# Computing for mathematics handout 8 - Extracting solutions from outputs of solvers

Lecturer: Vince Knight

Office: M1.30

email: knightva@cf.ac.uk

**Office hours: Thursday 1300-1500**

## What you have learnt this week:

Some basic Sage code to solve differential equations:

- ODEs;  
- Systems of ODEs;  
- Numerical solutions of ODEs (for when they can't be solved exactly).

## Extracting parts of an equation

In [handout 7](http://drvinceknight.github.io/Computing_for_mathematics/Handouts/handout07.html) we saw how to extract solutions to equations from the list output:

sols = solve(x ^ 2 - x - 1 == 0, x, solution\_dict=True)  
[d[x] for d in sols]

Another way to do this is to use .rhs():

sols = solve(x ^ 2 - x - 1 == 0, x)  
[eq.rhs() for eq in sols] # We are getting the right hand side of the solutions which are given in the form of equations: `x = ...`.

**This extends to the solutions of differential equations**.

t = var('t')  
y = function('y', t)  
x = function('x', t)  
sols = desolve\_system([diff(x, t) == 1 - y, diff(y, t) == 1 - x], [y,x])

If we take a look at sols, the output of desolve\_system is a list containing x(t) = ... and y(t) = ....

To extract the solutions we use the rhs() method:

x(t) = sols[0].rhs()  
y(t) = sols[1].rhs()

Now plotting these is straightforward:

p = plot(x, t, 0, 10, legend\_label="$x(t)$")  
p += plot(y, t, 0, 10, color='red', legend\_label="$y(t)$")  
p

## Numerical analysis

## What you should do next:

* **Start the next sheet**: make sure you spend time working on the sheet **BEFORE** the labs.
* **Start the coursework**
* Contribute to the wiki.
* To make the best use of the lab sessions turn up having finished your sheets;
* If anything is still unclear **please** come and see me during office hours.