Review of methods for Solving Nonlinear Equations

A. Donev, Spring 2021

Solving $f(x) = 0$, $x \in [a, b]$

for a continuous or continuously differentiable function is "easy" for <u>simple roots</u>, $|f'(x)| \neq 0$.

Two classical methods:

1. Bisection
2. Approximate $f(x)$ by a polynomial, find root of polynomial, and repeat (secant, Newton, Muller, Brent)

Questions to ask:

- Will the method converge almost always if I have a good guess x_0 for root x,

$$x_k \to x \quad \text{s.t. } f(x)=0$$

If $|x_0 - x| < \delta$, we want

$$e_k = |x^k - x| < e_{k-1}$$

- Once x^k is close to to root x, how fast does the method converge, i.e. what is the order of convergence

a) $$\frac{|e_{k+1}|}{|e_k|} \to C < 1$$

linear convergence

$$\Rightarrow \quad e_k = C^k e_0 \to 0$$

6) $$\frac{|e_{k+1}|}{|e_k|^p} \to C, \quad p > 1$$

$p = 2$ quadratic convergence

$1 < p \leq 2$ sublinear convergence

Need to know answer to these two questions for bisection, Secant, Newton, and also be able to analyze (with hints) convergence of a new method, or a known method on specific example ($f(x)$ and root x).

Bisection:

~~Only~~ works in 1D.

Given an $[a,b]$ s.t.

$$f(a)\cdot f(b) < 0$$

it is guaranteed to converge linearly with constant $C = \frac{1}{2}$, specifically

$$e_k = |x - x_k| \leq \frac{b-a}{2^{k+1}}$$

(we get extra $\frac{1}{2}$ by always outputting the midpoint of bisection interval at the end)

Converges slowly but surely if function changes sign on $[a,b]$

Algorithm: Bisection

$a_0 = a \quad , \quad b_0 = b$

For $k = 0, 1, \ldots, n-1$

$x_k = \dfrac{a_k + b_k}{2}$

If $f(x_k) f(a_k) < 0$

$a_{k+1} = a_k \; ; \; b_{k+1} = x_k$

else

$b_{k+1} = b_k \; ; \; a_{k+1} = x_k$

end

end

Return $x \in [a_n, b_n]$

$x \approx \dfrac{a_n + b_n}{2}$ (extra $\frac{1}{2}$)

Methods based on function approximation

Generic algorithm:

Start with interval $[a_0, b_0]$

Repeat:

a) Approximate $f(x)$ by a linear, quadratic, cubic etc. polynomial on $[a_k, b_k]$ and find roots in $[a_k, b_k]$, if any, choose one if multiple.

b) If no roots use bisection to update $[a_{k+1}, b_{k+1}]$ (safeguarded) & cycle back.

c) Update interval $[a_{k+1}, b_{k+1}]$ to contain root estimate,

$$[a_{k+1}, b_{k+1}] \subseteq [a_k, b_k]$$

d) Terminate if either

$$|f(x_k)| < \varepsilon_f \quad \left(\text{good if } f'(x_k) \text{ large}\right)$$

or

$$|x_{k+1} - x_k| < \varepsilon_x \quad \left(\text{good if convergence is rapid}\right)$$

Two specific algorithms in this class we studied are

<u>Secant</u> and <u>Newton</u>

(linear approximation),

but homework Muller / Brent

(quadratic approximation via interpolation)

Newton's method

Approximate

$$f(x) \simeq f_k(x) = f(x_k) + f'(x_k)(x - x_k)$$

$$f_k(x_{k+1}) = 0 \implies$$

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

Choose either $[a, x_{k+1}]$, $[x_{k+1}, b]$, $\left[a, \frac{a+b}{2}\right]$ or $\left[\frac{a+b}{2}, b\right]$ as next interval

Know Taylor series with remainder for analysis!

If method converges, it converges quadratically

$$\frac{|e_{k+1}|}{e_k^2} \to \left|\frac{f''(x)}{2f'(x)}\right| = C$$

if (approximately)

$$e_0 \leq \left|\frac{f'(x)}{f''(x)}\right|$$

If initial guess is not good enough, it may diverge or converge slowly / erratically.

Clearly doesn't work if $f'(x)=0$ (midterm question)

since $C \to \infty$ and we need $e_0 \leq 0$.

Secant method

Use linear interpolation to approximate $f(x)$ using x_k and x_{k-1} as nodes.

[Really doing extrapolation since we consider linear approximation outside of interval]

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

If x_0 and x_1 sufficiently close to root it will converge superlinearly

$$\frac{|e_{k+1}|}{|e_k|^q} \to C, \quad q = \frac{1}{2}(1+\sqrt{5})$$

(homework)

How to analyze a generic method for order of convergence?

Write method as fixed-point iteration

$$x_{k+1} = g(x_k)$$

s.t. $f(x) = 0 \iff x = g(x)$

If it converges it will converge at least linearly,

$$\frac{|e_{k+1}|}{|e_k|} \to |g'(x)|$$

So we want $|g'(x)| << 1$.

If $g'(x) = 0 \Rightarrow$ superlinear convergence

Newton's method in high-dims

$$\vec{f}(\vec{x}) = 0$$

Approximate by linear function

$$f(x) \simeq f(x_k) + \text{Linear Mapping}(\Delta x)$$

$$\Delta x = x - x_k$$

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

& solve linear system for Δx.

If $\vec{x}$ and $\vec{\Delta x}$ are vectors, then

$$\text{Linear Mapping}(\vec{\Delta x}) = \overleftrightarrow{J}\,\vec{\Delta x}$$

where $\overleftrightarrow{J}$ is some matrix.

Different methods to estimate $\overleftrightarrow{J}$ but for Newton's method use multivariate Taylor series:

$$J_{ij} = \frac{\partial f_i}{\partial x_j} \leftarrow \text{Jacobian matrix}$$

Newton's method:

Solve $\begin{cases} J(x_k)\,\Delta x_k = -f(x_k) \\ x_{k+1} = x_k + \Delta x_k \end{cases}$

If initial guess is sufficiently close, it will converge quadratically.

Note:
Never use matrix inverse to solve linear systems on computer

$$x_{k+1} = x_k - \cancel{J_k^{-1}}\, f_k$$