The Fine Art of Plumbing: Bringing New Advances in Computer Algebra into General Purpose Software

John May

Mathematical Software

MapleSoft

April 2009
Maple, other M’s, Sage, etc are general purpose computing software. So: User Driven.
Maple, other M’s, Sage, etc are general purpose computing software. So: User Driven.

- Users are not stupid.
Introduction

Maple, other M’s, Sage, etc are *general purpose* computing software. So: User Driven.

- Users are not stupid.
- Users are not computer algebraists
Introduction

Maple, other M’s, Sage, etc are *general purpose* computing software. So: User Driven.

- Users are not stupid.
- Users are not computer algebraists
- (Users might be rocket scientists)
Maple, other M’s, Sage, etc are *general purpose* computing software. So: User Driven.

- Users are not stupid.
- Users are not computer algebraists
- (Users might be rocket scientists)
- Users want enough rope to shoot themselves in the foot.
Example: solve

\[ \text{solve } x^2 + 3x - 2 \]

\[ -\frac{3}{2} \pm \frac{1}{2} \sqrt{17} \]
Example: solve

\[
solve \ x^2 + 3x - 2
\]

\[-3/2 \pm 1/2 \sqrt{17}\]

\[
solve \ \cos(x)^2 + 2 \cos(x) - 1
\]

arccos\((-3/2 \pm 1/2 \sqrt{17})\)
Example: solve

solve $x^2 + 3x - 2$

$-3/2 \pm 1/2 \sqrt{17}$

solve $\cos(x)^2 + 2 \cos(x) - 1$

$\arccos(-3/2 \pm 1/2 \sqrt{17})$

Actually...

$(1 - 2 b_1) \arccos \left(-3/2 + 1/2 \sqrt{17}\right) + 2 \pi z_1,$

$(1 - 2 b_2) \pi + (2 b_2 - 1) \arccos \left(3/2 + 1/2 \sqrt{17}\right) + 2 \pi z_2$
Example: solve

\[ \text{solve } \cos(x)^2 + 2.31 \cos(x) - 1.531 \]

\[ 1.003162480, 3.141592654 - 1.707275385i \]
Example: solve

\[ \text{solve } \cos(x)^2 + 2.31 \cos(x) - 1.531 \]

\[ 1.003162480, 3.141592654 - 1.707275385i \]

Or...

\[ (1 - 2 b_3) \arccos \left( -\frac{231}{200} + \frac{1}{200} \sqrt{114601} \right) + 2 \pi z_3, \]

\[ (-1 + 2 b_4) \arccos \left( \frac{231}{200} + \frac{1}{200} \sqrt{114601} \right) + (1 - 2 b_4) \pi + 2 \pi z_4 \]
Example: solve

solve \( \cos(x)^2 + 2.31 \cos(x) - 3.142 \ a \) (for \( x \))

\[
(1.0 - 2.0 \ b_1) \arccos \left( -1.1550 + 0.0050 \sqrt{53361.0 + 125660.0 \ a} \right)
\]

\[
+ 6.283185307 \ z_1 ,
\]

\[
(-1.0 + 2.0 \ b_1) \arccos \left( 1.1550 + 0.0050 \sqrt{53361.0 + 125660.0 \ a} \right)
\]

\[
+ 3.141592654 - 6.283185308 \ b_1 + 6.283185307 \ z_1
\]
Example: solve

\[
\text{solve } \cos(x)^2 + 2.31 \cos(x) - 3.142 a \text{ (for } x) \\

(1.0 - 2.0 b_1) \arccos \left(-1.1550 + 0.0050 \sqrt{53361.0 + 125660.0 a}\right) \\
+ 6.283185307 z_1, \\

(-1.0 + 2.0 b_1) \arccos \left(1.1550 + 0.0050 \sqrt{53361.0 + 125660.0 a}\right) \\
+ 3.141592654 - 6.283185308 b_1 + 6.283185307 z_1 \\
\]

...or maybe 3.142 was supposed to be \(\pi\). Probably not.
Example: solve

\[
\text{solve } \cos(x)^{2.1} + 2.31 \cos(x) - 3.1415 a \text{ (for } x) \\
(1.0 - 2.0 b_1) \arccos \left( \left( \text{RootOf} \left( 2000 \cdot Z^{21} + 4620 \cdot Z^{10} - 6283 \cdot a \right) \right)^{10} \right) + 6.283185307 z_1
\]
Example: solve

\[ \text{solve } \cos(x)^{2.1} + 2.31 \cos(x) - 3.1415 a \text{ (for } x) \]

\[ (1.0 - 2.0 b_1) \arccos \left( \left( \text{RootOf} (2000 \cdot Z^{21} + 4620 \cdot Z^{10} - 6283 \cdot a) \right)^{10} \right) \]
\[ + 6.283185307 z_1 \]

\[ \text{solve } \cos(x)^{2.1} + 2.31 \cos(x) \leq 3.1415 a \text{ (for } x) \]
Example: solve

\[
solve \cos(x)^{2.1} + 2.31 \cos(x) - 3.1415 a \text{ (for } x) \\
(1.0 - 2.0 b_1) \arccos \left( \left( \text{RootOf} \left( 2000 Z^{21} + 4620 Z^{10} - 6283 a \right) \right)^{10} \right) + 6.283185307 z_1
\]

\[
solve \cos(x)^{2.1} + 2.31 \cos(x) \leq 3.1415 a \text{ (for } x) \\
please make it stop
\]
Upshot

Bad News: User’s problems often are not the problems on which most of the research is done:

1. polynomial equation solving
2. linear system solving
3. polynomial system solving

Good News: User’s problems reduce to these things with a bit of (sometimes heuristic) work.

Bad News: there are many theory vs. practice gaps e.g. variable (not monomial) ordering in polynomial system solving.

User does not know what to do with a Gröbner basis or CAD.
Bad News: User’s problems often are not the problems on which most of the research is done:

1. polynomial equation solving  
2. linear system solving  
3. polynomial system solving

Good News: User’s problems reduce to these things with a bit of (sometimes heuristic) work.
Bad News: User’s problems often are not the problems on which most of the research is done:

1. polynomial equation solving
2. linear system solving
3. polynomial system solving

Good News: User’s problems reduce to these things with a bit of (sometimes heuristic) work.

Bad News: there are many theory vs. practice gaps

  e.g. variable (not monomial) ordering in polynomial system solving.
  e.g. user does not know what to do with a Gröbner basis or CAD.
Charts

Polynomials solved in $\approx 4000$ regression tests

Problems are often small - small problems must be done FAST. Time for preprocessing and dispatch can easily eat speed ups in low level routines.
Polynomials solved in $\approx 4000$ regression tests

Problems are often not complicated - Sophisticated algorithms often get beat by simple heuristics for special cases.
Final Platitudes

- There is a fine line to walk between special case handling and quick dispatch $\leadsto$ lots of experimental work.

\[ x^4 - x^3 + 3 = 0, \quad x^0 - 2x^1 + 3x^2 = 0, \ldots, \quad 999x^{998} - 1000x^{999} + 1001x^{1000} = 0 \]

- User interface can really get in the way: explicit back-substituted solutions to large systems are "wall paper." Same for explicit case discussions of systems with parameters.

- Theoretic seeming computational results can really be practical: one bottleneck for the polynomial system solver in Maple 10 was factoring polynomials with coefficients in algebraic extensions of $\mathbb{Q}$. 
Final Platitudes

- There is a fine line to walk between special case handling and quick dispatch $\leadsto$ lots of experimental work.
- Dispatch needs to be smart: e.g.
  \[ x_0^4 - 2x_0^3 + 3 = 0, \quad x_0 - 2x_1 + 3x_2 = 0, \]
  \[ \ldots, \]
  \[ 999x_{998} - 1000x_{999} + 1001x_{1000} = 0 \]
Final Platitudes

- There is a fine line to walk between special case handling and quick dispatch \(\sim\) lots of experimental work.

- Dispatch needs to be smart: e.g.
  \[ x_0^4 - 2x_0^3 + 3 = 0, \quad x_0 - 2x_1 + 3x_2 = 0, \]
  \[ \ldots, \]
  \[ 999x_{998} - 1000x_{999} + 1001x_{1000} = 0 \]

- User interface can really get in the way: explicit back-substituted solutions to large systems are “wall paper”. Same for explicit case discussions of systems with parameters.
• There is a fine line to walk between special case handling and quick dispatch $\leadsto$ lots of experimental work.

• Dispatch needs to be smart: e.g.
  
  \[ x_0^4 - 2x_0^3 + 3 = 0, \quad x_0 - 2x_1 + 3x_2 = 0, \]
  
  \[ \ldots, \]
  
  \[ 999x_{998} - 1000x_{999} + 1001x_{1000} = 0 \]

• User interface can really get in the way: explicit back-substituted solutions to large systems are “wall paper”. Same for explicit case discussions of systems with parameters.

• Theoretic seeming computational results can really be practical: one bottleneck for the polynomial system solver in Maple 10 was factoring polynomials with coefficients in algebraic extensions of $\mathbb{Q}$. 
It never ENDS