Systems of Nonlinear Equations

Paul Heckbert

Computer Science Department
Carnegie Mellon University
Nonlinear Systems

One dimensional:

\[ f : \mathbb{R} \rightarrow \mathbb{R} \]

-e.g. \( f(x) = ax - b = 0 \) (linear)

or \( f(x) = \sin(2x) - x = 0 \) (nonlinear)

Multidimensional

\[ f : \mathbb{R}^n \rightarrow \mathbb{R}^n \]

-e.g. \( f(x) = Ax - b = 0 \) (linear system)

or \{\( x^2 + y^2 + xy - 1 = 0, \ x^2 + y^2 - xy - 1 = 0 \}\) (nonlinear: two ellipses)

also known as root finding
Four 1-D Root-Finding Methods

- Bisection Method
- Newton’s Method
- Secant Method
- Linear Fractional Interpolation
Bisection Method (Binary Search)

Given $f()$, tol, and $[a,b]$ such that $\text{sign}(f(a)) \neq \text{sign}(f(b))$:

while $b-a>\text{tol}$

\[ m = \frac{a+b}{2} \quad \text{midpoint} \]

if $\text{sign}(f(a)) = \text{sign}(f(m))$: then

\[ a = m \quad \text{recurse on right half} \]

else

\[ b = m \quad \text{recurse on left half} \]

Guaranteed convergence! (if you can find initial $a,b$)

but only at a linear rate: $|\text{error}_{k+1}| \leq c|\text{error}_k|$, $c<1$
Newton’s Method

Given \( f() \), \( f'() \), tol, and initial guess \( x_0 \)

\[
k = 0 \\
do \\
\quad x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \\
\quad k++ \\
while x_k - x_{k-1} > tol
\]

Can diverge (especially if \( f'' \approx 0 \)).
If it converges, does so at quadratic rate: \(|e_{k+1}| \leq c|e_k|^2\).
Requires derivative computation.
Secant Method

Like Newton, but approximates slope using point pairs

Given $f()$, tol, and initial guesses $x_0$, $x_1$

$k = 1$
do

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

$k++$

while $x_k - x_{k-1} < \text{tol}$

Can diverge.

If it converges, does so at rate 1.618: $|e_{k+1}| \leq c|e_k|^{1.618}$. 
Linear Fractional Interpolation

Instead of linear approximation, fit with fractional linear approximation: \( f(x) \approx \frac{x-u}{vx-w} \) for some \( u,v,w \)

Given \( f() \), tol, and initial guesses \( a, b, c, f_a=f(a), f_b=f(b) \)
do

\[
h = \frac{(a-c)(b-c)(f_a-f_b)f_c}{[(a-c)(f_c-f_b)f_a-(b-c)(f_c-f_a)f_b]}
\]

\[
a = b; \quad f_a = f_b
\]

\[
b = c; \quad f_b = f_c
\]

\[
c += h; \quad f_c = f(c)
\]
while \( h > \text{tol} \)
If it converges, does so at rate 1.839: \(|e_{k+1}| \leq c|e_k|^{1.839}\).
1-D Root-Finding Summary

- Bisection: safe: never diverges, but slow.
- Newton’s method: risky but fast!
- Secant method & linear fractional interpolation: less risky, mid-speed.

- Hybrids: use a safe method where function poorly behaved, Newton’s method where it’s well-behaved.
Multidimensional Newton’s Method

\( f \) and \( x \) are \( n \)-vectors

First order Taylor series approx.: \( f(x+h) \approx f(x) + f'(x)h \)

\[
\begin{align*}
\begin{bmatrix}
  f_1 \\
  \vdots \\
  f_n 
\end{bmatrix}
  &=
  \begin{bmatrix}
    x_1 \\
    \vdots \\
    x_n 
  \end{bmatrix} \\
  f(x+h) &= 0 \\[2pt]
  \text{implies } h &= -J^{-1}f \\
  \text{so iteration should be } \\
  x_{k+1} &= x_k - J^{-1}(x_k)f(x_k)
\end{align*}
\]

Method requires calculation of Jacobian – can be expensive
Example: Intersection of Ellipses

2 equations in 2 unknowns:

\[ x^2 + y^2 + xy - 1 = 0 \]

\[ x^2 + y^2 - xy - 1 = 0 \]