Ch 10.2: Newton's method for systems

Tuesday, September 23, 2025 2:18 PM

Let $F(\vec{x}) = \begin{bmatrix} f_1(\vec{x}) \\ \vdots \\ f_n(\vec{x}) \end{bmatrix}$, $F: \mathbb{R}^n \to \mathbb{R}^n$. Look for a solution $F(\vec{x}) = \vec{0}$.

n equations, n variables

For now, these must match

Multi-dim. Taylor Series:

$$F(\vec{x}) = \underbrace{F(\vec{x}^{(k)}) + \overbrace{J_F}^{\text{Jacobian}}(\vec{x}^{(k)}) \cdot (\vec{x} - \vec{x}^{(k)})}_{\text{this is our model } m(\vec{x})} + \ldots \text{higher order terms that we'll neglect} \ldots$$

Solve for a root of our simple model

Find $\vec{x}$ s.t.

$$0 = F(\vec{x}^{(k)}) + J_F(\vec{x}^{(k)}) \cdot (\vec{x} - \vec{x}^{(k)})$$

Recall the Jacobian

$$\left(J_F(\vec{x})\right)_{ij} = \frac{\partial f_i}{\partial x_j}(\vec{x})$$

i.e.

$$J_F(\vec{x}) = \begin{bmatrix} -\nabla f_1(\vec{x})^T- \\ \vdots \\ -\nabla f_n(\vec{x})^T- \end{bmatrix}$$

i.e. NEWTON'S METHOD

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \underbrace{J_F(\vec{x}^{(k)})^{-1} \cdot F(\vec{x}^{(k)})}$$

$$[\ ] = [\ ] - [\quad]^{-1} \cdot [\ ]$$

use LU factorization

or backslash (Matlab)

or np.linalg.solve(J, F)

NOT inverse.

generalizes 1D case

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}.$$

Like 1D case,

1. need not converge — diverge, or $f'(x)=0$ or $J(\vec{x})$ is singular
2. for nice problems, if started close enough, converges quadratically.
3. computing derivatives (Jacobian) is annoying

   ... autodiff helps ...

Unlike 1D case,

- Solving $J_F(\vec{x}^{(k)})^{-1} F(x^{(k)})$ costs $O(n^3)$ flops in general ... i.e. it gets worse as dimension n increases

  So it's rare to use Newton in very high dimension
- Often approximately solve, eg. via a Krylov Subspace method we'll see later... and use Jacobian-vector-products from your autodiff software.

Ch 10.2, p. 2: Affine Invariance of Newton's method

Wednesday, September 24, 2025 2:23 PM

Affine change-of-variables won't affect convergence rate

($\Rightarrow$ Newton is impervious to some types of ill-conditioning)

$F(\vec{x}) = \vec{0}$, Newton is $\vec{x} \leftarrow \vec{x} - J_F(\vec{x})^{-1} F(\vec{x})$

invertible

try $\tilde{F}(\vec{y}) = \vec{0}$, $\tilde{F}(\vec{y}) := F(\underbrace{A\vec{y} + \vec{b}}_{\text{affine transformation of } \vec{y}})$

This is an equivalent problem, in this sense

$$\left\{ \begin{array}{l} \tilde{F}(\vec{y}) = 0 \Rightarrow \vec{x} = A\vec{y} + \vec{b} \text{ is a root of } F \\ F(\vec{x}) = 0 \Rightarrow \vec{y} = A^{-1}(\vec{x} - \vec{b}) \text{ is a root of } \tilde{F} \end{array} \right.$$

run Newton on $\tilde{F}$:

$$\vec{y} \leftarrow \vec{y} - J_{\tilde{F}}(\vec{y}) \cdot \tilde{F}(\vec{y})$$

$J_{\tilde{F}}(\vec{y}) = J_F(A\vec{y} + \vec{b}) \cdot A$ via chain rule

$$\vec{y} \leftarrow \vec{y} - \left(J_F(A\vec{y} + \vec{b})\, A \right)^{-1} F(A\vec{y} + b)$$

$$\vec{y} \leftarrow \vec{y} - A^{-1} J_F(A\vec{y} + \vec{b})^{-1} F(A\vec{y} + \vec{b})$$

$$A\vec{y} \leftarrow A\vec{y} - J_F(A\vec{y} + \vec{b})^{-1} F(A\vec{y} + \vec{b})$$

$$A\vec{y} + b \leftarrow A\vec{y} + b - J_F(A\vec{y} + b)^{-1} F(A\vec{y} + b)$$

$\vec{x} := A\vec{y} + \vec{b}$

$$\vec{x} \leftarrow \vec{x} - J_F(\vec{x})^{-1} F(\vec{x})$$

which is our original Newton iteration