Workshop on

Computer Algebra System (CAS)

Session: Solving Equations with CAS

10. 04. 2016

P. K. Yadav & K. Taneja

khusboo.taneja@sharda.ac.in

Department of Civil Engineering
Dept. of Computer Sci. & Engg.

SHARDA UNIVERSITY
Beyond Boundaries

Solving Equations with CAS

Engineering and Science is all about equations. We have different types of equations: Algebraic (polynomial), parametric equations, the calculus equations- differential equations, integral equations etc.

The actual Engineering or Scientific challenge is finding the solution of the equations.

The challenge has become bigger due to bigger Engineering or Scientific demand.

The use of computer can now let us solve equations more efficiently and CAS is one of the best “Analytical” tool as opposed to “Numerical” Tool- e.g., MATLAB, Numpy etc.

We now learn to use CAS **Maxima** for solving equations.

 P. K. Yadav

Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

The solve function

 P. K. Yadav

Solving Equation and Equations

The functions:

1. **solve(eqn, var)**
2. **solve([eqn1, eqn2,...], [var1, var2,...])**

solves the **algebraic** equation (expression or function) for the variable (in 1) or variables (in 2) **var** in equation (in 1) or equations (in 2) and returns a **list** of solutions in **var**.

If `expr` is not an 'equation', the equation `expr= 0` is assumed.

Variants of the `solve` and related function can be found using `appropos(solve)` and described using, e.g., `describe(solve)`.

Let us learn the `solve` functions from examples.

Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

The function solve(eq, var)

1. The quadratic equation (you all know this!):

```
(%i1) solve(a*x^2+b*x+c=0,x);
```

$$(\%o1) \quad \left[x = -\frac{\sqrt{b^2-4\,a\,c}+b}{2\,a}, x = \frac{\sqrt{b^2-4\,a\,c}-b}{2\,a}\right]$$

let us define, a, b and c and find the numerical value of x.

```
(%i1) [a,b,c]: [5,3,2];
```

$$(\%o1) \quad [x = -2, x = -1]$$

You may want to check your solution function ev(eq1,sol). The `eq1` is your equation and `sol` is the solution.

The solve function

The function solve(eq, var)

OK, that was easy, now

1. Let us try to solve the cubic equation: $ax^3 + bx^2 + cx + d = 0$ in *WxMaxima* , and
2. The quartic equation $ax^4 + bx^3 + cx^2 + d * x + e = 0$

You may want to use $[a, b, c, d] : [6, -3, 1, -1]$ and $e = 125$

And, get the numerical values using `numer` or `float` function.

You may get a complex result.

The solve function

The function solve([eq1,eq2,...], [var1,var2,...])

Now we attempt to solve set of equations using solve. The `solve` function will now include all equations separated by ',' and enclosed in **[ ]**. The same is done for variables.

Let us look at an example

```
(%i1) eq2: [3*x+2*y=5, 6*x+y=0]
```

$$(\%o1) \quad \left[x=-\frac{5}{9}, x=\frac{10}{3}\right]$$

How about solving: $\sqrt{x}+y=0$ and $\sqrt{x}=1$ for x and y

For that you will have to use another function called: `to_poly_solve`. Check the *WxMaxima* for more.

You may want to learn more on that by yourself.

The solve function

The Engineering Problems- Decay

The radioactive decay model is:

$$DEq : q = q_0 \exp \frac{-t}{\tau}$$

We can find the rate of reaction (τ) by first solving the equation using `solve` function and substituting the values $q_0 = 10g, q = 4g \text{ and } t = 2s$ using another function subst([val1, val2,. . .], sol).

We should get $\tau = 2.18$.

Now if half-life ($t_{1/2}$) is to be found, we can use the relation

$$t_{1/2} = \frac{\ln(2)}{\tau}$$

We obtain $t_{1/2} = 0.317$

Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

The Engineering Problems- The Motion Trajectory

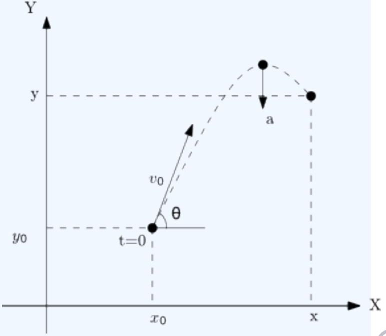
The two-dimensional motion under constant acceleration

Let a ball is moving with constant acceleration in the $y-$ direction only (see Figure). The equations for the position of the ball in the x and y directions at any time t are given by equations EqX and EqY:

EqX: $x = x_0 + v_0 \cos\theta t$
and

EqY: $y = y_0 + v_0 \sin\theta t + \frac{1}{2}at^2$

v_0 (the initial velocity) and a (the acceleration), have to be obtained provided that initial position (x_0, y_0), θ and t are known.



The Motion Trajectory

Using function `solve`, we can solve two simultaneous equation *EqX* and *EqY* for v_0 and a

```
(%i1) SolXY: solve([EqX,EqY], [v0, a])
```

$$(\%o1) \qquad v0 = -\frac{x0 - x}{\cos}(\theta)t\,,$$

$$a = \frac{\cos(\theta)2y - 2y0 + 2\sin(\theta)x0 - 2\sin(\theta)x}{t^2\cos(\theta)}$$

Let us use $x0 = 0, x = 2m, y0 = 2m, y = 8m, \theta = \pi/6$, and $t = 2s$, then substituting them we get $v0$ and a

$$v0 = 1.154m/s \text{ and } a = 2.422m/s^2$$

Check them in *WxMaxima* .

The Motion Trajectory

Few notes on the trajectory problem before we move to another problem:

A) `solve` will not be able to obtain solution for *theta*, because it is not algebraic and can not be linearly isolated.

B) A solution for $x0$ and t is allowed because both terms are algebraic in the equations, even if t is quadratic.

For the case A): a numerical solution using function `find_root` can be used after substituting all the known values in the equations.

You may want to try the above in *WxMaxima* .

Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

SHARDA UNIVERSITY Beyond Boundaries

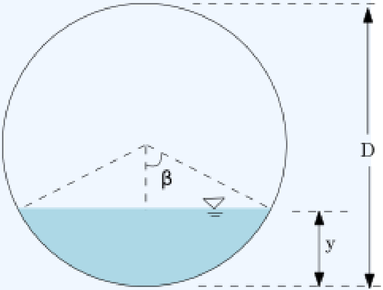
The Engineering Problems- The open channel problem

Let us solve the Manning's equation for a circular open channel.

The cross-section of a circular channel is characterized by its diameter D, and its depth y. These two variables are related by the half-angle β, such that $\cos(\beta) = 1 - 2(y/D)$.

Let us start with Manning's equation:

$$\mathrm{EqM} : v = \frac{kR^{2/3}S^{1/2}}{n}$$



Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

The solve function

The open channel problem

Next, we define the continuity equation, EqQ.

$$\mathrm{EqQ} : Q = Av$$

And, then combine two equations using `subst` function

$$\mathrm{EqMQ} : subst(EqM, EqQ)$$

Next, we substitute the definition of the hydraulic radius:

$$\mathrm{EqMQ} : subst(R = A/P, EqMQ)$$

Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

The solve function

The open channel problem

Next, we substitute the definitions of the area, A, and wetted perimeter, P, for a circular cross-section in terms of the half-angle β to produce equation EqMQC:

$$\mathrm{EqMQC} : subst([A = \frac{D^2}{4}(\beta - \sin(\beta)\cos(\beta)), P = \beta D], EqMQ)$$

Next, we replace the half-angle β in terms of the depth y and diameter D, to produce equation EqMQCy:

$$\mathrm{EqMQCy} : subst(beta = \mathrm{acos}(1 - 2y/D), EqMQC)$$

Finally, we substitute the parameters of the problem as follows, $k = 1.486, D = 5ft, Q = 2.5ft^3/s, S = 0.000023, andn = 0.012,$ to create equation

Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

The find_root function

The open channel problem

$$\mathrm{EqMQCy1} : subst([k = 1.486, D = 5, Q = 2.5, S = 0.000023, n = 0.012], EqMQCy)$$

This is the equation we need to solve for y. The equation is not algebraic (see arccos), and therefore the `solve` function will not work. Hence, a new function: *find_root(exp, var, a, b)* is introduced.

The `exp` = expression,
The `var` = the variable
The `a and b` = the limit within which the root is to be searched.

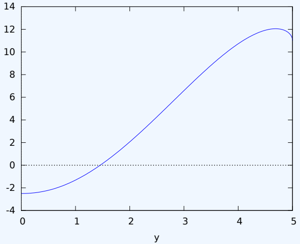
The open channel problem

Question: Which interval contains the root ($f(x) = 0$)?

We need to plot the problem for an easy answer. Let us do that.

For plotting, we use **wxplot2d(fn,[y,0,5])**

The plot shows that `fn=0` will be between 1 and 2. We use this as our interval.



Finally, we find the critical depth or y, using: find_root(fn=0, y, 1,2) and get, $y = 1.45$

Dept. of Civil Engineering
Dept. of Computer Sci. & Engg.
Sharda University

Conclusions

This was a very brief introduction to the solving of equation using **Maxima** .

Maxima can be used to solve varieties of ODE, and some PDE. More detail on it can be found in the **Maxima** manual.

Moreover, the *WxMaxima* provide us with a very handy interface to solve mathematical problems intuitively.

You may want to check following function in the **Maxima** manual: `ode2, linsolve, dsolve` for solving different types of equations.

We realized the importance of graphics when solving a complicated problem. We next focus on plotting.

That was introduction to Solving Eqautions using **Maxima** . Let us get advanced

and learn to visualize maths.

SHARDA UNIVERSITY

Beyond Boundaries