроя коробки: длина стороны вырезаемых квадратов должна равняться одной шестой длины стороны исходного квадрата. На рисунке 2 это решение дополнительно проверяется графически для коробки, сделанной из квадратной заготовки со стороной, длина которой равна одному метру. Из такой "метровой" заготовки можно сделать коробку, максимальный объем которой составит 74.071 литра[1].

$$V_1(A,x):=(A-2\,x)^2\cdot x \qquad \frac{\mathrm{d}}{\mathrm{d}x}V_1(A,x)=0 \xrightarrow{solve,\,x} \begin{bmatrix} \frac{A}{2} \\ \frac{A}{6} \end{bmatrix} \qquad V_1\left(1\ m, \frac{1}{6}\ m\right)=74.074\ L$$

$$x:=0\ m, 1\ cm..0.5\ m$$

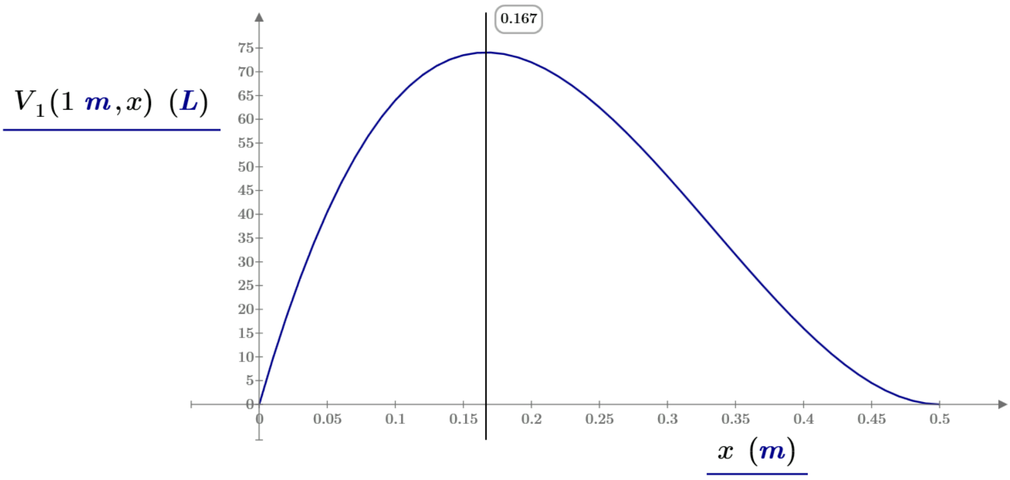


Рис. 2. Решение задачи о коробке максимального объема (Mathcad Prime)

Показанное на рисунке 2 решение, повторяем, можно найти во многих бумажных и интернетовских источниках. Но задача об оптимальной коробке имеет довольно неожиданное и интересное продолжение [3], которым пренебрегали в докомпьютерную эру.

Можно четыре квадрата, отрезанных от исходной заготовки (см. левый верхний кадр анимации на рис. 1) не выбрасывать, а пустить в дело – сделать из них четыре новые одинаковые коробки меньшего размера по той же схеме раскроя. Из шестнадцати (4·4) новых отрезанных квадратов можно опять же сделать новые коробки. Из шестидесяти четырех (16·4) новых отрезанных квадратов снова же можно сделать еще меньшие новые коробки и т.д. до бесконечности. В математике такие объекты, образуемые повторением одной и той же операции, но на новом уровне, получили название *фракталов*. Если в том же Интернете провести поиск по данному ключевому слову, то можно найти изображения и описания фракталов различной формы, в том числе и состоящих из квадратов или кубов, а также параллелепипедов (коробок) с квадратным основанием уменьшающегося размера, которые мы только-что описали.

[1] Литры к ответу приписал сам пакет Mathcad по умолчанию. Но литры – это единицы *вместимости*, а не объема. Тут литры по идее нужно заменить на единицы длины в кубе (единицы объема), но мы оставили все как есть, понимая, что коробка создается для того, чтобы *вмещать* в себя что-то.

Но наш "коробочный" фрактал можно попытаться *оптимизировать* – определить размеры сторон квадратных вырезок, при которых суммарный объем полученных коробок будет максимальным.

На рисунке 3 показано нахождение оптимального размера сторон четырех квадратов, вырезаемых из исходного квадрата со стороной `A` и из которых вырезаются 16 квадратов меньшего размера. Ясно, что при втором (последнем) шаге раскроя маленьких квадратов нужно сохранить пропорцию 1/6, чтобы суммарный объем четырех коробок малого размера был максимален. Эта пропорция (`x/6`) и зафиксирована в функции пользователя на рис. 3. Тут, как правило, многие полагают, что в пропорции раскроя большой коробки сохранится пропорция 1/6. Но это не так – эта пропорция несколько большей, чем одна шестая.

$$V_5(A,x) := V_1(A,x) + 4 \cdot \left(x - 2\,\frac{x}{6}\right)^2 \cdot \frac{x}{6}$$

$$\frac{d}{dx} V_5(A,x) = 0 \xrightarrow{solve,\,x} \begin{bmatrix} \dfrac{9 \cdot A}{29} + \dfrac{3 \cdot \sqrt{7} \cdot A}{58} \\[2ex] \dfrac{9 \cdot A}{29} - \dfrac{3 \cdot \sqrt{7} \cdot A}{58} \end{bmatrix}$$

$$A := 1\ \mathbf{m} \qquad \begin{bmatrix} \dfrac{9 \cdot A}{29} + \dfrac{3 \cdot \sqrt{7} \cdot A}{58} \\[2ex] \dfrac{9 \cdot A}{29} - \dfrac{3 \cdot \sqrt{7} \cdot A}{58} \end{bmatrix} = \begin{bmatrix} 44.719 \\ 17.35 \end{bmatrix} \mathbf{cm}$$

$$V_5\left(A, \frac{9 \cdot A}{29} - \frac{3 \cdot \sqrt{7} \cdot A}{58}\right) = 75.529\ \mathbf{L}$$

Рис. 3. Задача о пяти коробках – решение 1 (Mathcad Prime)

Максимальный объем пяти коробок стал почти на полтора литра больше максимального объема одной коробки. Прирост небольшой (2%), но мы гонимся не за вместимостью, а за… математикой и ее компьютерными приложениями.

Задачу о пяти коробках – одной большой и четырех маленьких можно решить и по-иному, забыв на время про пропорцию 1/6 (раскрой четырех маленьких квадратов – см. рис. 3) и проанализировав функцию V_5 не одного (рис. 3), а двух аргументов x (пропорция для первого шага раскроя) и y (пропорция для второго шага раскроя – см. рис. 4).

Суммарный объем пяти коробок $V_5(x,y) := (1-2x)^2 \cdot x + 4 \cdot (x - 2y \cdot x)^2 \cdot y \cdot x$

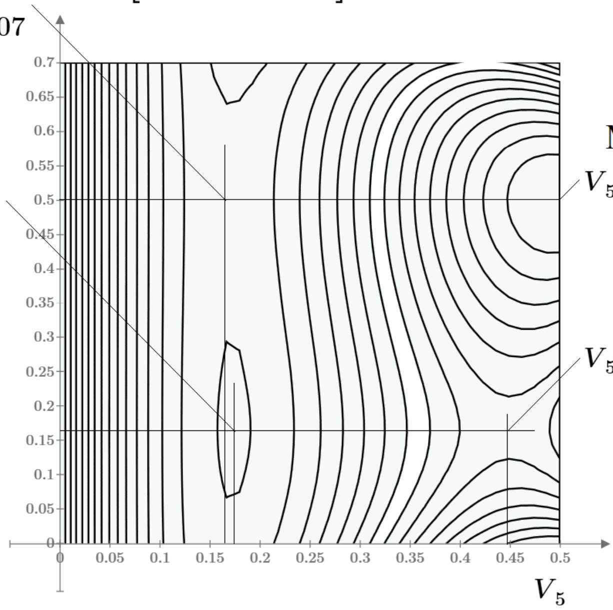
$$a := \begin{bmatrix} \frac{d}{dx} V_5(x,y) = 0 \\ \frac{d}{dy} V_5(x,y) = 0 \end{bmatrix} \xrightarrow{solve, \begin{bmatrix} x \\ y \end{bmatrix}} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} \\ \frac{3 \cdot \sqrt{7}}{58} + \frac{9}{29} & \frac{1}{6} \\ \frac{9}{29} - \frac{3 \cdot \sqrt{7}}{58} & \frac{1}{6} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.1666667 & 0.5 \\ 0.447194 & 0.1666667 \\ 0.1734956 & 0.1666667 \end{bmatrix}$$

Седло

$V_5(a_{2,1}, a_{2,2}) = 0.07407$

Максимум

$V_5(a_{4,1}, a_{4,2}) = 0.07553$



Минимум

$V_5(a_{1,1}, a_{1,2}) = 0$

Седло

$V_5(a_{3,1}, a_{3,2}) = 0.031$

Рис. 4. Задача о пяти коробках – решение 2 (Mathcad 15)

В решение на рисунке 4 вводится функция пользователя V_5 с двумя аргументами[2]: x и y. Далее решается система двух алгебраических уравнений: равенство нулю частных производных[3] функции V_5 по x и по y. Это необходимое условие экстремума для функции двух аргументов. Координаты точек экстремума функции V_5 обязаны удовлетворять этой системе. Далее можно в каждой найденной точке проверить выполнение достаточного условия экстремума. Но можно результат получить не аналитически, а графически – построить линии уровня функции двух аргументов – и см. рис. 4. Полученная система уравнений имеет четыре корня, один из которых

[2] От единиц измерения мы тут и далее отказались, допустив, что A равно просто единице, а не одному метру. Дело в том, что трехмерная графика пакета Mathcad не поддерживает единицы измерения. Хотя единицы измерения тут очень полезны – они позволяют избегать ряда ошибок при формировании функций объема, не позволяя складывать, например, квадратные метры и кубические метры.

[3] В среде Mathcad 15 был символ частной производной (округлое d). В среде Mathcad Prime, где сделан расчет, показанный на рис. 4, его уже нет. И это правильно. Символ частной производной был взят из ручных расчетов, где у функции не показывали аргументы. В среде Mathcad аргументы фиксируются всегда. Поэтому символ частной производной здесь будет лишним.

совпадет с решением, показанным на рис. 3, и является искомой точкой максимума. Три другие корня – это точка минимума (0.5 0.5) и две так называемых "седловые точки", что ясно видно из графиков линий уровня, показанных на рис. 4. Минимумы, максимумы и "седла" можно также увидеть и на поверхности, построенной с помощью авторского Mathcad-сайта – см. рис. 5.

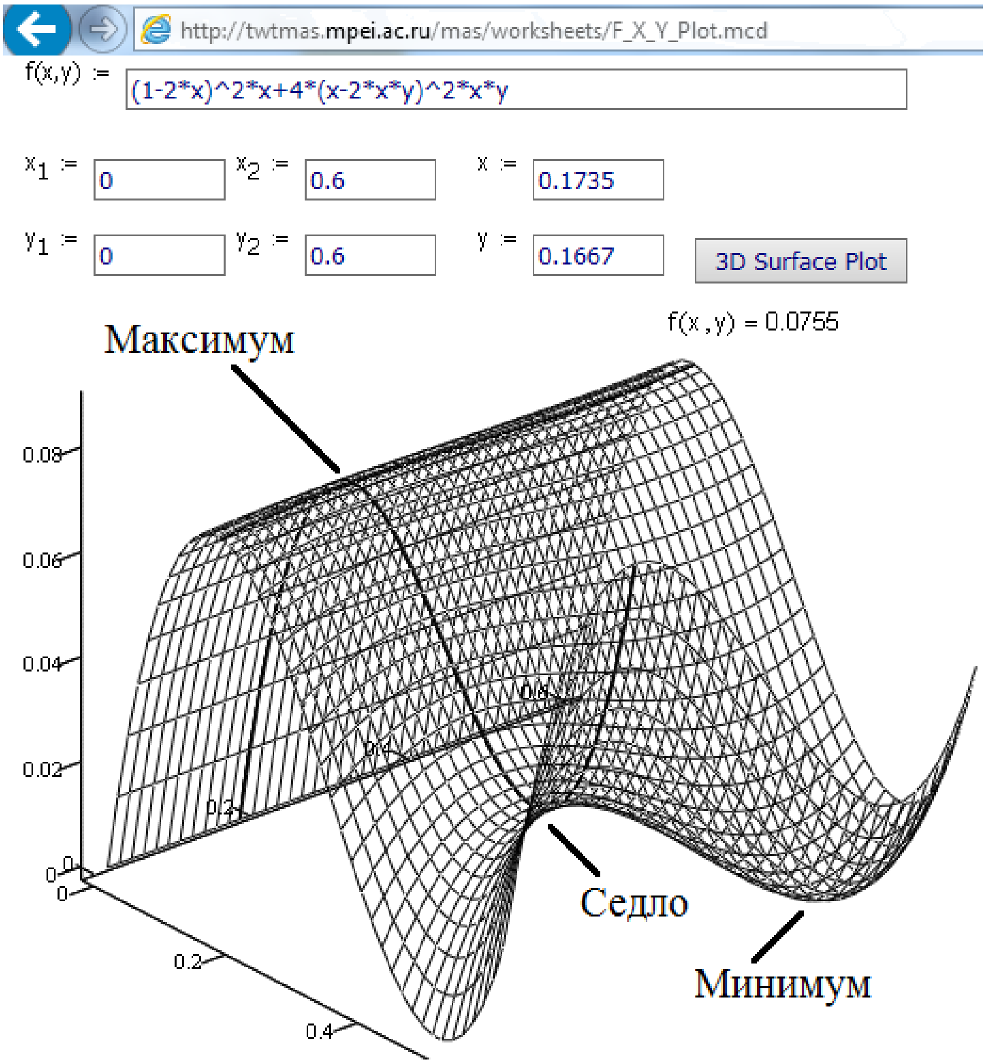
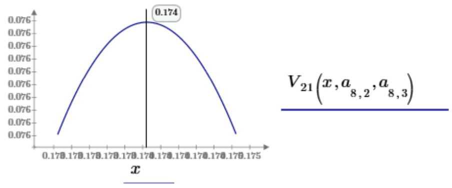


Рис. 5. Построение поверхности функции двух аргументов с помощью Mathcad-севера

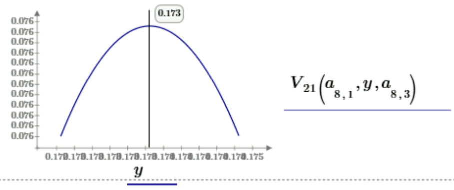
$$V_{21}(x,y,z) := (1-2\,x)^2 \cdot x + 4 \cdot (x - 2\,y \cdot x)^2 \cdot y \cdot x + 4 \cdot 4 \cdot (y \cdot x - 2\,z \cdot y \cdot x)^2 \cdot z \cdot y \cdot x$$

$$a := \begin{bmatrix} \frac{d}{dx} V_{21}(x,y,z) = 0 \\ \frac{d}{dy} V_{21}(x,y,z) = 0 \\ \frac{d}{dz} V_{21}(x,y,z) = 0 \end{bmatrix} \xrightarrow{solve, \begin{bmatrix} x \\ y \\ z \end{bmatrix}} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{2} \\ \frac{3 \cdot \sqrt{7}}{58} + \frac{9}{29} & \frac{1}{6} & \frac{1}{2} \\ \frac{9}{29} - \frac{3 \cdot \sqrt{7}}{58} & \frac{1}{6} & \frac{1}{2} \\ \frac{\sqrt{270207 \cdot \sqrt{7} + 5952978}}{16794} + \frac{7 \cdot \sqrt{7}}{2799} + \frac{886}{2799} & \frac{3 \cdot \sqrt{7}}{58} + \frac{9}{29} & \frac{1}{6} \\ \frac{7 \cdot \sqrt{7}}{2799} - \frac{\sqrt{270207 \cdot \sqrt{7} + 5952978}}{16794} + \frac{886}{2799} & \frac{3 \cdot \sqrt{7}}{58} + \frac{9}{29} & \frac{1}{6} \\ \frac{\sqrt{5952978 - 270207 \cdot \sqrt{7}}}{16794} - \frac{7 \cdot \sqrt{7}}{2799} + \frac{886}{2799} & \frac{9}{29} - \frac{3 \cdot \sqrt{7}}{58} & \frac{1}{6} \\ \frac{886}{2799} - \frac{7 \cdot \sqrt{7}}{2799} - \frac{\sqrt{5952978 - 270207 \cdot \sqrt{7}}}{16794} & \frac{9}{29} - \frac{3 \cdot \sqrt{7}}{58} & \frac{1}{6} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.166666666666667 & 0.5 & 0.5 \\ 0.447194033330927 & 0.166666666666667 & 0.5 \\ 0.173495621841487 & 0.166666666666667 & 0.5 \\ 0.476917060684446 & 0.447194033330927 & 0.166666666666667 \\ 0.169399666130454 & 0.447194033330927 & 0.166666666666667 \\ 0.446204781796901 & 0.173495621841487 & 0.166666666666667 \\ 0.173644979419639 & 0.173495621841487 & 0.166666666666667 \end{bmatrix}$$

$$\Delta := 0.001 \qquad x := a_{8,1} - \Delta, a_{8,1} - \Delta + \frac{\Delta}{300} \,..\, a_{8,1} + \Delta$$



$$y := a_{8,2} - \Delta, a_{8,2} - \Delta + \frac{\Delta}{300} \,..\, a_{8,2} + \Delta$$



$$z := a_{8,3} - \Delta, a_{8,3} - \Delta + \frac{\Delta}{300} \,..\, a_{8,3} + \Delta$$

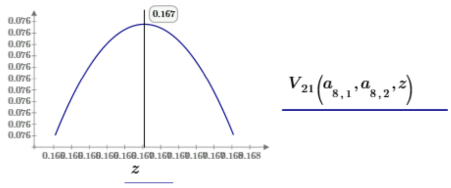


Рис. 5а. Задача о 21 коробках – решение 1 (Mathcad Prime)

$$V_{85}(x,y,z,w):=(1-2\ x)^2\cdot x+4\cdot(x-2\ y\cdot x)^2\cdot y\cdot x+4\cdot 4\cdot(y\cdot x-2\ z\cdot y\cdot x)^2\cdot z\cdot y\cdot x+4\cdot 4\cdot 4\cdot(z\cdot y\cdot x-2\ w\cdot z\cdot y\cdot x)^2\cdot w\cdot z\cdot y\cdot x$$

$$a:=\begin{bmatrix}\frac{d}{dx}V_{85}(x,y,z,w)=0\\ \frac{d}{dy}V_{85}(x,y,z,w)=0\\ \frac{d}{dz}V_{85}(x,y,z,w)=0\\ \frac{d}{dw}V_{85}(x,y,z,w)=0\end{bmatrix}\xrightarrow{solve,\begin{bmatrix}x\\y\\z\\w\end{bmatrix}} ?=\begin{bmatrix}
0.5 & 0.5 & 0.5 & 0.5\\
0.166666666666667 & 0.5 & 0.5 & 0.5\\
0.447194033330927 & 0.166666666666667 & 0.5 & 0.5\\
0.173495621841487 & 0.166666666666667 & 0.5 & 0.5\\
0.476917060684446 & 0.447194033330927 & 0.166666666666667 & 0.5\\
0.169399666130454 & 0.447194033330927 & 0.166666666666667 & 0.5\\
0.446204781796901 & 0.173495621841487 & 0.166666666666667 & 0.5\\
0.173644979419639 & 0.173495621841487 & 0.166666666666667 & 0.5\\
0.489109888173501 & 0.476917060684446 & 0.447194033330927 & 0.166666666666667\\
0.167912869184576 & 0.476917060684446 & 0.447194033330927 & 0.166666666666667\\
0.446787788723599 & 0.169399666130454 & 0.447194033330927 & 0.166666666666667\\
0.47654511533674 & 0.446204781796901 & 0.173495621841487 & 0.166666666666667\\
0.446184107289249 & 0.173644979419639 & 0.173495621841487 & 0.166666666666667\\
0.173556845683853 & 0.169399666130454 & 0.447194033330927 & 0.166666666666667\\
0.169446642317966 & 0.446204781796901 & 0.173495621841487 & 0.166666666666667\\
0.1736481106798 & 0.173644979419639 & 0.173495621841487 & 0.166666666666667
\end{bmatrix}$$

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

$$\sum_{i=0}^{3} 4^i = 85$$

Рис. 5b. Задача о 85 коробках – решение 1 (Mathcad Prime)

На рисунке 5 максимум попадает на почти горизонтальную прямую, и создается впечатление, что точки максимума заполняют всю прямую, прорисованную на вершине этого "хребта". Нужно развернуть график или изменить масштаб по вертикальной оси, чтобы максимум выделялся на общем фоне. У автора есть мечта установить в аудитории лазерный 3D-проектор и показывать поверхности и другие объемные объекты, подобные тому, какой показан на рис. 5, под потолком, поворачивая их по трем осям. Сейчас так проводятся различные эстрадные шоу[4]. Поверхность, показанную на рис. 5, можно также распечатать на 3-D принтере.

Оптимальные пропорции раскроя можно определить не только с помощью символьной, но и с помощью численной математики Mathcad, запрограммировав несложный алгоритм поиска максимума функции – см. рис. 6.

Представьте себе, что вы находитесь у подножия холма и вам нужно забраться на его вершину, которую не видно из-за густого тумана. Вы ощупываетесь вокруг себя – делаете шаги вправо, влево, вперед и назад, а затем переходите в точку, где подъем был максимален. В новой точке вы повторяете эту операцию до тех пор, пока новые шаги не найдут точки подъема. Тогда вы вдвое уменьшаете шаг и повторяете поиск вокруг себя до тех пор, пока ваш шаг не станет меньше наперед заданной величины. Тут вы делаете вывод о том, что вы забрались на вершину холма. Это, наверно, самый простейший алгоритм поиска максимума функции. Но не глобального, а локального максимума – вы можете забраться не на самый высокий холм в гряде холмов, а на

[4] Когда автор предложил одному своему коллеге-математику читать лекции в современной аудитории с компьютером, Интернетом, экраном, мультимедийным проектором и иллюстрировать лекционный материал "живыми" решениями задач в средах каких-либо математических программ, то он полушутя-полусерьезно ответил, что можно еще привлечь и "пританцовку", какая сопровождает певцов на эстраде. Доказал теорему — и тут на сцену, пардон, в аудиторию с визгом вбегают полуголые девицы и лихо отплясывают канкан... Можно лекцию читать также и под фонограмму, а под потолком аудитории устроить лазерное шоу — показывать, например, поверхность решения дифференциального уравнения в частных производных или настоящую 3D анимацию...

ближайший, к примеру. На рисунке 6 показана функция, созданная в среде Mathcad 15, возвращающая координаты шагов подъема на вершину холма – в точку максимума.

```
MaxTwoSep(f, x, D) := | (L ← last(x)   fmax ← f(x)   j ← p ← n ← 0   M⟨0⟩ ← x)
                      | while  D > 10^-10
                      |   | p ← 1
                      |   | while  p ∧ n < 100
                      |   |   | p ← 0
                      |   |   | for  i ∈ 0 .. L
                      |   |   |   for  X ∈ -D, D
                      |   |   |     | (xi ← xi + X   F ← f(x))
                      |   |   |     | (p ← X   j ← i   fmax ← F)  if  F > fmax
                      |   |   |     | xi ← xi - X
                      |   |   | (xj ← xj + p    n ← n + 1    M⟨n⟩ ← x)
                      |   | D ← D/2
                      | M
```

Рис. 6. Поиск максимума функции методом "Два шага" (Mathcad 15)

Программа на рис. 6 при поиске максимума делает два шага величиной D от очередной точки приближения (первая точка – это аргумент-вектор x) и переносит опорную точку туда, где значение оптимизируемой функции максимально. Эти два шага длинной X делаются по всем координатам анализируемой функции-вектора f (их число L) в отрицательном (-D) и положительном (4) в направлениях. По мере приближения к максимуму шаги укорачиваются. Эта процедура повторяется до тех пор, пока шаг не станет короче заданного значения погрешности ($D < 10^{-10}$ – см. рис. 6).

$$f(x) := \left(1 - 2x_0\right)^2 \cdot x_0 + 4\left(x_0 - 2 \cdot x_0 \cdot x_1\right)^2 \cdot \left(x_0 \cdot x_1\right)$$

$$M := \mathrm{MaxTwoSep}\left[f, \begin{pmatrix} 0.4 \\ 0.4 \end{pmatrix}, 0.01\right] = \begin{pmatrix} 0.4 & 0.39 & 0.38 & 0.37 & 0.36 \ldots & 0.173495621681213 & 0.173495622873306 & 0.173495622873306 \\ 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \ldots & 0.166666679382324 & 0.166666679382324 & 0.166666679382324 \end{pmatrix}$$

$x := \left(M^T\right)^{\langle 0 \rangle}$ $\quad y := \left(M^T\right)^{\langle 1 \rangle}$ $\quad z(x, y) := f\left(\begin{pmatrix} x \\ y \end{pmatrix}\right)$ $\quad n := \mathrm{cols}(M) = 101$

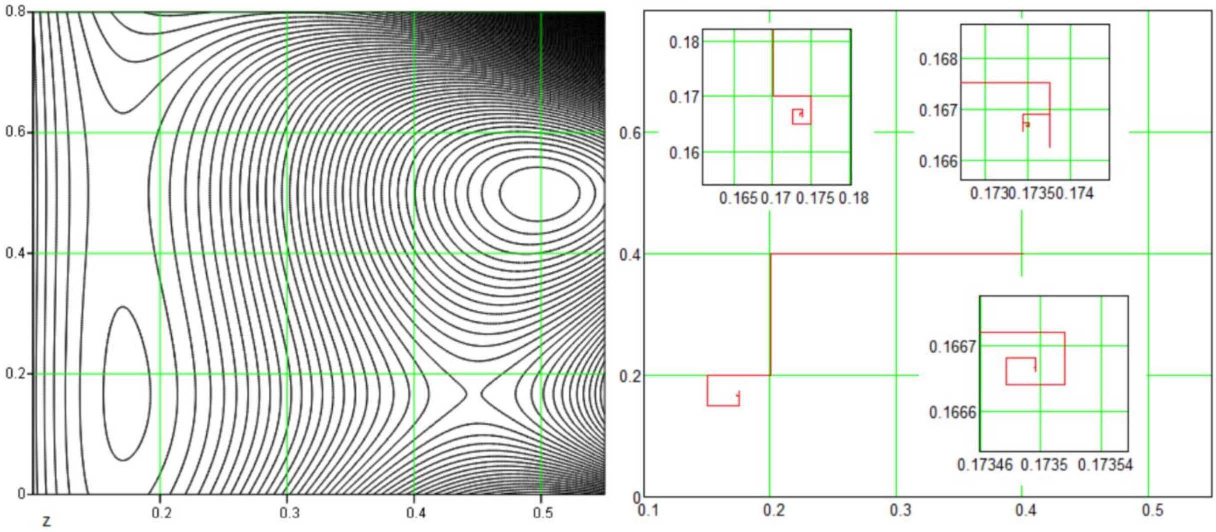


Рис. 7. Траектория поиска максимума у пятикоробочной функции методом "Два шага" (Mathcad 15)

На рисунке 7 показано, как программа-функция на рис. 6 была использована для поиска оптимального раскроя пяти коробок. При этом показана траектория подъема на высоту и увеличения "лупой" этой траектории вблизи искомой точки. В среде Mathcad есть подобная встроенная функция с именем `Maximize`, и мы ее еще будем использовать – см. рис. 8 ниже. Но эта функция возвращает только финальную точку поиска и не ясно, какой алгоритм встроен в нее. Правда, в среде Mathcad 15 можно было узнать и поменять алгоритм и при особых изощрениях можно было получить траекторию поиска [4].

Примечание.

Разработано большое количество изощренных алгоритмов оптимизации. Теперь нам трудно понять, что было побудительным мотивом их создания – избежание ложных и/или неточных решений или просто желание ускорить расчет на старых тихоходных ЭВМ. Сейчас скорость компьютеров существенно возросла и возник интерес к простейшим понятным (открытым) алгоритмам, которые несложно доработать своими руками. Из рисунка 7 видно, что было сделано более ста шагов итерации (см. переменную `n`), прежде, чем найден максимум функции. Если бы здесь был использован, например, метод деформируемого многогранника (другие его названия – метод Нелдера—Мида, симплекс-метод, метод амебы[5] – см. рис. 8), то ответ был бы найден не за сотые, а за тысячные доли секунды, не за сотню, а за десяток шагов. Но раньше "на старых добрых тихоходных ЭВМ" эти отрезки времени составляли часы или минуты, соответственно. Поэтому-то (но не только поэтому) приходилось выдумывать сложные алгоритмы

---

[5] Футуристическая амеба (в нашем случае для функции двух аргументов она треугольная) переваливаясь с боку на бок (рис. 8), переползает на вершину холма, уменьшаясь при этом в размерах (сжимаясь), пока не найдет искомую точку, обрамленную на рис. 4 и 7 линиями одного уровня.

оптимизации. На сайте статьи есть ссылки на анимации методов два шага и метода амебы. Мы прервали расчет не потому, что достигнута заданная точность, а потому, что выбрано заданное число итераций (100 – см. рис. 6). А их может быть бесчисленное множество, если выбрана неправильная начальная точка, и мы шагаем в правый верхний угол, где функция двух аргументов стремиться к бесконечности. На графиках рисунка 7 можно провести некую разделительную линию, разделяющую точки, от которых первое приближение дает правильный ответ. Быстродействие компьютеров возрождают и методы простого перебора, когда при не слишком большом числе аргументов перебираются все значения функции на линии, на плоскости (рис. 4 и 7), в объеме или в более мерном пространстве и запоминается точка, где функция имеет максимальное значение. Так можно найти не просто максимум – локальный максимум из многих возможных, а глобальный максимум в определенной области. Можно сочетать метод перебора с более изощренными методами, находя для них перебором с большим шагом точку для первого приближения. Годится тут и метод Монте Карло, когда область поиска не сканируется, а в не случайным образом "бросают" точки.

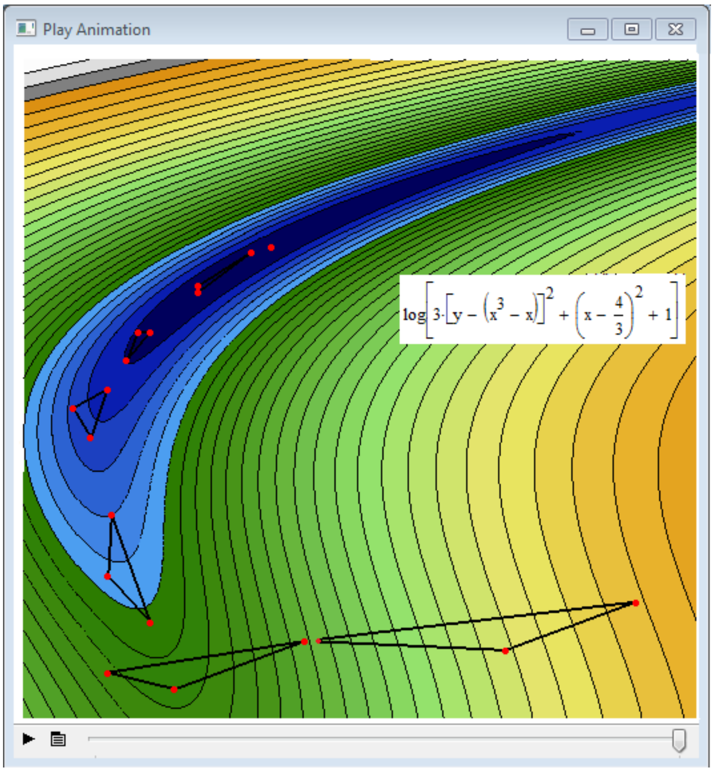


Рис. 8. Кадр анимации метода деформируемого многогранника

Пойдем дальше! Из вырезанных 16 квадратов также можно изготовить новые коробки[6], получив 64 новых квадратных обрезков, из которых… Вот вам и обещанный фрактал. Его можно

[6] На сайте статьи размещены решения, подобные тому, какое показано на рис. 4, для трех шагов раскроя (21 коробка – анализ функции трех аргументов) и четырех шагов раскроя (85 коробок – анализ функции четырех аргументов). Функцию трех и более аргументов уже нельзя отобразить графически так, как мы это делали для функции двух аргументов – см. рис. 4.5. По функции с более чем двумя аргументами можно нарисовать линии сечений и убедиться, что в точке максимума линии сечений также имеют максимальные значения. Это можно увидеть на сайте статьи для функции V_{21}(x, y, z).

отобразить графически, если, например, на плоскости выставить все коробки ("мал-мала меньше"), не двигая их, а только загибая вверх их стенки.

На рисунке 8 показана авторская формула, по которой можно рассчитать суммарный объем коробок при любом числе шагов раскроя исходной квадратной заготовки и при любой пропорции раскроя. Аргументом функции V_n будет не список (ряд переменных, разделенных запятой – см. 4), а вектор, число элементов которого равно числу шагов раскроя, а значения элементов – пропорции раскроя.

$$V_n(x) := \sum_{i=0}^{\mathrm{last}(x)} \left(4^i \cdot \prod_{j=0}^{i} x_j \cdot \left(\mathbf{if}\left(i = 0, 1, \prod_{j=0}^{i-1} x_j \right) - 2 \cdot \prod_{j=0}^{i} x_j \right)^2 \right)$$

$$V_n\left(\begin{bmatrix} \frac{1}{6} \end{bmatrix}\right) = 0.07407 \qquad V_n\left(\begin{bmatrix} \frac{9}{29} - \frac{3 \cdot \sqrt{7}}{58} \\ \frac{1}{6} \end{bmatrix}\right) = 0.07553 \qquad V_{21}(x) := V_n\left(\begin{bmatrix} x \\ \frac{9}{29} - \frac{3 \cdot \sqrt{7}}{58} \\ \frac{1}{6} \end{bmatrix}\right)$$

$$x := 0, 0.001 .. 0.5$$

$$x_3 := 0.2$$

$$x_3 := \mathbf{Maximize}\left(V_{21}, x_3\right) = 0.173645$$

$$V_{21}\left(x_3\right) = 0.0755599$$

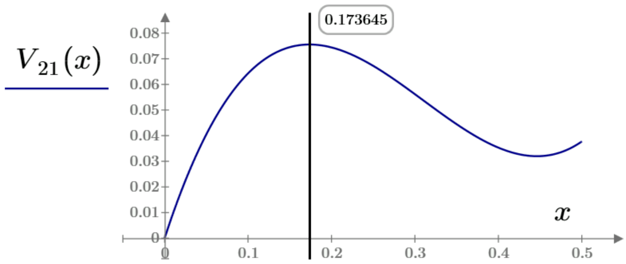


Рис. 8. Задача о произвольном числе шагов квадратной заготовки (Mathcad Prime)

На рисунке 8 функция V_n с одним оператором суммы и тремя операторами произведения, обобщающая функции V_1, и V_5, показанные на рис. 2, 3 и 4, протестирована на решениях, найденных для одной коробки (рис. 2) и для пяти коробок (рис. 3 и 4). Все совпало! Задача о 21 коробке (одна большая, четыре средние и 16 маленьких) на рис. 8 решена средствами не символьной, а численной математики – вызовом функции `Maximaze`[7] с начальным предположением `x: = 0.2`. Символьная математика не справилась с этой задачей из-за того, в частности, что в анализируемой функции V_n присутствует функция `if` ("гроза" символьной математики). Можно, конечно, попытаться избавиться от функции `if`, заменив ее функцией-ступенькой (функцией Хэвисайда), но можно предположить, что ответ (если он все-таки будет получен) окажется слишком громоздким. Кроме того, есть более простой способ решения данной задачи при числе шагов раскроя квадратной заготовки, стремящемся к… бесконечности. А фрактал и бесконечность – это, как говорится, "два сапога – пара".

История этого решения такова. В [3] была сделана попытка анализа раскроя 5 (1+4), 21 (1+4+16) и 85 (1+4+16+64) коробок. После выхода этой книги в свет было получено письмо от анонимного читателя, который предложил решение для любого числа шагов раскроя, последний

[7] Тут можно было задействовать и нашу самодельную функцию `MaxTwoStep`, показанную на рис. 6. Она может работать при любом числе аргументов оптимизируемой функции.

из которых имеет пропорцию 1/6, определение которого было показано на рис. 2. На рисунке 10 показана эта *рекуррентная формула*: задается значение $x_0 = 1/6$ (пропорция раскроя самых маленьких последних коробочек[8]), далее находится пропорция для предыдущих шагов раскроя x_1, x_2, x_3 и т.д. до x_n.

$$N := 7 \qquad x_0 := \frac{1}{6} \qquad i := 0..N$$

$$x_{i+1} := \frac{1}{4 + 2 \cdot \sqrt{1 - 2\, x_i \cdot \left(1 - 2\, x_i\right)}}$$

$$V_n(\mathrm{reverse}(x)) = 0.075560532301892$$

$$n := \sum_{i=0}^{N} 4^i = 21845$$

$$x = \begin{bmatrix} 0.166666666666667 \\ 0.173495621841487 \\ 0.173644979419639 \\ 0.1736481106798 \\ 0.173648176263916 \\ 0.173648177637545 \\ 0.173648177666315 \\ 0.173648177666917 \\ 0.17364817766693 \end{bmatrix}$$

Рис. 10. Задача о бесконечном числе коробок с максимальным суммарным объемом (Mathcad Prime)

Решение, показанное на рис. 10, было опубликовано в [5]. Кто автор этого решения и верно ли оно – остается загадкой. На сайте статьи, повторяем, показаны аналитические решения задач с 21 и 85 коробками с выводом численных результатов, которые совпали с теми, какие показаны вектором на рис. 10.

Но до бесконечности нам еще очень далеко. В расчете на рисунке 10 было сделано только восемь шагов раскроя коробки, для которых были найдены численные значения оптимальных пропорций вырезок. Если число шагов раскроя увеличивать, то будет достигнут предел точности численной математики Mathcad, и изменения в ответе (в векторе на рис. 10) не будут фиксироваться.

Оптимизацию бесконечного числа коробок можно провести, если сделать допущение – пропорция раскроя квадратов сделать постоянной для всех шагов раскроя. На рисунках 11 и 13 показано решение такой задачи.

[8] По идее эти самые маленькие коробки находятся на втором краю… бесконечности. Но бесконечность применительно к данной задаче о коробках имеет левый край (первый шаг раскроя), но не имеет правого края. Вот такое противоречие: по бесконечности мы "прокатываемся" с двух концов – раскраивая коробки (одно направление) и рассчитывая пропорции раскроя (второе, обратное направление).

$$\sum_{i=1}^{\infty} 4^{i-1}\cdot x^{i}\cdot\left(x^{i-1}-2\,x^{i}\right)^{2} \rightarrow \begin{cases} \dfrac{x-x\cdot\infty+\infty}{x^{2}} & \text{if } x=-\left(\dfrac{1}{4}\cdot 2^{\frac{1}{3}}\right)-\dfrac{2^{\frac{1}{3}}\cdot\sqrt{3}}{4}\cdot 1\mathrm{i} \vee x=-\left(\dfrac{1}{4}\cdot 2^{\frac{1}{3}}\right)+\dfrac{2^{\frac{1}{3}}\cdot\sqrt{3}}{4}\cdot 1\mathrm{i} \vee x=\dfrac{1}{2}\cdot 2^{\frac{1}{3}} \\ \dfrac{(2\cdot x-1)^{2}\cdot\left(\lim\limits_{i\rightarrow\infty} 2^{2\cdot i}\cdot x^{3\cdot i}-4\cdot x^{3}\right)}{4\cdot x^{2}\cdot\left(4\cdot x^{3}-1\right)} & \text{else if } x\neq-\left(\dfrac{1}{4}\cdot 2^{\frac{1}{3}}\right)-\dfrac{2^{\frac{1}{3}}\cdot\sqrt{3}}{4}\cdot 1\mathrm{i} \wedge x\neq-\left(\dfrac{1}{4}\cdot 2^{\frac{1}{3}}\right)+\dfrac{2^{\frac{1}{3}}\cdot\sqrt{3}}{4}\cdot 1\mathrm{i} \wedge x\neq\dfrac{1}{2}\cdot 2^{\frac{1}{3}} \end{cases}$$

Рис. 11. Задача о бесконечном числе коробок с максимальным суммарным объемом и с одинаковыми пропорциями раскроя (Mathcad Prime)

На рисунке 11 с помощью оператора суммы формируется формула, задающая суммарный объем бесконечного числа коробок с пропорцией x в раскрое, и сделана попытка найти формулу для этого бесконечного ряда. Пакет Mathcad Prime (рис. 11) выдал "абсолютно точный и абсолютно бесполезный ответ[9]", в то время как Mathcad 12 выдавал вполне приемлемый результат, и мы его покажем ниже. Это связано с тем, что в эти две версии Mathcad встроена разная символьная математика[10].

Была сделана попытка найти формулу для сходящегося ряда на сайте пакета Mathematica – см. рис. 12. Этот сайт не стал выдавать "страшную" формулу (см. рис. 11), а сообщил, что ряд сходится при определенных значениях параметра x, а именно при $4|x|^3<1$.

[9] Три варианта анекдота про математика, выдающего "абсолютно точный и абсолютно бесполезный ответ". (1) Похороны. Прохожий спрашивает: "Кого хоронят?". Ответ: "Вон того – кто в гробу лежит!". (2) Воздушный шар вырвался из облаков. Летящие кричат человеку на земле: "Где мы находимся?". Ответ: "Вы находитесь в корзине воздушного шара!". (3) Туристы немного заблудились и спрашивают прохожего: "Мы правильно идем к электричке?". Ответ: "Нет, неправильно! Не строем и не в ногу!" (это был преподаватель математики в военном училище).

[10] Mathcad изначально создавался как пакет численной математики, к которому впоследствии (в 5-й версии) было прикреплено ядро символьной математики из пакета Maple, которое затем (в 13-й версии) было заменено на ядро символьной математики из пакета MuPAD. Это ядро перешло и в Mathcad Prime.

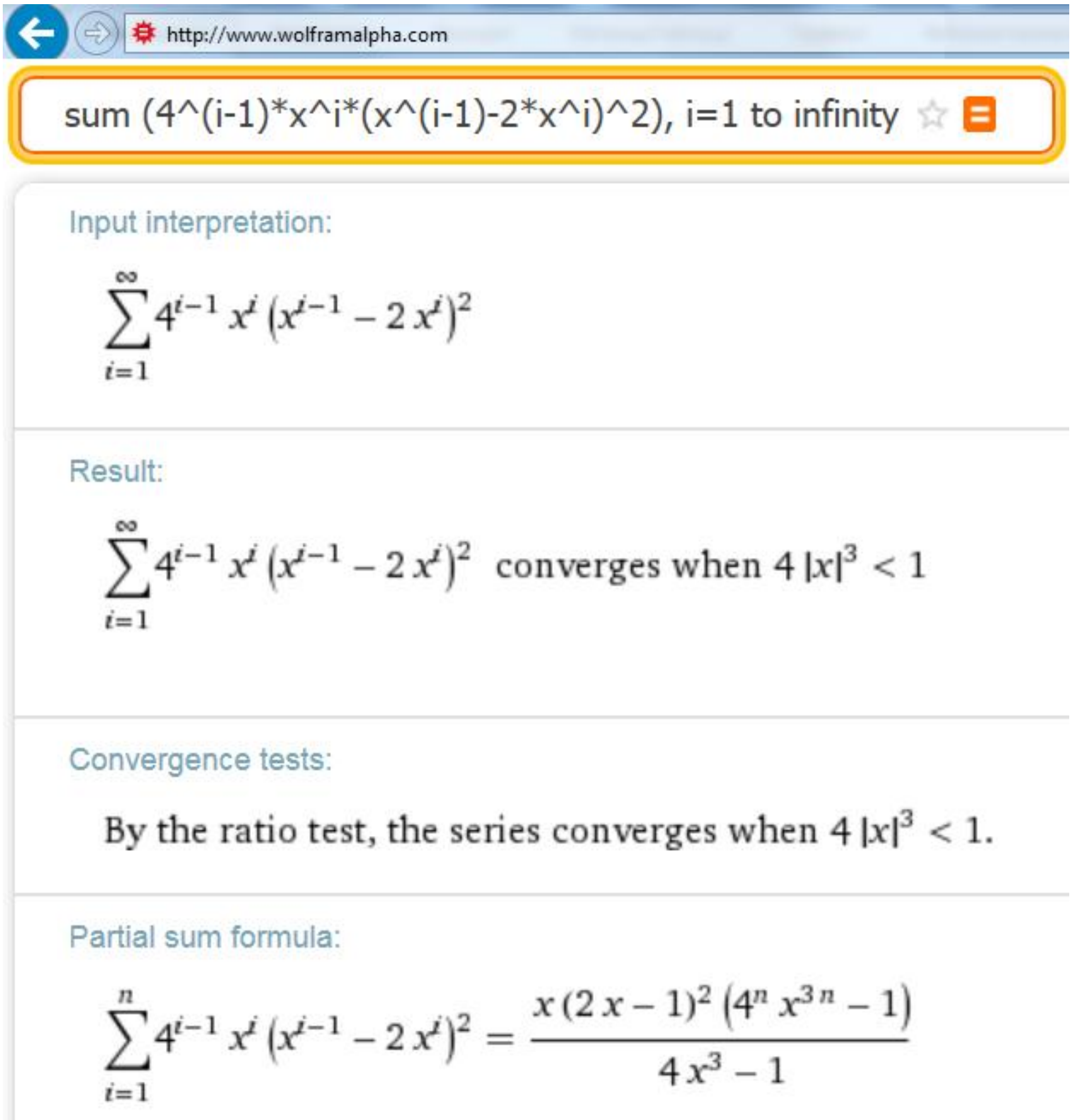


Рис. 12. Попытка упрощения суммы

$$V(x) := \sum_{i=1}^{\infty} 4^{i-1} \cdot x^{i} \cdot \left(x^{i-1} - 2\ x^{i}\right)^{2} \xrightarrow{assume\,,\,|x| < \sqrt[3]{\frac{1}{4}}} -\frac{x \cdot (2 \cdot x - 1)^{2}}{4 \cdot x^{3} - 1}$$

$$\frac{\mathrm{d}}{\mathrm{d}x} V(x) = 0 \xrightarrow[float\,,\,250]{solve} \begin{bmatrix} 0.5 \\ 0.7660444431189780352023\ldots 685107618 \\ -0.939692620785908384054 10 \ldots 4687190687 \\ 0.1736481776669303488517 16\ldots 002083068 \end{bmatrix}$$

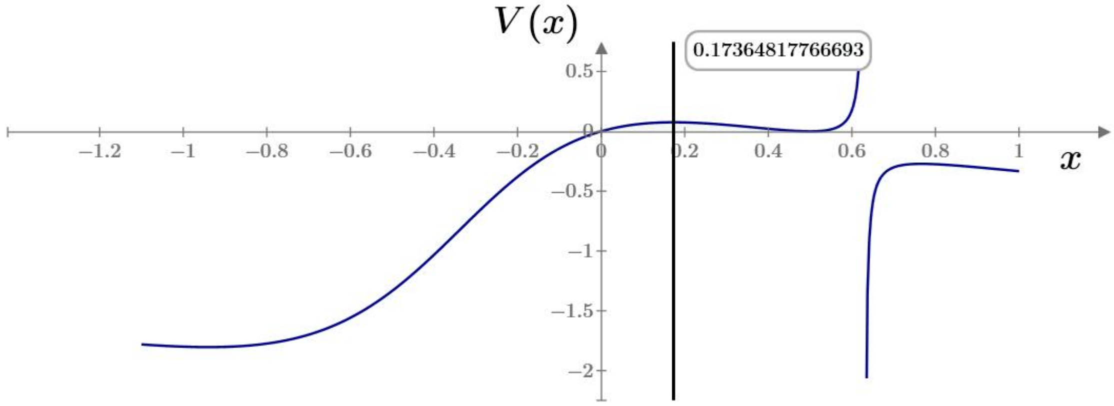


Рис. 13. Решение задачи о бесконечном числе коробок с максимальным суммарным объемом и с одинаковыми пропорциями раскроя (Mathcad Prime)

Дополнительное условие по параметру x было сообщено через ключевое слово `assume` (предполагать) пакету Mathcad Prime, и он справился с задачей: нашел сумму сходящегося ряда и нашел корни соответствующего уравнения – см. рис. 13. Их оказалось четыре, последний из них – решение задачи.

Числа и их аналитические выражения, характеризующие параметры раскроя квадратной заготовки для коробок с максимальным суммарным объемом, можно рассматривать как новые математические константы. Таких констант, характеризующих реальные физические и математические объекты, не просто много, а бесконечно много. И мы это еще раз доказали расчетом, показанным на рис. 10. Есть такой сайт http://www.people.fas.harvard.edu/~sfinch, который поддерживает Стивен Финч (Steven Finch[11] – см. http://www.people.fas.harvard.edu/~sfinch/csolve) с описанием различных констант. Наши "коробочные" константы должны найти там свое законное место.

Вывод.

В настоящее время нарабатываются новые задачи и методы их решения для зарождающегося учебного курса для школ и вузов под условным названием ФизМатИнформика (ФМИ). Задачи, опубликованные в [1] и в этой статье, один из вкладов в этот процесс, который

[11] Он был одним из технических писателей по пакету Mathcad.

для авторов должен зафиксироваться не определенном этапе выходом в свет в 2016 году в Издательстве Лань книги "Физико-математические этюды с Mathcad и Интернет". При этом стоит не только придумывать новые примеры, но и обратить внимание на "старые добрые задачи", новую жизнь которым дают современные информационные технологии.

Литература:

1. Очков В.Ф., Богомолова Е.П., Иванов Д.А. Программное уравнение или ФМИ // Cloud of Science. Т. 2, № 3. 2015. С. 473-515. URL: http://twt.mpei.ac.ru/ochkov/PMI.pdf
2. Очков В.Ф. Живые кинематические схемы в Mathcad // Открытое образование. 2013. № 3. С. 27–33. URL: http://twt.mpei.ac.ru/ochkov/Mathcad-15/kinematic.html
3. Очков В.Ф. Mathcad 8 Pro для студентов и инженеров. М.: КомпьютерПресс, 1999. URL: http://twt.mpei.ac.ru/ochkov/mc8Pro.book
4. Очков В.Ф., Богомолова Е.П. Путешествие окружности в треугольнике, а треугольника в ложбине или Сам себе компьютерный режиссер // Открытое образование. № 2. 2015. С. 24-32. URL: http://twt.mpei.ac.ru/ochkov/TrianglInCircle.pdf
5. Очков В.Ф. Mathcad 12 для студентов и инженеров. БХВ-Петербург 2005. URL: http://twt.mpei.ac.ru/ochkov/Mathcad_12