Mathematics Methods for Computer Science

Instructor: Xubo Yang

SJTU-SE DALAB

Lecture

Nonlinear Systems II: Multiple Variables

Today's Root-Finding Problems

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

One "Easy" Instance

$$f(\vec{x}) = A\vec{x} - \vec{b}$$

Usual Assumption

For $f : \mathbb{R}^n \to \mathbb{R}^m$, assume

$$n \geq m.$$

Examples (whiteboard)

Common Examples

On whiteboard:

1. Implicit integration ($n = m$)
2. Projecting onto constraints ($n > m$)
   E.g., Robotics (inverse kinematics)

Jacobian

$$(Df)_{ij} \equiv \frac{\partial f_i}{\partial x_j}$$

对于多变量问题，使用上述雅各比矩阵的形式表示各个导数

Jacobian

$$(Df)_{ij} \equiv \frac{\partial f_i}{\partial x_j}$$

How big is Df for $f : \mathbb{R}^n \to \mathbb{R}^m$?

First-Order Approximation of $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\vec{x}) \approx f(\vec{x}_k) + Df(\vec{x}_k) \cdot (\vec{x} - \vec{x}_k)$$

First-Order Approximation of $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\vec{x}) \approx f(\vec{x}_k) + Df(\vec{x}_k) \cdot (\vec{x} - \vec{x}_k)$$

Newton's Method:

$$\vec{x}_{k+1} = \vec{x}_k - \left[Df\left(\vec{x}_k\right)\right]^{-1} f\left(\vec{x}_k\right)$$

First-Order Approximation of $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\vec{x}) \approx f(\vec{x}_k) + Df(\vec{x}_k) \cdot (\vec{x} - \vec{x}_k)$$

Newton's Method:

$$\vec{x}_{k+1} = \vec{x}_k - [Df(\vec{x}_k)]^{-1} f(\vec{x}_k)$$

Review: Do we explicitly compute $[Df(\vec{x}_k)]^{-1}$

Convergence Sketch

1. $x_{\vec{k+1}} = g(\vec{x}_k)$ converges when the maximum-magnitude eigenvalue of Dg is less than 1 (当最大的特征值<1时收敛)
2. Extend observations about (quadratic) convergence in multiple dimensions

Two Problems

1. Differentiation is hard
2. $Df(\vec{x}_k)$ changes every iteration

Extend Secant Method?

Extend Secant Method?

Direct extensions are **not obvious!**

Observation: Directional Derivative

(方向导数)

$$D_{\vec{v}}f = Df \cdot \vec{v}$$

Secant-Like Approximation

$$J \cdot (\vec{x}_k - \vec{x}_{k-1})$$
$$\approx f(\vec{x}_k) - f(\vec{x}_{k-1})$$
$$\text{where } J \approx Df(\vec{x}_k)$$

Secant-Like Approximation

$$J \cdot (\vec{x}_k - \vec{x}_{k-1})$$
$$\approx f(\vec{x}_k) - f(\vec{x}_{k-1})$$
$$\text{where } J \approx Df(\vec{x}_k)$$

"Broyden's Method"

Broyden's Method: Outline

- Maintain current iterate $\vec{x}_k$ and approximation J_k of Jacobian near $\vec{x}_k$
- Update $\vec{x}_k$ using Newton-like step
- Update J_k using secant-like formula

Deriving the Broyden Step

$$\text{minimize}_{J_k} \|J_k - J_{k-1}\|^2_{\text{Fro}}$$

$$\text{such that } J_k \cdot (\vec{x}_k - \vec{x}_{k-1}) = f(\vec{x}_k) - f(\vec{x}_{k-1})$$

Deriving the Broyden Step

$$\text{minimize}_{J_k} \|J_k - J_{k-1}\|^2_{\text{Fro}}$$
$$\text{such that } J_k \cdot (\vec{x}_k - \vec{x}_{k-1}) = f(\vec{x}_k) - f(\vec{x}_{k-1})$$

$$J_k = J_{k-1} + \frac{(f(\vec{x}_k) - f(\vec{x}_{k-1}) - J_{k-1} \cdot \Delta\vec{x})}{\|\vec{x}_k - \vec{x}_{k-1}\|^2_2} (\Delta\vec{x})^\top$$

(使用上面的式子对J_k进行更新)

The Newton Step

$$\vec{x}_{k+1} = \vec{x}_k - J_k^{-1} f(\vec{x}_k)$$