Obliczenia Naukowe

Wykład 11:Pakiety do obliczeń: naukowych i inżynierskich Przegląd i porównanie

Bartek Wilczyński

21.5.2018

Plan na dziś

- Pakiety do obliczeń: przegląd zastosowań
- różnice w zapotrzebowaniu: naukowcy, inżynierowie, statystycy/medycy
- Matlab/octave/scipy – obliczenia numeryczne
- Mathematica/Maxima/Sympy – obliczenia symboliczne
- Sage i SageCloud – zintegrowany pakiet opensource
- Excel?

Typowi użytkownicy pakietów obliczeniowych

- Inżynierowie i projektanci (budownictwo, lotnictwo, motoryzacja, itp.)
- Naukowcy doświadczalni (fizycy, chemicy, materiałoznawcy, itp.)
- Statystycy (zastosowania w medycynie, ekonomii, biologii molekularnej, psychologii, socjologii, ubezpieczeniach, itp.)
- Matematycy (przede wszystkim matematyka stosowana)

Obliczenia naukowe

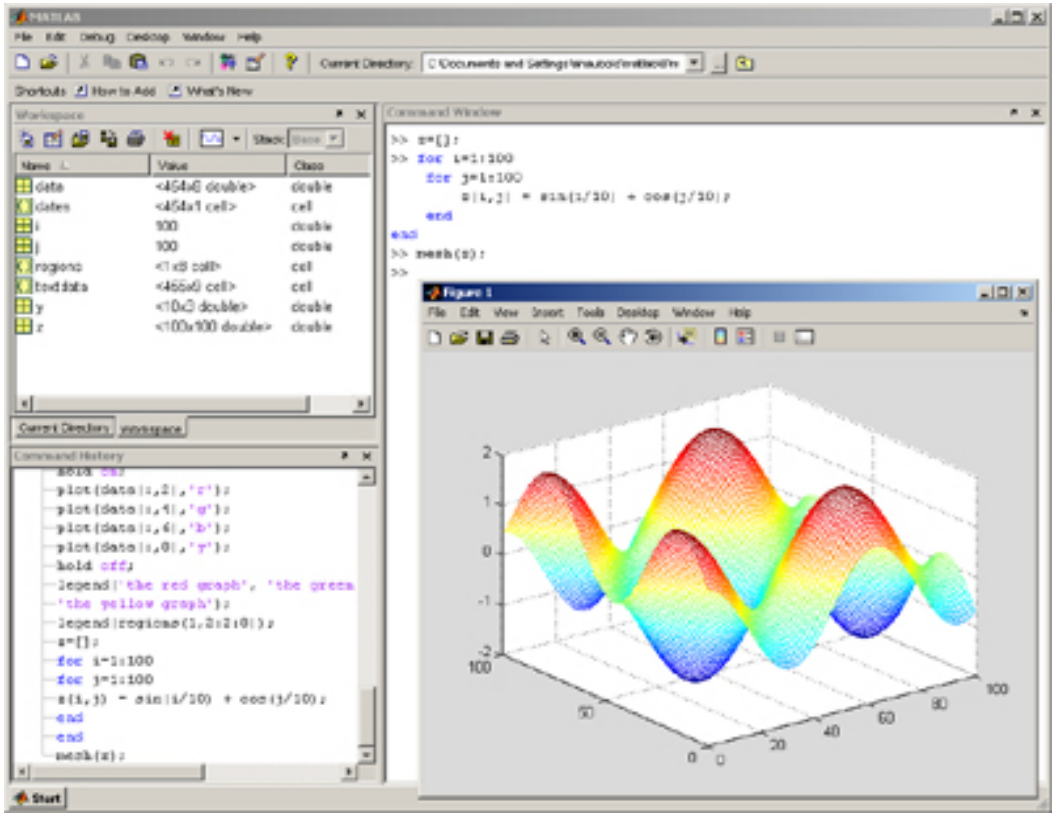
- Komputer jako “potężniejszy kalkulator”
- W zasadzie wszystko można zaprogramować samemu, ale każdemu mogą się przydać:
  - Interfejs użytkownika łatwiejszy niż typowego kompilatora
  - Możliwość zaawansowanej grafiki
  - Dobrze przetestowane standardowe procedury
  - Interfejsy do urządzeń
  - Wsparcie fachowców

Matlab i pakiety “inżynierskie”

- Rozwijany w latach 70'tych przeze Cleve Moler'a jako narzędzie dla studentów informatyki, aby nie musieli używać zaawansowanych bibliotek fortranu
- Firma mathworks powstaje w 1984 i wydaje pierwszą wersję Matlab'a
- Najpopularniejszy wśród inżynierów, dobre całki numeryczne, rozwiązywanie równań i wykresy (również 3d)
- Bardzo popularny także do przetwarzania sygnałów i symulacji (simulink)
- Licencja komercyjna – niedrogi dla studentów, droższy dla uczelni, bardzo drogi dla przemysłu

Toolbox'y Matlab'a

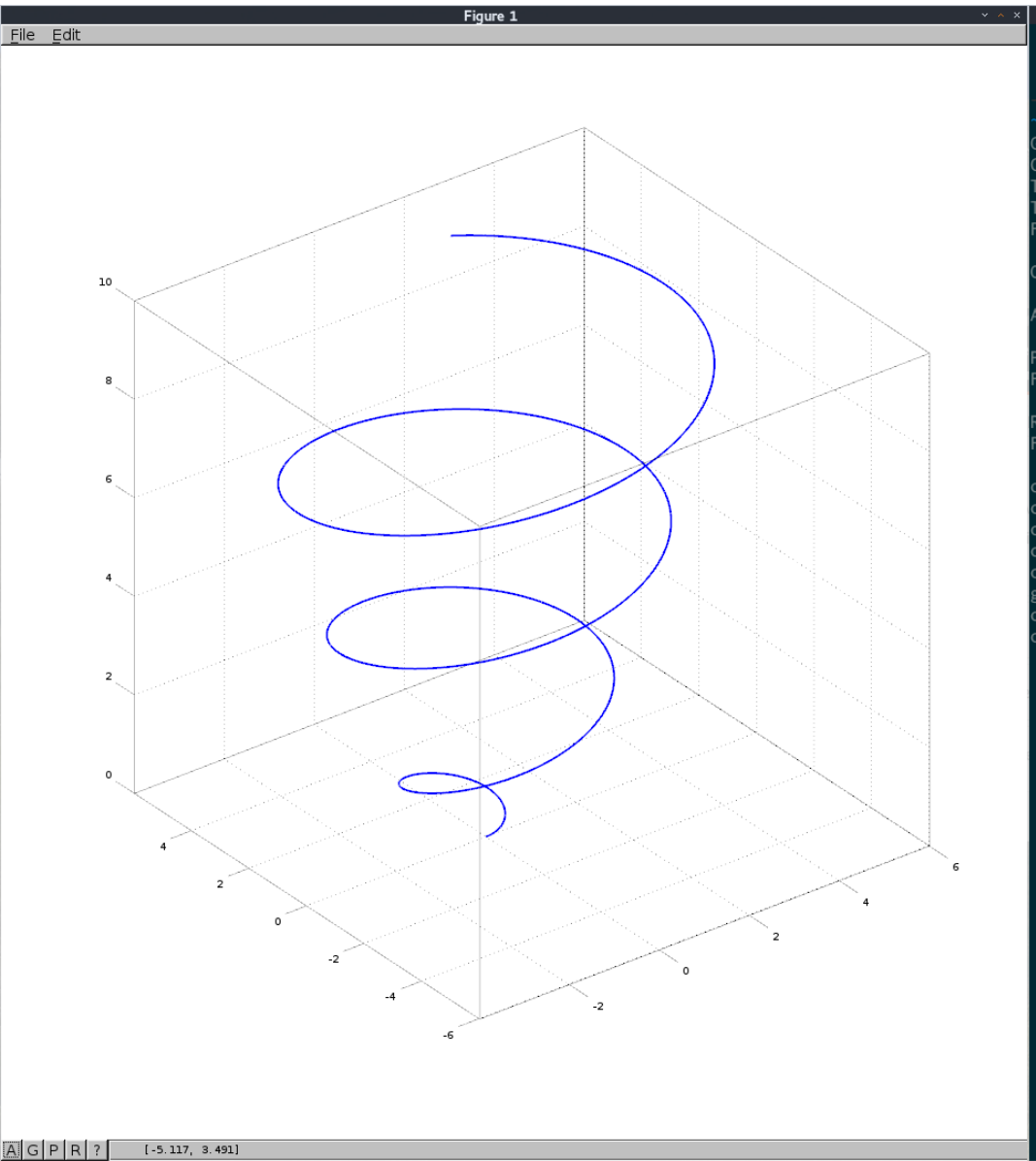
- Wiele dodatkowych (płatnych) bibliotek dla specjalistów
  - Symbolic math
  - Image processing
  - Financial toolbox
  - Bioinformatics
  - Optimization
  - SimBiology

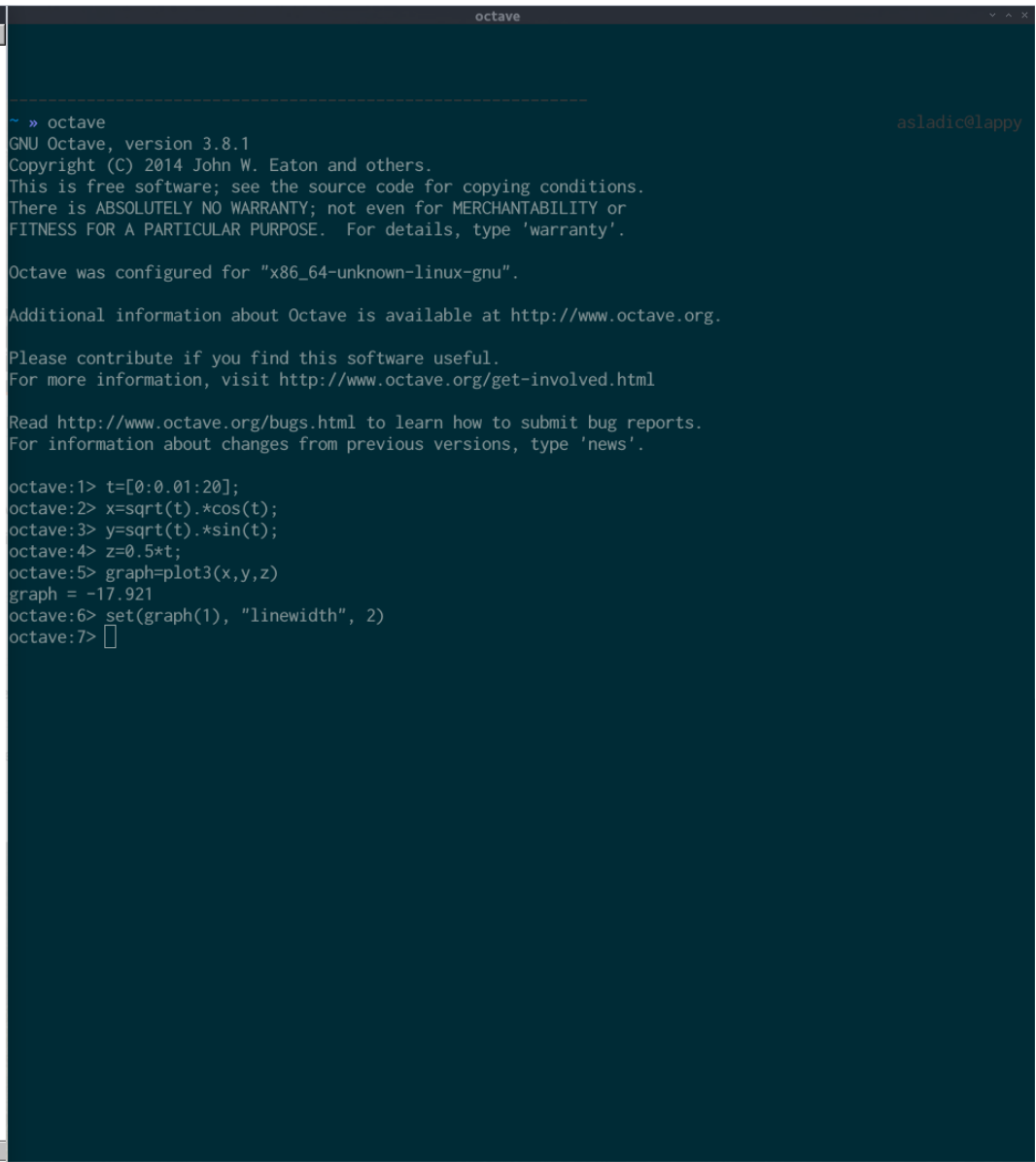


Alternatywy openSource

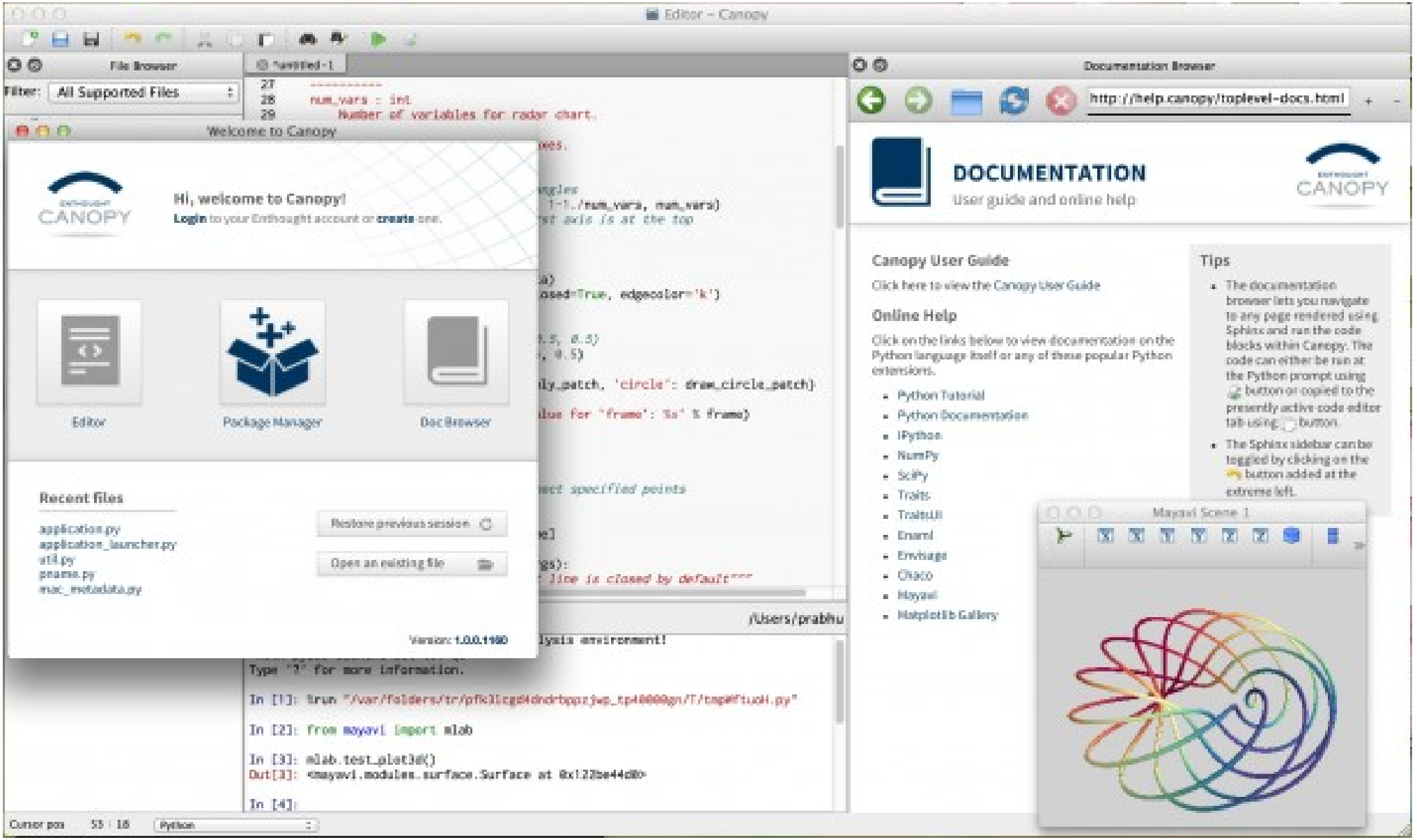
- GNU Octave (rozpoczęty w 1988, wydania od 1992, rozwijany przez John'a W. Eatona, chemika z University of Wisconsin-Madison)
  - W zasadzie kompatybilny z Matlab'em
  - John W. Eaton Inc. - consulting
- Scipy stack – zestaw bibliotek python'a do obliczeń naukowych
  - Wiele bibliotek, rozwijanych przez niezależne grupy
  - System pakietów, edytor i dystrybucja organizowana przez firmę Enthought, również komercyjne dystrybucje i konsulting
  - Wiele konferencji tematycznych dla naukowców I pracowników przemysłu - także źródło dochodu

Interfejs Octave





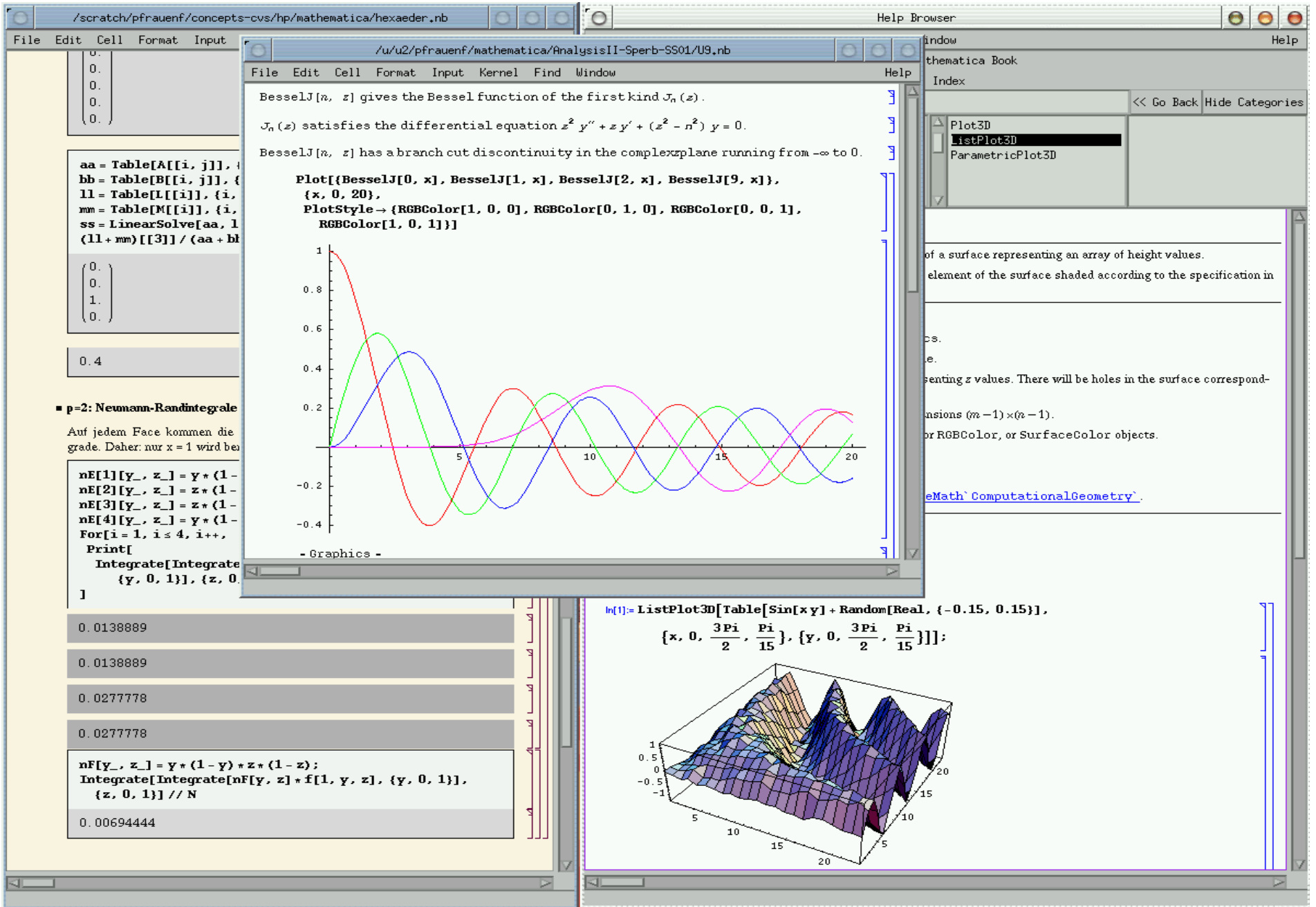
Interfejs Enthought Canopy



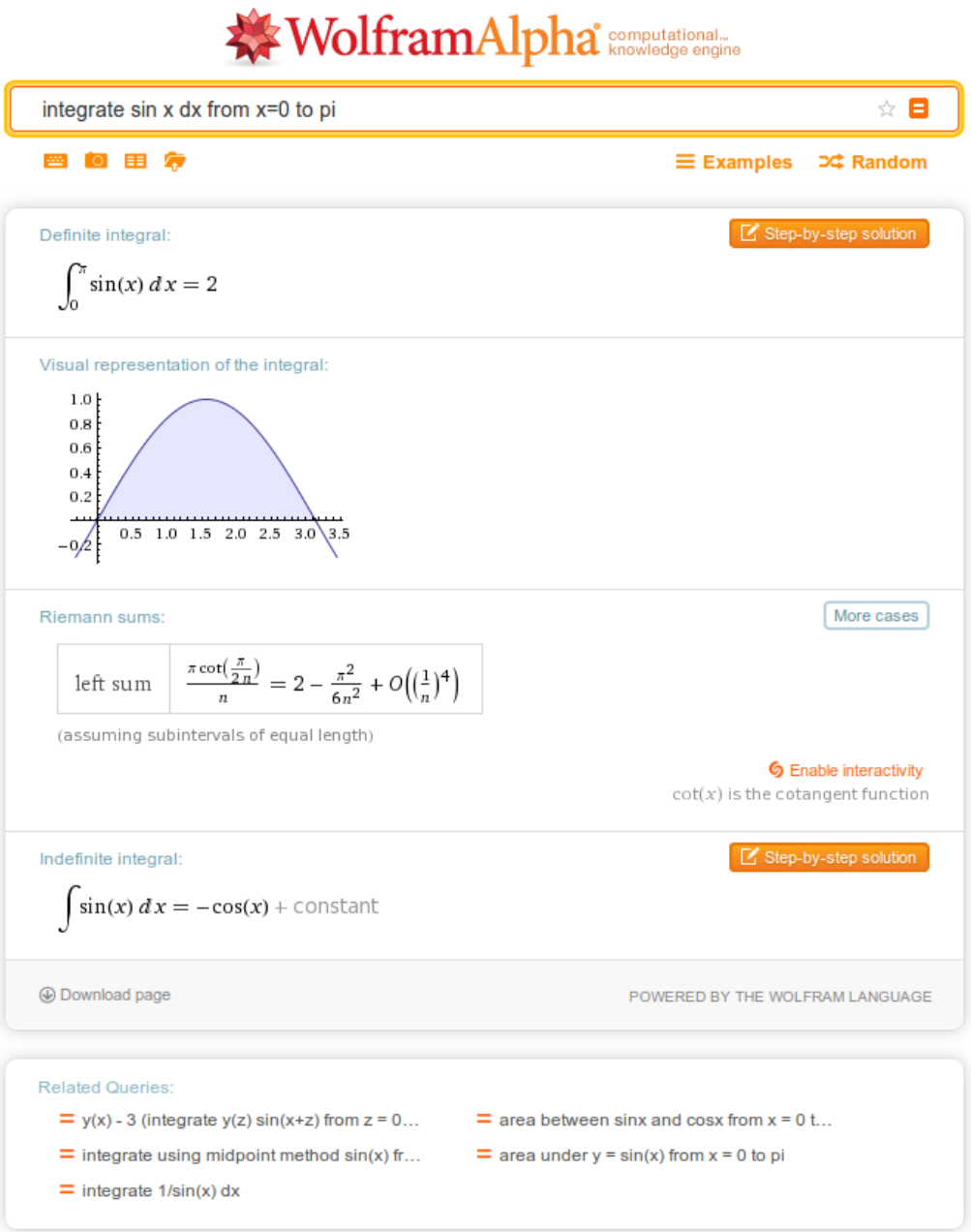
Obliczenia symboliczne – Mathematica i podobne

- Opracowana w latach 1980'tych przez Stephen'a Wolframa
- Jeden z pierwszych w historii pakietów umożliwiających obliczenia symboliczne
- Bardzo popularna wśród studentów amerykańskich, którzy muszą “zaliczyć” rachunek różniczkowy
- Obecnie także w wersji online: Wolfram Alpha
- Konkurencyjne pakiety: Maple, Mathcad, Symbolic math toolbox w matlabie (dawny muPAD)

Mathematica - interfejs



Wolfram Alpha – mathematica online



- Interfejs online umożliwiający korzystanie z wielu narzędzi do obliczeń symbolicznych
- Duża część funkcjonalności darmowa, ale wiele funkcji (np. rozpisywanie rozwiązania na kroki) - płatna

Obliczenia numeryczne a symboliczne

- W obliczeniach symbolicznych próbujemy obejść problem numerycznych zaokrągleń i przybliżeń poprzez opis równań algebraicznych explicite
- Tego typu pakiety pozwalają na dokładne odwzorowanie równań, jednak cierpią z powodu heurystycznych metod rozwiązania
- Proste operacje w takich pakietach są prostsze niż “na kartce” ale trudne często mogą nastręczać więcej problemów niż korzyści
- Na pewno są skuteczne do sprawdzania, czy nie pomyliliśmy się w obliczeniach

Typowe funkcje pakietów symbolicznych

- Przekształcanie, upraszczanie wzorów
- Rozwiązywanie równań i układów równań algebraicznych
- Znajdowanie granic wyrażeń i ciągów liczbowych
- Całkowanie i różniczkowanie symboliczne
- Wykresy
- Ładne formatowanie wzorów matematycznych (często przy użyciu LaTeX'a)

Maxima - Obliczenia symboliczne Open Source

- Maxima (1992-), a wcześniej Macsyma (1968-1982)
- Wydana w 1998 na licencji GPL
- Napisana w języku lisp
- Wiele konkurencyjnych interfejsów (WXMaxima, Gmaxima itp)
- Skupiona na obliczeniach symbolicznych

Projekt SymPy

- Projekt narzędzi do obliczeń symbolicznych dla języka python
- Powstaje od ok. 2005 roku, obecnie osiągnął wersję 1.0
- Napisany w pythonie, kładzie nacisk na czytelność kodu i rozszerzalność, niekoniecznie na szybkość i pełność systemu
- Zawiera podstawowe funkcjonalności (zmienne symboliczne, granice, równania, różniczkowanie, całkowanie symboliczne)
- Dobrze integruje się z innymi pakietami w pythonie

Przykład użycia sympy

https://notch.mimuw.edu.pl:8888/notebooks/Symbolicznie.ipynb

jupyter Symbolicznie Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

In [37]:
```
import sympy,math
print math.sqrt(8),sympy.sqrt(8)
print math.sqrt(8)**2, sympy.sqrt(8)**2
sympy.Rational(1,3),1./3
```

```
2.82842712475 2*sqrt(2)
8.0 8
```

Out[37]: (1/3, 0.3333333333333333)

In [26]:
```
from sympy import symbols
x, y = symbols('x y')
expr = x + 2*y # wyrażenia symboliczne
print expr
print expr - x**2
```

```
x + 2*y
-x**2 + x + 2*y
```

In [30]:
```
from sympy import expand, factor #wymnażanie
expanded_expr = expand(x*expr)
print x*expr, expanded_expr
```

```
x*(x + 2*y) x**2 + 2*x*y
```

In [31]:
```
factor(expanded_expr) # grupowanie
```

Out[31]: x*(x + 2*y)

In [19]:
```
sympy.diff(sin(x)*exp(x), x) # różniczkowanie
```

Out[19]: exp(x)*sin(x) + exp(x)*cos(x)

In [34]:
```
sympy.integrate(exp(x)*sin(x) + exp(x)*cos(x), x) #całkowanie
```

Out[34]: exp(x)*sin(x)

In [35]:
```
sympy.limit(sin(x)/x, x, 0) # granice
```

Out[35]: 1

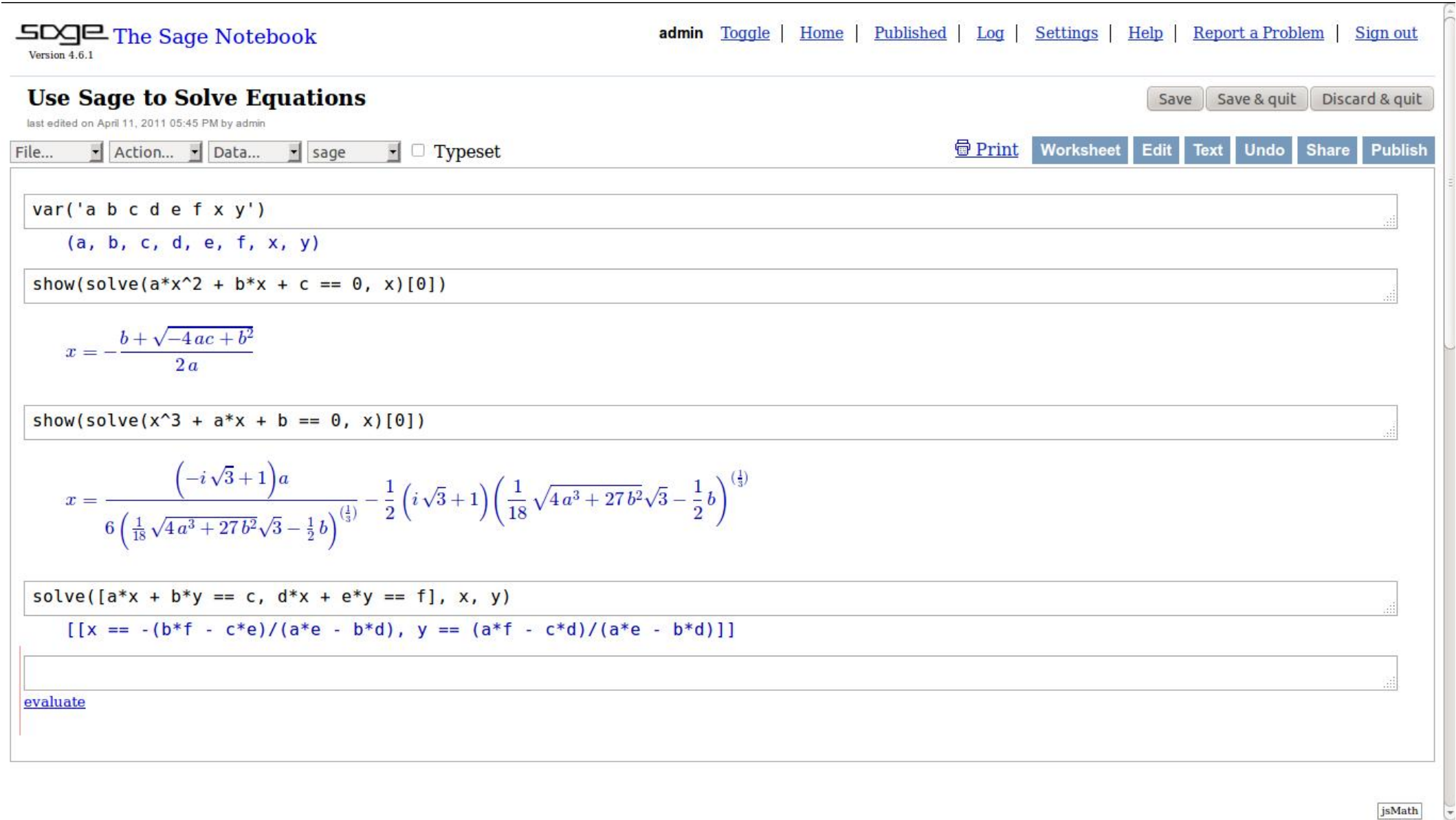
In [22]:
```
from sympy import latex
latex(Integral(cos(x)**2, (x, 0, pi))) # wypisywanie w LaTeX'u
```

Out[22]: '\\int_{0}^{\\pi} \\cos^{2}{\\left (x \\right)}\\, dx'

SAGE notebook

- Stosunkowo nowy projekt
- Połączenie wielu środowisk obliczeniowych
  - Python (Numpy, Scipy, Sympy, matplotlib, Networkx)
  - Maxima
  - R
  - GAP, FLINT, GD, JMOL, PALP, Singular
- Środowisko w przeglądarce, sesja na serwerze lub “w chmurze”

Interfejs SAGE



aaa - SageMathCloud - Mozilla Firefox

aaa - SageMathCloud

https://cloud.sagemath.com/projects/bc74

Search

Projects | aaa | Bartek Wilczynski | About | 133ms

Files | New | Log | Find | Settings | 2016-05-30-002158.sagews

Help | in | out

Modes | Help | # | Data | Control | Program | %sage

x | Plots | Calculus | Linear | Graphs | Number Theory | Rings

```
1
2
3  integrate(1 + x + x^2, x)
```

```
4    1/3*x^3 + 1/2*x^2 + x
```

```
5
6
7  show(integrate(1 + x + x^2, x))
8  %var x, theta
```

9

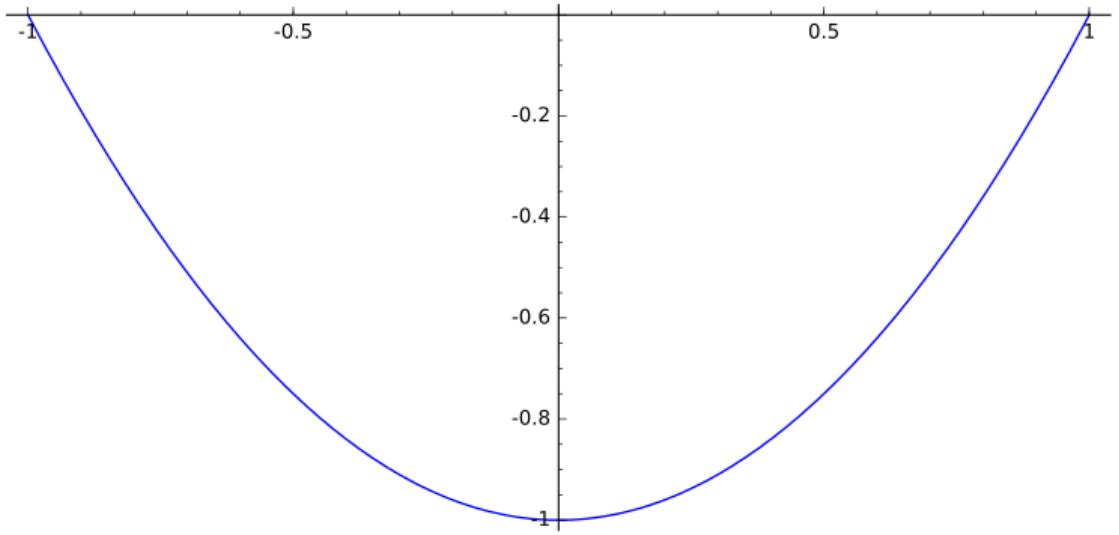
$$\frac{1}{3}x^3 + \frac{1}{2}x^2 + x$$

```
10
11 solve(x**2-1,x)
12
```

```
13   [x == -1, x == 1]
```

```
14
15
16 plot(x**2-1)
```

17




```
18
```

Jupyter notebook

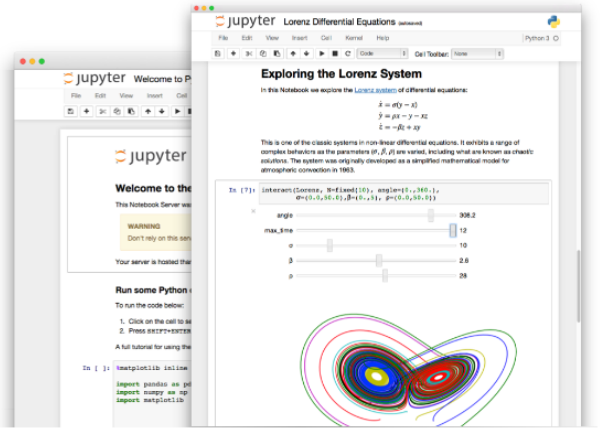
Project Jupyter | H

jupyter.org

Bartek

jupyter

Install About Resources Documentation NBViewer Widgets Blog Donate



The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.

Language of choice

The Notebook has support for over 40 programming languages, including those popular in Data Science such as Python, R, Julia and Scala.

Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.

Interactive widgets

Code can produce rich output such as images, videos, LaTeX, and JavaScript. Interactive widgets can be used to manipulate and visualize data in realtime.

Big data integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, dplyr, etc.

Excel?

- Najpopularniejszy pakiet do obliczeń
- Bardzo prosty interfejs
- Często stosowany również w bio-informatyce
- Ma spore ograniczenia (np. Maksymalna liczba linii w arkuszu), które utrudniają rozwój projektów prowadzonych w arkuszu
- Brak możliwości efektywnego testowania,
- Brak debuggerów
- Ma wiele funkcji, które warto znać, zwłaszcza, że często dane do obróbki dostajemy właśnie w Excel'u

Ziemann et al. Genome Biology (2016) 17:177
DOI 10.1186/s13059-016-1044-7

Genome Biology

COMMENT Open Access

Gene name errors are widespread in the scientific literature

Mark Ziemann[1], Yotam Eren[1,2] and Assam El-Osta[1,3*]

Abstract

The spreadsheet software Microsoft Excel, when used with default settings, is known to convert gene names to dates and floating-point numbers. A programmatic scan of leading genomics journals reveals that approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions.

Keywords: Microsoft Excel, Gene symbol, Supplementary data

Abbreviations: GEO, Gene Expression Omnibus; JIF, journal impact factor

The problem of Excel software (Microsoft Corp., Redmond, WA, USA) inadvertently converting gene symbols to dates and floating-point numbers was originally described in 2004 [1]. For example, gene symbols such as *SEPT2* (Septin 2) and *MARCH1* [Membrane-Associated Ring Finger (C3HC4) 1, E3 Ubiquitin Protein Ligase] are converted by default to '2-Sep' and '1-Mar', respectively. Furthermore, RIKEN identifiers were described to be automatically converted to floating point numbers (i.e. from accession '2310009E13' to '2.31E+13'). Since that report, we have uncovered further instances where frequently reused. Our aim here is to raise awareness of the problem.

We downloaded and screened supplementary files from 18 journals published between 2005 and 2015 using a suite of shell scripts. Excel files (.xls and.xlsx suffixes) were converted to tabular separated files (tsv) with ssconvert (v1.12.9). Each sheet within the Excel file was converted to a separate tsv file. Each column of data in the tsv file was screened for the presence of gene symbols. If the first 20 rows of a column contained five or more gene symbols, then it was suspected to be a list of gene symbols, and then a regular expression (regex) search of the entire column was applied to identify gene symbol errors. Official gene symbols from Ensembl version 82, accessed November 2015, were obtained for *Arabidopsis thaliana*, *Caenorhabditis elegans*, *Drosophila melanogaster*, *Danio rerio*, *Escherichia coli*, *Gallus gallus*, *Homo sapiens*, *Mus musculus*, *Oryza sativa* and *Saccharomyces cerevisiae* [2]. The regex search used was similar to that described previously by Zeeberg and colleagues [1], with the added screen for dates in other formats (e.g. DD/MM/YY and MM-DD-YY). To expedite analysis of supplementary files from multi-disciplinary journals, we limited the articles screened to those that have the keyword 'genome' in the title or abstract (*Science*, *Nature* and *PLoS One*). Excel files (.xls and.xlsx) deposited in NCBI Gene Expression Omnibus (GEO) [3] were also

Table 1 Results of the systematic screen of supplementary Excel files for gene name conversion errors

Journal[a]	Number of Excel files screened	Number of gene lists found	Number of papers with gene lists	Number of supplementary files affected	Number of papers affected	Number of gene names converted
PLoS One	7783	2202	994	220	170	4240
BMC Genomics	11464	1650	801	218	158	4932
Genome Res	2607	580	251	114	68	3180
Nucleic Acids Res	2117	540	315	88	67	1661
Genome Biol	2678	664	257	97	63	1878
Genes Dev	932	395	190	75	55	1593
Hum Mol Genet	980	372	168	48	27	1724
Nature	482	150	74	27	23	1375
BMC Bioinformatics	1790	235	152	26	21	534
RNA	569	127	77	20	15	1341
Nat Genet	264	70	37	12	9	178
Bioinformatics	731	112	67	11	6	339
PLoS Comput Biol	177	79	32	6	6	46
PLoS Biol	143	54	29	7	5	206
Mol Biol Evol	995	112	79	7	4	56
Science	172	36	19	7	3	451
Genome Biol Evol	490	32	25	2	2	121
DNA Res	801	57	30	2	2	6
Total	35175	7467	3597	987	704	23861

[a]The 18 journals investigated are ordered by the number of papers affected by gene name conversion errors