Mathematics 5.

Teemu Weckroth, September 7, 2023.

Landau's $\mathcal{O}$-symbol.

Let f and g be given functions. If there exists a positive r and a positive, finite M such that

$$|f(x)| \leq M|g(x)| \text{ for all } x \in [-r, r],$$

we say that $f(x)$ is of order $g(x)$ for x going to zero. Symbolically we write this as

$$f(x) = \mathcal{O}(g(x)) \text{ for } x \to 0.$$

If $f(x) = \mathcal{O}(x^p)$ and $g(x) = \mathcal{O}(x^q)$ for $x \to 0$ with $p \geq 0, q \geq 0$, then

- $f(x) = \mathcal{O}(x^s)$ for $x \to 0$ with $0 \leq s \leq p$
- $\alpha f(x) + \beta g(x) = \mathcal{O}(x^r)$ for $x \to 0$ with $r = \min\{p, q\}$ for all $\alpha, \beta \in \mathbb{R}$
- $f(x)g(x) = \mathcal{O}(x^{p+q})$ for $x \to 0$
- $\frac{f(x)}{x^t} = \mathcal{O}(x^{p-t})$ for $x \to 0$ if $0 \leq t \leq p$

Taylor expansions.

Assume $f(x) \in C^{n+1}[a, b]$ is given. Then for all $c, x \in (a, b)$ there exists a number ξ between c and x such that

$$f(x) = T_n(x) + R_n(x),$$

in which

$$T_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(c)}{k!}(x - c)^k,$$

and

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - c)^{n+1} = \mathcal{O}((x - c)^{n+1}).$$

Lagrange interpolation - linear.

Assume two data points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ of a function f are given. The linear Lagrange basis polynomials $L_{i1},\ i \in \{0, 1\}$ are defined by

$$L_{01} = \frac{x - x_1}{x_0 - x_1} \text{ and } L_{11} = \frac{x - x_0}{x_1 - x_0},$$

and the linear Lagrange interpolation polynomial L_1 by

$$L_1(x) = f(x_0)L_{01}(x) + f(x_1)L_{11}(x).$$

Assume $f \in C^2[a, b]$ with $x_0, x_1 \in [a, b]$. Then for each $x \in [a, b]$ there exists a $\xi \in (a, b)$ such that

$$f(x) - L_1(x) = \frac{1}{2}(x - x_0)(x - x_1)f''(\xi).$$

Furthermore it holds that

$$|f(x) - L_1(x)| \leq \frac{1}{8}(x_1 - x_0)^2 \max_{\gamma \in (a,b)} |f''(\gamma)|.$$

Assume two distinct measured data points $(x_0, \hat{f}(x_0))$ and $(x_1, \hat{f}(x_1))$ of a function $f \in C^2[a, b]$ are given with $x_0, x_1 \in [a, b]$. Also assume $|f(x_0) - \hat{f}(x_0)| \leq \varepsilon$ and $|f(x_1) - \hat{f}(x_1)| \leq \varepsilon$ and $\hat{L}_1$ is the linear Lagrange interpolation polynomial using the measured values. Then for all $x \in [a, b]$ it holds that

$$|L_1(x) - \hat{L}_1(x)| \leq \frac{|x_1 - x| + |x - x_0|}{|x_1 - x_0|}\varepsilon.$$

Lagrange interpolation - general.

Assume $n + 1$ data points $(x_0, f(x_0))$, $(x_1, f(x_1))$, ..., $(x_n, f(x_n))$ of a function f are given. The Lagrange basis polynomials L_{kn}, $k \in \{0, 1, \dots, n\}$ are defined by

$$L_{kn} = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j},$$

and the n^{th} degree Lagrange interpolation polynomial L_n by

$$L_n(x) = \sum_{k=0}^{n} f(x_k)L_{kn}(x).$$

Assume $f \in C^{n+1}[a, b]$ with $x_0, \dots, x_n \in [a, b]$. Then for each $x \in [a, b]$ there exists a $\xi \in (a, b)$ such that

$$f(x) - L_n(x) = \frac{1}{(n+1)!}\prod_{k=0}^{n}(x - x_k)f^{(n+1)}(\xi).$$

Furthermore it holds that

$$|L_n(x) - \hat{L}_n(x)| \leq \sum_{k=0}^{n} |L_{kn}(x)|\varepsilon,$$

with ε an upper bound for measurement errors.

Spline interpolation.

Consider nodes $a = x_0 < x_1 < \cdots < x_n = b$ and a function $f \in C[a, b]$. A spline of degree p is a piecewise polynomial $s \in C^{p-1}[a, b]$ that assumes the values of f in the nodes, and is defined by

$$s(x) = \begin{cases} s_0(x), & x \in [x_0, x_1), \\ s_1(x), & x \in [x_1, x_2), \\ \vdots & \vdots \\ s_{n-1}(x), & x \in [x_{n-1}, x_n], \end{cases}$$

where s_k, $k \in \{0, 1, \dots, n-1\}$ are polynomials of degree p.

Numerical integration.

Assume $f \in C[x_L, x_R]$ and $x_M = \frac{1}{2}(x_L + x_R)$. We can define the following integration rules:

left rectangle rule
$$\int_{x_L}^{x_R} f(x)\,\mathrm{d}x \approx (x_R - x_L)f(x_L)$$

right rectangle rule
$$\int_{x_L}^{x_R} f(x)\,\mathrm{d}x \approx (x_R - x_L)f(x_R)$$

midpoint rule
$$\int_{x_L}^{x_R} f(x)\,\mathrm{d}x \approx (x_R - x_L)f(x_M)$$

trapezoidal rule
$$\int_{x_L}^{x_R} f(x)\,\mathrm{d}x \approx \frac{1}{2}(x_R - x_L)(f(x_L) + f(x_R))$$

Simpson's rule
$$\int_{x_L}^{x_R} f(x)\,\mathrm{d}x \approx \frac{1}{6}(x_R - x_L)(f(x_L) + 4f(x_M) + f(x_R))$$

Numerical integration - accuracy.

Assume $f \in C^q[x_L, x_R]$, $x_M = \frac{1}{2}(x_L + x_R)$ and $m_q = \max_{x \in [x_L, x_R]} \left|f^{(q)}(x)\right|$.

Then it holds that:

$$\left|\int_{x_L}^{x_R} f(x)\,\mathrm{d}x - (x_R - x_L)f(x_L)\right| \leq \frac{1}{2}m_1(x_R - x_L)^2$$

$$\left|\int_{x_L}^{x_R} f(x)\,\mathrm{d}x - (x_R - x_L)f(x_R)\right| \leq \frac{1}{2}m_1(x_R - x_L)^2$$

$$\left|\int_{x_L}^{x_R} f(x)\,\mathrm{d}x - (x_R - x_L)f(x_M)\right| \leq \frac{1}{24}m_2(x_R - x_L)^3$$

$$\left|\int_{x_L}^{x_R} f(x)\,\mathrm{d}x - \frac{1}{2}(x_R - x_L)(f(x_L) + f(x_R))\right| \leq \frac{1}{12}m_2(x_R - x_L)^3$$

$$\left|\int_{x_L}^{x_R} f(x)\,\mathrm{d}x - \frac{1}{6}(x_R - x_L)(f(x_L) + 4f(x_M) + f(x_R))\right| \leq \frac{1}{2880}m_4(x_R - x_L)^5$$

Composite numerical integration.

Assume $f \in C[a, b]$ and $a = x_0 < x_1 < \cdots < x_n = b$ are given nodes. If I_k is an integration rule that approximates $\int_{x_{k-1}}^{x_k} f(x)\,\mathrm{d}x$, we can approximate $\int_a^b f(x)\,\mathrm{d}x$ with the composite integration rule

$$I = \sum_{k=1}^{n} I_k.$$

Composite numerical integration - accuracy.

Assume f is defined on $[a, b]$ and $a = x_0 < x_1 < \cdots < x_n = b$ with $x_k = a + kh$ and $h = \frac{b-a}{n}$. Also assume I is a composite integration rule based on the integration rule I_k for which we know $|\int_{x_{k-1}}^{x_k} f(x)\,\mathrm{d}x - I_k| \leq c_k h^{p+1}$. Then

$$\left|\int_a^b f(x)\,\mathrm{d}x - I\right| \leq c(b - a)h^p,$$

where $c = \max\{c_1, c_2, \dots, c_n\}$.

Composite numerical integration - errors.

Assume I is a composite integration rule based on the function f and $\hat{I}$ is the same composite integration rule based on the function $\hat{f}$ containing measurement and/or rounding error for which we know $|f(x) - \hat{f}(x)| = \varepsilon(x) \leq \varepsilon_{\max}$. Then

$$\left|\int_a^b f(x)\,\mathrm{d}x - \hat{I}\right| \leq \left|\int_a^b f(x)\,\mathrm{d}x - I\right| + \left|I - \hat{I}\right|.$$

Initial-value problems.

A first-order initial-value problem for a scalar function y of the independent variable t is a system of the form

$$\begin{cases} y' = f(t,y), & \text{for } t \geq t_0, \\ y(t_0) = y_0, \end{cases}$$

where f is a known function and y_0 is a known value.

Initial-value problems - stability.

Consider the two first-order initial-value problems

$$\begin{cases} y' = f(t,y), & t \geq t_0, \\ y(t_0) = y_0, \end{cases} \text{ and } \begin{cases} \tilde{y}' = f(t,\tilde{y}), & t \geq t_0, \\ \tilde{y}(t_0) = y_0 + \varepsilon_0, \end{cases}$$

where f is a known function and y_0 and ε_0 are known values. Define $\varepsilon(t) = \tilde{y}(t) - y(t)$. The left initial-value problem is stable if

$$|\varepsilon(t)| < \infty \text{ for all } t \geq t_0,$$

and absolutely stable if it is stable and

$$\lim_{t \to \infty} |\varepsilon(t)| = 0.$$

Define $\lambda = \frac{\partial f}{\partial y}(\hat{t}, \hat{y})$. The left initial-value problem is stable for typical value of $\hat{t}$ and $\hat{y}$ if $\lambda \leq 0$ and absolutely stable if $\lambda < 0$.

Test equation.

The differential equation

$$y' = \lambda y$$

is called the test equation.

Forward Euler method.

Consider the initial-value problem

$$\begin{cases} y' = f(t,y), & \text{for } t \geq t_0, \\ y(t_0) = y_0. \end{cases}$$

Define w_n as the numerical approximation of $y(t_n)$. Then the Forward Euler method is defined by the discrete dynamical system

$$\begin{cases} w_{n+1} = w_n + \Delta t f(t_n, w_n), & \text{for } n \geq 0, \\ w_0 = y_0, \end{cases}$$

where $t_n = t_0 + n\Delta t$.

Local truncation error $\tau_{n+1} = \mathcal{O}(\Delta t)$.

Stability attained if $\Delta t \leq -\frac{2}{\lambda}$.

Backward Euler method.

Consider the initial-value problem

$$\begin{cases} y' = f(t,y), & \text{for } t \geq t_0, \\ y(t_0) = y_0. \end{cases}$$

Define w_n as the numerical approximation of $y(t_n)$. Then the Backward Euler method is defined by the discrete dynamical system

$$\begin{cases} w_{n+1} = w_n + \Delta t f(t_{n+1}, w_{n+1}), & \text{for } n \geq 0, \\ w_0 = y_0, \end{cases}$$

where $t_n = t_0 + n\Delta t$.

Local truncation error $\tau_{n+1} = \mathcal{O}(\Delta t)$.

Stability attained for all Δt.

Trapezoidal method.

Consider the initial-value problem

$$\begin{cases} y' = f(t,y), & \text{for } t \geq t_0, \\ y(t_0) = y_0. \end{cases}$$

Define w_n as the numerical approximation of $y(t_n)$. Then the Trapezoidal method is defined by the discrete dynamical system

$$\begin{cases} w_{n+1} = w_n + \frac{1}{2}\Delta t(f(t_n, w_n) + f(t_{n+1}, w_{n+1})), & \text{for } n \geq 0, \\ w_0 = y_0, \end{cases}$$

where $t_n = t_0 + n\Delta t$.

Local truncation error $\tau_{n+1} = \mathcal{O}(\Delta t^2)$.

Stability attained for all Δt.

Modified Euler method.

Consider the initial-value problem

$$\begin{cases} y' = f(t,y), & \text{for } t \geq t_0, \\ y(t_0) = y_0. \end{cases}$$

Define w_n as the numerical approximation of $y(t_n)$. Then the Modified Euler method is defined by the discrete dynamical system

$$\begin{cases} \overline{w}_{n+1} = w_n + \Delta t f(t_n, w_n), & \text{for } n \geq 0, \\ w_{n+1} = w_n + \frac{1}{2}\Delta t(f(t_n, w_n) + f(t_{n+1}, \overline{w}_{n+1})), & \text{for } n \geq 0, \\ w_0 = y_0, \end{cases}$$

where $t_n = t_0 + n\Delta t$.

Local truncation error $\tau_{n+1} = \mathcal{O}(\Delta t^2)$.

Stability attained if $\Delta t \leq -\frac{2}{\lambda}$.

RK4 method.

Consider the initial-value problem

$$\begin{cases} y' = f(t,y), & \text{for } t \geq t_0, \\ y(t_0) = y_0. \end{cases}$$

Define w_n as the numerical approximation of $y(t_n)$. Then the Fourth-order method of Runge-Kutta (RK4 method) is defined by the discrete dynamical system

$$\begin{cases} k_1 = \Delta t f(t_n, w_n), & \text{for } n \geq 0, \\ k_2 = \Delta t f(t_n + \frac{1}{2}\Delta t, w_n + \frac{1}{2}k_1), & \text{for } n \geq 0, \\ k_3 = \Delta t f(t_n + \frac{1}{2}\Delta t, w_n + \frac{1}{2}k_2), & \text{for } n \geq 0, \\ k_4 = \Delta t f(t_n + \Delta t, w_n + k_3), & \text{for } n \geq 0, \\ w_{n+1} = w_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), & \text{for } n \geq 0, \end{cases}$$

where $t_n = t_0 + n\Delta t$ and $w_0 = y_0$.

Local truncation error $\tau_{n+1} = \mathcal{O}(\Delta t^4)$.

Stability attained if $\Delta t \leq \frac{2.8}{|\lambda|}$.

Numerical stability.

Consider the two first-order initial-value problems

$$\begin{cases} y' = f(t,y), & t \geq t_0, \\ y(t_0) = y_0, \end{cases} \text{ and } \begin{cases} \tilde{y}' = f(t,\tilde{y}), & t \geq t_0, \\ \tilde{y}(t_0) = y_0 + \varepsilon_0, \end{cases}$$

where f is a known function and y_0 and ε_0 are known values. Assume a time-integration method has been applied to both initial-value problems, leading to the numerical solutions $w_n, n = 0, 1, \ldots$ and $\tilde{w}_n, n = 0, 1, \ldots$. Define $\bar{\varepsilon}_n = \tilde{w}_n - w_n$. The used time-integration method applied to the left initial-value problem is (numerically) stable if

$$|\bar{\varepsilon}_n| < \infty \text{ for all } n \geq 0.$$

Amplification factor Q.

The amplification factor Q of a time-integration method is defined by applying the time-integration method to the test equation. It satisfies the equation

$$w_{n+1} = Q(\lambda\Delta t) w_n.$$

forward Euler	$Q(\lambda\Delta t) = 1 + \lambda\Delta t$
backward Euler	$Q(\lambda\Delta t) = \frac{1}{1 - \lambda\Delta t}$
trapezoidal	$Q(\lambda\Delta t) = \frac{1 + \frac{1}{2}\lambda\Delta t}{1 - \frac{1}{2}\lambda\Delta t}$
modified Euler	$Q(\lambda\Delta t) = 1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2$
RK4	$Q(\lambda\Delta t) = 1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2 + \frac{1}{6}(\lambda\Delta t)^3 + \frac{1}{24}(\lambda\Delta t)^4$

Numerical stability - amplification factor.

Consider a time-integration method with amplification factor Q applied to the initial-value problem

$$\begin{cases} y' = f(t,y), & t \geq t_0, \\ y(t_0) = y_0, \end{cases}$$

and define $\lambda = \frac{\partial f}{\partial y}(\hat{t}, \hat{y})$. The chosen time-integration method applied to the initial-value problem is stable if

$$|Q(\lambda\Delta t)| \leq 1$$

for typical values of $\hat{t}$ and $\hat{y}$.

Local truncation error τ.

Given is that $y_{n+1} = y(t_{n+1})$ is the exact solution and that z_{n+1} is the numerical approximation of y_{n+1} with a chosen numerical time-integration method with starting point $y_n = y(t_n)$ and time step Δt. The local truncation error at time step $n+1$, τ_{n+1}, is then defined as

$$\tau_{n+1} = \frac{y_{n+1} - z_{n+1}}{\Delta t}.$$

To obtain a formula for the local truncation error the following steps can be performed:

1. Formulate a Taylor expansion of y_{n+1} around $\Delta t = 0$;
2. Formulate a Taylor expansion of z_{n+1} around $\Delta t = 0$;
3. Substitute 1. and 2. into the definition;
4. Simplify.

Given is a time-integration method with amplification factor Q. Then the local truncation error for this method applied to the test equation $y' = \lambda y$ is given by

$$\tau_{n+1} = \frac{e^{\lambda \Delta t} - Q(\lambda \Delta t)}{\Delta t} y(t_n).$$

To obtain a formula for the local truncation error for the test equation the following steps can be performed:

1. Formulate a Taylor expansion of $e^{\lambda \Delta t}$ around $\Delta t = 0$;
2. Formulate a Taylor expansion of $Q(\lambda \Delta t)$ around $\Delta t = 0$;
3. Substitute 1. and 2. into the equation above;
4. Simplify.

Global truncation error e.

Given is that $y_{n+1} = y(t_{n+1})$ is the exact solution and that w_{n+1} is the numerical approximation of y_{n+1} with a chosen solution time-integration method with time step Δt after $n+1$ time steps. The global truncation error at time t_{n+1}, e_{n+1}, is then defined as

$$e_{n+1} = y_{n+1} - w_{n+1}.$$

Consistency and convergence.

A numerical time-integration method is called consistent if

$$\lim_{\Delta t \to 0} \tau_{n+1} = 0,$$

where $T = (n+1)\Delta t$ is a fixed time. If it is consistent and $\tau_{n+1} = \mathcal{O}(\Delta t^p)$, we call the method consistent of order p.
A numerical time-integration method is called convergent if

$$\lim_{\Delta t \to 0} e_{n+1} = 0,$$

where $T = (n+1)\Delta t$ is a fixed time. If it is convergent and $e_{n+1} = \mathcal{O}(\Delta t^p)$, we call the method convergent of order p.
If a numerical method is both stable and consistent, it is convergent. In that case the global and local truncation errors are of the same order.

Error estimation.

Define $w_N^{\Delta t}$ as the numerical approximation of $y(t)$ using N time steps of size Δt and $e(t, \Delta t) = y(t) - w_N^{\Delta t}$ as the global truncation error at time t using time steps of size Δt.
Consider a time-integration method that is convergent of order p and assume Δt is small enough. Then the global truncation error can be estimated by

$$e(t, \Delta t) = c_p(t) \Delta t^p = \frac{w_N^{\Delta t} - w_{N/2}^{2\Delta t}}{2^p - 1}.$$

Consider a time-integration method that is convergent, but the order p is unknown and assume Δt is small enough. Then the order of the global truncation error can be estimated from

$$2^p = \frac{w_{N/2}^{2\Delta t} - w_{N/4}^{4\Delta t}}{w_n^{\Delta t} - w_{N/2}^{2\Delta t}}.$$

Assume $Q(h)$ is an approximation of an unknown value M. For sufficiently small h the truncation error $M - Q(h)$ can be estimated by

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

if p is known. If p is not known, it can be estimated from

$$2^p = \frac{Q(2h) - Q(4h)}{Q(h) - Q(2h)}.$$

Stability of systems of differential equations.

Consider a vector-valued initial-value problem

$$\begin{cases} \mathbf{y}' = A\mathbf{y} + \mathbf{g}(t), & t \geq t_0, \\ \mathbf{y}(t_0) = \mathbf{y}_0, \end{cases}$$

with A an $n \times n$-matrix. Define λ_j, $j = 1, 2, \dots, n$ as the eigenvalues of A, repeated according to their algebraic multiplicity. The system is stable if

$$\mathrm{Re}(\lambda_j) \leq 0$$

for $j = 1, 2, \dots, n$.
Consider the vector-valued initial-value problem

$$\begin{cases} \mathbf{y}' = \mathbf{f}(t, \mathbf{y}), & t \geq t_0, \\ \mathbf{y}(t_0) = \mathbf{y}_0, \end{cases}$$

with $\mathbf{y} \in \mathbb{R}^n$. Define λ_j, $j = 1, 2, \dots, n$ as the eigenvalues of the Jacobian matrix $J(\hat{t}, \hat{\mathbf{y}})$ of $\mathbf{f}$, repeated according to their algebraic multiplicity, and typical values of $\hat{t}$ and $\hat{\mathbf{y}}$. The system is stable if

$$\mathrm{Re}(\lambda_j) < 0,$$

for $j = 1, 2, \dots, n$.
A numerical time-integration method with amplification factor Q is stable if

$$|Q(\lambda_j \Delta t)| \leq 1,$$

for $j = 1, 2, \dots, n$.

Stability regions.

Consider a numerical time-integration method with amplification factor Q. The stability region S of this method is defined as

$$S = \{\lambda \Delta t \in \mathbb{C} : |Q(\lambda \Delta t)| \leq 1\}.$$

Stiff systems.

Consider a solution $\mathbf{y}(t)$ of some initial-value problem, and assume it contains a rapidly-decaying part and a slowly-varying part. Then we call the rapidly-decaying part the transient of the solution, and the initial-value problem is called a stiff system.

Superstability.

Consider a numerical time-integration method with amplification factor Q applied to an n-dimensional initial-value problem with eigenvalues λ_j, $j = 1, \dots, n$. This method is superstable if it is stable and

$$\lim_{\mathrm{Re}(\lambda_j \Delta t) \to -\infty} |Q(\lambda_j \Delta t)| < 1$$

for $j = 1, \dots, n$.

All explicit methods are never superstable.

The Backward Euler method is always superstable.

The Trapezoidal method is never superstable.

First-order derivative.

Consider a function $f \in C[a, b]$. Then the first-order derivative f' is defined by

$$f' = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h},$$

if this limit exists.

Finite differences.

Consider a function $f \in C^1[a, b]$. The value $f'(x)$ can be approximated by the following finite differences:

$$\begin{aligned} \text{forward difference} \quad & Q_f(h) = \frac{f(x+h) - f(x)}{h}, \\ \text{backward difference} \quad & Q_b(h) = \frac{f(x) - f(x-h)}{h}, \\ \text{central difference} \quad & Q_c(h) = \frac{f(x+h) - f(x-h)}{2h}. \end{aligned}$$

Truncation error R.

Consider a function $f \in C^1[a, b]$ and a finite difference $Q(h)$ as an approximation of $f'(x)$. Then the truncation error $R(h)$ is defined as

$$R(h) = f'(x) - Q(h).$$

Consider a function $f \in C^3[a, b]$. Then:

$$\begin{aligned} R_f(h) &= f'(x) - Q_f(h) = -\frac{h}{2} f''(\xi) = \mathcal{O}(h) && \text{with } \xi \in (x, x+h), \\ R_b(h) &= f'(x) - Q_b(h) = \frac{h}{2} f''(\eta) = \mathcal{O}(h) && \text{with } \eta \in (x-h, x), \\ R_c(h) &= f'(x) - Q_c(h) = -\frac{h^2}{6} f'''(\zeta) = \mathcal{O}(h^2) && \text{with } \zeta \in (x-h, x+h). \end{aligned}$$

Rounding error S, total error E.

Consider a function $f \in C^1[a,b]$ and a finite difference $Q(h)$ as an approximation of $f'(x)$. Also assume $\hat{f}$ is the function f including rounding errors and $\hat{Q}(h)$ is the finite difference based on $\hat{f}$. Then the rounding error $S(h)$ is defined as

$$S(h) = |Q(h) - \hat{Q}(h)|,$$

and the total error $E(h)$ is defined as

$$E(h) = |f'(x) - \hat{Q}(h)|.$$

Boundary value problems.

A second-order boundary value problem is the (unknown) variable $y(x)$ on $[K,L]$ is a system of the form

$$\begin{cases} -(p(x)y'(x))' + r(x)y'(x) + q(x)y(x) = f(x), & K \le x \le L, \\ a_K y(K) + b_K y'(K) = c_K, \\ a_L y(L) + b_L y'(L) = c_L, \end{cases}$$

where the last two equations are called boundary conditions and all constants and functions are known. Assume $p(x) > 0$ and $q(x) \ge 0$.

Boundary conditions.

Assume a boundary condition of the form

$$a_K y(K) + b_K y'(K) = c_K$$

is given.

If $a_K \neq 0,\ b_K = 0$, we call the BC a Dirichlet boundary condition.

If $a_K = 0,\ b_K \neq 0$, we call the BC a Neumann boundary condition.

If $a_K \neq 0,\ b_K \neq 0$, we call the BC a Robin or mixed boundary condition.

Finite difference method.

Assume a boundary-value problem in $y(x)$ on $[K,L]$ is given. The finite difference method constructs a system of the form

$$A\mathbf{w} = \mathbf{f},$$

where w_j is an approximation of $y(x_j)$ and $x_j = K + j\Delta x$ and $(n+1)\Delta x = L - K$.
Algorithm:

1. Define the grid size $\Delta x = \frac{L-K}{n+1}$ and $x_j = K + j\Delta x$.
2. Evaluate the differential equation in x_j and replace all derivatives with finite differences.
3. Approximate $y(x_j)$ by w_j and ignore all remainder terms.
4. Incorporate the boundary condition at $x = K$.
5. Incorporate the boundary condition at $x = L$.
6. Formulate A, $\mathbf{w}$, and $\mathbf{f}$ such that $\mathbf{w}$ contains all unknown values and $A\mathbf{w} = \mathbf{f}$.
7. Solve $A\mathbf{w} = \mathbf{f}$ for $\mathbf{w}$.

Discretisation - $(p(x)y'(x))'$.

The term

$$(p(x)y'(x))'\big|_{x=x_j}$$

is discretised with the $\mathcal{O}(\Delta x^2)$ finite difference

$$\frac{p(x_{j+1/2})(w_{j+1} - w_j) - p(x_{j-1/2})(w_j - w_{j-1})}{\Delta x^2}.$$

Discretisation - $p(K)y'(K)$.

The term

$$p(K)y'(K)$$

is discretised with the $\mathcal{O}(\Delta x^2)$ finite difference

$$\frac{1}{2}\left(p(x_{-1/2})\frac{w_0 - w_{-1}}{\Delta x} + p(x_{1/2})\frac{w_1 - w_0}{\Delta x}\right),$$

for the term coming from $(p(x)y'(x))'$ and with the $\mathcal{O}(\Delta x^2)$ finite difference

$$p(x_0)\frac{w_1 - w_{-1}}{2\Delta x}$$

for the term coming from $r(x)y'(x)$.

Discretisation - $p(L)y'(L)$.

The term

$$p(L)y'(L)$$

is discretised with the $\mathcal{O}(\Delta x^2)$ finite difference

$$\frac{1}{2}\left(p(x_{n+1/2})\frac{w_{n+1} - w_n}{\Delta x} + p(x_{n+3/2})\frac{w_{n+2} - w_{n+1}}{\Delta x}\right),$$

for the term coming from $(p(x)y'(x))'$ and with the $\mathcal{O}(\Delta x^2)$ finite difference

$$p(x_{n+1})\frac{w_{n+2} - w_n}{2\Delta x}$$

for the term coming from $r(x)y'(x)$.

Upwind discretisations.

An upwind discretisation of the term $vy'(x_j)$ in the differential equation $-y'' + vy' + q(x)y = f(x)$ is given by

$$vy'(x_j) \approx \begin{cases} v\dfrac{w_j - w_{j-1}}{\Delta x}, \text{ if } v \ge 0, \\ v\dfrac{w_{j+1} - w_j}{\Delta x}, \text{ if } v < 0. \end{cases}$$

Norms.

The Euclidean norm $||\mathbf{x}||$ of a vector $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$||\mathbf{x}|| = \sqrt{\sum_{i=1}^{n} x_i^2},$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$.
The natural matrix norm $||A||$ of an $n \times n$-matrix A related to the Euclidean norm is defined as

$$||A|| = \max_{||x||=1} ||A\mathbf{x}||,$$

where $\mathbf{x} \in \mathbb{R}^n$.

Gershgorin circle theorem.

The eigenvalues of an $n \times n$-matrix A are located in the complex plane in the union of circles

$$|z - a_{ii}| \le \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|,$$

where $z \in \mathbb{C}$.

Condition number.

The condition number $\kappa(A)$ of an $n \times n$-matrix A is defined as

$$\kappa(A) = ||A|| \cdot ||A^{-1}||.$$

If A is a symmetric $n \times n$-matrix with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$, then

$$\kappa(A) = \frac{|\lambda|_{\max}}{|\lambda|_{\min}}.$$

Local errors.

The local truncation error $\boldsymbol{\varepsilon}$ of the finite difference method scheme $A\mathbf{w} = \mathbf{f}$ is defined as

$$\boldsymbol{\varepsilon} = A\mathbf{y} - \mathbf{f}$$

where $\mathbf{y}$ contains the exact solutions at the same grid nodes as the approximations in $\mathbf{w}$.
A finite difference method scheme $A\mathbf{w} = \mathbf{f}$ is consistent if

$$\lim_{\Delta x \to 0} ||\boldsymbol{\varepsilon}|| = 0.$$

Stability.

The finite difference method scheme $A\mathbf{w} = \mathbf{f}$ is stable if there exists a constant C independent of Δx such that

$$||A^{-1}|| \le C \text{ for } x \to 0.$$

Global errors.

The global truncation error $\mathbf{e}$ of the finite difference method scheme $A\mathbf{w} = \mathbf{f}$ is defined as

$$\mathbf{e} = \mathbf{y} - \mathbf{w},$$

where $\mathbf{y}$ contains the exact solutions at the same grid nodes as the approximations in $\mathbf{w}$.
A finite difference method scheme $A\mathbf{w} = \mathbf{f}$ is convergent if

$$\lim_{\Delta x \to 0} ||\mathbf{e}|| = 0.$$

Equations and solutions.

A solution p of an equation is of the form

$$f(x) = 0$$

is called a root of the equation and a zero of the function f.
Assume $f \in C[a,b]$ and $f(x)f(y) < 0$ for some $x \in [a,b]$ and some $y \in [a,b]$. Then f has a zero in $[a,b]$.
A solution p of an equation of the form

$$g(x) = x$$

is called a fixed point of the function g.
Assume $g \in C[a,b]$ and $g(x) \in [a,b]$ for all $x \in [a,b]$. Then g has a fixed point in $[a,b]$.
Assume $g \in C^1[a,b]$ and that g has a fixed point in $[a,b]$. If there is a positive value $k < 1$ such that

$$|g'(x)| \le k \text{ for all } x \in [a,b],$$

then p is unique on $[a,b]$.

Convergence.

Assume some numerical method generates a sequence $\{p_n\} = p_0, p_1, p_2, \ldots$. If $\lim_{n\to\infty} p_n = p$, we call the sequence convergent (to p). If there exists $\lambda > 0$ and $\alpha > 0$ such that

$$\lim_{n\to\infty} \frac{|p - p_{n+1}|}{|p - p_n|^\alpha} = \lambda,$$

then $\{p_n\}$ converges to p with order α and asymptotic constant λ.

If $\alpha = 1$ the sequence is linearly convergent.

If $\alpha = 2$ the sequence is quadratically convergent.

Assume $\{p_n\}$ satisfies

$$|p - p_n| \leq k|p - p_{n-1}|, \text{ for } n = 1, 2, \ldots$$

with $0 \leq k \leq 1$. Then $\{p_n\}$ converges to p.
Stopping criteria (examples):

- $|p - p_n| < \varepsilon$
- $|p_n - p_{n-1}| < \varepsilon$
- $\frac{|p_n - p_{n-1}|}{|p_n|} < \varepsilon$
- $|f(p_n)| < \varepsilon$

Bisection method.

Assume f is continuous on the interval $[a, b]$ and $f(a)f(b) < 0$. Then the Bisection method produce a sequence $\{p_n\}$ that converges to a zero $p \in [a, b]$ of f with order $\alpha = 1$, asymptotic constant $\lambda = \frac{1}{2}$, and

$$|p - p_n| \leq \frac{b - a}{2^{n+1}}, \text{ for } n = 0, 1, 2, \ldots.$$

Algorithm:

1. Set $n = 0,\ a_0 = a,\ b_0 = b$.
2. For $n = 0, 1, 2, 3, \ldots$:
   (a) $p_n = \frac{1}{2}(a_n + b_n)$
   (b) If $f(a_n)f(p_n) < 0$, set $a_{n+1} = a_n,\ b_{n+1} = p_n$.
   (c) Otherwise set $a_{n+1} = p_n,\ b_{n+1} = b_n$.
   (d) If converged, stop.

Fixed-point iteration.

Assume $g \in C[a, b]$. Then the fixed-point iteration produces a sequence $\{p_n\}$ that could converge to a fixed point p of g.
Algorithm:

1. Set $n = 0$ and pick $p_0 \in [a, b]$.
2. For $n = 0, 1, 2, 3, \ldots$:
   (a) $p_{n+1} = g(p_n)$
   (b) If converged, stop.

Assume $g \in C^1[a, b]$, $g(x) \in [a, b]$ for all $x \in [a, b]$ and that there is a positive value $k < 1$ such that

$$|g'(x) \leq k| \text{ for all } x \in [a, b].$$

Then the fixed-point iteration $p_{n+1} = g(p_n)$ converges to a unique fixed point $p \in [a, b]$ for each $p_0 \in [a, b]$. If $|g'(p)| \neq 0$, then the fixed-point iteration converges to p with order $\alpha = 1$ and asymptotic constant $\lambda = |g'(p)|$. If $|g'(p)| = 0$, then $\alpha > 1$.

Newton-Raphson method.

Assume $f \in C^1[a, b]$, a root $\in [a, b]$ of f exists and $f'(x) \neq 0$ on $[a, b]$. Then the Newton-Raphson method produces a sequence $\{p_n\}$ that could converge to a zero p of f.
Algorithm:

1. Set $n = 0$ and choose p_0.
2. For $n = 0, 1, 2, 3, \ldots$:
   (a) $p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}$
   (b) If converged, stop.

Assume $f \in C^2[a, b]$ and that f has a zero $p \in [a, b]$ with $f'(p) \neq 0$. Then there is a $\delta > 0$ such that the Newton-Raphson method produces a sequence $\{p_n\}$ that converges to p for any $p_0 \in [p - \delta, p + \delta]$ with order $\alpha = 2$ and asymptotic constant $\lambda = \left|\frac{f''(p)}{2f'(p)}\right|$.

Newton-Raphson method variants.

If $f'(p_n)$ is unknown or unavailable in $p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}$ the following variants can be used:

quasi-Newton method $\quad f'(p_n) = \dfrac{f(p_n + h) - f(p_n)}{h}$ with $h > 0$ small

secant method $\quad f'(p_n) = \dfrac{f(p_n) - f(p_{n-1})}{p_n - p_{n-1}}$

Regula-Falsi method $\quad$ Use the Bisection method with the change

$$p_n = a_n - f(a_n)\frac{b_n - a_n}{f(b_n) - f(a_n)}$$

Systems of non-linear equations.

A system of non-linear equations is a system of the form

$$\text{fixed-point form} \quad \begin{cases} g_1(x_1, \ldots, x_m) &= x_1, \\ g_2(x_1, \ldots, x_m) &= x_2, \\ &\vdots \\ g_m(x_1, \ldots, x_m) &= x_m, \end{cases}$$

or of the form

$$\text{general form} \quad \begin{cases} f_1(x_1, \ldots, x_m) &= 0, \\ f_2(x_1, \ldots, x_m) &= 0, \\ &\vdots \\ f_m(x_1, \ldots, x_m) &= 0. \end{cases}$$

In vector notation:

$$\mathbf{g}(\mathbf{x}) = \mathbf{x} \text{ or } \mathbf{f}(\mathbf{x}) = \mathbf{0}.$$

Fixed-point iteration for systems.

Assume $\mathbf{g} \in C(\mathbb{R}^m)$. Then the fixed-point iteration produces a sequence $\{\mathbf{p}^{(n)}\}$ that could converge to a fixed point $\mathbf{p}$ of $\mathbf{g}$.
Algorithm:

1. Set $n = 0$ and pick $\mathbf{p}^{(0)} \in \mathbb{R}^m$.
2. For $n = 0, 1, 2, 3, \ldots$:
   (a) $\mathbf{p}^{(n+1)} = \mathbf{g}(\mathbf{p}^{(n)})$
   (b) If converged, stop.

Newton-Raphson method for systems.

Assume $\mathbf{f} \in C(\mathbb{R}^m)$ and a root $\mathbf{p}$ of $\mathbf{f}$ exists. Then the Newton-Raphson method produces a sequence $\{\mathbf{p}^{(n)}\}$ that could converge to a zero $\mathbf{p}$ of $\mathbf{f}$.
Algorithm:

1. Set $n = 0$ and choose $\mathbf{p}^{(0)}$.
2. For $n = 0, 1, 2, 3, \ldots$:
   (a) Solve for $\mathbf{s}^{(n)}$: $J(\mathbf{p}^{(n)})\mathbf{s}^{(n)} = -\mathbf{f}(\mathbf{p}^{(n)})$.
   (b) $\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} + \mathbf{s}^{(n)}$
   (c) If converged, stop.

Initial-and-boundary-value problems.

An initial-and-boundary-value problem in the function $y = y(x, t)$ is a problem of the form

$$\begin{cases} \dfrac{\partial y}{\partial t} = k(x,t)\dfrac{\partial}{\partial x}\left(p(x,t)\dfrac{\partial y}{\partial x}\right) - r(x,t)\dfrac{\partial y}{\partial x} - q(x,t)y + f(x,t), & \begin{matrix} L \leq x \leq K, \\ t \geq t_0, \end{matrix} \\ y(L,t) + p(L,t)\dfrac{\partial y}{\partial x} = c_L(t),\ t \geq t_0, \\ y(K,t) + p(K,t)\dfrac{\partial y}{\partial x} = c_K(t),\ t \geq t_0, \\ y(x,0) = y_0(x),\ L \leq x \leq K. \end{cases}$$

Semi-discretisation.

A semi-discretisation of an initial-and-boundary-value problem is a system of the form

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = K\mathbf{u} + \mathbf{r}(t),\ t \geq t_0. \\ \mathbf{u}(t_0) = \mathbf{y}_0, \end{cases}$$

where $\mathbf{u} = \mathbf{u}(t)$ is a vector containing the approximations $u_i(t)$ of $y(x_i, t)$. The semi-discretisation is obtained by applying the finite-difference method to the problem in the spatial coordinate.
Any time-integration method above can be applied to the semi-discretisation. We use $\mathbf{w}^j$ as the approximation of $\mathbf{u}(t_j)$. The i-th component of $\mathbf{w}^j$, w_i^j, is an approximation of $u_i(t_j)$ and therefore an approximation of $y(x_i, t_j)$.
The local truncation error is defined as

$$\boldsymbol{\tau}^{j+1} = \frac{\mathbf{y}^{j+1} - \mathbf{z}^{j+1}}{\Delta t},$$

where $y_i^{j+1} = y(x_i, t_j)$ and $\mathbf{z}^{j+1}$ is the result of applying one time step with $\mathbf{w}^j = \mathbf{y}^j$.