Risp 39: Teacher Notes

Suggested use: to introduce/consolidate/revise numerical methods, polynomial equations

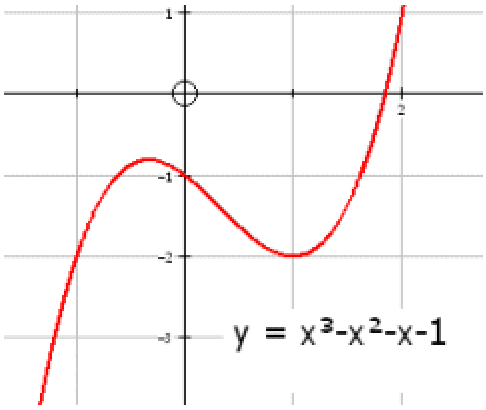
This piece of polynomial work is a nice entry into the world of numerical methods and iterative procedures, although it will work just as well as a review exercise.

I find when I start on numerical methods, to find approximate roots for tricky polynomials, I need a simple set of equations that get gradually more difficult as the degree increases, that do not have trivial solutions, and where the rearrangement methods in particular does not do anything too nasty. This set needs too to be easy to remember as the polynomials are typed into graphing programs. The functions this risp employs fulfil all these criteria.

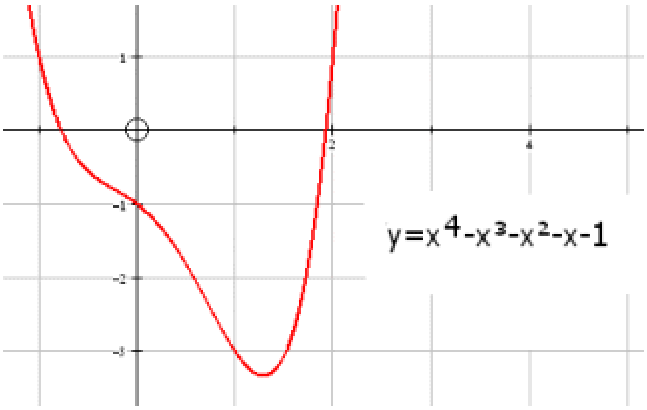
$x - 1 = 0$; we have no problem here.

$x^2 - x - 1 = 0$; more difficult, but we have a formula that gives the exact answers, to this and every quadratic.

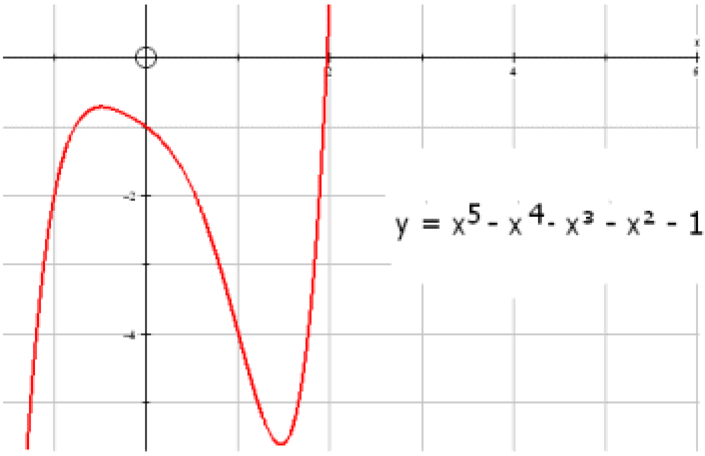
$x^3 - x^2 - x - 1 = 0$; much harder – there is a formula, that will work for all cubics, but it is tough, although not beyond a keen A Level student. The history of this is excellent fare – the Italian mathematicians in the sixteenth century, Ferror, Tartaglia, Cardano, Fiore and Ferrari and their duels make a fine story. (Search for 'solving the cubic' on Wikipedia.)



$x^4 - x^3 - x^2 - x - 1 = 0$; again, a tough formula to give the exact answers to this and all other quartics does exist.



$x^5 - x^4 - x^3 - x^2 - x - 1 = 0$; quintics and above cannot in general be solved exactly using an algebraic formula. The history again is interesting here; Abel, Ruffini and Galois are the main players this time.



Playing with $y = x^n - x^{n-1} - x^{n-2} .. - 1$ on a graphing program, a student will quickly develop the hypothesis that y = 0 has two roots (ever closer to 2 and -1 as n increases) if n is even, and one root (ever closer to 2 as n increases) if n is odd. We can find the roots accurately by homing in on the graph, but using an iterative procedure works well here.

Taking n = 3, we are trying to solve $x^3 = x^2 + x + 1$, or $x = \sqrt[3]{(x^2 + x + 1)}$.

This gives us the iteration $x_n = \sqrt[3]{(x^2_{n-1} + x_{n-1} + 1)}$, which converges to 1.839... when $x_0 = 1$.

This same technique works for n even to find the root near 2.

$x_n = \sqrt[4]{(x^3_{n-1} + x^2_{n-1} + x_{n-1} + 1)}$ gives the root as 1.928...

With n even, can we find the root near -1 this way? The fourth root is always positive (if it exists), so no. However, the iteration above gives the root near 2 when x_0 is negative.

But if we tweak the formula slightly as follows:

$x_n = \sqrt[3]{(x^4_{n-1} - x^2_{n-1} - x_{n-1} - 1)}$, we eventually get –0.7748 after entering $x_0 = –0.5$, so we can find the root near –1.

Students will tire of typing in long equations for the curves. Better is to multiply both sides of:

$y = x_n - x_{n-1} - x_{n-2} - .. - 1$ by $(x - 1)$, saying that when $x = 1$, $y = 2 - n$.

Most things cancel, giving the curve $y = \dfrac{x^{n+1} - 2x^n + 1}{x - 1}$.

Your graphing package should deal with this as long as n is integral, and increasing n with the constant controller illustrates the flip-flop nature of the curves nicely. It also makes it easy to see what happens to the curve as n tends to infinity. (Making n negative is another ball-game!)

For all n, the curve goes through (0, -1) and (2, 1).

If n is even, (-1, 1) is on the curve, if n is odd, (-1, -2) is on the curve.

So as n tends to infinity, and the curve gets steeper close to 2 and -1,
so the roots will tend to 2 (and -1 if n is even.)

The nice thing here is that this simple set of polynomials provides an excellent starting example for iterative techniques, an area where is it easy to embark on equations that behave in ways that we hoped they would not!

www.risps.co.uk