Computational Methods

Solving Equations

Solving Equations

- Solving scalar equations is an elemental task that arises in a wide range of applications
  - Corresponds to finding parameters that will achieve a particular outcome

$$f(x) = b$$

- Solving an equation is equivalent to root (or zero) finding for a related function

$$\tilde{f}(x) = f(x) - b$$

  - Can be solved by division for linear functions
  - Not analytically solvable in general since many non-linear functions are not easily invertible

Root Finding

- Numeric root finding algorithms are generally iterative algorithms
  - Each iteration attempts to find a value for x that is closer to the root of the system
- Numeric algorithms consider and often rely on the basic properties of the root and of the function
  - Continuity
  - Differentiability
  - Existence of root
  - Uniqueness or multiplicity of root

Existence

- Determining existence and uniqueness of solutions to the root finding problem can be complex
- Existence of a solution
  - A bracket is an interval $[a,b]$ for which $f(a)f(b)<0$
  - If a bracket exists for a continuous function $f(x)$ then the function has at least one root x^*.
- Number of solutions for a function is often difficult to determine
  - For polynomials the number of solutions is equal to the order of the polynomial

Uniqueness and Multiplicity

- Whether a function has a unique root can influence the solution approach taken
  - Linear functions mostly have a unique root (if it exists)
  - Non-linear function frequently have multiple roots
    - “local uniqueness” can be evaluated
- Multiplicity captures local non-uniqueness of a root
  - At non-simple roots (roots with multiplicity > 1) multiple roots coincide
  - The multiplicity of a root is the order of the lowest derivative that does not vanishes at *x**

$$m : f\left(x^*\right) = f'\left(x^*\right) = f^{(m-1)}\left(x^*\right) = 0 \;;\; f^{(m)}\left(x^*\right) \neq 0$$

Sensitivity and Conditioning

- Sensitivity of the root finding problem can be measured in terms of the condition number
  - Condition number for the root finding problem is the opposite of the one for the evaluation problem

    Absolute condition number (since *f(x*)=0*): $cond = \frac{1}{\left|f'\left(x^*\right)\right|}$
    - Root finding for a root is ill-conditioned if derivative is ≈0
    - Root finding at a multiple root is ill conditioned
- Approximation in backward or forward error
  - $\left|f(\hat{x})\right| \approx 0$ corresponds to small residual
  - $\left|\hat{x} - x^*\right| \approx 0$ represents closeness of solution

Convergence

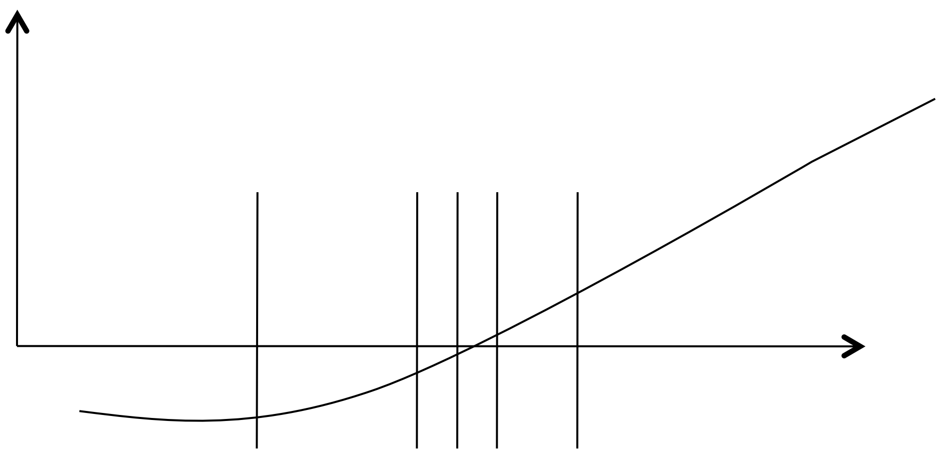
- For iterative methods it is generally important to evaluate convergence rate to estimate performance and complexity
  - Iteration error $e_k = x_k - x^*$
    - In interval methods, iteration error can be bounded by the width of the interval
  - Iterations converge with rate *r* if for constant *C*

$$\lim_{k \to \infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C$$

    - *r=1* linear convergence
    - *r>1* superlinear

Interval Bisection Method

- Bisection starts with an initial bracket
  - Determine function value in the middle of the bracket
  - Construct new bracket including the new point and one of the previous bracket end points
  - Repeat until the bracket has reached the termination width (corresponding to the remaining error bound)



Interval Bisection Method

- Requirements and Applicability
  - Bisection has only limited requirements for *f*
    - Function has to be continuous (but not differentiable)
    - Uses only the sign of the function value
- Convergence
  - Error can be measured by the width of the bracket
    - Halving of bracket yields linear convergence (*r=1, C=0.5*)
- Accuracy and Complexity
  - Iteration number is independent of function $\left\lceil \log_2\left(\frac{b-a}{tolerance}\right)\right\rceil$
    - Accuracy is a function of the number of iterations $|x-x^*| \leq \frac{b-a}{2^{n+1}}$
    - Complexity of each iteration equals one evaluation of the function

Fixed Point Iteration

- Fixed point iteration for root finding is an example of a redefinition of the problem
  - Fixed-point iteration uses a second, related function to compute the point for the next iteration

$$f(x) = 0 \quad \Leftrightarrow \quad x = g(x)$$

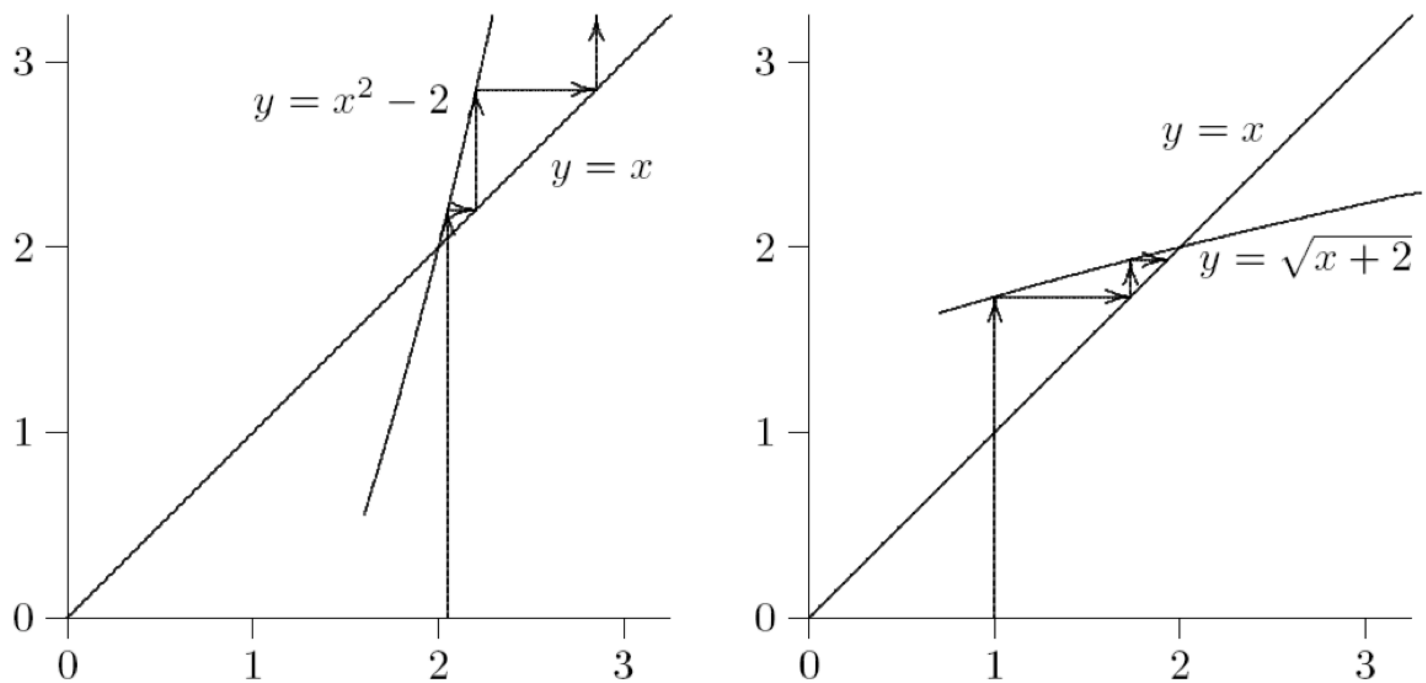
    - Fixed point of this function is the root of the original function
    - There can be many fixed-point problems for a given function *f*
  - Point for next iteration is computed using this function

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

- Fixed point iteration also called functional iteration

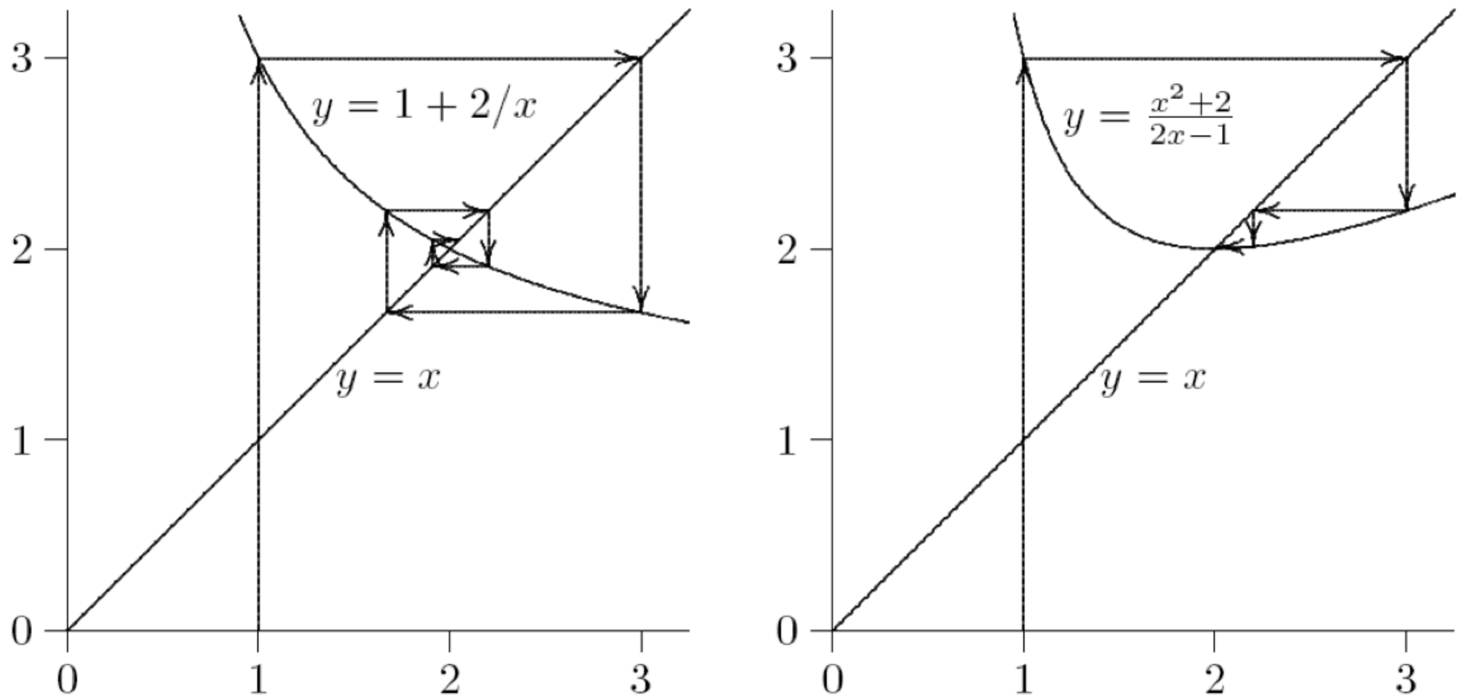
Fixed Point Iteration

$f(x)=x^2-x-2$



Fixed Point Iteration

$f(x)=x^2-x-2$



Fixed Point Iteration

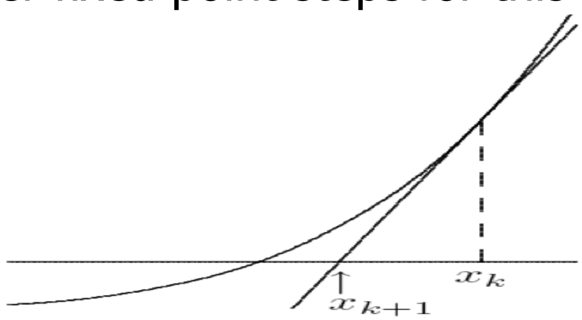
- Requirements and Applicability
  - Requires construction of function *g* for the function *f*
    - Function *g* has to be continuous and differentiable
- Convergence
  - Convergence is only guaranteed if $|g'(x^*)|<1$
    - Fixed point iteration is often only locally convergent
    - If $|g'(x^*)|<1$ *then the error converges at least linearly*
- Accuracy and Complexity
  - Accuracy is no longer tied strictly to iteration number
    - Need termination criterion $|x_k - x_{k-1}| \leq tolerance$
    - Each iteration requires one evaluation of *g*

Newton's Method

- Newton's method uses a locally linear approximation of the function *f*

$$f_x(h) = f(x+h) \approx f(x) + hf'(x)$$

  - Interpretation 1:
    - Iterate over root finding of the approximation function $f_x(h)$
  - Interpretation 2:
    - Approximation leads to a fixed point function $g(x) = x - \frac{f(x)}{f'(x)}$
    - Iterate over fixed point steps for this function *g*



Newton's Method

- Requirements and Applicability
  - Requires continuous and twice differentiable f
    - Both f and f' have to be known
- Convergence
  - Locally convergent
    - Converges quadratically for simple roots (i.e. multiplicity 1)
    - Converges linearly or sublinearly for a multiple root
- Accuracy and Complexity
  - Accuracy is not strictly tied to iteration number
    - Need termination criterion
    - Each iteration requires one evaluation of f and of f'

Modified Newton Method

- Newton's method can be modified to generally yield quadratic convergence by modifying *g* for a root with multiplicity *m*

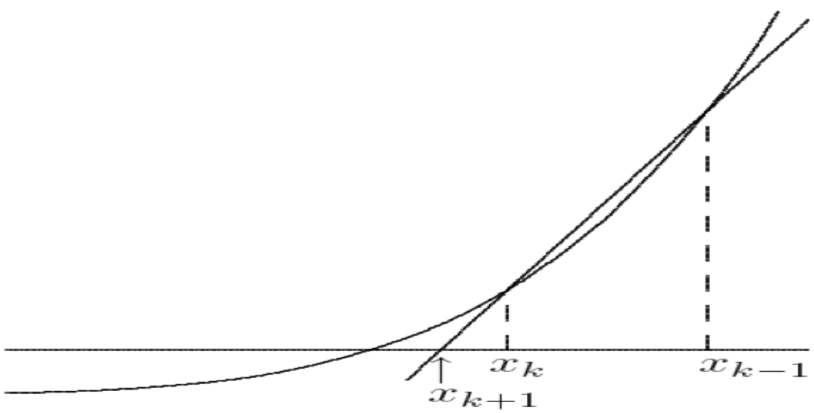
$$g(x) = x - \frac{mf(x)}{f'(x)}$$

  - Modified formulation changes the step size for multiple roots to avoid the drop in convergence rate
    - Modified Newton method converges quadratically for all roots
    - Requires knowledge about the multiplicity of a root (and thus the calculation of higher derivatives

Secant Method

- To avoid the need to know the derivative of *f*, Newton's method can be modified to replace it with a local approximation using the secant through the last two iterated points

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



Secant Method

- Requirements and Applicability
  - Requires continuous and differentiable *f*
    - Only *f* has to be known
- Convergence
  - Locally convergent
    - Converges superlinear (~1.62) for simple roots (i.e. multiplicity 1)
- Accuracy and Complexity
  - Accuracy is not strictly tied to iteration number
    - Need termination criterion
    - Each iteration requires one evaluation of *f* (first requires 2 evaluations)

Muller’s Method

- To accelerate convergence it is possible to use higher order interpolations
  - Muller’s method uses quadratic interpolation
    - Using the last 3 points, fit a second order polynomial (parabula)
    - Use the closes root as the next point (use alternative if no intersection point exists
  - Usually converges locally with superlinear rate (~1.84)
    - Interpolation might not have an intersection which requires an alternate option
  - Each iteration requires a second order polynomial fit operation

Inverse Interpolation

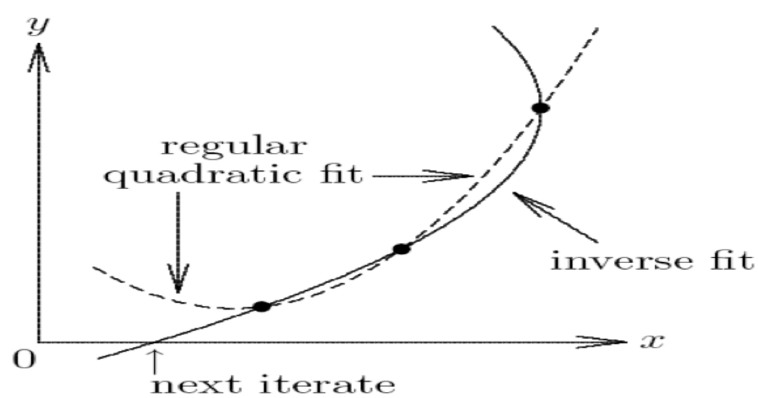
- To avoid lacking roots of the approximate function use an inverse interpolation function

$$x \approx p(f(x))$$

- Inverse Quadratic Interpolation (IQI)
  - Using the last 3 points, fit a second order polynomial (parabula)

$$p(y) = x_{k-2}\frac{(y-f(x_{k-1}))(y-f(x_k))}{(f(x_{k-2})-f(x_{k-1}))(f(x_{k-2})-f(x_k))} + x_{k-1}\frac{(y-f(x_{k-2}))(y-f(x_k))}{(f(x_{k-1})-f(x_{k-2}))(f(x_{k-1})-f(x_k))} + x_k\frac{(y-f(x_{k-2}))(y-f(x_{k-1}))}{(f(x_k)-f(x_{k-2}))(f(x_k)-f(x_{k-1}))}$$

  - Use the root as the next point



Hybrid Methods

- Hybrid methods combine features of others to accelerate root finding while preserving useful properties
  - Brent's Method
    - Guaranteed convergence from Bisection method
    - Fast convergence from Inverse quadratic interpolation and secant methods
    - Basic operation occurs using an initial bracket and a point within it
      - IQI is used first and if backward error decreases and new point cuts bracket in less than half, it is used to modify bracket.
      - If not, secant method is used
      - If none reduces the bracket sufficiently, the bisection method is applied.

Solving Equations

- Finding a set of parameters that leads to a particular solution for an equation is a common problem in science and engineering applications
  - Determining numeric solution for inverse kinematic problems
  - Specifying network requirements for a specific layout
  - Computing specification parameters for a circuit
- Iterative solutions can be used to efficiently find solutions fro arbitrary equations
  - Increasing convergence rates often reduce the ability to guarantee convergence
  - Problem reformulations can increase accuracy of the solution

Example Applications

- Robotics: Compute forward kinematics for a 2D closed kinematic chain
- Vision: Compute distance from the vergence angle of a symmetric stereo system
- Networks: Compute number of nodes for a particular bandwidth
- Systems: Compute the buffer size for a network interface card to limit dropped packets