# Оптимизация функции одного аргумента

Из круглой жестянки по несложной технологии изготавливается *пожарное ведро*[1] (см. п. 1 на рис. 1): вырезается сектор, затем полученная выкройка сворачивается в конус (точка `a` подтягивается к точке `c`), а шов сваривается (паяется).

[1] Батунер Л. М., Позин М. Е. Математические методы в химической технике. — Л.: Химия, 1960. В этой книге описывалось не пожарное ведро, а конус, изготавливаемый загибом сектора на круглом листе фильтровальной бумаги. Спрашивается, какой должен быть загиб, чтобы этот фильтр работал оптимально.

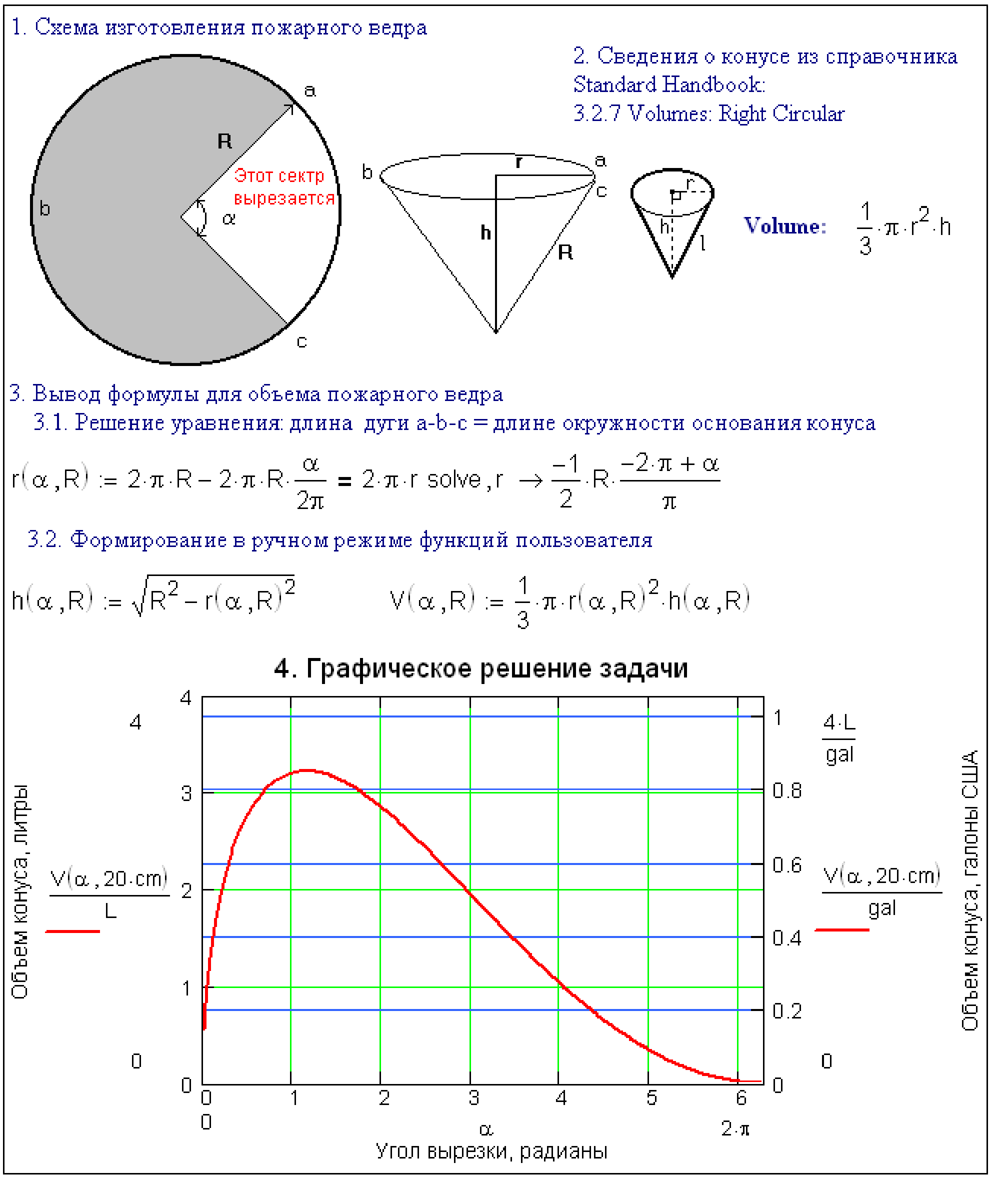


**Рис. 1.** Задача о пожарном ведре: схема решения

Требуется найти угол вырезки α, при котором объем ведра будет *максимальным*.

И при решении данной задачи мы должны вывести зависимость объема ведра V от угла вырезки α. Далее можно взять первую производную от этой функции, приравнять ее к нулю и найти корень полученного уравнения. Не обойтись тут и без второй производной, если нужно убедиться, что найденное решение — максимум, а не минимум или точка перегиба, где первая производная также равна нулю. Жестянщик, которому поручат сделать такое пожарное ведро, скорее всего, незнаком с математическим анализом, азы которого мы только что изложили. Но в среде Mathcad поставленная задача вполне окажется по плечу "компьютеризированному" жестянщику.

Рисунок заготовки ведра и самого ведра в п. 1 на рис. 1 сделан с помощью графического редактора Paint и перенесен в Mathcad-документ через буфер обмена (см. рис. 1.58).

В п. 2 на рис. 1 скопированы данные о геометрии конуса из *стандартного справочника* Mathcad, который удобен тем, что входит в состав пакета и всегда находится под рукой (см. рис. 1.4). Перенос данных из справочника в Mathcad-документ также автоматизирован, что исключает их искажение — вручную переписывая формулу из книги, немудрено и ошибиться.

В п. 3 на рис. 1 записаны операторы символьных (аналитических) преобразований: оператором `solve` решается алгебраическое уравнение, а тандемом `r(α, R):=...→` формируется функция пользователя с именем `r` (п. 3.1). Далее формируются две другие функции: `h` и `V` (п. 3.2). Зависимости выводятся из несложной геометрии круга и конуса: дуга *abc* выкройки ($2\pi \cdot R - 2\pi \cdot R \cdot \alpha/2\pi$) становится длиной окружности в основании конуса ($2\pi \cdot r$), а высота конуса `h`, радиус его основания `r` и радиус заготовки `R` — это стороны прямоугольного треугольника, длины которых связаны теоремой Пифагора.

***Примечание***

Решение задачи на компьютере — это только один из этапов общего процесса решения задачи. Другой немаловажный этап — это постановка задачи человеком, перекладывание словесного описания задачи на язык математики и компьютера.

Прежде чем искать максимум функции, необходимо убедиться, что он существует. Лучший же способ увидеть максимум — просмотреть график функции. В среде Mathcad, как уже отмечалось, есть семь видов графиков *(см. разд. 1.6)*, первый из которых (**График X-Y**) отображен на рис. 1. Здесь график построен по "двухшаговой" технологии: задается вид функции и сразу отдается команда на вставку графика в Mathcad-документ. По умолчанию значение ординаты графика меняется от –10 до 10. После построения наброска графика его нужно будет отформатировать — изменить разброс по оси *x* и другие установки по умолчанию.

На графике в районе угла, равного 1 радиан (60—70 угловых градусов), отчетливо виден максимум функции. Как его уточнить?

На рис. 2 показано аналитическое (символьное) решение данной задачи через поиск корней уравнения: первая производная функции `V(α, R)` равна нулю. Таких точек у функции `V(α, R)` оказалось три — два максимума ($\alpha_1$ и $\alpha_2$)[2] и один локальный ми-

---

[2] У нас на рис. 2 решение тандемом `■:=...■→` (см. рис. 1.16) заносится в вектор, три элемента которых — переменные с текстовыми индексами. Внимание! Порядок вывода корней

нимум ($\alpha_0$). Задача решалась так. Сначала оператором ■ `solve,` ■ → выводились на дисплей корни уравнения $V'(\alpha,\ R)=0$, а после того, как стало ясно, что уравнение в среде Mathcad решается и корней три, то оператор ■ `solve,` ■ → встраивался в оператор ■ := ■, в левый операнд (местозаменитель) которого вставлен вектор с тремя элементами по числу найденных корней. Элементы вектора хранят переменные $\alpha_0$, $\alpha_1$ и $\alpha_2$ (здесь не числовые, а текстовые индексы $\alpha$`.0`, $\alpha$`.1` и $\alpha$`.2`), третья из которых $\alpha_2$ — наш искомый ответ, который выводится на печать в угловых градусах и накладывается маркером на график функции $V(\alpha,\ R)$ при `R=20 см`. При аналитическом решении задачи в отличие от графического или численного (см. рис. 3) переменная `R` может оставаться пустой и не принимать никакого числового значения. На то оно и *символьное* решение. Это одно из преимуществ символьной математики — значение переменной `R` не влияет на решение, и его можно не задавать. При *численном* же решении задачи (см. рис. 3 и 4) всегда будет оставаться открытым вопрос, а не изменится ли решение, если изменить значение переменной `R` либо какой-нибудь другой константы. Беглый взгляд на задачу говорит о том, что нет, не измениться. Но в других задачах, как правило, такого однозначного ответа с "первого, беглого" взгляда не видно. Но если перейти от "одноведерной" к "двухведерной" задаче — задаче, когда из круглой заготовки вырезаются два ведра — к поиску корней производной функции $V(\alpha,\ R)\ +\ V(2\pi-\alpha,\ R)$, то символьная математика Mathcad даст сбой. Нужно переходить к численным методам, которые, как правило, выдают один ответ из множества возможных и не с абсолютной, а с ограниченной точностью.

---

уравнения оператором `solve` может меняться при очередном открытии файла в иной версии Mathcad — там где, например, было $2\pi$, может оказаться другой корень.

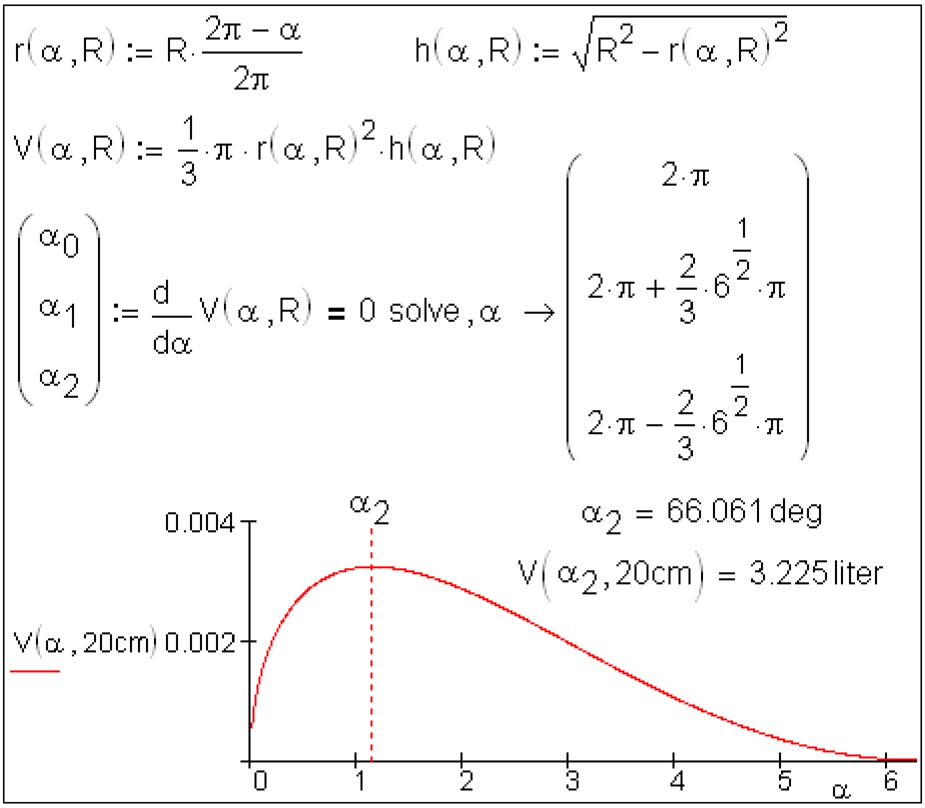


**Рис. 2.** Аналитическое решение задачи о пожарном ведре

Самый старый способ численного поиска максимумов и минимумов в среде Mathcad, работавший еще в DOS-версиях, — это модификация блока `Given/Find`, когда функция `Find` заменяется на функцию `MinErr`. Блок `Given/Find` предназначен для решения систем аналитических уравнений. Если у системы нет решения, то функция `Find` возвращает сообщение об ошибке. Если в такой ситуации функцию `Find` заменить на функцию `MinErr`, то она вернет уже не сообщение об ошибке, как функция `Find`, а координаты точки, дающей минимальную (min) ошибку (err), *невязку* системы уравнений. На рис. 3 показано численное решение задачи о пожарном ведре через блок `Given/MinErr`, с учетом того факта, что при `R=20 см` объем нашего конуса не превышает четырех литров (`4L`).

$$r(\alpha, R) := R \cdot \frac{\alpha - 2\pi}{2\pi} \qquad h(\alpha, R) := \sqrt{R^2 - r(\alpha, R)^2}$$

$$V(\alpha, R) := \frac{1}{3} \cdot \pi \cdot r(\alpha, R)^2 \cdot h(\alpha, R) \qquad R \equiv 20cm \qquad \alpha := 1rad$$

$$\text{Given} \qquad V(\alpha, R) = 4L \qquad \alpha := \text{Minerr}(\alpha) = 1.153\,rad$$

$$\alpha = 66.061\,deg \qquad V(\alpha, R) = 3.225\,L$$

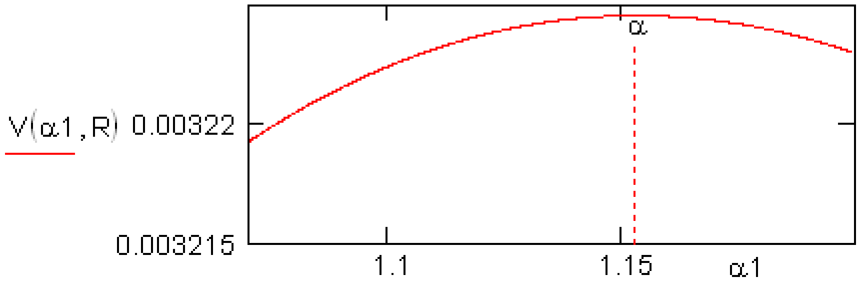


**Рис. 3.** Численное решение задачи о пожарном ведре

Численный метод, реализованный на рис. 3, требует первого приближения (`α:=1 rad`) и возвращает координаты ближайшей к первому приближению точки, дающей минимальное отклонение объема ведра от заданных нами четырех литров. Здесь может быть любое другое значение, большее искомого максимума, но не совсем "любое". При слишком большом значении, численные методы, заложенные в функцию `MinErr`, могут давать сбой, т. к. разница между значениями анализируемой функции и заданными заведомо максимальным значением нивелируется[3].

Попробуем усложнить задачу о пожарном ведре и перейдем к уже упоминавшейся нами трехведерной задаче. Что если вырезанный из заготовки сектор не выбрасывать, а скручивать из него второй конус? Вместимость двух ведер, естественно, будет больше вместимости одного ведра. Вопрос на пари: как необходимо раскроить заготовку, чтобы суммарный объем двух ведер был максимальным? Большинство опрошенных, опираясь на здравый смысл, ответят, что заготовку нужно разрезать пополам по диаметру, и... проиграют пари.

На рис. 4 показано численное решение "двухведерной задачи" в среде Mathcad. Аналитическое решение тут, как уже было отмечено ранее, не проходит[4]. В основном

---

[3] Здесь можно почитать и поварьировать встроенными переменными `TOL` и `CTOL`.

[4] Хотя его можно "дожать": преобразовать или упростить выражения. Но стоит ли!? Для учебных целей — да, а для практических — еще вопрос.

оно (численное решение) повторяет решение, показанное на рис. 3, но имеет свои особенности.

$$r(\alpha, R) := R \cdot \frac{\alpha - 2\pi}{2\pi} \qquad h(\alpha, R) := \sqrt{R^2 - r(\alpha, R)^2} \qquad V(\alpha, R) := \frac{1}{3} \cdot \pi \cdot r(\alpha, R)^2 \cdot h(\alpha, R)$$

$$R \equiv 20cm \qquad V_2(\alpha) := V(\alpha, R) + V(2\pi - \alpha, R)$$

$$\alpha := \begin{pmatrix} 2 \\ 4.5 \end{pmatrix} rad \qquad \alpha := Maximize(V_2, \alpha) = \begin{pmatrix} 2.036 \\ 4.247 \end{pmatrix} rad \qquad \alpha = \begin{pmatrix} 116.645 \\ 243.355 \end{pmatrix} deg$$

$$\alpha_0 + \alpha_1 = 360\,deg \qquad \overrightarrow{V_2(\alpha)} = \begin{pmatrix} 3.653 \\ 3.653 \end{pmatrix} L \qquad V_2(180deg) = 3.628\,L$$

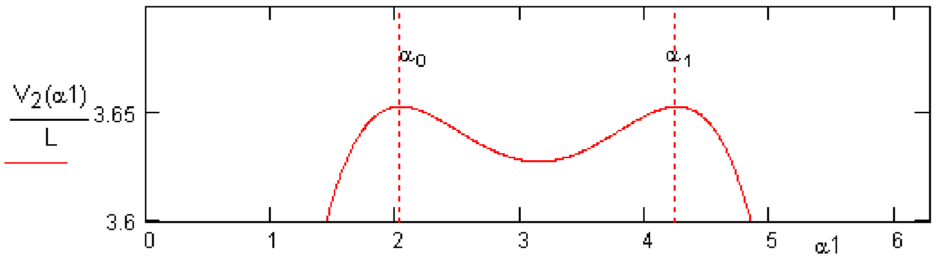


**Рис. 4.** Численное решение задачи о двух пожарных ведрах

Для поиска максимума объема ведра используется не функция `MinErr`, как на рис. 3, а функция `Maximize`, введенная в Mathcad начиная с 8-й версии для численного решения оптимизационных задач. Основное преимущество функции `Maximize`, перед функцией `MinErr` в том, что функция `Maximize` (как и ее "напарница" функция `Minimize`) не требует примерного знания максимального (минимального) значения анализируемой функции, у которой ищут экстремум. С другой стороны, функция `MinErr` тоже имеет преимущества. Главная из них описана пословицей "Старый конь борозды не испортит!". "Старость" функции `MinErr` определяет ее надежность. Очень часто можно видеть ситуацию, когда функции `Minimaze` и `Maximize` дают сбой, а функция `MinErr` спасает положение. В качестве первого приближения в решении, показанном на рис. 4, взят не скаляр, а *вектор* — два значения угла вырезки вблизи максимумов, определенных визуально (`2` и `4.5` радиан). В такой ситуации функция `Minimize` вернула нам также два уточненных значения, симметрично расположенных по отношению к значению π.

### *Примечание*

Последнее, в принципе, еще нужно доказать. У нас получилось, что $\alpha_0+\alpha_1=2\pi$, но это *примерно* равно, т. к. использовались численные (приближенные), а не аналитические методы решения задачи.

Пожарное ведро делается в виде конуса для того, чтобы его нельзя было поставить на пол, а потом использовать не по прямому назначению (для стирки, например) — такое ведро свалится на бок[5]. Такую же форму (конус без ножки) имеет бокал "Пей до дна!". Решение задачи о двух пожарных ведрах также "валится на бок" — на "левый" ($\alpha<180°$) или на "правый" ($\alpha>180°$).

Не выбросив меньший сектор и сделав из него второе пожарное ведро, мы мало что выиграли — второе ведро дало небольшую прибавку в объеме. Раскрой заготовки для двух ведер не по диаметру ($\alpha=180$), а хитрым способом, дал совсем ничтожный выигрыш по суммарному объему ведер. Но нам важен не результат, а сам процесс расчета: "Цель ничто, движение — все!".

[5] Автор, служа в армии (см. **http://twt.mpei.ac.ru/ochkov/Army.htm**) и мо́я там, как водится, полы, ставил пожарное ведро (а другого не было) в перевернутую табуретку. Отсюда и "любовь" к пожарным ведрам, вылившаяся на страницы этой книги.