CS 450: Numerical Anlaysis

Lecture 14

Chapter 5 – Nonlinear Equations

Newton's Method for Systems of Nonlinear Equations

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

March 5, 2018

Review Solving a Nonlinear Equation

- Newton's and secant method provide basic approaches for solving a univariate nonlinear equation:

$$x_{k+1}^{\textit{Newton}} = x_k - f(x_k)/f'(x_k)$$

$$x_{k+1}^{\textit{Secant}} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

- Inverse (quadratic) interpolation can provide better convergence:

  Interpolate quadratic polynomial q so that $q(f(x_i)) = x_i$ for $i \in \{k, k-1, k-2\}$, then pick approximate root as $x_{k+1} = q(0)$.

Systems of Nonlinear Equations

- Given $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_m(\boldsymbol{x}) \end{bmatrix}$ for $\boldsymbol{x} \in \mathbb{R}^n$, seek $\boldsymbol{x}^* \in \mathbb{R}^n$ so that $\boldsymbol{f}(\boldsymbol{x}^*) = \mathbf{0}$

  $\boldsymbol{x}^*$ must simultaneously set to zero all components of $\boldsymbol{f}$:
  $f_1(\boldsymbol{x}^*) = \cdots = f_m(\boldsymbol{x}^*) = 0.$

- At a particular point $\boldsymbol{x}$, the *Jacobian* of $\boldsymbol{f}$, describes how $\boldsymbol{f}$ changes in a given direction of change in $\boldsymbol{x}$,

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

  Our local approximation is given by

$$\boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta x}) \approx \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{J_f}(\boldsymbol{x})\boldsymbol{\delta x},$$

  note that when $m = 1$ the Jacobian corresponds to the gradient of $\boldsymbol{f}$.

Multivariate Fixed-Point and Newton Iteration

- Fixed-point iteration $\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k)$ achieves local convergence so long as $|\lambda_{\text{max}}(\boldsymbol{J_g}(\boldsymbol{x}^*))| < 1$:
  Given starting point $\boldsymbol{x}_0$ close enough to $\boldsymbol{x}^\star$, we will have $|\lambda_{max}(\boldsymbol{J_g}(\boldsymbol{x}_i))|, \forall i$.
- Newton's method corresponds to the fixed-point iteration

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$$

Note that generally Newton's method iteration has a fixed-point $\bar{\boldsymbol{x}}$ whenever $\boldsymbol{f}(\bar{\boldsymbol{x}}) = \boldsymbol{0}$, i.e. we have found a root of $\boldsymbol{f}$, namely $\boldsymbol{x}^* = \bar{\boldsymbol{x}}$.
A necessary assumption is that $\boldsymbol{J_f}(\boldsymbol{x}^*)$ is nonsingular, otherwise we can find nonzero solutions $\boldsymbol{y}$ to $\boldsymbol{J_f}(\boldsymbol{x}^*)\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$.

Convergence of Newton Iteration

- Newton's method achieves quadratic local convergence if $||\boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x}^*)||$ is bounded:

$$
\begin{aligned}
\boldsymbol{e}_k &= \boldsymbol{x}_k - \boldsymbol{x}^* = \boldsymbol{g}(\boldsymbol{x}_{k-1}) - \boldsymbol{x}^* \\
&= \boldsymbol{x}_{k-1} - \boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x}_{k-1})\boldsymbol{f}(\boldsymbol{x}_{k-1}) - \boldsymbol{x}^* \\
&= \boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x}_{k-1})(\boldsymbol{f}(\boldsymbol{x}_{k-1}) - \boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}_{k-1})(\boldsymbol{x}^* - \boldsymbol{x}_{k-1})) \\
||\boldsymbol{e}_k|| &\leq ||\boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x}_{k-1})||O(||\boldsymbol{x}^* - \boldsymbol{x}_{k-1}||^2) \\
&= ||\boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x}_{k-1})||O(||\boldsymbol{e}_{k-1}||^2)
\end{aligned}
$$

Convergence of Newton Iteration (II)

- Quadratic convergence is achieved when the Jacobian of a fixed-point iteration is zero at the solution, which is true for Newton's method:

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x}) &= \boldsymbol{x} - \boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) \\ \boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}) &= \boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x})\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}) - \sum_i f_i(\boldsymbol{x})\boldsymbol{H}_{\boldsymbol{f}}^{(i)}(\boldsymbol{x}) \\ &= \boldsymbol{I} - \boldsymbol{I} - \boldsymbol{O} = \boldsymbol{O} \end{aligned}$$

where $\boldsymbol{H}_{\boldsymbol{f}}^{(i)}$ is the ith component of the derivative of $\boldsymbol{J}_{\boldsymbol{f}}^{-1}(\boldsymbol{x})$ of $\boldsymbol{f}$.

Estimating the Jacobian using Finite Differences

- To obtain $\boldsymbol{J_f}(\boldsymbol{x}_k)$ at iteration k, can use finite differences:
  For $n = 1$, we have $\boldsymbol{J_f} \approx (1/h)(\boldsymbol{f}(x_k + h) - \boldsymbol{f}(x_k))$.
  More generally, the ith column of $\boldsymbol{j}_i$ of the Jacobian is given by

$$\boldsymbol{j}_i \approx (1/h)(\boldsymbol{f}(\boldsymbol{x}_k + h\boldsymbol{e}_i) - \boldsymbol{f}(\boldsymbol{x}_k)).$$

- $n + 1$ function evaluations are needed: $\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x} + h\boldsymbol{e}_i) \forall i \in \{1, \dots, n\}$, which correspond to $m(n + 1)$ scalar function evaluations.

Cost of Multivariate Newton Iteration

- What is the cost of solving $\boldsymbol{J_f}(\boldsymbol{x}_k)\boldsymbol{s}_k = \boldsymbol{f}(\boldsymbol{x}_k)$?
  $O(n^3)$
- What is the cost of Newton's iteration overall?
  For k steps, $O(n^3 k + k n^2 \gamma_{\textit{func-eval}})$.