

Maple

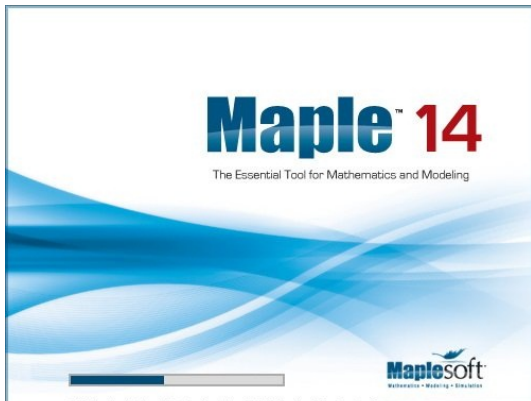
Christoph Koutschan

Research Institute for Symbolic Computation
Johannes Kepler Universität Linz, Austria

Computer Algebra Systems
18.10.2010



Maple



© Maplesoft, a division of Waterloo Maple Inc., 2010 Maplesoft and Maple are trademarks



Maple



Maple



Maplesoft is located in Waterloo, Canada.

Maple is a commercial product:

- student edition: US\$99
- academic edition: US\$995
- professional edition: US\$1895



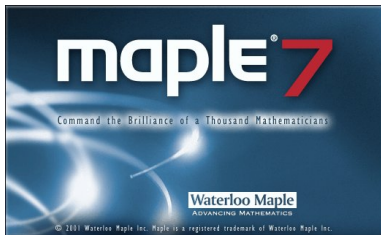
Maple Slogans



- Maple 6: Performance • Innovation



Maple Slogans



- Maple 6: Performance • Innovation
- Maple 7-9: Command the Brilliance of a Thousand Mathematicians



Maple Slogans



- Maple 6: Performance • Innovation
- Maple 7-9: Command the Brilliance of a Thousand Mathematicians



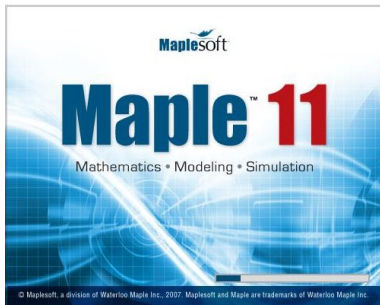
Maple Slogans



- Maple 6: Performance • Innovation
- Maple 7-9: Command the Brilliance of a Thousand Mathematicians
- Maple 10: Harnessing the Power of Mathematics



Maple Slogans



- Maple 6: Performance • Innovation
- Maple 7-9: Command the Brilliance of a Thousand Mathematicians
- Maple 10: Harnessing the Power of Mathematics
- Maple 11: Mathematics • Modeling • Simulation



Maple Slogans



- Maple 6: Performance • Innovation
- Maple 7-9: Command the Brilliance of a Thousand Mathematicians
- Maple 10: Harnessing the Power of Mathematics
- Maple 11: Mathematics • Modeling • Simulation
- Maple 12-14: The Essential Tool for Mathematics and Modeling



History

- First idea: November 1980 at the University of Waterloo (as an alternative to Macsyma)
- 1982: Maple 1.0
- 1985: Maple 3.3 (first commercial version)
- 1988: Waterloo Maple Inc. founded by Keith Geddes and Gaston Gonnet
- 1990: Maple V
- 2003: standard interface (Java based, two-dimensional input)
- 2010: Maple 14



Keith Geddes



- born 1948
- 1973 PhD in Computer Science at University of Toronto
- Director of the Symbolic Computation Group (University of Waterloo)
- Scientific Director at the Ontario Research Centre for Computer Algebra



Gaston Gonnet



- 1977 PhD in Computer Science (University of Waterloo)
- 1980 co-founder of the Symbolic Computation Group at the University of Waterloo
- quitted the Maple company in the mid 90s
- currently professor at ETH Zürich



Some other mathematical software that used to use Maple:

- MathCad (1994–2006): Maple-derived algebra engine, Mathsoft Kernel Maple (MKM)
- Symbolic Math Toolbox in MATLAB contained a portion of the Maple 10 engine
- some versions of the mathematical editor Scientific Workplace included Maple as a computational engine



Architecture

Maple consists of four parts:

- kernel: most procedures written in C
- interface (worksheet)
- library: many packages, e.g., LinearAlgebra, Groebner, etc., written in Maple language
- share library: user contributions Maple Application Center, visit <http://www.mapleapps.com>.

As Mathematica, Maple uses the GNU Multiple Precision library (GMP) for big integer arithmetic



Interface

- standard notebook
- classical notebook



Quick start into Maple

Now, let's jump into the cold water. . .



Quick start into Maple

Now, let's jump into the cold water...

```
2+3^4;
```

```
expand((x+4)^2);
```

```
x+x=2*x;
```

```
x := 3;
```

```
sin(Pi);
```

```
{1,1,2,3,5};
```



Some (confusing) differences between Maple and Mathematica:

	Maple	Mathematica
new line	⟨Shift⟩ + ⟨Return⟩	⟨Return⟩
execute	⟨Return⟩	⟨Shift⟩ + ⟨Return⟩
semicolon	required at line end	no output
colon	no output	syntax error
=	equation	assignment
==	syntax error	equation
:=	assignment	delayed assignment
{...}	set	list/set
[...]	list	function application
(...)	function application/grouping	grouping
commands	lower case letter (usually)	upper case letter



Getting Help

?expand

is the same as
info(expand)

Furthermore there are:

usage(expand);

example(expand);

related(expand);



Assignments

```
x := 4;
```

```
x := "x";
```

```
x := 'x';
```

```
restart;
```

Aliasing:

```
alias(s=sin(x),c=cos(x));
```

```
alias(J=BesselJ);
```

```
diff(BesselJ(n,sin(x)),x$3);
```



Functions in Maple

```
f := x -> sin(x);  
f(Pi);  
diff(f(z),z);  
diff(f(x),x$3);
```



Basic commands

For polynomials (and rational functions):

`expand`, `factor`, `degree`, `collect`, `numer`, `denom`, `indets`, `gcd`,
`lcm`, `gcdex`, etc.

For integers:

`ifactor`, `igcd`, `ilcm`, `igcdex`, `isqrt`, `iroot`, etc.



Summation and Integration

Note the difference between

- `sum`: symbolic summation
- `add`: add up an explicit sequence of values

Examples:

```
sum(i, i=0..n);
```

```
add(i, i=0..100);
```

The same applies to `product` and `mul`.

Definite and indefinite integrals:

```
int(sin(x), x);
```

```
int(sin(x), x=0..Pi);
```

Inert commands: `Sum`, `Int`, `Product`



Evaluation and Substitution

How can we force Maple to answer the question $\sqrt{5} > 2$?



Evaluation and Substitution

How can we force Maple to answer the question $\sqrt{5} > 2$?

Use special commands for evaluation (`simplify` does not help!):

- `evalf`: numeric evaluation
- `evalb`: boolean evaluation
- `eval`: symbolic evaluation at some point

Examples:

```
evalb(evalf(sqrt(5))>2);
```

```
subs(x=0, sin(x));
```

```
eval(sin(x), x=0);
```



Maple packages

```
?index,packages  
with(LinearAlgebra);  
LinearAlgebra[RandomMatrix](2);
```



Maple packages

```
?index,packages
```

```
with(LinearAlgebra);
```

```
LinearAlgebra[RandomMatrix](2);
```

Doing some linear algebra:

```
A := Matrix([[1,2,3],[3,4,5],[9,8,7]]);
```

```
b := Vector([5,6,7]);
```

```
A.b;
```

```
<A|b>;
```



Programming in Maple

- procedural programming language
- dynamically typed (try ?type)
- certain keywords:
 - `if ... then ... elif ... else ... fi`
 - `for ... from ... to ... by ... while ... do ... od`
 - `for ... in ... while ... do ... end do`
 - `proc(...)` ... `end`

Experiment: try `if+while`; in Maple and compare with `If+While` in Mathematica!

Why are there no keywords in Mathematica?



The unavoidable Fibonacci example



The unavoidable Fibonacci example

```
fib := proc(n::nonnegint)
  option remember;
  if n<2 then
    return(n);
  else
    return(fib(n-1)+fib(n-2));
  fi;
end:
```



Experiments

View code of procedure that are written in Maple language:

```
print(Groebner[Basis]);  
interface(verboseproc=3);  
print(factor);  
print(diff);
```

View internal structure of an expression:

```
dismantle(x-3*y^2);  
dismantle[hex](x-3*y^2);
```

