newton’s method and root-finding

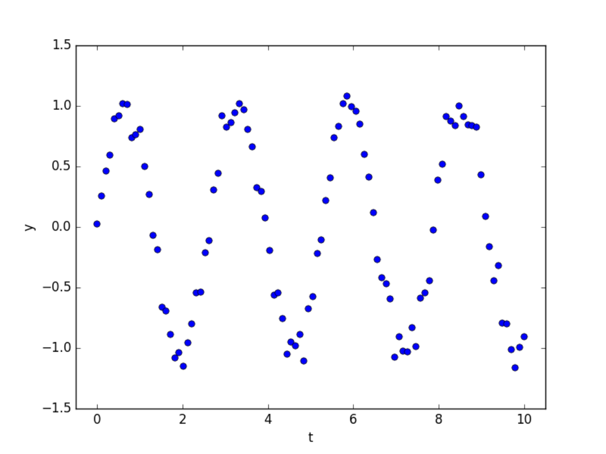
Pete Sentz

Department of Computer Science
University of Illinois at Urbana-Champaign

objectives

- Solve $f(x) = 0$ using Newton's method
- Establish properties of Newton's method
- Apply root-finding to optimization problem
- Solve non-linear least squares using optimization

some data



What are some properties of this data?

properties of data

- $|y_i| \leqslant 1$ (approximately)
- Data is apparently periodic
- $y(0) \approx 0$
- $\implies y_i \approx \sin(kt_i)$
- Why is this different from Tuesday?

linear least squares

Let's take a step back. Suppose the problem were $y_i = k\sin(t_i)$ (unknown coefficient):

$$\begin{bmatrix} \sin(t_1) \\ \sin(t_2) \\ \vdots \\ \sin(t_m) \end{bmatrix} k \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

This is just a $m \times n$ linear least squares problem where $n = 1$. (Same theory applies)

non-linear least squares

But now we have $y_i \approx \sin(kt_i)$ (unknown *basis function*):

$$\begin{bmatrix} ? \\ ? \\ \vdots \\ ? \end{bmatrix} k \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Any ideas?

minimize the residual

$$\min \sum_{i=1}^{m} (y_i - \sin(kt_i))^2$$

Important: the data (x_i, y_i) is fixed (we know it). The residual is a function of k (the unknown).

How do we minimize a function of a single variable?

minimize the residual

$$r(k) = \sum_{i=1}^{m} (y_i - \sin(kt_i))^2$$

Differentiate with respect to k and set equal to zero.

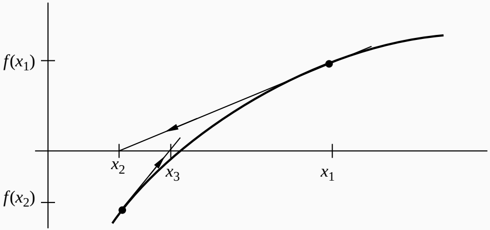
$$0 = r'(k) = -2 \sum_{i=1}^{m} t_i \cos(kt_i)(y_i - \sin(kt_i))$$

Any volunteers?

root-finding

- Would to solve $f(x) = 0$ for general functions
- A value of x that satisfies $f(x) = 0$ is called a root
- Even for polynomials, cannot be done in finite number of steps (Abel/Ruffini/Galois)
- Need iterative method

newton's method



For a current guess x_k, use $f(x_k)$ and the slope $f'(x_k)$ to predict where $f(x)$ crosses the x axis.

newton's method

Use linear approximation of $f(x)$ centered at x_k

$$f(x_k + \Delta x) \approx f(x_k) + f'(x_k)\Delta x$$

Substitute $\Delta x = x_{k+1} - x_k$ to get

$$f(x_{k+1}) \approx f(x_k) + (x_{k+1} - x_k)\, f'(x_k)$$

newton's method

Goal is to find x such that $f(x) = 0$.

Set $f(x_{k+1}) = 0$ and solve for x_{k+1}

$$0 = f(x_k) + (x_{k+1} - x_k)\, f'(x_k)$$

or, solving for x_{k+1}

$$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}}$$

newton's method algorithm

```
initialize:  x_0 = ... # inital guess
for k = 0, 1, 2, ...
    x_{k+1} = x_k - f(x_k)/f'(x_k)
    if converged, stop
end
```

convergence criteria

An automatic root-finding procedure needs to monitor progress toward the root and stop when current guess is close enough to real root.

- Convergence checking will avoid searching to unnecessary accuracy.
- Check how close successive approximations are to each other

$$|x_{k+1} - x_k| < \delta_x$$

- Check how close $f(x)$ is to zero at the current guess.

$$|f(x_{k+1})| < \delta_f$$

newton's method properties

- Highly dependent on initial guess
- Quadratic convergence once it is sufficiently close to the root
- HOWEVER: if $f'(x) = 0$ as well, only has linear convergence
- Is not guaranteed to converge at all, depending on function or initial guess

finding square roots

Newton's method can be used to find square roots. If $x = \sqrt{C}$, then $x^2 - C = 0$. Define as a function:

$$f(x) = x^2 - C = 0$$

First derivative is

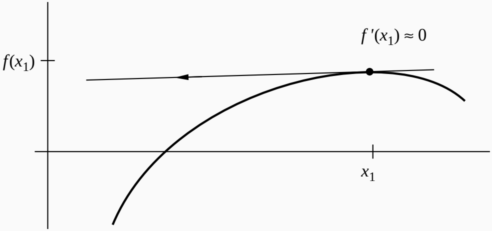
$$f'(x) = 2x$$

The iteration formula is

$$x_{k+1} = x_k - \frac{x_k^2 - C}{2x_k} = \frac{1}{2}\left(x_k + \frac{C}{x_k}\right)$$

Also known as the ”Babylonian Method” for computing square roots.

divergence of newton's method



Since

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

the new guess, x_{k+1}, will be far from the old guess whenever $f'(x_k) \approx 0$

newton's method for optimization

- Minimizing $f(x) \implies f'(x) = 0$
- So now we are searching for zeros of $f'(x)$
- What is Newton's Method for this?

$$x_{k+1} = x_k - \frac{f'(x)}{f''(x)}$$

- If there are many local minima/maxima then $f'(x)$ has many zeros
- Initial guess is *very* important in this case.
- Actual implementation is virtually the same as root-finding.
- Rather than linear approximation, is using quadratic approximation to $f(x)$ (first 3 terms of Taylor Series) and uses minimum as next guess

newton's method for optimization

Can now use Newton's Method to solve non-linear least squares problem from before

$$r(k) = \sum_{i=1}^{m} (y_i - \sin(kt_i))^2$$

$$r'(k) = -2\sum_{i=1}^{m} t_i \cos(kt_i)(y_i - \sin(kt_i))$$

$$r''(k) = 2\sum_{i=1}^{m} t_i^2 \left[(y - \sin(kt_i))\sin(kt_i) + \cos^2(kt_i)\right]$$

(Good thing we have a computer). Iteration:

$$k_{new} = k - \frac{r'(k)}{r''(k)}$$

newton’s method for higher dimensions

- Newton’s Method can be generalized for functions of several variables
- Both root finding and optimization are important in higher dimensions
- Generalizations of first and second derivatives are needed in this case i.e. Jacobian matrix, gradient, and Hessian matrix