410 Tutorial 3: Cool Newton and Derivatives

September 28, 2017

1 Convergence of Newton's Method

Last week we looked at using Newton's Method to find the eigenvalues of a simple problem in quantum mechanics. You may have noticed that for some starting guesses the method converged to one root, while for other, very close, guesses it converged to a completely separate root (If you want to try this out, create a very deep potential well and set your initial guess intermediate between two roots).

The newton iteration for a given function is given by:

$$x^{n+1} = x^n - \frac{f(x^n)}{f'(x^n)} \tag{1}$$

For any given root, it can be shown that Newton's method will definitely converge when the function is well approximate near the root (and initial guess) as,

$$f(x) \approx xf'(x) + \epsilon x^2 \tag{2}$$

where ϵ is small. If this isn't the case, the iteration may "jump" to a new root, or even diverge away to infinity.

1.1 Julia Sets

From Wikipedia:

> In the context of complex dynamics, a topic of mathematics, the Julia set and the Fatou set are two complementary sets (Julia 'laces' and Fatou 'dusts') defined from a function. Informally, the Fatou set of the function consists of values with the property that all nearby values behave similarly under repeated iteration of the function, and the Julia set consists of values such that an arbitrarily small perturbation can cause drastic changes in the sequence of iterated function values. Thus the behavior of the function on the Fatou set is 'regular', while on the Julia set its behavior is 'chaotic'.

Basically, the Julia sets are the sets of function values for which the iteration,

$$z^{n+1} \to g(z^n) \tag{3}$$

is arbitrarily sensitive to our initial z^0. For $g(z^n) = z^n - \frac{f(z^n)}{f'(z^n)}$, it is easy to see that this is just the Newton iteration extended into the complex plane. Therefore, one might expect that if we try to label the regions of the complex plane which converge to a given root, the boundaries of these regions will result in a Julia set!

Start from the `tutorial_03_roots_template`:

- create vectors `x` and `y` ranging between `-w` and `w`
- use this to create the complex plane `z` with those bounds
- define the functions `f` and `f'` for matrix operations with $f(z) = z^3 - 1$
- define the Newton iteration for the above problem

In the while loop:

- perform a Newton iteration and store the result in `z_new`.
- identify values in the matrix which are greater than the `divergence` variable and set those to `NaN`.
- try to work through the logic in the code
- explore the resulting fractal

2 Numeric Differentiation

2.1 Analytics

Using the Taylor series expansion for $f(x)$,

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2!}h^2 f''(x) + \dots \tag{4}$$

write out first, second and fourth order approximations (in h) to $f'(x)$.

Assuming that every number represented on the computer has error proportional to ϵ, derive expressions for the leading order error expressions (both in terms of ϵ and h) for each derivative approximation derived above (If pressed for time, just do this for the first order approximation).

Use these expression to find an optimal h for performing differentiation with these various schemes.

2.2 Numerics

Using the provided code templates for first, second and fourth order differentiation,

- plot analytic and numeric derivatives of $sin(x)$ for x between 0 and 10 with various values of h. E.g. analytic derivative, $h = 1$, 0.5, 0.25, 0.125.
- In a subplot, plot the error in these approximations.
- do the same for second order derivative approximations. Are the results as expected?

The differentiation templates were written to accept `h` as a vector provided that it has the same dimensions as x. Setting x as a vector of 1's and h as a logarithmically spaced vector ranging between 1 and 1e-15,

- plot the derivatives on a semilogx plot for the analytic and numeric solutions
- plot the errors in the approximations on a loglog plot
- are the minima of each function located in the expected location?