Reference sheets

Forward difference:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}.$$

Backward difference:

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}.$$

Centered difference:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

Centered difference for f'':

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}.$$

Richardson extrapolation

Suppose

$$F(h) = a_0 + a_1 h^p + O(h^r),$$

as $h \to 0$ for some p and r, with $r > p$. Then

$$a_0 = F(h) + \frac{F(h) - F(h/q)}{q^{-p} - 1}.$$

Automatic differentiation: forward mode vs Reverse mode

Suppose $f : \mathbb{R}^n \to \mathbb{R}^m$. Forward mode is faster when m is much larger than n. Reverse mode is faster when n is much larger than m. Reverse mode takes a lot more space (except for some special cases).

Method of normal equations

$$A^T A x = A^T b.$$

Solving least-squares with QR factorization

$$A = QR.$$
$$c = Q^T b.$$
$$R_1 x = c_1.$$

Householder transformations

$$H = I - 2\frac{vv^T}{v^T v}.$$

Also,

$$Ha = \alpha e_1$$

if

$$v := a - \alpha e_1$$

and $\alpha = \pm \|a\|_2$.

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H'_k \end{bmatrix}.$$

Givens rotations

$$Ga = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix},$$

where

$$c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}, \qquad s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}.$$

If $|a_1| > |a_2|$,

$$t = \frac{a_2}{a_1}, \qquad c = \frac{1}{\sqrt{1+t^2}}, \qquad s = ct.$$

If $|a_2| > |a_1|$,

$$\tau = \frac{a_1}{a_2}, \qquad s = \frac{1}{\sqrt{1+\tau^2}}, \qquad c = s\tau.$$

Theorem 1. *Any $A \in \mathbb{R}^{m \times n} (m \geq n)$ can be factored $A = QR$ where Q is an $m \times m$ orthogonal matrix and R is an $m \times n$ upper triangular matrix.*

Theorem 2 (Bolzano's)**.** *If f is a continuous function on a closed interval $[a, b]$, and $f(a)$ and $f(b)$ differ in sign, then there must be a root within the interval $[a, b]$.*

Convergence rates of an iterative method

An iterative method converges with **rate** r if

$$\lim_{k \to \infty} \frac{\left\| e^{(k+1)} \right\|}{\left\| e^{(k)} \right\|^r} = C$$

for some finite constant $C > 0$.

Secant method

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

Systems of nonlinear equations

Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, its first derivative is the **Jacobian**, which is the m-by-n matrix

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

Newton's method for system of nonlinear equations

$$\begin{aligned} \nabla f(x^{(k)})h^{(k)} &= -f(x^{(k)}) \\ x^{(k+1)} &= x^{(k)} + h^{(k)} \end{aligned}$$

If the initial point $x^{(0)}$ is close enough to a simple root, Newton's method converges quadratically.

Newton's method generally does not converge to a singular root (i.e. a root x^* where $\nabla f(x^*) = 0$) or if it does, it does so very slowly (linear convergence at best).

Broyden's method

$$\begin{aligned} B^{(k)}h^{(k)} &= -f(x^{(k)}) \\ x^{(k+1)} &= x^{(k)} + h^{(k)} \\ y^{(k)} &= f(x^{(k+1)}) - f(x^{(k)}) \\ B^{(k+1)} &= B^{(k)} + \frac{\left(y^{(k)} - B^{(k)}h^{(k)}\right)\left(h^{(k)}\right)^T}{\left(h^{(k)}\right)^T h^{(k)}} \end{aligned}$$

Successive Parabolic Interpolation

The minimum of the parabola interpolating u, v, and w is given by

$$v - \frac{1}{2}\frac{(v-u)^2(f_v - f_w) - (v-w)^2(f_v - f_u)}{(v-u)(f_v - f_w) - (v-w)(f_v - f_u)}.$$

Derivatives of multivariate functions

Suppose $f : \mathbb{R}^n \to \mathbb{R}$. The **gradient** of f is the n-vector of partial derivatives

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}.$$

The **Hessian matrix** of $f : \mathbb{R}^n \to \mathbb{R}$ is the matrix

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

That is,

$$\left[\nabla^2 f(x)\right]_{ij} = \frac{\partial^2 f}{\partial x_j \partial x_i}.$$

Newton's method for unconstrained optimization

$$\begin{aligned} \left(\nabla^2 f(x^{(k)})\right) h^{(k)} &= -\nabla f(x^{(k)}) \\ x^{(k+1)} &= x^{(k)} + h^{(k)} \end{aligned}$$

BFGS method for unconstrained optimization

$$\begin{aligned} B^{(k)}h^{(k)} &= -\nabla f(x^{(k)}) \\ x^{(k+1)} &= x^{(k)} + h^{(k)} \\ y^{(k)} &= \nabla f(x^{(k+1)}) - \nabla f(x^{(k)}) \\ B^{(k+1)} &= B^{(k)} + \frac{y^{(k)}(y^{(k)})^T}{(y^{(k)})^T h^{(k)}} - \frac{B^{(k)}h^{(k)}(h^{(k)})^T B^{(k)}}{(h^{(k)})^T B^{(k)} h^{(k)}} \end{aligned}$$

Euler's method:

$$y_{k+1} = y_k + h_k f(t_k, y_k),$$

Adams-Bashforth second order (AB2):

$$y_{k+1} = y_k + \frac{3h}{2} f(t_k, y_k) - \frac{h}{2} f(t_{k-1}, y_{k-1}).$$

The second-order Runge-Kutta method or Heun's method

$$y_{k+1} = y_k + \frac{h}{2}(s_1 + s_2),$$

where

$$\begin{aligned} s_1 &= f(t_k, y_k), \\ s_2 &= f(t_k + h, y_k + hs_1). \end{aligned}$$

Stiffness

Model ODE:

$$\frac{dy}{dt} = ay, \quad a < 0.$$

Backward Euler:

$$y_{k+1} = y_k + hf(t_{k+1}, y_{k+1}).$$

The implicit trapezoid method:

$$y_{k+1} = y_k + h\left(\frac{f(t_k, y_k) + f(t_{k+1}, y_{k+1})}{2}\right).$$

Solving least-squares problems with SVD

The least squares solution to $Ax \cong b$ is

$$x = \sum_{\sigma_i \neq 0} \frac{u_i^T b}{\sigma_i} v_i,$$

where u_i is the ith column of U, v_i is the ith column of V.

Low-rank Approximation

Let $A \in \mathbb{R}^{m\times n}$ whose SVD is $A = U\Sigma V^T$. Let

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T.$$

Then $\text{rank}(A_k) \le k$ and

$$\begin{aligned} \|A - A_k\|_F &= \min\{\|A - B\|_F : B \in \mathbb{R}^{m\times n} \\ &\qquad \text{satisfying rank}(B) \le k\}. \end{aligned}$$

(Moore-Penrose) Pseudoinverse

- The pseudoinverse of a scalar σ is

$$\sigma^+ = \begin{cases} \frac{1}{\sigma} & \text{if } \sigma \ne 0, \\ 0 & \text{otherwise.} \end{cases}$$

- The pseudoinverse of a matrix $A \in \mathbb{R}^{m\times n}$, is given by

$$A^+ = V\Sigma^+ U^T,$$

  where $A = U\Sigma V^T$ is the SVD of A and Σ^+ is the $n \times m$ matrix with diagonals entries $\sigma_1^+, \sigma_2^+, \ldots$.

- $A^+ b$ is the least-squares solution to $Ax \cong b$.

Eigenvalues and Eigenvectors

Rayleigh quotient

$$\lambda = \frac{(x^{(k)})^T A x^{(k)}}{(x^{(k)})^T x^{(k)}}.$$

Theorem 3. *If $A \in \mathbb{R}^{n\times n}$ is invertible, then A and A^{-1} have the same eigenvectors, and the eigenvalues of A^{-1} are reciprocals of the eigenvalues of A (i.e. if λ is an eigenvalue of A, then $1/\lambda$ is an eigenvalue of A^{-1}).*

Inverse Shifted Power Method

$$x^{(k+1)} = (A - \sigma I)^{-1} x^{(k)}$$

Theorem 4. *Suppose $(A - \sigma I)$ is invertible. Then*

- *A and $(A - \sigma I)^{-1}$ have the same eigenvectors.*
- *λ is an eigenvalue of A if and only if $1/(\lambda - \sigma)$ is an eigenvalue of $(A - \sigma I)^{-1}$.*

QR iteration

1. $M^{(0)} = A$
2. For $k = 0, 1, 2, \ldots$
   - Factor $M^{(k)} = Q^{(k)} R^{(k)}$.
   - Define $M^{(k+1)} := R^{(k)} Q^{(k)}$.

Finite Difference Method for BVP

Mesh points $t_i = a + ih, i = 0, \ldots, n+1$, where $h = (b - a)/(n+1)$.

The Least-squares method for the scalar Poisson equation:

$$\sum_{j=1}^{n} \left(\int_a^b \phi_j''(t)\phi_i''(t)dt \right) x_j = \int_a^b f(t)\phi_i''(t)dt,$$

for $i = 1, \ldots, n$, which is $Ax = b$ where

$$a_{ij} = \int_a^b \phi_j''(t)\phi_i''(t)dt,$$

and

$$b_i = \int_a^b f(t)\phi_i''(t)dt.$$

The Galerkin method for the scalar Poisson equation:

$$-\sum_{j=1}^{n} \left(\int_a^b \phi_j'(t)\phi_i'(t)dt \right) x_j = \int_a^b f(t)\phi_i(t)dt,$$

for i satisfying $\phi_i(a) = \phi_i(b) = 0$, which is $Ax = b$ where

$$a_{ij} = -\int_a^b \phi_j'(t)\phi_i'(t)dt,$$

and

$$b_i = \int_a^b f(t)\phi_i(t)dt.$$

PDE

Common spatial mesh points: $x_i = i\Delta x$, $i = 0, \ldots, n+1$, where $\Delta x = 1/(n+1)$.

Common temporal mesh points: $t_k = k\Delta t$, $k = 0, 1, \ldots$.

Crank-Nicolson method

$$\begin{aligned} u_i^{k+1} &= u_i^k + c\frac{\Delta t}{(\Delta x)^2}\left(u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}\right. \\ &\qquad \left. + u_{i+1}^k - 2u_i^k + u_{i-1}^k\right), \end{aligned}$$

Simulated Annealing

A typical acceptance probabiity for simulated annealing is

$$p(k, f(z^{(k)}), f(x^{(k)})) = \min\left\{1, e^{\frac{-\left(f(z^{(k)}) - f(x^{(k)})\right)}{T_k}}\right\}.$$

Hajek cooling schedule is

$$T_k = \frac{\gamma}{\log(k+2)},$$

where $\gamma > 0$.

Particle swarm optimization (PSO)

$$\begin{aligned} x_i^{(k+1)} &= x_i^{(k)} + v_i^{(k)} \\ v_i^{(k+1)} &= \omega v_i^{(k)} + c_1 r_i^{(k)} \circ (p_i^{(k)} - x_i^{(k)}) + c_2 s_i^{(k)} \circ (g^{(k)} - x_i^{(k)}), \end{aligned}$$

where $r_i^{(k)}$ and $s_i^{(k)}$ are random n-vectors with entries uniformly in the interval $(0, 1)$ and ω, c_1, and c_2 are constants.

Nelder-Mead

The centroid p_g is computed by

$$p_g = \sum_{i=0}^{n-1} \frac{p_i}{n}.$$

Reflection: $p_r = p_g + \rho(p_g - p_n)$.
Expansion: $p_e = p_g + \chi(p_r - p_g)$.
Outside contraction: $p_c = p_g + \gamma(p_r - p_g)$.
Inside contraction: $p_c = p_g + \gamma(p_n - p_g)$.
Shrinkage: $v_i = p_0 + \sigma(p_i - p_0)$.

Jacobi Method

$$x_i^{(k+1)} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}}{a_{ii}}.$$

Gauss-Seidel Method

$$x_i^{(k+1)} = \frac{b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)}}{a_{ii}}.$$

Successive Over-Relaxation (SOR)

$$x^{(k+1)} = x^{(k)} + \omega \left(x_{GS}^{(k+1)} - x^{(k)} \right),$$

where $x_{GS}^{(k+1)}$ is the Gauss-Seidel step.