Principia Computer Algebra

Christoph Koutschan

Research Institute for Symbolic Computation
Johannes Kepler Universität Linz, Austria

Computer Algebra Systems
25.10.2010

Principia Computer Algebra

1. Never trust a CAS!
2. Keep control of your expressions!
3. Know what's behind!
4. Never use `simplify`!

1. Never trust a CAS!

Strange integral

Type the following indefinite integral into Mathematica:

```
Integrate[Exp[xy], x]
```

Strange integral

Type the following indefinite integral into Mathematica:
`Integrate[Exp[xy], x]`

$\longrightarrow$ Seemingly wrong output is mostly caused by incorrect input.

Other common instance:

```
x=1
...
Exponent[x^2+1,x]
```

$\longrightarrow$ use `Clear[x]`

But there are enough "real" bugs as well...

What is a polynomial?

```
PolynomialQ[1/x, {x,y}]
PolynomialQ[1/x, y]
PolynomialQ[1/x, {}]
PolynomialQ[1/x]
PolynomialQ[Sin[x], {x,y}]
PolynomialQ[Sin[x], y]
PolynomialQ[Sin[x], {}]
PolynomialQ[Sin[x]]
PolynomialQ[Sqrt[x], {}]
PolynomialQ[Sqrt[x^2-1], {}]
```

Try also Maple:

```
type(sqrt(x^2-1), polynom(anything, []));
```

Chebyshev polynomials

Try the following in Mathematica:

```
Table[ChebyshevU[n, -1], n, -5, 5]
```

and compare with Maple:

```
[seq(simplify(ChebyshevU(n, -1),'ChebyshevU'),
n=-5..5)];
```

A limit problem

What is the problem in

```
Limit[x^k*Binomial[-2,k], k->Infinity,
Assumptions->Abs[x]<1]
```

A limit problem

What is the problem in
`Limit[x^k*Binomial[-2,k], k->Infinity, Assumptions->Abs[x]<1]`

Hint: try `FullSimplify[Binomial[-2,k]]`

A limit problem

What is the problem in
`Limit[x^k*Binomial[-2,k], k->Infinity, Assumptions->Abs[x]<1]`

Hint: try `FullSimplify[Binomial[-2,k]]`

There is a general problem with generic results:
What should `Sin[Pi]/Sin[x]` evaluate to?

A binomial sum

We consider the double summation problem

$$\sum_{i=0}^{m}\sum_{j=0}^{n}(-1)^{i+j}\binom{i+j}{i}\binom{m}{i}\binom{n}{j}$$

in Mathematica

```
Sum[(-1)^(i+j)*Binomial[i+j,i]*Binomial[m,i]*Binomial[n,j],
{i,0,m}, {j,0,n}]
```

and Maple

```
sum(sum((-1)^(i+j)*binomial(i+j,i)*binomial(m,i)*
binomial(n,j), i=0..m), j=0..n);
```

$\longrightarrow$ What happens if we set $m = n$?

the square root bug (`Sqrt[x^2] == x`)

Limits of Zeta functions

```
as = n>1 && Element[n,Integers] && a>0 && b>0 && k<n
expr1 = x^k*Zeta[n, a+b*x]
expr2 = x^k*Zeta[n, 1+a+b*x]
Limit[expr1, x->Infinity, Assumptions->as]
Limit[expr2, x->Infinity, Assumptions->as]
Limit[expr1-expr2, x->Infinity, Assumptions->as]
```

Numerical Evaluation

Compare Mathematica

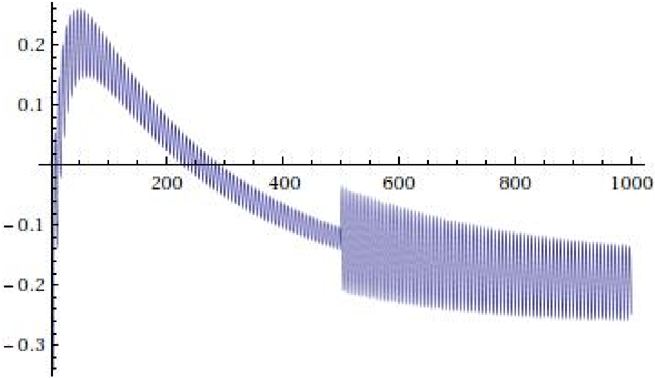
```
N[StruveL[1+I, 700*I], 25]
```

and Maple

```
evalf(StruveL(1+I, 700*I), 25);
```

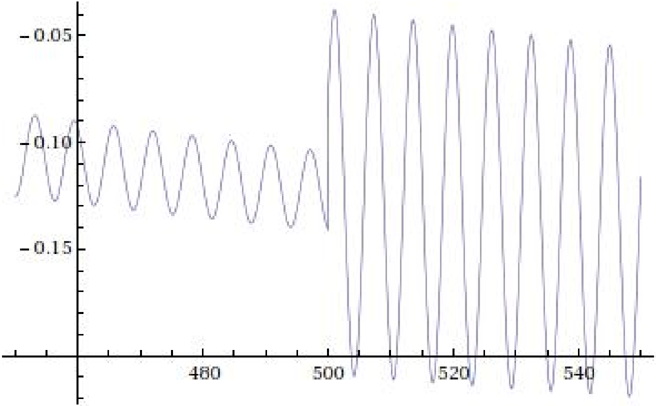
Numerical Evaluation

Mathematica: `Plot[Re[StruveL[1+I, x*I]], {x,0,1000}]`



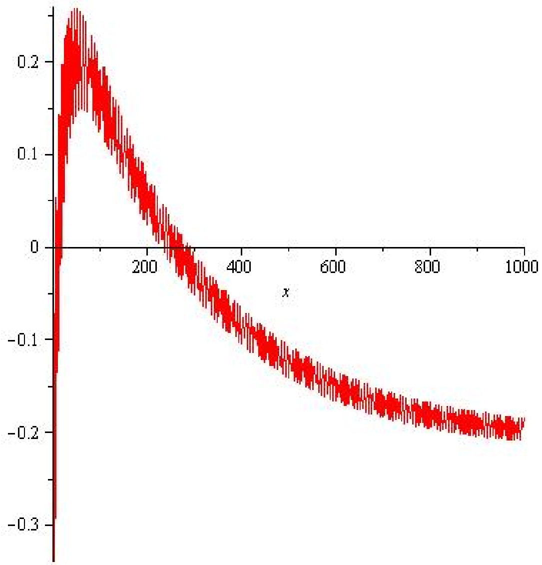
Numerical Evaluation

Mathematica: `Plot[Re[StruveL[1+I, x*I]], {x,480,520}]`



Numerical Evaluation

Maple: `plot(Re(StruveL(1+I, I*x)), x=0..1000);`



Principia Computer Algebra

2. Keep control of your expressions!

PolynomialGCD (1)

We write a little program that computes the gcd of two polynomials using Euklid's algorithm.

PolynomialGCD (1)

We write a little program that computes the gcd of two polynomials using Euklid's algorithm.

```
MyPolynomialGCD[p1_, p2_, x_] :=
Module[{e1, e2},
  If[p1*p2 === 0, Return[p1+p2]];
  While[p2 =!= 0,
    While[(e1=Exponent[p1,x]) >= (e2=Exponent[p2,x]),
      p1 = p1 - (Coefficient[p1,x,e1]/Coefficient[p2,x,e2])
    ];
  {p1, p2} = {p2, p1};
  ];
  Return[p1];
];
```

What's going wrong? Try `MyPolynomialGCD[x^2-1, x-1, x]`

PolynomialGCD (2)

```
MyPolynomialGCD[pp1_, pp2_, x_] :=
Module[{p1=pp1, p2=pp2, e1, e2},
  If[p1*p2 === 0, Return[p1 + p2]];
  While[p2 =!= 0,
    While[(e1=Exponent[p1,x]) >= (e2=Exponent[p2,x]),
      p1 = p1 - (Coefficient[p1,x,e1]/Coefficient[p2,x,e2]
    ];
    {p1, p2} = {p2, p1};
  ];
  Return[p1];
];
```

What's going wrong? Try `MyPolynomialGCD[x^2-1, x-1, x]`

PolynomialGCD (3)

```
MyPolynomialGCD[pp1_, pp2_, x_] :=
Module[{p1=pp1, p2=pp2, e1, e2},
  If[p1*p2 === 0, Return[p1+p2]];
  While[p2 =!= 0,
    While[(e1=Exponent[p1,x]) >= (e2=Exponent[p2,x]),
      p1 = Expand[p1 -
        (Coefficient[p1,x,e1]/Coefficient[p2,x,e2])*x^(e1-e
    ];
    {p1, p2} = {p2, p1};
  ];
  Return[p1];
];
```

What's going wrong?

Try `MyPolynomialGCD[x^2-1, (y-1)*x-1, x]`

PolynomialGCD (4)

```
MyPolynomialGCD[pp1_, pp2_, x_] :=
Module[{p1=pp1, p2=pp2, e1, e2},
  If[p1*p2 === 0, Return[p1+p2]];
  While[p2 =!= 0,
    While[(e1=Exponent[p1,x]) >= (e2=Exponent[p2,x]),
      p1 = Together[p1 -
        (Coefficient[p1,x,e1]/Coefficient[p2,x,e2])*x^(e1-e
    ];
    {p1, p2} = {p2, p1};
  ];
  Return[p1];
];
```

Nullspace

Compute the nullspace of a small matrix over $\mathbb{Q}(x)$.

Compare Mathematica

```
mat = Table[x-RandomInteger[10], 9, 10]
Timing[ns = NullSpace[mat];]
```

and Maple

```
with(LinearAlgebra);
mat := Matrix((i,j)->x,9,10) + RandomMatrix(9,10);
t := time(); NullSpace(mat); time()-t;
```

Determinant

We compute a Vandermonde determinant.

With Mathematica:

```
mat = Table[1/(x+i)^j, {i,0,7}, {j,0,7}];
Timing[det = Det[mat];]
ByteCount[det]
Timing[Together[det]]
```

With Maple:

```
with(LinearAlgebra);
mat := Matrix((i,j) -> 1/(x+i-1)^(j-1), 8, 8);
t := time(); Determinant(mat); time()-t;
```

3. Know what's behind!

Examples (Mathematica):

```
Exponent[x^2+x+1-x*(x-1),x]
```

Examples (Maple):

```
degree(x^2+x+1-x*(x-1),x);
```

Examples (Mathematica):

```
Exponent[x^2+x+1-x*(x-1),x]
Timing[Exponent[Product[x+y+z+i, i,100], x]]
```

Examples (Maple):

```
degree(x^2+x+1-x*(x-1),x);
lcoeff(x^2+x+1-x*(x-1),x);
```

Examples (Mathematica):

```
Exponent[x^2+x+1-x*(x-1),x]
Timing[Exponent[Product[x+y+z+i, i,100], x]]
Denominator[1/(x+1)-x/(x-1)]
```

Examples (Maple):

```
degree(x^2+x+1-x*(x-1),x);
lcoeff(x^2+x+1-x*(x-1),x);
denom(1/(x+1)-x/(x-1));
```

Examples (Mathematica):

```
Exponent[x^2+x+1-x*(x-1),x]
Timing[Exponent[Product[x+y+z+i, i,100], x]]
Denominator[1/(x+1)-x/(x-1)]
Denominator[x^(-n)]
```

Examples (Maple):

```
degree(x^2+x+1-x*(x-1),x);
lcoeff(x^2+x+1-x*(x-1),x);
denom(1/(x+1)-x/(x-1));
denom(x^(-n));
denom(1/x^n);
```

Examples (Mathematica):

```
Exponent[x^2+x+1-x*(x-1),x]
Timing[Exponent[Product[x+y+z+i, i,100], x]]
Denominator[1/(x+1)-x/(x-1)]
Denominator[x^(-n)]
```

Examples (Maple):

```
degree(x^2+x+1-x*(x-1),x);
lcoeff(x^2+x+1-x*(x-1),x);
denom(1/(x+1)-x/(x-1));
denom(x^(-n));
denom(1/x^n);
```

$\longrightarrow$ Many commands do not work mathematically but syntactically.

Internal representation

Try the following (in Mathematica):

What is the largest integer appearing in the expression $\frac{x}{20} + \frac{y}{21}$?

```
Max[Cases[x/20 + y/21, _Integer, Infinity]]
```

Test whether $A \in \mathbb{Q}^{3\times 3}$:

```
A = Table[1/(i+j-1), {i,3}, {j,3}]
MatchQ[A, (_Rational)....]
```

Get rid of the $O(\cdot)$ term in the series expansion:

```
ser = Series[Log[1-x], x,0,10]
ser - Last[ser]
```

Internal representation

Try the following (in Mathematica):

What is the largest integer appearing in the expression $\frac{x}{20} + \frac{y}{21}$?

```
Max[Cases[x/20 + y/21, _Integer, Infinity]]
```

Test whether $A \in \mathbb{Q}^{3\times 3}$:

```
A = Table[1/(i+j-1), {i,3}, {j,3}]
MatchQ[A, (_Rational)....]
```

Get rid of the $O(\cdot)$ term in the series expansion:

```
ser = Series[Log[1-x], x,0,10]
ser - Last[ser]
```

$\longrightarrow$ Often it is useful to know about the internal representation.

4. Never use simplify!

Why?

- Result is unpredictable: compare
  `Simplify[x^4+4(x^3+x)+6x^2+1]`
  `Simplify[(x^2+1) (x^2-1)]`
- You don't know what's behind.
- Usually a more specific command can be (and should be) used.

$\longrightarrow$ Why are there two different commands, `Factor` and `FactorInteger`?

We consider the following expression:

```
RandomPolynomial[x_, d_] :=
Sum[RandomInteger[100]*x^i, i, 0, d]
expr =
Sum[RandomPolynomial[x,5]/RandomPolynomial[x,5],50];
```

Now compare how long these manipulations take:

```
Timing[Expand[expr];]
Timing[Together[expr];]
Timing[Simplify[expr];]
Timing[FullSimplify[expr];]
```

Trigonometric functions

```
Simplify[Sin[x]^2+Cos[x]^2]

Expand[Sin[x]^2+Cos[x]^2, Trig->True]
```

Factorials

```
Simplify[(n+1)!/n!]

FunctionExpand[(n+1)!/n!]
```

Logical expressions

```
Simplify[x && (x || y)]
```

```
LogicalExpand[x && (x || y)]
```

Other commands in Mathematica for logical simplification:

- `Reduce`
- `Resolve`
- `CylindricalDecomposition`

Solving equations

Similar things hold for other general purpose commands like `solve`, etc.

Instead, use special purpose solvers like

- `NullSpace`
- `GroebnerBasis`
- `fsolve`