Этюд 2

Скаляр и вектор в компьютерных вычислениях

Сайт этюда: **https://www.ptcusercommunity.com/thread/127828**

На лекциях по информатике и математике в Московском энергетическом институте авторы иногда делают опросы (см. также начало этюда 14) для того, чтобы уточнить, что студенты изучали раньше в школе и что они изучают сейчас в вузе на параллельных курсах физики, химии, термодинамики, теоретической механики и других учебных дисциплин. На вопрос о том, знают ли они, что такое *скалярная* физическая величина и *векторная* физическая величина, все студенты хором отвечают: "Знаем! Нам все уши в школе на уроках физики прожужжали, объясняя это!". "Знаем" они выкрикивают вслух, а про "уши", конечно, умалчивают. Но в студенческой реакции на вопрос можно услышать данное продолжение ответа… Все школьники и студенты безошибочно знают, что масса — это скалярная величина, а вес (сила) — векторная и что задачи по физике нужно решать с учетом этих фундаментальных понятий.

Скаляр и вектор присутствует и в языках программирования, электронных таблицах и математических пакетах, которые в настоящее время широко используются при преподавании информатики и других учебных дисциплин в школе и вузе.

В языках программирования есть понятие *массива* (array) — одномерного, двумерного и многомерного. Электронные таблицы (CuperCalc, VisiCalc, Excel и др.) создавались именно для того, чтобы хранить и обрабатывать эти самые таблицы — одну из разновидностей массива данных. В имени языка программирования технических расчетов MatLab, который часто пытаются сравнить с Mathcad, — Mat означает матрицу (a **Mat**rix), a Lab — лабораторию (a **Lab**oratory). В математических пакетах, в частности, в пакете Mathcad понятие массива прописано в явном виде. Пользователь может ввести в расчет *массив* (матрицу) с одним столбцом, который в среде Mathcad обозначен как *вектор*. Но при этом нужно четко понимать, что маткадовский вектор несет в себе как минимум две смысловые нагрузки, которые мы попытаемся уяснить на примере решения двух несложных задач.

На рисунке 2.1 показан расчет ускорения тела с заданной массой, если известны три силы, приложенные к телу в трех направлениях — по оси x, y и z (наша задача решается в трехмерном пространстве). Но мы можем перевести ее и на плоскость (двумерное пространство) и на линию. А можно перейти и в четырехмерное и "болеемерное" пространство, которое часто описывается в научной фантастике, но которое реально присутствует в математике.

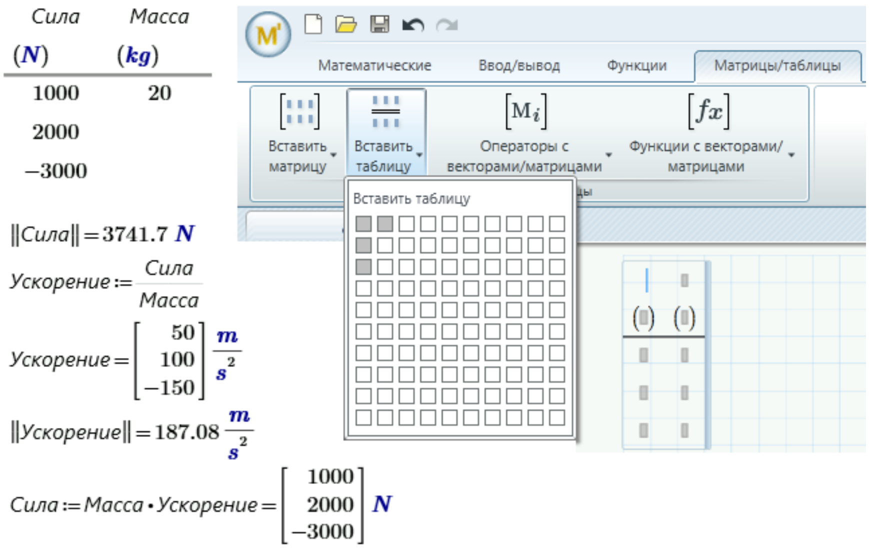

Рис. 2.1. Задача о движении тела в пространстве (Mathcad Prime)

В расчете, показанном на рис. 2.1, мы ввели две исходные величины — Сила в виде вектора с тремя компонентами (1000, 2000 и минус 3000 ньютон) и Масса в виде скаляра со значением 20 кг. Затем мы рассчитали длину вектора ||Сила|| — определили абсолютное значение силы, приложенной к телу, и вектор Ускорение (три его составляющие по осям x, y и z). Многие пользователи Mathcad ошибочно полагают, что длина вектора — это число компонентов в нем, а не квадратный корень из суммы квадратов значений компонентов. Эта путаница связана и с тем, что в среде Mathcad число компонентов вектора подсчитывает встроенная функция с именем длина (см. рис. 2.2 ниже). Оператор же вычисления "настоящей" длины вектора (нормы вектора) ||**v**|| появился только в Mathcad Prime.

Примечание

Мы воспользовались одним удобным инструментом — *таблицей*, введенной в Mathcad Prime. Введенную прямоугольную таблица размером 3 на 2 (три строки и два столбца — см. правую часть рис. 2.1) мы заполнили частично, дав силе три значения (вектор), а массе только одно (скаляр).

На рисунке 2.2 показано решение в среде предыдущей версии Mathcad — Mathcad 15 нашей задачи об ускорении тела.

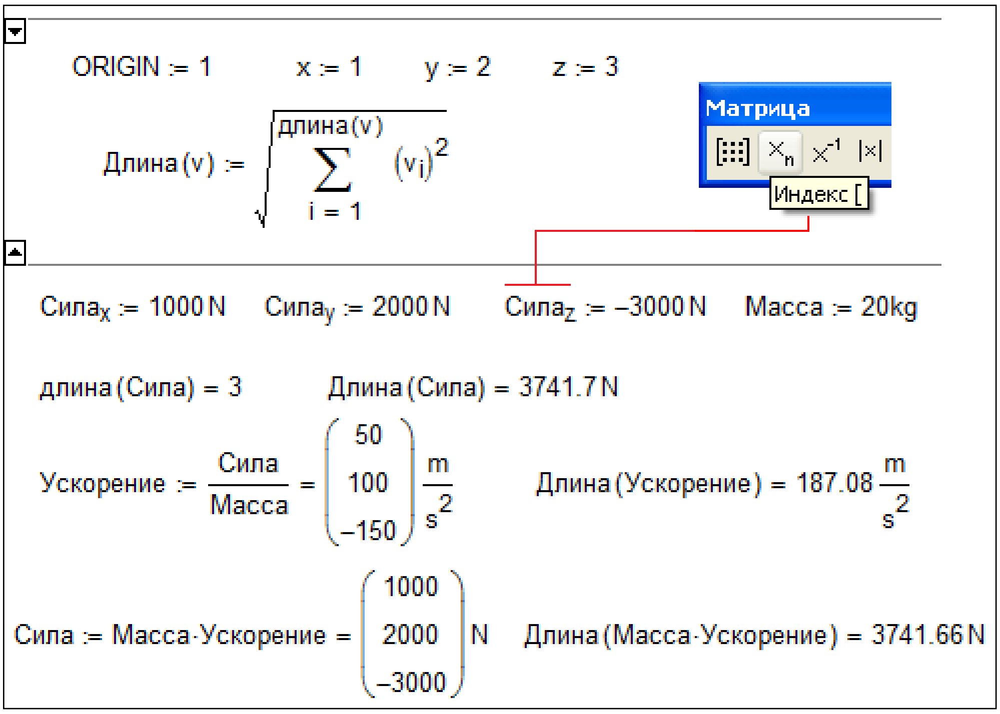

Рис. 2.2. Задача о движении тела в пространстве (Mathcad 15)

В варианте решения задачи, показанном на рис. 2.2, значение силы вводится не вектором с тремя компонентами, как на рис. 2.1, а покоординатно: Сила$_x$, Сила$_y$ и Сила$_z$. Для этого в выделенной области расчета (см. верхнюю часть рис. 2.2), во-первых, системной переменной ORIGIN[1] присвоено значение единицы (по умолчанию оно равно нулю — почему, скажем чуть позже), а во-вторых, переменным `x`, `y` и `z` присвоены значения 1, 2 и 3, соответственно. Это сделано для того, чтобы компоненты вектора Сила обозначать не как Сила$_1$, Сила$_2$ и Сила$_3$, а как Сила$_x$, Сила$_y$ и Сила$_z$, что более соответствует "физике" задачи: разложение вектора по трем координатам: `x`, `y` и `z`.

Глядя на рис. 2.1 и 2.2 можно представить себе реальную силу (стрелку-вектор) в 3741.7 N (ньютон), приложенную к телу и раскладывающуюся по координатам `x`, `y` и `z` со значениями 1000, 2000 и минус 3000 N соответственно. Такие векторы-стрелки часто рисуют в учебниках и задачниках по физике, иллюстрируя силы (см., например, рис. 7.7 в этюде 7), направления движения, электрические и магнитные поля и прочие векторные физические величины. Кстати, в различных учебниках и популярных изданиях по физике к векторам силы привязывают… Лебедя, Рака и Щуку из знаменитой басни Крылова, и анализируют, смог бы воз сдвинутся с места[2]. В этюде 9 мы будем рассматривать силы, приложенные к планете или спутнику вследствие действия закона всемирного тяготения. Эти силы будут разложены по двум или трем осям пространства.

---

[1] В книге описан только один расчет, где системная переменная `ORIGIN` равна не нулю и не единице, а двойке – см. рис. 16.2 в этюде 16.

[2] Авторов данной книги можно уподобить этим трем персонажам. Но если у Крылова "воз и ныне там", то авторам все-таки удалось издать книгу, несмотря на то, что они "тянули" ее в разные стороны: в сторону математики, в сторону физики и в сторону информатики.

Второй тип данных, который в среде Mathcad имеет также форму вектора, но не является по сути (физической) таковым — *это массив данных в одном столбце*. Что это такое, мы поясним рисунком 2.3, где показано решение задачи линейной интерполяции по точкам.

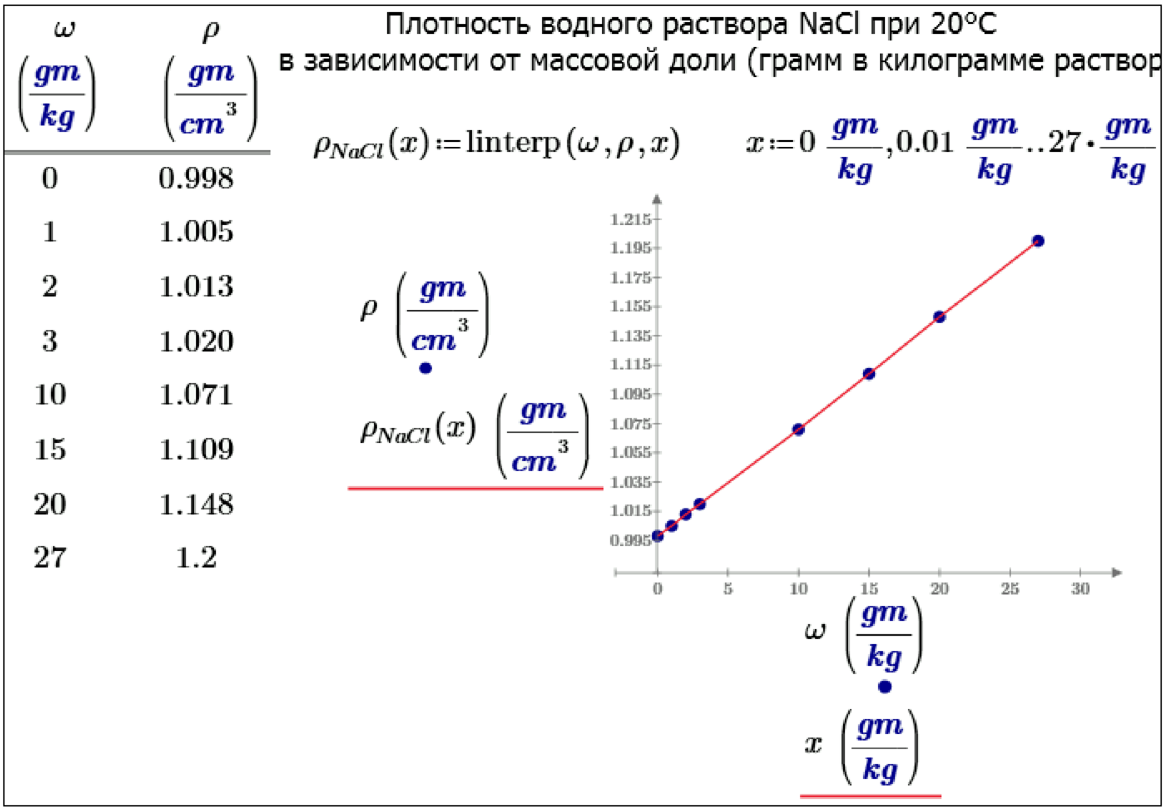

Рис. 2.3. Сглаживание точек в среде Mathcad

Суть задачи такова. Имеются дискретные значения массовой доли (отношения массы растворенного вещества к массе раствора) водного раствора поваренной соли ω и его плотности ρ. Необходимо создать непрерывную функцию, которая бы возвращала значение плотности раствора соли в промежуточных точках. Такая функция нужна при пересчете видов концентраций — см. рис. 1.6 и 1.7 в этюде 1. Подробнее об этой статистической операции будет рассказано в этюде 14.

Координаты сглаживаемых точек в задаче на рис. 2.3 хранятся в двух массивах данных (векторах) с именами ω и ρ. Эти массивы по форме, повторяем, являются векторами, но это не векторы в "физическом" смысле того слова, который был подчеркнут на рис. 2.1 и 2.2. У векторов ω и ρ в задаче, показанной на рис. 2.3, конечно, можно посчитать длину (квадратный корень из суммы квадратов компонентов), но эта величина не будет иметь никакого физического смысла, того смысла, который ясно просматривается в задаче, показанной на рис. 2.1 и 2.2. На рис. 2.3 фактически показана *база данных* с двумя *полями* (ω и ρ) и с 8 *записями*, которую мы храним в виде двух векторов одного размера. Эту базу данных можно отображать графически, обрабатывать статистически (что и показано на рис. 2.3), сортировать по полям, фильтровать по записям и т. д., но к какому-то n-мерному пространству эти "векторы" никакого отношения не имеют.

Примечание, касающееся и этюда 1

Через точки, показанные на рис. 2.3, можно провести прямую линию и описать зависимость плотности и концентрации водного раствора поваренной соли простой формулой $\rho = a + b \cdot \omega$ (см. описание этой процедуры в этюде 14). Но при этом будет потеряна физическая суть формулы, базирующаяся на таблице замеренных значений плотности и концентрации. Так что лучше оставить формулу (программу) в виде, показанном на рис. 2.3, вставив в функцию два вектора ω и ρ, и сделав ее тем самым автономной. В функции хорошо будет виден диапазон применимости формулы — от 0 до 27% по массовой доле, что помогает исключить ошибку экстраполяция.

Второе отличие массивов, показанных на рис. 2.1 и 2.2, с одной стороны, и на рис. 2.3, с другой стороны, в том, что нумерация *компонентов* вектора, как правило, начинается с единицы, а *элементов* массива — с нуля. На рисунке 2.3 эта установка (ORIGIN=0) зафиксирована умолчанием. Третье отличие: составляющие векторов, показанных на рис. 2.1 и 2.2, обычно называют *компонентами* (координатами вектора), а векторов, показанных на рис. 2.3, — *элементами* (элементами матрицы-столбца). Но это не столь важно.

Как, наверное, уже отметил читатель, задачи, отображенные на рис. 2.1, 2.2 и 2.3, решаются с использованием *единиц физических величин*: метров (m), секунд (*s*), килограммов (kg) и др. Это очень мощный инструмент пакета Mathcad [4], позволяющий эффективно контролировать правильность расчетов (не складывать метры с килограммами, грубо говоря) и организовать удобный ввод-вывод численных значений переменных. Но тут встает интересный вопрос в технологическом и смысловом плане — может ли массив/вектор Mathcad иметь элементы с различной размерностью, когда, например, один элемент имеет размерность длины, второй — времени, а третий вообще безразмерный. Традиционный Mathcad, например, Mathcad 15 такой возможности не допускал. В среде Mathcad Prime это стало допустимым. Технологическая сторона вопроса тут решена, а что со смысловой стороной? Есть ли реальные математические задачи, решение которых требует использования векторов и массивов с различной размерностью элементов?! На рисунке 2.4 такая задача показана: даны две точки на плоскости с координатами x_1–y_1 и x_2–y_2, необходимо найти значения коэффициентов a и b уравнения прямой $y(x) = a + b \cdot x$, проходящей через эти две точки. На рисунке 2.4 показано, как можно эту задачу решить в среде Mathcad Prime.

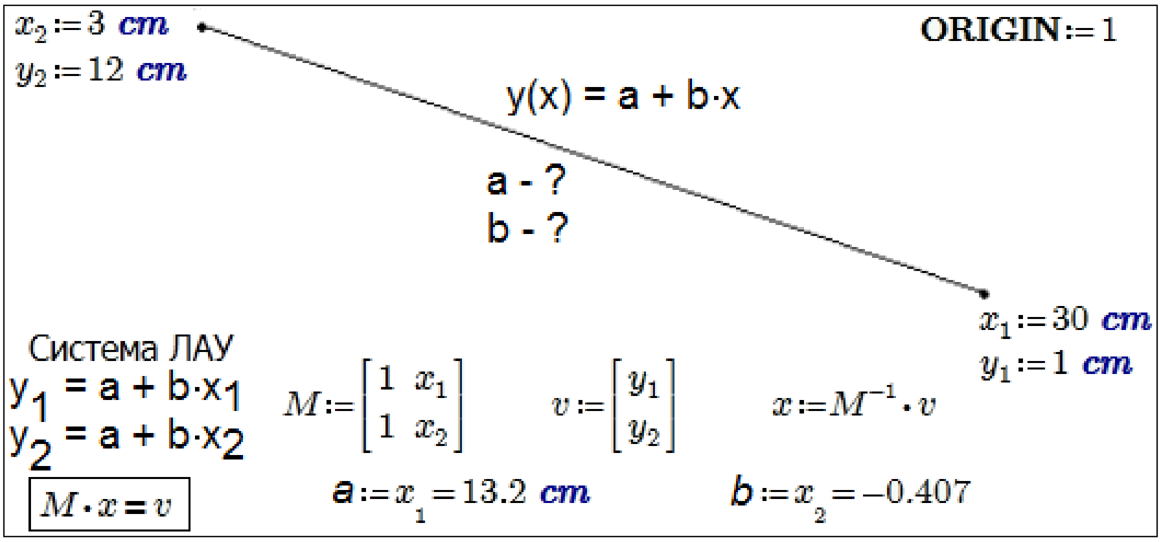

Рис. 2.4. Задача об уравнении прямой линии

Задачу об уравнении прямой линии, проходящей через две точки на плоскости[3], можно свести к решению системы двух линейных алгебраических уравнений (ЛАУ) в векторно-матричной форме `M·x = v`, где `M` — квадратная матрица коэффициентов при неизвестных `x`, а `v` — вектор свободных членов. Решение системы — это нахождение значений вектора `x`, состоящего из двух компонентов, первый из которых хранит величину с размерностью длины, а второй безразмерный. Матрица `M` также хранит разноразмерные величины, одни из которых (первый столбец) безразмерные, а другие (второй столбец) имеют размерность длины. Здесь также возникает вопрос — можно ли конструкции `v` и `x`, задействованные в задаче на рис. 2.4, называть векторами с тем "физическим" смыслом, который просматривается в этом слове в задаче, показанной на рис. 2.1 и 2.2. Нет, конечно! Тем не менее, в курсе высшей математики под названием "Линейная алгебра", один из разделов которого посвящен решению систем линейных алгебраических уравнений, конструкции x и v называют векторами, а конструкцию `M` — матрицей.

Вектор по критериям классического программирования (BASIC, Pascal и др. языки) — это просто одномерный массив, т.е. массив с одним индексом, а матрица — двумерный массив, т.е. массив с двумя индексами. В программировании используют и *многомерные* (трех-, четырех- и т.д.) массивы. Если скаляр можно уподобить точке, вектор — отрезку прямой линии, а матрицу — прямоугольнику, то трехмерный массив — параллелепипеду. В среде Mathcad эту объемную конструкцию можно создать в виде *вложенного массива*, пример которого показан на рис. 2.5, где показано хранение дискретных значений функции `f(x, y, z)`: `f=0.81` при `x=0.5, y=15` и `z=0.2, f=0.64` при `x=1, y=10` и `z=0.24` и т.д.

---

[3] Мы эту задачу будем решать в этюде 8 (рис. 8.5), когда будем моделировать движение дворников автомобиля, закрепленных в двух точках.

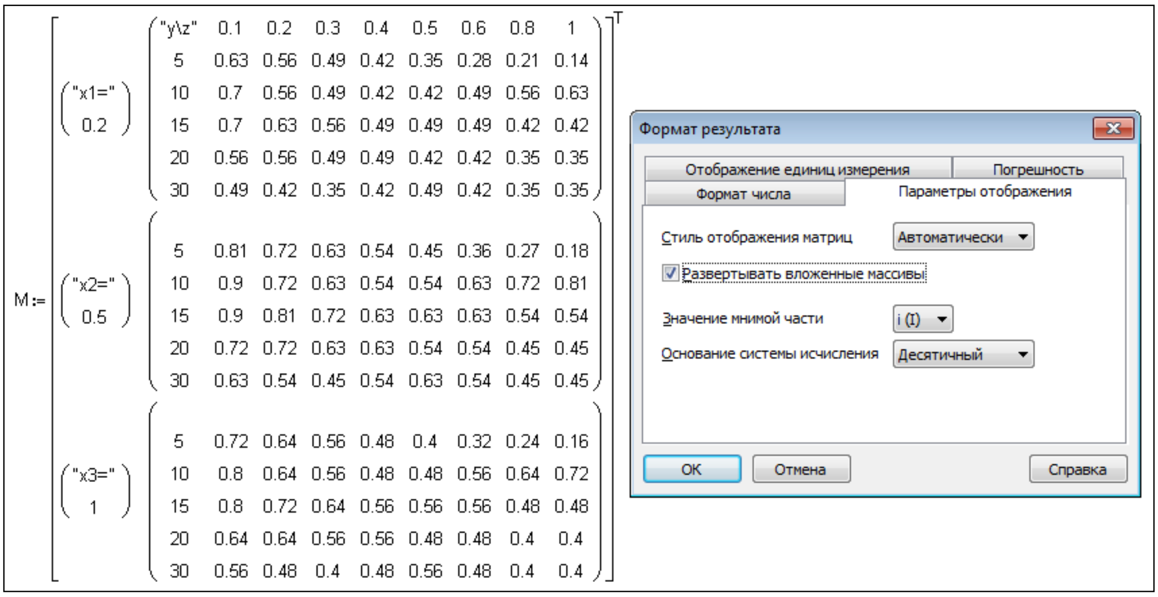

Рис. 2.5. Пример вложенного массива в Mathcad

Вложенный массив — это вектор или матрица ("простой" массив), один, два или более элементов которого это не скаляр, а новый вектор или матрица. Если говорить о "физике" решаемых в среде Mathcad задач, то тут нужно упомянуть *тензор*, который соответствует трех- и более мерному массиву. Типичный вложенный массив в среде Mathcad — это вектор, хранящий три матрицы — интенсивность трех цветов изображения (см. рис. 5.7 в этюде 5).

Если вернутся к началу этюда, где упомянут опыт общения авторов со студентами, изучающими информатику и математику в школе и вузе, то можно отметить такой феномен. Первый автор этой книги читает студентам курс информатики почти четверть века (см. **http://twt.mpei.ac.ru/ochkov/Potoki.htm**). Двадцать лет назад бывшие школьники, а теперь студенты были очень заинтересованы в изучении программирования, они, как они выражались, "тащились" от программирования. Теперь же у многих студентов на программирование возникла некая "аллергия". Причин у этого печального явления, как можно предположить, две. Первая в том, что это общешкольная проблема, связанная с неприятием того, что преподают в школе. У многих людей, например, наблюдается стойкое отвращение к литературным произведениям, которые изучали в школе. Двадцать лет назад программирование в школе почти не изучали, поэтому в те времена у студентов был огромный интерес к этому предмету. А теперь его, увы, нет.

Вторая причина потери у школьников, а потом и у студентов интереса к программированию заключается в том, что преподаватели часто дают студентам для программирования уж очень "занудные" задачи. Эти задачи в прошлом веке, может быть, были сами по себе интересны. Но сейчас для решения таких задач появились новые средства, исключающие традиционное программирование. И школьники-студенты это интуитивно чувствуют или даже прямо знают об этом.

Конкретный пример. На рисунке 2.2 можно видеть, как современными средствами Mathcad (оператором суммы) была создана функция пользователя с именем длина, возвращающая "физическую" длину вектора. Но многие преподаватели информатики в школе и в вузе по-прежнему предлагают школьникам и студентам создавать такие и другие подобные примитивные функции средствами традиционного программирования (BASIC, Pascal, C и др.), не зная или умалчивая о том, что в современных программных оболочках многое уже давно реализовано встроенными функциями и операторами. Кстати, функцию пользователя с именем длина можно создать средствами программирования Mathcad (см. рис. 2.6) с помощью цикла for (цикла с параметром).

```
ORIGIN ≡ 1
Length(v) := | "Длина вектора"
             | Sum ← 0
             | for  i ∈ 1.. длина(v)
             |   Sum ← Sum + (v_i)^2
             | return √Sum
```

$$\text{Сила} := \begin{pmatrix} 1000\,N \\ 2000\,N \\ -3000\,N \end{pmatrix} \qquad \text{длина}(\text{Сила}) = 3 \qquad \text{Length}(\text{Сила}) = 3741.66\,N$$

Рис. 2.6. Программа в среде Mathcad

В Mathcad есть, например, очень интересная функция `match(z, A)`, возвращающая координаты элемента со значением `z` в матрице `A`. Если элементов со значением `z` в матрице `A` окажется несколько, то функция `match` вернет тот самый вложенный массив (см. рис. 2.5) — вектор, элементы (или компоненты) которого будут новыми векторами с двумя элементами — с номером строки и номером столбца матрицы `A`, где "сидит" величина `z`. Эту функцию мы будем использовать в этюде 15 в программе решения задачи коммивояжера (рис. 15.10). Можно, конечно, поручить школьникам или студентам создать функцию `match` через программирование с циклами и альтернативами. Но лучше давать студентам более занимательные задачи, те задачи, у которых решения еще нет или оно еще не так "вылизано". Заставлять студентов создавать функцию `match` это все равно, что заставить их написать программу расчета синуса через разложение в ряд Тейлора, умалчивая при этом, что синус есть в любом калькуляторе и встроен в любой язык программирования.

Эту проблему можно поставить шире и перенести ее на другие школьные и вузовские дисциплины — физику [31], математику [32], химию [5].

Дело в том, что изучение многих учебных дисциплин ориентировано на докомпьютерные и даже докалькуляторные методы решения задач. И это мы отмечали во Введении и этюде 1. Возьмем, к примеру, физику, с которой начался данный этюд. Решая задачу по этой дисциплине, школьник или студент должен вспомнить или найти в учебнике формулу или набор формул, по которой можно решить конкретную задачу. Сами же эти формулы являются, как правило, частными случаями решения систем уравнений (алгебраических, дифференциальных, интегральных и др.), отображающих фундаментальные законы сохранения вещества, энергии; перехода энергии из одной формы в другую и т. д. и т. п. Упор на использование готовых решений, а не на постановку задачи в общем виде был сделан когда-то давно по простой и понятной причине. В те времена, когда создавались методики решения этих задач, не было эффективных и доступных средств решения уравнений и систем, отображающих фундаментальные законы физики, химии и других научных (учебных) дисциплин. Теперь же такие средства появились. И это требует кардинального пересмотра методик решения задач и переписывания почти всех учебников и задачников. Более подробно мы эту тему описали в этюде 1.