Maxima CAS presentation 2015-12-01

Chelton Evans

Abstract

Maxima is a popular copyleft CAS (Computer Algebra System) which can be used for both symbolic and numerical calculation. Chelton Evans will present a series of examples on Maxima's programming language from a pragmatic view (cheat sheet provided) to show how Maxima can be used: For example as a calculator, graph plotting, and solving a quadratic equation. He will offer an introduction to solving an equation $f(x) = 0$ and why this is important. e.g. if a cannon fires - where does it hit? He will also provide an introduction to Newton's method for calculating the square root of two and a more complex solver, Newton's method with third order convergence, with an elementary discussion on numerical solvers and their merit.

Introduction

Introducing Maxima: examples; working towards a research problem investigating 3rd order convergence; showing how to go about forming and solving problems in a CAS environment.

Reference: G. Polya, "How To Solve It" - the best introduction to computer science for problem solving.

Computer Algebra System (CAS)

- Prior to CAS, maths was implemented in other computer languages e.g. basic Tandy PC-8
- HP reverse Polish, high level programming on the stack HP-28S
- the majority of mathematical research uses CAS
- Different CAS systems have similar functionality but very different usability making the knowledge hard to transfer

Example

Inside a cell, enter the commands, multiple commands with the Enter key. To evaluate $\mathrm{shift} + \mathrm{enter}$ key.

$$\mathrm{expand}((x+y)^5)$$
$$y^5 + 5\,x\,y^4 + 10\,x^2\,y^3 + 10\,x^3\,y^2 + 5\,x^4\,y + x^5$$

Maxima availability

- maxima.sourceforge.net
- # apt-get install maxima
- # apt-get install wxMaxima
- MaximaOnAndroid
- Platforms: *nix, Windows, Android

Why use Maxima?

Mathematical laboratory - problem solving, "How to solve it" - Polya

- Primarily symbolic calculation
- Symbolic and numerical calculation in one package
- Big calculator
- Applications: calculus, differential equations, numerical analysis
- free, alternative to commercial software such as Mathematica, Maple, Matlab (primarily numeric). Similar software Sage.

What is symbolic calculation?

- unknown - no limitation, an open ended form of computation
- large numbers (representation)- both in integers and numerical data
- exact forms e.g. $\sqrt{2}$
- embedding knowledge
- algebra manipulation, substitution
- matrices, lists and other data structures
- generalising processing functions to build and evaluate other functions

Using Maxima

• a library call does not modify the original object but makes a new one

• cheat sheet for obtuse syntax

- the syntax or way to do a particular task is often not easy to remember

• there is no sane library - all large scale software has its issues

Infinite evaluation or colliding definitions

Maxima state can be become unstable/corrupted

• Maxima > Restart Maxima - runaway processing,
Maxima > Interrupt - often fails
• Depending on what you are doing, edit and re-define the offending code/procedure

Programming data structures

- list []
- access elements with the array operator
- create a list $\mathrm{makelist}(\mathrm{f04}(a), a, 1, 10)$
- append makes a new list and adds to its end

- set $\{\}$

```
/* Convert */
s1: {2, 5, 9};
s2: listify(s1);
s2[2];

/* Add to end of a list */
t3: [3, 4, 7];
t3: append( t3, [-1]);
```

Functions and procedures

• last statement is the return value
• declare local variables at start in square brackets with commas
e.g. $[y, z]$
• greatest debugging technique - print to the screen

```
f01(n) := block
(
  [y],
  y: 0,
  while n > 1 do
  (
    y: y+n^3,
print(y),
    n: n-1
  ),
  y
);
```

Cannon ball example

The trajectory of a cannon ball

```
can01(a,b) := a*x^2+b*x;
plot2d( can01(-2,5), [x,0,6] );
find_root( can01(-2,5)=0,x,0,6 );
find_root( can01(-2,5)=0,x,1,6 );
diff( can01(-2,5),x);
solve(5-4*x=0,x);
```

Solving $f(x_n) = 0$

- equilibrium can be expressed as an equation of 0:

$$f = g \text{ then } f - g = 0$$

- most problems can be transformed into solving for 0
  e.g. define a metric or distance function with a solution at 0
- local vs global solution and strategy

Derivative

- ratio of two indefinite decreasing infinitesimals

$$f'(x) = \frac{dy}{dx}$$

$$dy \to 0, \; dx \to 0$$

Taylor series

- representing a function about a point $x = a$
- perfect information
- continuity - a piece of string
- algebraic and numerical real world are connected
- even processes without explicit functions can have these numerically constructed

$$f(x) = f(a) + f'(a)(x-a) + f^{(2)}(a)\frac{(x-a)^2}{2!} + f^{(3)}(a)\frac{(x-a)^3}{3!} +$$

$$a = 0: \; f(x) = f(0) + f'(0)(x) + f^{(2)}(0)\frac{x^2}{2!} + f^{(3)}(0)\frac{x^3}{3!} + \dots$$

$$\mathrm{taylor}(\sin(x), x, 0, 6); \;\; x\frac{x^3}{6} + \frac{x^5}{120} + \dots$$

Visual example of Taylor series

- use of a function to make this easier

```
tay(n) := taylor(sin(x),x,0,n);
plot2d( [tay(2), tay(4), tay(6), tay(8), tay(10), tay(12)],
```

Derivation of Newton's method

- Solve $f(x) = 0$

$$f(x_{n+1}) = f(x_n) + (x_{n+1} - x_n)f'(x_n) + \frac{(x_{n+1} - x_n)^2}{2!} f^{(2)}(x_n) + \dots$$

(truncate)

$$f(x_{n+1}) = f(x_n) + (x_{n+1} - x_n)f'(x_n)$$

(assume $f(x_n) \to 0$ decreases)

$$0 = f(x_n) + (x_{n+1} - x_n)f'(x_n) \qquad \text{(solve for } x_{n+1}\text{)}$$

$$0 = \frac{f(x_n)}{f'(x_n)} + x_{n+1} - x_n$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Newton's method as a function

$$f(x) \qquad (f(x) = 0)$$

$$Df(x) \qquad \text{(derivative)}$$

$$\mathrm{newt}(Df, f, x) = x - \frac{f(x)}{Df(x)}$$

Third order Newton method

- Newton's method is a 2nd order approximation
- Arithmetic Newton (AN) method is 3rd order

$$
\begin{aligned}
z_n &= \mathrm{newt}(Df, f, x_n) && \text{(next better approximation)} \\
Df_2 &= \frac{1}{2}(Df(z_n) + Df(x_n)) && \text{(better derivative approximation)} \\
x_{n+1} &= \mathrm{newt}(Df_2, x_n)
\end{aligned}
$$

- AN with explicit mathematics

$$z_n = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$Dg_n = \frac{1}{2}(f'(z_n) + f'(x_n))$$

$$x_{n+1} = x_n - \frac{f(x_n)}{Dg_n}$$

Osama Yusuf Ababneh, *New Newton's Method with Third-order Convergence for Solving Nonlinear Equations*