Numerical analysis

Chapter 1: Solving equations

Equation solving methods are important for effectivizing computer calculations.

The bisection method

Bracketing a root

def 1.1: $f(x)$ has a root r if $f(r) = 0$

th. 1.2: if f is cont. in $[a,b]$ and $f(a)f(b) < 0$ a root $r \in [a,b]$ exists

Finding the roots of a function can be made more efficient by repetitively narrowing our search interval (increasing certainty).

This is called bracketing.

Bisection method

if c = middle, $f(c) = 0$: end, c is a root

else narrow until the TOLerance is reached

How accurate & fast?

Error : $|x_c - r| < \frac{b-a}{2^{n+1}}$

No. evals : $n+2$

A solution is correct within p decimals if error $< 0{,}5 \times 10^{-p}$

Fixed point iteration

def 1.4: the real num. r is a fixed point of g if $g(r) = r$

Such values are found by repetively feeding the function the result of the previous iteration, until the value stops changing

Geometry

If we graphically represent these steps we get a cobweb diagram, spiralling around r in a spiral pattern

Linear convergence

The slope of the function determines the spiralling direction: whether it converges or diverges

Convergence means the error grows smaller whilst the opposite holds for divergence

def 1.5: the rate of convergence is

$$\lim_{i \to \infty} \frac{e_{i+1}}{e_i} = S$$ where e_n is error at step n

def 1.7: an iterative method is locally convergent within a neighborhood if all guesses within it converge to r

Stopping criteria

Knowing when to stop with FPI is hard
we need a stopping criterion

Absolute

$$|x_{i+1} - x_i| < TOL$$

Relative

If the solution is too near zero

$$\frac{|x_{i+1} - x_i|}{|x_{i+1}|} < TOL$$

Hybrid

$$\frac{|x_{i+1} - x_i|}{\max(|x_{i+1}|, \theta)}$$

FPI is only locally convergent, and even then linearly

Unlike bisection method, errors grow/shrink at different rates, so FPI may either be slower or faster than it depending on circumstance

Limits of accuracy

Some calculations cannot be made accurate on a computer, to the same degree as others.

Forward and backward error

def. 1.8: given $x_a \approx r$ where r is a root of f

the backward error is $|f(x_a)|$

and the forward error is $|r - x_a|$

Backward error

How much the function f would need to change to nullify the error

Forward error

How much the solution x_a would need to change to nullify the error

Root multiplicity

A root is a multiple root of multiplicity m if $0 = f(r) = f'(r) = \dots = f^{(m-1)}(r)$ but $f^{(m)}(r) \neq 0$

A root is simple if $m = 1$

Wilhinson polynomial

Hard to determine roots due to cancellation of nearly equal large numbers

Sensitivity of root finding

A problem is sensitive if small errors in the input lead to large errors in the output.

sensitivity factor $= \Delta r = \dfrac{\epsilon g(r)}{f'(r)}$

where r is root for f
and $r + \Delta r$ is root for $f + \epsilon g$

error magnification factor $= \dfrac{\text{relative forward err.}}{\text{relative backward err.}}$

Newton's method

This method takes into account the differential at the current estimate, and adjusts it by an inverse factor. It is a form of FPI

If the function is slowing down, eg. nearing a root, we hasten the movement.

Therefore, this method is even faster than previous ones

x_0 = initial guess

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

Quadratic convergence

def 1.10: a method is quadratically convergent if $M = \lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} < \infty$

Root finding w/o derivatives

Secant method

Converges almost as quickly as Newton's method

Replaces tangent w/ secant or (in variants) an approx. parabola

Original

Replace $f'(x_i)$ w/ $\dfrac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$

in Newton's method

x_0, x_1 init. guesses

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

for $i \in \mathbb{N}^+$

The convergence is superlinear e.g. between linear and quad. convergent.

Regula Falsi

Like bisection, but midpoint
is calc. with a secant-like approx

The new point is guaranteed to
lie in [a, b] unlike secant method

Muller's method

Use a parabola by using three
prev. guesses

Check for intersections

Inverse quad. interpolation (IQI)

Like Muller's, but parabola is
of form $x = p(y)$ (laying down)
so it only intersects at one point.

Three init guesses

Converges faster than og. secant

Brent's method

Combines IQI, secant method and bisection method

Uses backwards error as metric

1. apply IQI
2. bracketing interval should be halved
   if not, apply secant
3. if it still fails, apply bisection