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

John May

Mathematical Software

Maplesoft

April 2009

Introduction

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

Introduction

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)

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)
- Users want enough rope to shoot themselves in the foot.

Example: solve

`solve` $x^2 + 3x - 2$

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

Example: solve

`solve` $x^2 + 3x - 2$

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

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

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

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

`solve` $cos(x)^2 + 2.31\,cos(x) - 1.531$

$$1.003162480,\ 3.141592654 - 1.707275385i$$

Example: solve

`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

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

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

$$+6.283185307\, z_1,$$

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

$$+3.141592654 - 6.283185308\, b_1 + 6.283185307\, z_1$$

...or maybe 3.142 was supposed to be π. Probably not.

Example: solve

`solve` $cos(x)^{2.1} + 2.31\, cos(x) - 3.1415\, a$ (for x)

$$(1.0 - 2.0\, b_1) \arccos\left(\left(\mathrm{RootOf}\left(2000\, \_Z^{21} + 4620\, \_Z^{10} - 6283\, a\right)\right)^{10}\right)$$

$$+6.283185307\, z_1$$

Example: solve

`solve` $cos(x)^{2.1} + 2.31\,cos(x) - 3.1415\,a$ (for x)

$$(1.0 - 2.0\,b_1)\,\mathrm{arccos}\left(\left(\mathrm{RootOf}\left(2000\,\_Z^{21} + 4620\,\_Z^{10} - 6283\,a\right)\right)^{10}\right)$$
$$+6.283185307\,z_1$$

`solve` $cos(x)^{2.1} + 2.31\,cos(x) \leq 3.1415\,a$ (for x)

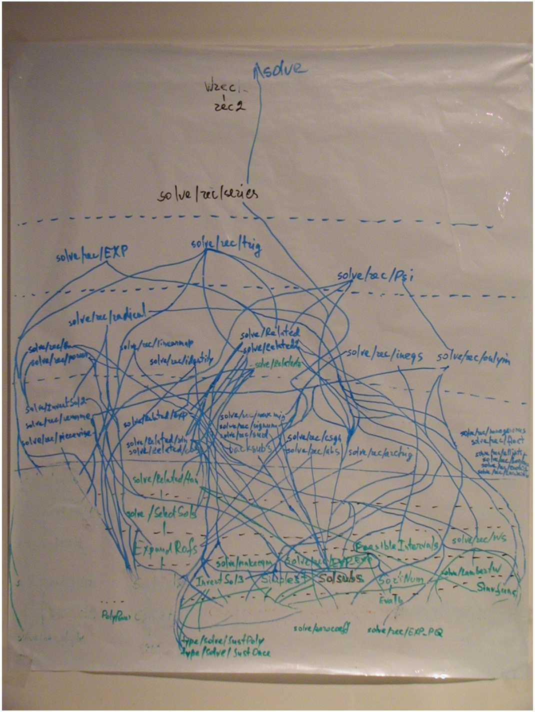
Example: solve

`solve` $cos(x)^{2.1} + 2.31\,cos(x) - 3.1415\,a$ (for x)

$$(1.0 - 2.0\,b_1)\arccos\left(\left(\mathrm{RootOf}\left(2000\,\_Z^{21} + 4620\,\_Z^{10} - 6283\,a\right)\right)^{10}\right)$$
$$+6.283185307\,z_1$$

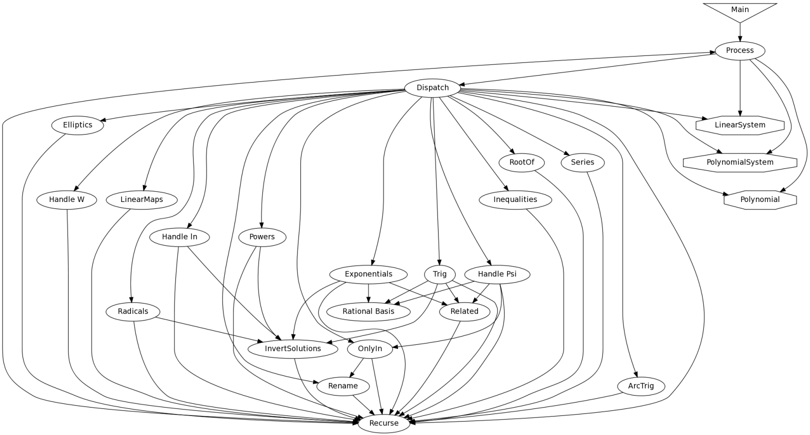
`solve` $cos(x)^{2.1} + 2.31\,cos(x) \leq 3.1415\,a$ (for x)

please make it stop



solve
solve/rec/series
solve/rec/EXP
solve/rec/trig
solve/rec/Psi
solve/rec/radical
solve/Related
solve/rec/ineqs
solve/rec/identity
solve/rec/piecewise
solve/rec/csgn
solve/rec/abs
solve/rec/arctrig
solve/rec/floor
solve/Related/tan
solve/SelectSols
ExpandRofs
PossibleIntervals
solve/rec/WS
solve/makeeqn
InvertSol3
Simplest
Solsubs
solve/lambertW
SinvFuncs
solve/zerocoeff
solve/rec/EXP_PQ
type/solve/SustPoly
type/solve/SustOnce

Solve Call Graph



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

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.

Upshot

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

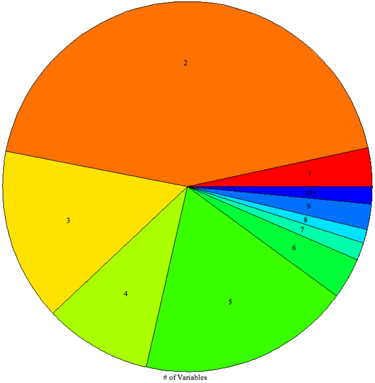
1. polynomial equation solving
2. linear system solving
3. polynomial system solving

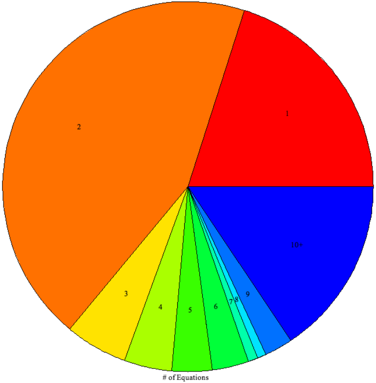
Good News: User's problems reduce to these things with a bit of (sometimes heuristic) work.

Bad News: there are many theory vs. practice gaps
e.g. variable (not monomial) ordering in polynomial system solving.
e.g. user does not know what to do with a Gröbner basis or CAD.

Charts

Polynomials solved in $\approx$ 4000 regression tests

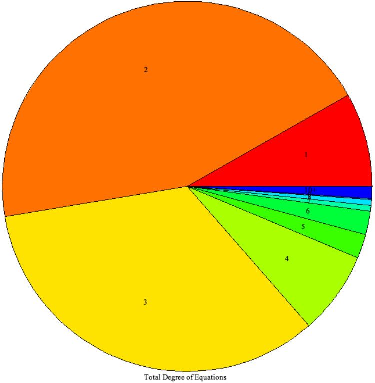


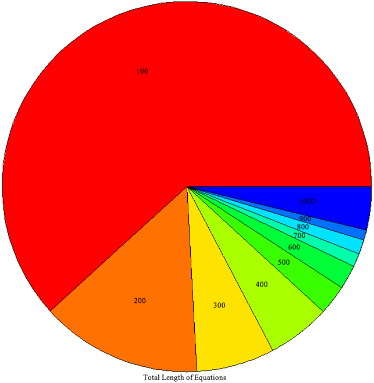


Problems are often small -
small problems must be done FAST. Time for preprocessing and dispatch can easily eat speed ups in low level routines.

Charts

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 $\rightsquigarrow$ lots of experimental work.

Final Platitudes

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

Final Platitudes

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

Final Platitudes

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