Autumn Quarter 2005
Math. Methods Problem Set 3: Newton's Method and Numerical Integration

October 19, 2005

1   Newton's method

Use Newton's method to find the first 5 positive solutions of $\sin x = 10e^{-x}$.

2   Newton's method as a dynamical system

Consider the iterated-map dynamical system defined by applying Newton's method to an attempt to find the roots of $x^2 + 1 = 0$, with real x. Are there fixed points? Are there periodic orbits? Are these stable? What is the attractor basin of the stable solutions, if any? Are there any chaotic orbits?

   Extra credit: Iterated maps can be defined on the complex plane instead of on the Real axis. The above problem has a pair of roots on the imaginary axis, which are attracting stable fixed points of the associated Newton map (show this). What are the attractor basins of these points in the complex plane? Are there chaotic solutions? Note that to do this problem you will need to either modify `mapExplorer` or write your own iteration routines, since `mapExplorer` is not set up to handle complex numbers. Remember, a complex constant in Python is defined by an expression like `2.5 + 3.2j`. You can make a complex number out of a pair of real numbers `a` and `b` by writing `z = complex(a,b)`. The real and imaginary parts can be accessed as `z.real` and z.imag.

3 Numerical evaluation of definite integrals

Study the convergence of the trapezoidal rule approximation of

$$\int_a^b \sqrt{x}dx \tag{1}$$

to the exact value for $n = 2, 4, 8, 16, ...4096$. Save your results in a list for plotting and other future uses. Is it quadratic in $1/n$ as expected? Now look at how rapidly the result converges if you use Romberg extrapolation (helped by the polynomial interpolation/extrapolation routine `polint`) for the same sequence of n.

First do the above for $a = 1, b = 10$. Then try again for $a = 0$ and compare results. Does Romberg extrapolation improve your convergence as much in this case? What is the reason for the difference between the two cases?

Finally,just for fun, use the trapezoidal rule with Romberg extrapolation to evaluate the integral

$$\int_0^\pi e^{-\cos x}dx \tag{2}$$

If you begin with a single interval ($n = 1$), how many refinement steps do you need to achieve 6 decimal place accuracy?

4 Programming Project: A better trapezoidal rule integrator

The simplest form of the trapezoidal rule integrator, `dumbTrap(f,interval,n)` is wasteful if you've already computed the result for `n = N` and then want to recompute it for `n = 2N`. The recomputation computes the sum over all the points, whereas half the work was already done when you computed the result for `n = N`.

Develop a class called `betterTrap`, which has `dumbTrap` as one of its methods, and has the function being integrated, the interval of integration, and the current values of `n` and `dumbTrap(f,interval,n)` as members. Include a method `refine(...)`, which computes the trapezoidal rule approximation for `2n` in terms of the previous value, without throwing away the part you've already computed. The `refine(...)` method should update the value of `n` and of the estimate.

Note: The `__init__` method should take the initial value of `n` and the function `f` as its argument, and compute the initial value of the trapezoidal rule approximation.