Use of Open-source Mathematics Software in degree level courses at Sheridan College Content delivery, assessment and evaluation

Victor Ralevich, Ph.D.
Sheridan College, Oakville, Ontario, Canada

THE ROLE OF TECHNOLOGY IN ASSESSMENT AND EVALUATION OF MATHEMATICS LEARNING
FIELDS MATHEMATICS EDUCATION FORUM
February 25, 2012, Fields Institute, 222 College Street, Toronto

Sheridan

Introduction

Brief overview of mathematics software, with the emphasis on use of free and open-source software platform SAGE in some of the advanced courses taught at the Sheridan College, Ontario.

We use SAGE for:

Meaningful non-trivial exercises courses such as:

- Algorithms and data structures
- CPU architecture
- Introduction to Cryptology
- Advanced Cryptology, etc.

Bachelor of Applied Information Sciences (Information Systems Security) program includes in its curriculum:

- *Number theory* (divisibility, primality testing, Euler totient function, congruencies, simultaneous congruency equations, pseudoprimality testing on large numbers)

- *Abstract algebra* (groups, rings, integral domains, fields, finite, polynomials over finite fields)

- *Complexity theory*

- *Information theory*

Particularly sensitive and complex topics which cannot be covered properly without use of more mathematics software:

- Efficient implementation of multiple precision arithmetic computation with large integers, and in Z_m
- Use in RSA, AES, ECC, ElGamal and other cryptographic algorithms
- Fast exponentiation of large integers (modulo n)
- Probabilistic primality testing (Miller-Rabin, etc)
- Discrete fast Fourier transforms

What Math Software is Available and Useful?

Magma

Software to Solve Computationally Hard Problems in Pure Mathematics

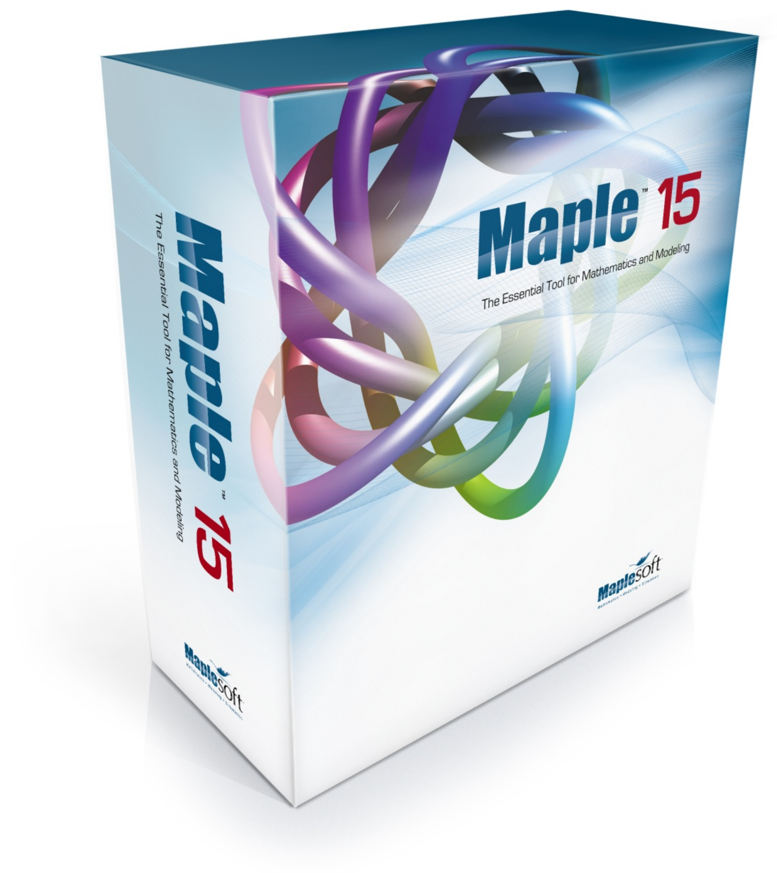
Origins: 1973 as Cayley, then renamed Magma in 1993. University setting – not a company. **Not for profit.**

Mission: "Develop computer techniques for solving symbolic problems in mathematics, with particular emphasis on the areas of algebra, number theory and geometry."

Funding: Australian Research Council and License Fees
List price: The educational price is $1150 per copy.

Sheridan

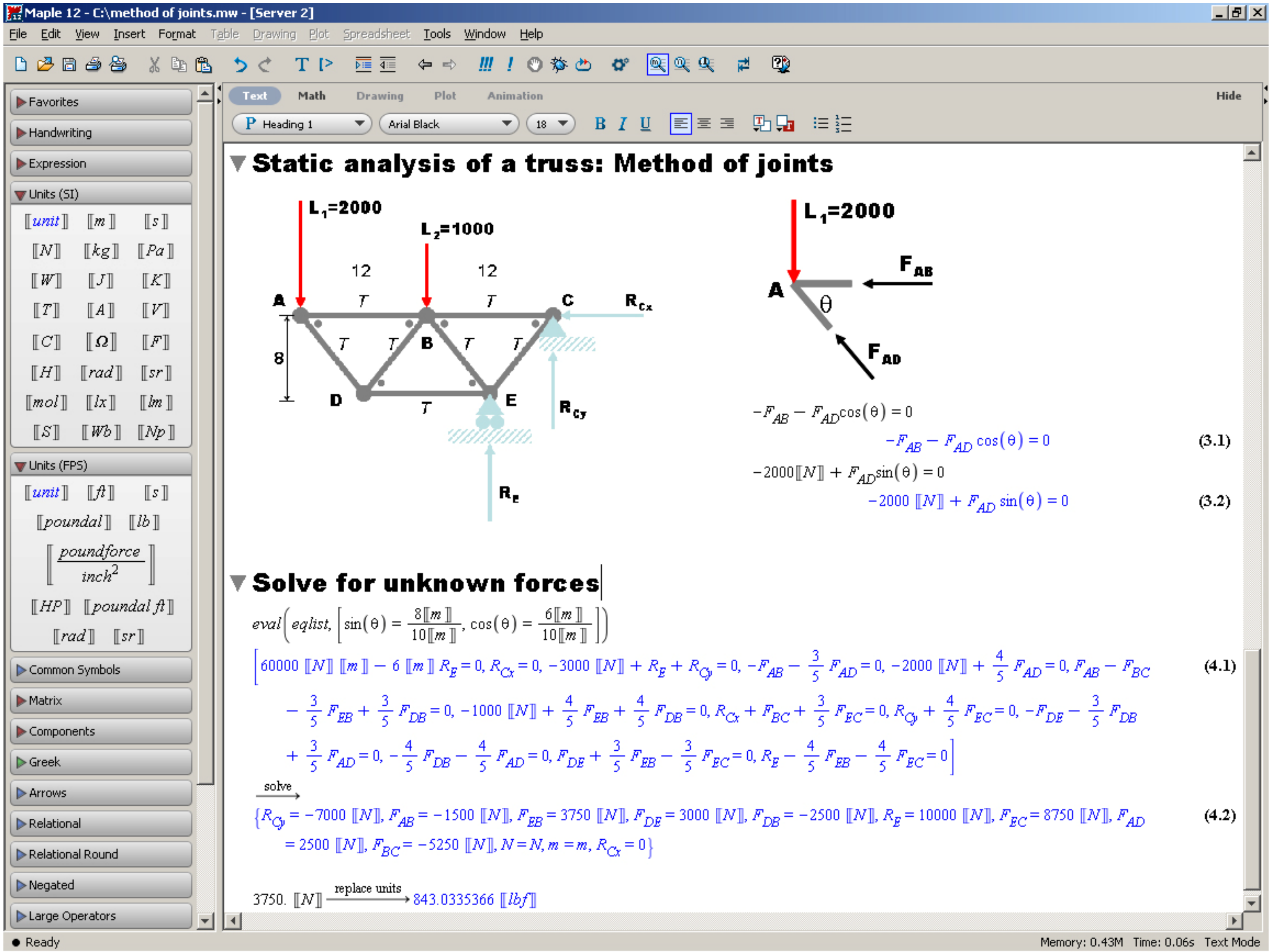
Maple



Maple is a general-purpose commercial computer algebra system. It was first developed in 1980 by the Symbolic Computation Group at the University of Waterloo in Waterloo, Ontario, Canada.

Since 1988, it has been developed and sold commercially by Waterloo Maple Inc. (also known as Maplesoft), a Canadian company also based in Waterloo, Ontario. The current major version is version 15 which was released in April 2011.

Sheridan

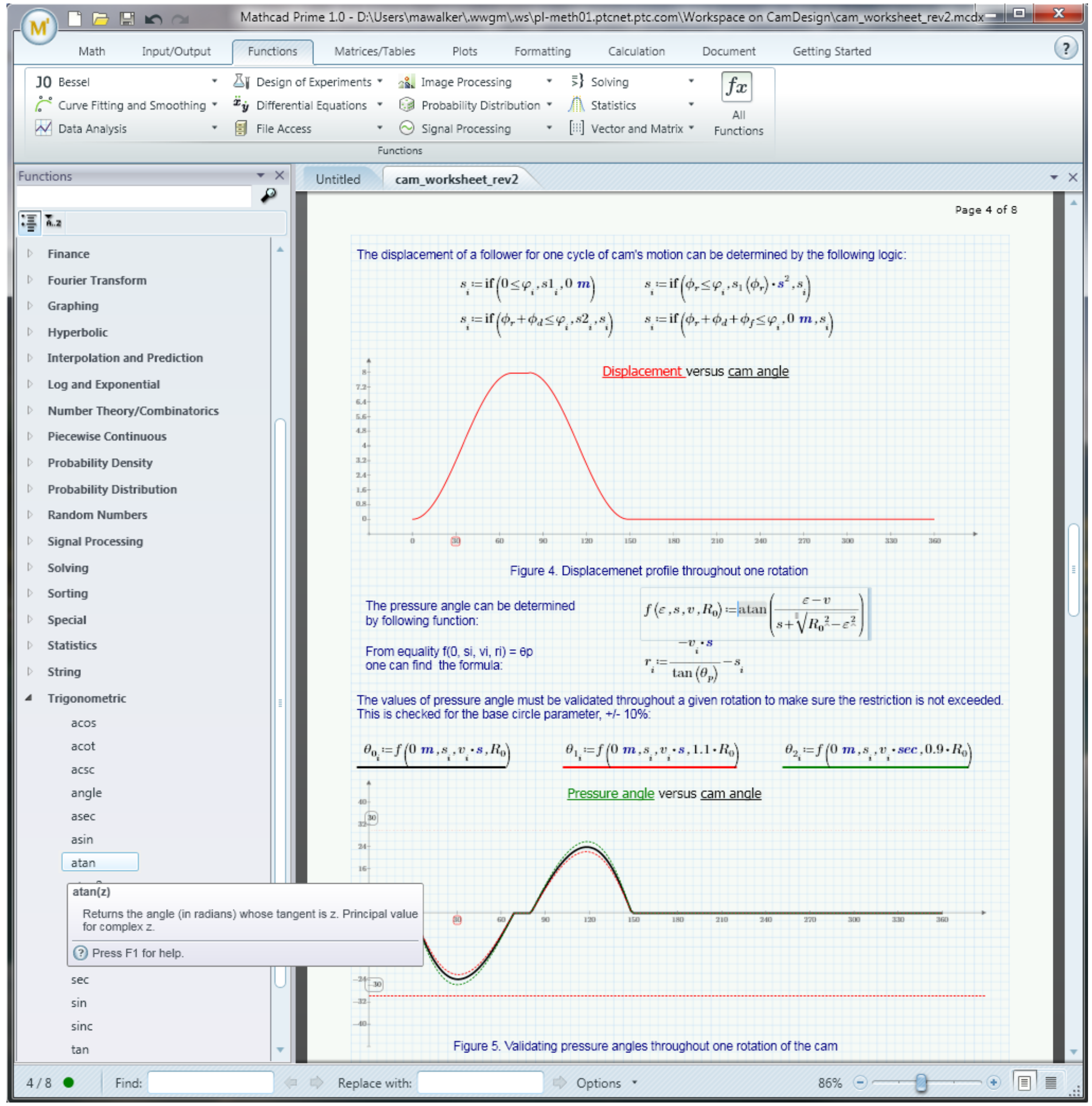


Maple - Screenshot

MathCad

Was the first to introduce live editing of typeset mathematical notation, combined with its automatic computations.

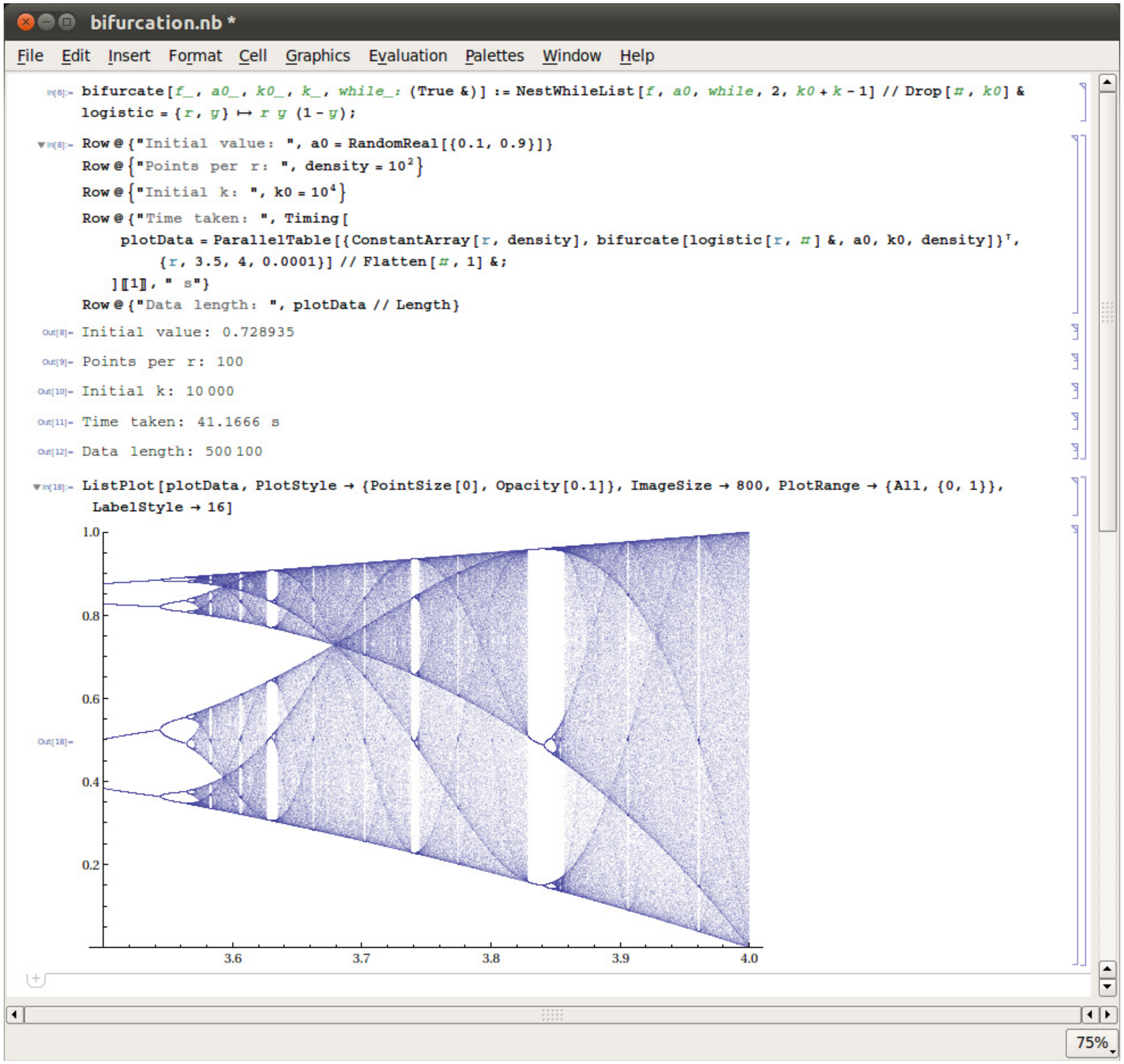
Mathcad includes some of the capabilities of a *computer algebra* system but is primarily oriented towards ease of use and *numerical engineering applications*.



Sheridan

Mathematica

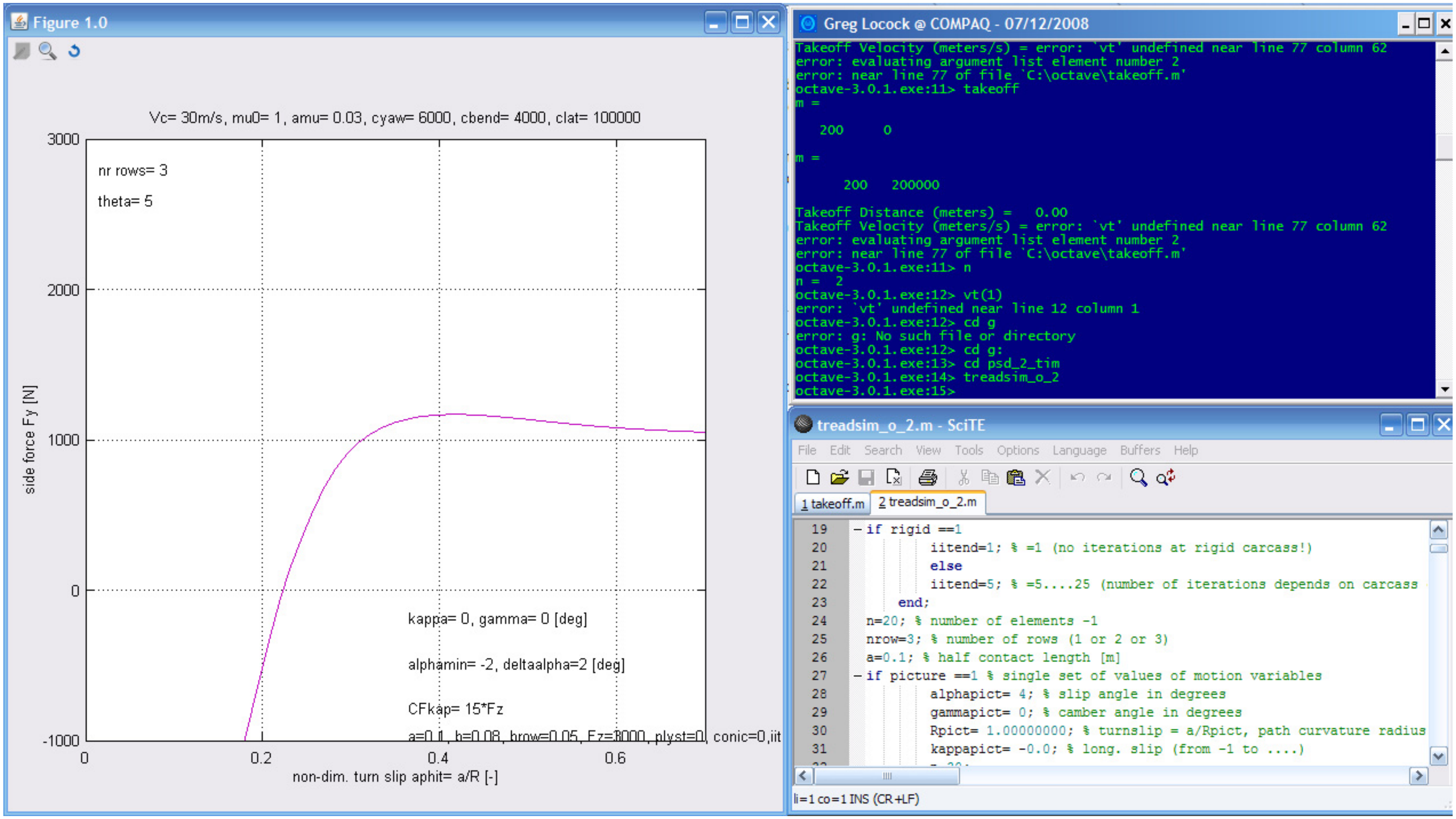
- Mathematica is a computational software program used in scientific, engineering, and mathematical fields and other areas of technical computing.
- It was conceived by Stephen Wolfram and is developed by Wolfram Research of Champaign, Illinois.
- List price: $2,495/copy; $1095/copy for professors; $139.95/copy for students; $1,995/copy for government employees.



Mathematica Screenshot

GNU Octave

- Origins: Started in January 2005 by William Stein by combining together the open source programs PARI, Maxima, Python, Singular and GAP.
- Mission statement: “Create a viable open source free alternative to Magma, Maple, Mathematica, and MATLAB which uses a standard modern language.”
- List price: $0
- Volunteers: About 50, with at least 20 regular contributors.
- Annual budget: Currently about $50K.
- Estimated number of users: Between 200 and 1000.



Octave - Screenshot

Sheridan

Sage

- **Sage** is an open source computer algebra system that supports teaching, study and research in mathematics.

- Its features cover many aspects of mathematics, including *algebra*, *combinatorics*, *numerical mathematics*, *number theory*, and *calculus*.

- It combines numerous open source packages and provides access to their functionalities via a common interface, namely, a *Python* based programming language supporting procedural, functional and object-oriented constructs..

Sheridan

- Sage can be used as a powerful desktop calculator, as a tool to help undergraduate students study mathematics, or as a programming environment for prototyping algorithms and research in algorithmic aspects of mathematics.

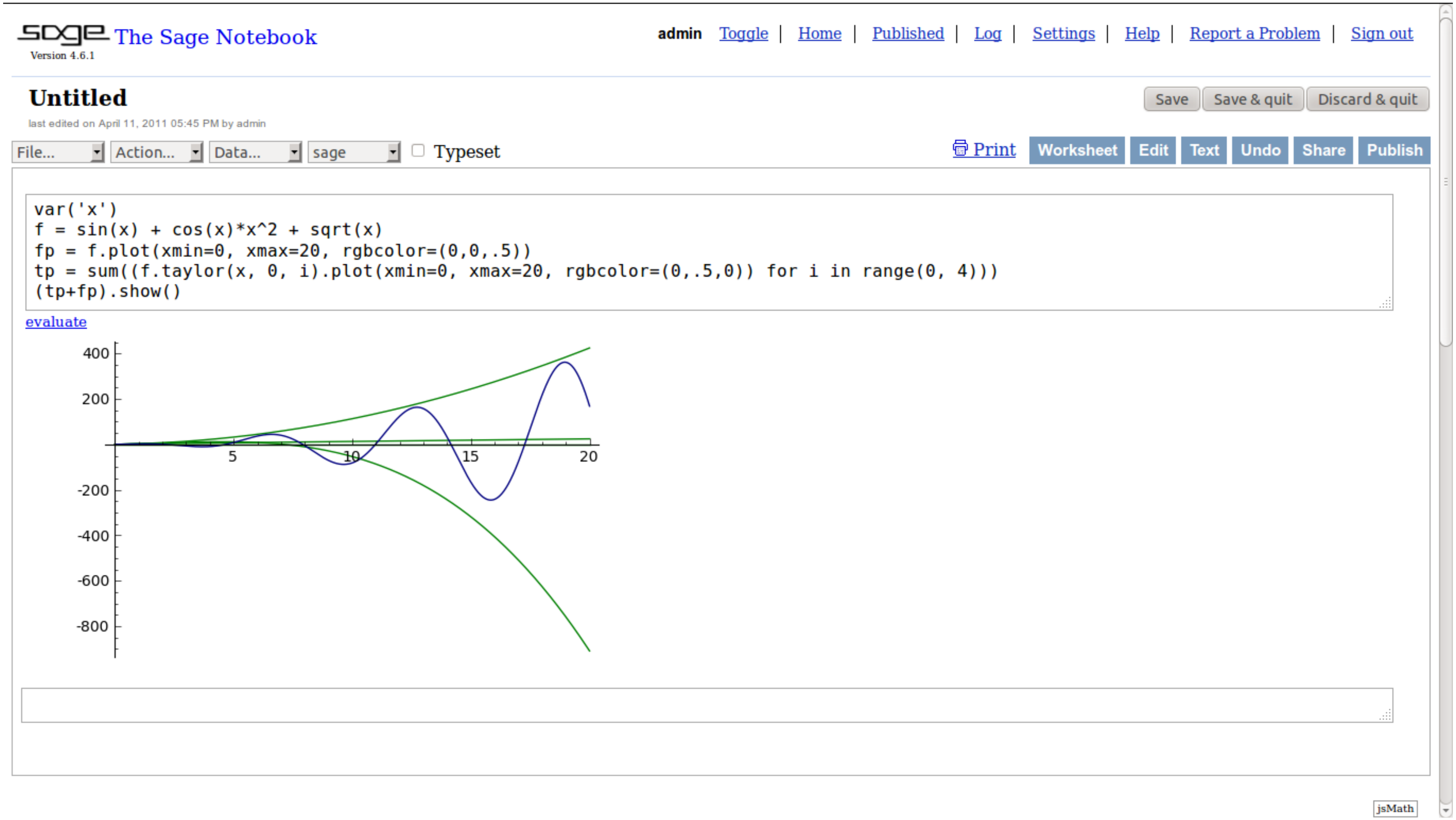
- Sage is available free of charge and can be downloaded from the following website:

  http://www.sagemath.org

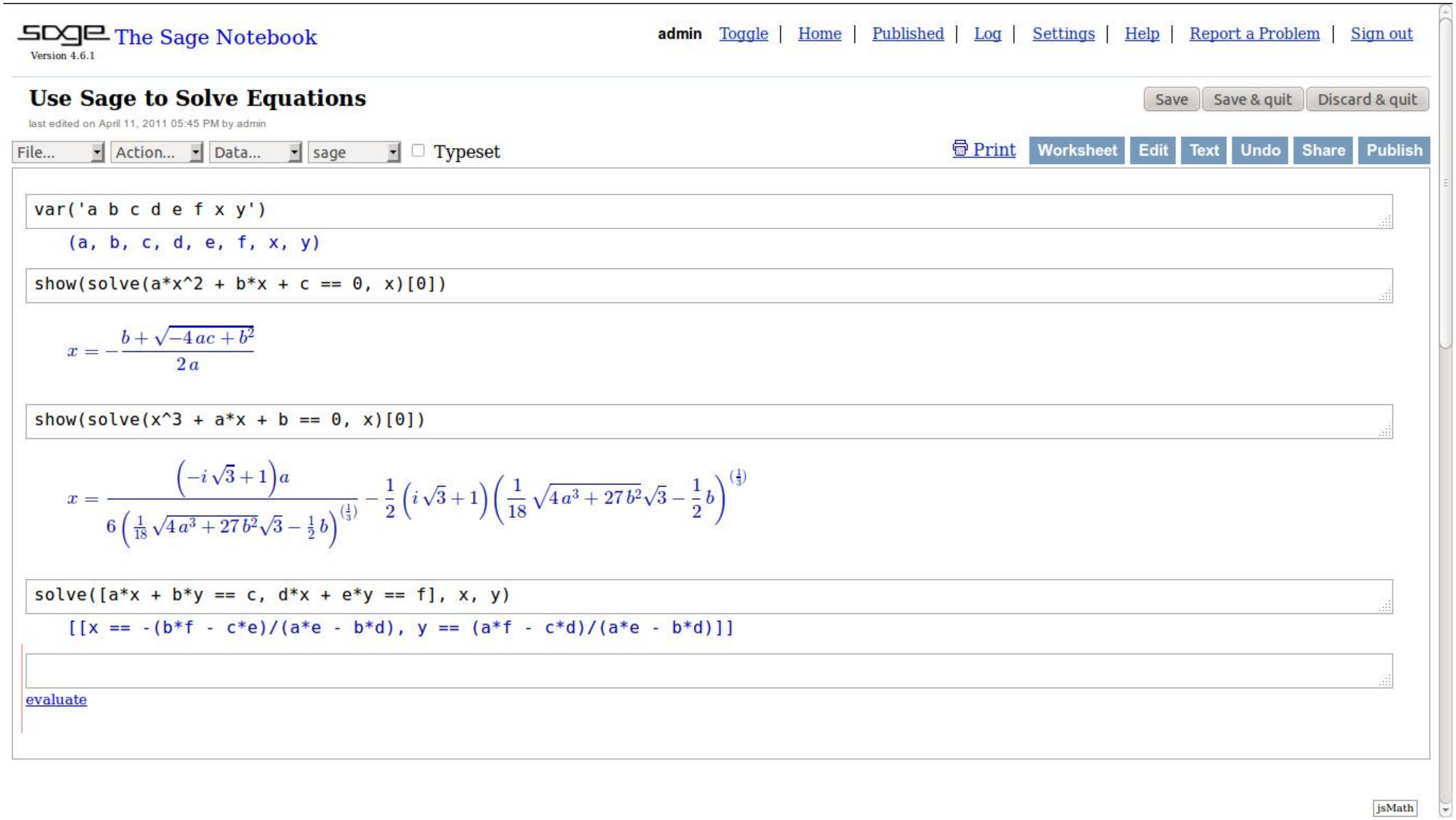
- The starter and leader of the Sage project, William Stein, is a mathematician at the University of Washington.

Sheridan

- Two modes of operation:
  - Downloadable install version for Linux or Oracle VM Virtualbox for MS Windows, or
  - Online command line or as Sage Notebook
- The default interface to Sage is command line based, but there is a graphical user interface to the software as well in the form of the Sage notebook.



Sage – Screenshot (graphics plotting)



Sage – Screenshot (typeset)

Sheridan

Sage Features

- Sage is built out of nearly 100 open-source packages and features a unified interface.

- Sage can be used to study elementary and advanced, pure and applied mathematics.

- This includes a huge range of mathematics, including basic algebra, calculus, elementary to very advanced number theory, cryptography, numerical computation, commutative algebra, group theory, combinatorics, graph theory, exact linear algebra and much more.

Using the Sage shell – Random Examples

```
sage: integral(x*sin(x^2), x)
-1/2*cos(x^2)

sage: integral(x/(x^2+1), x, 0, 1)
1/2*log(2)
```

```
sage: [sqrt(i) for i in srange(0,10,.1)]
```

```
sage: plot(sin,0,2)+plot(cos,0,2,rgbcolor='red')
```

Elementary Number Theory Examples

```
sage: primes_first_n(20)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37,
41, 43, 47, 53, 59, 61,
67, 71]
sage: gcd(18, 27)
9
sage: mod(23, 5)
3
sage: euler_phi(20)
8

sage: factorial(50)
30414093201713378043612608166064768844377641
568960512000000000000L
```


Sheridan

Generating pair of keys for RSA

1. Choose two primes p and q and let n = pq.
2. Let $e \in \mathbb{Z}, e > 0, \gcd(e, \varphi(n)) = 1$.
3. Compute a values for $d \in \mathbb{Z}, d > 0, de \equiv 1 \ (mod\ \varphi(n))$.
4. *Public key* is the pair (n, e) and *private key* is (p, q, d).
5. For any two non-zero integer m<n, encrypt m using $c \equiv m^e (mod\ n)$.
6. Decrypt c using $m \equiv c^d (mod\ n)$.

Sheridan

Use of Sage to generate RSA keys (example)

If p is prime and $M_p = 2^p - 1$ is also prime, then M_p is called a Mersenne prime. For example, for primes p = 31 and p = 61, M_p are Mersenne primes.

```
sage:  p = (2^31) - 1
sage:  is_prime(p)
True
sage:  q = (2^61) - 1
sage:  is_prime(q)
True
sage:  n = p * q ; n
4951760154835678088235319297
```


Sheridan

A word of warning: choice of p and q as Mersenne primes, and with so many digits far apart from each other, is a very bad choice in terms of cryptographic security.

```
sage: e = ZZ.random_element(euler_phi(n))
sage: while gcd(e, euler_phi(n)) != 1:
....: e = ZZ.random_element(euler_phi(n))
....:
sage: e
185056762330061596630395 4877
sage: e < n
True
sage: bezout = xgcd(e, euler_phi(n)) ; bezout
(1, 4460824882019967172592779313, -
1667095708515377925087033035)
sage: d=Integer(mod(bezout[1],euler_phi(n))) ; d
4460824882019967172592779313
sage: mod(d * e, euler_phi(n))
```


Sheridan

In this example RSA public key is:

(n, e) = (49517601548356780882353l9297,
1850567623300615966303954877)

and private key is:

(p, q, d) = (2147483647, 2305843009213693951,
4460824882019967172592779313)

Sheridan

Example of Sage programming

Find two numbers, both greater than 100,000 that have a greatest common divisor of exactly 3.

```
sage: while (p < 100000): p = random_prime(1000000)
....:
sage: p
586139
sage: q = 1
sage: while (q < 100000): q = random_prime(1000000)
....:
sage: q
938591
sage: A = 3*p
sage: B = 3*q
sage: xgcd(A,B)
(3, -380028, 237323)
```


Sheridan

How do we use Sage for evaluation?

1. In-class exercises

2. Homework and assignments

3. Group projects that involve use of Sage or similar tools

- Use of laptop during exams is not possible – we cannot isolate and control environment.
- Major concern may be – network communication among students in real-time.
- For the purpose of homework and projects, plagiarism among students is easy to recognize.
- Plagiarism accomplished by downloading solutions from the Web is virtually impossible to detect or prove.

Thank you!

Questions?

My contact information:

victor.ralevich@sheridanc.on.ca

Sheridan