Derivative matrix

- Gradient matrix:

$$f: R^n \to R \Rightarrow \nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{x_n} \end{bmatrix} : R^n$$

- Hessian matrix:

$$f: \mathbb{R}^n \to \mathbb{R} \to H(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

- Jacobian matrix:

$$f: R^n \to R^m \Rightarrow J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} : m \times n$$

$$J_{ij} = \frac{\partial f_i}{\partial x_j}, \quad J = \left[\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n} \right]$$

I Unconstrained minimization

1. Method of steepest descent

⊕ $F(\lambda) = f(x' + \lambda u)$ - search for best direction

$$\left.\frac{df(x')}{d\lambda}\right|_{u} = \nabla^T f(x') u \geq -\|\nabla^T f(x)\| \|u\|$$

$$u = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$$

⊕ $u^{i+1^T} u^i = 0 \Rightarrow$ slow to converge due to zigzag pattern

At x^{i-1}, line search through u^i to give x^i

→ condition for optimal descent is for λ_i

$$\left.\frac{df(x^{i-1} + \lambda_i u^i)}{d\lambda_i}\right|_{u^i} = 0$$

$$\Rightarrow \nabla^T f(u^i) u^i = 0, \quad u^{i+1} = -\frac{\nabla f(x^i)}{\|\nabla f(x^i)\|}$$

⊕ convergence criteria:

- $\|x^{i} - x^{i-1}\| < \varepsilon_1$
- $\|\nabla f(x^i)\| \leq \varepsilon_2$
- $|f(x^i) - f(x^{i-1})| < \varepsilon_3$

pwc

⊕ suggicient for improvement: Armijo, curvature

2. Conjugate gradient methods

⊕ converge in a finite step for positive-definite quadratic function:

$$f(x) = \frac{1}{2} x^T A x + b^T x + c$$

⊕ $u^i A u^{i+1} = 0$

⊕ obvious choice: eigenvectors if A is symmetric

$$u^i A u^j = u^i \lambda u^j = \lambda u^i u^j = 0$$

eigenvectors of symmetric matrix are orthgornal

⊕ Fletcher-Roeves direction:

- $u^1 = -\nabla f(x^0)$
- $u^{i+1} = -\nabla f(x^i) + B_i u^i$

$$B_i = \frac{\| \nabla f(x^i) \|^2}{\| \nabla f(x^{i-1}) \|^2}$$

pwc

3. Second order line search method - Newton

⊕ Taylor's expansion at the optimal point x^*

$$x^* = x^i + u$$

$$f(x^*) = f(x^i) + \nabla^T f(x^i) u + \frac{1}{2} u^T H(x^i) u$$

$$\nabla f(x^*) = \nabla f(x^i) + H(x^i) u = 0$$

$$\Rightarrow u = -H(x^i)^{-1} \nabla f(x^i)$$

⊕ can set : $x^i = x^{i-1} + \lambda_i u^i$

⊕ compute H and H^{-1} is expensive → use an update scheme

→ use quasi-newton method such as:

- DFP
- BFGS

→ achieve quadratic termination.

~~5. Trust Region Methods~~

~~(*) Given a few methods~~

4. Gauss-Newton method for non-linear least-squares problem :

$$\min \quad g(x) = f_1^2(x) + \dots + f_m^2(x)$$

$$f(x) : R^n \to R^m$$

We apply the usual Newton's direction. $-H_g(x)\,\nabla g(x)$

$$\nabla g(x) = \begin{bmatrix} 2\frac{\partial f}{\partial x_1}^T f \\ \vdots \\ 2\frac{\partial f}{\partial x_n}^T f \end{bmatrix} = 2J_f^T(x)\,f(x)$$

$$\cancel{H_{ij}(x)} \quad g_j(x) = 2\sum_i f_i \frac{\partial f_i}{\partial x_j} \qquad i : 1, n \quad j : 1, .. n$$

$$\to H_{jk}(x) = 2\sum_i \left(f_i \frac{\partial^2 f_i}{\partial x_j \partial x_k} + \frac{\partial f_i}{\partial x_k}\frac{\partial f_i}{\partial x_j} \right)$$

$$\simeq 2\sum_i \frac{\partial f_i}{\partial x_k}\frac{\partial f_i}{\partial x_j}$$

$$\to H_{jk}(x) \simeq 2J^T J$$

$\to$ Gauss Newton Step $= -(2J^T J)^{-1}\, 2J^T f$

$$= -(J^T J)^{-1} J^T f$$

$\to$ No Need to compute Hessian

5. Levenberg- Marquadt algorithm for non-linear least square:

⊕ For ~~s~~starting point far from solution
⇒ GNA can be slow to converge especially is the quadratic assumption is strong.
→ Should use gradient - descent in this case:

⊕ Add a damping parameter:

$$-(J^T J + \lambda I)^{-1} J^T f$$

$\lambda \to 0 \Rightarrow$ GN update

$\lambda \uparrow \Rightarrow$ Gradient - descent update with smaller step size, ignore H effect

unacceptable step $\to \lambda$ is $\uparrow$ untill acceptable
follows the gradient more

⊕ Unity - step curve algorithm use case:

$$\min \sum_{j=1}^{m} W_j (\tilde{R}_j - R_j)^2 + \beta \sum_{i=1}^{n-2} \left[\frac{f(t_{i+1}, t_{i+2}) - f(t_i, t_{i+1})}{t_{i+2} - t_i} + \frac{f(t_i, t_{i+1}) - f(t_{i-1}, t_i)}{t_{i+1} - t_2} \right]^2$$

R_j: input rate of input instrutments

t_i: bucket points: MPCs

$f(t_i, t_{i+1})$: continuously compounded forward rate.

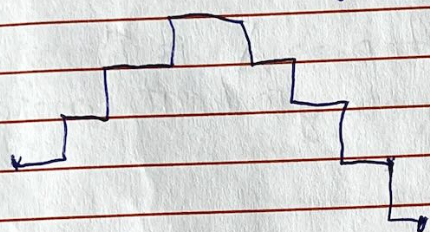
R_j and f_i are functions of the ln DF

$$f(t_i, t_{i+1}) = \frac{-\ln(D(t_{i+1}, D_{t_i}))}{t_{i+1} - t_i}$$

$$= \frac{-(\ln(D_{t_{i+1}}) - \ln D_{t_i})}{t_{i+1} - t_i}$$

$R_j <$ Fwd Rat / swap rate } solved from ln DF.

problem with quadratic penalty: prefer series of small steps size leading in wavy pattern



Approximate $|x|$ with $\sqrt{x^2+\varepsilon}$