Conservative, Symplectic, and Exponential Integrators

John C. Bowman
Department of Mathematical and Statistical Sciences
University of Alberta

December 1, 2024

www.math.ualberta.ca/∼bowman/talks

Outline

- Symplectic Integrators
- Conservative Integrators
- Exponential Integrators
  - Robust Embedded Pairs
  - Schur Decomposition
- Conclusions

Initial Value Problems

- Given $\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, suppose $\boldsymbol{x} \in \mathbb{R}^n$ evolves according to

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t),$$

with the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$.

Initial Value Problems

- Given $\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, suppose $\boldsymbol{x} \in \mathbb{R}^n$ evolves according to

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t),$$

with the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$.

- Hamiltonian subclass: $n = 2k$ and $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$, where $\boldsymbol{q}, \boldsymbol{p} \in \mathbb{R}^k$ satisfy

$$\frac{d\boldsymbol{q}}{dt} = \frac{\partial H}{\partial \boldsymbol{p}},$$

$$\frac{d\boldsymbol{p}}{dt} = -\frac{\partial H}{\partial \boldsymbol{q}},$$

for some function $H(\boldsymbol{q}, \boldsymbol{p}, t) : \mathbb{R}^{n+1} \to \mathbb{R}$.

Initial Value Problems

- Given $\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, suppose $\boldsymbol{x} \in \mathbb{R}^n$ evolves according to

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t),$$

  with the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$.

- Hamiltonian subclass: $n = 2k$ and $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$, where $\boldsymbol{q}, \boldsymbol{p} \in \mathbb{R}^k$ satisfy

$$\frac{d\boldsymbol{q}}{dt} = \frac{\partial H}{\partial \boldsymbol{p}},$$

$$\frac{d\boldsymbol{p}}{dt} = -\frac{\partial H}{\partial \boldsymbol{q}},$$

  for some function $H(\boldsymbol{q}, \boldsymbol{p}, t) : \mathbb{R}^{n+1} \to \mathbb{R}$.

- Often, the Hamiltonian H has no explicit dependence on t.

Structure-Preserving Discretizations

- Symplectic integration: conserves phase space structure of Hamilton's equations; the time step map is a canonical transformation. [Ruth 1983], [Channell & Scovel 1990], [Sanz-Serna & Calvo 1994]

Structure-Preserving Discretizations

- Symplectic integration: conserves phase space structure of Hamilton's equations; the time step map is a canonical transformation. [Ruth 1983], [Channell & Scovel 1990], [Sanz-Serna & Calvo 1994]

- Conservative integration: conserves first integrals. [Bowman *et al.* 1997], [Shadwick *et al.* 1999], [Kotovych & Bowman 2002], [Wan *et al.* 2017]

Structure-Preserving Discretizations

- Symplectic integration: conserves phase space structure of Hamilton's equations; the time step map is a canonical transformation. [Ruth 1983], [Channell & Scovel 1990], [Sanz-Serna & Calvo 1994]
- Conservative integration: conserves first integrals. [Bowman *et al.* 1997], [Shadwick *et al.* 1999], [Kotovych & Bowman 2002], [Wan *et al.* 2017]
- Positivity: preserves positive semi-definiteness of covariance matrices. [Bowman & Krommes 1997]

Structure-Preserving Discretizations

- Symplectic integration: conserves phase space structure of Hamilton's equations; the time step map is a canonical transformation. [Ruth 1983], [Channell & Scovel 1990], [Sanz-Serna & Calvo 1994]

- Conservative integration: conserves first integrals. [Bowman *et al.* 1997], [Shadwick *et al.* 1999], [Kotovych & Bowman 2002], [Wan *et al.* 2017]

- Positivity: preserves positive semi-definiteness of covariance matrices. [Bowman & Krommes 1997]

- Unitary integration: conserves trace of probability density matrix. [Shadwick & Buell 1997]

Structure-Preserving Discretizations

- Symplectic integration: conserves phase space structure of Hamilton's equations; the time step map is a canonical transformation. [Ruth 1983], [Channell & Scovel 1990], [Sanz-Serna & Calvo 1994]

- Conservative integration: conserves first integrals. [Bowman *et al.* 1997], [Shadwick *et al.* 1999], [Kotovych & Bowman 2002], [Wan *et al.* 2017]

- Positivity: preserves positive semi-definiteness of covariance matrices. [Bowman & Krommes 1997]

- Unitary integration: conserves trace of probability density matrix. [Shadwick & Buell 1997]

- Exponential integrators: Yield exact evolution on linear time scale

Symplectic vs. Conservative Integration

Theorem: (Ge and Marsden 1988) A C^1 symplectic map M with no explicit time-dependence will conserve a C^1 time-independent Hamiltonian $H : \mathbb{R}^n \to \mathbb{R} \iff M$ is identical to the exact evolution, up to a reparametrization of time.

Proof:

- A C^1 symplectic scheme is a canonical map M corresponding to some approximate C^1 Hamiltonian $\tilde{H}_{\tau(\boldsymbol{x},t)} : \mathbb{R}^{n+1} \to \mathbb{R}$, where the label τ denotes the time step.

Symplectic vs. Conservative Integration

Theorem: (Ge and Marsden 1988) A C^1 symplectic map M with no explicit time-dependence will conserve a C^1 time-independent Hamiltonian $H : \mathbb{R}^n \to \mathbb{R} \iff M$ is identical to the exact evolution, up to a reparametrization of time.

Proof:

- A C^1 symplectic scheme is a canonical map M corresponding to some approximate C^1 Hamiltonian $\tilde{H}_{\tau(\boldsymbol{x},t)} : \mathbb{R}^{n+1} \to \mathbb{R}$, where the label τ denotes the time step.
- If the mapping M does not depend explicitly on time, it can be generated by the approximate Hamiltonian $K(\boldsymbol{x}) = \tilde{H}_\tau(\boldsymbol{x}, 0)$.

- Suppose the symplectic map conserves the true Hamiltonian H:

$$0 = \frac{dH}{dt} = \frac{\partial H}{\partial q_i}\frac{dq_i}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i}{dt} + \cancel{\frac{\partial H}{\partial t}} = [H, K],$$

where

$$[H, K] = \frac{\partial H}{\partial q_i}\frac{\partial K}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial K}{\partial q_i}.$$

- Suppose the symplectic map conserves the true Hamiltonian H:

$$0 = \frac{dH}{dt} = \frac{\partial H}{\partial q_i}\frac{dq_i}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i}{dt} + \cancel{\frac{\partial H}{\partial t}} = [H, K],$$

where

$$[H, K] = \frac{\partial H}{\partial q_i}\frac{\partial K}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial K}{\partial q_i}.$$

- Implicit function theorem: in a neighbourhood of $\boldsymbol{x}_0 \in \mathbb{R}^n$ $\exists$ a C^1 function $\phi : \mathbb{R} \to \mathbb{R} \ni$

$$H(\boldsymbol{x}) = \phi(K(\boldsymbol{x})) \quad \text{or} \quad K(\boldsymbol{x}) = \phi(H(\boldsymbol{x})) \iff [H, K] = 0.$$

- Suppose the symplectic map conserves the true Hamiltonian H:

$$0 = \frac{dH}{dt} = \frac{\partial H}{\partial q_i}\frac{dq_i}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i}{dt} + \frac{\partial H\!\!\!/}{\partial t\!\!\!/} = [H, K],$$

where

$$[H, K] = \frac{\partial H}{\partial q_i}\frac{\partial K}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial K}{\partial q_i}.$$

- Implicit function theorem: in a neighbourhood of $\boldsymbol{x}_0 \in \mathbb{R}^n$ $\exists$ a C^1 function $\phi : \mathbb{R} \to \mathbb{R} \ni$

$$H(\boldsymbol{x}) = \phi(K(\boldsymbol{x})) \quad \text{or} \quad K(\boldsymbol{x}) = \phi(H(\boldsymbol{x})) \iff [H, K] = 0.$$

- Consequently, the trajectories in $\mathbb{R}^n$ generated by the Hamiltonians H and K coincide.

Conservative Integration

- Traditional numerical discretizations of nonlinear initial value problems are based on polynomial functions of the time step.

Conservative Integration

- Traditional numerical discretizations of nonlinear initial value problems are based on polynomial functions of the time step.

- They typically yield spurious secular drifts of nonlinear first integrals of motion (e.g. total energy).

  $\Rightarrow$ the numerical solution will *not* remain on the energy surface defined by the initial conditions!

Conservative Integration

- Traditional numerical discretizations of nonlinear initial value problems are based on polynomial functions of the time step.
- They typically yield spurious secular drifts of nonlinear first integrals of motion (e.g. total energy).

  $\Rightarrow$ the numerical solution will *not* remain on the energy surface defined by the initial conditions!
- There exists a class of nontraditional explicit algorithms that exactly conserve nonlinear invariants to *all orders* in the time step (to machine precision).

Three-Wave Problem

- Truncated Fourier-transformed Euler equations for an inviscid 2D fluid:

$$\frac{dx_1}{dt} = f_1 = M_1 x_2 x_3,$$
$$\frac{dx_2}{dt} = f_2 = M_2 x_3 x_1,$$
$$\frac{dx_3}{dt} = f_3 = M_3 x_1 x_2,$$

where $M_1 + M_2 + M_3 = 0$.

Three-Wave Problem

- Truncated Fourier-transformed Euler equations for an inviscid 2D fluid:

$$\frac{dx_1}{dt} = f_1 = M_1 x_2 x_3,$$
$$\frac{dx_2}{dt} = f_2 = M_2 x_3 x_1,$$
$$\frac{dx_3}{dt} = f_3 = M_3 x_1 x_2,$$

where $M_1 + M_2 + M_3 = 0$.

- Then

$$\sum_k f_k x_k = 0 \Rightarrow \text{ energy } E \doteq \frac{1}{2}\sum_k x_k^2 \text{ is conserved.}$$

Secular Energy Growth

- Energy is not conserved by conventional discretizations.

Secular Energy Growth

- Energy is not conserved by conventional discretizations.
- The Euler method

$$x_{k,i+1} = x_{k,i} + hf_k$$

yields a monotonically increasing energy:

$$\begin{aligned} E_{i+1} &= \frac{1}{2} \sum_k \left[x_k^2 + 2hf_k x_k + h^2 S_k^2 \right] \\ &= E(t) + \frac{1}{2} h^2 \sum_k S_k^2. \end{aligned}$$

Conservative Euler Algorithm

- Determine a modification of the original equations of motion leading to *exact* energy conservation:

$$\frac{dx_k}{dt} = f_k + g_k.$$

Conservative Euler Algorithm

- Determine a modification of the original equations of motion leading to *exact* energy conservation:

$$\frac{dx_k}{dt} = f_k + g_k.$$

- Euler's method predicts the new energy

$$\begin{aligned} E_{i+1} &= \frac{1}{2}\sum_k \left[x_{k,i} + h(f_k + g_k)\right]^2 \\ &= E_i + \frac{1}{2}\sum_k \underbrace{\left[2hg_k x_{k,i} + h^2(f_k + g_k)^2\right]}_{\text{set to 0}}. \end{aligned}$$

- Solving for g_k yields the C–Euler discretization:

$$x_{k,i+1} = \operatorname{sgn} x_{k,i+1} \sqrt{x_{k,i}^2 + 2hf_k x_{k,i}}.$$

- Solving for g_k yields the C–Euler discretization:

$$x_{k,i+1} = \operatorname{sgn} x_{k,i+1} \sqrt{x_{k,i}^2 + 2hf_k x_{k,i}}.$$

- Reduces to Euler's method as $h \to 0$:

$$\begin{aligned} x_{k,i+1} &= x_{k,i}\sqrt{1 + 2h\frac{f_k}{x_{k,i}}} \\ &= x_{k,i} + hf_k + \mathcal{O}(h^2). \end{aligned}$$

- Solving for g_k yields the C–Euler discretization:

$$x_{k,i+1} = \operatorname{sgn} x_{k,i+1} \sqrt{x_{k,i}^2 + 2hf_k x_{k,i}}.$$

- Reduces to Euler's method as $h \to 0$:

$$\begin{aligned} x_{k,i+1} &= x_{k,i}\sqrt{1 + 2h\frac{f_k}{x_{k,i}}} \\ &= x_{k,i} + hf_k + \mathcal{O}(h^2). \end{aligned}$$

- C–Euler is just the usual Euler algorithm applied to

$$\frac{dx_k^2}{dt} = 2f_k x_k.$$

Lemma: ([Shampine 1986]) Let $\boldsymbol{x}$ and $\boldsymbol{c}$ be vectors in $\mathbb{R}^n$. If $\boldsymbol{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ has values orthogonal to $\boldsymbol{c}$, so that $I = \boldsymbol{c} \cdot \boldsymbol{x}$ is a linear invariant of

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t),$$

then each stage of the explicit s-stage discretization

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_0 + h \sum_{j=0}^{i} a_{ij} \boldsymbol{f}(\boldsymbol{x}_j, t + a_i h), \qquad i = 0, \ldots, s-1,$$

also conserves I, where h is the time step and $a_{ij} \in \mathbb{R}$.

Higher-Order Conservative Integration

- Find a transformation $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^n$ such that the nonlinear invariants are linear functions of $\boldsymbol{\xi} = \boldsymbol{T}(\boldsymbol{x})$.

Higher-Order Conservative Integration

- Find a transformation $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^n$ such that the nonlinear invariants are linear functions of $\boldsymbol{\xi} = \boldsymbol{T}(\boldsymbol{x})$.
- The new value of $\boldsymbol{x}$ is then obtained by inverse transformation:

$$\boldsymbol{x}_{i+1} = \boldsymbol{T}^{-1}(\boldsymbol{\xi}_{i+1}).$$

Higher-Order Conservative Integration

- Find a transformation $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^n$ such that the nonlinear invariants are linear functions of $\boldsymbol{\xi} = \boldsymbol{T}(\boldsymbol{x})$.

- The new value of $\boldsymbol{x}$ is then obtained by inverse transformation:

$$\boldsymbol{x}_{i+1} = \boldsymbol{T}^{-1}(\boldsymbol{\xi}_{i+1}).$$

- Problem: T may not be invertible!

  – Solution 1: Reduce the time step.

  – Solution 2: Use a traditional integrator for that time step.

  – Solution 3: Use an implicit backwards step [Shadwick & Bowman SIAM J. Appl. Math. **59**, 1112 (1999), Appendix A].

Higher-Order Conservative Integration

- Find a transformation $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^n$ such that the nonlinear invariants are linear functions of $\boldsymbol{\xi} = \boldsymbol{T}(\boldsymbol{x})$.

- The new value of $\boldsymbol{x}$ is then obtained by inverse transformation:

$$\boldsymbol{x}_{i+1} = \boldsymbol{T}^{-1}(\boldsymbol{\xi}_{i+1}).$$

- Problem: T may not be invertible!

  – Solution 1: Reduce the time step.

  – Solution 2: Use a traditional integrator for that time step.

  – Solution 3: Use an implicit backwards step [Shadwick & Bowman SIAM J. Appl. Math. **59**, 1112 (1999), Appendix A].

- Only the final corrector stage needs to be computed in the transformed space.

Error Analysis: 1D Autonomous Case

- Exact solution (everything on RHS evaluated at x_i):

$$x_{i+1} = x_i + hf + \frac{h^2}{2} f' f + \frac{h^3}{6} (f'' f^2 + f'^2 f) + \mathcal{O}(h^4);$$

Error Analysis: 1D Autonomous Case

- Exact solution (everything on RHS evaluated at x_i):

$$x_{i+1} = x_i + hf + \frac{h^2}{2} f' f + \frac{h^3}{6}(f'' f^2 + f'^2 f) + \mathcal{O}(h^4);$$

- When $T'(x_i) \neq 0$, C–PC yields the solution

$$x_{i+1} = x_i + hf + \frac{h^2}{2} f' f + \frac{h^3}{4}\left(f'' f^2 + \frac{T'''}{3T'} f^3 \right) + \mathcal{O}(h^4),$$

where all of the derivatives are evaluated at x_i.

Error Analysis: 1D Autonomous Case

- Exact solution (everything on RHS evaluated at x_i):

$$x_{i+1} = x_i + hf + \frac{h^2}{2}f'f + \frac{h^3}{6}(f''f^2 + f'^2 f) + \mathcal{O}(h^4);$$

- When $T'(x_i) \neq 0$, C–PC yields the solution

$$x_{i+1} = x_i + hf + \frac{h^2}{2}f'f + \frac{h^3}{4}\left(f''f^2 + \frac{T'''}{3T'}f^3\right) + \mathcal{O}(h^4),$$

where all of the derivatives are evaluated at x_i.

- On setting $T(x) = x$, the C–PC solution reduces to the conventional PC.

Error Analysis: 1D Autonomous Case

- Exact solution (everything on RHS evaluated at x_i):

$$x_{i+1} = x_i + hf + \frac{h^2}{2} f' f + \frac{h^3}{6} (f'' f^2 + f'^2 f) + \mathcal{O}(h^4);$$

- When $T'(x_i) \neq 0$, C–PC yields the solution

$$x_{i+1} = x_i + hf + \frac{h^2}{2} f' f + \frac{h^3}{4} \left(f'' f^2 + \frac{T'''}{3T'} f^3 \right) + \mathcal{O}(h^4),$$

where all of the derivatives are evaluated at x_i.

- On setting $T(x) = x$, the C–PC solution reduces to the conventional PC.

- C–PC and PC are both accurate to second order in h; for $T(x) = x^2$, they agree through third order in h.

Singular Case

- When $T'(x_i) = 0$, the conservative corrector reduces to

$$x_{i+1} = T^{-1}\left(T(x_i) + \frac{h}{2}T'(\tilde{x})f(\tilde{x})\right),$$

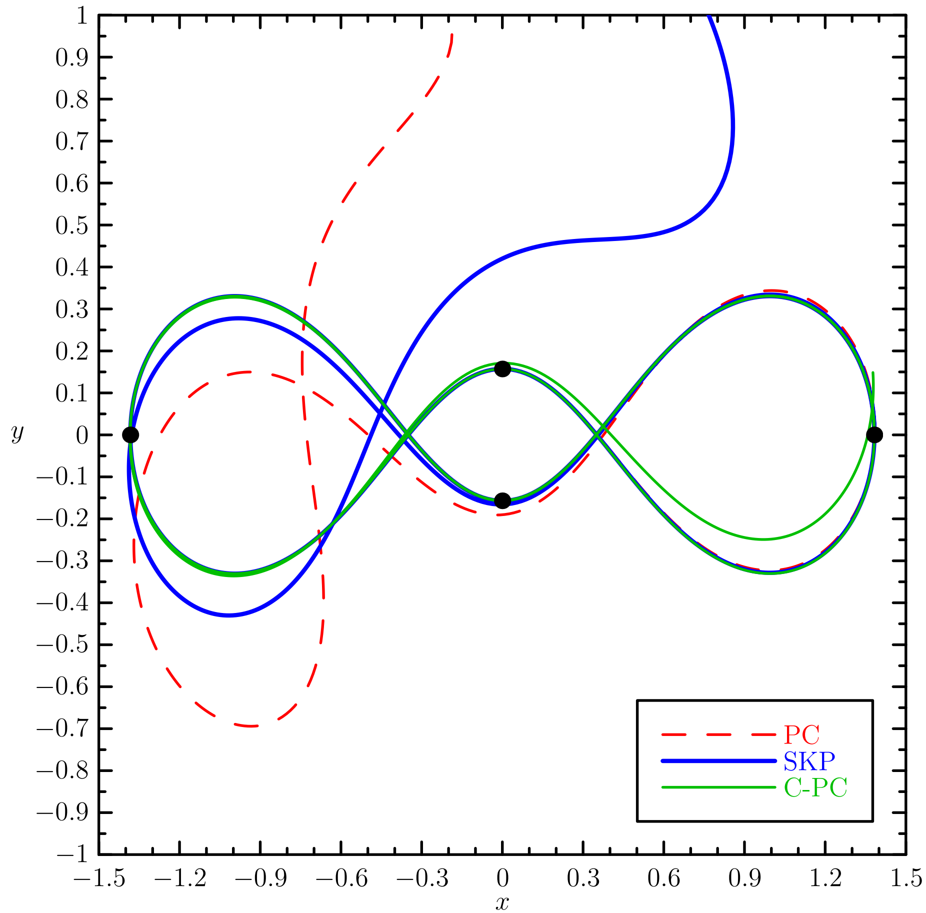
Singular Case

- When $T'(x_i) = 0$, the conservative corrector reduces to

$$x_{i+1} = T^{-1}\left(T(x_i) + \frac{h}{2}T'(\tilde{x})f(\tilde{x})\right),$$

- If T and f are analytic, the existence of a solution is guaranteed as $h \to 0^+$ if the points at which T' vanishes are isolated.

Four-Body Choreography



PC, symplectic SKP, and C–PC solutions

Conservative Symplectic Integrators

- Conservative variational symplectic integrators based on explicitly time-dependent symplectic maps have been proposed for certain mechanics problems. [Kane, Marsden, and Ortiz 1999]

Conservative Symplectic Integrators

- Conservative variational symplectic integrators based on explicitly time-dependent symplectic maps have been proposed for certain mechanics problems. [Kane, Marsden, and Ortiz 1999]

- These integrators circumvent the conditions of the Ge–Marsden theorem!

Linear Stiffness

- Consider for $y : \mathbb{R} \to \mathbb{R}$ and $L > 0$ the equation

$$\frac{dy}{dt} = -Ly,$$

with the initial condition $y(0) = y_0 \neq 0$.

Linear Stiffness

- Consider for $y : \mathbb{R} \to \mathbb{R}$ and $L > 0$ the equation

$$\frac{dy}{dt} = -Ly,$$

  with the initial condition $y(0) = y_0 \neq 0$.

- We know that the exact solution to this equation is given by

$$y(t) = y_0 e^{-tL}.$$

Linear Stiffness

- Consider for $y : \mathbb{R} \to \mathbb{R}$ and $L > 0$ the equation

$$\frac{dy}{dt} = -Ly,$$

with the initial condition $y(0) = y_0 \neq 0$.

- We know that the exact solution to this equation is given by

$$y(t) = y_0 e^{-tL}.$$

- Apply Euler's method with time step h:

$$y_{i+1} = (1 - hL)y_i.$$

Linear Stiffness

- Consider for $y : \mathbb{R} \to \mathbb{R}$ and $L > 0$ the equation

$$\frac{dy}{dt} = -Ly,$$

with the initial condition $y(0) = y_0 \neq 0$.

- We know that the exact solution to this equation is given by

$$y(t) = y_0 e^{-tL}.$$

- Apply Euler's method with time step h:

$$y_{i+1} = (1 - hL)y_i.$$

- For $hL \geq 2$, y_n does not converge to the correct steady-state solution.

Linear Stiffness

- Consider for $y : \mathbb{R} \to \mathbb{R}$ and $L > 0$ the equation

$$\frac{dy}{dt} = -Ly,$$

  with the initial condition $y(0) = y_0 \neq 0$.

- We know that the exact solution to this equation is given by

$$y(t) = y_0 e^{-tL}.$$

- Apply Euler's method with time step h:

$$y_{i+1} = (1 - hL)y_i.$$

- For $hL \geq 2$, y_n does not converge to the correct steady-state solution.
- If L is large, the time step is then forced to be unreasonably small.

- This phenomenon of linear stiffness manifests itself in general driven systems of ODEs in $\mathbb{R}^n$:

$$\frac{dy}{dt} + Ly = f(y).$$

- This phenomenon of linear stiffness manifests itself in general driven systems of ODEs in $\mathbb{R}^n$:

$$\frac{dy}{dt} + Ly = f(y).$$

- When the eigenvalues of L are large compared to the eigenvalues of f', a similar problem will occur.

Notation

$$\frac{dy}{dt} = f(t, y), \qquad y(0) = y_0,$$

- General s-stage Runge–Kutta scheme (scalar case):

$$y_{i+1} = y_0 + h \sum_{j=0}^{i} a_{ij} f(c_j h, y_j), \qquad i = 0, \ldots, s-1.$$

0 is the initial time; h is the time step;

y_s is the approximation to $y(h)$;

a_{ij} are the Runge–Kutta weights;

c_j are the step fractions for stage j.

Butcher Tableau ($s = 3$):

$$c_0 = 0, \qquad c_{i+1} = \sum_{j=0}^{i} a_{ij}.$$

$$\begin{array}{c|ccc} 0 & & & \\ c_1 & a_{00} & & \\ c_2 & a_{10} & a_{11} & \\ \hline 1 & a_{20} & a_{21} & a_{22} \end{array}$$

Exponential Integrators

- Circumvent linear stiffness by applying a scheme that is exact on the time scale of the linear part of the problem.

Exponential Integrators

- Circumvent linear stiffness by applying a scheme that is exact on the time scale of the linear part of the problem.
- Consider

$$\frac{dy}{dt} + Ly = f(y).$$

Exponential Integrators

- Circumvent linear stiffness by applying a scheme that is exact on the time scale of the linear part of the problem.

- Consider

$$\frac{dy}{dt} + Ly = f(y).$$

- Rewrite the above equation as

$$\frac{d(e^{tL}y)}{dt} = e^{tL}f(y)$$

  and integrate to obtain

$$y(h) = e^{-hL}y(0) + \int_0^h e^{-(h-s)L} f(y(0+s))ds.$$

Exponential Integrators

- Circumvent linear stiffness by applying a scheme that is exact on the time scale of the linear part of the problem.
- Consider

$$\frac{dy}{dt} + Ly = f(y).$$

- Rewrite the above equation as

$$\frac{d(e^{tL}y)}{dt} = e^{tL}f(y)$$

and integrate to obtain

$$y(h) = e^{-hL}y(0) + \int_0^h e^{-(h-s)L}f(y(0+s))ds.$$

- A quadrature rule is used to approximate the integral, while treating the exponential term exactly.

Stiff-Order Conditions

$$y_{i+1} = e^{-hL} y_0 + h \sum_{j=0}^{i} a_{ij}(-hL) f(y_j), \quad i = 0, ..., s-1.$$

- The weights a_{ij} are constructed from linear combinations of e^x and truncations of its Taylor series:

$$\varphi_0(x) = e^x$$

$$\varphi_{k+1}(x) = \frac{\varphi_k(x) - \frac{1}{k!}}{x} \quad \text{for } k \geq 0,$$

with $\varphi_k(0) = \frac{1}{k!}$.

Stiff-Order Conditions

$$y_{i+1} = e^{-hL} y_0 + h \sum_{j=0}^{i} a_{ij}(-hL) f(y_j), \quad i = 0, \ldots, s-1.$$

- The weights a_{ij} are constructed from linear combinations of e^x and truncations of its Taylor series:

$$\varphi_0(x) = e^x$$

$$\varphi_{k+1}(x) = \frac{\varphi_k(x) - \frac{1}{k!}}{x} \quad \text{for } k \geq 0,$$

with $\varphi_k(0) = \frac{1}{k!}$.

- Care must be exercised when evaluating φ near 0; see the C++ routines at www.math.ualberta.ca/~bowman/phi.h.

Stiff-Order Conditions

$$y_{i+1} = e^{-hL} y_0 + h \sum_{j=0}^{i} a_{ij}(-hL) f(y_j), \quad i = 0, \ldots, s-1.$$

- The weights a_{ij} are constructed from linear combinations of e^x and truncations of its Taylor series:

$$\varphi_0(x) = e^x$$
$$\varphi_{k+1}(x) = \frac{\varphi_k(x) - \frac{1}{k!}}{x} \quad \text{for } k \geq 0,$$

with $\varphi_k(0) = \frac{1}{k!}$.

- Care must be exercised when evaluating φ near 0; see the C++ routines at `www.math.ualberta.ca/~bowman/phi.h`.

- A set of *stiff-order conditions* on the weights were shown by Hochbruck and Ostermann to be *sufficient* to avoid *order reduction* when L has large eigenvalues.

Exponential Euler Algorithm

$$y_{i+1} = e^{-hL} y_i + \frac{1 - e^{-hL}}{L} f(y_i),$$

- Also called Exponentially Fitted Euler, ETD Euler, filtered Euler, Lie–Euler.

Exponential Euler Algorithm

$$y_{i+1} = e^{-hL} y_i + \frac{1 - e^{-hL}}{L} f(y_i),$$

- Also called Exponentially Fitted Euler, ETD Euler, filtered Euler, Lie–Euler.

- If it has a fixed point, it must satisfy $y = \frac{f(y)}{L}$; this is then a fixed point of the ODE.

Exponential Euler Algorithm

$$y_{i+1} = e^{-hL} y_i + \frac{1 - e^{-hL}}{L} f(y_i),$$

- Also called Exponentially Fitted Euler, ETD Euler, filtered Euler, Lie–Euler.

- If it has a fixed point, it must satisfy $y = \dfrac{f(y)}{L}$; this is then a fixed point of the ODE.

- In contrast, the popular Integrating Factor method (I-Euler).

$$y_{i+1} = e^{-hL}(y_i + hf_i)$$

can at best have an incorrect fixed point: $y = \dfrac{hf(y)}{e^{Lh} - 1}$.

Exponential Euler Algorithm

$$y_{i+1} = e^{-hL} y_i + \frac{1 - e^{-hL}}{L} f(y_i),$$

- Also called Exponentially Fitted Euler, ETD Euler, filtered Euler, Lie–Euler.

- If it has a fixed point, it must satisfy $y = \dfrac{f(y)}{L}$; this is then a fixed point of the ODE.

- In contrast, the popular Integrating Factor method (I-Euler).

$$y_{i+1} = e^{-hL}(y_i + hf_i)$$

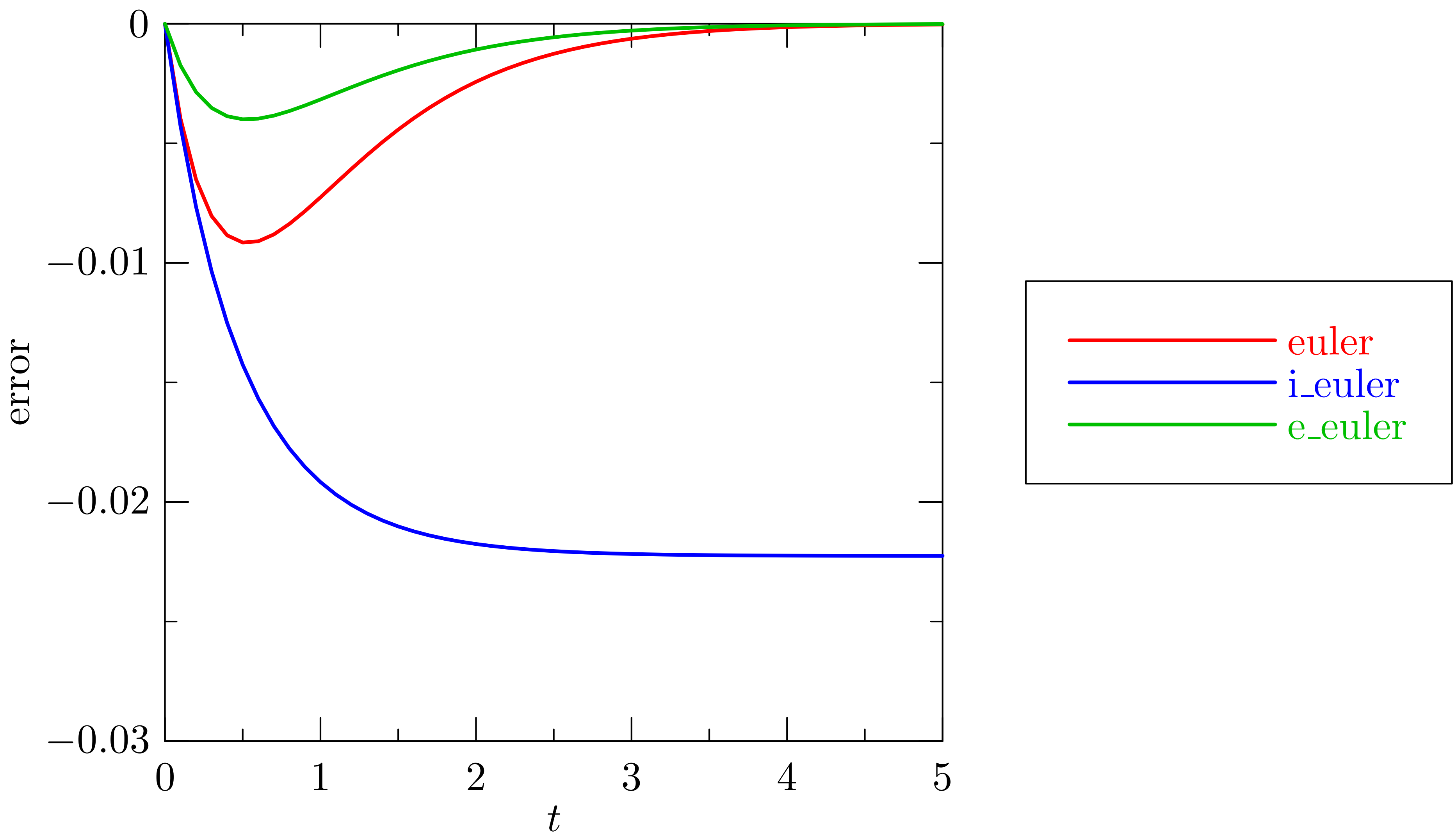
  can at best have an incorrect fixed point: $y = \dfrac{hf(y)}{e^{Lh} - 1}$.

- As $h \to 0$ the Euler method is recovered:

$$y_{i+1} = y_i + hf(y_i).$$

Comparison of Euler Integrators

$$\frac{dy}{dt} + y = \cos y, \qquad y(0) = 1.$$



History of Exponential Integrators

- Certaine [1960]: Exponential Adams-Moulton
- Nørsett [1969]: Exponential Adams-Bashforth
- Verwer [1977] and van der Houwen [1977]: Exponential linear multistep method
- Friedli [1978]: Exponential Runge–Kutta
- Hochbruck *et al.* [1998]: Exponential integrators up to order 4
- Beylkin *et al.* [1998]: Exact Linear Part (ELP)
- Cox & Matthews [2002]: ETDRK3, ETDRK4; worst case: stiff order 2
- Lu [2003]: Efficient Matrix Exponential
- Hochbruck & Ostermann [2005a]: Explicit Exponential Runge–Kutta; stiff order conditions.

Embedded Pairs for Adaptive Time-Stepping

- An adaptive pair is *robust* if the order of the low-order method is never equal to the order n of the high-order method for any source function $G(t) = F(t, y(t))$ with a nonzero derivative of order less than n.

Embedded Pairs for Adaptive Time-Stepping

- An adaptive pair is *robust* if the order of the low-order method is never equal to the order n of the high-order method for any source function $G(t) = F(t, y(t))$ with a nonzero derivative of order less than n.

- A nonrobust method can mislead the time step adjustment algorithm into adopting too large a time step, leading to catastrophic loss of accuracy.

(3,2) Robust Embedded Pair ERK32ZB

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\varphi_1\left(-\frac{hL}{2}\right) & & & \\
\frac{3}{4} & \frac{3}{4}\varphi_1\left(-\frac{3hL}{4}\right) - a_{11} & \frac{9}{8}\varphi_2(-\frac{3hL}{4}) + \frac{3}{8}\varphi_2(-\frac{hL}{2}) & & \\
\hline
1 & \varphi_1 - a_{21} - a_{22} - a_{23} & \frac{3}{4}\varphi_2 - \frac{1}{4}\varphi_3 & \frac{5}{6}\varphi_2 + \frac{1}{6}\varphi_3 & \\
1 & a_{30} & a_{31} & a_{32} & a_{33},
\end{array}
$$

where $\varphi_i = \varphi_i(-hL)$ and

$$
\begin{aligned}
a_{30} &= \frac{29}{18}\varphi_1 + \frac{7}{6}\varphi_1\left(-\frac{3hL}{4}\right) + \frac{9}{14}\varphi_1\left(-\frac{hL}{2}\right) + \frac{3}{4}\varphi_2 \\
&\quad + \frac{2}{7}\varphi_2\left(-\frac{3hL}{4}\right) + \frac{1}{12}\varphi_2\left(-\frac{hL}{2}\right) - \frac{8083}{420}\varphi_3 + \frac{11}{30}\varphi_3\left(-\frac{hL}{2}\right) \\
a_{31} &= -\frac{1}{9}\varphi_1 - \frac{1}{6}\varphi_1\left(-\frac{3hL}{4}\right) - \frac{1}{2}\varphi_2 \\
&\quad - \frac{1}{7}\varphi_2\left(-\frac{3hL}{4}\right) - \frac{1}{3}\varphi_2\left(-\frac{hL}{2}\right) + \frac{1}{6}\varphi_3 + \frac{1}{6}\varphi_3\left(-\frac{hL}{2}\right) \\
a_{32} &= \frac{2}{3}\varphi_1 - \frac{1}{2}\varphi_1\left(-\frac{3hL}{4}\right) - \frac{1}{7}\varphi_1\left(-\frac{hL}{2}\right) + \frac{1}{3}\varphi_2 \\
&\quad - \frac{1}{7}\varphi_2\left(-\frac{3hL}{4}\right) - \frac{1}{5}\varphi_3\left(-\frac{hL}{2}\right) \\
a_{33} &= -\frac{7}{6}\varphi_1 - \frac{1}{2}\varphi_1\left(-\frac{3hL}{4}\right) - \frac{1}{2}\varphi_1\left(-\frac{hL}{2}\right) - \frac{7}{12}\varphi_2 \\
&\quad + \frac{1}{4}\varphi_2\left(-\frac{hL}{2}\right) + \frac{2671}{140}\varphi_3 - \frac{1}{3}\varphi_3\left(-\frac{hL}{2}\right).
\end{aligned}
$$

(4,3) Robust Embedded Pair ERK43ZB

$$
\begin{array}{c|ccccc}
0 & & & & & \\
\frac{1}{6} & \frac{1}{6}\varphi_1\left(-\frac{hL}{6}\right) & & & & \\
\frac{1}{2} & \frac{1}{2}\varphi_1\left(-\frac{hL}{2}\right) - a_{11} & a_{11} & & & \\
\frac{1}{2} & \frac{1}{2}\varphi_1\left(-\frac{hL}{2}\right) - a_{21} - a_{22} & a_{21} & a_{22} & & \\
\hline
1 & \varphi_1 - a_{31} - a_{32} - a_{33} & a_{31} & a_{32} & a_{33} & \\
1 & \varphi_1 - \frac{67}{9}\varphi_2 + \frac{52}{3}\varphi_3 & 8\varphi_2 - 24\varphi_3 & \frac{26}{3}\varphi_3 - \frac{11}{9}\varphi_2 & a_{43} & a_{44},
\end{array}
$$

where $\varphi_i = \varphi_i(-hL)$ and

$$
\begin{aligned}
a_{11} &= \frac{3}{2}\varphi_2\left(-\frac{hL}{2}\right) + \frac{1}{2}\varphi_2\left(-\frac{hL}{6}\right)\\
a_{21} &= \frac{19}{60}\varphi_1 + \frac{1}{2}\varphi_1\left(-\frac{hL}{2}\right) + \frac{1}{2}\varphi_1\left(-\frac{hL}{6}\right)\\
&\quad + 2\varphi_2\left(-\frac{hL}{2}\right) + \frac{13}{6}\varphi_2\left(-\frac{hL}{6}\right) + \frac{3}{5}\varphi_3\left(-\frac{hL}{2}\right)\\
a_{22} &= -\frac{19}{180}\varphi_1 - \frac{1}{6}\varphi_1\left(-\frac{hL}{2}\right) - \frac{1}{6}\varphi_1\left(-\frac{hL}{6}\right)\\
&\quad - \frac{1}{6}\varphi_2\left(-\frac{hL}{2}\right) + \frac{1}{9}\varphi_2\left(-\frac{hL}{6}\right) - \frac{1}{5}\varphi_3\left(-\frac{hL}{2}\right)\\
a_{33} &= \varphi_2 + \varphi_2\left(-\frac{hL}{2}\right) - 6\varphi_3 - 3\varphi_3\left(-\frac{hL}{2}\right)\\
a_{31} &= 3\varphi_2 - \frac{9}{2}\varphi_2\left(-\frac{hL}{2}\right) - \frac{5}{2}\varphi_2\left(-\frac{hL}{6}\right) + 6a_{33} + a_{21}\\
a_{32} &= 6\varphi_3 + 3\varphi_3\left(-\frac{hL}{2}\right) - 2a_{33} + a_{22}\\
a_{43} &= \frac{7}{9}\varphi_2 - \frac{10}{3}\varphi_3, \qquad a_{44} = \frac{4}{3}\varphi_3 - \frac{1}{9}\varphi_2.
\end{aligned}
$$

Test Problem

- We illustrate robustness by comparing ERK43ZB to ERK43DK [Ding & Kang 2017] for a test problem from Hochbruck–Ostermann:

Test Problem

- We illustrate robustness by comparing ERK43ZB to ERK43DK [Ding & Kang 2017] for a test problem from Hochbruck–Ostermann:
- For $x \in [0, 1]$ and $t \geq 0$:

$$\frac{\partial y}{\partial t}(x,t) - \frac{\partial^2 y}{\partial x^2}(x,t) = H(x,t) + \Phi(x,t).$$

$$H(x,t) = \frac{1}{1 + y(x,t)^2}.$$

Test Problem

- We illustrate robustness by comparing ERK43ZB to ERK43DK [Ding & Kang 2017] for a test problem from Hochbruck–Ostermann:
- For $x \in [0, 1]$ and $t \geq 0$:

$$\frac{\partial y}{\partial t}(x,t) - \frac{\partial^2 y}{\partial x^2}(x,t) = H(x,t) + \Phi(x,t).$$

$$H(x,t) = \frac{1}{1 + y(x,t)^2}.$$

- Φ is chosen so that the exact solution is

$$y(x,t) = x(1-x)e^t.$$

Test Problem

- We illustrate robustness by comparing ERK43ZB to ERK43DK [Ding & Kang 2017] for a test problem from Hochbruck–Ostermann:
- For $x \in [0, 1]$ and $t \geq 0$:

$$\frac{\partial y}{\partial t}(x,t) - \frac{\partial^2 y}{\partial x^2}(x,t) = H(x,t) + \Phi(x,t).$$

$$H(x,t) = \frac{1}{1 + y(x,t)^2}.$$

- Φ is chosen so that the exact solution is

$$y(x,t) = x(1-x)e^t.$$

- 200 spatial grid points, evolve from $t = 0$ to $t = 3$.

Test Problem

- We illustrate robustness by comparing ERK43ZB to ERK43DK [Ding & Kang 2017] for a test problem from Hochbruck–Ostermann:
- For $x \in [0, 1]$ and $t \geq 0$:

$$\frac{\partial y}{\partial t}(x,t) - \frac{\partial^2 y}{\partial x^2}(x,t) = H(x,t) + \Phi(x,t).$$

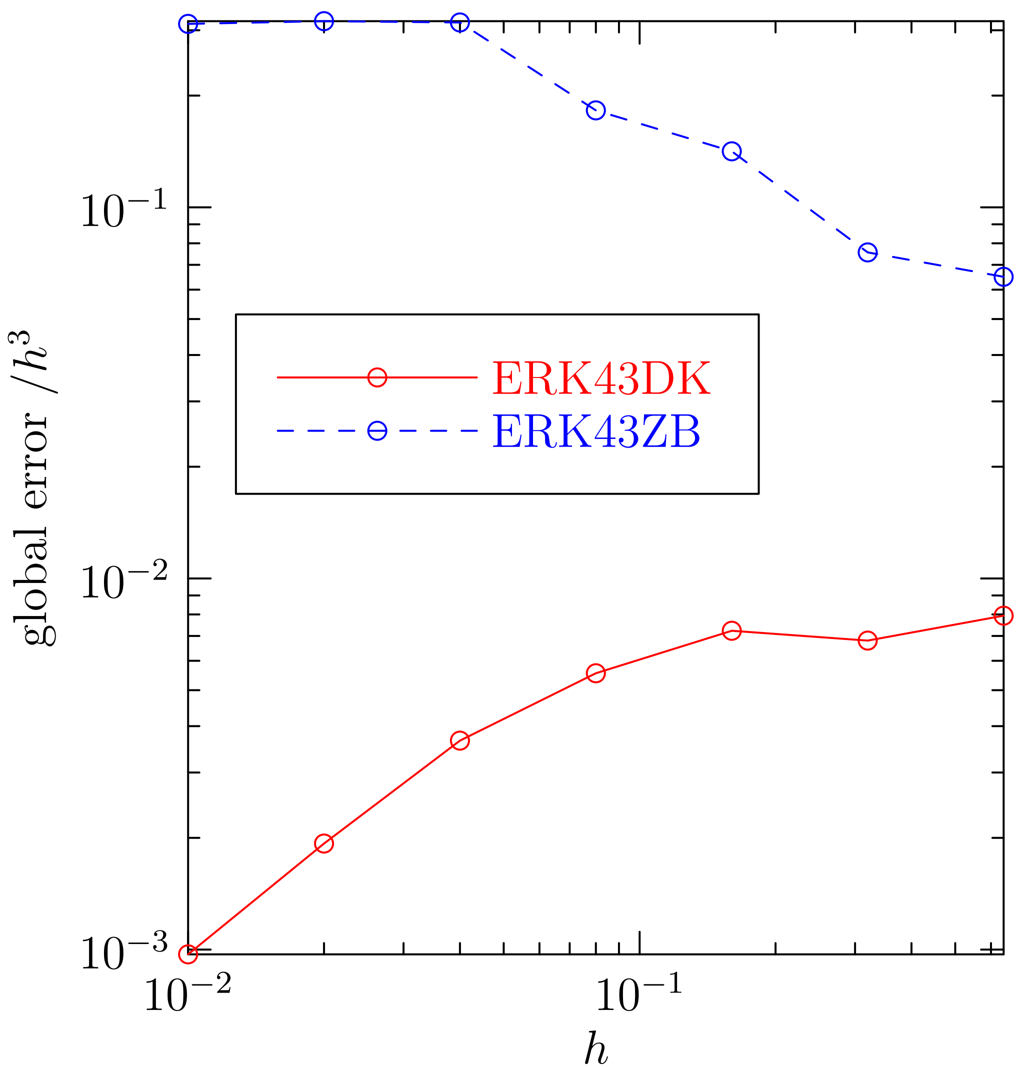
$$H(x,t) = \frac{1}{1 + y(x,t)^2}.$$

- Φ is chosen so that the exact solution is

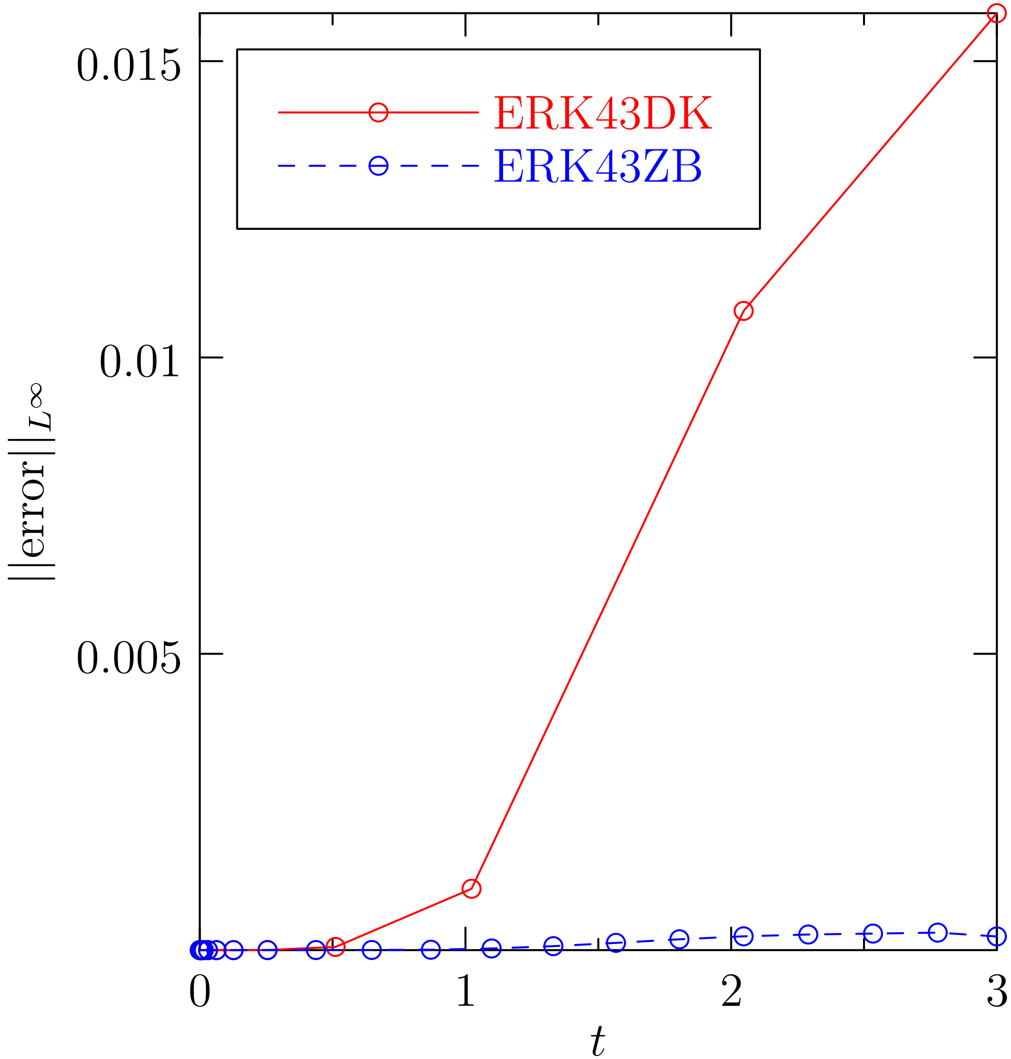
$$y(x,t) = x(1-x)e^t.$$

- 200 spatial grid points, evolve from $t = 0$ to $t = 3$.
- We calculate the matrix φ_k functions with the help of Padé approximants, along with repeated scaling and squaring.

Robust vs. Non-Robust Third-Order Estimate

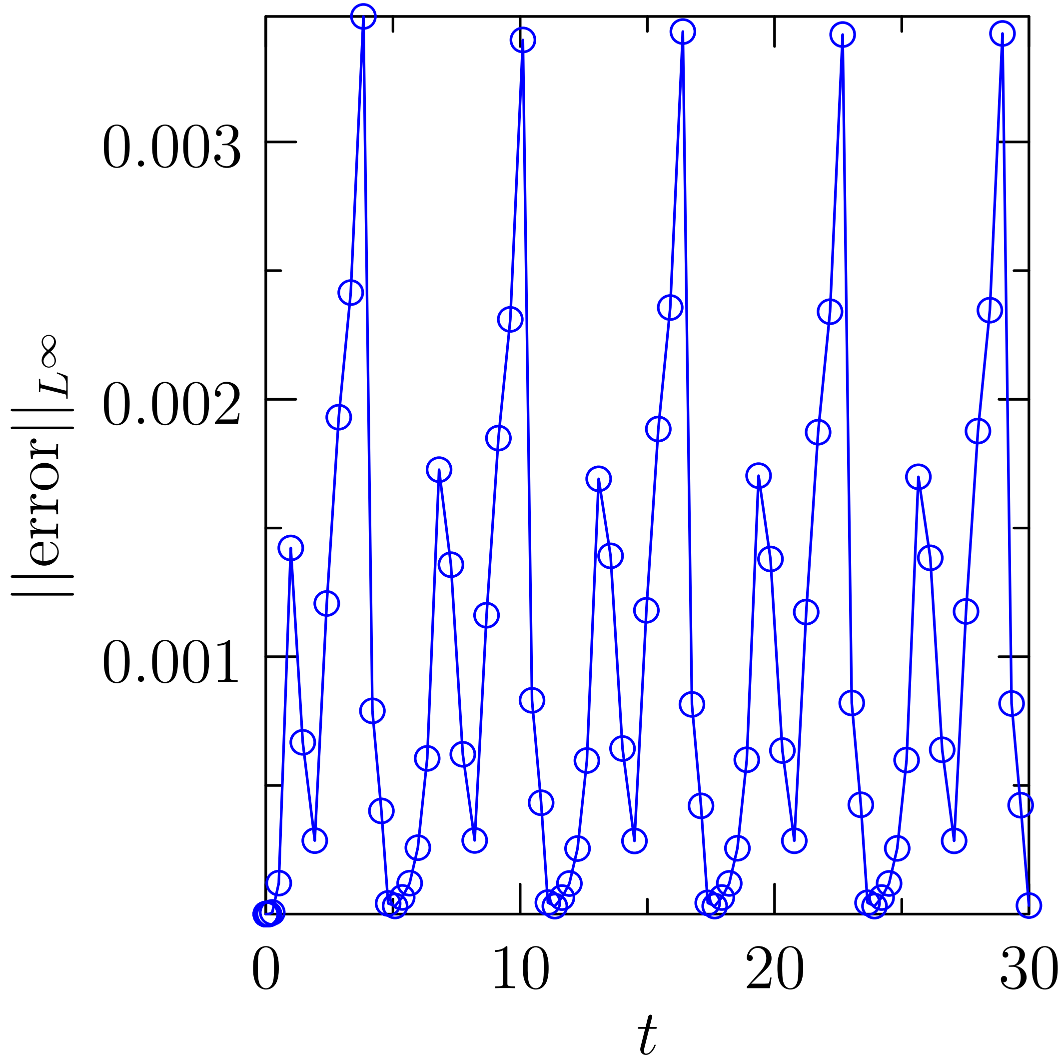


Robust vs. Non-Robust Time Evolution



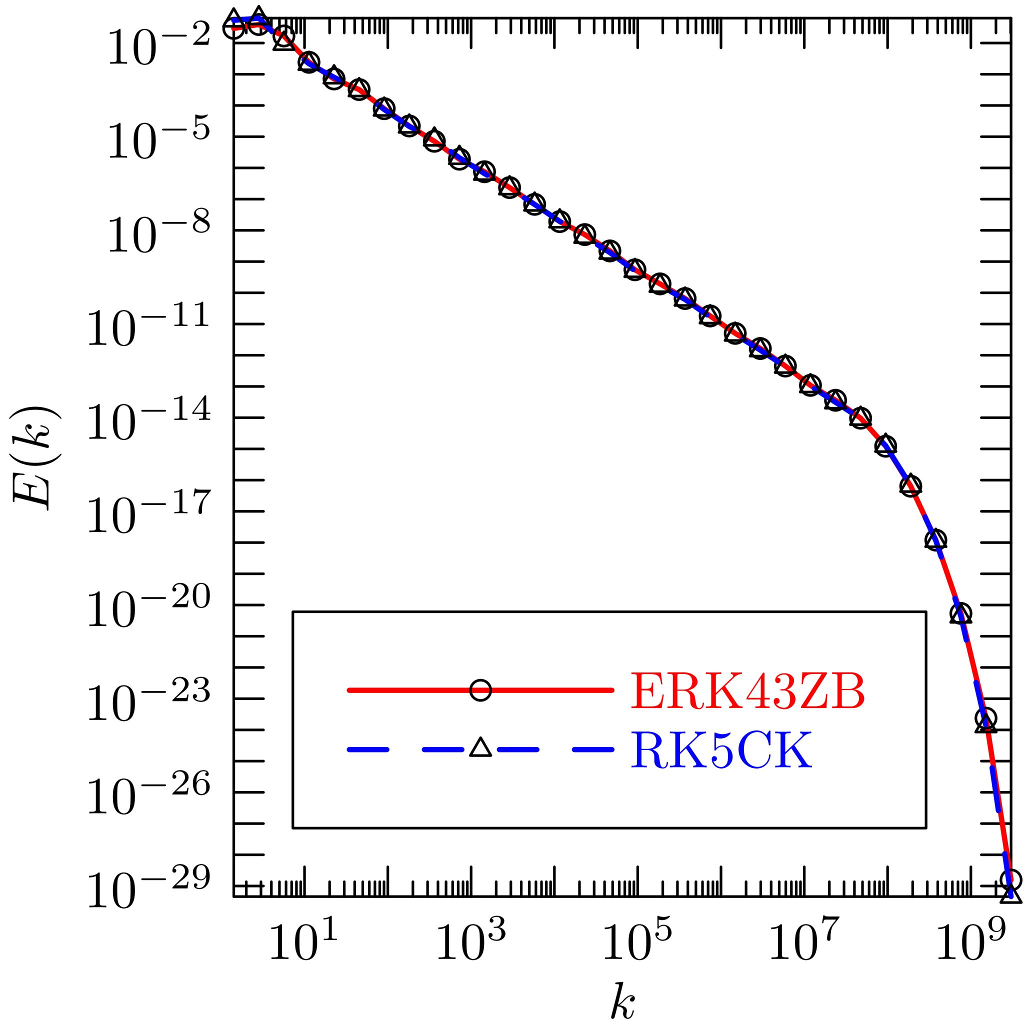
Adaptive Performance of ERK43ZB

- Choose Φ such that $y(x,t) = 10(1-x)x(1+\sin t)+2$:



GOY Shell Model of 3D Turbulence

- ERK43ZB runs over 3 times faster than the classical Cash–Karp (5,4) pair on a shell model of 3D turbulence exhibiting both linear and nonlinear stiffness:



Schur Decomposition

- When L is a nondiagonal matrix, the matrix exponentials required by exponential integrators are computationally expensive.

Schur Decomposition

- When L is a nondiagonal matrix, the matrix exponentials required by exponential integrators are computationally expensive.
- Consider the Schur decomposition of L:

$$L = UTU^\dagger,$$

where U is a *unitary* matrix and T is an upper triangular matrix.

Schur Decomposition

- When L is a nondiagonal matrix, the matrix exponentials required by exponential integrators are computationally expensive.

- Consider the Schur decomposition of L:

$$L = UTU^{\dagger},$$

where U is a *unitary* matrix and T is an upper triangular matrix.

- Decompose $T = D + S$, where D is a diagonal matrix and S is a strictly upper triangular matrix.

Schur Decomposition

- When L is a nondiagonal matrix, the matrix exponentials required by exponential integrators are computationally expensive.

- Consider the Schur decomposition of L:

$$L = UTU^{\dagger},$$

where U is a *unitary* matrix and T is an upper triangular matrix.

- Decompose $T = D + S$, where D is a diagonal matrix and S is a strictly upper triangular matrix.

- We obtain

$$\frac{dy}{dt} + U(D+S)U^{\dagger}y = F(t, y).$$

- Multiply by $U^\dagger$ on the left:

$$\frac{d(U^\dagger y)}{dt} + (D + S)U^\dagger y = U^\dagger F(t, y).$$

- Multiply by $U^\dagger$ on the left:

$$\frac{d(U^\dagger y)}{dt} + (D + S)U^\dagger y = U^\dagger F(t, y).$$

- In terms of the transformed variable $Y = U^\dagger y$:

$$\frac{dY}{dt} + DY = U^\dagger F(t, UY) - SY.$$

- Multiply by $U^\dagger$ on the left:

$$\frac{d(U^\dagger y)}{dt} + (D+S)U^\dagger y = U^\dagger F(t,y).$$

- In terms of the transformed variable $Y = U^\dagger y$:

$$\frac{dY}{dt} + DY = U^\dagger F(t,UY) - SY.$$

- This transformation allows us to replace exponentials of a full matrix with a diagonal matrix of scalar exponentials.

- Multiply by $U^\dagger$ on the left:

$$\frac{d(U^\dagger y)}{dt} + (D + S)U^\dagger y = U^\dagger F(t, y).$$

- In terms of the transformed variable $Y = U^\dagger y$:

$$\frac{dY}{dt} + DY = U^\dagger F(t, UY) - SY.$$

- This transformation allows us to replace exponentials of a full matrix with a diagonal matrix of scalar exponentials.

- Being diagonal, the φ_k functions now require far less storage.

- Multiply by $U^\dagger$ on the left:

$$\frac{d(U^\dagger y)}{dt} + (D+S)U^\dagger y = U^\dagger F(t,y).$$

- In terms of the transformed variable $Y = U^\dagger y$:

$$\frac{dY}{dt} + DY = U^\dagger F(t, UY) - SY.$$

- This transformation allows us to replace exponentials of a full matrix with a diagonal matrix of scalar exponentials.
- Being diagonal, the φ_k functions now require far less storage.
- Although the computation of the Schur decomposition of L is expensive, *it only has to be done once.*

- Multiply by $U^\dagger$ on the left:

$$\frac{d(U^\dagger y)}{dt} + (D+S)U^\dagger y = U^\dagger F(t,y).$$

- In terms of the transformed variable $Y = U^\dagger y$:

$$\frac{dY}{dt} + DY = U^\dagger F(t,UY) - SY.$$

- This transformation allows us to replace exponentials of a full matrix with a diagonal matrix of scalar exponentials.

- Being diagonal, the φ_k functions now require far less storage.

- Although the computation of the Schur decomposition of L is expensive, *it only has to be done once.*

- The explicit treatment of the upper triangular matrix S contributes to the overall error, but *does not contribute to stiffness.*

- Moreover, many matrices encountered in practice are *normal*: they commute with their Hermitian adjoint.

- Moreover, many matrices encountered in practice are *normal*: they commute with their Hermitian adjoint.
- For normal matrices, $S = 0$:

Lemma: The triangle matrix in the Schur decomposition of a normal matrix is diagonal.

- Moreover, many matrices encountered in practice are *normal*: they commute with their Hermitian adjoint.

- For normal matrices, $S = 0$:

  Lemma: The triangle matrix in the Schur decomposition of a normal matrix is diagonal.

- With the optimization afforded by Schur decomposition, embedded ERK methods for step size adjustment becomes computationally viable, *even when L is a nondiagonal matrix.*

- Moreover, many matrices encountered in practice are *normal*: they commute with their Hermitian adjoint.
- For normal matrices, $S = 0$:

  Lemma: The triangle matrix in the Schur decomposition of a normal matrix is diagonal.
- With the optimization afforded by Schur decomposition, embedded ERK methods for step size adjustment becomes computationally viable, *even when L is a nondiagonal matrix.*
- An adaptive exponential method requires re-evaluating the φ_k functions whenever the step size is adjusted.

- Moreover, many matrices encountered in practice are *normal*: they commute with their Hermitian adjoint.
- For normal matrices, $S = 0$:

  Lemma: The triangle matrix in the Schur decomposition of a normal matrix is diagonal.
- With the optimization afforded by Schur decomposition, embedded ERK methods for step size adjustment becomes computationally viable, *even when L is a nondiagonal matrix.*
- An adaptive exponential method requires re-evaluating the φ_k functions whenever the step size is adjusted.
- However, since these are now functions of diagonal matrices, there is no longer a huge computational cost.

Claim: The term Sy does not incorporate any of the stiffness inherent in the linear term Ly.

Proof:

- On defining the integrating factor $I(t) = e^{tD}$ and $\tilde{y}(t) = I(t)y(t)$, we can transform the autonomous case to

$$\frac{d\tilde{y}}{dt} = I(t)U^{\dagger}F(UI^{-1}(t)\tilde{y}) - \tilde{S}\tilde{y},$$

where $\tilde{S} = I(t)SI^{-1}(t)$ is an $m \times m$ strictly upper triangular matrix.

Claim: The term Sy does not incorporate any of the stiffness inherent in the linear term Ly.

Proof:

- On defining the integrating factor $I(t) = e^{tD}$ and $\tilde{y}(t) = I(t)y(t)$, we can transform the autonomous case to

$$\frac{d\tilde{y}}{dt} = I(t)U^{\dagger}F(UI^{-1}(t)\tilde{y}) - \tilde{S}\tilde{y},$$

where $\tilde{S} = I(t)SI^{-1}(t)$ is an $m \times m$ strictly upper triangular matrix.

- If the stiffness only enters through the linear term Ly and not through $F(y)$, the first term on the right-hand side will not contribute any additional stiffness.

- When $F = 0$, we obtain the triangular system of equations

$$\frac{d\tilde{y}_i}{dt} = \sum_{j=i+1}^{m} \tilde{S}_{ij}\tilde{y}_j \text{ for } i = 1, \ldots, m-1 \quad \text{and} \quad \frac{d\tilde{y}_m}{dt} = 0,$$

which can be solved recursively to obtain the general solution as a polynomial in t.

- When $F = 0$, we obtain the triangular system of equations

$$\frac{d\tilde{y}_i}{dt} = \sum_{j=i+1}^{m} \tilde{S}_{ij}\tilde{y}_j \text{ for } i = 1, \ldots, m-1 \quad \text{and} \quad \frac{d\tilde{y}_m}{dt} = 0,$$

  which can be solved recursively to obtain the general solution as a polynomial in t.

- Stiffness arises only when nearby solution curves approach the solution curve of interest at exponentially fast rates.

- When $F = 0$, we obtain the triangular system of equations

$$\frac{d\tilde{y}_i}{dt} = \sum_{j=i+1}^{m} \tilde{S}_{ij}\tilde{y}_j \text{ for } i = 1, \ldots, m-1 \quad \text{and} \quad \frac{d\tilde{y}_m}{dt} = 0,$$

which can be solved recursively to obtain the general solution as a polynomial in t.

- Stiffness arises only when nearby solution curves approach the solution curve of interest at exponentially fast rates.

- Thus, the decomposed system of equations is not stiff; it can in fact be solved exactly by a classical Runge–Kutta method whose order is at least the degree of the solution polynomials.

- When $F = 0$, we obtain the triangular system of equations

$$\frac{d\tilde{y}_i}{dt} = \sum_{j=i+1}^{m} \tilde{S}_{ij}\tilde{y}_j \text{ for } i = 1, \ldots, m-1 \quad \text{and} \quad \frac{d\tilde{y}_m}{dt} = 0,$$

  which can be solved recursively to obtain the general solution as a polynomial in t.

- Stiffness arises only when nearby solution curves approach the solution curve of interest at exponentially fast rates.

- Thus, the decomposed system of equations is not stiff; it can in fact be solved exactly by a classical Runge–Kutta method whose order is at least the degree of the solution polynomials.

- By linear superposition, the system is not stiff even when F is linear (and, in particular, when F is constant).

- When $F = 0$, we obtain the triangular system of equations

$$\frac{d\tilde{y}_i}{dt} = \sum_{j=i+1}^{m} \tilde{S}_{ij}\tilde{y}_j \text{ for } i = 1, \ldots, m-1 \quad \text{and} \quad \frac{d\tilde{y}_m}{dt} = 0,$$

  which can be solved recursively to obtain the general solution as a polynomial in t.

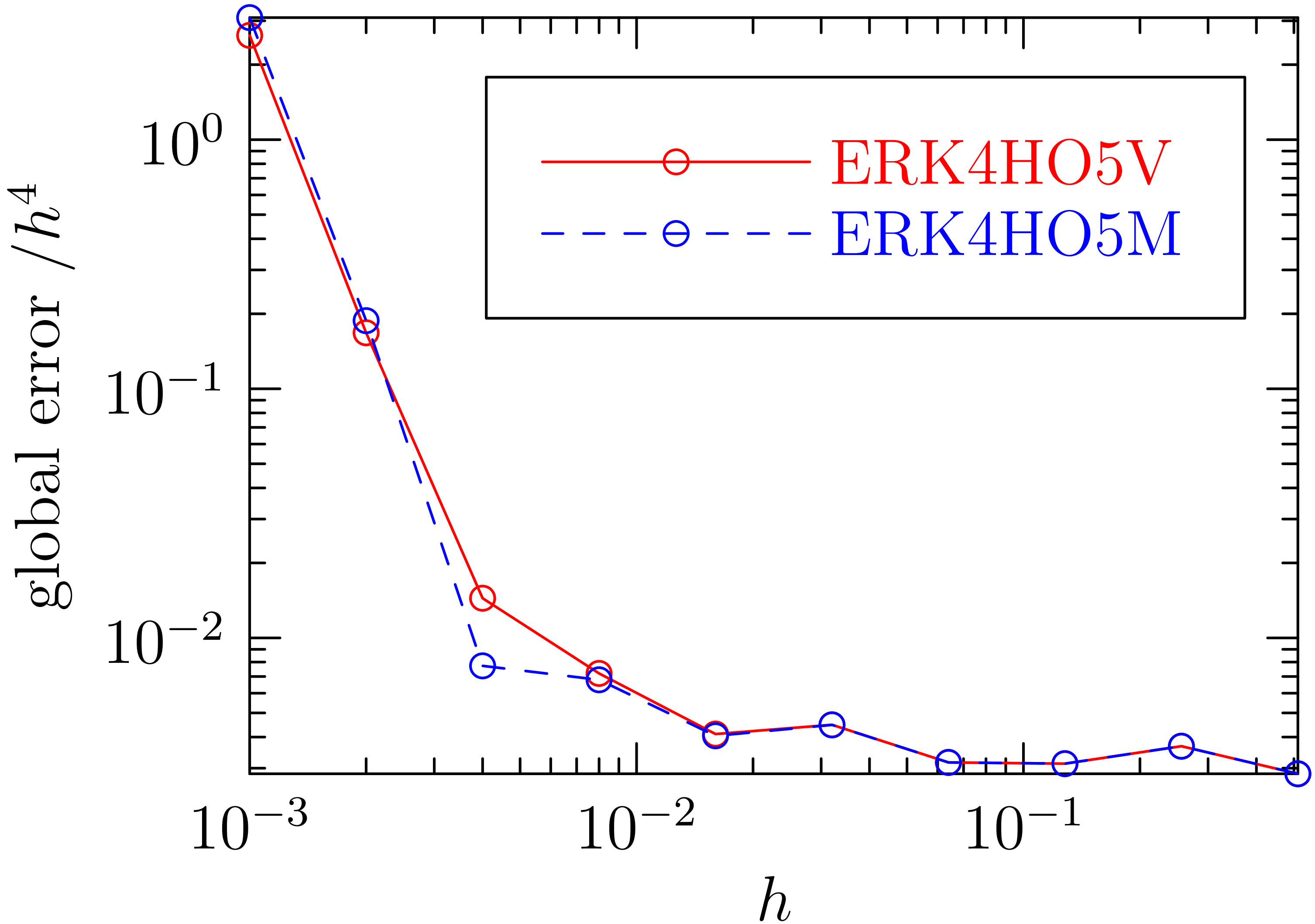
- Stiffness arises only when nearby solution curves approach the solution curve of interest at exponentially fast rates.

- Thus, the decomposed system of equations is not stiff; it can in fact be solved exactly by a classical Runge–Kutta method whose order is at least the degree of the solution polynomials.

- By linear superposition, the system is not stiff even when F is linear (and, in particular, when F is constant).

- The linear stiffness is thus *entirely contained within the diagonal term* DY.

Schur Decomposition vs. Full Solution



Conclusions

- Numerical discretizations that preserve physically relevant structure or known analytic properties are desirable.

Conclusions

- Numerical discretizations that preserve physically relevant structure or known analytic properties are desirable.
- Traditional numerical discretizations of conservative systems generically yield artificial secular drifts of nonlinear invariants.

Conclusions

- Numerical discretizations that preserve physically relevant structure or known analytic properties are desirable.
- Traditional numerical discretizations of conservative systems generically yield artificial secular drifts of nonlinear invariants.
- New exactly conservative but explicit integration algorithms have been developed.

Conclusions

- Numerical discretizations that preserve physically relevant structure or known analytic properties are desirable.
- Traditional numerical discretizations of conservative systems generically yield artificial secular drifts of nonlinear invariants.
- New exactly conservative but explicit integration algorithms have been developed.
- The transformation technique is relevant to integrable and nonintegrable Hamiltonian systems and even to non-Hamiltonian systems such as force-dissipative turbulence.

Conclusions

- Numerical discretizations that preserve physically relevant structure or known analytic properties are desirable.
- Traditional numerical discretizations of conservative systems generically yield artificial secular drifts of nonlinear invariants.
- New exactly conservative but explicit integration algorithms have been developed.
- The transformation technique is relevant to integrable and nonintegrable Hamiltonian systems and even to non-Hamiltonian systems such as force-dissipative turbulence.
- Exponential integrators are explicit schemes for ODEs with a stiff linearity.

Conclusions

- Numerical discretizations that preserve physically relevant structure or known analytic properties are desirable.
- Traditional numerical discretizations of conservative systems generically yield artificial secular drifts of nonlinear invariants.
- New exactly conservative but explicit integration algorithms have been developed.
- The transformation technique is relevant to integrable and nonintegrable Hamiltonian systems and even to non-Hamiltonian systems such as force-dissipative turbulence.
- Exponential integrators are explicit schemes for ODEs with a stiff linearity.
- When the nonlinear source is constant, the time-stepping algorithm is precisely the analytical solution to the corresponding first-order linear ODE.

- Unlike integrating factor methods, exponential integrators have the correct fixed point behaviour.

- Unlike integrating factor methods, exponential integrators have the correct fixed point behaviour.
- We derived adaptive ERK pairs by symbolically solving the Hochbruck–Ostermann stiff-order conditions.

- Unlike integrating factor methods, exponential integrators have the correct fixed point behaviour.
- We derived adaptive ERK pairs by symbolically solving the Hochbruck–Ostermann stiff-order conditions.
- A key requirement is that the pair be robust: if the nonlinear source function has nonzero total time derivatives, the order of the low-order estimate should never exceed its design value.

- Unlike integrating factor methods, exponential integrators have the correct fixed point behaviour.

- We derived adaptive ERK pairs by symbolically solving the Hochbruck–Ostermann stiff-order conditions.

- A key requirement is that the pair be robust: if the nonlinear source function has nonzero total time derivatives, the order of the low-order estimate should never exceed its design value.

- New robust exponential Runge–Kutta (3,2) and (4,3) embedded pairs are well-suited to initial value problems with a dominant linearity.

- Unlike integrating factor methods, exponential integrators have the correct fixed point behaviour.
- We derived adaptive ERK pairs by symbolically solving the Hochbruck–Ostermann stiff-order conditions.
- A key requirement is that the pair be robust: if the nonlinear source function has nonzero total time derivatives, the order of the low-order estimate should never exceed its design value.
- New robust exponential Runge–Kutta (3,2) and (4,3) embedded pairs are well-suited to initial value problems with a dominant linearity.
- A Schur decomposition avoids the need for computing matrix exponentials, while still circumventing linear stiffness.

References

[Beylkin *et al.* 1998] G. Beylkin, J. M. Keiser, & L. Vozovoi, J. Comp. Phys., **147**:362, 1998.

[Bowman & Krommes 1997] J. C. Bowman & J. A. Krommes, Phys. Plasmas, **4**:3895, 1997.

[Bowman *et al.* 1997] J. C. Bowman, B. A. Shadwick, & P. J. Morrison, "Exactly conservative integrators," in *15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, edited by A. Sydow, volume 2, pp. 595–600, Berlin, 1997, Wissenschaft & Technik.

[Certaine 1960] J. Certaine, Mathematical methods for digital computers, **1**:128, 1960.

[Channell & Scovel 1990] P. J. Channell & J. C. Scovel, Nonlinearity, **3**:231, 1990.

[Cox & Matthews 2002] S. Cox & P. Matthews, J. Comp. Phys., **176**:430, 2002.

[Ding & Kang 2017] X. Ding & S. Kang, arXiv preprint arXiv:1703.09622, 2017.

[Friedli 1978] A. Friedli, Lecture Notes in Mathematics, **631**:214, 1978.

[Ge Zhong & Marsden 1988] Ge Zhong & J. E. Marsden, Phys. Lett. A, **133**:134, 1988.

[Hochbruck & Ostermann 2005a] M. Hochbruck & A. Ostermann, SIAM J. Numer. Anal., **43**:1069, 2005.

[Hochbruck & Ostermann 2005b] M. Hochbruck & A. Ostermann, Appl. Numer. Math., **53**:323, 2005.

[Hochbruck *et al.* 1998] M. Hochbruck, C. Lubich, & H. Selfhofer, SIAM J. Sci. Comput., **19**:1552, 1998.

[Kane *et al.* 1999] C. Kane, J. E. Marsden, & M. Ortiz, J. Math. Phys., **40**:3353, 1999.

[Kotovych & Bowman 2002] O. Kotovych & J. C. Bowman, J. Phys. A.: Math. Gen., **35**:7849, 2002.

[Lu 2003] Y. Y. Lu, J. Comput. Appl. Math., **161**:203, 2003.

[Nørsett 1969] S. Nørsett, Lecture Notes in Mathematics, **109**:214, 1969.

[Ruth 1983] R. D. Ruth, IEEE Trans. Nucl. Sci., **NS-30**:2669, 1983.

[Sanz-Serna & Calvo 1994] J. M. Sanz-Serna & M. P. Calvo, *Numerical Hamiltonian Problems*, volume 7 of *Applied Mathematics and Mathematical Computation*, Chapman and Hall, London, 1994.

[Shadwick & Buell 1997] B. A. Shadwick & W. F. Buell, Phys. Rev. Lett., **79**:5189, 1997.

[Shadwick *et al.* 1999] B. A. Shadwick, J. C. Bowman, & P. J. Morrison, SIAM J. Appl. Math., **59**:1112, 1999.

[Shampine 1986] L. F. Shampine, Computers & Mathematics with Applications, **12**:1287, 1986.

[van der Houwen 1977] P. J. van der Houwen, *Construction of integration formulas for initial value problems*, North-Holland Publishing Co., Amsterdam, 1977, North-Holland Series in Applied Mathematics and Mechanics, Vol. 19.

[Verwer 1977] J. Verwer, Numer. Math., **27**:143, 1977.

[Wan *et al.* 2017] A. T. Wan, A. Bihlo, & J.-C. Nave, SIAM Journal on Numerical Analysis, **55**:2255, 2017.