Lecture 4: Numerical solution of ordinary differential equations

Habib Ammari

Department of Mathematics, ETH Zürich

Numerical solution of ODEs

- General explicit one-step method:
  - **Consistency;**
  - **Stability;**
  - **Convergence.**
- High-order methods:
  - **Taylor methods;**
  - **Integral equation method;**
  - **Runge-Kutta methods.**
- Multi-step methods.

Numerical solution of ODEs

- Stiff equations and systems.
- Perturbation theories for differential equations:
    - **Regular perturbation theory;**
    - **Singular perturbation theory.**

Numerical solution of ODEs

- Consistency, stability and convergence
- Consider

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = f(t,x), \quad t \in [0,T], \\ x(0) = x_0, \quad x_0 \in \mathbb{R}. \end{cases}$$

- $f \in \mathcal{C}^0([0,t] \times \mathbb{R})$: Lipschitz condition.
- Start at the initial time $t = 0$;
- Introduce successive discretization points

$$t_0 = 0 < t_1 < t_2 < \ldots,$$

continuing on until we reach the final time T.

- Uniform **step size**:

$$\Delta t := t_{k+1} - t_k > 0,$$

does not dependent on k and assumed to be relatively small, with $t_k = k\Delta t$.

- Suppose that $K = T/(\Delta t)$: an integer.

Numerical solution of ODEs

- General **explicit one-step method**:

$$x^{k+1} = x^k + \Delta t\, \Phi(t_k, x^k, \Delta t),$$

  for some continuous function $\Phi(t, x, h)$.

- Taking in succession $k = 0, 1, \ldots, K-1$, **one-step** at a time $\Rightarrow$ the approximate values x^k of x at t_k: obtained.
- **Explicit** scheme: x^{k+1} obtained from x^k; x^{k+1} appears only on the left-hand side.

Numerical solution of ODEs

- Truncation error of the numerical scheme:

$$T_k(\Delta t) = \frac{x(t_{k+1}) - x(t_k)}{\Delta t} - \Phi(t_k, x(t_k), \Delta t).$$

- As $\Delta t \to 0$, $k \to +\infty$, $k\Delta t = t$,

$$T_k(\Delta t) \to \frac{dx}{dt} - \Phi(t, x, 0).$$

- DEFINITION: Consistency
  - Numerical scheme consistent with the ODE if

$$\Phi(t, x, 0) = f(t, x) \quad \text{for all } t \in [0, T] \text{ and } x \in \mathbb{R}.$$

Numerical solution of ODEs

- DEFINITION: Stability
  - Numerical scheme stable if Φ: Lipschitz continuous in x, i.e., there exist positive constants C_Φ and h_0 s.t.

$$|\Phi(t,x,h) - \Phi(t,y,h)| \leq C_\Phi |x-y|, \ t \in [0,T], h \in [0,h_0], x,y \in \mathbb{R}.$$

- **Global error** of the numerical scheme:

$$e_k = x^k - x(t_k).$$

- DEFINITION: Convergence
  - Numerical scheme: convergent if

$$|e_k| \to 0 \quad \text{as } \Delta t \to 0, k \to +\infty, k\Delta t = t \in [0,T].$$

Numerical solution of ODEs

- THEOREM: Dahlquist-Lax equivalence theorem
  - Numerical scheme: convergent iff consistent and stable.

Numerical solution of ODEs

- PROOF:
  - $$x(t_{k+1}) - x(t_k) = \int_{t_k}^{t_{k+1}} f(s, x(s))\, ds;$$
  - $\Rightarrow$
    $$x(t_{k+1}) - x(t_k) = (\Delta t) f(t_k, x(t_k)) + \int_{t_k}^{t_{k+1}} \big[f(s, x(s)) - f(t_k, x(t_k))\big]\, ds.$$
  - $\Rightarrow$
    $$\Big| x(t_{k+1}) - x(t_k) - (\Delta t) f(t_k, x(t_k)) \Big|$$
    $$= \Big| \int_{t_k}^{t_{k+1}} \big[f(s, x(s)) - f(t_k, x(t_k))\big]\, ds \Big| \leq (\Delta t)\, \omega_1(\Delta t).$$

Numerical solution of ODEs

- $\omega_1(\Delta t)$:

$$\omega_1(\Delta t) := \sup\left\{|f(t,x(t)) - f(s,x(s))|, 0 \leq s,t \leq T, |t-s| \leq \Delta t\right\}.$$

  - $\omega_1(\Delta t) \to 0$ as $\Delta t \to 0$.
  - If f: Lipschitz in t, then $\omega_1(\Delta t) = O(\Delta t)$.

Numerical solution of ODEs

- From
$$e_{k+1} - e_k = x^{k+1} - x^k - (x(t_{k+1}) - x(t_k)),$$
- $\Rightarrow$
$$e_{k+1} - e_k = \Delta t \, \Phi(t_k, x^k, \Delta t) - (x(t_{k+1}) - x(t_k)).$$
- Or equivalently,
$$e_{k+1} - e_k = \Delta t \left[\Phi(t_k, x^k, \Delta t) - f(t_k, x(t_k))\right] - \left[x(t_{k+1}) - x(t_k) - \Delta t \, f(t_k, x(t_k))\right].$$
- Write
$$\begin{aligned} e_{k+1} - e_k \quad &= \quad \Delta t \left[\Phi(t_k, x^k, \Delta t) - \Phi(t_k, x(t_k), \Delta t) + \Phi(t_k, x(t_k), \Delta t)\right. \\ &\quad \left. - f(t_k, x(t_k))\right] - \left[x(t_{k+1}) - x(t_k) - \Delta t \, f(t_k, x(t_k))\right]. \end{aligned}$$

Numerical solution of ODEs

- Let

$$\omega_2(\Delta t) := \sup \left\{ |\Phi(t, x, h) - f(t, x)|, t \in [0, T], x \in \mathbb{R}, 0 < h \leq (\Delta t) \right\}.$$

- Consistency $\Rightarrow$

$$\left| \Phi(t_k, x(t_k), \Delta t) - f(t_k, x(t_k)) \right| \leq \omega_2(\Delta t) \to 0 \text{ as } \Delta t \to 0.$$

- Stability condition $\Rightarrow$

$$\left| \Phi(t_k, x^k, \Delta t) - \Phi(t_k, x(t_k), \Delta t) \right| \leq C_\Phi |e_k|.$$

Numerical solution of ODEs

- $\Rightarrow$

$$|e_{k+1}| \leq (1 + C_\Phi \Delta t)|e_k| + \Delta t \omega_3(\Delta t), \quad 0 \leq k \leq K-1;$$

- $K = T/(\Delta t)$ and $\omega_3(\Delta t) := \omega_1(\Delta t) + \omega_2(\Delta t) \to 0$ as $\Delta t \to 0$.

Numerical solution of ODEs

- By induction,

$$|e_{k+1}| \leq (1 + C_\Phi \Delta t)^k |e_0| + (\Delta t)\, \omega_3(\Delta t) \sum_{l=0}^{k-1} (1 + C_\Phi \Delta t)^l, \quad 0 \leq k \leq K.$$

- $$\sum_{l=0}^{k-1} (1 + C_\Phi \Delta t)^l = \frac{(1 + C_\Phi \Delta t)^k - 1}{C_\Phi \Delta t},$$

  and

$$(1 + C_\Phi \Delta t)^K \leq (1 + C_\Phi \frac{T}{K})^K \leq e^{C_\Phi T}.$$

- $\Rightarrow$

$$|e_k| \leq e^{C_\Phi T} |e_0| + \frac{e^{C_\Phi T} - 1}{C_\Phi} \omega_3(\Delta t).$$

- If $e_0 = 0$, then as $\Delta t \to 0$, $k \to +\infty$ s.t. $k\Delta t = t \in [0, T]$

$$\lim_{k \to +\infty} |e_k| = 0.$$

Numerical solution of ODEs

- DEFINITION:
  - An explicit one-step method: **order** p if there exist positive constants h_0 and C s.t.

$$|T_k(\Delta t)| \leq C(\Delta t)^p, \quad 0 < \Delta t \leq h_0, k = 0, \ldots, K-1;$$

$T_k(\Delta t)$: truncation error.

Numerical solution of ODEs

- If the explicit one-step method: stable $\Rightarrow$ global error: bounded by the truncation error.
- PROPOSITION:
  - Consider the explicit one-step scheme with Φ satisfying the stability condition.
  - Suppose that $e_0 = 0$.
  - Then

$$|e_{k+1}| \leq \frac{(e^{C_\Phi T} - 1)}{C_\Phi} \max_{0 \leq l \leq k} |T_l(\Delta t)| \quad \text{for } k = 0, \ldots, K-1;$$

  - T_l: truncation error and e_k: global error.

Numerical solution of ODEs

- PROOF:
  - $$e_{k+1}-e_k = -(\Delta t)T_k(\Delta t)+(\Delta t)\Big[\Phi(t_k, x^k, \Delta t)-\Phi(t_k, x(t_k), \Delta t)\Big].$$
  - $\Rightarrow$ $$\begin{aligned} |e_{k+1}| &\leq (1+C_\Phi(\Delta t))|e_k| + (\Delta t)|T_k(\Delta t)| \\ &\leq (1+C_\Phi(\Delta t))|e_k| + (\Delta t)\max_{0\leq l\leq k}|T_l(\Delta t)|. \end{aligned}$$

Numerical solution of ODEs

- Explicit Euler's method
  - $\Phi(t, x, h) = f(t, x)$.
  - Explicit Euler scheme:

$$x^{k+1} = x^k + (\Delta t) f(t, x^k).$$

Numerical solution of ODEs

- THEOREM:
  - Suppose that f satisfies the Lipschitz condition;
  - Suppose that f: Lipschitz with respect to t.
  - Then the explicit Euler scheme: convergent and the global error e_k: of order Δt.
  - If $f \in \mathcal{C}^1$, then the scheme: of order one.

Numerical solution of ODEs

- PROOF:
  - f satisfies the Lipschitz condition $\Rightarrow$ numerical scheme with $\Phi(t, x, h) = f(t, x)$: stable.
  - $\Phi(t, x, 0) = f(t, x)$ for all $t \in [0, T]$ and $x \in \mathbb{R} \Rightarrow$ numerical scheme: consistent.
  - $\Rightarrow$ convergence.
  - f: Lipschitz in $t \Rightarrow \omega_1(\Delta t) = O(\Delta t)$.
  - $\omega_2(\Delta t) = 0 \Rightarrow \omega_3(\Delta t) = O(\Delta t)$.
  - $\Rightarrow |e_k| = O(\Delta t)$ for $1 \leq k \leq K$.

Numerical solution of ODEs

- $f \in \mathcal{C}^1 \Rightarrow x \in \mathcal{C}^2$.
- Mean-value theorem $\Rightarrow$

$$T_k(\Delta t) = \frac{1}{\Delta t}\Big(x(t_{k+1}) - x(t_k)\Big) - f(t_k, x(t_k))$$

$$= \frac{1}{\Delta t}\Big(x(t_k) + (\Delta t)\frac{dx}{dt}(t_k) + \frac{(\Delta t)^2}{2}\frac{d^2x}{dt^2}(\tau) - x(t_k)\Big) - f(t_k, x(t_k))$$

$$= \frac{\Delta t}{2}\frac{d^2x}{dt^2}(\tau),$$

for some $\tau \in [t_k, t_{k+1}]$.

- $\Rightarrow$ Scheme: first order.

Numerical solution of ODEs

- High-order methods:
  - In general, the order of a numerical solution method governs both the accuracy of its approximations and the speed of convergence to the true solution as the step size $\Delta t \to 0$.
  - Explicit Euler method: only a first order scheme;
  - Devise simple numerical methods that enjoy a higher order of accuracy.
  - The higher the order, the more accurate the numerical scheme, and hence the larger the step size that can be used to produce the solution to a desired accuracy.
  - However, this should be balanced with the fact that higher order methods inevitably require more computational effort at each step.

Numerical solution of ODEs

- High-order methods:
  - **Taylor methods;**
  - **Integral equation method;**
  - **Runge-Kutta methods.**

Numerical solution of ODEs

- Taylor methods
- Explicit Euler scheme: based on a first order Taylor approximation to the solution.
- Taylor expansion of the solution $x(t)$ at the discretization points t_{k+1}:

$$x(t_{k+1}) = x(t_k) + (\Delta t)\frac{dx}{dt}(t_k) + \frac{(\Delta t)^2}{2}\frac{d^2x}{dt^2}(t_k) + \frac{(\Delta t)^3}{6}\frac{d^3x}{dt^3}(t_k) + \dots .$$

- Evaluate the first derivative term by using the differential equation

$$\frac{dx}{dt} = f(t,x).$$

Numerical solution of ODEs

- Second derivative can be found by differentiating the equation with respect to t:

$$\frac{d^2x}{dt^2} = \frac{d}{dt}f(t,x) = \frac{\partial f}{\partial t}(t,x) + \frac{\partial f}{\partial x}(t,x)\frac{dx}{dt}.$$

- **Second order Taylor method**

$$(*) \quad x^{k+1} = x^k + (\Delta t)f(t_k, x^k) + \frac{(\Delta t)^2}{2}\left(\frac{\partial f}{\partial t}(t_k, x^k) + \frac{\partial f}{\partial x}(t_k, x^k)f(t_k, x^k)\right).$$

Numerical solution of ODEs

- Proposition:
  - Suppose that $f \in \mathcal{C}^2$.
  - Then $(*)$: of second order.

Numerical solution of ODEs

- Proof:
  - $f \in \mathcal{C}^2 \Rightarrow x \in \mathcal{C}^3$.
  - $\Rightarrow$ truncation error T_k given by

$$T_k(\Delta t) = \frac{(\Delta t)^2}{6} \frac{d^3 x}{dt^3}(\tau),$$

for some $\tau \in [t_k, t_{k+1}]$ and so, $(*)$: of second order.

Numerical solution of ODEs

- Drawbacks of higher order Taylor methods:
  (i) Owing to their dependence upon the partial derivatives of f, f needs to be smooth;
  (ii) Efficient evaluation of the terms in the Taylor approximation and avoidance of round off errors.

Numerical solution of ODEs

- Integral equation method
- Avoid the complications inherent in a direct Taylor expansion.
- $x(t)$ coincides with the solution to the **integral equation**

$$x(t) = x_0 + \int_0^t f(s, x(s))\, ds, \quad t \in [0, T].$$

Starting at the discretization point t_k instead of 0, and integrating until time $t = t_{k+1}$ gives

$$(**) \quad x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} f(s, x(s))\, ds.$$

- Implicitly computes the value of the solution at the subsequent discretization point.

Numerical solution of ODEs

- Compare formula $(**)$ with the explicit Euler method

$$x^{k+1} = x^k + (\Delta t) f(t_k, x^k).$$

- $\Rightarrow$ Approximation of the integral by

$$\int_{t_k}^{t_{k+1}} f(s, x(s))\, ds \approx (\Delta t) f(t_k, x(t_k)).$$

- Left endpoint rule for numerical integration.

Numerical solution of ODEs

- Left endpoint rule for numerical integration:

- **Left endpoint rule**: not an especially accurate method of numerical integration.
- Better methods include the Trapezoid rule:

Numerical solution of ODEs

- Numerical integration formulas for continuous functions.

  (i) **Trapezoidal rule**:

  $$\int_{t_k}^{t_{k+1}} g(s)\, ds \approx \frac{\Delta t}{2}\Big(g(t_{k+1}) + g(t_k)\Big);$$

  (ii) **Simpson's rule**:

  $$\int_{t_k}^{t_{k+1}} g(s)\, ds \approx \frac{\Delta t}{6}\Big(g(t_{k+1}) + 4g(\frac{t_k + t_{k+1}}{2}) + g(t_k)\Big);$$

  (iii) Trapezoidal rule: **exact for polynomials of order one**;
  Simpson's rule: **exact for polynomials of second order**.

Numerical solution of ODEs

- Use the more accurate Trapezoidal approximation

$$\int_{t_k}^{t_{k+1}} f(s, x(s))\, ds \approx \frac{(\Delta t)}{2} \Big[f(t_k, x(t_k)) + f(t_{k+1}, x(t_{k+1})) \Big].$$

- Trapezoidal scheme:

$$x^{k+1} = x^k + \frac{(\Delta t)}{2} \Big[f(t_k, x^k) + f(t_{k+1}, x^{k+1}) \Big].$$

- Trapezoidal scheme: implicit numerical method.

Numerical solution of ODEs

- Proposition:
  - Suppose that $f \in \mathcal{C}^2$ and

$$(***) \quad \frac{(\Delta t) C_f}{2} < 1;$$

    C_f: Lipschitz constant for f in x.
  - Trapezoidal scheme: convergent and of second order.

Numerical solution of ODEs

- Proof:
  - Consistency:

$$\Phi(t, x, \Delta t) := \frac{1}{2}\left[f(t, x) + f(t + \Delta t, x + (\Delta t)\Phi(t, x, \Delta t))\right].$$

  - $\Delta t = 0$.

Numerical solution of ODEs

- Stability:
  - $$\left|\Phi(t,x,\Delta t)-\Phi(t,y,\Delta t)\right| \leq C_f|x-y| + \frac{\Delta t}{2} C_f \left|\Phi(t,x,\Delta t)-\Phi(t,y,\Delta t)\right|.$$
  - $\Rightarrow$ $$\left(1-\frac{(\Delta t)C_f}{2}\right)\left|\Phi(t,x,\Delta t)-\Phi(t,y,\Delta t)\right| \leq C_f|x-y|.$$
  - $\Rightarrow$ Stability holds with $$C_\Phi = \frac{C_f}{1-\frac{(\Delta t)C_f}{2}},$$ provided that Δt satisfies $(***)$.

Numerical solution of ODEs

- Second order scheme:
    - By the mean-value theorem,

$$
\begin{aligned}
T_k(\Delta t) &= \frac{x(t_{k+1}) - x(t_k)}{\Delta t} \\
&\quad - \frac{1}{2}\Big[f(t_k, x(t_k)) + f(t_{k+1}, x(t_{k+1})) \Big] \\
&= -\frac{1}{12}(\Delta t)^2 \frac{d^3 x}{dt^3}(\tau),
\end{aligned}
$$

for some $\tau \in [t_k, t_{k+1}] \Rightarrow$ second order scheme, provided that $f \in \mathcal{C}^2$ (and consequently $x \in \mathcal{C}^3$).

Numerical solution of ODEs

- An alternative scheme: replace x^{k+1} by $x^k + (\Delta t) f(t_k, x^k)$.
- $\Rightarrow$ Improved Euler scheme:

$$x^{k+1} = x^k + \frac{(\Delta t)}{2}\left[f(t_k, x^k) + f(t_{k+1}, \mathbf{x^k} + (\mathbf{\Delta t})\mathbf{f}(\mathbf{t_k}, \mathbf{x^k}))\right].$$

- Proposition: Improved Euler scheme: convergent and of second order.
- Improved Euler scheme: performs comparably to the Trapezoidal scheme, and significantly better than the Euler scheme.
- Alternative numerical approximations to the integral equation $\Rightarrow$ a range of numerical solution schemes.

Numerical solution of ODEs

- **Midpoint rule**:

$$\int_{t_k}^{t_{k+1}} f(s, x(s))\, ds \approx (\Delta t) f(t_k + \frac{\Delta t}{2}, x(t_k + \frac{\Delta t}{2})).$$

- Midpoint rule: same order of accuracy as the trapezoid rule.
- **Midpoint scheme:** approximate $x(t_k + \frac{\Delta t}{2})$ by $x^k + \frac{\Delta t}{2} f(t_k, x^k)$,

$$x^{k+1} = x^k + (\Delta t) f\big(t_k + \frac{\Delta t}{2}, x^k + \frac{\Delta t}{2} f(t_k, x^k)\big).$$

- Midpoint scheme: of second order.

Numerical solution of ODEs

- Example of linear systems
- Consider the linear system of ODEs

$$\begin{cases} \frac{dx}{dt} = Ax(t), \quad t \in [0, +\infty[, \\ x(0) = x_0 \in \mathbb{R}^d. \end{cases}$$

- $A \in \mathbb{M}_d(\mathbb{C})$: independent of t.
- DEFINITION:
  - A one-step numerical scheme for solving the linear system of ODEs: **stable** if there exists a positive constant C_0 s.t.

$$|x^{k+1}| \leq C_0 |x^0| \quad \text{for all } k \in \mathbb{N}.$$

Numerical solution of ODEs

- Consider the following schemes:
  (i) Explicit Euler's scheme:
$$x^{k+1} = x^k + (\Delta t)Ax^k;$$
  (ii) Implicit Euler's scheme:
$$x^{k+1} = x^k + (\Delta t)Ax^{k+1};$$
  (iii) Trapezoidal scheme:
$$x^{k+1} = x^k + \frac{(\Delta t)}{2}\left[Ax^k + Ax^{k+1}\right],$$
  with $k \in \mathbb{N}$, and $x^0 = x_0$.

Numerical solution of ODEs

- Proposition:

  Suppose that $\Re\lambda_j < 0$ for all j. The following results hold:

  (i) Explicit Euler scheme: stable for Δt small enough;
  (ii) Implicit Euler scheme: unconditionally stable;
  (iii) Trapezoidal scheme: unconditionally stable.

Numerical solution of ODEs

- Proof:
  - Consider the explicit Euler scheme. By a change of basis,

$$\widetilde{x}^{k+1} = (I + \Delta t(D+N))^k \widetilde{x}^0,$$

    where $\widetilde{x}^k = Cx^k$.
  - If $\widetilde{x}^0 \in E_j$, then

$$\widetilde{x}^k = \sum_{l=0}^{\min\{k,d\}} C_k^l (1+\Delta t \lambda_j)^{k-l} (\Delta t)^l N^l \widetilde{x}^0,$$

    C_k^l: binomial coefficient.

Numerical solution of ODEs

- If $|1+(\Delta t)\lambda_j| < 1$, then $\widetilde{x}^k$: bounded.
- If $|1+(\Delta t)\lambda_j| > 1$, then one can find $\widetilde{x}^0$ s.t. $|\widetilde{x}^k| \to +\infty$ (exponentially) as $k \to +\infty$.
- If $|1+(\Delta t)\lambda_j| = 1$ and $N \neq 0$, then for all $\widetilde{x}^0$ s.t. $N\widetilde{x}^0 \neq 0, N^2\widetilde{x}^0 = 0$,

$$\widetilde{x}^k = (1+(\Delta t)\lambda_j)^k \widetilde{x}^0 + (1+(\Delta t)\lambda_j)^{k-1} k\Delta t N\widetilde{x}^0$$

  goes to infinity as $k \to +\infty$.
- Stability condition $|1+(\Delta t)\lambda_j| < 1 \Leftrightarrow$

$$\Delta t < -2\frac{\Re\lambda_j}{|\lambda_j|^2},$$

  holds for Δt small enough.

Numerical solution of ODEs

- Implicit Euler scheme:

$$\tilde{x}^{k+1} = (I - \Delta t(D+N))^{-k} \tilde{x}^0 .$$

- All the eigenvalues of the matrix $(I - \Delta t(D+N))^{-1}$: of modulus strictly smaller than 1.
- $\Rightarrow$ Implicit Euler scheme: unconditionally stable.
- Trapezoidal scheme:

$$\tilde{x}^{k+1} = (I - \frac{(\Delta t)}{2}(D+N))^{-k}(I + \frac{(\Delta t)}{2}(D+N))^{k} \tilde{x}^0 .$$

- Stability condition:

$$|1 + \frac{(\Delta t)}{2}\lambda_j| < |1 - \frac{(\Delta t)}{2}\lambda_j|,$$

holds for all $\Delta t > 0$ since $\Re \lambda_j < 0$.

Numerical solution of ODEs

- REMARK: Explicit and implicit Euler schemes: of order one; Trapezoidal scheme: of order two.

Numerical solution of ODEs

- Runge-Kutta methods:
  - By far the most popular and powerful general-purpose numerical methods for integrating ODEs.
  - Idea behind: evaluate f at carefully chosen values of its arguments, t and x, in order to create an accurate approximation (as accurate as a higher-order Taylor expansion) of $x(t + \Delta t)$ without evaluating derivatives of f.

Numerical solution of ODEs

- Runge-Kutta schemes: derived by matching **multivariable Taylor series expansions** of $f(t, x)$ with the Taylor series expansion of $x(t + \Delta t)$.
- To find the right values of t and x at which to evaluate f:
  - Take a Taylor expansion of f evaluated at these (unknown) values;
  - Match the resulting numerical scheme to a Taylor series expansion of $x(t + \Delta t)$ around t.

Numerical solution of ODEs

- Generalization of Taylor's theorem to functions of two variables:

  THEOREM:

  - $f(t,x) \in \mathcal{C}^{n+1}([0, T] \times \mathbb{R})$. Let $(t_0, x_0) \in [0, T] \times \mathbb{R}$.
  - There exist $t_0 \leq \tau \leq t$, $x_0 \leq \xi \leq x$, s.t.

$$f(t,x) = P_n(t,x) + R_n(t,x),$$

  - $P_n(t,x)$: nth Taylor polynomial of f around (t_0, x_0);
  - $R_n(t,x)$: remainder term associated with $P_n(t,x)$.

Numerical solution of ODEs

-
$$P_n(t,x) = f(t_0,x_0) + \left[(t-t_0)\frac{\partial f}{\partial t}(t_0,x_0) + (x-x_0)\frac{\partial f}{\partial x}(t_0,x_0)\right]$$
$$+\left[\frac{(t-t_0)^2}{2}\frac{\partial^2 f}{\partial t^2}(t_0,x_0) + (t-t_0)(x-x_0)\frac{\partial^2 f}{\partial t\partial x}(t_0,x_0)\right.$$
$$\left.+\frac{(x-x_0)^2}{2}\frac{\partial^2 f}{\partial x^2}(t_0,x_0)\right]$$
$$\ldots + \left[\frac{1}{n!}\sum_{j=0}^{n} C_j^n (t-t_0)^{n-j}(x-x_0)^j \frac{\partial^n f}{\partial t^{n-j}\partial x^j}(t_0,x_0)\right];$$

-
$$R_n(t,x) = \frac{1}{(n+1)!}\sum_{j=0}^{n+1} C_j^{n+1}(t-t_0)^{n+1-j}(x-x_0)^j \frac{\partial^{n+1} f}{\partial t^{n+1-j}\partial x^j}(\tau,\xi).$$

Numerical solution of ODEs

- Illustration: obtain a second-order accurate method (truncation error $O((\Delta t)^2)$).
- Match

$$x + \Delta t f(t,x) + \frac{(\Delta t)^2}{2}\Big[\frac{\partial f}{\partial t}(t,x) + \frac{\partial f}{\partial x}(t,x) f(t,x)\Big] + \frac{(\Delta t)^3}{6}\frac{d^2}{dt^2}[f(\tau,x)]$$

  to

$$x + (\Delta t) f(t+\alpha_1, x+\beta_1),$$

  $\tau \in [t, t+\Delta t]$ and α_1 and β_1: to be found.
- Match

$$f(t,x) + \frac{(\Delta t)}{2}\Big[\frac{\partial f}{\partial t}(t,x) + \frac{\partial f}{\partial x}(t,x) f(t,x)\Big] + \frac{(\Delta t)^2}{6}\frac{d^2}{dt^2}[f(t,x)]$$

  with $f(t+\alpha_1, x+\beta_1)$ at least up to terms of the order of $O(\Delta t)$.

Numerical solution of ODEs

- Multivariable version of Taylor's theorem to f,

$$f(t+\alpha_1, x+\beta_1) = f(t,x) + \alpha_1 \frac{\partial f}{\partial t}(t,x) + \beta_1 \frac{\partial f}{\partial x}(t,x) + \frac{\alpha_1^2}{2}\frac{\partial^2 f}{\partial t^2}(\tau,\xi)$$
$$+\alpha_1\beta_1 \frac{\partial^2 f}{\partial t \partial x}(\tau,\xi) + \frac{\beta_1^2}{2}\frac{\partial^2 f}{\partial x^2}(\tau,\xi),$$

  $t \leq \tau \leq t+\alpha_1$ and $x \leq \xi \leq x+\beta_1$.

- $\Rightarrow$

$$\alpha_1 = \frac{\Delta t}{2} \quad \text{and} \quad \beta_1 = \frac{\Delta t}{2} f(t,x).$$

- $\Rightarrow$ Resulting numerical scheme: explicit midpoint method: the simplest example of a Runge-Kutta method of second order.
- Improved Euler method: also another often-used Runge-Kutta method.

Numerical solution of ODEs

- General Runge-Kutta method:

$$x^{k+1} = x^k + \Delta t \sum_{i=1}^{m} c_i f(t_{i,k}, x_{i,k}),$$

  m: number of terms in the method.

- Each $t_{i,k}$ denotes a point in $[t_k, t_{k+1}]$.
- Second argument $x_{i,k} \approx x(t_{i,k})$ can be viewed as an approximation to the solution at the point $t_{i,k}$.
- To construct an nth order Runge-Kutta method, we need to take at least $m \geq n$ terms.

Numerical solution of ODEs

- Best-known Runge-Kutta method: **fourth-order Runge-Kutta method**, which uses four evaluations of f during each step.

$$\begin{cases} \kappa_1 := f(t_k, x^k), \\ \kappa_2 := f(t_k + \frac{\Delta t}{2}, x^k + \frac{\Delta t}{2}\kappa_1), \\ \kappa_3 := f(t_k + \frac{\Delta t}{2}, x^k + \frac{\Delta t}{2}\kappa_2), \\ \kappa_4 := f(t_{k+1}, x^k + \Delta t \kappa_3), \\ x^{k+1} = x^k + \dfrac{(\Delta t)}{6}(\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4). \end{cases}$$

- Values of f at the midpoint in time: given four times as much weight as values at the endpoints t_k and t_{k+1} (similar to Simpson's rule from numerical integration).

Numerical solution of ODEs

- Construction of Runge-Kutta methods:
  - Construct Runge-Kutta methods by generalizing collocation methods.
  - Discuss their consistency, stability, and order.

Numerical solution of ODEs

- Collocation methods:
- $\mathcal{P}_m$: space of real polynomials of degree $\leq m$.
- Interpolating polynomial:
  - Given a set of m **distinct** quadrature points $c_1 < c_2 < \ldots < c_m$ in $\mathbb{R}$, and corresponding data $g_1, \ldots, g_m$;
  - There exists a unique polynomial, $P(t) \in \mathcal{P}_{m-1}$ s.t.

$$P(c_i) = g_i, i = 1, \ldots, m.$$

Numerical solution of ODEs

- DEFINITION:
  - Define the ith Lagrange interpolating polynomial $l_i(t)$, $i = 1, \ldots, m$, for the set of quadrature points $\{c_j\}$ by

$$l_i(t) := \prod_{j \neq i, j=1}^{m} \frac{t - c_j}{c_i - c_j}.$$

- Set of Lagrange interpolating polynomials: form a basis of $\mathcal{P}_{m-1}$;
- Interpolating polynomial P corresponding to the data $\{g_j\}$ given by

$$P(t) := \sum_{i=1}^{m} g_i l_i(t).$$

Numerical solution of ODEs

- Consider a smooth function g on $[0, 1]$.
- Approximate the integral of g on $[0, 1]$ by exactly integrating the Lagrange interpolating polynomial of order $m - 1$ based on m quadrature points $0 \leq c_1 < c_2 < \ldots < c_m \leq 1$.
- Data: values of g at the quadrature points $g_i = g(c_i)$, $i = 1, \ldots, m$.

Numerical solution of ODEs

- Define the weights

$$b_i = \int_0^1 l_i(s)\, ds.$$

- Quadrature formula:

$$\int_0^1 g(s)\, ds \approx \int_0^1 \sum_{i=1}^{m} g_i l_i(s)\, ds = \sum_{i=1}^{m} b_i g(c_i).$$

Numerical solution of ODEs

- f: smooth function on $[0, T]$; $t_k = k\Delta t$ for $k = 0, \ldots, K = T/(\Delta t)$: discretization points in $[0, T]$.
- $\int_{t_k}^{t_{k+1}} f(s)\, ds$ can be approximated by

$$\int_{t_k}^{t_{k+1}} f(s)\, ds = (\Delta t) \int_0^1 f(t_k + \Delta t \tau)\, d\tau \approx (\Delta t) \sum_{i=1}^{m} b_i f(t_k + (\Delta t) c_i).$$

Numerical solution of ODEs

- x: polynomial of degree m satisfying

$$\begin{cases} x(0) = x_0, \\ \frac{dx}{dt}(c_i \Delta t) = F_i, \end{cases}$$

  $F_i \in \mathbb{R}, i = 1, \ldots, m.$

- Lagrange interpolation formula $\Rightarrow$ for t in the first time-step interval $[0, \Delta t]$,

$$\frac{dx}{dt}(t) = \sum_{i=1}^{m} F_i l_i(\frac{t}{\Delta t}).$$

Numerical solution of ODEs

- Integrating over the intervals $[0, c_i \Delta t] \Rightarrow$

$$x(c_i \Delta t) = x_0 + (\Delta t) \sum_{j=1}^{m} F_j \int_0^{c_i} l_j(s)\, ds = x_0 + (\Delta t) \sum_{j=1}^{m} a_{ij} F_j,$$

  for $i = 1, \ldots, m$, with

$$a_{ij} := \int_0^{c_i} l_j(s)\, ds.$$

- Integrating over $[0, \Delta t] \Rightarrow$

$$x(\Delta t) = x_0 + (\Delta t) \sum_{i=1}^{m} F_i \int_0^{1} l_i(s)\, ds = x_0 + (\Delta t) \sum_{i=1}^{m} b_i F_i.$$

Numerical solution of ODEs

- Writing $dx/dt = f(x(t))$, on the first time step interval $[0, \Delta t]$,

$$\begin{cases} F_i = f(x_0 + (\Delta t) \sum_{j=1}^{m} a_{ij} F_j), \quad i = 1, \ldots, m, \\ x(\Delta t) = x_0 + (\Delta t) \sum_{i=1}^{m} b_i F_i. \end{cases}$$

- Similarly, we have on $[t_k, t_{k+1}]$

$$\begin{cases} F_{i,k} = f(x(t_k) + (\Delta t) \sum_{j=1}^{m} a_{ij} F_{j,k}), \quad i = 1, \ldots, m, \\ x(t_{k+1}) = x(t_k) + (\Delta t) \sum_{i=1}^{m} b_i F_{i,k}. \end{cases}$$

- In the **collocation method**: one first solves the coupled nonlinear system to obtain $F_{i,k}$, $i = 1, \ldots, m$, and then computes $x(t_{k+1})$ from $x(t_k)$.

Numerical solution of ODEs

- REMARK:
  - $$t^{l-1} = \sum_{i=1}^{m} c_i^{l-1} l_i(t), \quad t \in [0,1], l = 1, \ldots, m,$$
  - $\Rightarrow$
    $$\sum_{i=1}^{m} b_i c_i^{l-1} = \frac{1}{l}, \quad l = 1, \ldots, m,$$
    and
    $$\sum_{j=1}^{m} a_{ij} c_j^{l-1} = \frac{c_i^l}{l}, \quad i, l = 1, \ldots, m.$$

Numerical solution of ODEs

- Runge-Kutta methods as generalized collocation methods
  - In the collocation method, the coefficients b_i and a_{ij}: defined by certain integrals of the Lagrange interpolating polynomials associated with a chosen set of quadrature nodes c_i, $i = 1, \ldots, m$.
  - Natural generalization of collocation methods: obtained by allowing the coefficients c_i, b_i, and a_{ij} to take arbitrary values, not necessary related to quadrature formulas.

Numerical solution of ODEs

- No longer assume the c_i to be distinct.
- However, assume that

$$c_i = \sum_{j=1}^{m} a_{ij}, \quad i = 1, \ldots, m.$$

- $\Rightarrow$ Class of Runge-Kutta methods for solving the ODE,

$$\begin{cases} F_{i,k} = f(t_{i,k}, x^k + (\Delta t) \sum_{j=1}^{m} a_{ij} F_{j,k}), \\ x^{k+1} = x^k + (\Delta t) \sum_{i=1}^{m} b_i F_{i,k}, \end{cases}$$

$t_{i,k} = t_k + c_i \Delta t$, or equivalently,

$$\begin{cases} x_{i,k} = x^k + (\Delta t) \sum_{j=1}^{m} a_{ij} f(t_{j,k}, x_{j,k}), \\ x^{k+1} = x^k + (\Delta t) \sum_{i=1}^{m} b_i f(t_{i,k}, x_{i,k}). \end{cases}$$

Numerical solution of ODEs

- Let

$$\kappa_j := f(t + c_j \Delta t, x_j);$$

- Define Φ by

$$\begin{cases} x_i = x + (\Delta t) \displaystyle\sum_{j=1}^{m} a_{ij} \kappa_j, \\ \Phi(t, x, \Delta t) = \displaystyle\sum_{i=1}^{m} b_i f(t + c_i \Delta t, x_i). \end{cases}$$

- $\Rightarrow$ One step method.
- If $a_{ij} = 0$ for $j \geq i$ $\Rightarrow$ scheme: explicit.

Numerical solution of ODEs

- EXAMPLES:
  - Explicit Euler's method and Trapezoidal scheme: Runge-Kutta methods.
  - Explicit Euler's method: $m = 1, b_1 = 1, a_{11} = 0.$

Numerical solution of ODEs

- Trapezoidal scheme:
  $m = 2, b_1 = b_2 = 1/2, a_{11} = a_{12} = 0, a_{21} = a_{22} = 1/2.$

Numerical solution of ODEs

- Fourth-order Runge-Kutta method: $m = 4$, $c_1 = 0$, $c_2 = c_3 = 1/2$, $c_4 = 1$, $b_1 = 1/6$, $b_2 = b_3 = 1/3$, $b_4 = 1/6$, $a_{21} = a_{32} = 1/2$, $a_{43} = 1$, and all the other a_{ij} entries are zero.

Numerical solution of ODEs

- Consistency, stability, convergence, and order of Runge-Kutta methods
- Runge-Kutta scheme: consistent iff

$$\sum_{j=1}^{m} b_j = 1.$$

Numerical solution of ODEs

- Stability:
  - $|A| = (|a_{ij}|)_{i,j=1}^m$.
  - **Spectral radius** $\rho(|A|)$ of the matrix $|A|$:

$$\rho(|A|) := \max\{|\lambda_j|, \lambda_j : \text{eigenvalue of } |A|\}.$$

Numerical solution of ODEs

- THEOREM:
  - C_f: Lipschitz constant for f.
  - Suppose
$$(\Delta t) C_f \rho(|A|) < 1.$$
  - Then the **Runge-Kutta method**: stable.

Numerical solution of ODEs

- PROOF:
  -

$$\Phi(t,x,\Delta t)-\Phi(t,y,\Delta t)=\sum_{i=1}^{m} b_i\Big[f(t+c_i\Delta t,x_i)-f(t+c_i\Delta t,y_i)\Big],$$

with

$$x_i = x+(\Delta t)\sum_{j=1}^{m} a_{ij} f(t+c_j\Delta t, x_j),$$

and

$$y_i = y+(\Delta t)\sum_{j=1}^{m} a_{ij} f(t+c_j\Delta t, y_j).$$

Numerical solution of ODEs

- $\Rightarrow$

$$x_i - y_i = x - y + (\Delta t)\sum_{j=1}^{m} a_{ij}\Big[f(t + c_j\Delta t, x_j) - f(t + c_j\Delta t, y_j)\Big].$$

- $\Rightarrow$ For $i = 1, \ldots, m$,

$$|x_i - y_i| \leq |x - y| + (\Delta t)C_f\sum_{j=1}^{m} |a_{ij}||x_j - y_j|.$$

Numerical solution of ODEs

- X and Y:

$$X = \begin{bmatrix} |x_1 - y_1| \\ \vdots \\ |x_m - y_m| \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} |x - y| \\ \vdots \\ |x - y| \end{bmatrix}.$$

- $X \leq Y + (\Delta t) C_f |A| X, \Rightarrow$

$$X \leq (I - (\Delta t) C_f |A|)^{-1} Y,$$

  provided that $(\Delta t) C_f \rho(|A|) < 1$.

- $\Rightarrow$ stability of the Runge-Kutta scheme.

Numerical solution of ODEs

- Dahlquist-Lax equivalence theorem $\Rightarrow$ Runge-Kutta scheme: convergent provided that $\sum_{j=1}^{m} b_j = 1$ and $(\Delta t) C_f \rho(|A|) < 1$ hold.

Numerical solution of ODEs

- Order of the Runge-Kutta scheme: compute the order as $\Delta t \to 0$ of the truncation error

$$T_k(\Delta t) = \frac{x(t_{k+1}) - x(t_k)}{\Delta t} - \Phi(t_k, x(t_k), \Delta t).$$

- Write

$$T_k(\Delta t) = \frac{x(t_{k+1}) - x(t_k)}{\Delta t} - \sum_{i=1}^{m} b_i f(t_k + c_i \Delta t, x(t_k) + \Delta t \sum_{j=1}^{m} a_{ij} \kappa_j).$$

- Suppose that f: smooth enough $\Rightarrow$

$$f(t_k + c_i \Delta t, x(t_k) + \Delta t \sum_{j=1}^{m} a_{ij} \kappa_j)$$

$$= f(t_k, x(t_k)) + \Delta t \left[c_i \frac{\partial f}{\partial t}(t_k, x(t_k)) + (\sum_{j=1} a_{ij} \kappa_j) \frac{\partial f}{\partial x}(t_k, x(t_k)) \right]$$

$$+ O((\Delta t)^2).$$

Numerical solution of ODEs

- $$\sum_{j=1} a_{ij}\kappa_j = (\sum_{j=1} a_{ij}) f(t_k, x(t_k)) + O(\Delta t) = c_i f(t_k, x(t_k)) + O(\Delta t).$$

Numerical solution of ODEs

- $\Rightarrow$

$$
\begin{aligned}
& f(t_k + c_i \Delta t, x(t_k) + \Delta t \sum_{j=1}^{m} a_{ij} \kappa_j) \\
& \quad = f(t_k, x(t_k)) + \Delta t c_i \left[\frac{\partial f}{\partial t}(t_k, x(t_k)) + \frac{\partial f}{\partial x}(t_k, x(t_k)) f(t_k, x(t_k)) \right] \\
& \qquad + O((\Delta t)^2).
\end{aligned}
$$

Numerical solution of ODEs

- THEOREM:
  - Assume that f: smooth enough.
  - Then the Runge-Kutta scheme: of order 2 provided that the conditions

$$\sum_{j=1}^{m} b_j = 1$$

and

$$\sum_{i=1}^{m} b_i c_i = \frac{1}{2}$$

hold.

Numerical solution of ODEs

- Higher-order Taylor expansions ⇒
- THEOREM:
    - Assume that f: smooth enough.
    - Then the Runge-Kutta scheme: of order 3 provided that the conditions

$$\sum_{j=1}^{m} b_j = 1,$$

$$\sum_{i=1}^{m} b_i c_i = \frac{1}{2},$$

and

$$\sum_{i=1}^{m} b_i c_i^2 = \frac{1}{3}, \quad \sum_{i=1}^{m} \sum_{j=1}^{m} b_i a_{ij} c_j = \frac{1}{6}$$

hold.

Numerical solution of ODEs

- Of Order 4 provided that **in addition**

$$\sum_{i=1}^m b_i c_i^3 = \frac{1}{4}, \quad \sum_{i=1}^m \sum_{j=1}^m b_i c_i a_{ij} c_j = \frac{1}{8}, \quad \sum_{i=1}^m \sum_{j=1}^m b_i c_i a_{ij} c_j^2 = \frac{1}{12},$$

$$\sum_{i=1}^m \sum_{j=1}^m \sum_{l=1}^m b_i a_{ij} a_{jl} c_l = \frac{1}{24}$$

  hold.

- The (fourth-order) Runge-Kutta scheme: of order 4.

Numerical solution of ODEs

- Multi-step methods
- Runge-Kutta methods: improvement over Euler's methods in terms of accuracy, but achieved by investing additional computational effort.
- The fourth-order Runge-Kutta method involves four function evaluations per step.

Numerical solution of ODEs

- For comparison, by considering three consecutive points t_{k-1}, t_k, t_{k+1}, integrating the differential equation between t_{k-1} and t_{k+1}, and applying **Simpson's rule** to approximate the resulting integral yields

$$x(t_{k+1}) = x(t_{k-1}) + \int_{t_{k-1}}^{t_{k+1}} f(s, x(s))\, ds$$

$$\approx x(t_{k-1}) + \frac{(\Delta t)}{3}\Big[f(t_{k-1}, x(t_{k-1})) + 4f(t_k, x(t_k)) + f(t_{k+1}, x(t_{k+1}))\Big],$$

$\Rightarrow$

$$x^{k+1} = x^{k-1} + \frac{(\Delta t)}{3}\Big[f(t_{k-1}, x^{k-1}) + 4f(t_k, x^k) + f(t_{k+1}, x^{k+1})\Big].$$

- Need two preceding values, x^k and x^{k-1} in order to calculate x^{k+1}: **two-step method**.
- In contrast with the one-step methods: only a single value of x^k required to compute the next approximation x^{k+1}.

Numerical solution of ODEs

- General n-step method:

$$\sum_{j=0}^{n} \alpha_j x^{k+j} = (\Delta t) \sum_{j=0}^{n} \beta_j f(t_{k+j}, x^{k+j}),$$

  α_j and β_j: real constants and $\alpha_n \neq 0$.

- If $\beta_n = 0$, then x^{k+n}: obtained explicitly from previous values of x^j and $f(t_j, x^j) \Rightarrow$ n-step method: **explicit**. Otherwise, the n-step method: **implicit**.

Numerical solution of ODEs

- EXAMPLE:

  (i) Two-step Adams-Bashforth method: explicit two-step method

$$x^{k+2} = x^{k+1} + \frac{(\Delta t)}{2}\Big[3f(t_{k+1}, x^{k+1}) - f(t_k, x^k)\Big];$$

  (ii) Three-step Adams-Bashforth method: explicit three-step method

$$x^{k+3} = x^{k+2} + \frac{(\Delta t)}{12}\Big[23f(t_{k+2}, x^{k+2}) - 16f(t_{k+1}, x^{k+1}) + f(t_k, x^k)\Big];$$

Numerical solution of ODEs

(iii) Four-step Adams-Bashforth method: explicit four-step method

$$x^{k+4} = x^{k+3} + \frac{(\Delta t)}{24}\Big[55f(t_{k+3}, x^{k+3}) - 59f(t_{k+2}, x^{k+2})$$

$$+37f(t_{k+1}, x^{k+1}) - 9f(t_k, x^k)\Big];$$

(iv) Two-step Adams-Moulton method: implicit two-step method

$$x^{k+2} = x^{k+1} + \frac{(\Delta t)}{12}\Big[5f(t_{k+2}, x^{k+2}) + 8f(t_{k+1}, x^{k+1}) + f(t_k, x^k)\Big];$$

(v) Three-step Adams-Moulton method: implicit three-step method

$$x^{k+3} = x^{k+2} + \frac{(\Delta t)}{24}\Big[9f(t_{k+3}, x^{k+3}) + 19f(t_{k+2}, x^{k+2}) - 5f(t_{k+1}, x^{k+1}) - 9f(t_k, x^k)\Big].$$

Numerical solution of ODEs

- Construction of linear multi-step methods
- Suppose that $x^k, k \in \mathbb{N}$: sequence of real numbers.
- Shift operator E, forward difference operator Δ_+ and backward difference operator Δ_-:

$$E : x^k \mapsto x^{k+1}, \quad \Delta_+ : x^k \mapsto x^{k+1} - x^k, \quad \Delta_- : x^k \mapsto x^k - x^{k-1}.$$

- $\Delta_+ = E - I$ and $\Delta_- = I - E^{-1} \Rightarrow$ for any $n \in \mathbb{N}$,

$$(E - I)^n = \sum_{j=0}^{n} (-1)^j C_j^n E^{n-j}$$

and

$$(I - E^{-1})^n = \sum_{j=0}^{n} (-1)^j C_j^n E^{-j}.$$

Numerical solution of ODEs

- $\Rightarrow$

$$\Delta_+^n x^k = \sum_{j=0}^{n} (-1)^j C_j^n x^{k+n-j}$$

and

$$\Delta_-^n x^k = \sum_{j=0}^{n} (-1)^j C_j^n x^{k-j}.$$

Numerical solution of ODEs

- $y(t) \in \mathcal{C}^\infty(\mathbb{R})$; $t_k = k\Delta t, \Delta t > 0$.
- Taylor series $\Rightarrow$ for any $s \in \mathbb{N}$,

$$E^s y(t_k) = y(t_k + s\Delta t) = \Big(\sum_{l=0}^{+\infty} \frac{1}{l!}(s\Delta t \frac{\partial}{\partial t})^l y\Big)(t_k) = \big(e^{s(\Delta t)\frac{\partial}{\partial t}} y\big)(t_k),$$

- $\Rightarrow$

$$E^s = e^{s(\Delta t)\frac{\partial}{\partial t}}.$$

- Formally,

$$(\Delta t)\frac{\partial}{\partial t} = \ln E = -\ln(I - \Delta_-) = \Delta_- + \frac{1}{2}\Delta_-^2 + \frac{1}{3}\Delta_-^3 + \ldots$$

Numerical solution of ODEs

- $x(t)$: solution of ODE:

$$(\Delta t) f(t_k, x(t_k)) = \left(\Delta_- + \frac{1}{2}\Delta_-^2 + \frac{1}{3}\Delta_-^3 + \ldots \right) x(t_k).$$

- Successive truncation of the infinite series $\Rightarrow$

$$x^k - x^{k-1} = (\Delta t) f(t_k, x^k),$$

$$\frac{3}{2}x^k - 2x^{k-1} + \frac{1}{2}x^{k-2} = (\Delta t) f(t_k, x^k),$$

$$\frac{11}{6}x^k - 3x^{k-1} + \frac{3}{2}x^{k-2} - \frac{1}{3}x^{k-3} = (\Delta t) f(t_k, x^k),$$

  and so on.

- Class of implicit multi-step methods: backward differentiation formulas.

Numerical solution of ODEs

- Similarly,

$$E^{-1}((\Delta t)\frac{\partial}{\partial t}) = (\Delta t)\frac{\partial}{\partial t}E^{-1} = -(I - \Delta_-)\ln(I - \Delta_-).$$

- $\Rightarrow$

$$((\Delta t)\frac{\partial}{\partial t}) = -E(I - \Delta_-)\ln(I - \Delta_-) = -(I - \Delta_-)\ln(I - \Delta_-)E.$$

- $\Rightarrow$

$$(\Delta t)f(t_k, x(t_k)) = \left(\Delta_- - \frac{1}{2}\Delta_-^2 - \frac{1}{6}\Delta_-^3 + \dots\right)x(t_{k+1}).$$

Numerical solution of ODEs

- Successive truncation of the infinite series $\Rightarrow$ explicit numerical schemes:

$$
\begin{aligned}
& x^{k+1} - x^{k} = (\Delta t) f(t_k, x^k), \\
& \frac{1}{2} x^{k+1} - \frac{1}{2} x^{k-1} = (\Delta t) f(t_k, x^k), \\
& \frac{1}{3} x^{k+1} + \frac{1}{2} x^{k} - x^{k-1} + \frac{1}{6} x^{k-2} = (\Delta t) f(t_k, x^k), \\
& \vdots
\end{aligned}
$$

- The first of these numerical scheme: explicit Euler method, while the second: explicit mid-point method.

Numerical solution of ODEs

- Construct further classes of multi-step methods:
- For $y \in \mathcal{C}^\infty$,
$$D^{-1}y(t_k) = y(t_0) + \int_{t_0}^{t_k} y(s)\,ds,$$
and
$$(E - I)D^{-1}y(t_k) = \int_{t_k}^{t_{k+1}} y(s)\,ds.$$
- $$(E - I)D^{-1} = \Delta_+ D^{-1} = E\Delta_- D^{-1} = (\Delta t)E\Delta_-((\Delta t)D)^{-1},$$

Numerical solution of ODEs

- $\Rightarrow$

$$(E - I)D^{-1} = -(\Delta t)E\Delta_-\big(\ln(I - \Delta_-)\big)^{-1}.$$

Numerical solution of ODEs

- $$(E-I)D^{-1} = E\Delta_- D^{-1} = \Delta_- E D^{-1} = \Delta_-(DE^{-1})^{-1} = (\Delta t)\Delta_-\left((\Delta t)DE^{-1}\right)^{-1}.$$

- $\Rightarrow$
$$(E-I)D^{-1} = -(\Delta t)\Delta_-\Big((I-\Delta_-)\ln(I-\Delta_-)\Big)^{-1}.$$

Numerical solution of ODEs

- $$x(t_{k+1}) - x(t_k) = \int_{t_k}^{t_{k+1}} f(s, x(s))\, ds = (E - I)D^{-1} f(t_k, x(t_k)),$$
- $\Rightarrow$

$$x(t_{k+1}) - x(t_k) = \begin{cases} -(\Delta t)\Delta_- \big((I - \Delta_-)\ln(I - \Delta_-)\big)^{-1} f(t_k, x(t_k)) \\ -(\Delta t) E \Delta_- \big(\ln(I - \Delta_-)\big)^{-1} f(t_k, x(t_k)). \end{cases}$$

Numerical solution of ODEs

- Expand $\ln(I - \Delta_-)$ into a Taylor series on the right-hand side $\Rightarrow$

$$x(t_{k+1}) - x(t_k) = (\Delta t)\left[I + \frac{1}{2}\Delta_- + \frac{5}{12}\Delta_-^2 + \frac{3}{8}\Delta_-^3 + \dots\right] f(t_k, x(t_k))$$

and

$$x(t_{k+1}) - x(t_k) = (\Delta t)\left[I - \frac{1}{2}\Delta_- - \frac{1}{12}\Delta_-^2 - \frac{1}{24}\Delta_-^3 + \dots\right] f(t_{k+1}, x(t_{k+1})).$$

- Successive truncations $\Rightarrow$ families of (explicit) Adams-Bashforth methods and of (implicit) Adams-Moulton methods.

Numerical solution of ODEs

- Consistency, stability, and convergence
- Introduce the concepts of consistency, stability, and convergence for analyzing linear multi-step methods.

Numerical solution of ODEs

- DEFINITION: Consistency
  - The n-step method: **consistent** with the ODE if the truncation error defined by

$$T_k = \frac{\sum_{j=0}^{n} \left[\alpha_j x(t_{k+j}) - (\Delta t)\beta_j \frac{dx}{dt}(t_{k+j})\right]}{(\Delta t)\sum_{j=0}^{n} \beta_j}$$

is s.t. for any $\epsilon > 0$ there exists h_0 for which

$$|T_k| \leq \epsilon \quad \text{for } 0 < \Delta t \leq h_0$$

and any $(n+1)$ points $\big((t_j, x(t_j)), \ldots, (t_{j+n}, x(t_{j+n}))\big)$ on any solution $x(t)$.

Numerical solution of ODEs

- DEFINITION: Stability
  - The n-step method: stable if there exists a constant C s.t., for any two sequences (x^k) and $(\widetilde{x}^k)$ which have been generated by the same formulas but different initial data $x^0, x^1, \ldots, x^{k-1}$ and $\widetilde{x}^0, \widetilde{x}^1, \ldots, \widetilde{x}^{k-1}$, respectively,

$$|x^k - \widetilde{x}^k| \leq C \max\{|x^0 - \widetilde{x}^0|, |x^1 - \widetilde{x}^1|, \ldots, |x^{k-1} - \widetilde{x}^{k-1}|\}$$

as $\Delta t \to 0$.

Numerical solution of ODEs

- THEOREM: Convergence
  - Suppose that the n-step method: consistent with the ODE.
  - Stability condition: necessary and sufficient for the convergence.
  - If $x \in \mathcal{C}^{p+1}$ and the truncation error $O((\Delta t)^p)$, then the global error $e_k = x(t_k) - x^k$: $O((\Delta t)^p)$.

Numerical solution of ODEs

- Stiff equations and systems:
- Let $\epsilon > 0$: small parameter. Consider the initial value problem

$$\begin{cases} \frac{\mathrm{d}x(t)}{\mathrm{d}t} = -\frac{1}{\epsilon}x(t), \quad t \in [0, T], \\ x(0) = 1, \end{cases}$$

- Exponential solution $x(t) = e^{-t/\epsilon}$.
- Explicit Euler method with step size Δt:

$$x^{k+1} = (1 - \frac{\Delta t}{\epsilon})x^k, \quad x^0 = 1,$$

with solution

$$x^k = (1 - \frac{\Delta t}{\epsilon})^k.$$

Numerical solution of ODEs

- $\epsilon > 0 \Rightarrow$ exact solution: exponentially decaying and positive.
- If $1 - \frac{\Delta t}{\epsilon} < -1$, then the iterates grow exponentially fast in magnitude, with alternating signs.
- Numerical solution: nowhere close to the true solution.
- If $-1 < 1 - \frac{\Delta t}{\epsilon} < 0$, then the numerical solution decays in magnitude, but continue to alternate between positive and negative values.
- To correctly model the qualitative features of the solution and obtain a numerically accurate solution: choose the step size Δt so as to ensure that $1 - \frac{\Delta t}{\epsilon} > 0$, and hence $\Delta t < \epsilon$.
- stiff differential equation.

Numerical solution of ODEs

- In general, an equation or system: stiff if it has one or more very rapidly decaying solutions.
- In the case of the autonomous constant coefficient linear system: stiffness occurs whenever the coefficient matrix A has an eigenvalues λ_{j_0} with large negative real part: $\Re\lambda_{j_0} \ll 0$, resulting in a very rapidly decaying eigensolution.
- It only takes one such eigensolution to render the equation stiff, and ruin the numerical computation of even well behaved solutions.
- Even though the component of the actual solution corresponding to λ_{j_0}: almost irrelevant, its presence continues to render the numerical solution to the system very difficult.
- Most of the numerical methods: suffer from instability due to stiffness for sufficiently small positive ϵ.
- Stiff equations require more sophisticated numerical schemes to integrate.

Numerical solution of ODEs

- Perturbation theories for differential equations
  - Regular perturbation theory;
  - Singular perturbation theory.

Numerical solution of ODEs

- Regular perturbation theory:
- $\epsilon > 0$: small parameter and consider

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = f(t, x, \epsilon), \quad t \in [0, T], \\ x(0) = x_0, \quad x_0 \in \mathbb{R}. \end{cases}$$

- $f \in \mathcal{C}^1 \Rightarrow$ regular perturbation problem.
- Taylor expansion of $x(t, \epsilon) \in \mathcal{C}^1$:

$$x(t, \epsilon) = x^{(0)}(t) + \epsilon x^{(1)}(t) + o(\epsilon)$$

with respect to ϵ in a neighborhood of 0.

Numerical solution of ODEs

- $x^{(0)}$:

$$\begin{cases} \dfrac{\mathrm{d}x^{(0)}}{\mathrm{d}t} = f_0(t, x^{(0)}), \quad t \in [0, T], \\ x^{(0)}(0) = x_0, \quad x_0 \in \mathbb{R}, \end{cases}$$

  $f_0(t, x) := f(t, x, 0)$.

- $x^{(1)}(t) = \frac{\partial x}{\partial \epsilon}(t, 0)$:

$$\begin{cases} \dfrac{\mathrm{d}x^{(1)}}{\mathrm{d}t} = \dfrac{\partial f}{\partial x}(t, x^{(0)}, 0)x^{(1)} + \dfrac{\partial f}{\partial \epsilon}(t, x^{(0)}, 0), \quad t \in [0, T], \\ x^{(1)}(0) = 0. \end{cases}$$

- Compute numerically $x^{(0)}$ and $x^{(1)}$.

Numerical solution of ODEs

- Singular perturbation theory:
- Consider
$$\begin{cases} \epsilon \dfrac{d^2 x}{dt^2} = f(t, x, \dfrac{dx}{dt}), \quad t \in [0, T], \\ x(0) = x_0, \quad x(T) = x_1. \end{cases}$$
- Singular perturbation problem: order reduction when $\epsilon = 0$.

Numerical solution of ODEs

- Consider the linear, scalar and of second-order ODE:

$$\begin{cases} \epsilon \dfrac{d^2x}{dt^2} + 2\dfrac{dx}{dt} + x = 0, \quad t \in [0,1], \\ x(0) = 0, \quad x(1) = 1. \end{cases}$$

-

$$\alpha(\epsilon) := \frac{1-\sqrt{1-\epsilon}}{\epsilon} \quad \text{and} \quad \beta(\epsilon) := 1 + \sqrt{1-\epsilon}.$$

-

$$x(t,\epsilon) = \frac{e^{-\alpha t} - e^{-\beta t/\epsilon}}{e^{-\alpha} - e^{-\beta/\epsilon}}, \quad t \in [0,1].$$

- $x(t,\epsilon)$: involves two terms which vary on widely different length-scales.

Numerical solution of ODEs

- Behavior of $x(t, \epsilon)$ as $\epsilon \to 0^+$.
- Asymptotic behavior: nonuniform;
- There are two cases $\to$ matching outer and inner solutions.

Numerical solution of ODEs

(i) Outer limit: $t > 0$ fixed and $\epsilon \to 0^+$. Then $x(t,\epsilon) \to x^{(0)}(t)$,

$$x^{(0)}(t) := e^{(1-t)/2}.$$

- Leading-order outer solution satisfies the boundary condition at $t = 1$ but not the boundary condition at $t = 0$. Indeed, $x^{(0)}(0) = e^{1/2}$.

(ii) Inner limit: $t/\epsilon = \tau$ fixed and $\epsilon \to 0^+$. Then $x(\epsilon\tau,\epsilon) \to X^{(0)}(\tau) := e^{1/2}(1 - e^{-2\tau})$.

- Leading-order inner solution satisfies the boundary condition at $t = 0$ but not the one at $t = 1$, which corresponds to $\tau = 1/\epsilon$. Indeed, $\lim_{\tau\to+\infty} X^{(0)}(\tau) = e^{1/2}$.

(iii) Matching: Both the inner and outer expansions: valid in the region $\epsilon \ll t \ll 1$, corresponding to $t \to 0$ and $\tau \to +\infty$ as $\epsilon \to 0^+$. They satisfy the matching condition

$$\lim_{t\to 0^+} x^{(0)}(t) = \lim_{\tau\to+\infty} X^{(0)}(\tau).$$

Numerical solution of ODEs

- Construct an asymptotic solution without relying on the fact that we can solve it exactly.
- Outer solution:

$$x(t, \epsilon) = x^{(0)}(t) + \epsilon x^{(1)}(t) + O(\epsilon^2).$$

- Use this expansion and equate the coefficients of the leading-order terms to zero.
- $\Rightarrow$

$$\begin{cases} 2\dfrac{dx^{(0)}}{dt} + x^{(0)} = 0, \quad t \in [0, 1], \\ x^{(0)}(1) = 1. \end{cases}$$

Numerical solution of ODEs

- Inner solution.
- Suppose that there is a boundary layer at $t = 0$ of width $\delta(\epsilon)$, and introduce a stretched variable $\tau = t/\delta$.
- Look for an inner solution $X(\tau, \epsilon) = x(t, \epsilon)$.

Numerical solution of ODEs

- $$\frac{d}{dt} = \frac{1}{\delta}\frac{d}{d\tau},$$

  $\Rightarrow X$ satisfies

  $$\frac{\epsilon}{\delta^2}\frac{d^2X}{d\tau^2} + \frac{2}{\delta}\frac{dX}{d\tau} + X = 0.$$

- Two possible dominant balances:

  (i) $\delta = 1$, leading to the outer solution;

  (ii) $\delta = \epsilon$, leading to the inner solution.

- $\Rightarrow$ Boundary layer thickness: of the order of ϵ, and the appropriate inner variable: $\tau = t/\epsilon$.

Numerical solution of ODEs

- Equation for X:

$$\begin{cases} \dfrac{d^2X}{d\tau^2} + 2\dfrac{dX}{d\tau} + \epsilon X = 0, \\ X(0,\epsilon) = 0. \end{cases}$$

- Impose only the boundary condition at $\tau = 0$, since we do not expect the inner expansion to be valid outside the boundary layer where $t = O(\epsilon)$.
- Seek an inner expansion

$$X(\tau,\epsilon) = X^{(0)}(\tau) + \epsilon X^{(1)}(\tau) + O(\epsilon^2)$$

and find that

$$\begin{cases} \dfrac{d^2X^{(0)}}{d\tau^2} + 2\dfrac{dX^{(0)}}{d\tau} = 0, \\ X^{(0)}(0) = 0. \end{cases}$$

Numerical solution of ODEs

- General solution:

$$X^{(0)}(\tau) = c(1 - e^{-2\tau}),$$

c: arbitrary constant of integration.

- Determine the unknown constant c by requiring that the inner solution matches with the outer solution.
- Matching condition:

$$\lim_{t \to 0^+} x^{(0)}(t) = \lim_{\tau \to +\infty} X^{(0)}(\tau),$$

$\Rightarrow c = e^{1/2}$.

Numerical solution of ODEs

- Asymptotic solution as $\epsilon \to 0^+$:

$$x(t,\epsilon) = \begin{cases} e^{1/2}(1 - e^{-2\tau}) & \text{as } \epsilon \to 0^+ \text{ with } t/\epsilon \text{ fixed,} \\ e^{(1-t)/2} & \text{as } \epsilon \to 0^+ \text{ with } t \text{ fixed.} \end{cases}$$