Ch 10.3: secant method, quasi-Newton methods

Tuesday, September 23, 2025 2:19 PM

In 1D, the **secant method** approximates Newton by replacing

$$f'(x_k) \text{ with } \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

In higher dimensions, we can do the same, but have many options, so we get a whole class of **quasi-Newton methods**

For large dimensions, these are state-of-the-art

- you don't have to program the Jacobian — nice
- you don't have to solve a large linear system — nice!

Goal: approximate $J_F(\vec{x}^{(k)})$ and/or approximate the Newton step

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

$F: \mathbb{R}^n \to \mathbb{R}^n$, J_F its Jacobian, $J_F: \mathbb{R}^n \to \mathbb{R}^{n \times n}$

let's call our approximation B_k

so we'll replace $\vec{x}^{(k+1)} = \vec{x}^{(k)} - J_F(\vec{x}^{(k)})^{-1} F(\vec{x}^{(k)})$

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

Considerations:

Taylor expand

$$\underbrace{F(\vec{x}^{(k-1)})}_{F_{k-1}} = \underbrace{F(\vec{x}^{(k)})}_{F_k} + J_F(\vec{x}^{(k)}) \cdot (\vec{x}^{(k-1)} - \vec{x}^{(k)}) + \text{higher order terms}$$

F_k shorthand notation

so

$$F_k - F_{k-1} \approx J_F(\vec{x}^{(k)}) \cdot (\vec{x}^{(k)} - \vec{x}^{(k-1)})$$

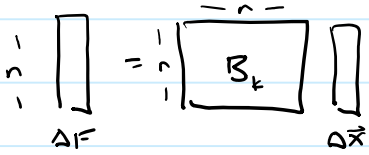
hence let's impose

$$\underbrace{F_k - F_{k-1}}_{\Delta F_{k-1}} = B_k \underbrace{(\vec{x}^{(k)} - \vec{x}^{(k-1)})}_{\Delta \vec{x}^{(k-1)}}$$

"secant equations"

In 1D, this is 1 equation, 1 unknown, exactly 1 unique solution: $b_k = \frac{f_k - f_{k-1}}{x_k - x_{k-1}}$ ... the secant method

but in n-dim, this is



n equations, n^2 unknowns!

No unique answer ... hence many quasi-Newton methods

Friday, September 26, 2025 10:10 AM

So, we impose $\Delta F_{k-1} = B_k \cdot \Delta \vec{x}^{(k-1)}$ but we need more criterion

Main idea: choose B_k to be a perturbation of B_{k-1}

One particular method is BROYDEN'S METHOD:

insist $B_k = B_{k-1} + \vec{u}\,\vec{v}^T$, a low-rank perturbation.

Then find $\vec{u}, \vec{v}$ via imposing secant equations:

$$\Delta F = (B_{k-1} + \vec{u}\vec{v}^T)\Delta x,$$

$$\vec{u} \cdot (\underbrace{\vec{v}^T \cdot \Delta x}_{\text{scalar}}) = \underbrace{\Delta F - B_{k-1} \cdot \Delta x}_{\vec{r}}$$

i.e.

if this is $\vec{0}$, set $\vec{u} = \vec{v} = \vec{0}$ i.e. $B_{k+1} = B_k$

so $\vec{u} = \frac{1}{\vec{v}^T \Delta x} \cdot \vec{r}$

Not quite satisfactory ($\vec{u}$ in terms of $\vec{v}$ ... what is $\vec{v}$?)

Frobenius norm

$\|A\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}$

so also ask for $\|B_k - B_{k-1}\|_F^2$ ← Frobenius norm to be as small as possible, i.e. min $\|\vec{u}\vec{v}^T\|_F^2$

Same as:

$\|A\|_F = \sqrt{\text{tr}(A^T A)}$

$$\|\vec{u}\vec{v}^T\|_F^2 = \text{tr}\left((\vec{u}\vec{v}^T)^T \cdot \vec{u}\vec{v}^T\right)$$

$$= \text{tr}\left(\vec{v} \cdot \vec{u}^T \cdot \vec{u} \cdot \vec{v}^T\right)$$

$$= \text{tr}\left(\underbrace{\vec{u}^T\vec{u}}_{\text{scalar}} \cdot \underbrace{\vec{v}^T\vec{v}}_{\text{scalar}}\right)$$ via circular property of trace: $\text{tr}(ABC) = \text{tr}(CAB)$

$$= (\vec{u}^T\vec{u}) \cdot (\vec{v}^T\vec{v})$$

$$= \|\vec{u}\|_2^2 \cdot \|\vec{v}\|_2^2$$

So...

$$\min \|\vec{u}\|_2^2 \cdot \|\vec{v}\|_2^2$$

$$= \min \frac{1}{(\vec{v}^T\Delta x)^2} \cdot \|\vec{r}\|^2 \cdot \|\vec{v}\|^2 = \|\vec{r}\|^2 \cdot \min_{\vec{v}} \frac{\|\vec{v}\|^2}{(\vec{v}^T\Delta x)^2}$$

↑ constant; ↓ want large

Cauchy-Schwarz: $\vec{v}^T\Delta\vec{x} \le \|\vec{v}\| \cdot \|\Delta x\|$ and is an equality when $\vec{v} = \alpha \cdot \Delta\vec{x}$

So for denominator as large as possible, $\vec{v} = \alpha \cdot \Delta\vec{x}$,

hence $\vec{u} \cdot \vec{v}^T = \frac{1}{\vec{v}^T \cdot \Delta x} \cdot \vec{r} \cdot \vec{v}^T = \frac{1}{\alpha \cdot \Delta\vec{x}^T \Delta x} \cdot \vec{r} \cdot \alpha \cdot \Delta\vec{x}^T = \boxed{\frac{\vec{r} \cdot \Delta\vec{x}^T}{\|\Delta\vec{x}\|_2^2}}$

Friday, September 26, 2025 10:45 AM

so for BROYDEN

$$B_k = B_{k-1} + \frac{1}{\|\Delta\vec{x}_{k-1}\|_2^2} \cdot \vec{r}_{k-1} \cdot \Delta\vec{x}_{k-1}$$

where $\Delta\vec{x}_{k-1} = \vec{x}^{(k)} - \vec{x}^{(k-1)}$

$\vec{r}_{k-1} = \Delta F_{k-1} - B_{k-1} \cdot \Delta\vec{x}_{k-1}$,

$\Delta F_{k-1} = F(\vec{x}^{(k)}) - F(\vec{x}^{(k-1)})$

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

and in practice we update B_k^{-1} directly (not B_k)

using the Sherman-Morrison formula

$$B_k^{-1} = B_{k-1}^{-1} + \frac{\Delta\vec{x}_{k-1} - B_{k-1}^{-1} \cdot \Delta F_{k-1}}{\Delta\vec{x}_{k-1}^T B_{k-1}^{-1} \Delta F_{k-1}} \cdot \left((B_{k-1}^{-1})^T \Delta x_{k-1}^T \right)^T$$

So costs $O(n^2)$ per iteration, not $O(n^3)$.

In practice, you want to add a stepsize which you find via a linesearch. $\vec{x}^{(k+1)} = \vec{x}^{(k)} - \eta \cdot B_k^{-1} F(\vec{x}^{(k)})$

Since 1980's, variants of quasi-Newton with "limited memory" are just $O(n)$ cost and state-of-the-art performance (especially for minimization problems...

"BFGS" is most famous quasi-Newton method,

"L-BFGS" is limited memory variant)

See a textbook for implementation details

(choose B_0, choose stepsize η, etc.)

Nocedal & Wright (2004) "Numerical Optimization" is good