Lecture 12

Solving nonlinear equations

Many equations are not solvable analytically. For these, numerical techniques can be helpful.

Relaxation method

Consider

$$x = 2 - e^{-x}$$

This nonlinear equation is not solvable by analytical methods.

However there is a simple numerical method to solve it: just iterate the equation, i.e., guess an initial value $x = x_0$ & plug in

$$x_1 = 2 - e^{-x_0}$$

$$x_2 = 2 - e^{-x_1}$$

$$\vdots$$

$$x_{i+1} = 2 - e^{x_i}$$

If we're lucky, this procedure converges to a fixed point, which is a solution of the equation

Bring up 1-iterate.py for example of this method.

When it works, the method is good & easy to program

Issues w/ this method:

1) Equation must be in the form

$$x = f(x)$$

But sometimes you can get it w/ a rearrangement, e.g.,

$$\log x + x^2 - 1 = 0 \iff x = e^{1-x^2}$$

2) Equation might have more than one solution, but method converges to only one of them. To get around this, change initial value

3) For some functions, it might not converge at all. E.g.,

$$x = e^{1-x^2}$$

can see that solution is $x=1$

Bring up 2-iterate.py

Useful idea is to invert function (if possible). E.g.

$$x = e^{1-x^2} \iff \log x = 1 - x^2$$

$$\iff x = \sqrt{1 - \log x}$$

Bring up 3-iterate.py

Method converges...

Can try to understand when convergence happens

Suppose $f(x)$ has sufficient smoothness & equation is $x = f(x)$ w/ solution for $x = x^*$.

Then consider how relaxation method does when x is close to x^*

Taylor expanding, the value x_1 after an iteration in terms of previous value x_0 is

$$x_1 = f(x_0) = f(x^*) + (x_0 - x^*) f'(x^*) + \dots$$

But we know that $f(x^*) = x^*$ so

$$x_1 - x^* = (x_0 - x^*) f'(x^*) + \dots$$

If higher order terms are small, then we can interpret the above as saying that

the distance between the guess + true solution shrinks or expands by a factor of $|f'(x^*)|$ when $|f'(x^*)| < 1$ or $|f'(x^*)| > 1$, respectively.

So relaxation method converges iff

$$|f'(x^*)| < 1$$

So we can see why this method failed before. We took

$$f(x) = e^{1-x^2} \quad \text{w/} \quad x^* = 1$$

$$\Rightarrow \quad |f'(x^*)| = \left| \left[-2x\, e^{1-x^2} \right]_{x=1} \right|$$

$$= 2$$

$\Rightarrow$ does not converge

On the other hand, if we invert the equation, let's see what happens.

So we have

$$x = f(x) \quad \Rightarrow \quad f^{-1}(x) = x$$

define $v = f^{-1}(x)$ & desired derivative is $\frac{du}{dx}$.

But then we know that

$$x = f(u) \quad \Rightarrow \quad \frac{dx}{du} = f'(u)$$

$$= f'(f^{-1}(x))$$

$$\Rightarrow \frac{du}{dx} = \frac{1}{f'(f^{-1}(x))}$$

But since $f^{-1}(x^*) = x^*$ we have that

$$\frac{df^{-1}(x)}{dx}\bigg|_{x=x^*} = \frac{1}{f'(x^*)}$$

So if $|f'(x^*)| > 1$

this means you can invert everything & be guaranteed convergence.

Unfortunately, we can't invert all equations. So sometimes we cannot guarantee convergence

What about

$$x = x^2 + \sin x$$

solution is @ $x = 0$

but applying relaxation method gives

$$|f'(x^*)| = 2 > 1$$

There is the relation

$$x = \sin^{-1}(x - x^2)$$

not a true inverse! but we can actually use this to converge

Rate of convergence

it is exponentially fast if $|f'(x^*)| < 1$

not very practical:

1) we'd like to know exactly how accurate the answer is

2) we'd like to stop as soon as desired accuracy is reached, e.g., if each iteration takes a long time.

Let ε_i be error on i^{th} estimate, so that

$$x^* = x_i + \varepsilon_i \quad \& \quad x^* = x_{i+1} + \varepsilon_{i+1}$$

Neglecting higher order terms, we have (from Taylor expansions)

$$\varepsilon_{i+1} = \varepsilon_i f'(x^*)$$

$$\& \quad x^* = x_i + \varepsilon_i = x_i + \frac{\varepsilon_{i+1}}{f'(x^*)}$$

Now using

$$x^* = x_{i+1} + \varepsilon_{i+1}$$

we have

$$x_{i+1} + \varepsilon_{i+1} = x_i + \frac{\varepsilon_{i+1}}{f'(x^*)}$$

$$\Rightarrow \varepsilon_{i+1}\left(1 - \frac{1}{f'(x^*)}\right) = x_i - x_{i+1}$$

$$\Rightarrow \varepsilon_{i+1} = \frac{x_i - x_{i+1}}{1 - 1/f'(x^*)}$$

Assuming that $x_i \approx x^*$ so that

$$f'(x_i) \approx f(x^*)$$

we get

$$\varepsilon_{i+1} = \frac{x_i - x_{i+1}}{1 - 1/f'(x_i)}$$

so this is estimated error & we can keep repeating until this error falls below some threshold.

What if we don't know the derivative? Then just estimate it

Consider three successive points

x_i, x_{i+1}, x_{i+2}

From before, we have

$$\varepsilon_{i+2} = \frac{x_{i+1} - x_{i+2}}{1 - 1/f'(x^*)}$$

$$\approx \frac{x_{i+1} - x_{i+2}}{1 - 1/f'(x_i)}$$

Then approximate

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i+1})}{x_i - x_{i+1}}$$

But $x_{i+1} = f(x_i)$ & $x_{i+2} = f(x_{i+1})$

$$\Rightarrow f'(x_i) \approx \frac{x_{i+1} - x_{i+2}}{x_i - x_{i+1}}$$

Substitute back in to get

$$\varepsilon_{i+2} \approx \frac{x_{i+1} - x_{i+2}}{1 - \frac{x_{i+1} - x_{i+2}}{x_i - x_{i+1}}} = \frac{(x_{i+1} - x_{i+2})^2}{2x_{i+1} - x_i - x_{i+2}}$$

So we can estimate error even if we don't know the derivative

Relaxation method for 2 or more variables

Suppose N equations & N variables

Rewrite as $x_1 = f_1(x_1, \ldots, x_N)$

$\vdots$

$x_N = f_N(x_1, \ldots, x_N)$

Then choose starting values & apply repeatedly. May or may not converge

condition for single-variable convergence was $|f'(x^*)| < 1$

for multivariable case, it is that

$|\lambda_i| < 1$ ~~$\forall i \in \{1, \ldots, N\}$~~ where

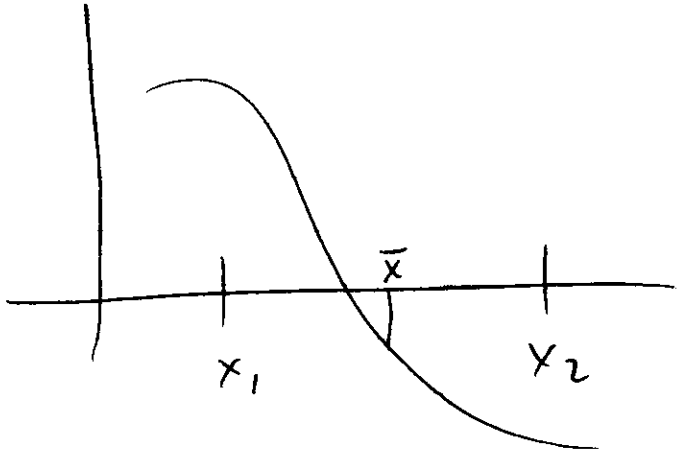
λ_i are eigenvalues of Jacobian matrix w/ entries $\partial f_i / \partial x_j$ evaluated @ fixed point

spectral radius of

Another method for finding solutions is the bisection method or binary search

- specify an interval $[x_1, x_2]$ where we would like a solution
- rearrange equation so that it has the form $f(x) = 0$ & goal is to find roots

Calculate $f(x_1)$ & $f(x_2)$



Suppose $f(x_1) > 0$ & $f(x_2) < 0$.

If $f(x)$ is continuous, then $\exists$ one point between x_1 & x_2 such that f is zero

Also If $\exists$ exactly one root in $[x_1, x_2]$, then $f(x_1)$ & $f(x_2)$ must have

Let $\bar{x} = \frac{x_1 + x_2}{2}$ be midpoint.

Now evaluate $f(\bar{x})$.

Could be that $f(\bar{x}) = 0$,
in which case we're done

If not, then $f(\bar{x})$ is either > 0 or < 0 & so has same sign as either $f(x_1)$ or $f(x_2)$ & opposite sign to the other. If $f(\bar{x})$ has opposite sign to $f(x_1)$ then x_1 & $\bar{x}$ bracket a root. But we have shortened the distance between x_1 & $\bar{x}$ by a factor of two.

Just repeat this process

Algorithm:

1. Given x_1, x_2 check if $f(x_1)$ & $f(x_2)$ have opposite signs. Pick $\varepsilon > 0$.
2. Calculate $\bar{x} = \frac{x_1 + x_2}{2}$ & $f(\bar{x})$
3. If $f(\bar{x})$ has same sign as $f(x_1)$, set $x_1 = \bar{x}$ & else set $x_2 = \bar{x}$.
4. If $|x_1 - x_2| > \varepsilon$ repeat, otherwise set root $= \frac{x_1 + x_2}{2}$

accuracy improves exponentially

Suppose initial distance is $\Delta = x_2 - x_1$

distance is halved at each step, so that after N we get distance

$= \frac{\Delta}{2^N}$ (calculation stops when $\varepsilon = \frac{\Delta}{2^N}$,

implying that $N = \log_2\left(\frac{\Delta}{\varepsilon}\right)$

E.g. suppose $\Delta = 10^{10}$ &

$\varepsilon = 10^{-10}$

$\Rightarrow N = \log_2\left(\frac{10^{10}}{10^{-10}}\right) = \log_2(10^{20})$

≈ 67

Disadvantages of binary search / bisection:

1) doesn't work if $f(x_1)$ & $f(x_2)$ have same sign (however, you might know more about function)

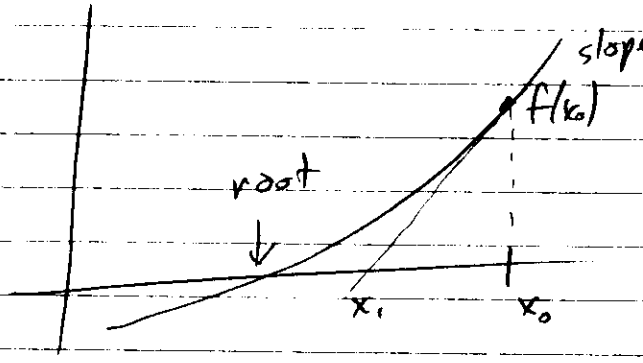
2) cannot find even-order polynomial roots such as for $(1-x)^2$ or $(2-3x)^4$

3) does not extend to multiple dimensions

Newton's method (Newton - Raphson)

again
convert finding a solution to that of finding a root

Start w/ a guess x & then use slope @ x to make next guess



need derivative $f'(x)$ in order to make next guess x_1

$$f'(x_0) \equiv \frac{f(x_0)}{x_0 - x_1}$$

so

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

keep on iterating until convergence occurs

How good is accuracy?

Suppose x^* is true value of root

then Taylor expand about estimate x_i

$$f(x^*) = f(x_i) + (x^* - x_i) f'(x_i) + \frac{1}{2}(x^* - x_i)^2 f''(x_i) + \dots$$

But x^* is such that $f(x^*) = 0$

so LHS vanishes

Divide by $f'(x)$ & rearrange as

$$x^* = \underbrace{\left[x_i - \frac{f(x_i)}{f'(x_i)} \right]}_{\text{this is the new estimate } x_{i+1}} - \frac{1}{2}(x^* - x_i)^2 \frac{f''(x_i)}{f'(x_i)} + \dots$$

So $x^* = x_{i+1} - \frac{1}{2}(x^* - x_i)^2 \frac{f''(x_i)}{f'(x_i)}$

ith error defined by $x^* = x_i + \varepsilon_i$ & i+1 error $x^* = x_{i+1} + \varepsilon_{i+1}$

$$\Rightarrow \quad \varepsilon_{i+1} = -\frac{1}{2}\frac{f''(x_i)}{f'(x_i)}\varepsilon_i^2$$

$\Rightarrow$ Newton's method has quadratic convergence

(converges very quickly

Suppose $-\frac{1}{2}\frac{f''(x)}{f'(x)}$ is constant near ~~root~~

then error after N iterations is

$$\varepsilon \approx \frac{(c\,\varepsilon_0)^{2^N}}{c}$$

doubly exponentially fast convergence

Disadvantages

1) May not know derivative, but can estimate numerically

2) might not converge if $f'(x)$ is very small — could give a larger error