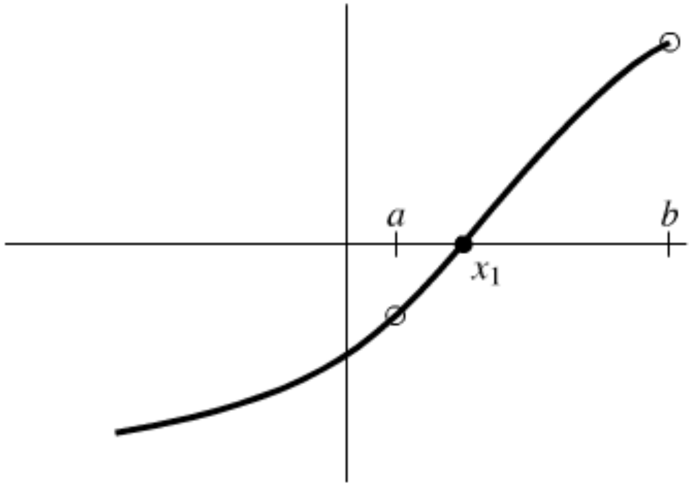
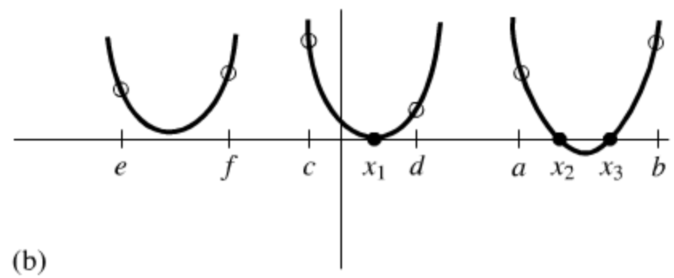
LECTURE 7: Nonlinear Equations and 1-D Optimization

- The topic is related to optimization (Lecture 8) so we cover solving nonlinear equations and 1-d optimization here
- $f(x) = 0$ (either in 1-d or many dimensions)
- In 1-d we can bracket the root and then find it, in many dims we cannot
- Bracketing in 1-d: if $f(x) < 0$ at a and $f(x) > 0$ at $b > a$ (or the other way around) and f(x) is continuous then there is a root at $a < x < b$

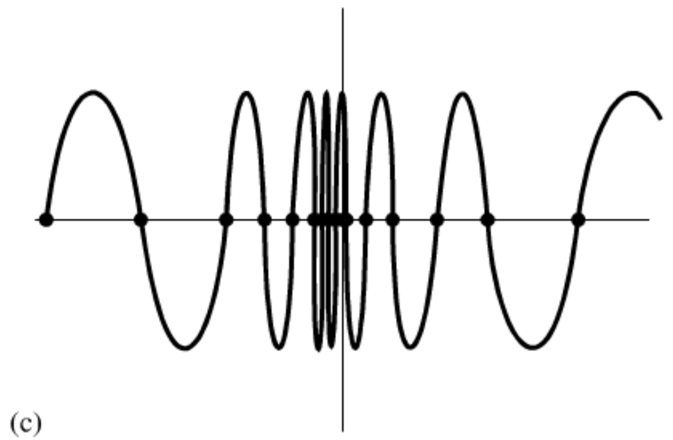


Other Situations:

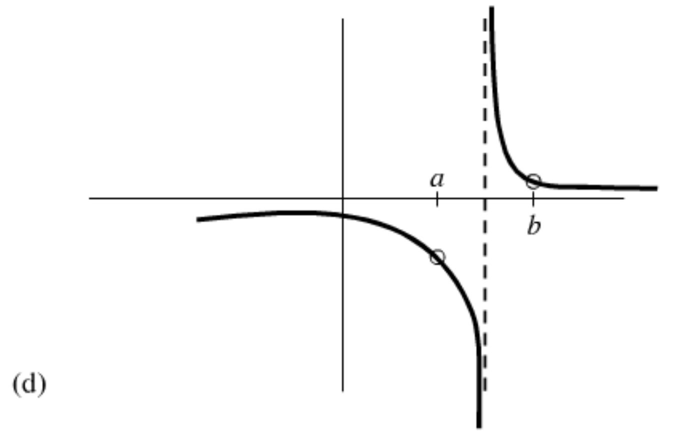
- No roots or one or two roots but no sign change:



- Many roots:



- Singularity:

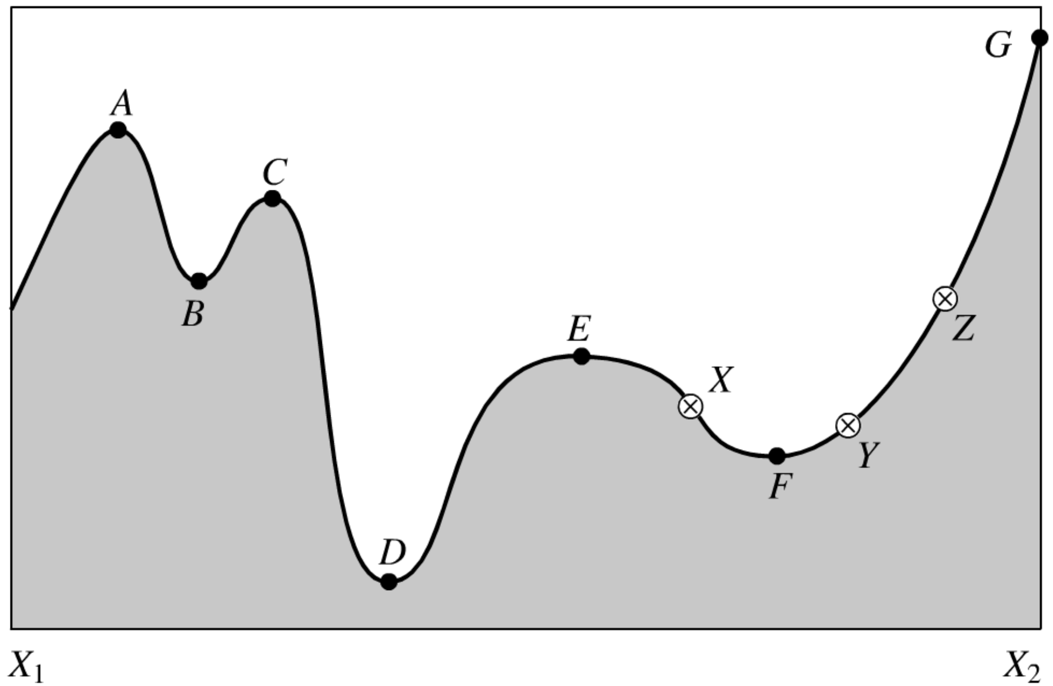


Bisection for Bracketing

- We can use bisection: divide interval by 2, evaluate at the new position, and choose left or right half-interval depending on where the function has opposite sign. Number of steps is $log_2[(b\text{-}a)/\varepsilon]$, where e is the error tolerance. The method must succeed.
- Error at next step is $\varepsilon_{n+1} = \varepsilon_n/2$, so converges linearly
- Higher order methods scale as $\varepsilon_{n+1} = c\varepsilon_n^{\,m}$, with $m > 1$

1-d Optimization: Local and Global Extrema, Bracketing

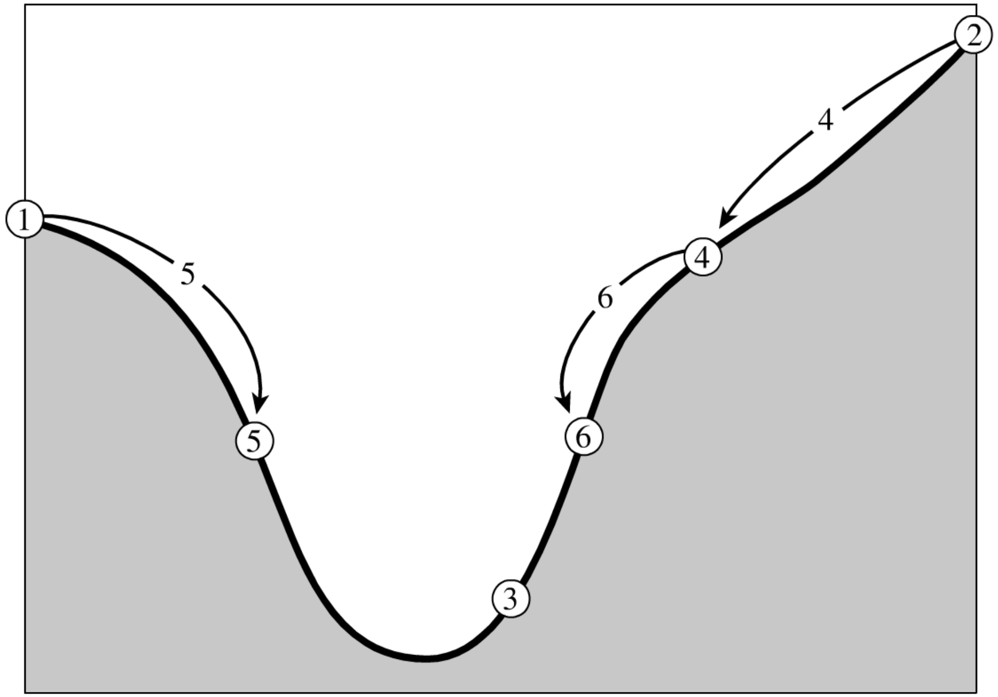
- Optimization: minimization or maximization
- In most cases only local minimum (B,D,F) or local maximum (A,C,E,G) can be found, difficult to prove they are global minimum (D) or global maximum (G)
- We bracket a local minimum if we find $f(X) > f(Y)$ and $f(Z) > f(Y)$ for $X < Y < Z$.



Golden Ratio Search

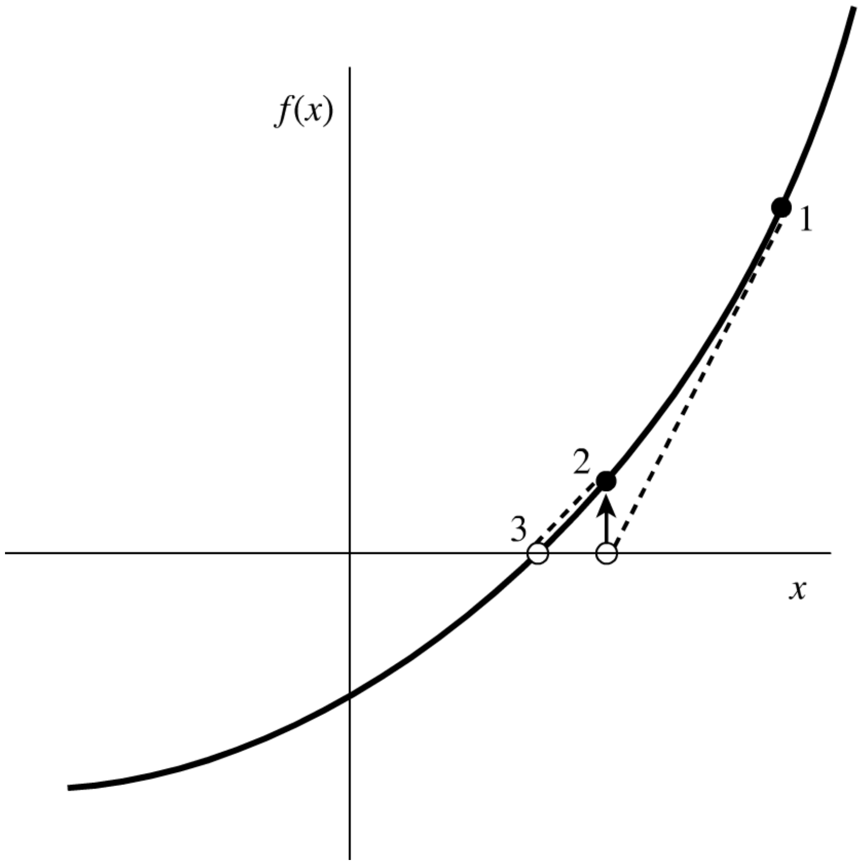
- Remember that we need a triplet of points to bracket $a < b < c$ such that $f(b)$ is less than $f(a)$ and $f(c)$
- Suppose w = $(b-a)/(c-a)$. We evaluate at x, define $(x-b)/(c-a) = z$. The next bracketing segment will be either w+z or 1-w.

  To minimize the error choose these two to be equal: $z = 1 - 2w$.
  But w was also chosen this way, so $z/(1-w) = w$,
  and $w = (3-5^{1/2})/2 = 0.382$,
  $1 - w = 0.618$,
  Golden Ratio
  $(1/0.618=1.618=(1+5^{1/2})/2)$.

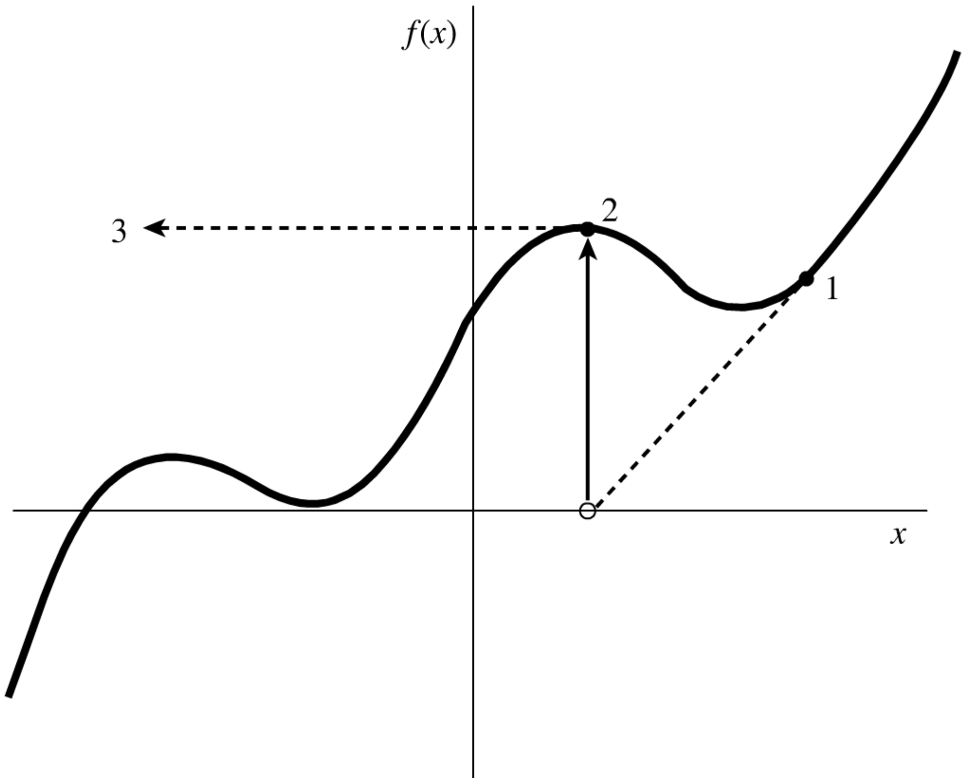


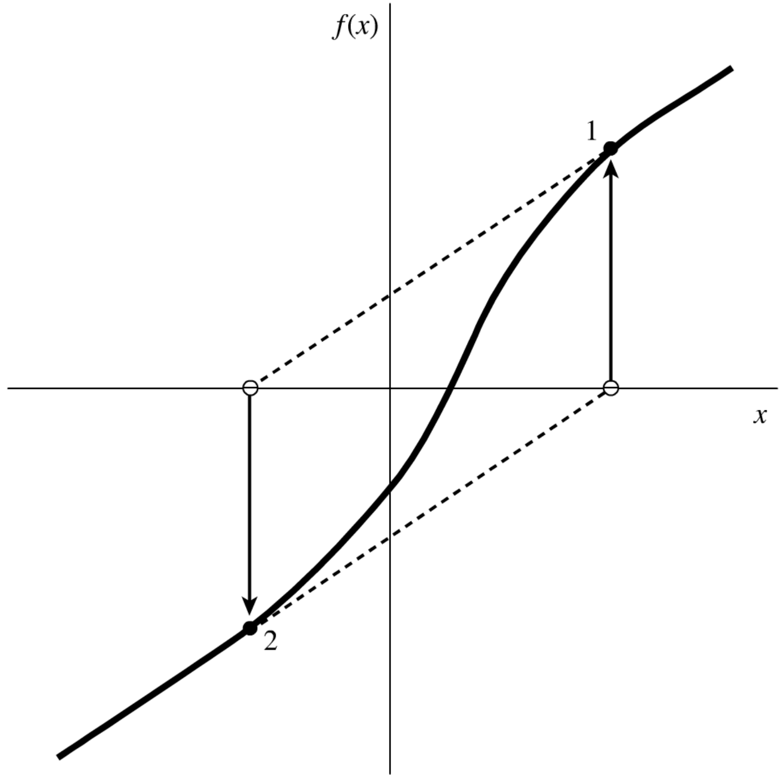
Newton(-Raphson) Method

- Most celebrated of all methods, we will use it extensively in higher dimensions
- Requires a gradient:
  $f(x+\delta) = f(x) + \delta f'(x) + \ldots$
- We want $f(x+\delta) = 0$, hence
  $\delta = -f(x)/f'(x)$
- Rate of convergence is quadratic (NR 9.4)
  $\varepsilon_{i+1} = \varepsilon_i^2 f''(x)/(2f'(x))$



Newton-Raphson is not Failure-free



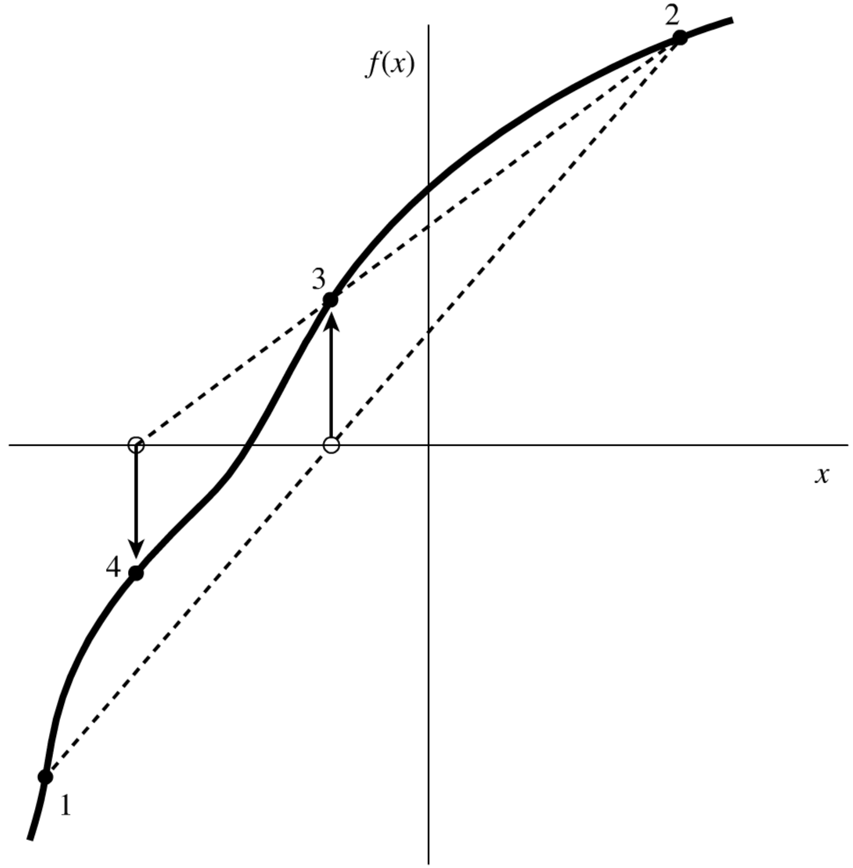


Newton-Raphson for 1-d Optimization

- Expand function to 2nd order (note: we did this already when expanding log likelihood)
- $f(x+\delta) = f(x) + \delta f'(x) + \delta^2 f''(x)/2 + \ldots$
- Expand its derivative $f'(x+\delta) = f'(x) + \delta f''(x) + \ldots$
- Extremum requires $f'(x+\delta) = 0$ hence $\delta = -f'(x)/f''(x)$
- This requires f'': Newton's optimization method
- In least square problems we sometimes only need f'^2: Gauss-Newton method (next lecture)

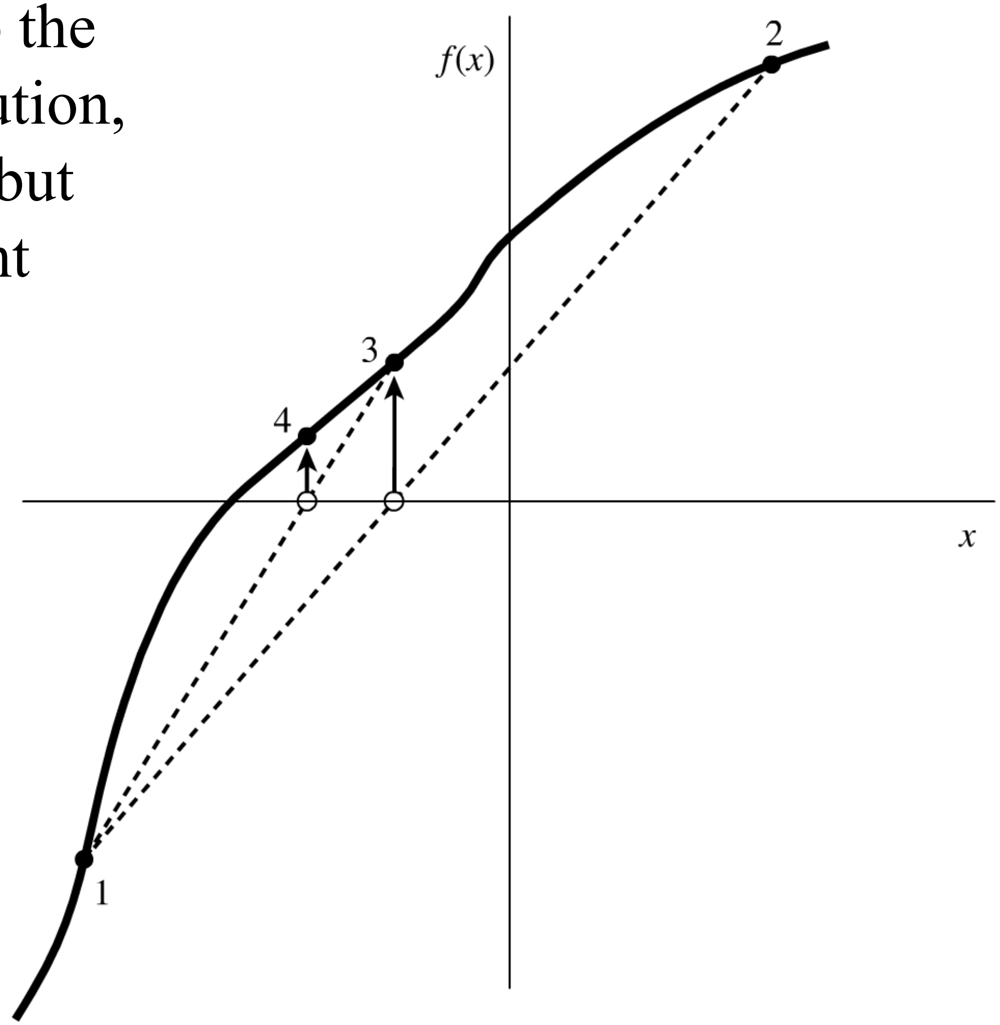
Secant Method for Nonlinear Equations

- Newton's method using numerical evaluation of a gradient defined across the entire interval:
  $f'(x_2) = [f(x_2)-f(x_1)]/(x_2-x_1)$
- $x_3 = x_2 - f(x_2)/f'(x_2)$
- Can fail, since does not always bracket
- m = 1.618 (golden ratio), a lot faster than bisection

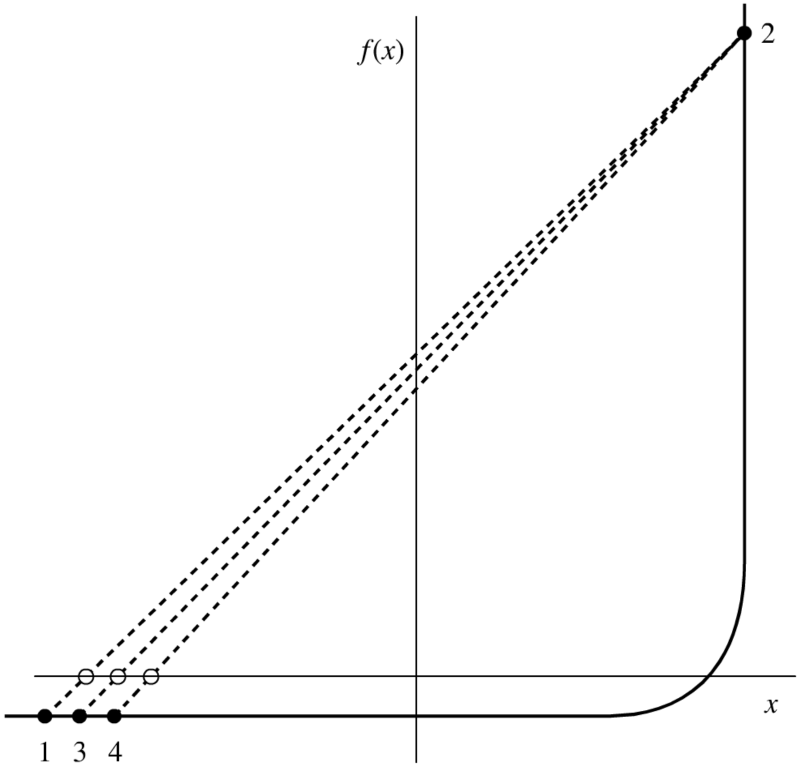


False Position Method for Nonlinear Equations

- Similar to secant, but keep the points that bracket the solution, so guaranteed to succeed, but with more steps than secant



Sometimes convergence can be slow

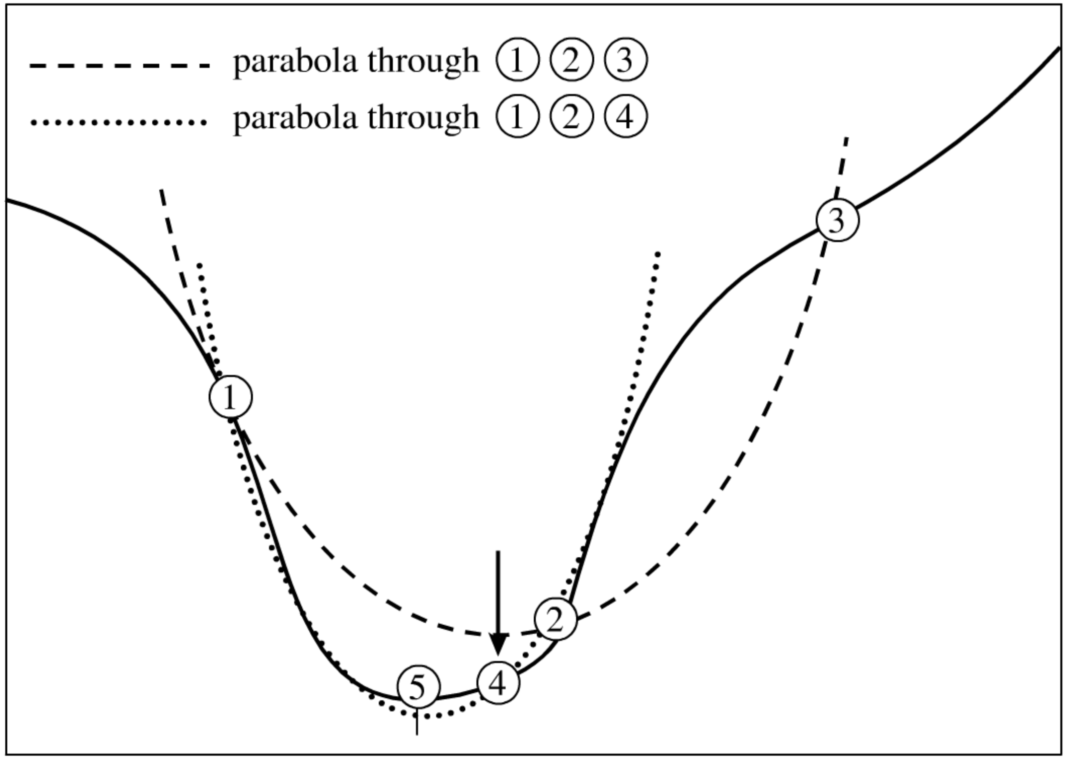


Better methods without derivatives such as Ridders or Brent's method combine these basic techniques: use these as default option and (optionally) switch to Newton once the solution is guaranteed for a higher convergence rate

Parabolic Method for 1-d Optimization

- Approximate the function of *a, b, c* as a parabola

$$x = b - \frac{1}{2} \frac{(b-a)^2[f(b) - f(c)] - (b-c)^2[f(b) - f(a)]}{(b-a)[f(b) - f(c)] - (b-c)[f(b) - f(a)]}$$

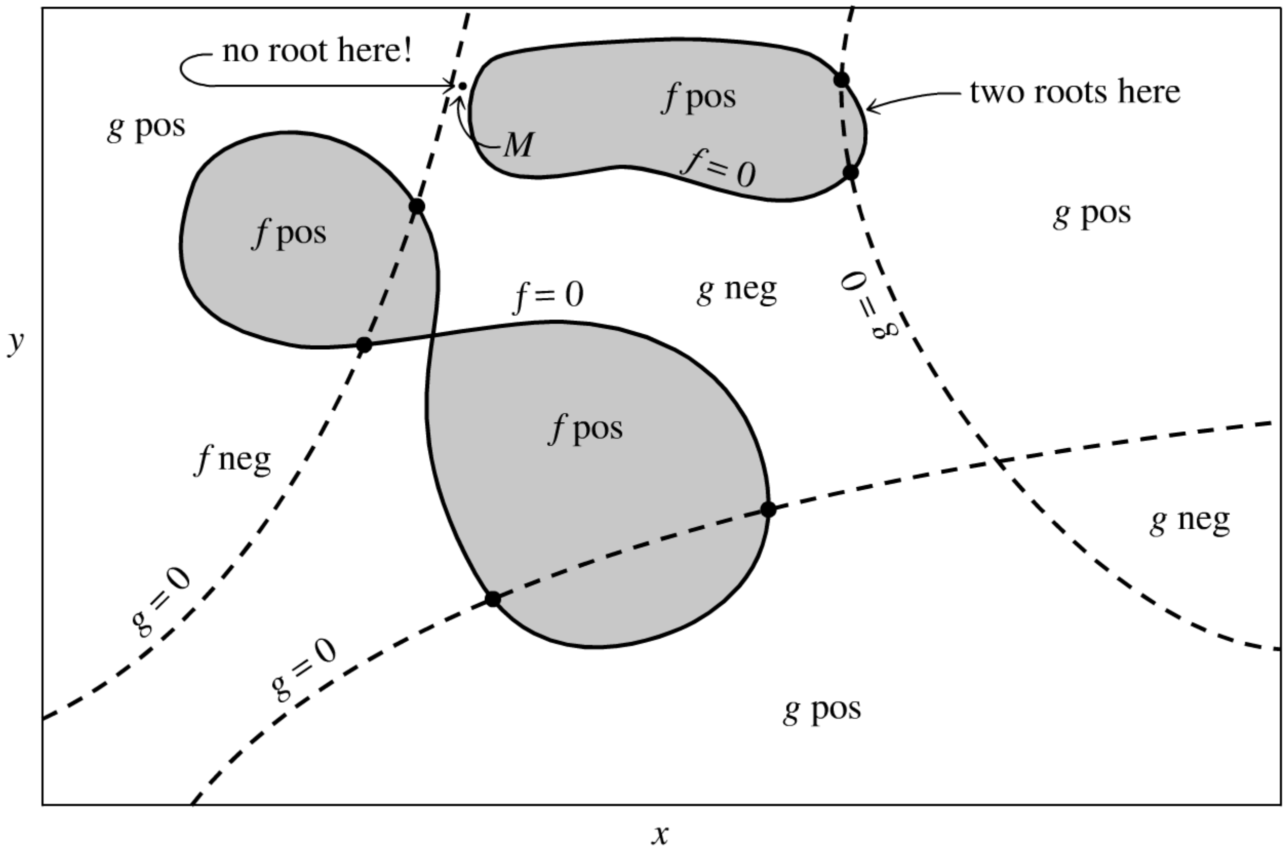


Gradient Descent in 1-d

- Suppose we do not have f'', but we have f': so we know the direction of function descent. We can take a small step in that direction: $\delta = -\eta f'(x)$. We must choose the sign of η to descend (if minimum is what we want) and it must be small enough not to overshoot.
- We can make a secant version of this method by evaluating gradient with finite difference: $f'(x_2) = [f(x_2)-f(x_1)]/(x_2-x_1)$

Nonlinear Equations in Many Dimensions

- $f(x,y) = 0$ and $g(x,y) = 0$: but the two functions f and g are unrelated, so it is difficult to look for general methods that will find all solutions



Newton-Raphson in Higher Dimensions

- Assume N functions

$$F_i(x_0, x_1, \ldots, x_{N-1}) = 0 \qquad i = 0, 1, \ldots, N-1.$$

- Taylor expand $F_i(\mathbf{x} + \delta\mathbf{x}) = F_i(\mathbf{x}) + \sum_{j=0}^{N-1} \frac{\partial F_i}{\partial x_j}\delta x_j + O(\delta\mathbf{x}^2).$
- Define Jacobian $J_{ij} \equiv \frac{\partial F_i}{\partial x_j}$
- In matrix notation $\mathbf{F}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{F}(\mathbf{x}) + \mathbf{J} \cdot \delta\mathbf{x} + O(\delta\mathbf{x}^2).$
- Setting $\mathbf{F}(\mathbf{x} + \delta\mathbf{x}) = 0$, we find $\mathbf{J} \cdot \delta\mathbf{x} = -\mathbf{F}.$
- This is a matrix equations: solve with LU
- Update $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \delta\mathbf{x}$ and iterate again

Globally Convergent Methods and Secant Methods

- If quadratic approximation in N-R method is not accurate taking a full step may make the solution worse. Instead one can do a line search backtracking and combine it with a descent direction (or use a thrust region).
- When derivatives are not available we can approximate them: multi-dimensional secant method (Broyden's method).
- Both of these methods have clear analogies in optimization and since the latter is more important for data science we will explain the concepts in optimization lecture next.

Relaxation Methods

- Another class of methods solving $x = f(x)$
- Take $x = 2\text{-}e^{-x}$, start at $x_0 = 1$ and evaluate $f(x_0) = 2\text{-}e^{-1} = 1.63 = x_1$
- Now use this solution again: $f(x_1) = 2\text{-}e^{-1.63} = 1.80 = x_2$
- Correct solution is $x = 1.84140\ldots$
- If there are multiple solutions which one one converges to depends on the starting point
- Convergence is not guaranteed: suppose x^0 is exact solution: $x_{n+1} = f(x_n) = f(x^0) + (x_n\text{-}x^0)f'(x^0) + \ldots$ since $x^0 = f(x^0)$ we get $x_{n+1}\text{-}x^0 = f'(x^0)(x_n\text{-}x^0)$ so this converges if $|f'(x^0)| < 1$
- When this is not satisfied we can try to invert the equation to get $u = f^{-1}(u)$ so that $|f'^{-1}(u)| < 1$

Relaxation Methods in Many Dimensions

- Same idea: write equations as $x = f(x,y)$ and $y = g(x,y)$, use some good starting point and see if you converge
- Easily generalized to N variables and equations
- Simple, and (sometimes) works!
- Again impossible to find all the solutions unless we know something about their structure

Over-relaxation

- We can accelerate the convergence:
- $\Delta x_n = x_{n+1} - x_n = f(x_n) - x_n$
- $x_{n+1} = x_n + (1+\omega)Dx_n$
- if $\omega = 0$ this is relaxation method
- If $\omega > 0$ this is over-relaxation method
- No general theory for how to select ω : trial and error

Literature

- *Numerical Recipes*, Press et al., Chapter 9, 10
- *Computational Physics,* M. Newman, Chapter 6