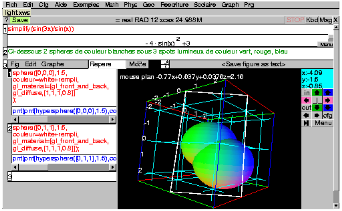
Introduction to Computer Algebra

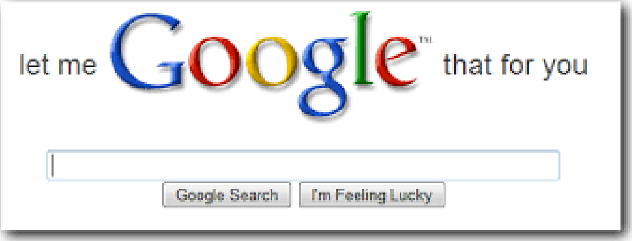
Carlos D'Andrea

Oslo, December 1st 2016



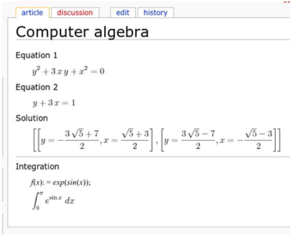
What is Computer Algebra?

What is Computer Algebra?



Computer Algebra

Computer Algebra



article | discussion | edit | history

Computer algebra

Equation 1

$y^2 + 3\,x\,y + x^2 = 0$

Equation 2

$y + 3\,x = 1$

Solution

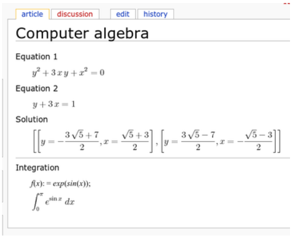
$\left[\left[y = -\frac{3\sqrt{5}+7}{2}, x = \frac{\sqrt{5}+3}{2}\right], \left[y = \frac{3\sqrt{5}-7}{2}, x = -\frac{\sqrt{5}-3}{2}\right]\right]$

Integration

$f(x) := exp(sin(x));$

$\int_0^{\pi} e^{\sin x}\, dx$

Computer Algebra



article | discussion | edit | history

Computer algebra

Equation 1

$$y^2+3\,x\,y+x^2=0$$

Equation 2

$$y+3\,x=1$$

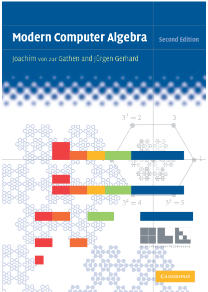
Solution

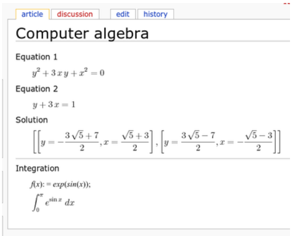
$$\left[\left[y=-\frac{3\sqrt{5}+7}{2}, x=\frac{\sqrt{5}+3}{2}\right], \left[y=\frac{3\sqrt{5}-7}{2}, x=-\frac{\sqrt{5}-3}{2}\right]\right]$$

Integration

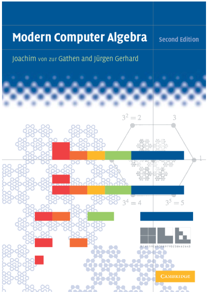
$f(x) := exp(sin(x));$

$$\int_0^{\pi} e^{\sin x}\,dx$$

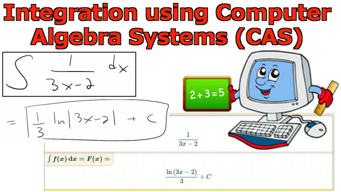




article
discussion
edit
history
Computer algebra
Equation 1
$y^2+3xy+x^2=0$
Equation 2
$y+3x=1$
Solution
$\left[\left[y=-\frac{3\sqrt{5}+7}{2}, x=\frac{\sqrt{5}+3}{2}\right], \left[y=\frac{3\sqrt{5}-7}{2}, x=-\frac{\sqrt{5}-3}{2}\right]\right]$
Integration
f(x): = exp(sin(x));
$\int_0^{\pi} e^{\sin x}\, dx$

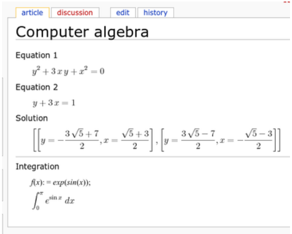


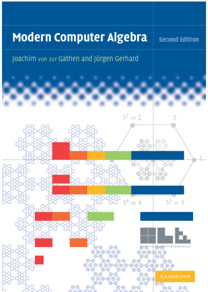
Modern Computer Algebra
Second Edition
Joachim von zur Gathen and Jürgen Gerhard
CAMBRIDGE

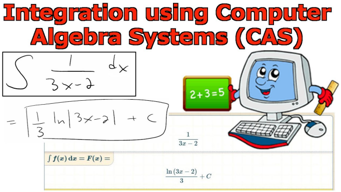


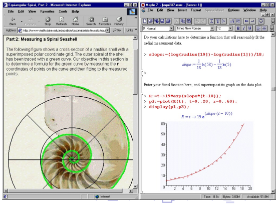
Integration using Computer Algebra Systems (CAS)
$\int \frac{1}{3x-2}\,dx$
$= \frac{1}{3}\ln|3x-2| + C$
2+3=5
$\frac{1}{3x-2}$
$\int f(x)\,dx = F(x) =$
$\frac{\ln(3x-2)}{3}+C$

Computer Algebra

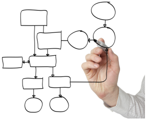




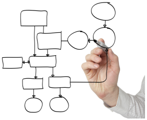




From Wikipedia (Symbolic Computation)

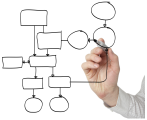


From Wikipedia (Symbolic Computation)



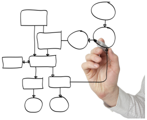
"...is a scientific area that refers to the study and development of

From Wikipedia (Symbolic Computation)



"...is a scientific area that refers to the study and development of **algorithms** and **software**

From Wikipedia (Symbolic Computation)



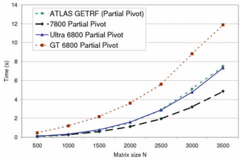
"...is a scientific area that refers to the study and development of **algorithms** and **software** for manipulating mathematical **expressions** and other mathematical **objects**..."

From Wikipedia (Symbolic Computation)

"A large part of the work in the field consists in revisiting classical algebra in order to

From Wikipedia (Symbolic Computation)

"A large part of the work in the field consists in revisiting classical algebra in order to make it **effective** and to discover **efficient** algorithms to implement this effectiveness"



“Effective” Operations

“Effective” Operations

- Boolean decisions: $=, \neq, (>, <)$

“Effective” Operations

- Boolean decisions: $=, \neq, (>, <)$
- Arithmetic Operations over “computable” rings: $\mathbb{Z}, \mathbb{Q}, \mathbb{F}_q, \mathbb{Q}[x_1, \ldots, x_n], \ldots$

"Effective" Operations

- Boolean decisions: $=, \neq, (>, <)$
- Arithmetic Operations over "computable" rings: $\mathbb{Z}, \mathbb{Q}, \mathbb{F}_q, \mathbb{Q}[x_1, \ldots, x_n], \ldots$
- Finite-Dimensional Linear Algebra

What is “an algorithm” for CA?

What is “an algorithm” for CA?

Given $A, B \in \mathbb{Z}$,

What is "an algorithm" for CA?

Given $A,\ B \in \mathbb{Z}$,compute the product

$$A \cdot B \in \mathbb{Z}$$

What is “an algorithm” for CA?

Given A, $B \in \mathbb{Z}$, compute the product

$$A \cdot B \in \mathbb{Z}$$

$$\begin{array}{r}
\overset{1}{5127} \\
\text{x } 4265 \\
\hline
25635 \\
307620 \\
1025400
\end{array}$$

$$\begin{array}{r}
453 \\
\times \; 1001205 \\
\hline
2265 \\
906\cdot \\
453 \\
453\cdot\cdot \\
\hline
453545865
\end{array}$$

Example: The Multiplication Algorithm

- Input: $A, B \in \mathbb{Z}$

Example: The Multiplication Algorithm

- Input: $A, B \in \mathbb{Z}$
- Output: $A \cdot B \in \mathbb{Z}$

Example: The Multiplication Algorithm

- Input: $A, B \in \mathbb{Z}$
- Output: $A \cdot B \in \mathbb{Z}$
- Procedure:

```
   36
 x 24
 ----
  144     (4 x 36 = 144)
+ 720     (2 x 36 = 72,
 ----     with a 0 added in
  864     the ones' position)
```

Effective and Efficient

Effective:

Effective and Efficient

Effective: the procedure must finish after a finite number of operations,

Effective and Efficient

Effective: the procedure must finish after a finite number of operations, and give the right answer

Effective and Efficient

Effective: the procedure must finish after a finite number of operations, and give the right answer

Efficient:

Effective and Efficient

Effective: the procedure must finish after a finite number of operations, and give the right answer

Efficient: the procedure must be as **short** as possible

Effective and Efficient

Effective: the procedure must finish after a finite number of operations, and give the right answer

Efficient: the procedure must be as **short** as possible and use as **less space** as possible

Short and less Space

- Input: $A, B \in \mathbb{Z}$

Short and less Space

- Input: $A, B \in \mathbb{Z}$ of N digits

Short and less Space

- Input: $A, B \in \mathbb{Z}$ of N digits
- Output: $A \cdot B \in \mathbb{Z}$

Short and less Space

- Input: $A, B \in \mathbb{Z}$ of N digits
- Output: $A \cdot B \in \mathbb{Z}$

  Output's size is around $2N$ digits

Short and less Space

- Input: $A, B \in \mathbb{Z}$ of N digits
- Output: $A \cdot B \in \mathbb{Z}$

Output's size is around $2N$ digits

High school algorithm takes around $\mathcal{O}(N^2)$ operations

Short and less Space

- Input: $A, B \in \mathbb{Z}$ of N digits
- Output: $A \cdot B \in \mathbb{Z}$

Output's size is around $2N$ digits

High school algorithm takes around $\mathcal{O}(N^2)$ operations

It is effective

Short and less Space

- Input: $A, B \in \mathbb{Z}$ of N digits
- Output: $A \cdot B \in \mathbb{Z}$

Output's size is around $2N$ digits

High school algorithm takes around $\mathcal{O}(N^2)$ operations

It is effective, but is it efficient?

Efficient Multiplication

Karatsuba's algorithm (1960): $\mathcal{O}(N^{1.585})$

Multiplication of large numbers					
X	9876256719 8712349865				
	0054739234 5643218574				Y
Decomposition					
X	L	9876256719 * 0054739234		L	Y
	R	+ 8712349865 * 5643218574 +		R	
	S	= 18588606584 * 5697957808 =		S	
Partial multiplications					
P	1	540618727585413246	=	X_L * Y_L	
	2	49165694581354392510	=	X_R * Y_R	
	3	105917096025143007872	=	X_S * Y_S	
Result					
P_1 * 10 ^ N + (P_3 - P_2 - P_1) * 10 ^ (N/2) + P_2					
N = Max(Length_X, Length_Y) and if N odd then N = N + 1					
54 061 873 320 649 151 811 197 715 741 354 392 510					

Toom-Cook's algorithm (1966):

$\mathcal{O}(N^{\log(2k-1)/\log k}),\ k \geq 3$

Toom-Cook's algorithm (1966): $\mathcal{O}(N^{\log(2k-1)/\log k}),\ k \geq 3$

Schönhage-Strassen's algorithm (1971): $\mathcal{O}(N \log N \log\log N)$

$$W = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & & \omega^{2n-1} \\ 1 & \omega^2 & \omega^4 & & \omega^{4n-2} \\ & & & & \\ 1 & \omega^{2n-1} & \omega^{4n-2} & & \omega^{2n^2-n} \end{pmatrix}$$

Efficient Multiplication

Fürer's algorithm (2007):
$\mathcal{O}(N \log N \, 2^{\mathcal{O}(\log^* N)})$

$$W = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & & \omega^{2n-1} \\ 1 & \omega^2 & \omega^4 & & \omega^{4n-2} \\ & & & & \\ 1 & \omega^{2n-1} & \omega^{4n-2} & & \omega^{2n^2-n} \end{pmatrix}$$

Efficient Multiplication

Fürer's algorithm (2007):

$$\mathcal{O}(N \log N \, 2^{\mathcal{O}(\log^* N)})$$

$$W = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & & \omega^{2n-1} \\ 1 & \omega^2 & \omega^4 & & \omega^{4n-2} \\ & & & & \\ 1 & \omega^{2n-1} & \omega^{4n-2} & & \omega^{2n^2-n} \end{pmatrix}$$

Is it the most efficient??

Numerical vs Computer Algebra

Numerical vs Computer Algebra

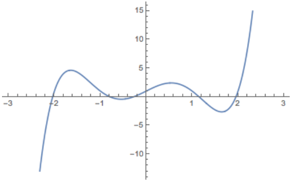
"As **numerical** software are highly efficient for approximate numerical computation, it is common, in **computer algebra**

Numerical vs Computer Algebra

"As **numerical** software are highly efficient for approximate numerical computation, it is common, in **computer algebra**, to emphasize on **exact computation** with **exactly represented data**"

Compute the roots of

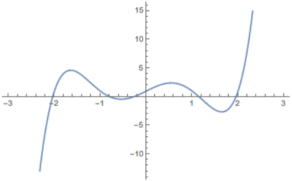
$$x^5 - 5x^3 + 4x + 1$$



Numerical vs Computer Algebra

Compute the roots of

$$x^5 - 5x^3 + 4x + 1$$

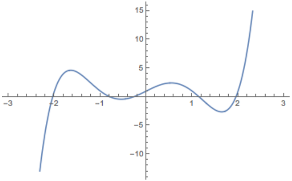


Numerical "solution":

Numerical vs Computer Algebra

Compute the roots of

$$x^5 - 5x^3 + 4x + 1$$

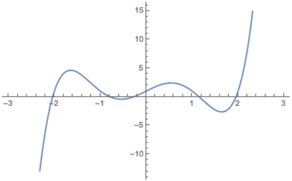


Numerical "solution":

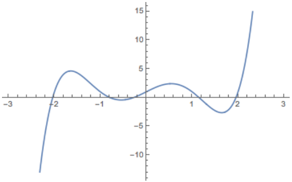
$-2.0385; -0.790734; -0.275834; 1.15098; 1.95408$

Exactly Represented Data

Exactly Represented Data

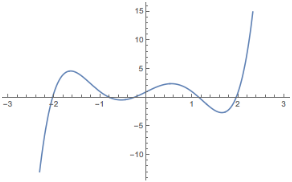


Exactly Represented Data



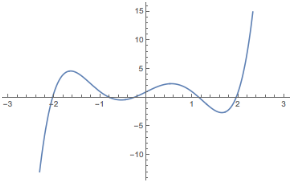
- $x^5 - 5x^3 + 4x + 1$

Exactly Represented Data



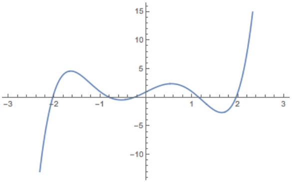
- $x^5 - 5x^3 + 4x + 1$
- $[-2.5, -2]; [-2, -0.5]; [-0.5, 0]; [0, 1.5]; [1.5, 2]$

Exactly Represented Data



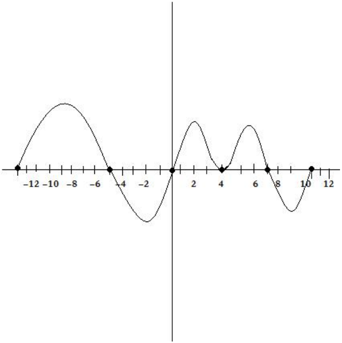
- $x^5 - 5x^3 + 4x + 1$
- $[-2.5, -2]; [-2, -0.5]; [-0.5, 0]; [0, 1.5]; [1.5, 2]$
- $\{+, -, +, -, +\}$, $\{-, +, +, -, +\}$, $\{+, +, -, -, +\}$, $\{-, -, +, +, +\}$, $\{+, +, +, +, +\}$

Exactly Represented Data

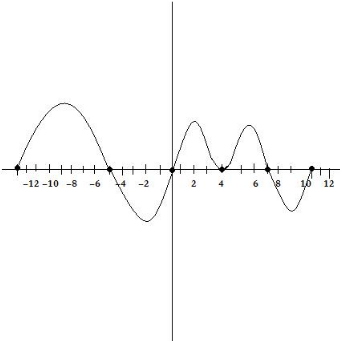


- $x^5 - 5x^3 + 4x + 1$
- $[-2.5, -2]; [-2, -0.5]; [-0.5, 0]; [0, 1.5]; [1.5, 2]$
- $\{+, -, +, -, +\}$, $\{-, +, +, -, +\}$, $\{+, +, -, -, +\}$, $\{-, -, +, +, +\}$, $\{+, +, +, +, +\}$
- $-2.5;\ -0.75;\ -0.25;\ 1;\ 2$

Computational Challenge

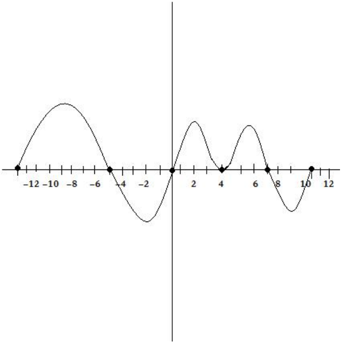


Computational Challenge



Given $f(x) \in \mathbb{Z}[x]$, compute a set of **small** approximate roots of it

Computational Challenge



Given $f(x) \in \mathbb{Z}[x]$, compute a set of **small** approximate roots of it **fast**

What is a small number?

What is a small number?

The size of an integer N is its number of digits:

What is a small number?

The size of an integer N is its number of digits: $\log N$

What is a small number?

The size of an integer N is its number of digits: $\log N$

The size of a fraction $\frac{N_1}{N_2}$ will be

What is a small number?

The size of an integer N is its number of digits: $\log N$

The size of a fraction $\frac{N_1}{N_2}$ will be $\max\{\log N_1, \log N_2\}$

What is a small number?

The size of an integer N is its number of digits: $\log N$

The size of a fraction $\frac{N_1}{N_2}$ will be $\max\{\log N_1, \log N_2\}$

Size of Algebraic Numbers

What is “the size” of $\sqrt{2}$?

Size of Algebraic Numbers

What is "the size" of $\sqrt{2}$?

Given $\alpha \in \mathbb{C}$ the root of

$$(*) \quad a_0 + a_1 x + \ldots + a_n x^n \in \mathbb{Z}[x]$$

Size of Algebraic Numbers

What is "the size" of $\sqrt{2}$?

Given $\alpha \in \mathbb{C}$ the root of

$$(*) \quad a_0 + a_1 x + \ldots + a_n x^n \in \mathbb{Z}[x]$$

The "size" of α is

$$\left(n, \ \max\{\log |a_i|_{1 \leq i \leq n}\}\right)$$

Size of Algebraic Numbers

What is "the size" of $\sqrt{2}$?

Given $\alpha \in \mathbb{C}$ the root of

$$(*) \quad a_0 + a_1 x + \ldots + a_n x^n \in \mathbb{Z}[x]$$

The "size" of α is

$$\left(n, \max\{\log |a_i|_{1 \leq i \leq n}\}\right)$$

if $(*)$ is the minimal polynomial of α

With this definition...

With this definition...

- size of 2016 :

With this definition...

- size of $2016: \ (1, \log 2016 \approx 11)$

With this definition...

- size of 2016 : $(1, \log 2016 \approx 11)$
- size of $\sqrt{2}$:

With this definition...

- size of 2016 : $(1, \log 2016 \approx 11)$
- size of $\sqrt{2}$: $(2, \log 2 = 1)$

With this definition...

- size of $2016: \ (1, \log 2016 \approx 11)$
- size of $\sqrt{2}: (2, \log 2 = 1)$
- size of $-\frac{1}{2} + \frac{\sqrt{3}}{2} i:$

With this definition...

- size of $2016 : \ (1, \log 2016 \approx 11)$
- size of $\sqrt{2} : (2, \log 2 = 1)$
- size of $-\frac{1}{2} + \frac{\sqrt{3}}{2} i : (2, \log 1 = 0)$

With this definition...

- size of 2016 : $(1, \log 2016 \approx 11)$
- size of $\sqrt{2}$: $(2, \log 2 = 1)$
- size of $-\frac{1}{2} + \frac{\sqrt{3}}{2} i$: $(2, \log 1 = 0)$
- size of $\sqrt[3]{2}$:

With this definition...

- size of 2016 : $(1, \log 2016 \approx 11)$
- size of $\sqrt{2}$: $(2, \log 2 = 1)$
- size of $-\frac{1}{2} + \frac{\sqrt{3}}{2} i$: $(2, \log 1 = 0)$
- size of $\sqrt[3]{2}$: $(3, \log 2 = 1)$

How does the size grow?

Over the integers: size of
N

How does the size grow?

Over the integers: size of $N = (1, \log N)$

How does the size grow?

Over the integers: size of

$$N = (1, \log N)$$

Size of

- $N_1 + N_2$:

How does the size grow?

Over the integers: size of

$$N = (1, \log N)$$

Size of

- $N_1 + N_2 : \quad (1, \max\{\log N_1, \log N_2\})$

How does the size grow?

Over the integers: size of

$$N = (1, \log N)$$

Size of

- $N_1 + N_2 : \quad (1, \max\{\log N_1, \log N_2\})$
- $N_1 \cdot N_2 :$

How does the size grow?

Over the integers: size of

$$N = (1, \log N)$$

Size of

- $N_1 + N_2: \quad (1, \max\{\log N_1, \log N_2\})$
- $N_1 \cdot N_2: \quad (1, \log N_1 + \log N_2)$

Size of algebraic numbers

Size of:

- $\sqrt{2} \cdot \sqrt[3]{2}$:

Size of algebraic numbers

Size of:

- $\sqrt{2} \cdot \sqrt[3]{2} : (6, \log 32 = 5)$

Size of algebraic numbers

Size of:

- $\sqrt{2} \cdot \sqrt[3]{2} : (6, \log 32 = 5)$
  $(2^{\frac{1}{2}+\frac{1}{3}} = 2^{\frac{5}{6}}$

Size of algebraic numbers

Size of:

- $\sqrt{2}\cdot\sqrt[3]{2}$: $(6, \log 32 = 5)$ $(2^{\frac{1}{2}+\frac{1}{3}} = 2^{\frac{5}{6}} \leftrightarrow x^6 - 2^5 = 0)$
- $\sqrt{2}+\sqrt[3]{2}$:???

Computer Algebra helps!

$$\begin{cases} x^2 - 2 = 0 \\ (y - x)^3 - 2 = 0 \end{cases}$$

Computer Algebra helps!

$$\begin{cases} x^2 - 2 = 0 \\ (y-x)^3 - 2 = 0 \end{cases}$$

$$\downarrow$$

$$\begin{cases} x^2 - 2 = 0 \\ \mathbf{y^6 - 6y^4 - 4y^3 + 12y^2 - 24y = 4} \end{cases}$$

Computer Algebra helps!

$$\begin{cases} x^2 - 2 = 0 \\ (y-x)^3 - 2 = 0 \end{cases}$$

$$\downarrow$$

$$\begin{cases} x^2 - 2 = 0 \\ \mathbf{y^6 - 6y^4 - 4y^3 + 12y^2 - 24y = 4} \end{cases}$$

$$\boxed{\text{size of } \sqrt{2} + \sqrt[3]{2} = (6, \log 24 \approx 4.5)}$$

In general...

In general...

if the sizes of α_1, α_2 are (d_1, L_1), (d_2, L_2),

In general...

if the sizes of α_1, α_2 are (d_1, L_1), (d_2, L_2), the sizes of both $\alpha_1 + \alpha_2$ and $\alpha_1 \cdot \alpha_2$ is of the order of

In general...

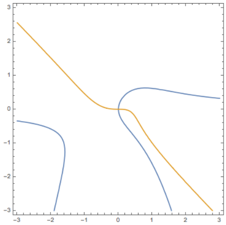
if the sizes of α_1, α_2 are (d_1, L_1), (d_2, L_2), the sizes of both $\alpha_1 + \alpha_2$ and $\alpha_1 \cdot \alpha_2$ is of the order of $(d_1 \cdot d_2, d_1 L_2 + d_2 L_1)$

Triangulating systems of equations

$$\begin{cases} x^2y + y^2 - x = 0 \\ x^3 + y^3 - xy + y = 0 \end{cases}$$

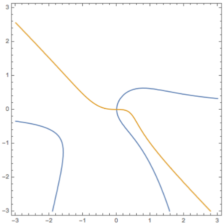
Triangulating systems of equations

$$\begin{cases} x^2y + y^2 - x = 0 \\ x^3 + y^3 - xy + y = 0 \end{cases}$$



Triangulating systems of equations

$$\begin{cases} x^2y + y^2 - x = 0 \\ x^3 + y^3 - xy + y = 0 \end{cases}$$



$$\begin{cases} x^3 + y^3 - xy + y = 0 \\ \mathbf{y^9 + 2y^7 + y^5 - 4y^4 + y = 0} \end{cases}$$

What can you get from here?

What can you get from here?

- Number of solutions of a system of equations

What can you get from here?

- Number of solutions of a system of equations
- "Location" of the solutions

What can you get from here?

- Number of solutions of a system of equations
- "Location" of the solutions
- Dimension, degree, size, ...

What can you get from here?

- Number of solutions of a system of equations
- "Location" of the solutions
- Dimension, degree, size, ...
- Symbolic Integration

What can you get from here?

- Number of solutions of a system of equations
- “Location” of the solutions
- Dimension, degree, size, ...
- Symbolic Integration
- Factorization of polynomials,

What can you get from here?

- Number of solutions of a system of equations
- "Location" of the solutions
- Dimension, degree, size, ...
- Symbolic Integration
- Factorization of polynomials, matrices,

What can you get from here?

- Number of solutions of a system of equations
- "Location" of the solutions
- Dimension, degree, size, ...
- Symbolic Integration
- Factorization of polynomials, matrices, differential operators,...

“Triangulation” $=$ Elimination of variables

"Triangulation" = Elimination of variables

Find "the condition" on

$a_{10}, a_{11}, a_{20}, a_{21}$ so that the system

$$\begin{cases} a_{10}x_0 + a_{11}x_1 = 0 \\ a_{20}x_0 + a_{21}x_1 = 0 \end{cases}$$

has a solution different from $(0,0)$

The General System with Parameters

The General System with Parameters

For $\mathbf{a} = (a_1, \ldots, a_N)$, k, $n \in \mathbb{N}$ let $f_1(\mathbf{a}, x_1, \ldots, x_n), \ldots, f_k(\mathbf{a}, x_1, \ldots, x_n) \in \mathbb{K}[\mathbf{a}, x_1, \ldots, x_n]$. Find conditions on $\mathbf{a}$ such that

$$\left\{\begin{array}{rcl} f_1(\mathbf{a}, x_1, \ldots, x_n) & = & 0 \\ f_2(\mathbf{a}, x_1, \ldots, x_n) & = & 0 \\ \vdots & \vdots & \vdots \\ f_k(\mathbf{a}, x_1, \ldots, x_n) & = & 0 \end{array}\right.$$

has a solution

Solution?

- Depends on the ground field

Solution?

- Depends on the ground field
- There is not necessarily a “closed” condition

Solution?

- Depends on the ground field
- There is not necessarily a "closed" condition
- Tools from Geometry are needed!

The "simplest" example

$$k = n = 1,$$

$$a_0 + a_1 x_1 + a_2 {x_1}^2 + \ldots + a_d {x_1}^d = 0$$

The "simplest" example

$$k = n = 1,$$

$$a_0 + a_1 x_1 + a_2 {x_1}^2 + \ldots + a_d {x_1}^d = 0$$

Conditions?

Known and "universal" examples

Known and "universal" examples

$$\left\{\begin{array}{rcl} a_{11}x_1 + \ldots + a_{1n}x_n & = & 0 \\ a_{21}x_1 + \ldots + a_{2n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ a_{k1}x_1 + \ldots + a_{kn}x_n & = & 0 \end{array}\right.$$

with $k \geq n$

Known and “universal” examples

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n & = & 0 \\ a_{21}x_1 + \ldots + a_{2n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ a_{k1}x_1 + \ldots + a_{kn}x_n & = & 0 \end{cases}$$

with $k \geq n$

Conditions: all maximal minors of $\left(a_{ij}\right)_{1\leq i\leq k,\, 1\leq j\leq n}$ equal to zero

Another Classical Example

Another Classical Example

$$\begin{cases} a_{11}v_1 + \ldots + a_{1n}v_n &= \lambda v_1 \\ a_{21}v_1 + \ldots + a_{2n}v_n &= \lambda v_2 \\ \quad\quad\quad \vdots \quad\quad\quad\quad \vdots & \quad \vdots \\ a_{n1}v_1 + \ldots + a_{nn}v_n &= \lambda v_n \end{cases}$$

Another Classical Example

$$\begin{cases} a_{11}v_1 + \ldots + a_{1n}v_n & = & \lambda v_1 \\ a_{21}v_1 + \ldots + a_{2n}v_n & = & \lambda v_2 \\ \\ \vdots \qquad\qquad \vdots & & \vdots \\ a_{n1}v_1 + \ldots + a_{nn}v_n & = & \lambda v_n \end{cases}$$

<u>Condition:</u> $C_A(\lambda) = 0$

Geometry

$$V = \{(\boldsymbol{a}, x_1, \ldots, x_n) : f_1(\boldsymbol{a}, x_1, \ldots, x_n) = 0, \ldots f_k(\boldsymbol{a}, x_1, \ldots, x_n) = 0\}$$

Geometry

$$V = \{(a, x_1, \ldots, x_n) : f_1(a, x_1, \ldots, x_n) = 0, \ldots f_k(a, x_1, \ldots, x_n) = 0\}$$

$$\begin{array}{ccc} V & \subset & \mathbb{K}^N \times \mathbb{K}^n \\ & & \\ \downarrow \pi_1 & & \downarrow \pi_1 \\ & & \\ \pi_1(V) & \subset & \mathbb{K}^N \end{array}$$

Geometry

$$V = \{(\mathbf{a}, x_1, \ldots, x_n) : f_1(\mathbf{a}, x_1, \ldots, x_n) = 0, \ldots f_k(\mathbf{a}, x_1, \ldots, x_n) = 0\}$$

$$\begin{array}{ccc} V & \subset & \mathbb{K}^N \times \mathbb{K}^n \\ \downarrow \pi_1 & & \downarrow \pi_1 \\ \pi_1(V) & \subset & \mathbb{K}^N \end{array}$$

The set of conditions is $\pi_1(\mathbf{V})$, not necessarily described by zeroes of polynomials

Elimination Theorem

$$V = \{(\boldsymbol{a}, x_0, x_1, \ldots, x_n) : f_1(\boldsymbol{a}, x_0, x_1, \ldots, x_n) = 0, \ldots f_k(\boldsymbol{a}, x_0, x_1, \ldots, x_n) = 0\}$$

Elimination Theorem

$$V = \{(\mathbf{a}, x_0, x_1, \ldots, x_n) : f_1(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0, \ldots f_k(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0\}$$

$$\begin{array}{ccc} V & \subset & \mathbb{K}^N \times \mathbb{P}^n \\ \downarrow \pi_1 & & \downarrow \pi_1 \\ \pi_1(V) & \subset & \mathbb{K}^N \end{array}$$

Elimination Theorem

$$V = \{(\mathbf{a}, x_0, x_1, \ldots, x_n) : \, f_1(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0, \ldots f_k(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0\}$$

$$\begin{array}{ccc} V & \subset & \mathbb{K}^N \times \mathbb{P}^n \\ \downarrow \pi_1 & & \downarrow \pi_1 \\ \pi_1(V) & \subset & \mathbb{K}^N \end{array}$$

$$\pi_1(V) = \{p_1(\mathbf{a}) = 0, \ldots, p_\ell(\mathbf{a}) = 0\}$$

"One" Condition

$$V = \{(\boldsymbol{a}, x_0, x_1, \ldots, x_n) : f_1(\boldsymbol{a}, x_0, x_1, \ldots, x_n) = 0, \ldots, f_{n+1}(\boldsymbol{a}, x_0, x_1, \ldots, x_n) = 0\}$$

"One" Condition

$$V = \{(\mathbf{a}, x_0, x_1, \ldots, x_n) : f_1(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0, \ldots, f_{n+1}(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0\}$$

$$\begin{array}{ccc} V & \subset & \mathbb{K}^N \times \mathbb{P}^n \\ \downarrow \pi_1 & & \downarrow \pi_1 \\ \pi_1(V) & \subset & \mathbb{K}^N \end{array}$$

"One" Condition

$$V = \{(\mathbf{a}, x_0, x_1, \ldots, x_n) : f_1(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0, \ldots, f_{n+1}(\mathbf{a}, x_0, x_1, \ldots, x_n) = 0\}$$

$$\begin{array}{ccc} V & \subset & \mathbb{K}^N \times \mathbb{P}^n \\ \downarrow \pi_1 & & \downarrow \pi_1 \\ \pi_1(V) & \subset & \mathbb{K}^N \end{array}$$

$$\boxed{\pi_1(V) = \{p_1(\mathbf{a}) = 0\}}$$

Example 1

$$\left\{\begin{array}{rcl} a_{00}x_0 + a_{01}x_1 + \ldots + a_{0n}x_n & = & 0 \\ a_{10}x_0 + a_{11}x_1 + \ldots + a_{1n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ a_{n0}x_0 + a_{n1}x_1 + \ldots + a_{nn}x_n & = & 0 \end{array}\right.$$

Example 1

$$\begin{cases} a_{00}x_0 + a_{01}x_1 + \ldots + a_{0n}x_n & = & 0 \\ a_{10}x_0 + a_{11}x_1 + \ldots + a_{1n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ a_{n0}x_0 + a_{n1}x_1 + \ldots + a_{nn}x_n & = & 0 \end{cases}$$

$$\boxed{p_1(a) = \det(a_{ij})}$$

Example 2

$$\begin{cases} f_1 = a_{10}{x_0}^{d_1} + a_{11}{x_0}^{d_1-1}x_1 + \ldots + a_{1d_1}{x_1}^{d_1} \\ f_2 = a_{20}{x_0}^{d_2} + a_{21}{x_0}^{d_2-1}x_1 + \ldots + a_{2d_2}{x_1}^{d_2} \end{cases}$$

Example 2

$$\begin{cases} f_1 = a_{10}{x_0}^{d_1} + a_{11}{x_0}^{d_1-1}x_1 + \ldots + a_{1d_1}{x_1}^{d_1} \\ f_2 = a_{20}{x_0}^{d_2} + a_{21}{x_0}^{d_2-1}x_1 + \ldots + a_{2d_2}{x_1}^{d_2} \end{cases}$$

$$\boxed{p_1(a) = \det \begin{pmatrix} a_{10} & a_{11} & \ldots & a_{1d_1} & 0 & \ldots & 0 \\ 0 & a_{10} & \ldots & a_{1d_1-1} & a_{1d_1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \ldots & \ldots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{10} & \ldots & \ldots & a_{1d_1} \\ a_{20} & a_{21} & \ldots & a_{2d_2} & 0 & \ldots & 0 \\ 0 & a_{20} & \ldots & a_{2d_2-1} & a_{2d_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \ldots & \ldots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{20} & \ldots & \ldots & a_{2d_2} \end{pmatrix}}$$

Example 3

$$\begin{cases} f_1 = \sum_{\alpha_0+\ldots+\alpha_n=d_1} a_{1,\alpha_0,\ldots,\alpha_n} x_0{}^{\alpha_0} \ldots x_n{}^{\alpha_n} \\ f_2 = \sum_{\alpha_0+\ldots+\alpha_n=d_2} a_{2,\alpha_0,\ldots,\alpha_n} x_0{}^{\alpha_0} \ldots x_n{}^{\alpha_n} \\ \vdots \\ f_{n+1} = \sum_{\alpha_0+\ldots+\alpha_n=d_{n+1}} a_{n+1,\alpha_0,\ldots,\alpha_n} x_0{}^{\alpha_0} \ldots x_n{}^{\alpha_n} \end{cases}$$

Example 3

$$\begin{cases} f_1 = \sum_{\alpha_0+\ldots+\alpha_n=d_1} a_{1,\alpha_0,\ldots,\alpha_n} x_0{}^{\alpha_0} \ldots x_n{}^{\alpha_n} \\ f_2 = \sum_{\alpha_0+\ldots+\alpha_n=d_2} a_{2,\alpha_0,\ldots,\alpha_n} x_0{}^{\alpha_0} \ldots x_n{}^{\alpha_n} \\ \vdots \\ f_{n+1} = \sum_{\alpha_0+\ldots+\alpha_n=d_{n+1}} a_{n+1,\alpha_0,\ldots,\alpha_n} x_0{}^{\alpha_0} \ldots x_n{}^{\alpha_n} \end{cases}$$

$$\boxed{\mathrm{Res}(f_1, f_2, \ldots, f_{n+1})}$$

Effective tools

Linear **Polynomial**

Effective tools

Linear

Determinants

Polynomial

Resultants

Effective tools

Linear

Determinants

Cramer's rule

Polynomial

Resultants

u-resultants

Effective tools

Linear	**Polynomial**
Determinants	Resultants
Cramer's rule	u-resultants
Gauss elimination	Gröbner Bases

Effective tools

Linear	**Polynomial**
Determinants	Resultants
Cramer's rule	u-resultants
Gauss elimination	Gröbner Bases
Triangulation	Triangular systems

Effective tools

Linear	**Polynomial**
Determinants	Resultants
Cramer's rule	u-resultants
Gauss elimination	Gröbner Bases
Triangulation	Triangular systems
$\vdots$	$\vdots$

How "efficient" is all this?

How "efficient" is all this?

The size of the solutions of $\left\{\begin{array}{lcl} f_1(x_1,\ldots,x_n) & = & 0 \\ f_2(x_1,\ldots,x_n) & = & 0 \\ \vdots & \vdots & \vdots \\ f_n(x_1,\ldots,x_n) & = & 0 \end{array}\right.$

How "efficient" is all this?

The size of the solutions of
$$\begin{cases} f_1(x_1,\ldots,x_n) &= & 0 \\ f_2(x_1,\ldots,x_n) &= & 0 \\ \vdots & \vdots & \vdots \\ f_n(x_1,\ldots,x_n) &= & 0 \end{cases}$$
where size of $f_i = (d, L)$

How "efficient" is all this?

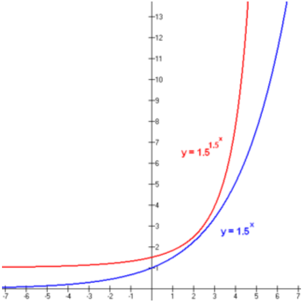
The size of the solutions of $\left\{\begin{array}{rcl} f_1(x_1,\ldots,x_n) & = & 0 \\ f_2(x_1,\ldots,x_n) & = & 0 \\ \vdots & \vdots & \vdots \\ f_n(x_1,\ldots,x_n) & = & 0 \end{array}\right.$

where size of $f_i = (d, L)$

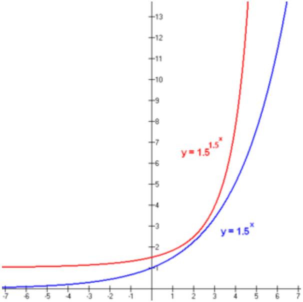
is bounded by and generically equal to

$$\boxed{\left(d^n, nd^{n-1}L\right)}$$

The output is already exponential!!!

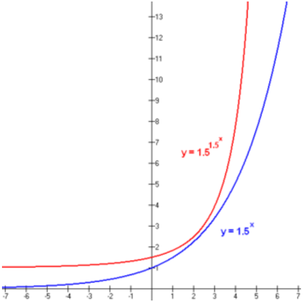


The output is already exponential!!!



And moreover:

The output is already exponential!!!



And moreover: Complexity of Computing Gröbner bases is doubly exponential

Changing the model

Changing the model

- Probabilistic algorithms

Changing the model

- Probabilistic algorithms
- Computations "over the Reals"

Changing the model

- Probabilistic algorithms
- Computations "over the Reals"
- Homotopy methods
- ...

Computing over the reals: BSS-machine

Computing over the reals: BSS-machine

is a **Random Access Machine**

Computing over the reals: BSS-machine

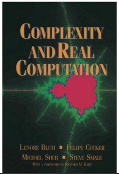
is a **Random Access Machine**

with registers that can store

arbitrary real numbers

Computing over the reals: BSS-machine

is a **Random Access Machine** with registers that can store **arbitrary real numbers** and that can compute rational functions over reals at unit cost

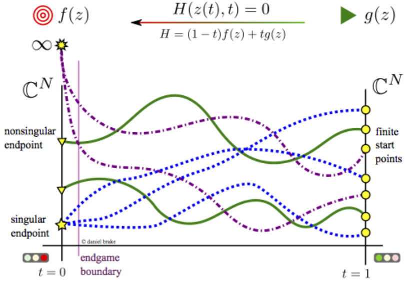


Steve Smale's 17th's problem (1998)

Steve Smale's 17th's problem (1998)

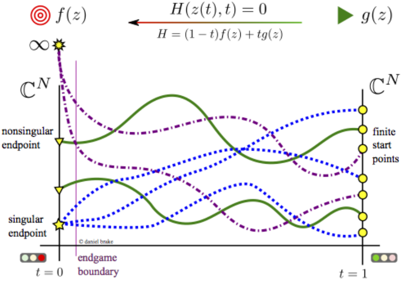
is there an algorithm which computes an **approximate solution** of a system of polynomials in **time polynomial on the average,** in the size of the input ?

Homotopies



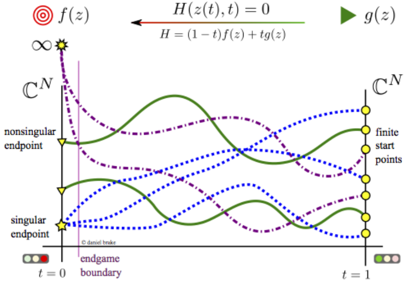
$f(z)$
$H(z(t),t) = 0$
$H = (1-t)f(z) + tg(z)$
$g(z)$
∞
$\mathbb{C}^N$
$\mathbb{C}^N$
nonsingular endpoint
finite start points
singular endpoint
© daniel brake
endgame boundary
$t = 0$
$t = 1$

Homotopies



- Start with an "easy" system

Homotopies



- Start with an "easy" system
- "Chase" the roots with an homotopy + Newton's method

Numerical Algebraic Geometry

By using homotopies, one can compute

Numerical Algebraic Geometry

By using homotopies, one can compute

- Irreducible components

Numerical Algebraic Geometry

By using homotopies, one can compute

- Irreducible components
- Multiplicities

Numerical Algebraic Geometry

By using homotopies, one can compute

- Irreducible components
- Multiplicities
- Irreducible decomposition

Numerical Algebraic Geometry

By using homotopies, one can compute

- Irreducible components
- Multiplicities
- Irreducible decomposition
- Dimension

Numerical Algebraic Geometry

By using homotopies, one can compute

- Irreducible components
- Multiplicities
- Irreducible decomposition
- Dimension
- . . .

Popular software in Computer Algebra

- Maple
- Mathematica
- Bertini
- CoCoA
- Macaulay2
- SageMath
- Singular

Maple

Maple

- Developed by MapleSoft

Maple

- Developed by MapleSoft
- Core Team: Waterloo (Canada)

Maple

- Developed by MapleSoft
- Core Team: Waterloo (Canada)
- http://www.maplesoft.com/

Mathematica

Mathematica

- Developed by Wolfram

Mathematica

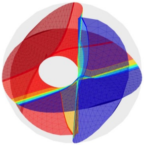
- Developed by Wolfram
- Core Team: Champaign, IL (USA)

Mathematica

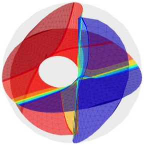
- Developed by Wolfram
- Core Team: Champaign, IL (USA)

https://www.wolfram.com/mathematica/

Bertini

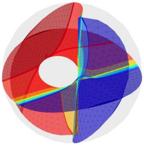


Bertini



- Free software

Bertini

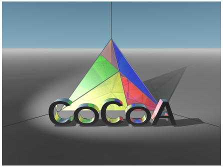


- Free software
- Core Team: University of Notre Dame (USA)

Bertini

- Free software
- Core Team: University of Notre Dame (USA)
- https://bertini.nd.edu/

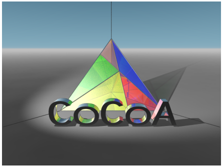
CoCoA



CoCoA

- Free software

CoCoA



- Free software
- Core Team: University of Genoa (Italy)

CoCoA

- Free software
- Core Team: University of Genoa (Italy)
- http://cocoa.dima.unige.it/

Macaulay2

Macaulay2

- Free software

Macaulay2

- Free software
- Core Team: University of Illinois at Urbana-Champaign (USA)

Macaulay2

- Free software
- Core Team: University of Illinois at Urbana-Champaign (USA)

http://www.math.uiuc.edu/Macaulay2/

SageMath

SageMath

- Free open-source

SageMath

- Free open-source
- Core Team: Worlwide

SageMath

- Free open-source
- Core Team: Worlwide
- http://www.sagemath.org/

Singular

Singular

- Free software

Singular

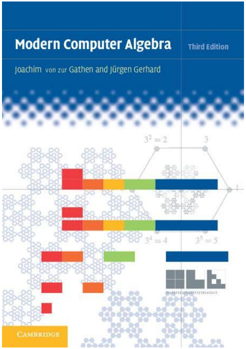
- Free software
- Core Team: University of Kaiserlautern (Germany)

Singular

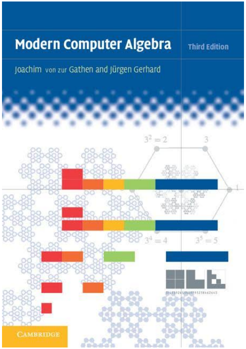
- Free software
- Core Team: University of Kaiserlautern (Germany)
- https://www.singular.uni-kl.de/

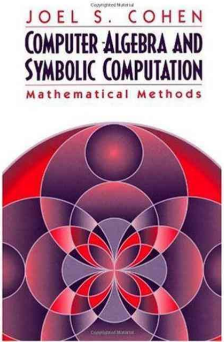
To Learn More about this Area...

To Learn More about this Area...

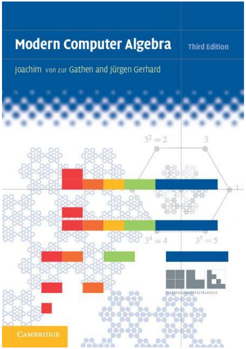


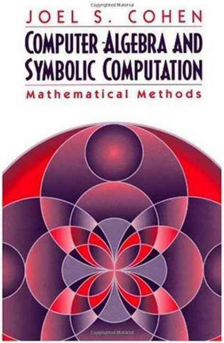
To Learn More about this Area...

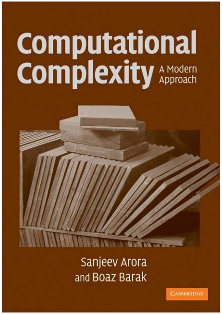




To Learn More about this Area...

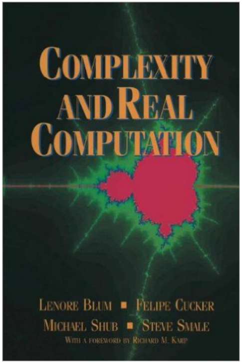




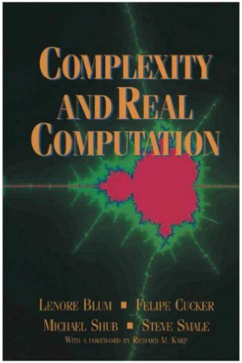


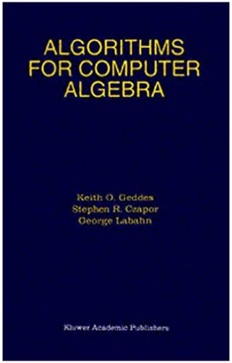
To Learn More about this Area...

To Learn More about this Area...

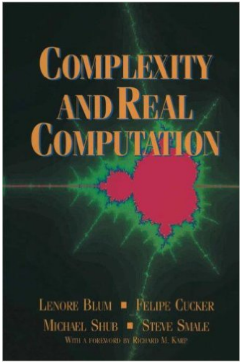


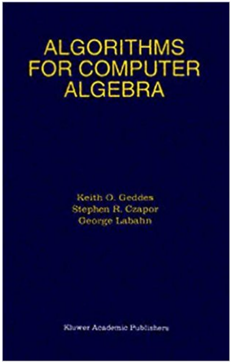
To Learn More about this Area...

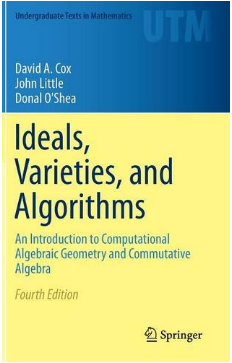




To Learn More about this Area...

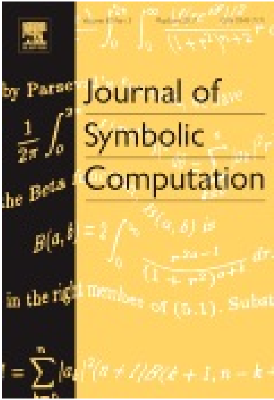




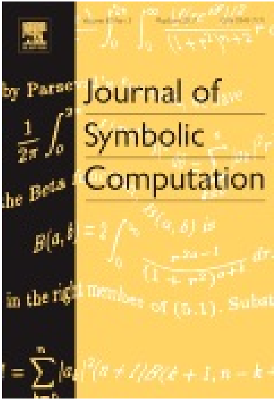


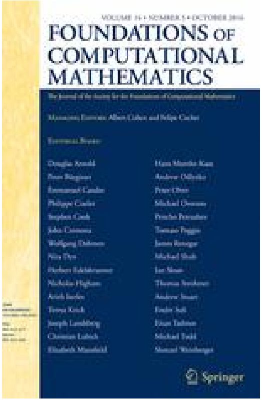
To publish in this area

To publish in this area

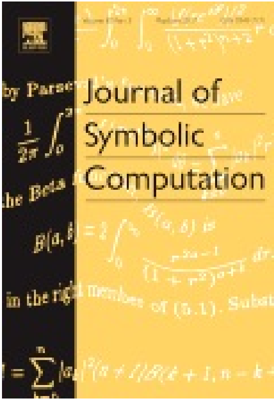


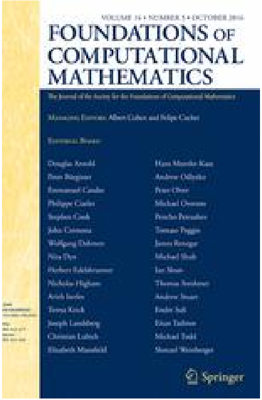
To publish in this area

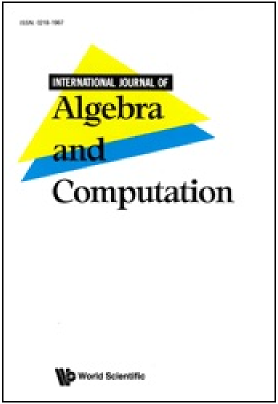




To publish in this area



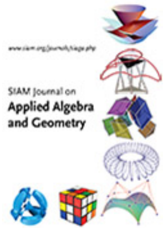




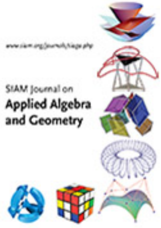
To publish in this area

To publish in this area

To publish in this area



To publish in this area



Thanks!