Uppsala universitet
Institutionen för informationsteknologi

Selected formulas in Scientific Computing

1. Floating point numbers and roundoff errors

A floating point number $fl(x)$ is represented as

$$fl(x) = \hat{m} \cdot \beta^e, \quad \hat{m} = \pm(d_0.d_1d_2 \ldots d_{p-1}), \quad 0 \leq d_k < \beta, \quad d_0 \neq 0, \quad L \leq e \leq U,$$

where β denotes the basis and p precision.
A floating point system is defined by $\mathbb{F}(\beta, p, L, U)$.
Machine epsilon (unit roundoff) $\epsilon_M = \frac{1}{2}\beta^{1-p}$ can be defined as the smallest number such that $fl(1 + \epsilon_M) > 1$. In double precision $\epsilon_M \approx 10^{-16}$.

2. Linear and non-linear equations

Newton-Raphsons method: $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
For systems: $x_{k+1} = x_k - [F']^{-1}F(x_k)$ with x_k and $F(x_k)$ being vectors and F' being a Jacobian.
Fixed-point iteration for $x = g(x) : x_{k+1} = g(x_k)$
Convergence ratio, convergence order

$$\lim_{k \to \infty} \frac{|x_{k+1} - x_*|}{|x_k - x_*|^r} = C$$

with C being a constant, and r denotes the order of convergence ($r = 1$ means, for example, linear convergence).
General error estimate

$$|x_k - x_*| \leq \frac{|f(x_k)|}{\min|f'(x)|}$$

3. Norms (vector and matrix norms) and condition number

$$\|x\|_2 = \sqrt{|x_1|^2 + \ldots + |x_n|^2} \quad \|x\|_1 = \textstyle\sum_i |x_i| \quad \|x\|_\infty = \max_i\{|x_i|\}$$
$$\|A\|_1 = \max_j(\textstyle\sum_i |a_{ij}|) \quad \|A\|_\infty = \max_i(\textstyle\sum_j |a_{ij}|) \quad \|A\|_2 = \sqrt{\lambda_{\max}(A^TA)} = \sigma_1$$

$\mathrm{cond}_2(A) = \|A\|_2 \cdot \|A^+\|_2 = \frac{\sigma_1}{\sigma_r}$, where r is the rank of A. If A square and non-singular: $A^+ = A^{-1}$.
Condition number $\mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\|$ (for A square and non-singular) measures the sensitivity to disturbances in the system of equations $Ax = b$. There holds

$$\frac{\|\Delta x\|}{\|x\|} \leq \mathrm{cond}(A)\frac{\|\Delta b\|}{\|b\|}$$

with $\Delta x = x - \hat{x}$ and $\Delta b = b - \hat{b}$.

4. Taylor expansion

Taylor expansion of $y(x_k+h)$ around the point x_k:
$y(x_k+h) = y(x_k) + hy'(x_k) + \frac{h^2}{2!}y''(x_k) + \frac{h^3}{3!}y'''(x_k) + \mathcal{O}(h^4)$

5. Ordinary differential equations

ODE, initial value problem: $\begin{cases} y'(t) = f(t, y(t)) \ , \quad t \geq a \\ y(a) = \alpha \end{cases}$

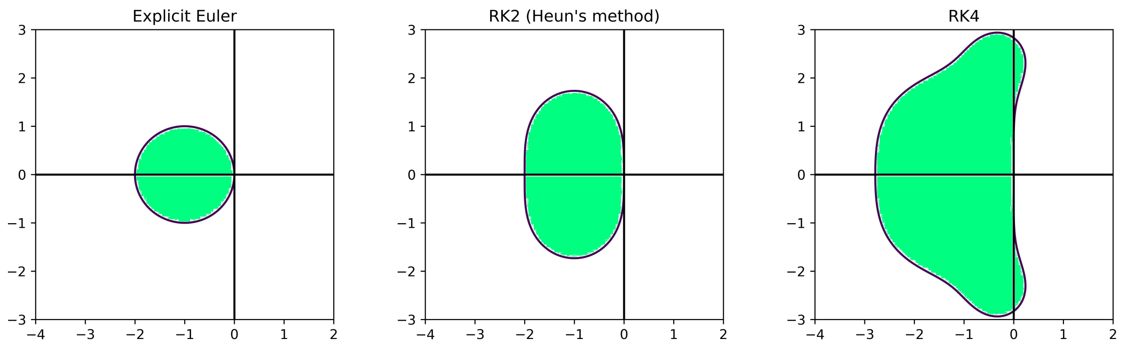
Euler's method (explicit Euler): $y_{i+1} = y_i + hf(t_i, y_i)$, order of accuracy: 1
Implicit Euler (Euler backward): $y_{i+1} = y_i + hf(t_{i+1}, y_{i+1})$, order of accuracy: 1
Trapezoidal rule (for ODEs): $y_{i+1} = y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_{i+1}))$, order of accuracy: 2
Heun's method (a 2-stage Runge-Kutta method):

$$\begin{cases} K_1 = f(t_i, y_i) \\ K_2 = f(t_{i+1}, y_i + hK_1) \\ y_{i+1} = y_i + \frac{h}{2}(K_1 + K_2) \quad \text{Order of accuracy: 2} \end{cases}$$

Classical Runge-Kutta (a 4-stage RK-method:)

$$\begin{cases} K_1 = f(t_i, y_i) \\ K_2 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}K_1) \\ K_3 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}K_2) \\ K_4 = f(t_{i+1}, y_i + hK_3) \\ y_{i+1} = y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad \text{Order of accuracy: 4} \end{cases}$$

The stability regions for explicit Euler, Heun's method, and RK4 are given by:



The stability intervals are given by

- Explicit Euler: SI $= [-2, 0]$
- Heun's method: SI $= [-2, 0]$
- RK4: SI $\approx [-2.78, 0]$

The stability region for implicit Euler and the Trapezoidal rule (as given above) includes the entire left complex plane, i.e., it includes $\{z = \text{Re}(z) + i\text{Im}(z) \mid \text{Re}(z) \leq 0\}$.

6. Numerical integration

Trapezoidal rule
Computation on one sub interval, step length $h_k = x_{k+1} - x_k$

$$\int_{x_k}^{x_{k+1}} f(x)\,dx \approx \frac{h_k}{2}[f(x_k) + f(x_{k+1})]$$

Composite formula on whole integration interval $[a\ , b]$ and equidistant step length $h = h_k$:

$$\int_a^b f(x)\,dx \approx \frac{h}{2}[f(x_0) + 2f(x_1) + \ldots + 2f(x_{N-1}) + f(x_N)]$$

Discretization error R on whole integration interval $[a\ ,\ b]$, i.e. $\int_a^b f(x)\,dx = T(h) + R$ is

$$R = -\frac{(b-a)}{12}h^2 f''(\xi) \text{ or } \mathcal{O}(h^2).$$

The function error (upper limit): $(b-a)\cdot\epsilon$, where ϵ is an upper limit for the absolute error in each function evaluation $f(x_k)$.

Simpson's rule
Computation on one double interval, step length h

$$\int_{x_k}^{x_{k+2}} f(x)\,dx \approx \frac{h}{3}[f(x_k) + 4f(x_{k+1}) + f(x_{k+2})]$$

Composite formula on whole integration interval $[a\ b]$ and equidistant step length $h = h_k$:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)]$$

Discretization error R on the integration interval $[a\ ,\ b]$, i.e, $\int_a^b f(x)\,dx = S(h) + R$ is

$$R = -\frac{(b-a)}{180}h^4 f''''(\xi) \text{ or } \mathcal{O}(h^4).$$

Function error: same as for trapezoidal rule, see above.

7. Richardson extrapolation

If $Q(h)$ och $Q(2h)$ are two computations (e.g. computation of an integral or an ODE) using a method with order of accuracy p, step length h and double step length $2h$, then

$$R(h) = \frac{Q(h) - Q(2h)}{2^p - 1}$$

is an estimate of the leading term in the discretization (truncation) error in $Q(h)$.
This can also be used to improve the accuracy in $Q(h)$ by

$$\tilde{Q}(h) = Q(h) + \frac{Q(h) - Q(2h)}{2^p - 1}.$$

8. Numerical differentiation

For numerical differentiation so called difference formulas are used

$$
\begin{array}{ll}
f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}, & \text{central difference} \\
f'(x) \approx \frac{f(x+h)-f(x)}{h}, & \text{forward difference} \\
f'(x) \approx \frac{f(x)-f(x-h)}{h}, & \text{backward difference} \\
f''(x) \approx \frac{f(x+h)-2f(x)+f(x-h)}{h^2} &
\end{array}
$$

9. Monte Carlo methods

Some well-known distributions:
- Uniform distribution on $[a,b]$, $\mathcal{U}(a,b)$, with probability density function (pdf):

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a,b] \\ 0, & \text{o.w.} \end{cases}$$

- Exponential distribution $\mathcal{E}xp(\lambda)$ on $[0,\infty)$ with pdf

$$f(x) = \lambda \exp(-\lambda x), \quad \lambda > 0.$$

- Normal distribution $\mathcal{N}(\mu,\sigma^2)$ on $(-\infty,\infty)$ with pdf

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

with mean μ and variance σ^2.

Cumulative distribution function (cdf): $F(x) = \int_{-\infty}^{x} f(y)\, dy$, where $f(y)$ is the probability density function (pdf).

The general structure of a Monte Carlo simulation is

```
Input N (number of trials)
for i = 1:N
    Perform one stochastic simulation
    result(i) = the result of the simulation
end
Final result through some statistical calculation, such as
the mean of the result vector
```

Order of accuracy for Monte Carlo methods is $\mathcal{O}(\frac{1}{\sqrt{N}})$, where N is the number of samples. A 95% confidence interval: $|e| \leq 1.96\frac{\sigma}{\sqrt{N}}$, where N is the number of independent Monte Carlo simulations, σ is the standard deviation.

10. Regression and data analysis

Least squares problem: $\min_x \|b - Ax\|_2^2$
Normal equations: $A^T A x = A^T b$

QR factorization: $A = QR$ where Q orthogonal and R upper triangular.

Least squares solution via QR: $R_1 x = Q_1^T b$ (where R_1 and Q_1 are reduced forms). The residual is $r = \|Q_2^T b\|_2$.

Householder matrix: for $u \in \mathbb{R}^n$, $H = I - \frac{2}{u^T u} u u^T$

11. SVD and eigenvalues

SVD: $A = U \Sigma V^T$, left singular vectors in U, right singular vectors in V, singular values $\sigma_1, \ldots, \sigma_n$ on diagonal of Σ, with U and V orthogonal.

Pseudoinverse (Moore-Penrose): $A^+ = V_1 \Sigma_1^+ U_1^T$, where $\Sigma_1^+ = (\frac{1}{\text{non-zero elements in } \Sigma})^T$, and U_1 and V_1 are reduced forms.
Least squares solution: $x = A^+ b$ (norm-minimal solution when A is rank deficient), and $residual = \|U_2^T b\|_2$

Power method for the dominant eigenpair: $v = Av^{(k)},\ v^{(k+1)} = v / \|v\|_2,\ k = 0, 1, \ldots$
Rayleigh quotient: $\lambda = \frac{v^T A v}{v^T v}$, v an eigenvector of A. Replace v by $v^{(k+1)}$ to approximate λ in each step.
Order of convergence of power method: $e_k = \mathcal{O}\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right)$, k large.

QR iteration: Start with $A_0 = A$, for $k = 0, 1, \ldots$
(a) determine Q, R such that $A_k = QR$
(b) compute new $A_{k+1} = RQ$