Nonlinear Systems

Part II Roadmap

- Part I – Linear Algebra (units 1-12) $Ac = b$
- Part II – Optimization (units 13-20)
  - (units 13-16) Optimization -> Nonlinear Equations -> 1D roots/minima
  - (units 17-18) Computing/Avoiding Derivatives
  - (unit 19) Hack 1.0: "I give up" $H = I$ and J is mostly 0 (descent methods)
  - (unit 20) Hack 2.0: "It's an ODE!?" (adaptive learning rate and momentum)

linearize

line search

Theory

Methods

Recall: Jacobian (Unit 9)

- The Jacobian of $F(c) = \begin{pmatrix} F_1(c) \\ F_2(c) \\ \vdots \\ F_m(c) \end{pmatrix}$ has entries $J_{ik} = \frac{\partial F_i}{\partial c_k}(c)$

- Thus, the Jacobian $J(c) = F'(c) = \begin{pmatrix} \frac{\partial F_1}{\partial c_1}(c) & \frac{\partial F_1}{\partial c_2}(c) & \cdots & \frac{\partial F_1}{\partial c_n}(c) \\ \frac{\partial F_2}{\partial c_1}(c) & \frac{\partial F_2}{\partial c_2}(c) & \cdots & \frac{\partial F_2}{\partial c_n}(c) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial c_1}(c) & \frac{\partial F_m}{\partial c_2}(c) & \cdots & \frac{\partial F_m}{\partial c_n}(c) \end{pmatrix}$

Linearization

- Solving a nonlinear system of equations $F(c) = 0$ is difficult
- Linearize via the multidimensional version of the Taylor expansion:

$$F(c) \approx F(c^*) + F'(c^*)\,(c - c^*)$$

  - More valid when $\Delta c = c - c^*$ is small (i.e. for c close enough to c^*)
  - Alternatively written as $F(c) - F(c^*) \approx F'(c^*)\Delta c$
- The chain rule $\frac{dF(c)}{dt} = F'(c)\frac{dc}{dt}$ is valid for any variable t, and thus can be written in differential form as $dF(c) = F'(c)dc$
  - Often referred to as the total derivative
  - Using finite size differentials leads to the approximation: $\Delta F(c) \approx F'(c)\Delta c$
  - In 1D, $df = f'(c)dc$ and $\Delta f \approx f'(c)\Delta c$ are the usual $\frac{df}{dc} = f'(c)$ and $\frac{\Delta f}{\Delta c} \approx f'(c)$

Newton's Method

- An iterative method: start with c^0, recursively find: c^1, c^2, c^3, ...

- Based on $\Delta F(c) \approx F'(c)\Delta c$, write $F(c^{q+1}) - F(c^q) = F'(c^q)\Delta c^q$
  - Aiming for $F(c) = 0$ motivates setting $F(c^{q+1}) = 0$
  - Alternatively, set $F(c^{q+1}) = \beta F(c^q)$ where $0 \leq \beta < 1$ aims to slowly shrink $F(c^q)$ towards zero

- Solve the linear system $F'(c^q)\Delta c^q = (\beta - 1)F(c^q)$ for Δc^q
- Use $\Delta c^q = c^{q+1} - c^q$ to update $c^{q+1} = c^q + \Delta c^q$

Newton's Method

- Requires repeatedly solving a linear system, making robustness and efficiency for linear system solvers quite important
  - Need to consider size, rank, conditioning, symmetry, etc. of $F'(c^q)$

- $F'(c^q)$ may be difficult to compute, since it requires every first derivative
  - Newton's Method contains linearization errors, so approximations of $F'(c^q)$ are often valid/worthwhile (e.g. symmetric approximation, etc.)
  - More on this in units 17/18 on Computing/Avoiding Derivatives

- Generally speaking, there are no guarantees on convergence
  - May converge to any one of many roots when multiple roots exist, or not converge at all

Solving Linear Systems (Review)

- Theory, all matrices: SVD (units 3, 10, 11)
- Square, full rank, dense:
  - LU factorization with pivoting (unit 2)
  - Symmetric: Cholesky factorization (unit 4), Symmetric approximation (unit 4)
- Square, full rank, sparse (iterative solvers) (unit 5):
  - SPD (sometimes SPSD): Conjugate Gradients
  - Nonsymmetric/Indefinite: GMRES, MINRES, BiCGSTAB (not steepest descent)
- Tall, full rank (least squares to minimize residual) (unit 8):
  - normal equations (units 9, 10), QR, Gram-Schmidt, Householder (unit 10)
- Any size/rank (minimum norm solution) (unit 11):
  - Pseudo-Inverse, PCA approximation, Power Method (unit 11)
  - Levenberg-Marquardt (iteration too), Column Space Geometric Approach (unit 12)

Line Search

- Given the linearization errors in $F'(c^q)\Delta c^q = (\beta - 1)F(c^q)$, the resulting Δc^q can lead to a poor estimate for c^{q+1} via $c^{q+1} = c^q + \Delta c^q$

- Instead, Δc^q is often just used as a search direction, i.e. $c^{q+1} = c^q + \alpha^q \Delta c^q$
- The 1D (parameterized) line $c^{q+1}(\alpha) = c^q + \alpha \Delta c^q$ is the new domain
- Find an α with $F\left(c^{q+1}(\alpha)\right) = 0$ simultaneously for all equations

- Safe Set methods restrict α in various ways, e.g. $0 \leq \alpha \leq 1$

Line Search

- Since F is vector valued, consider $g(\alpha) = F\left(c^{q+1}(\alpha)\right)^T F\left(c^{q+1}(\alpha)\right) = 0$
- Since $g(\alpha) \geq 0$, solutions to $F\left(c^{q+1}(\alpha)\right) = 0$ are minima of $g(\alpha)$
- $g(\alpha)$ might be <u>strictly</u> positive (with no $g(\alpha) = 0$), but minimizing $g(\alpha)$ might still help to make progress towards an α with $F\left(c^{q+1}(\alpha)\right) = 0$

- <u>Option 1</u>: find simultaneous roots of the vector valued $F\left(c^{q+1}(\alpha)\right) = 0$
- <u>Option 2</u>: find roots of or minimize $g(\alpha) = \frac{1}{2} F^T\left(c^{q+1}(\alpha)\right) F\left(c^{q+1}(\alpha)\right)$, to find or make progress towards an α with $F\left(c^{q+1}(\alpha)\right) = 0$

Optimization Problems

- Minimize the scalar cost function $\hat{f}(c)$ by finding the critical points where $\nabla\hat{f}(c) = J_{\hat{f}}^T(c) = F(c) = 0$
- $F'(c^q)\Delta c^q = (\beta - 1)F(c^q)$ gives the search direction (as usual)
- Here, $F'(c) = J_F(c) = H_{\hat{f}}^T(c)$
- So, solve $H_{\hat{f}}^T(c^q)\Delta c^q = (\beta - 1)J_{\hat{f}}^T(c^q)$ to find the search direction Δc^q
- Option 1: find simultaneous roots of the vector valued $J_{\hat{f}}^T\left(c^{q+1}(\alpha)\right) = 0$, which are critical points of $\hat{f}(c)$
- Option 2: find roots of or minimize $g(\alpha) = \frac{1}{2}J_{\hat{f}}\left(c^{q+1}(\alpha)\right)J_{\hat{f}}^T\left(c^{q+1}(\alpha)\right)$, to find or make progress towards critical points of $\hat{f}(c)$
- Option 3: minimize $\hat{f}(c^{q+1}(\alpha))$ directly