Newton's method

Let $f(x)$ be a function. Using Taylor series we can approximate

$$f_T(x) = f_T(x_n + \Delta x) = f(x_n) + f'(x_n)\Delta x + \frac{1}{2} f''(x_n)\Delta x^2$$

To find where $f_T(x_n + \Delta x)$ is minimal, we want to choose Δx that minimizes.

$$0 = \frac{\partial}{\partial \Delta x}\left(f(x_n) + f'(x_n)\Delta x + \frac{1}{2} f''(x_n)\Delta x^2\right)$$

$$= f'(x_n) + f''(x)\Delta x$$

$$\Delta x = -\frac{f'(x_n)}{f''(x)}$$

In higher dimensions... optimization problem example

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\sum_{i=1}^{m} f_i(x)^2 = \underset{x}{\text{minimize}} \ \frac{1}{2} F^T(x) F(x)$$

where $F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{bmatrix}$

$$\nabla f = \frac{\partial}{\partial x}\left(\frac{1}{2} f_1(x)^2 + \frac{1}{2} f_2(x)^2 \dots \frac{1}{2} f_m(x)^2\right) \quad \text{Chain Rule}$$

$$= f_1(x)\frac{\partial}{\partial x} f_1(x) + f_2(x)\frac{\partial}{\partial x} f_2(x) + \dots f_m(x)\frac{\partial}{\partial x} f_m(x)$$

$$= \nabla F(x)\, F(x)$$ $\leftarrow$ ∇F is row vector, F is column vector

$$\nabla^2 f(x) = \frac{\partial}{\partial x}\left(\nabla F(x)\, F(x)\right)$$

Product Rule

$f'g + g'f$

$$= \nabla^2 F(x)\, F(x) + \nabla F(x)\, \nabla F(x)^T$$

transpose due to difference in orientation of F & ΔF

$$\underbrace{\sum_{i=1}^{m} \nabla^2 f_i(x)\, f_i(x) + \nabla F(x) \nabla F(x)^T}_{\text{Hessian}}$$

Back to newtons method...

~~Jacobian~~ ~~Hessian~~

~~J(x)~~ ~~H(x)~~

$$0 = f'(x) + f''(x)\,\Delta x \sim \underbrace{\nabla f(x)}_{g} + \underbrace{\nabla^2 f(x)}_{H}\,\Delta x$$

$$\Rightarrow \cancel{J(x) + H(x)\,\Delta x = 0} \qquad -g(x) = H(x)\,\Delta x$$

$$H(x)\,\Delta x = -g(x)$$

← solve iteratively or w/ decomposition

However!...

Gauss-Newton when initial condition is close to optimal solution

$$H(x) = \sum_{i=1}^{m} \nabla^2 f_i(x)\, f_i(x) + \nabla F(x) \nabla F(x)^T$$

when $F(x) \sim F(x^*) = 0$ where x^* is optimal

$f_i(x) \sim 0$

thus. $H(x) \sim \nabla F(x)\, \nabla F(x)^T$ $\qquad \nabla F(x) = J^T(x)$

$\sim J^T(x)\, J(x)$, ... because $\nabla F(x)$ was defined as rows stacked together and

$$g(x) = \nabla F(x) F(x)$$
$$= J^T(x) b$$
$$H(x) \sim \nabla F(x) \nabla F(x)^T$$
$$\approx J^T(x) J(x)$$

$J^T(x) J(x) = -J^T(x) b$ ← This is the same as minimizing residuals

$F(x^*) = F(x) + \nabla F(x) \Delta$ where $\Delta = x^* - x$

$0 = F(x) + \nabla F(x) \Delta$ because we want residual to be 0

minimize $\| Ax + b \|_2^2$

$$-(A^T A)^{-1} A^T b = \Delta$$

$$-(J^T J)^{-1} J^T b = \Delta$$

$$x_{new} = x + \Delta$$

Finally.. never perform inverse.

Use
1) Cholesky Decomposition
2.) QR Factorization.

1) $A = M^T M$ where M^T is lower triangular

$M^T \underbrace{Mx}_{y} = b$

$M^T y = b$ via forward substitution

$Mx = y$ via backward substitution

$A = J^T J \quad b = -J^T F(x)$

2. QR decomposition

In $A = M^T M$ where $M = QR$ and $b = M^T c$ ← linear combination of basis M^T

$Q \in R^{m \times n}$ and $Q^T Q = I_{n \times n}$

R upper triangular

$$Ax = b$$

$$M^T M x = b = M^T c$$

$$R^T Q^T Q R x = R^T Q^T c$$

$$R^T R x = R^T Q c$$

$$Rx = Qc$$ solve via backward substitution