Review: pros and cons of different methods

Monday, October 6, 2025 11:11 AM

① For root finding + fixed points $f(x)=0$, or $x=g(x)$

Univariate case

BISECTION

problem it solves: rootfinding

requirements: f, a continuous function

bracket $[a,b]$, $f(a)\cdot f(b) < 0$

convergence: linear, with rate $1/2$

i.e. error $e_k = (\frac{1}{2})^k \cdot e_0$

pros:
+ simple
+ if requirements met, it's guaranteed to work

cons:
- needs more than a starting guess, needs a bracket
- kind of slow convergence

FIXED PT. ITERATION

problem it solves: finds a fixed pt., $x=g(x)$

requirements: g, a continuous function

x_0, a starting point

convergence: converges if you can establish a domain D for which ① $x\in D \Rightarrow g(x)\in D$

② g is contractive on D

If p is the fixed pt., and g is differentiable,

convergence is $\begin{cases} \text{linear, w/ rate } |g'(p)| & \text{if } 0<|g'(p)|<1 \\ \text{superlinear} & \text{if } g'(p)=0 \end{cases}$

(and won't converge if $|g'(p)|>1$)

pros:
+ no derivative g' nor bracket $[a,b]$ needed
+ it's a "meta" algorithm: Newton is a special case

cons:
- need not converge...
- can be slow

p. 2, pros and cons

Tuesday, October 7, 2025 12:52 PM

NEWTON'S METHOD

problem it solves: rootfinding, $f(x)=0$ } recasts as fixed pt. problem, $g(x) = x - f(x)/f'(x)$

requirements: f and f'

x_0, initial guess

convergence: ① if f, f' and f'' are continuous, and x_0 is sufficiently close to a simple root p, it will converge to p

i.e. $f'(p) \neq 0$

② if additionally $|g''(x)|$ is bounded on some interval around p,

$g(x) = x - f(x)/f'(x)$

then convergence is quadratic (very fast!)

pros:
- \+ No bracket needed
- \+ Quadratic convergence is super fast

cons:
- − need derivative f'
- − fails if $f'(x_k) = 0$ at any iteration
- − needs x_0 to be "close enough" to the root
- − doesn't have quadratic convergence in all cases (ex: $f(x) = x^2$, root is $p=0$, $f'(p)=0$ so not a simple root)
- − can diverge
- − practical (robust) implementations are more likely to converge but also more complicated.

SECANT METHOD

Similar to Newton but...

- − not quadratic convergence ($\alpha = 2$) but still superlinear w/ order of convergence $\alpha = \frac{1+\sqrt{5}}{2} \approx 1.62$
- − supply two initial points, x_0 and x_1
- \+ don't need to use f'

p. 3, pros and cons

Tuesday, October 7, 2025 4:54 PM

Multivariate Case

BISECTION – not applicable

FIXED PT. ITERATION

- Similar to univariate case. $\vec{x} \leftarrow G(\vec{x})$

pro + no matrix inversion / solving linear equations needed

NEWTON'S METHOD

- Same pros/cons as univariate case. $\vec{x} \leftarrow \vec{x} - J_F(\vec{x})^{-1} \cdot F(\vec{x})$

con – matrix inversion costs $O(n^3)$, bad in high dimension

QUASI-NEWTON METHODS (ex. Broyden's)

- generalizes secant method, similar pros/cons

pro + avoids matrix inverse via tricks like Sherman-Morrison, so much cheaper than Newton in high dimension

Multivariate Optimization

STEEPEST DESCENT / GRADIENT DESCENT

* problem it solves: $\min_{\vec{x} \in \mathbb{R}^n} g(\vec{x})$, $g : \mathbb{R}^n \to \mathbb{R}$

requirements: $\vec{x}_0$ starting point
g and ∇g
learning rate / stepsize η

$\vec{x} \leftarrow \vec{x} - \eta \cdot \nabla g(\vec{x})$

convergence: think of as fixed-point iteration on $G(\vec{x}) = \vec{x} - \eta \cdot \nabla g(\vec{x})$, so similar

can diverge

if it converges, does so linearly or sublinearly

More likely (but not guaranteed) to converge to a "good" critical point

* it really solves find $\vec{x}$ s.t. $\nabla g(\vec{x}) = \vec{0}$ which gives the stationary points / critical points to the optimization problem

pros:
+ simple
+ scales well to large dimension
+ avoids local max, somewhat avoids saddle pts.

cons:
– may diverge
– may not return a global minimum
– convergence can be slow
– must choose a stepsize, or use a linesearch

no method can without further assumptions

p. 4, pros and cons

Tuesday, October 7, 2025 5:09 PM

NEWTON'S METHOD FOR OPTIMIZATION

same caveat as before

*problem it solves: $\min_{\vec{x} \in \mathbb{R}^n} g(\vec{x})$ $: \mathbb{R}^n \to \mathbb{R}$

requirements: Starting point $\vec{x}_0$
g and ∇g and $\nabla^2 g$

convergence: think of as Newton's method (for root-finding) applied to ∇g, so $\vec{x} \leftarrow \vec{x} - J_{\nabla g}(\vec{x})^{-1} \nabla g(\vec{x})$

i.e. $\vec{x} \leftarrow \vec{x} - \nabla^2 g(\vec{x})^{-1} \cdot \nabla g(\vec{x})$

need not converge, but when it does, usually fast

pros: + usually fast convergence

cons:
- unless you add a linesearch, it can easily converge to a local max or saddle pt.
- need not converge
- fails if $\nabla^2 g(\vec{x}_k)$ becomes singular
- needs $\nabla^2 g$ (Autodiff is partially helpful)
- needs to invert/solve $n \times n$ system, so $O(n^3)$

GAUSS-NEWTON

problem it solves: nonlinear least squares, $\min_{\vec{x} \in \mathbb{R}^n} g(\vec{x})$ where $g(\vec{x}) = \sum_{i=1}^{m} \frac{1}{2} f_i(\vec{x})^2$

requirements: starting pt. $\vec{x}_0$
F, and Jacobian J_F $\left(F(\vec{x}) = \begin{bmatrix} f_1(\vec{x}) \\ \vdots \\ f_m(\vec{x}) \end{bmatrix} \right)$, i.e. $J_F = \begin{bmatrix} -\nabla f_1^T - \\ \vdots \\ -\nabla f_m^T - \end{bmatrix}$

convergence: we didn't discuss, but it need not work. When it does, usually faster than gradient descent but slower than Newton

$\vec{x} \leftarrow \vec{x} - (J^T J)^{-1} J^T F$

pros:
+ allows $m \neq n$
+ fast
+ unlike Newton applied to g, it doesn't need $\nabla^2 g$

cons:
- must invert/solve $n \times n$ system, so $O(n^3)$
- may not converge (robust version Levenberg-Marquardt is better)