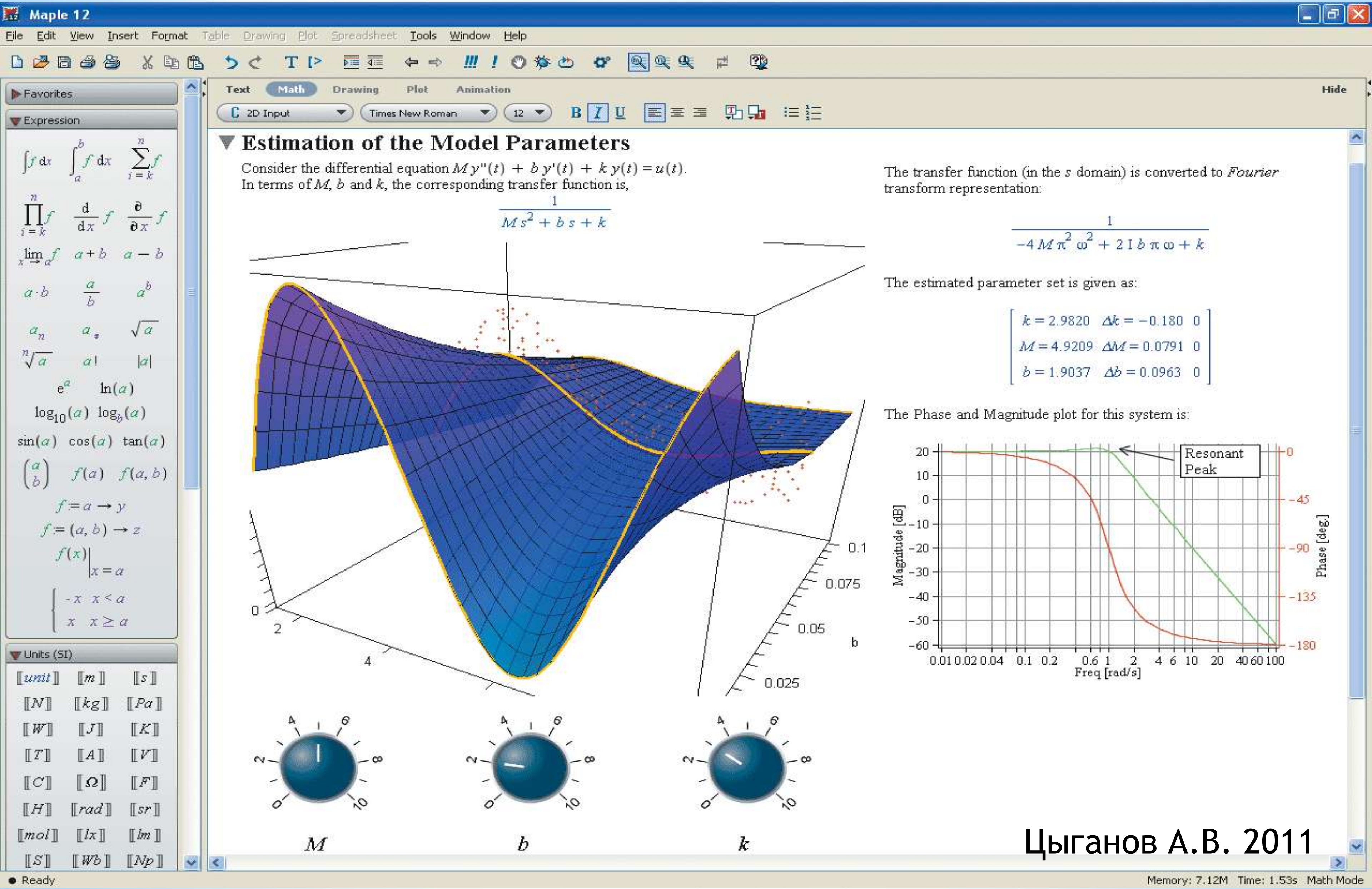
Алгоритмы символьных вычислений



Цыганов А.В. 2011

Недостойно одаренному человеку тратить, подобно рабу, часы на вычисления, которые безусловно можно было бы доверить любому лицу, если бы при этом применить машину.

Готфрид Вильгельм фон Лейбниц

Символьные вычисления — это преобразования и работа с математическими равенствами и формулами, как с последовательностью символов.

Численные расчёты оперируют приближёнными численными значениями, стоящими за математическими выражениями, т.е. числами (целыми и с плавающей запятой).

К примеру, решения уравнения $x^2 = 2x + 1$ "равны" **−0.41421356** и **2.41421356**.

Однако часто необходимо работать не с приближенным цифровым значением, а с точной величиной

$$\mathbf{1 \pm \sqrt{2}}.$$

С этого простейшего примера и начинается разница между численными и символьными вычислениями.

Но, самое главное, есть задачи, которые вообще невозможно решить численно.

Символьные вычисления или компьютерная алгебра — научная дисциплина, ставящая целью разработку алгоритмов и программного обеспечения для решения с помощью компьютера задач, в которых исходные данные и результаты имеют вид математических выражений, формул.

Системы компьютерной алгебры — **Maple, Mathematica, MATLAB, MATCAD**, **Sage** и др. — содержат процедуры выполнения базовых преобразований выражений и набор пакетов процедур, предназначенных для решения более специальных задач, как то интегрирование функций, преобразование и решение дифференциальных уравнений, нахождение пределов, доказательство комбинаторных тождеств и т.д.

В компьютерной алгебре имеется большое количество актуальных задач разной степени сложности, как теоретических, связанных с разработкой и модификацией алгоритмов, так и сугубо практических - написание модулей, встраиваемых в системы компьютерной алгебры.

ПО для аналитических (символьных) вычислений это очень мощный инструмент позволяющий автоматизировать наиболее рутинную и требующую повышенного внимания часть работы, оперирующий при этом аналитической записью данных, т. е. фактически математическими формулами. Такие ПО называют **средой программирования**, с той разницей, что в качестве элементов языка программирования выступают привычные человеку математические обозначения.

Недостатки систем символьных вычислений:

отсутствие компилятора в машинный код ; неудобная стыковка с другими языками программирования; неудобная обработка строк и ряда других типов, в том числе классов ООП; передача аргументов копированием и т.д

Поэтому можно работать прямо на Common Lisp, Python и языках которые позволяют RAD (Rapid Application Development -- быстрая разработка приложений☺.

Сравнение

Maple (паскаль и др.)	**Mathematica (C- подобно)**
restart	Clear
with(Student)	Needs["ComputerArithmetic`"]
sin(x)	Sin[x]
dsolve(diff(y(x),x$2)=sin(x),y(x))	DSolve[y''[x]==Sin[x],y[x],x]
int(sin(x),x=a..b)	Integrate[Sin[x],{x,a,b}]
plot(f(x),x=a..b,color=blue)	Plot[f[x],{x,a,b}, PlotStyle->Blue]
display(pict)	Show[pict]
subs(x=a,f(x))	f[x]/.x=a

Так что синтаксис не главное - главное удобство ☺.

Вычислим сумму в Maple 14

$$> combine\left(sum\left(sum\left((-1)^{i+j+k}\cdot p,\ k=1..j\right)+2\cdot r,\ i=1..k\right)\right)$$

$$2\,(k+1)\,r+\frac{1}{4}\,p\,(-1)^{2j+k}-\frac{1}{4}\,(-1)^{j+k}p-2\,r-\frac{1}{4}\,(-1)^{2j}p+\frac{1}{4}\,(-1)^{j}p$$

Вычислим тоже самое в Mathematica 8

“простой” набор

```
In[3]:= Sum[Sum[p*(-1)^(i+k)*(-1)^(j), {i, 1, k}] + 2*r, {k, 1, j}]
```

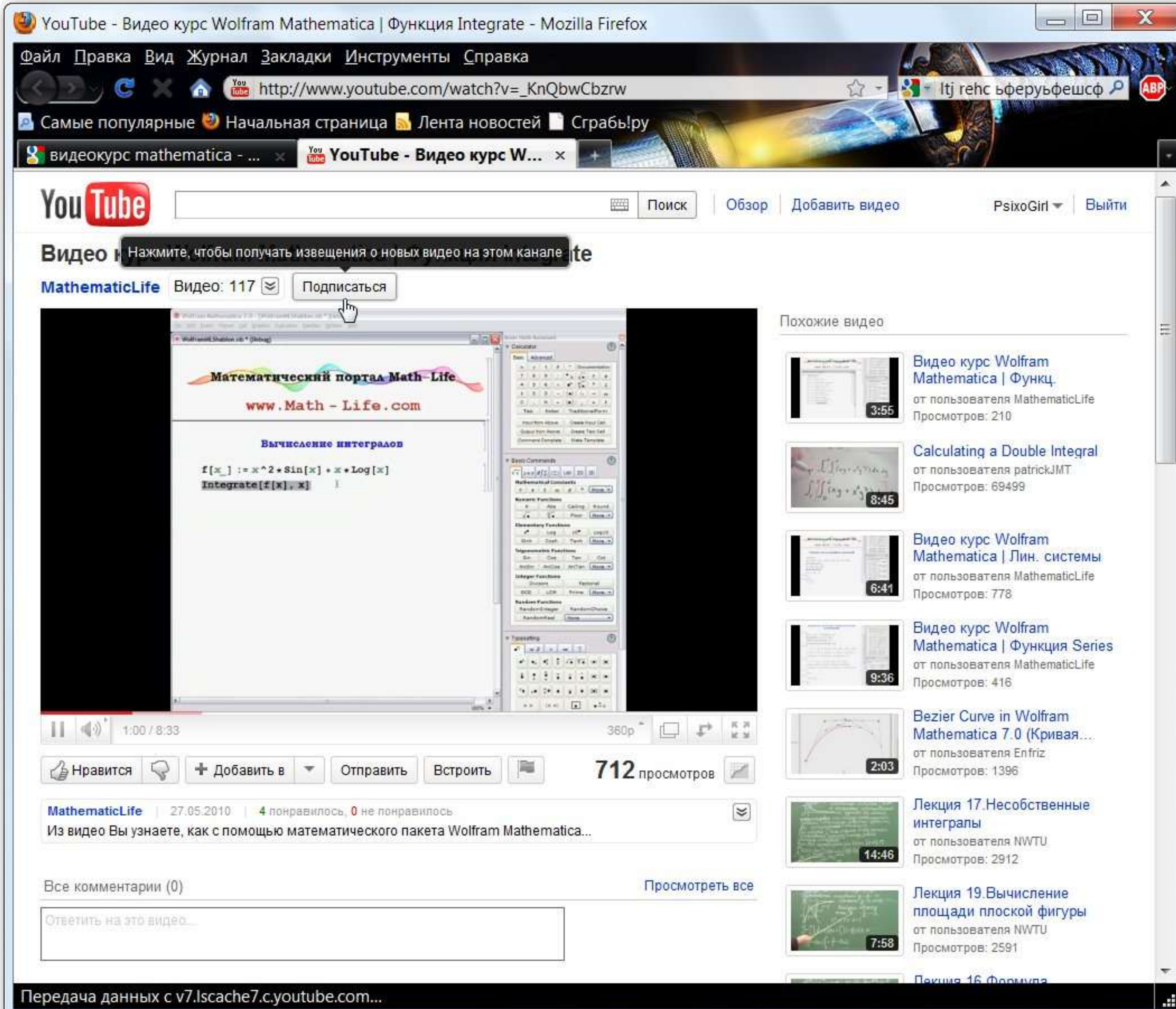
$$\text{Out[3]=}\ \frac{1}{2}\left(1+2\,(-1)^{2j}+2\,j^{3}+2\,p^{2}+2\,r\right)$$

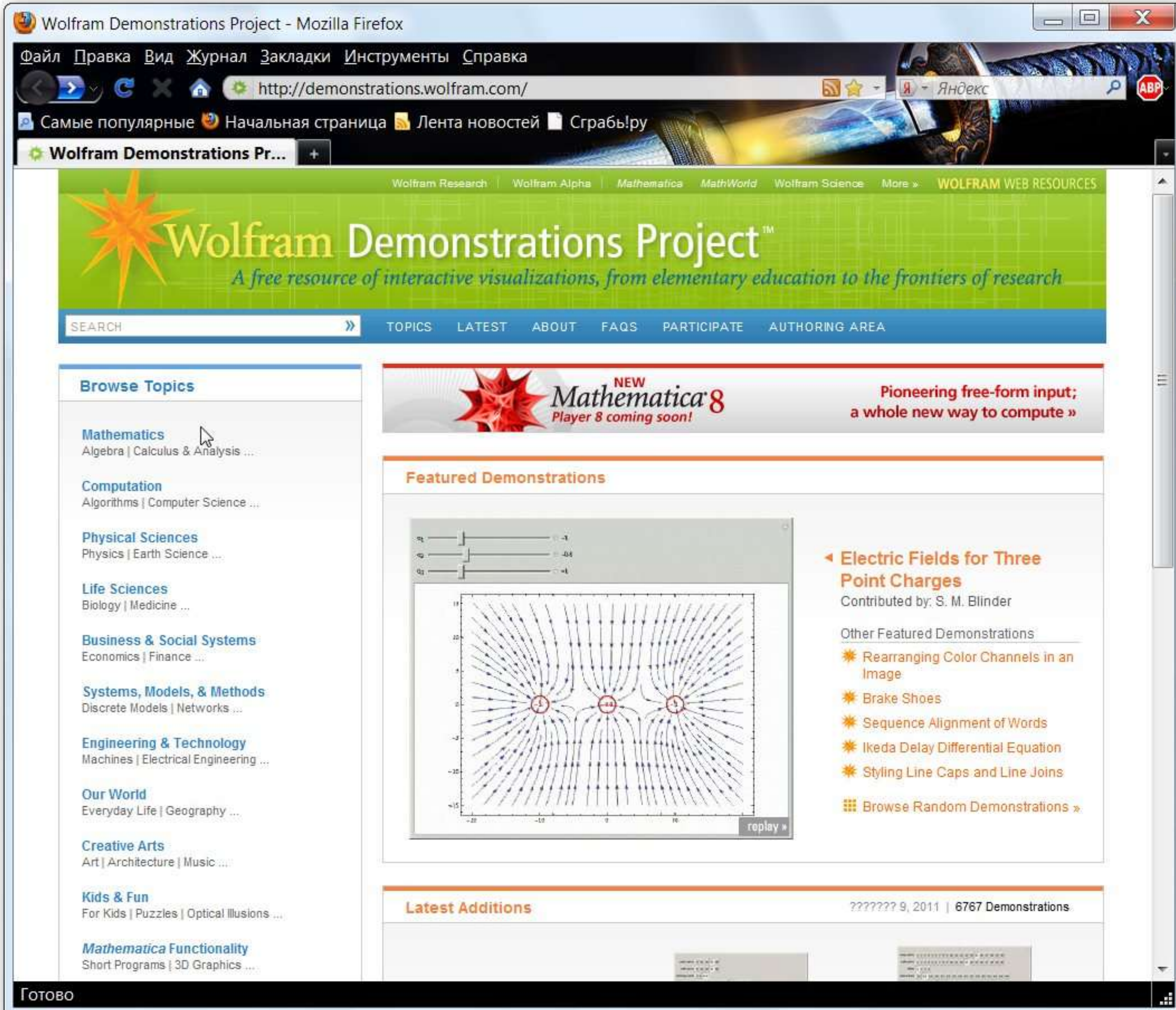
$$\text{In[2]:=}\ \sum_{i=1}^{k}\left(\left(\sum_{k=1}^{j}(-1)\,\hat{}\,(i+j+k)*p\right)+2*r\right)$$

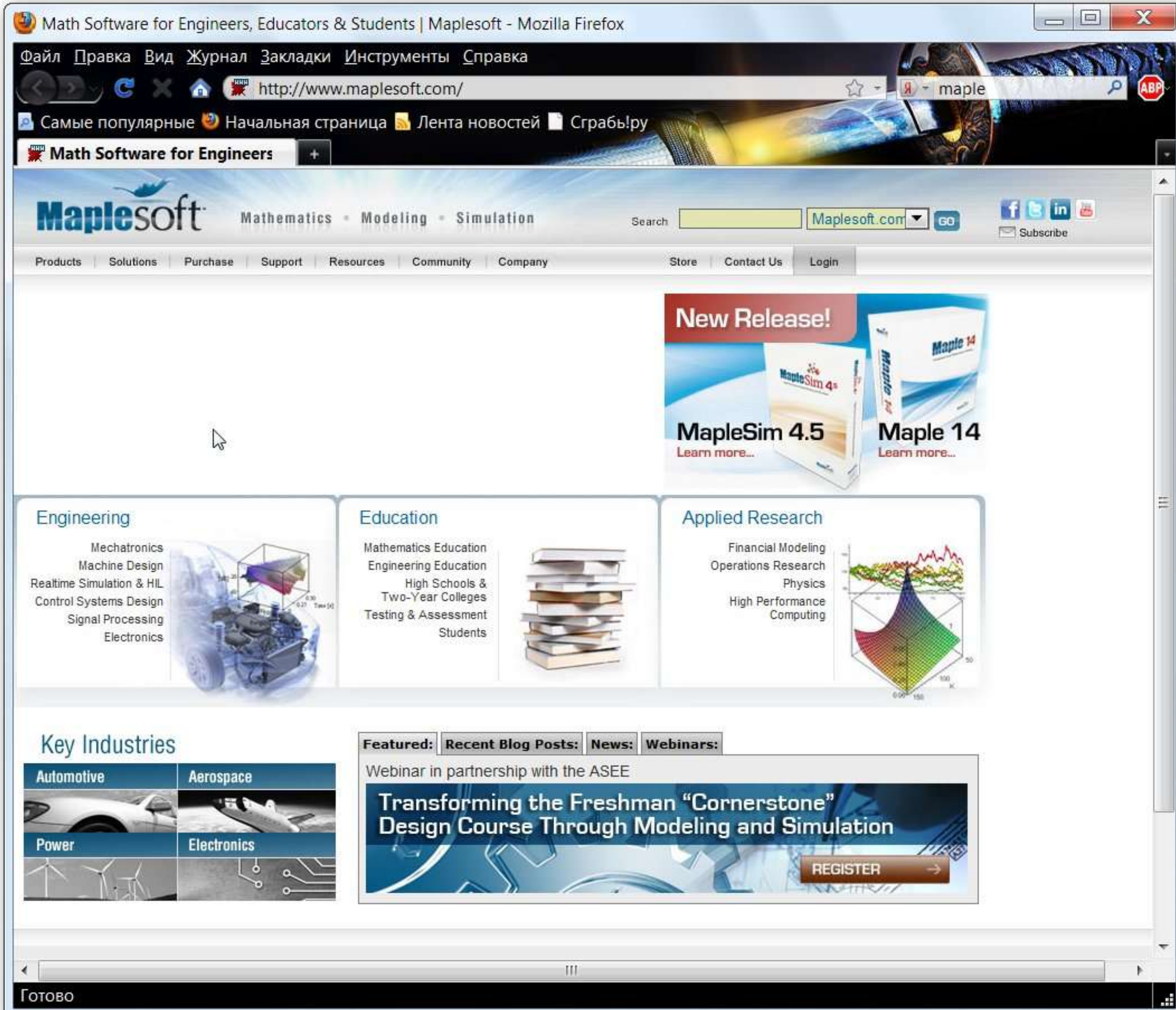
набор “палитра”

$$\text{Out[2]=}\ \frac{1}{4}\left((-1)^{j}\,p-(-1)^{2j}\,p-(-1)^{j+k}\,p+(-1)^{2j+k}\,p+8\,k\,r\right)$$

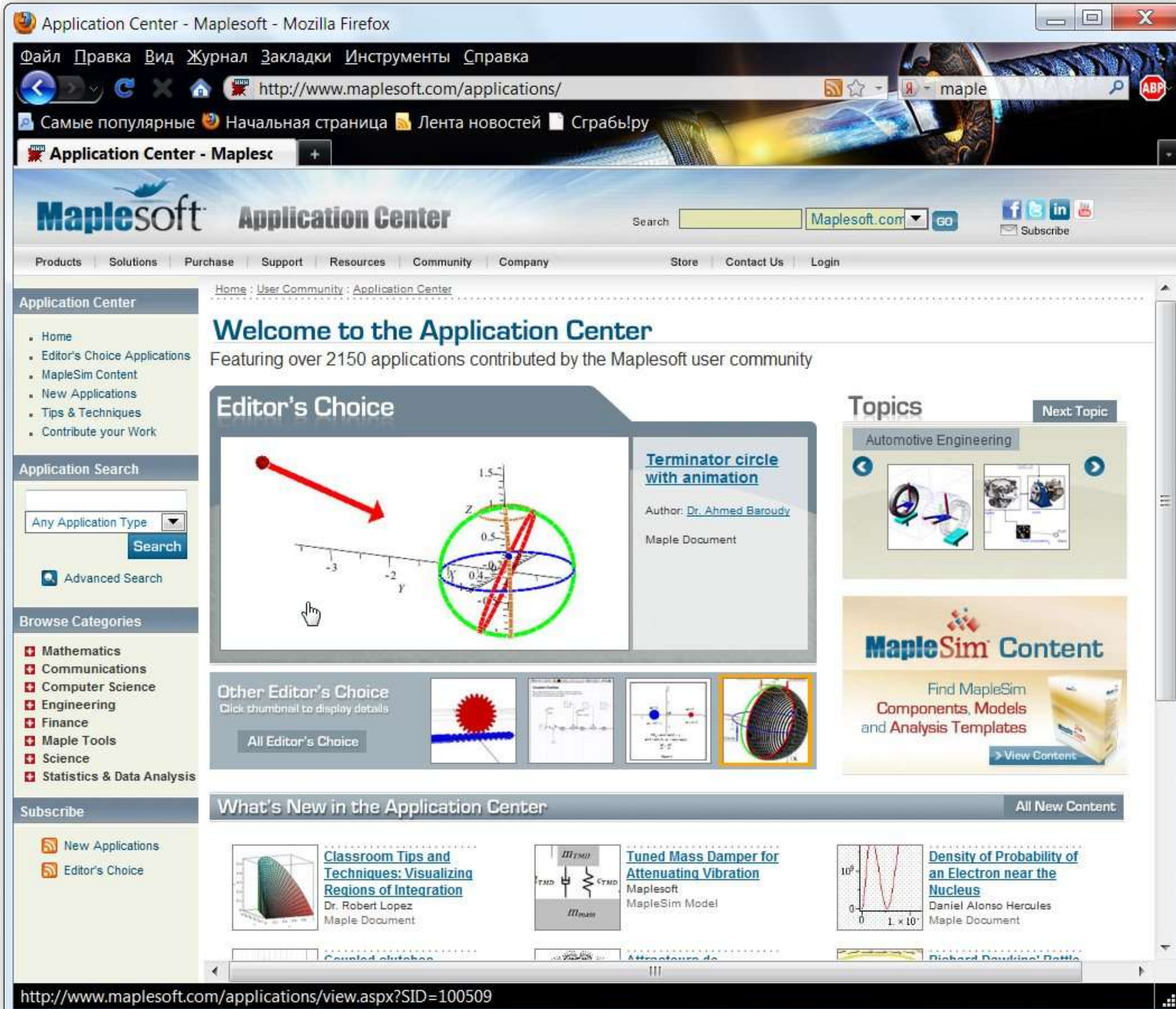
Формально три разных ответа – но это одно и тоже!!!!

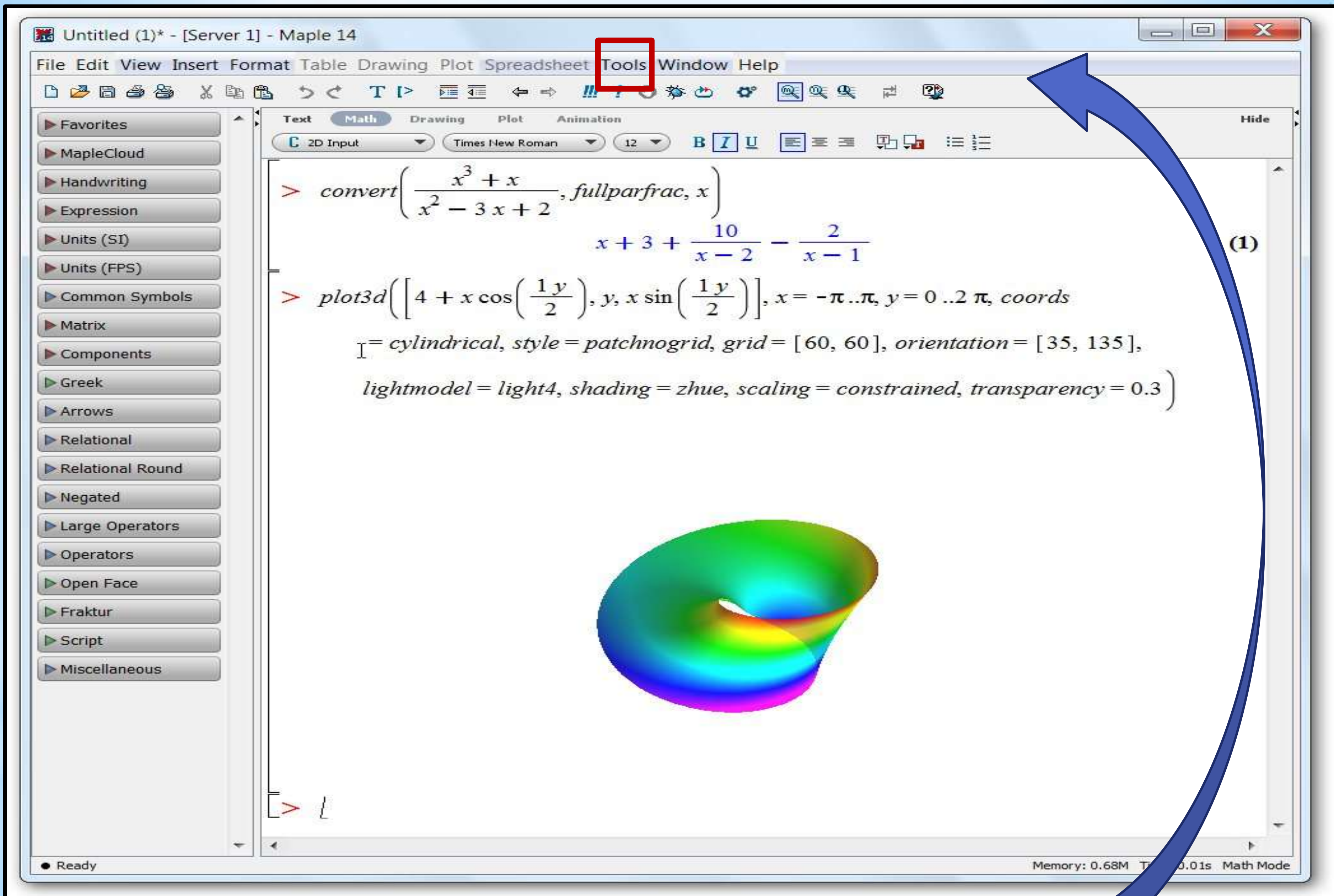

YouTube - Видео курс Wolfram Mathematica | Функция Integrate - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
http://www.youtube.com/watch?v=_KnQbwCbzrw
Itj rehc ьферуьфешсф
Самые популярные
Начальная страница
Лента новостей
Сграбь!ру
видеокурс mathematica - ...
YouTube - Видео курс W...
Поиск
Обзор
Добавить видео
PsixoGirl
Выйти
Видео
Нажмите, чтобы получать извещения о новых видео на этом канале
te
MathematicLife
Видео: 117
Подписаться
Математический портал Math-Life
www.Math-Life.com
Вычисление интегралов
f[x_] := x^2*Sin[x]+x*Log[x]
Integrate[f[x], x]
1:00 / 8:33
360p
Нравится
Добавить в
Отправить
Встроить
712 просмотров
MathematicLife
27.05.2010
4 понравилось, 0 не понравилось
Из видео Вы узнаете, как с помощью математического пакета Wolfram Mathematica...
Все комментарии (0)
Просмотреть все
Ответить на это видео...
Похожие видео
Видео курс Wolfram Mathematica | Функц.
от пользователя MathematicLife
Просмотров: 210
3:55
Calculating a Double Integral
от пользователя patrickJMT
Просмотров: 69499
8:45
Видео курс Wolfram Mathematica | Лин. системы
от пользователя MathematicLife
Просмотров: 778
6:41
Видео курс Wolfram Mathematica | Функция Series
от пользователя MathematicLife
Просмотров: 416
9:36
Bezier Curve in Wolfram Mathematica 7.0 (Кривая...
от пользователя Enfriz
Просмотров: 1396
2:03
Лекция 17.Несобственные интегралы
от пользователя NWTU
Просмотров: 2912
14:46
Лекция 19.Вычисление площади плоской фигуры
от пользователя NWTU
Просмотров: 2591
7:58
Передача данных с v7.lscache7.c.youtube.com...


Wolfram Demonstrations Project - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
http://demonstrations.wolfram.com/
Яндекс
Самые популярные
Начальная страница
Лента новостей
Сграбь!ру
Wolfram Demonstrations Pr...
Wolfram Research | Wolfram Alpha | Mathematica | MathWorld | Wolfram Science | More »
WOLFRAM WEB RESOURCES
Wolfram Demonstrations Project™
A free resource of interactive visualizations, from elementary education to the frontiers of research
SEARCH
TOPICS LATEST ABOUT FAQS PARTICIPATE AUTHORING AREA
Browse Topics
Mathematics
Algebra | Calculus & Analysis ...
Computation
Algorithms | Computer Science ...
Physical Sciences
Physics | Earth Science ...
Life Sciences
Biology | Medicine ...
Business & Social Systems
Economics | Finance ...
Systems, Models, & Methods
Discrete Models | Networks ...
Engineering & Technology
Machines | Electrical Engineering ...
Our World
Everyday Life | Geography ...
Creative Arts
Art | Architecture | Music ...
Kids & Fun
For Kids | Puzzles | Optical Illusions ...
Mathematica Functionality
Short Programs | 3D Graphics ...
NEW Mathematica 8
Player 8 coming soon!
Pioneering free-form input;
a whole new way to compute »
Featured Demonstrations
replay »
Electric Fields for Three Point Charges
Contributed by: S. M. Blinder
Other Featured Demonstrations
Rearranging Color Channels in an Image
Brake Shoes
Sequence Alignment of Words
Ikeda Delay Differential Equation
Styling Line Caps and Line Joins
Browse Random Demonstrations »
Latest Additions
??????? 9, 2011 | 6767 Demonstrations
Готово



Math Software for Engineers, Educators & Students | Maplesoft - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
http://www.maplesoft.com/
maple
Самые популярные Начальная страница Лента новостей Сграбь!ру
Math Software for Engineers
Maplesoft
Mathematics · Modeling · Simulation
Search
Maplesoft.com
GO
Subscribe
Products Solutions Purchase Support Resources Community Company
Store Contact Us Login
New Release!
MapleSim 4.5
Learn more...
Maple 14
Learn more...
Engineering
Mechatronics
Machine Design
Realtime Simulation & HIL
Control Systems Design
Signal Processing
Electronics
Education
Mathematics Education
Engineering Education
High Schools & Two-Year Colleges
Testing & Assessment
Students
Applied Research
Financial Modeling
Operations Research
Physics
High Performance Computing
Key Industries
Automotive
Aerospace
Power
Electronics
Featured: Recent Blog Posts: News: Webinars:
Webinar in partnership with the ASEE
Transforming the Freshman "Cornerstone" Design Course Through Modeling and Simulation
REGISTER
Готово


Application Center - Maplesoft - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
http://www.maplesoft.com/applications/
maple
Самые популярные Начальная страница Лента новостей Сграбь!ру
Application Center - Maplesc
Maplesoft Application Center
Search
Maplesoft.com
GO
Subscribe
Products Solutions Purchase Support Resources Community Company Store Contact Us Login
Home : User Community : Application Center
Application Center
Home
Editor's Choice Applications
MapleSim Content
New Applications
Tips & Techniques
Contribute your Work
Application Search
Any Application Type
Search
Advanced Search
Browse Categories
Mathematics
Communications
Computer Science
Engineering
Finance
Maple Tools
Science
Statistics & Data Analysis
Subscribe
New Applications
Editor's Choice
Welcome to the Application Center
Featuring over 2150 applications contributed by the Maplesoft user community
Editor's Choice
Terminator circle with animation
Author: Dr. Ahmed Baroudy
Maple Document
Other Editor's Choice
Click thumbnail to display details
All Editor's Choice
Topics
Next Topic
Automotive Engineering
MapleSim Content
Find MapleSim Components, Models and Analysis Templates
View Content
What's New in the Application Center
All New Content
Classroom Tips and Techniques: Visualizing Regions of Integration
Dr. Robert Lopez
Maple Document
Tuned Mass Damper for Attenuating Vibration
Maplesoft
MapleSim Model
Density of Probability of an Electron near the Nucleus
Daniel Alonso Hercules
Maple Document
http://www.maplesoft.com/applications/view.aspx?SID=100509



(файлы Maple, Mathematica и примеры практической работы)

Дифференцирование

Дифференцирование является алгоритмической процедурой, так как знание производных элементарных функций и следующих четырех правил:

1. **(a+b)'=a'+b' - дифференцирование суммы**
2. **(a b)'=a' b+a b' - дифференцирование произведения,**
3. **(a / b)'=(a' b-a b') / b^2 - дифференцирование отношения,**
4. **f(g(x))'=f ' (g(x)) g'(x) - дифференцирование сложной функции,**

позволяет нам продифференцировать произвольную функцию.

Фактически, настоящей **проблемой** при дифференцировании является **упрощение** результата, так как если его не упрощать, то производная от ***2x+1*** формально выдается в виде ***0x+2*1+0***.

Элементарные функции — функции, которые можно получить с помощью конечного числа арифметических действий и композиций из следующих **основных элементарных функций:**

1. **многочлен (полином)**
2. **рациональная**
3. **степенная**
4. **показательная и логарифмическая**
5. **тригонометрические и обратные тригонометрические.**

Каждую элементарную функцию можно задать формулой, то есть набором конечного числа символов, соответствующих используемым операциям.

Элементарные функции разделяются на алгебраические и трансцендентные.

Алгоритм дифференцирования достаточно прост и, поэтому, попробуем реализовать этот алгоритм для дифференцирования полиномов от одной переменной

$$P = a_n * x^n + a_{n-1} * x^{n-1} + \cdots + a_0 * x^0$$

Что нам надо уметь:

- ✓ отличать в выражении ***P*** переменную ***x*** от коэффициентов **a** и показателей степеней ***n***;
- ✓ различать математические операции - сумму, произведение и степень;
- ✓ задать правила дифференцирования.

Выделение различных частей выражения в **Maple** с помощью команды **op(i,e)** которая выделяет имя (операнд) под номером **i** из выражения **e** с учетом приоритета арифметических операций.

Определять тип переменных и выражений можно с помощью команды **type(e,t)** где **e**- выражение, а **t** - имя типа выражения.

В ответе получаем либо "правда", либо "ложь".

Типов выражений очень много и из простых типов можно составлять составные типы.

Пример:

`+` - сумма

`*` - произведение

string – строка, **symbol** – символ, **function**- функция

name^integer – имя в целой степени

anything^posint – что угодно в положительной целой степени

Процедура

```
1>  df:=proc(p::algebraic,  x::name)   local u,v;
2>    if type(p,numeric) then 0
3>    elif type(p,name) then
                 if p=x then 1 else 0 fi
4>    elif type(p,`+`) then map (df,p,x)
5>    elif type(p,`*`) then
               u:=op(1,p); v:=p/u;
               df(u,x)*v+df(v,x)*u
6>    elif type(p,anything^integer) then
               u:=op(1,p); v:=op(2,p);
               v*df(u,x)*u^(v-1)
7>    else ERROR(`Ошибка при дифференцировании полинома`,p)
      end if
    end proc:
```

В первой строке мы определяем имя процедуры **df**, типы входных данных и вводим две внутренних переменных **u,v** (это можно сделать и по умолчанию).

Во второй строке задано правило: производная от числа (тип numeric) есть ноль.

В третьей строке задано правило: производная от переменной (тип name) есть dx / dx =1 или dy / dx=0.

В четвертой строке задано правило дифференцирования суммы (тип \`+\`). Для дифференцирования слагаемых применяется та же процедура **df**, к которой мы обращаемся рекурсивно, используя команду **map** (данная команда позволяет применить процедуру **df** ко всем слагаемым сразу).

В пятой строке задано правило дифференцирования произведения (тип \`*\`). Для дифференцирования множителей, которым присваиваются имена **u** и **v**, применяется процедура **df** (в отличие от предыдущей строки мы вынуждены делать это явно)

В шестой строке задано правило дифференцирования степени переменной (тип anything^integer), при этом мы так же явно выделяем части выражения.

В седьмой строке для аварийного выхода из процедуры в случае возникновения ошибки используется команда ERROR (именно из больших букв).

Интегрирование

наука+исскуство

$$\int_0^\infty e^{-x}\,dx = 1,\ \int_0^\infty e^{-x^2}\,dx = \frac{\sqrt{\pi}}{2},$$

$$\text{and } \int_0^\infty e^{-x^3}\,dx = \Gamma\left(\frac{4}{3}\right).$$

функция Эйлера

$$\int_{-\infty}^{\infty} \frac{dx}{(e^x - x - 1)^2 + \pi^2} = \frac{1}{2}$$

функция Ламбера

$$\int_{-\infty}^{\infty} \frac{dx}{(e^x - x)^2 + \pi^2} = (1 - W(1))^{-1}$$

Загадочные интегралы

$$\int_0^\infty \operatorname{sinc}(x)\,dx = \frac{\pi}{2}$$

$$\int_0^\infty \operatorname{sinc}(x)\cdot\operatorname{sinc}(x/3)\,dx = \frac{\pi}{2}$$

$$\int_0^\infty \operatorname{sinc}(x)\cdot\operatorname{sinc}(x/3)\cdot\operatorname{sinc}(x/5)\,dx = \frac{\pi}{2}$$

$$\cdots$$

$$\int_0^\infty \operatorname{sinc}(x)\cdot\operatorname{sinc}(x/3)\cdot\operatorname{sinc}(x/5)\cdots\operatorname{sinc}(x/13)\,dx = \frac{\pi}{2}$$

$$\int_0^\infty \operatorname{sinc}(x)\cdot\operatorname{sinc}(x/3)\cdot\operatorname{sinc}(x/5)\cdots\operatorname{sinc}(x/15)\,dx = \frac{467807924713440738696537864469}{935615849440640907310521750000}\pi$$

Так же

$$\sum_{n=1}^{\infty}\prod_{k=0}^{N}\operatorname{sinc}\left(\frac{n}{2k+1}\right) = -\frac{1}{2}+\int_0^\infty\prod_{k=0}^{N}\operatorname{sinc}\left(\frac{x}{2k+1}\right)\,dx$$

Только и только при N меньше 40248.

R. Baillie, D. Borwein and J. M. Borwein, manuscript, http://users.cs.dal.ca/~jborwein/sinc-sums.pdf.

Элементарные функции по Лиувиллю

Элементарная функция y переменной x — аналитическая функция, которая может быть представлена как алгебраическая функция

$$y = \varphi(x, z_1, \dots, z_n)$$

от x и функций $z_1, \dots, z_n$ таких, что причём:

z_1 является логарифмом или экспонентой от некоторой алгебраической функции $g_1(x)$,

z_2 является логарифмом или экспонентой от некоторой алгебраической функции $g_2(x, z_1)$ и т.д.

Например, $y = \sin(x)$ — элементарная функция в этом смысле, поскольку она является алгебраической функцией e^{ix}.

Функция e^{e^x} тоже является элементарной, поскольку её можно представить в виде: $y = z_2$, где $z_2 = e^{z_1}$, где $z_1 = e^x$

Элементарные по Лиувиллю функции бесконечно дифференцируемы всюду, где они определены.

При этом производная элементарной функции всегда является элементарной функцией и может быть найдена за конечное число действий, так как по правилу дифференцирования сложной функции производная равна

$$y'(x) = \frac{d}{dx}\,\varphi(x, z_1, \dots, z_n) = \frac{\partial \varphi}{\partial x} + \sum_{i=1}^{n} \frac{\partial \varphi}{\partial z_i} * \frac{dz_i}{dx}$$

На практике обычно используют таблицу производных, а не так как мы делали при дифференцировании полиномов.

Интеграл элементарной по Лиувиллю функции не всегда сам является элементарной функцией. Наиболее распространённые функции, интегралы которых найдены, собраны в таблице интегралов.

В общем случае имеет место теорема Лиувилля:

Если интеграл от элементарной функции $\boldsymbol{y = \varphi(x, z_1, \dots, z_n)}$ сам является элементарной функцией, то он представим в виде

$$\boldsymbol{\int \varphi(x, z_1(x), \dots, z_n(x))\,dx = \sum_i^N A_i * \ln \vartheta_i(x, z_1, \dots, z_n) + \vartheta_0(x, z_1, \dots, z_n) + C}$$

где $\boldsymbol{A_i}$, $\boldsymbol{C}$ – комплексные числа, а $\boldsymbol{\vartheta_i(x, z_1, \dots, z_n)}$ - алгебраические функции своих аргументов.

Результаты Лиувилля, при всей своей значимости, совершенно не помогают в задаче нахождения первообразной, ибо процедуры приведения функции к интегрируемому виду автором предложено не было, более того Г. Х. Харди предположил, что такой процедуры вообще не может существовать.

Дифференциальная **теория Галуа** — позволила обобщить результат Лиувилля на произвольные элементарные функции и, более того, сделать доказательство конструктивным, т.е. таким, что с его помощью стало возможно получить желанную первообразную.

Роберт Риш из Калифорнийского университета в **1969-1970** годах опубликовал алгоритм, приводящий любую элементарную функцию к необходимому для интегрирования виду или определяющий, что такое приведение невозможно (и следовательно, интеграла как элементарной функции нет). На современных компьютерах эти конструкции были незамедлительно запрограммированы.

Первая реализация алгоритма Риша была выполнена **Джоэлом Мозесом** в рамках знаменитого «Project MAC» в MIT в 1971 г.

Дэвенпорт в 1981 г., основываясь на работе Риша и некоторых глубоких результатах дифференциальной алгебры и комплексного анализа, разработал алгоритм интегрирования чисто алгебраических функций и реализовал его в известной среде символьных вычислений **REDUCE**-2.

Программы Мозеса и Дэвенпорта ***для общего случая — интегрирования произвольных элементарных функций — были одинаково непригодны***.

Алгоритм Дэвенпорта ещё и обладал большой вычислительной сложностью, что его автор признавал и выражал надежду, что алгоритм можно упростить. Так и произошло: Барри Трагер из MIT в 1984 г. внёс серьёзные улучшения в алгоритм Дэвенпорта. Обновлённый алгоритм обладал гораздо большей практической ценностью и был реализован в математических программах **Axiom** и **Maple**.

Решающий шаг к практическому решению вопроса сделал Мануэль Бронштейн в 1990 г, обобщив алгоритм Трагера на произвольные элементарные функции.

Бронштейн был одним из ведущих разработчиков **Axiom**. Вместе с Трагером он реализовал в Axiom свой алгоритм, но лишь частично. В 2005 году Бронштейн скончался, спустя два года после смерти основателя **Axiom** Ричарда Дженкса.

При своей неполноте, имплементация Бронштейна — наиболее полная из имеющихся в программах символьных вычислений.

При возникновении нереализованного случая **Axiom** выдаёт ошибку, в отличие от процедур интегрирования **Maple** и **Mathematica**. То есть, если **Axiom** обнаруживает, что интеграл от данной функции не является элементарной функцией, то это правда. Если **Axiom** выдаёт ошибку, то интеграл может быть или не быть элементарной функцией.

Известны несложные примеры, показывающие несостоятельность процедур символьного интегрирования в современных математических пакетах.

Например, первообразная функции

$$\frac{x}{\sqrt{x^4+10x^2-96x-71}}$$

равна

$$-\frac{1}{8}\ln((x^6+15\,x^4-80\,x^3+27\,x^2-528\,x+781)\sqrt{x^4+10x^2-96x-71}-x^8-20\,x^6+128\,x^5-54\,x^4+1408\,x^3-3124\,x^2-10001)$$

однако в таком виде её находит только **Axiom**.

Прочие пакеты — **Mathematica, Maple, Matlab, Maxima, Reduce** — в лучшем случае выдают громоздкий ответ с участием эллиптических интегралов, т.е. неспособны выдать ответ в виде элементарной функции

Теорема Ричардсона:

Если есть функция f, получающаяся с помощью сложения, умножения и композиции из синуса, экспоненты и модуля, а коэффициенты f используют рациональные числа, π и $ln(2)$, то задача проверки утверждения «f тождественно равна нулю» алгоритмически неразрешима.

Функция модуля существенна в доказательстве Ричардсона, и есть основания полагать (да и сам автор теоремы надеется), что без неё задача Constant Problem всё таки разрешима, что даёт надежду на создание абсолютно честного алгоритма символьного интегрирования.

Constant Problem $$f = \frac{dY}{dx} - y = 0$$ $$Y = \int y dx$$

Класс элементарных функций по Лиувиллю кажется несколько надуманным.

Например, функция ошибки

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

не является элементарной, как доказал Лиувилль (гидродинамика, Прандтль).

Но чем она хуже экспоненты и логарифма, которые мы относим к элементарным функциям? На компьютере она может быть вычислена с произвольной точностью, и даже, как экспонента и логарифм, обладает элементарной производной.

Эту функцию вполне можно включить в «надэлементарные» расширения функций, так же как и эллиптические интегралы, и многие другие функции.

Литература:

1. Richard H. Enns, George C. McGuire, Nonlinear Physics With Maple for Scientists and Engineers. ISBN 0-8176-4119-X
2. Jon H. Davis, Differential Equations With Maple: An Interactive Approach. ISBN 0-8176-4181-5
3. Martha L. Abell, James P. Braselton, Differential Equations with Maple V. ISBN 0-12-041560-7
4. Franco Vivaldi, Experimental Mathematics with Maple. ISBN 1-58488-233-6
5. Ronald L. Greene, Classical Mechanics With Maple. ISBN 0-387-94512-1
6. Stephen Lynch, Dynamical Systems with Applications using Maple. ISBN 0-8176-4150-5
7. John F. Putz Maple Animation. 2003. ISBN 1-58488-378-2