CS 450: Numerical Anlaysis

Lecture 13

Chapter 5 – Nonlinear Equations

Existence, Conditioning, and 1D Methods for Nonlinear Equations

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

March 3, 2018

Solving Nonlinear Equations

- Solving (systems of) nonlinear equations corresponds to root finding:
  - $f(x^*) = 0$ *univariate nonlinear function*
  - $f(\boldsymbol{x}^*) = 0$ *multivariate, scalar-valued nonlinear function*
  - $\boldsymbol{f}(\boldsymbol{x}^*) = \mathbf{0}$ *multivariate, vector-valued nonlinear function*
- Root-finding can be reduced to finding a fixed-point $\boldsymbol{g}(\boldsymbol{x}^*) = \boldsymbol{x}^*$:
  - *various alternatives exist, including simple* $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$
  - *Newton's method uses (with Jacobian* $(\boldsymbol{J_f}(\boldsymbol{x}))_{ij} = \frac{\delta f_i}{\delta x_j}(\boldsymbol{x})$*),*

$$g(x) = x - f(x)/f'(x) \quad \textit{or more generally} \quad \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$$

which has the property $g'(x^*) = 0$*, or more generally* $\boldsymbol{J_g}(\boldsymbol{x}) = \boldsymbol{O}$

Nonexistence and Nonuniqueness of Solutions

- Solutions do not generally exist and are not generally unique, even in the univariate case:

  Consider functions that are strictly greater than zero or have many zeros.

- Solutions in the multivariate case correspond to intersections of hypersurfaces:

  The zeros of each equation define a hypersurface in $\mathbb{R}^n$, in the linear case, there are hyperplanes. Intersections of hypersurfaces for many equations, define the solutions, which are roots of all equations.

  Consider that two curves can intersect at many points in space. Two hypersurfaces in three-dimensional space may not intersect or may have multiple curves of intersection.

Conditions under which Solutions Exist

- *Intermediate value theorem* for univariate problems: *If for $x < y$, $sign(f(x)) \neq sign(f(y))$ and f is continuous, $\exists x^* \in [x, y], f(x^*) = 0$.*
- *Inverse function theorem* $\boldsymbol{J_f}(\boldsymbol{x})$ is nonsingular at $\boldsymbol{x}$ if $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$:

$$\boldsymbol{J_f}(\boldsymbol{x}) = \begin{bmatrix} \frac{df_1}{dx_1}(\boldsymbol{x}) & \cdots & \frac{df_1}{dx_n}(\boldsymbol{x}) \\ \vdots & & \vdots \\ \frac{df_m}{dx_1}(\boldsymbol{x}) & \cdots & \frac{df_m}{dx_n}(\boldsymbol{x}) \end{bmatrix}$$

  If $\boldsymbol{J_f}(\boldsymbol{x}^*)$ is singular, $\exists \boldsymbol{s} \neq \boldsymbol{0}$ so that $\boldsymbol{J_f}(\boldsymbol{x}^*)\boldsymbol{s} = \boldsymbol{0}$, which means a linear approximation cannot distinguish the soltion from a nearby point, $\boldsymbol{x}^* + \boldsymbol{s}$, which may or may not be another root.
- If a function has a unique fixed point in a given closed domain if it is is *contractive* and contained in that domain,

$$||\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{g}(\boldsymbol{z})|| \leq \gamma ||\boldsymbol{x} - \boldsymbol{z}||$$

  Contained implies that in the domain S, for any $\boldsymbol{x} \in S$, $\boldsymbol{g}(\boldsymbol{x}) \in S$, while contractive implies that the function is Lipschitz continuous in S.

Multiple Roots and Degeneracy

- If x^* is a root of f with multiplicity m,
  $f(x^*) = f'(x^*) = f''(x^*) = \cdots = f^{(m-1)}(x^*) = 0$:
  For some $t^{(0)}(x)$ *we have that*

$$\begin{aligned} f(x) &= (x - x^*)^m t^{(0)}(x) \\ f'(x) &= (x - x^*)^{m-1} t^{(0)}(x) + (x - x^*)^m t^{(0)'}(x) \\ &\equiv (x - x^*)^{m-1} t^{(1)}(x) \\ f^{(m-1)}(x) &= (x - x^*) t^{(m-1)}(x) \end{aligned}$$

  where $t^{(i)} = t^{(i-1)}(x) - (x - x^*) t^{(i-1)'}(x)$

- Increased multiplicity affects conditioning and convergence:
  When a root x^* *non-unit multiplicity,* $f'(x^*) = 0$*, so in a sense the problem of finding a particular root when two roots coincide is ill-posed.*

Conditioning of Nonlinear Equations

- Generally, we take interest in the absolute rather than relative conditioning of solving $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$:

  The sensitivity of solving a nonlinear equation, corresponds to the perturbation to the root due to a perturbation that has a bounded effect on the function. Without further knowledge of the specification of the function, it only makes sense to consider absolute perturbations to f, since a relative perturbation is undefined for $f(x^*) = 0$.

- The condition number of finding a root x^* of f is $1/|f'(x^*)|$ or $||\boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x}^*)||$:

  If we change f by a factor of at most δf at any point in the function while maintaining continuity, the root will shift by at most $|\delta f|/|f'(x^*)|$ assuming $|\delta f|$ is sufficiently small. This relationship is the converse of conditioning in functional evalution, where a perturbation to input x, results in a perturbation of at most $\kappa_{abs}(f) = |f'(x)|$ larger to the function value.

Bisection Algorithm

- Assume we know the desired root exists in a bracket $[a, b]$ and $sign(f(a)) \neq sign(f(b))$:
  - *note that multiple roots may exist in* $[a, b]$
  - *the condition of opposing sign is restrictive, we may want to find a root without knowing where a function is negative*
- Bisection subdivides the interval by a factor of two at each step by considering $f(c_k)$ at $c_k = (a_k + b_k)/2$:

$$[a_{k+1}, b_{k+1}] = \begin{cases} [c_k, b_k] & : sign(f(a_k)) = sign(f(c_k)) \\ [a_k, c_k] & : sign(f(b_k)) = sign(f(c_k)) \end{cases}$$

Rates of Convergence

- Let $\boldsymbol{x}_k$ be the kth iterate and $\boldsymbol{e}_k = \boldsymbol{x}_k = \boldsymbol{x}^*$ be the error, bisection obtains *linear convergence*, $\lim_{k\to\infty} ||\boldsymbol{e}_k||/||\boldsymbol{e}_{k-1}|| \leq C$:

  In bisection, working with the natural error bound given by bracket size,

  $$e_k = b_k - a_k = \frac{1}{2}(b_{k-1} - a_{k-1}) = \frac{1}{2}e_{k-1},$$

  so bisection achieve linear convergence with $C = 1/2$. With linear convergence, error $e_k \leq \epsilon$ is achieved after $O(\log_C(1/\epsilon))$ steps.

- rth order convergence implies that $||\boldsymbol{e}_k||/||\boldsymbol{e}_{k-1}||^r \leq C$

  rth order convergence implies the number of digits of correctness increases by a factor of r at each step. With rth order convergence, error $e_k \leq \epsilon$ is achieved after $O(\log_r(\log(1/\epsilon)))$ steps. Having achieved superlinear convergence ($r > 1$), methods differ only by constant factors in complexity.

Convergence of Fixed Point Iteration

- Fixed point iteration: $x_{k+1} = g(x_k)$ is locally linearly convergent if for $x^* = g(x^*)$, we have $|g'(x^*)| < 1$:

  By applying the intermediate value theorem to $g'(x)$ we can bound the error,

$$\begin{aligned} e_{k+1} = x_{k+1} - x^* = g(x_k) - g(x^*) \\ = g'(\theta_k)(x_k - x^*) \\ = g'(\theta_k)e_k, \quad \theta_k \in [x_k, x^*] \end{aligned}$$

- It is quadratically convergent if $g'(x^*) = 0$:

  Taylor's theorem allows us to show quadratic convergence,

$$\begin{aligned} e_{k+1} = x_{k+1} - x^* = g(x_k) - g(x^*) \\ = g''(\zeta_k)(x_k - x^*)^2/2 \\ = g''(\zeta_k)|e_k|^2/2, \quad \zeta_k \in [x_k, x^*] \end{aligned}$$

Newton's Method

- Newton's method is derived from a *Taylor series* expansion of f at x_k:

$$f(x) = \underbrace{f(x_k) + f'(x_k)(x - x_k)}_{\textit{secant line approximation}} + (1/2!)f''(x_k)(x - x_k)^2 + \cdots$$

- Newton's method is *quadratically convergent* if started sufficiently close to x^* so long as $f'(x^*) \neq 0$:

$$f(x^*) - f(x_{k+1}) \leq (1/2)f''(x_k)(x - x_k)^2 + \cdots = (1/2)f''(\xi_k)||e_k||^2, \quad \xi_k \in [x_k, x^*]$$

Secant Method

- The Secant method approximates $f'(x_k) \approx \frac{f(x_k)-f(x_{k-1})}{x_k - x_{k-1}}$:

  Usually this method is the cheapest approximation possible, since function values $f(x_k)$ and $f(x_{k-1})$ are already available. Approximation quality depends on magnitude $f(x_k) - f(x_{k-1})$ and $x_k - x_{k-1}$. If the two points are far apart, the derivative approximation may be bad locally, while if they are very close we have to take care in handling cancellation. A well-chosen finite-difference step at each x_k provides a more robust approximation, but requires another function evaluation.

- The convergence is *superlinear* but not quadratic:

  The error will now depend on the previous two errors, since we are using the previous two points, in simplified form,

$$e_k \leq e_{k-1} e_{k-2}$$

  Now note $\log(e_k) = \log(e_{k-1}) + \log(e_{k-2})$ is the Fibonacci sequence, which grows at a rate of $r = (1+\sqrt{5})/2$. Thus the (negative) exponent of the error increases by a factor of r at each step, i.e. the convergence rate is r.

Nonlinear Tangential Interpolants

- Secant method uses a linear interpolant based on points $f(x_k)$, $f(x_{k-1})$, could use more points and higher-order interpolant:

  Have points $(x_0, f(x_0)), \ldots, (x_k, f(x_k))$ *can fit polynomial to* $p(x_i) = f(x_i)$ *for some subset of points* $x_i \in S \subseteq \{x_0, \ldots x_k\}$.

- Quadratic interpolation (Muller's method) achieves convergence rate $r \approx 1.84$:

  Quadratic interpolation requires three points x_{k-2}, x_{k-1}, *and* x_k.

Achieving Global Convergence

- Hybrid bisection/Newton methods:

  Given a bracket (interval), can proceed with bisection until bracket is small then switch to Newton. Alternatively, can attempt Newton, check if it stays within bracket (safeguard) and proceed with change only if it does.

- Bounded (damped) step-size:

  Newton's method gives us a direction. Decreasing the step size in that direction trades off convergence rate for reliability. We will study how step sizes can be chosen in more detail in the context of optimization.