Dynamics and numerical time integration

Turn a differential model into a time step. Track accuracy, stability and structural properties separately in a computable physical or mathematical example.

Contents

1. **Find a crossing between accepted time steps**
   An integrator can approximate a trajectory accurately yet report a poor switch time. Construct a continuous interpolant, relate a state error to an event-time error and inspect what happens when a trajectory merely touches a threshold.
2. **A delayed feedback model needs a history, not one starting value**
   A controller responds to the measured displacement one time delay ago. Construct its solution in successive intervals, track propagated derivative discontinuities and derive the point at which increasing delay can destabilise negative feedback.
3. **Solving a two-point boundary problem without guessing the start slope**
   When endpoint conditions act on different sides of an interval, shooting can amplify a tiny initial error. Construct a cubic collocation system, test its continuous differential residual and quantify which boundary-value solution the residual actually certifies.
4. **Why stiff stability does not guarantee the expected order**
   A differential equation is constructed to have a smooth, known solution and a rapidly damped error mode. Use it to distinguish stability of the homogeneous equation, tracking of a moving target and the effect of off-grid stage forcing.
5. **How far does an energy-preserving rotation lag behind?**
   Integrate uniform circular motion by the implicit midpoint rule. The points stay exactly on the energy circle in exact arithmetic. Derive the angle advanced at each step and measure the accumulated timing error.
6. **From existence of a trajectory to the order of a time step**
   Establish what an initial-value solver approximates, derive Runge–Kutta order conditions for a general smooth vector field and construct an alternative high-order step by extrapolation.
7. **Why consistent multistep formulas can fail under refinement**
   Derive the root condition for constant steps, propagate a local defect to a fixed end time and isolate a growing parasitic mode with a controlled starting error.

8. Why an accurate position can hide an inaccurate constraint force

Derive an energy test for implicit Runge–Kutta stages, then recover missing derivatives in a constrained model and calculate the accuracy lost by a multiplier.

Dynamics and numerical time integration · 1. Find a crossing between accepted time steps

An integrator can approximate a trajectory accurately yet report a poor switch time. Construct a continuous interpolant, relate a state error to an event-time error and inspect what happens when a trajectory merely touches a threshold.

An event is a root of a composite function

Let $y' = f(t, y)$ and let a sensor or change of dynamics be described by $g(t, y) = 0$. Along an exact trajectory the scalar event function is $h(t) = g(t, y(t))$. A simple event at t_* satisfies

$$\dot{h}_* = g_t + g_y f \neq 0.$$

If approximate trajectory or sensor evaluation introduces error δh, linearisation around the crossing gives

$$\delta t \simeq -\delta h / \dot{h}_*.$$

A small state error can therefore imply a much larger time error if the crossing is nearly tangential. Reporting an event timestamp to machine precision after bisecting a low-order interpolant is not equivalent to knowing the physical crossing time that accurately.

Consider a dimensionless unit-frequency oscillator with $y = \cos t,\ y' = -\sin t$ during the first half cycle. Its first falling crossing of threshold $a \in (0, 1)$ occurs at $t_* = \arccos a$, with slope $-\sqrt{1 - a^2}$. At $a = 0.5$, a positive vertical interpolation error 10^{-3} can delay the crossing by about 0.00115 time units. At $a = 0.9999$ the same linear estimate is about 0.0707, comparable with or larger than the event time itself; the simple linear local approximation then also fails. At precisely $a = 1$ the surface is tangent at the maximum and sign-change tests can miss it altogether.

Construct a state and slope interpolant on one accepted interval

Suppose samples of state and derivative are available at $t_n, t_{n+1} = t_n + h$. Write $s = (t - t_n)/h$ and choose a cubic $p(s)$ matching $p(0) = y_n, p(1) = y_{n+1}, p_s(0) = h f_n, p_s(1) = h f_{n+1}$. Solving these four linear conditions gives

$$p = (2s^3 - 3s^2 + 1) y_n + (s^3 - 2s^2 + s) h f_n + (-2s^3 + 3s^2) y_{n+1} + (s^3 - s^2) h f_{n+1}.$$

For an exact scalar trajectory sufficiently smooth in time, the Hermite error is $y(t) - p(t) = y^{(4)}(\xi)(t - t_n)^2 (t - t_{n+1})^2 / 4!$, bounded by $h^4 \max |y^{(4)}| / 384$. Linear interpolation instead has error bounded by $h^2 \max |y''| / 8$. Combine either bound with a known lower bound on $|\dot{h}|$ in a bracket to bound event-time error. In an actual ODE solve, y_n, f_n are numerical, so interpolation and integrated state errors both contribute. A high-order interpolant cannot repair incorrect endpoint data or sample a discontinuity assumed smooth.

Bracket on the physical trajectory and restart after a transition

At each accepted interval compare event values at both ends. For a simple crossing of the requested direction, solve $g(t, p(t)) = 0$ inside that interval using a bracketed root finder. The bracket controls the algebraic root-finding error; independent time-step refinement measures trajectory error. If the dynamics change at the event, truncate the old step there, apply any specified reset $y^{+} = R(t_{*}, y^{-})$ and restart the integrator under the new right side. Propagating the old smooth interpolant through a change of dynamics can corrupt both state and subsequent event ordering.

If a surface stays exactly active over a finite interval, repeated detection at the same time can stall the integration. A model of contact, hysteresis or directional rearming may be physically appropriate, depending on what the event means. Merely nudging the time forward by an arbitrary tolerance can skip a nearby second event. Similarly two opposite crossings within one large accepted interval may yield equal endpoint signs. A maximum step guided by known physical timescales, supplementary extrema detection or an analytically enclosed event function can be needed. Accuracy control on the smooth state alone does not guarantee detection of every specified surface crossing.

Compare interpolation errors independently of the propagation error

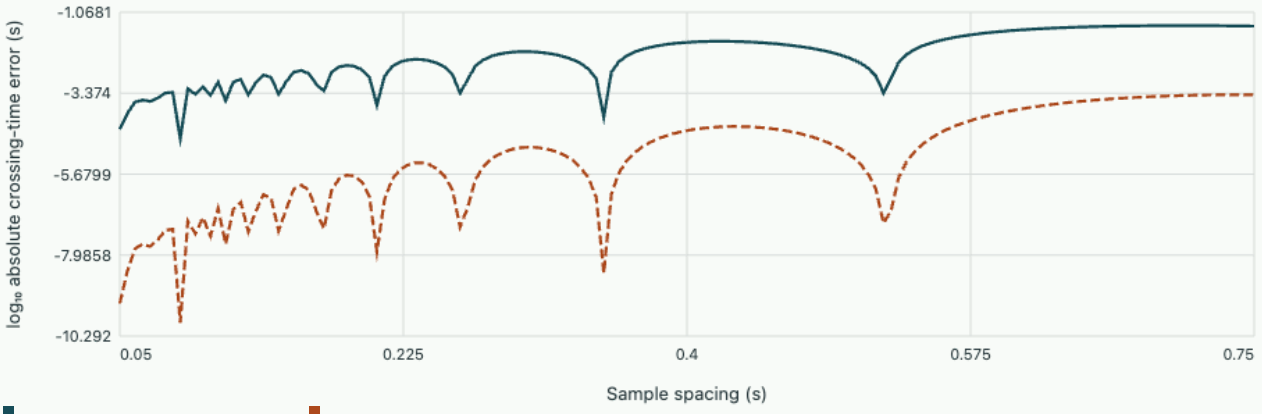
The calculator uses exact cosine endpoint samples and their derivatives, then locates the first falling crossing on a uniform time grid by linear or cubic Hermite interpolation. The curves show error against step spacing on a logarithmic vertical scale. Their irregular minima occur when a grid point happens to coincide with the event. Look for the error envelope over several spacings. At threshold 0.5 the exact result is 1.047198 s. Moving the threshold changes both the crossing slope and its position relative to the sampled grid.

Open this experiment online

Try the calculation

Crossing level **0.5**

Inspection step (s) **0.4**



Linear interpolation Cubic Hermite interpolation

Exact crossing 1.047198 s. At the selected step: linear 1.035331 s, Hermite 1.047161 s. Exact endpoints isolate interpolation error.

Distinguish uncertainty in the threshold from time discretisation

Two clocks can agree on an event and still report a poorly determined arrival time if the physical threshold is uncertain. For a calibrated level a the falling crossing solves $\cos t_* = a$, so implicit differentiation gives

$$dt_*/da = -1/\sqrt{1-a^2}.$$

A level error of 0.002 near $a = 0.5$ produces a timing uncertainty of about 0.00231 s for the unit-frequency benchmark. At $a = 0.9$ the same device gives 0.00459 s. Close to the turning point the trace is almost horizontal. Reducing the root finder's tolerance cannot remove that physical loss in sensitivity.

If the numerical trajectory also has a known absolute state error bound η, divide that bound by a lower bound on the crossing speed in the bracket. The result is a deterministic timing allowance. An independent random threshold error has a standard deviation and calls for a statistical interpretation. Adding those numbers in quadrature would need a probability model for both. Reporting the two separately preserves what the measurement and the solver actually established.

Tutorial · assess a reported event time

Get a crossing from just two samples

Use $y = \cos t,\ a = 0.5$ with samples at $t = 0.8, 1.2$. Compute the linearly interpolated time and its error. Why does its sign follow from curvature?

▼ **Work through the solution**

The endpoint values are 0.696707 and 0.362358; the fractional position is $(0.696707 - 0.5)/(0.696707 - 0.362358) = 0.5883$, giving $t_L \simeq 1.0353$. It is about 0.0119 s early. Over this interval $\cos t > 0$, so $y'' = -\cos t < 0$. A straight chord lies below the concave curve; the falling chord reaches the threshold before the true trajectory.

Bound a Hermite crossing-time error

On the same interval use $|y^{(4)}| \le 1,\ |\dot{y}| \ge \sin 0.8$. Give a simple guaranteed bound assuming exact endpoints and exact solution of the interpolating cubic root.

▼ **Work through the solution**

The state interpolation bound is

$$0.4^4/384 = 6.67 \times 10^{-5}.$$

Dividing by $\sin 0.8 = 0.7174$ gives 9.29×10^{-5} s. This uses monotonicity of the exact trajectory to convert a vertical defect at the computed root into a horizontal difference. It does not assume the cubic derivative equals the true derivative. Numerical endpoint error would need to be added in state units first.

Distinguish a tangency from a missed sign change

A gap sensor follows $h(t) = (t-1)^2 - \delta$. Compare event detection at $\delta = 0, 10^{-4}, -10^{-4}$ over the interval $[0.9, 1.1]$.

▼ **Work through the solution**

Both endpoints are positive in all three cases. At zero there is a repeated touching event. At positive δ there are two true crossings at 0.99 and 1.01 with opposite directions. At negative δ there is no root. Locating and evaluating the internal minimum distinguishes the cases when the uncertainty is below $|\delta|$. A sign test at just the endpoints cannot determine any of these distinctions, however accurately it evaluates those endpoints.

Further reading

- ETH Zürich · numerical methods for CSE, chapter 11 on numerical time integration
- SciPy solve_ivp · continuous output and sign-change event detection
- Course investigation · adaptive time stepping
- Compare time-step error with conservation

Dynamics and numerical time integration · 2. A delayed feedback model needs a history, not one starting value

A controller responds to the measured displacement one time delay ago. Construct its solution in successive intervals, track propagated derivative discontinuities and derive the point at which increasing delay can destabilise negative feedback.

Specify the additional data an ordinary initial value omits

For $t \geq 0$ consider $y'(t) = -ay(t-\tau)$ with positive constant gain a and delay $\tau > 0$. The derivative at the start depends on $y(-\tau)$; an initial value $y(0)$ alone is insufficient. Supply a history $y(t) = \phi(t)$ on $[-\tau, 0]$, continuous at zero for a continuous state. Two preparations sharing $y(0)$ but different earlier values can immediately develop different slopes. This is why treating a delay as a new algebraic coefficient in an ODE solver without a history buffer changes the physical problem.

Take constant history $\phi = 1$. For $0 \leq t \leq \tau$, the delayed input is one and $y = 1 - at$. For $\tau \leq t \leq 2\tau$, the derivative is $-a[1 - a(t-\tau)]$; integrate from $t = \tau$ to get

$$y = 1 - at + a^2(t-\tau)^2/2.$$

On the next interval another integral gives

$$y = 1 - at + a^2(t-\tau)^2/2 - a^3(t-2\tau)^3/6.$$

Each new interval solves an ordinary forced equation with known earlier right-side data. This is the method of steps. A smooth-looking initial history can nevertheless mismatch derivatives: here the left derivative at zero is zero while the equation supplies $-a$ on the right.

Trace how a step mismatch propagates through the delay

At $t = \tau$ continuity of y at zero makes y' continuous, but $y''(t) = -ay'(t-\tau)$ inherits the earlier jump in y'. At 2τ the discontinuity appears in the third derivative. High-order adaptive ODE error formulas assume smooth derivatives within an interval. Marking known breakpoints and integrating separately across the delay intervals avoids applying a high-degree interpolant through an unacknowledged loss of smoothness. With several delays, sums of delay lengths can generate additional event locations and their bookkeeping matters. A discontinuity in the physical forcing can also generate new fronts of reduced regularity.

Within a method-of-steps solver, values $y(t-\tau)$ usually lie between stored previous mesh points. Dense output supplies their interpolation. Its error feeds directly into the next interval's right side and can limit global accuracy even when the ODE time-step rule has higher formal order. Maintain enough history to cover the largest delay and requested outputs, and preserve the known one-sided behaviour at breakpoints. If a trial step is longer than the delay its stages may refer to the current unknown interval; an algorithm must either constrain that step or solve the resulting within-step interpolation

problem consistently. A generic lookup of the last stored value instead creates an unintended zero-order-held feedback law.

Background · Can smooth data prevent the first derivative jump?

A constant history $\phi(t) = 1$ has zero left derivative at the start. The equation $y' = -a\,y(t-\tau)$ demands right derivative $-a$, creating an immediate mismatch. A history with $\phi(-\tau) = 1$ and $\phi'(0^-) = -a$ instead matches the first derivative. For example, choose $\phi(t) = 1 - a(t+\tau)$ on the history interval.

Matching a first derivative does not guarantee a smooth second derivative. Differentiating the equation once shows that the next matching condition would be

$$\phi''(0^-) = -a\,\phi'(-\tau^+).$$

For the linear history, the two sides are zero and a^2. The mismatch has moved up by one derivative. A high-order interpolant across the start may therefore still lose order even though the trajectory and its tangent look continuous.

Derive a delay-induced change of oscillatory stability

For a constant-coefficient retarded equation try $y = e^{st}$. Substitution gives

$$s + ae^{-s\tau} = 0.$$

Unlike a polynomial characteristic equation, this transcendental equation generally has infinitely many roots. A pair crosses the imaginary axis when $s = i\omega$, giving $a\cos(\omega\tau) = 0,\ \omega - a\sin(\omega\tau) = 0$. For positive ω, the first crossing is at $\omega = a,\ a\tau = \pi/2$. With negligible delay the model behaves like the stable ODE $y' = -ay$. The crossing indicates that negative feedback acting on old information can become destabilising; stability of the zero-delay ODE is no guarantee for arbitrary delay.

To check crossing direction, differentiate $s + ae^{-s\tau} = 0$ at constant a, obtaining $ds/d\tau = -s^2/(1+\tau s)$ after using the characteristic equation. At $s = ia$ its real part is $a^2/[1+(a\tau)^2] > 0$. Increasing delay moves this pair into the unstable half-plane. A numerically observed sign-changing transient alone does not prove instability, as stable delay systems can oscillate while decaying. Compare an envelope over successive periods with the characteristic roots and refine both time discretisation and history interpolation. An unstable numerical discretisation can otherwise imitate a feedback-induced growth.

Compare a stored history with the method-of-steps solution

The delay is the time unit and the history on the preceding interval is one. The calculator integrates an explicit Euler update with an integer number of steps per delay so it can read the delayed state exactly

at a stored grid index. The comparison curve evaluates the exact piecewise polynomial. Its kth contribution starts after k minus one delay intervals and has coefficient given by the kth power of minus the dimensionless feedback divided by k factorial. This follows by repeatedly integrating the previous interval's polynomial, so it is independent of the Euler recurrence.

At feedback one, inspect the exact values after one, two and three delays before refining the grid. Euler already integrates the constant derivative on the first interval exactly, while errors enter on later intervals. Change the feedback across π/2 and compare the oscillation envelope. Six displayed delays are useful for checking the calculation but can be too short to measure slow growth close to threshold. A stable-looking short record does not replace the characteristic-root analysis. Because the grids here align with the delay, this experiment isolates stepping error and does not test a general off-grid history interpolator.

Open this experiment online

Try the calculation

Dimensionless feedback aτ **1**

Steps per delay **10**



Largest grid error 0.0642. First continuum instability at aτ = π/2 ≈ 1.571; current aτ = 1.00.

Tutorial · distinguish history dependence from numerical error

Compute three exact method-of-steps values

For $a = 1\,\mathrm{s}^{-1}$, $\tau = 1$ s and constant unit history, find $y(1), y(2), y(3)$. Does a negative response at two seconds prove instability?

▼ **Work through the solution**

The first value is zero. The second is

$$1 - 2 + 1/2 = -0.5.$$

The third is

$$1 - 3 + (3-1)^2/2 - (3-2)^3/6 = -1/6.$$

A negative value reflects delayed feedback overshoot. With $a\tau = 1 < \pi/2$ the zero solution is asymptotically stable for this model even though the transient crosses zero. The exact interval polynomials provide stringent tests for a history interpolator at the first few breakpoints.

Construct incompatible histories with the same initial value

Keep $a = \tau = 1$ and compare $\phi_1(t) = 1$ with $\phi_2(t) = 1 + t$ on $[-1, 0]$. Obtain the distinct states on the first interval.

▼ **Work through the solution**

For the first history $y_1 = 1 - t$. For the second, $\phi_2(t-1) = t$, so $y_2' = -t$ and $y_2 = 1 - t^2/2$. At $t = 1$ their values are 0 and 0.5 although both began at 1. Their initial right derivatives differ by one. A restart record containing only time and last state cannot reconstruct the interrupted integration; retaining the relevant past function is part of the state representation.

Compare physical and discrete stability boundaries

A controller has $a = 4\,\mathrm{s}^{-1}$. Estimate the first critical delay and the oscillation period there. What numerical checks would separate it from a coarse-step instability?

▼ **Work through the solution**

The first critical delay is

$$\pi/(2a) = 0.3927\,\mathrm{s}.$$

Its frequency is $a/(2\pi) = 0.6366$ Hz and period $2\pi/a = 1.571$ s. Change the delay on both sides of the predicted value, resolve the oscillation with multiple step sizes and compare measured growth with the characteristic equation. Keep history interpolation accurate while refining; fixed low-order delayed lookup can determine the observed stability error even as the main integrator step is reduced.

Further reading

- Shampine and Thompson · Solving DDEs in MATLAB

- SciML · a worked calculation with a history function and the method of steps
- MATLAB dde23 · history functions and derivative discontinuities
- Course investigation · multistep integration and starting histories
- Compare time-step error with conservation

Dynamics and numerical time integration · 3. Solving a two-point boundary problem without guessing the start slope

When endpoint conditions act on different sides of an interval, shooting can amplify a tiny initial error. Construct a cubic collocation system, test its continuous differential residual and quantify which boundary-value solution the residual actually certifies.

Sensitivity of shooting need not be sensitivity of the physical solution

Take $y'' = \kappa^2 y$ on $0 \le x \le 1$, with $y(0) = 1,\ y(1) = 0$. The solution $y = \sinh[\kappa(1-x)]/\sinh\kappa$ decays from the left. To shoot from the left, introduce unknown slope s, yielding endpoint $y(1) = \cosh\kappa + (s/\kappa)\sinh\kappa$. The correct slope is $-\kappa\coth\kappa$ and its error is magnified by $\sinh\kappa/\kappa$ at the far end. For $\kappa = 20$ this factor is about 1.21×10^7. Even an initial slope accurate to 10^{-10} can cause a far-end mismatch of order 10^{-3}. The mathematical boundary solution can remain well behaved while this particular representation suffers cancellation of large growing and decaying components.

A collocation boundary solver instead treats approximate states at many locations as simultaneous unknowns and enforces both boundary conditions directly. Write the equation as first-order system $Y' = (v, \kappa^2 y)^T = F(x, Y)$, $Y = (y, v)$. On each panel $[x_i, x_{i+1}]$ choose a cubic Hermite polynomial using unknown endpoint states and endpoint slopes fixed by F_i, F_{i+1}. This already imposes the differential equation at both endpoints. Require the polynomial derivative to satisfy the ODE also at the midpoint. Matching yields two equations for each panel, with the boundary equations closing the overall algebraic system.

Derive a residual with only three right-side evaluations per panel

For panel width h, Hermite interpolation at the midpoint gives

$$Y_m = (Y_i + Y_{i+1})/2 + h(F_i - F_{i+1})/8.$$

Its derivative there is $p'_m = 3(Y_{i+1} - Y_i)/(2h) - (F_i + F_{i+1})/4$. Equating this with $F_m = F(x_i + h/2, Y_m)$ yields

$$Y_{i+1} - Y_i - \frac{h}{6}(F_i + 4F_m + F_{i+1}) = 0.$$

This is a three-point Lobatto collocation relation, resembling Simpson quadrature with a midpoint state determined by the same cubic. Neighbouring panels share endpoint states and slopes, constructing a continuously differentiable piecewise approximation for a smooth right side. The nonlinear algebraic Jacobian has local block structure apart from boundary conditions and a few global unknowns. Dense finite differencing of the entire state would discard that structure and can dominate a large adaptive solve.

As a simple independent test take the scalar ODE $y' = \lambda y$. With $z = h\lambda$, substitution of Y_m yields a one-panel transfer $R(z) = (1 + z/2 + z^2/12)/(1 - z/2 + z^2/12)$, agreeing with e^z through z^4. The symmetry $R(-z) = 1/R(z)$ checks the derivation. At large negative z, $R(z) \to 1$, so the local solution formula is not strongly damping for unresolved stiff modes. Global enforcement of boundary conditions and adaptive spatial resolution remain important: a stable algebraic solve need not resolve a very thin physical layer.

Interpret a continuous residual through a stability estimate

A collocation residual vanishes at the points where it was imposed, so checking only those values tests the nonlinear algebraic solve, not the differential approximation between points. Evaluate the continuous defect $d(x) = p'(x) - F(x, p(x))$ at independent points or use its suitably weighted panel norm. Refine where the residual is large, especially where the solution changes rapidly, and enforce the boundary conditions independently. A mesh with many uniform far-interior points can inefficiently resolve a solution having a layer of thickness $1/\kappa$ near a single wall.

For a scalar linear elliptic boundary problem $-y'' + \kappa^2 y = f$ with zero endpoint error, an approximation with smooth enough derivatives has error governed by $-e'' + \kappa^2 e = r$. If $\|r\|_\infty \le \epsilon$, a maximum-principle argument gives $\|e\|_\infty \le \epsilon/\kappa^2$ for $\kappa > 0$: at an interior positive error maximum, $-e'' \ge 0$, and similarly for the negative minimum. This bound links a differential residual in the same formulation to solution error under the stated boundary conditions. A residual in a differently scaled first-order system requires its own inverse stability estimate. Near a nonunique boundary solution an apparently tiny defect can correspond to a much larger state error.

Reconstruct the entire cubic boundary solution

The calculator solves the same linear collocation equations as the written panel derivation. For this constant matrix, its growing and shrinking eigendirections decouple and one panel applies the stated rational transfer to each. The numerical field shown between nodes is the actual cubic Hermite interpolant through those collocation states and their slopes. The reference is the independently evaluated hyperbolic-sine boundary solution. Increase the layer parameter at fixed panel count first and look just inside the left boundary. Both curves hit the imposed endpoints even on a visibly inadequate mesh.

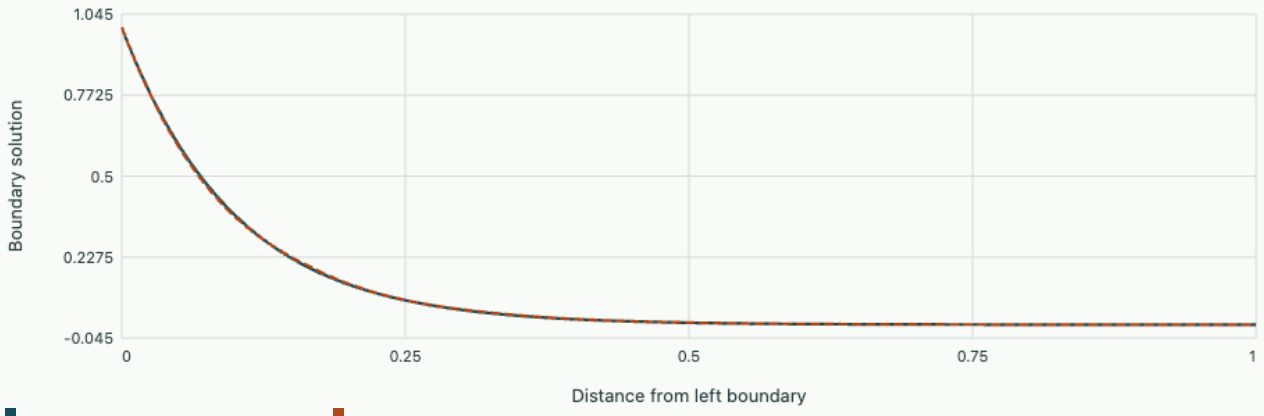
Then double the number of panels. The displayed maximum difference is measured on 241 independent evaluation locations and is a sampled diagnostic, not a guaranteed global norm. For large layer parameter the shooting endpoint amplification can be enormous even while the collocation solution becomes accurate upon mesh refinement. At only a few cells across the layer the cubic may bend below zero. A solution satisfying the nonlinear algebraic conditions of the chosen discretisation need not preserve positivity of the continuum field at every point.

Open this experiment online

Try the calculation

Layer parameter κ **10**

Uniform panels **6**



Sampled maximum field error 0.00674; shooting endpoint amplification sinh(κ)/κ = 1101.

Tutorial · distinguish mesh, boundary and algebraic errors

Check a cubic against an exactly representable solution

Choose $y' = 3x^2,\ y(0) = 0$ on a single unit panel. Verify the collocation residual using the candidate $y = x^3$ and identify the midpoint state from endpoint data.

▼ Work through the solution

Endpoint values are 0 and 1, slopes 0 and 3. The Hermite midpoint value is $1/2 - (3/8) = 1/8$, equal to $(1/2)^3$. The midpoint slope is $3/4$. Simpson's right side gives $[0 + 4(3/4) + 3]/6 = 1$, equal to the endpoint difference. Here the entire state polynomial, not just its grid values, is exact. This manufactured case checks signs and factors before using the solver on a stiff boundary layer.

Quantify endpoint sensitivity for one shooting trial

At $\kappa = 10$, an independent integration has a starting-slope error 2×10^{-7}. Estimate its endpoint error. Would measuring that endpoint residual accurately imply the underlying boundary problem is ill conditioned?

▼ Work through the solution

The amplification is $\sinh 10/10 \simeq 1101.3$, yielding endpoint error 2.20×10^{-4}. This exposes sensitivity of a one-way initial-value representation to the guessed slope. A boundary perturbation instead propagates through bounded functions $\sinh[\kappa(1-x)]/\sinh\kappa$ and $\sinh(\kappa x)/\sinh\kappa$. Its effect on interior y need not contain the huge shooting factor. Distinguish sensitivity of the original boundary data from cancellation in one numerical strategy.

Spot a false residual certificate

A computation checks only $p' - F = 0$ at each panel's two ends and midpoint and reports error below 10^{-12}. What evidence is still needed for a claimed 10^{-6} physical solution tolerance?

▼ **Work through the solution**

Those are exactly the collocation equations; their tiny residual tests the solved discrete system. Evaluate the defect off those nodes or estimate its norm by an independent quadrature, verify boundary mismatches, and relate these defects to solution error using a suitable stability bound or independent reference refinement. For the second-order model with $\kappa = 5$ and zero endpoint error a uniform residual below 25×10^{-6} in $-p'' + 25p - f$ suffices for the stated magnitude bound. A reported first-order-system residual with different units cannot be inserted unchanged.

Further reading

- Kierzenka and Shampine · a boundary-value solver controlling residual and error
- MATLAB bvp4c · three-stage Lobatto collocation
- Course investigation · sensitivity of boundary-value shooting
- Compare time-step error with conservation

Dynamics and numerical time integration · 4. Why stiff stability does not guarantee the expected order

A differential equation is constructed to have a smooth, known solution and a rapidly damped error mode. Use it to distinguish stability of the homogeneous equation, tracking of a moving target and the effect of off-grid stage forcing.

Choose an exact solution before increasing stiffness

Let $\lambda < 0$, let $\phi(t)$ be a smooth specified function, and solve the Prothero–Robinson model $y' = \lambda[y - \phi(t)] + \phi'(t)$, $y(0) = \phi(0)$. The exact solution is $y(t) = \phi(t)$ for every λ. For an initial mismatch e_0, the error evolves $e(t) = e_0 e^{\lambda t}$. Thus the observed slow solution and fast relaxation timescale can be varied independently. This makes the model an incisive verification case for a stiff integrator: changing λ leaves the exact waveform fixed while altering how errors created by discrete stages are damped.

A stability function $R(h\lambda)$ describes propagation of errors for the homogeneous test $y' = \lambda y$. A-stability constrains $|R(z)| \leq 1$ over the left half-plane. L-stability additionally requires $R(z) \to 0$ as $|z| \to \infty$ in the stable stiff limit. Neither definition alone describes how a method evaluates a time-dependent forcing at its internal stages. Stage order and stiff accuracy can matter when those stage errors are multiplied by large $|\lambda|$. Observing bounded output on a homogeneous damping test does not establish a method's full formal order on a forced stiff problem.

Calculate the error recurrence of backward Euler and the trapezoidal rule

Write $e_n = y_n - \phi_n,\ z = h\lambda$. Backward Euler evaluates the whole right side at t_{n+1}. Collect terms after subtracting the exact target:

$$(1 - z)e_{n+1} = e_n + \phi_n - \phi_{n+1} + h\phi'_{n+1}.$$

The forcing defect begins at $h^2\phi''_{n+1}/2$. Its effect on the next error is divided by $1 - z$. The rapidly decaying homogeneous mode is also divided by that factor. This can make backward Euler track a particular smooth imposed target surprisingly well in the very stiff regime even though its classical nonstiff order is one. Its behaviour should be assessed in the desired observables and step–stiffness range, rather than ranked solely by the formal method label.

For the trapezoidal rule the corresponding equation is

$$(1 - z/2)e_{n+1} = (1 + z/2)e_n + \phi_n - \phi_{n+1} + \tfrac{h}{2}(\phi'_n + \phi'_{n+1}).$$

The defect is the trapezoidal quadrature error in the derivative and begins at order h^3. However, its homogeneous factor tends to -1 when $z \to -\infty$, so a pre-existing fast error can survive as a sign-alternating numerical component for many steps. With $z = -100$ the factors are $1/101 = 0.00990$ for backward Euler and $-49/51 = -0.96078$ for trapezoid. Both schemes are A-stable. Only the first

immediately damps this unresolved mode. Event resets, inconsistent initial preparation or changing external loads can seed it.

A midpoint constraint can force endpoint drift

Implicit midpoint uses $y_{n+1} - y_n = h[\lambda((y_n + y_{n+1})/2 - \phi_m) + \phi'_m]$. For fixed h and large negative λ, finite numerical states require the stage average to approach ϕ_m. Thus $y_{n+1} \simeq 2\phi_m - y_n$. The endpoints inferred from this interior constraint need not land on ϕ_{n+1}. Compare with trapezoid, which shares exactly the same scalar homogeneous stability function but evaluates two different times and states for nonlinear or time-dependent forcing.

With $y_n = \phi_n$ and $\phi(t) = t^2$, the limiting endpoint error is

$$2\phi(t_n + h/2) - \phi_n - \phi_{n+1} = -h^2/2.$$

The exact finite-z midpoint error starting from the target is $zh^2/[4(1 - z/2)]$. The trapezoidal derivative quadrature, in contrast, is exact for this quadratic target and keeps $e_n = 0$ at every step started exactly on it. Their common stability function did not predict their different forced response. In the midpoint example an alternating error can cancel at some output times, so inspecting only the final point at an even number of steps may obscure the largest per-step error. Compare all sampled times and separate propagated from newly injected defects.

Separate injected forcing error from a starting mismatch

This experiment uses the target t squared with step 0.1 s. It propagates the exact error recurrences of backward Euler, trapezoid and midpoint rather than estimating the difference by subtracting two large trajectories. The reference also includes the exponentially decaying initial mismatch. Move the stiffness control until a time step contains a hundred fast relaxation times. With a nonzero initial error, trapezoid and midpoint preserve a visible alternating component even though the exact mismatch essentially disappears after one step.

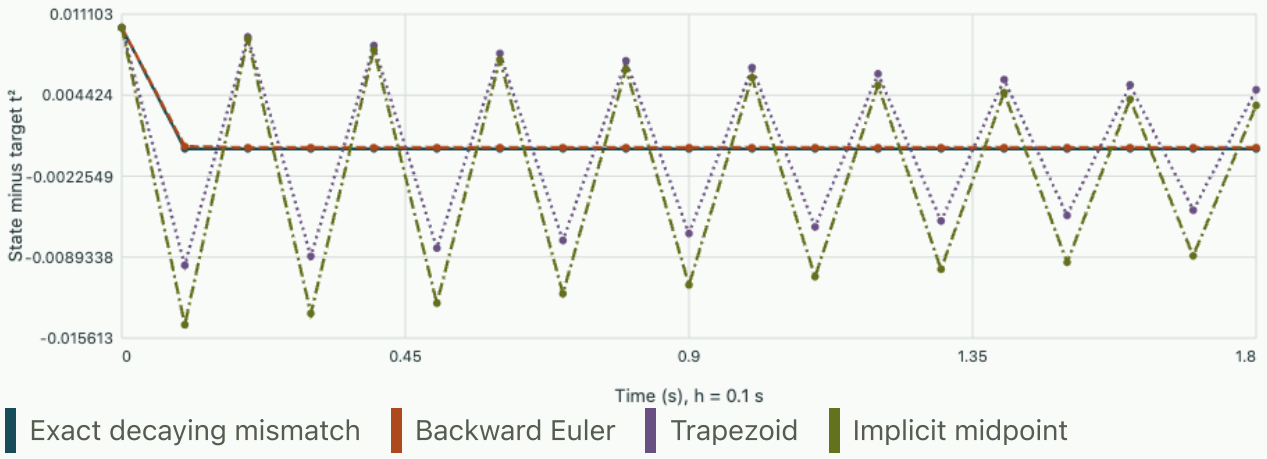
Set the initial mismatch to zero to isolate what an initially accurate trajectory gains from stage forcing. For this one quadratic target the trapezoidal derivative quadrature is exact and introduces no error, while midpoint still misses the endpoint because it constrains the value at the middle of the interval. Try a less stiff value and compare the peak and endpoint errors over all times. The demonstration is deliberately a manufactured exact target; substituting a cubic target would remove trapezoid’s special cancellation while keeping the same homogeneous stability function.

Open this experiment online

Try the calculation

log₁₀ of |hλ| 2

Initial mismatch 0.01



Exact decaying mismatch | Backward Euler | Trapezoid | Implicit midpoint

hλ = -100, homogeneous midpoint/trapezoid factor -0.9608, backward Euler factor 0.0099. Initial mismatch is shown relative to the target.

Tutorial · compare error mechanisms at fixed work

Compute a forced stiff step exactly

Set $h = 0.10, \lambda = -1000, \phi = t^2,\ y_0 = 0$. Compute errors after the first step for backward Euler, trapezoid and midpoint.

▼ **Work through the solution**

Here $z = -100$, $\phi(h) = 0.01$. Backward Euler defect is h^2, giving $e_1 = 0.01/101 = 9.90 \times 10^{-5}$. Trapezoid yields zero error on this target. Midpoint gives $zh^2/[4(1 - z/2)] = -0.004902$, so its predicted y_1 is only 0.005098. The exact target is smooth in all three experiments; the fast stiffness multiplies the way its curvature is sampled within the step.

Check an unresolved initial-layer perturbation

Keep $z = -100$ and introduce an initial error of 0.01. How large is the propagated part after two steps, excluding new forcing defects?

▼ **Work through the solution**

The exact propagated part is $0.01e^{-200}$, negligible here. Backward Euler gives

$$0.01/101^2 = 9.80 \times 10^{-7}.$$

Both trapezoid and implicit midpoint give $0.01(49/51)^2 = 0.00923$ with restored positive sign after the second step. Negative sign at alternate steps can be invisible if the code saves every second state. Monitoring stage and consecutive-step output helps identify this persistence.

Design a refinement that distinguishes ordinary order from stiff forcing error

Why are a study at fixed λ with $h \to 0$ and one at fixed $h\lambda = -100$ different experiments? What should be plotted?

▼ **Work through the solution**

At fixed λ, sufficiently small h eventually resolves the fast timescale and standard smooth-solution order can reappear. Holding $h\lambda = -100$ changes the differential equation's error-relaxation scale at every refinement, deliberately keeping that fast mode unresolved while the exact target stays fixed. Plot maximum error over all physical sample times, the homogeneous amplification and representative stage defects for both sequences. Report the exact initial mismatch as well as ϕ. Otherwise a measured order can conflate target tracking, unremoved fast transients and cancellation at one favourable output time.

Further reading

- ETH Zürich · numerical methods for CSE, time integrators and stability
- Prothero and Robinson · stability and accuracy of one-step methods for stiff systems
- Course investigation · stiff numerical stability
- Compare time-step error with conservation

Dynamics and numerical time integration · 5. How far does an energy-preserving rotation lag behind?

Integrate uniform circular motion by the implicit midpoint rule. The points stay exactly on the energy circle in exact arithmetic. Derive the angle advanced at each step and measure the accumulated timing error.

From a midpoint balance to a rotation matrix

Write a two-component oscillator as $v' = \omega J v$, where J sends (x, y) to $(-y, x)$. Choosing scaled coordinates makes both components comparable; for a mechanical oscillator one may take displacement and a suitably scaled speed. The generator is skew-symmetric, so $d(v^T v)/dt = \omega v^T (J^T + J) v = 0$. The exact motion has angular advance $h\omega$ over an interval h.

Midpoint approximates the velocity at the midpoint of the segment joining the old and new states. This yields $(I - aJ)v_{n+1} = (I + aJ)v_n$ with $a = h\omega/2$. Since $J^2 = -I$, multiplication by $I + aJ$ eliminates the inverse and gives

$$R_h = (I - aJ)^{-1}(I + aJ) = \frac{(1 - a^2)I + 2aJ}{1 + a^2}.$$

The coefficients of I and J have squares summing to one. They are the cosine and sine of the angle $\theta_h = 2 \arctan a$. Thus every update preserves the squared radius. The angle differs from the exact $h\omega$. Expanding the arctangent gives

$$\theta_h = h\omega - (h\omega)^3/12 + O((h\omega)^5).$$

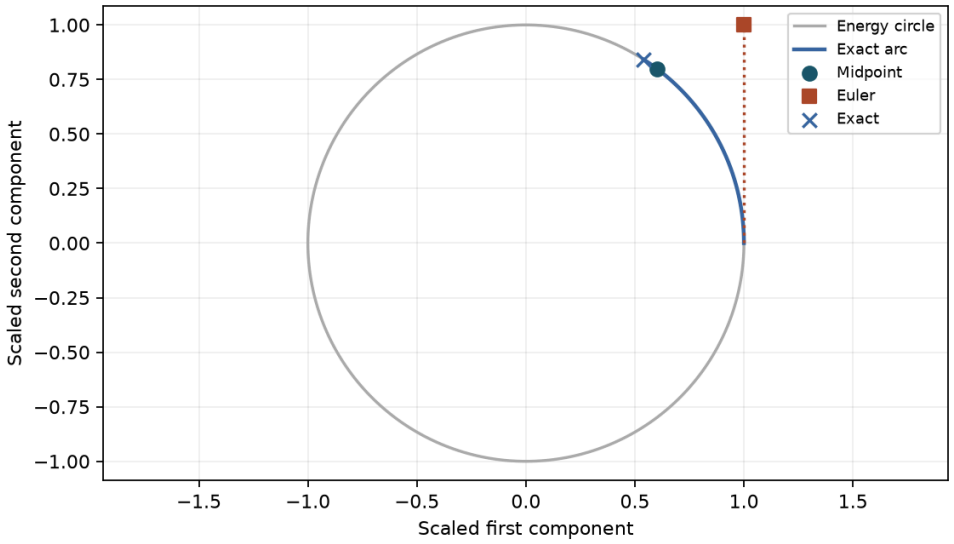
Over a fixed duration $T = nh$, the leading accumulated angle error is $-T\omega^3 h^2/12$. This phase lag is especially visible after many periods even when a graph of energy looks perfect.

For comparison, explicit Euler takes the tangent line to its end point. Its amplification is $\sqrt{1 + (h\omega)^2}$. Studying both amplitude and timing identifies different defects of the two updates.

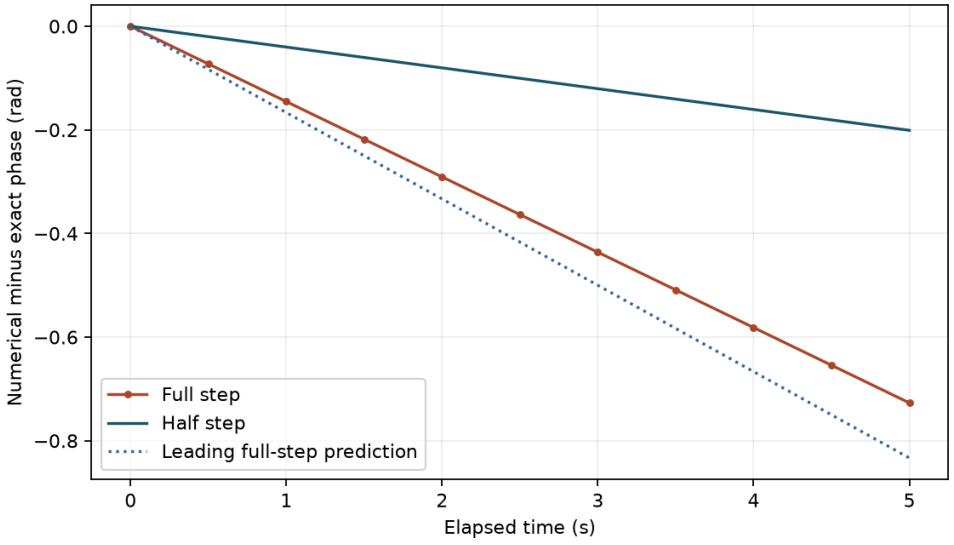
Put a clock on the unit circle

Let $\omega = 2$ rad/s, $h = 0.5$ s and $v_0 = (1, 0)$. The half-step parameter is 0.5. The rotation sends the first point to $(0.6, 0.8)$ instead of the exact $(\cos 1, \sin 1) \approx (0.5403, 0.8415)$. Both radii are one. The angular deficit is $1 - 2\arctan(0.5) \approx 0.072705$ rad per update. After ten updates, the numerical clock is behind by 0.72705 rad or about 0.364 s of physical time.

With $h = 0.25$ s over the same five seconds the deficit is $10 - 40\arctan(0.25) \approx 0.20085$ rad. Its ratio to the coarse-step error is close to one quarter. In the notebook the rotation is built by a linear solve and its angle is measured from a dot product and an oriented area. These are independent ways to check the hand formula. Counting full turns requires retaining unwrapped increments; a final point alone cannot identify the number of completed turns.



The three markers correspond to one physical time step. The midpoint result lies on the circle at its own numerical angle.



The full and half steps cover the same physical interval. The dotted trace is the leading asymptotic error for the coarse step.

Exercise · reverse the sign of time

Show why a forward midpoint update followed by one equal backward update returns to the starting state. Does forward Euler have the same property here?

▼ Work through the solution

Replacing a by $-a$ interchanges the two factors in the Cayley formula, so $R_{-h} = R_h^{-1}$. Their product is the identity. For explicit Euler the product is $(I - h\omega J)(I + h\omega J) = (1 + (h\omega)^2)I$. The reversal increases every radius when the angular speed and the step are nonzero.

Exercise · set a phase budget

For $\omega = 2$ rad/s, approximate a time step that keeps the accumulated phase lag below 0.02 rad over six seconds. Give an exact way to check the choice.

▼ **Work through the solution**

The leading magnitude is $T\omega^3 h^2/12 = 4h^2$. Equating this to 0.02 gives about 0.0707 s. Choosing 90 equal steps over six seconds gives $h = 1/15$ s. The exact lag is $12 - 180\arctan(1/15)$ rad, about 0.0177. The exact check is useful when the step is large enough that the cubic series term has limited accuracy.

Change parameters · Explore this notebook model

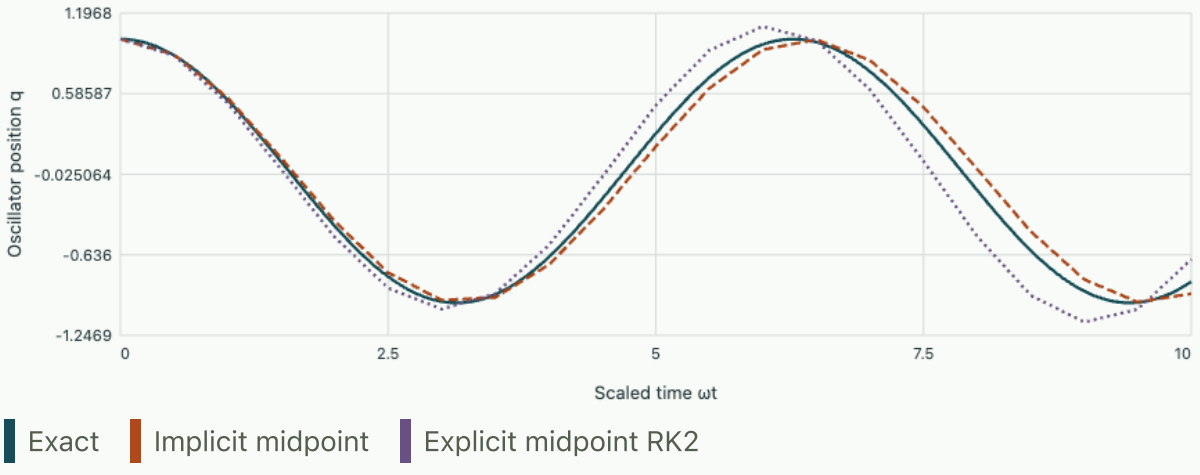
Change one input at a time and compare the recalculated curves with the argument above. The browser evaluates this small model locally; the accompanying numerical values state its assumptions and reference checks. Reset restores the reference inputs, and the data can be downloaded as CSV.

Open this calculation with adjustable inputs

Try the calculation

Time step h (ω = 1) **0.5**

Number of forward steps **20**



Implicit midpoint phase lag 0.2009 rad. Forward-backward distance from the start is 8.951e-16 for implicit midpoint and 0.3635 for explicit midpoint.

Open editable notebook Download .ipynb

Dynamics and numerical time integration · 6. From existence of a trajectory to the order of a time step

Establish what an initial-value solver approximates, derive Runge–Kutta order conditions for a general smooth vector field and construct an alternative high-order step by extrapolation.

Construct the trajectory on a short interval

Let $f(t, y)$ be continuous on a time–state cylinder, bounded there by M and Lipschitz in y with constant L. For initial value y_0, seek a continuous curve in the closed sup-norm ball of radius R around y_0. On an interval of length H, apply

$$(Tz)(t) = y_0 + \int_{t_0}^{t} f(s, z(s))\, ds.$$

If $HM \le R$ and $HL < 1$, T maps the ball into itself and is a contraction. Iteration from the constant curve changes by at most HL times the preceding change. The geometric series of increments converges uniformly and the limit satisfies the integral equation. Differentiation gives the ODE. Two such limits have distance at most HL times their distance, hence coincide. Overlapping short cylinders extend the trajectory until continuation fails. For $y' = y^2,\ y(0) = 1$, the explicit $1/(1-t)$ illustrates why local existence gives no promise at $t = 1$.

The perturbed integral inequality $e(t) \le e_0 + \int_0^t (Le(s) + d(s))ds$ leads, by an integrating factor for the right-hand side, to $e(t) \le e^{Lt}e_0 + \int_0^t e^{L(t-s)}d(s)ds$. It explains why a finite-time global error bound for a numerical solver needs an appropriate domain and a perturbation-growth estimate.

Collect elementary derivatives without losing branching terms

For an autonomous smooth f, the first three elementary differentials are $f,\ f'f,\ f''(f,f)$, together with the third-order chain $f'(f'f)$. Represent a derivative acting on child differentials by a node with one branch per argument. A single node has order and density one. Recursively, a tree $\tau = [\tau_1, \dots, \tau_k]$ has $|\tau| = 1 + \sum |\tau_j|$ and density $\gamma(\tau) = |\tau| \prod \gamma(\tau_j)$. This factor follows by integrating the power of time once when a root is added. Multiplicities of identical branches give the same symmetry factor on the exact and numerical sides.

Define stage weights $\Phi_i(\bullet) = 1$. Taylor expanding each RK stage about the initial state replaces an integral for every child by a weighted stage sum. Consequently

$$\Phi_i([\tau_1, \dots, \tau_k]) = \prod_{\ell=1}^{k} \sum_j a_{ij} \Phi_j(\tau_\ell),$$

$$\sum_i b_i \Phi_i(\tau) = 1/\gamma(\tau).$$

Matching every tree through order p gives classical order p for general sufficiently smooth vector fields. The local defect is order $p+1$, yielding global order p under the perturbation bound above. For

the fourth-order tree consisting of a root, a child, and two leaves on that child, the condition is $b^T A c^2 = 1/12$, with $c = A\mathbf{1}$ and the square componentwise. For a root with three leaves it is $b^T c^3 = 1/4$. A scalar linear test equation supplies only chains.

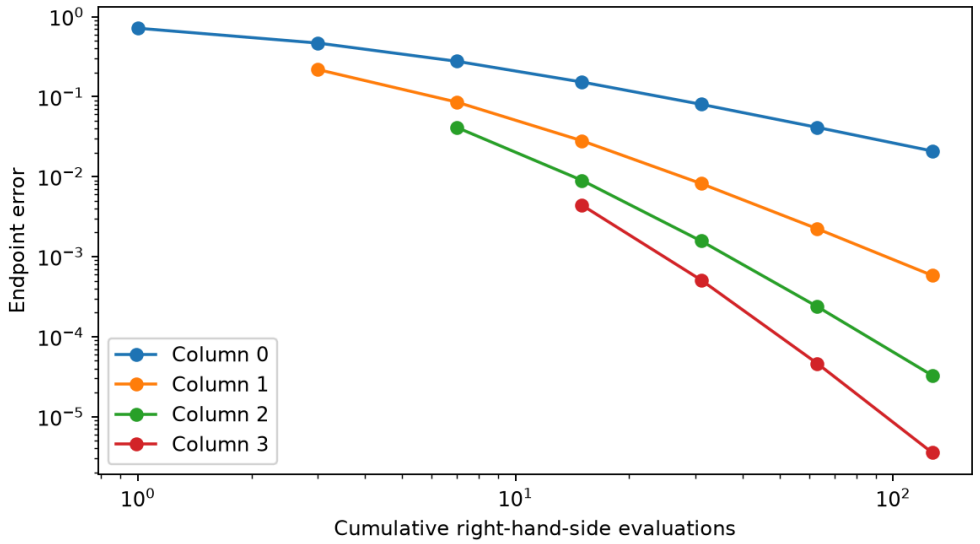
Select accuracy by spending more substeps

For a smooth problem over one macrostep H, let $T_{j,0}$ be its endpoint from $N_j = 2^j$ Euler substeps. At fixed H, assume the endpoint error has an asymptotic expansion in powers of $1/N_j$. This follows for smooth solutions by expanding the local error and propagating its coefficients. The leading term halves on doubling. Eliminate successive powers by

$$T_{j,k} = T_{j,k-1} + \frac{T_{j,k-1} - T_{j-1,k-1}}{2^k - 1}.$$

For $y' = y,\ y(0) = 1,\ H = 0.2$, the one- and two-substep estimates are 1.2 and 1.21, so the extrapolate is 1.22, close to $e^{0.2}$. Four substeps give 1.21550625. Extrapolating once gives 1.2210125 and twice gives 1.22135. Use the separation between successive columns to estimate the lower-order local error. For a column of global order $k + 1$, a local tolerance suggests a trial next H proportional to the tolerance ratio raised to $1/(k + 2)$ if using that column's error estimate. Compare accepted advance per function evaluation at candidate orders. Restart the expansion after a discontinuity and impose the stability restriction of the underlying method.

> **When can a correct extrapolation tableau be misleading?**
>
> An unresolved event or a nonsmooth forcing invalidates the common power expansion across grids. Agreement among columns can then mean that every grid missed the same feature. Split at a known event and test the reconstruction against the defect in the integral equation.



Each column cancels another inverse power of the substep count. Work includes all Euler subintegrations computed through the current row. The reference at time one is exp(1).

Tutorial on existence and order

Bound the Picard iteration for a linear cooling model

▼ **Work through the solution**

With $f = -2y + 1,\ y_0 = 1,\ R = 1, |f| \le 3$ and $L = 2$ on $[0, 2]$. Taking $H = 1/4$ gives $HM = 3/4$ and contraction ratio at most $1/2$. The sum of all remaining differences after an iterate is bounded by the last difference times $(1/2)/(1 - 1/2)$.

Distinguish two fourth-order calculations experimentally

▼ **Work through the solution**

A linear exponential tests the chain $b^T A^2 c$ and is blind to f''. To excite branching terms choose a smooth nonlinear vector field and compare to an independent fine solution at successively halved step sizes. A nonlinear test can reveal an incorrect coefficient, but the general order proof needs every independent tree condition.

Hairer, Nørsett and Wanner · nonstiff problems

Change parameters · From existence of a trajectory to the order of a time step

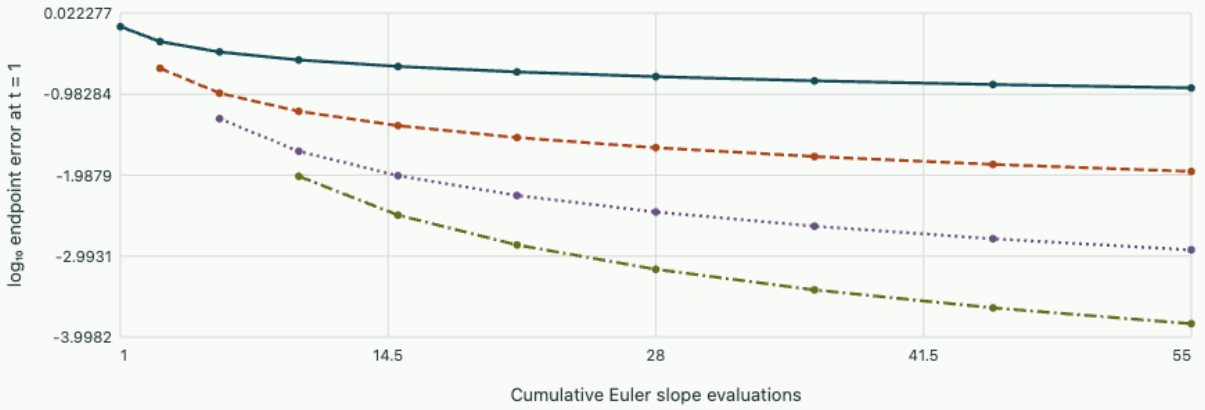
Change one input at a time and compare the recalculated curves with the argument above. The browser evaluates this small model locally; the accompanying numerical values state its assumptions and reference checks. Reset restores the reference inputs, and the data can be downloaded as CSV.

Open this calculation with adjustable inputs

Try the calculation

Growth rate λ **1**

Extrapolation columns **4**



The exact endpoint is exp(λ) = 2.7183. Each Euler subdivision has an expansion in 1/n; extrapolation cancels successive terms while reusing the computed table.

Dynamics and numerical time integration · 7. Why consistent multistep formulas can fail under refinement

Derive the root condition for constant steps, propagate a local defect to a fixed end time and isolate a growing parasitic mode with a controlled starting error.

Turn the recurrence into a propagation matrix

Write a k-step scheme as $\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}$ with $\alpha_k = 1$. Let $\rho(z) = \sum \alpha_j z^j$ and $\sigma(z) = \sum \beta_j z^j$. Substituting exact constant and linear solutions shows consistency requires $\rho(1) = 0,\ \rho'(1) = \sigma(1)$. For continuous differentiable exact solutions these conditions make the defect divided by h tend to zero. Higher derivatives determine the local order.

For $f = 0$ stack the latest k errors. Their propagation matrix is the companion matrix C of ρ. A root r of multiplicity s contributes modes $n^{\ell} r^{n},\ 0 \le \ell < s$ for nonzero roots. All powers of C stay bounded exactly when all characteristic roots lie in the closed unit disk and unit-modulus roots are simple. Interior repeated roots are harmless because exponential decay dominates their polynomial factors. This is null stability for constant step sequences.

Sum the defects over a fixed time

Let a convergent starter have maximum error E_0, the local discrete defect be $h\,\tau_n$, and f be Lipschitz with constant L on the solutions' domain. Repeated substitution using the bounded C^j first propagates the starter and each past perturbation. The Lipschitz terms contribute a sum of errors. For an implicit step take h sufficiently small to absorb its newest error term. With constants independent of h,

$$E_n \le C_1(E_0 + T \max|\tau_j|) + C_2 h \sum_{i<n} E_i \le C_1(E_0 + T \max|\tau_j|)\exp(C_2 T).$$

Thus consistency and null stability imply convergence. Conversely, convergence for arbitrary convergent starting values on smooth problems forces bounded homogeneous propagation. If a root lies outside the disk, a vanishing initial perturbation can grow exponentially over T/h steps. For a repeated unit root choose an initial perturbation proportional to h^{s-1} in the generalised mode so that its final error stays finite and nonzero. Consistency is forced by testing constants and linear-in-time solutions under convergence for all equations, taking an irreducible multistep formula and the usual normalisation. This equivalence concerns fixed-order, constant-step schemes. Changing step ratios changes the propagation matrices.

Control the perturbation before changing the grid

For $f = 0$, take $\rho = (z-1)(z-r)$, with $r = -1$ or $r = 2$. Choose $\sigma = 1 - r$ to make the associated formulas consistent. Initialise $y_0 = 1, y_1 = 1 + h^3$. The exact propagated endpoint error at $N = 1/h$ is

$$e_N = h^3(1 - r^N)/(1 - r).$$

For $r = -1$ every error stays $O(h^3)$. For $r = 2$ the growing factor eventually defeats every algebraic power. Exact starting data with no perturbations would hide this instability in the constant solution.

Transfer the experiment to the decaying state $y' = -y$ by choosing $\sigma(z) = (1 - r)z$. Now the middle coefficient in the recurrence is $1 + r - h(1 - r)$. A OneSimulation RK4 trajectory supplies a common starter and a separate full-step trajectory checked against e^{-1}. Inject h^3 into the starter and compare with the identical multistep solve without that injection. The difference isolates amplification along the recurrence.

OneSimulation experiment · Watch a vanishing starting error grow

Vary the number of steps and the injected starting error. Compare both perturbation amplification and the error of the reference trajectory.

Download the executable notebook · Python source

Adjust the experiment in the browser. Vary the parasitic root, the step count and the injected starting error independently. The first step is the RK4 polynomial for $y' = -y$; subsequent steps use the two-step recurrence above. The browser follows the perturbation in its own recurrence so that subtractive cancellation does not hide a tiny change. Compare $r = -1, 0, 1, 2$. At $r = 1$, the repeated root and vanishing differential term make this a degenerate, unstable formula. Hold the perturbation fixed to inspect amplification; then lower it by a factor of eight when doubling the number of steps to reproduce an h^3 perturbation. The notebook uses OneSimulation RK4 for its starter and reference trajectory.

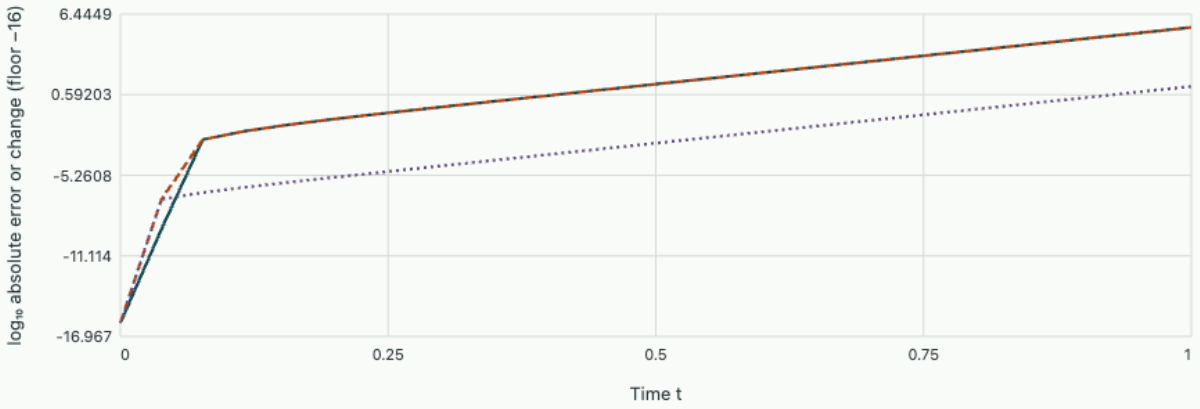
Open the live experiment

Try the calculation

Parasitic root r **2**

Steps to t = 1 **26**

$\log_{10}$ starting perturbation **-7**



Unperturbed discretisation error

Error with perturbed start

Propagated starting change

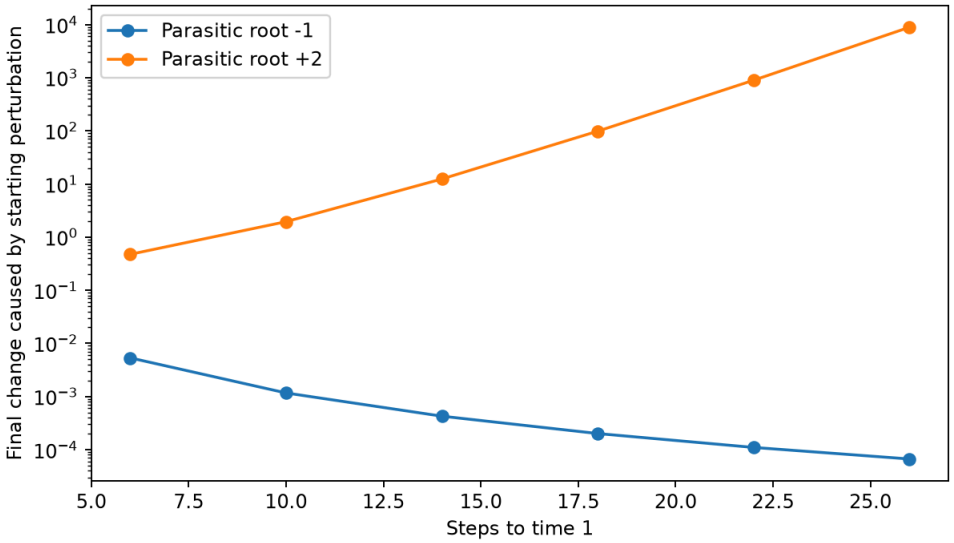
The characteristic roots at h = 0 are 1 and 2. The parasitic root lies outside the unit circle. A starting change of 1e-7 produces 15.6 change at t = 1. The error-change recurrence is evaluated separately to avoid subtracting two nearly equal trajectories.

```
"""Resolve a decaying signal with two consistent families of two-step methods."""
import matplotlib.pyplot as plt
import numpy as np
from onesim.numerics.time import ClassicalRungeKutta4TimeIntegrator

def decay(state, time):
    return (-state[0],)

def endpoint(parasitic, starter, h, n):
    previous, current = 1.0, starter
    for _ in range(1, n):
        previous, current = current, (1+parasitic-h*(1-parasitic))*current-parasitic*previous
    return current

method = ClassicalRungeKutta4TimeIntegrator()
counts = [6, 10, 14, 18, 22, 26]
fig, ax = plt.subplots(figsize=(7, 4), layout="constrained")
for parasitic in [-1, 2]:
    errors = []
    for n in counts:
        h = 1/n
        reference = (1.0,)
        starter = method.advance(reference, decay, h, time=0.)[0]
        for j in range(n):
            reference = method.advance(reference, decay, h, time=j*h)
        unperturbed = endpoint(parasitic, starter, h, n)
        perturbed = endpoint(parasitic, starter+h**3, h, n)
        errors.append(abs(perturbed-unperturbed))
```

```
        assert abs(reference[0] - np.exp(-1)) < 1e-5
    ax.semilogy(counts, errors, "o-", label=f"Parasitic root {parasitic:+d}")
    print(f"Root {parasitic:+d}: change at T=1 with 26 steps: {errors[-1]:.6g}")
print(f"Reference RK4 error with 26 steps: {abs(reference[0]-np.exp(-1)):.3g}")
ax.set(xlabel="Steps to time 1", ylabel="Final change caused by starting perturbation")
ax.legend()
fig.savefig("numerics-parasitic-root.png", dpi=170)
```

Output from the reference run

```
Root -1: change at T=1 with 26 steps: 6.6793e-05
Root +2: change at T=1 with 26 steps: 8875.58
Reference RK4 error with 26 steps: 6.93e-09
```



Terminal errors for two recurrences with distinct parasitic roots.

Tutorial on multiplicity and error order

Can two roots at one be tolerated if every local defect is tiny?

▼ Work through the solution

For $\rho = (z-1)^2$ and $f = 0$, choose $e_0 = 0, e_1 = h$. Differences stay equal to h, so $e_N = Nh = T$. The local residual is exactly zero while the starter converges and the endpoint fails to converge.

Separate the order of the starter from that of the method

▼ Work through the solution

A null-stable order-three formula has $\tau = O(h^3)$. If its starter has only $O(h^2)$ accuracy, the bound at fixed T becomes $O(h^2)$. Raising the order of the multistep formula alone cannot recover the lost order.

Hairer, Nørsett and Wanner · multistep methods

Dynamics and numerical time integration · 8. Why an accurate position can hide an inaccurate constraint force

Derive an energy test for implicit Runge–Kutta stages, then recover missing derivatives in a constrained model and calculate the accuracy lost by a multiplier.

Derive a nonlinear contractivity test

Suppose $\langle u - v, f(u) - f(v) \rangle \le 0$ for all relevant states. For two trajectories advanced by the same RK step, use D_i for each stage-state difference and g_i for the stage-slope differences. Substitute the stage identity $D_i = d_0 + h \sum_j a_{ij} g_j$ into the square of $d_1 = d_0 + h \sum_i b_i g_i$. Collecting symmetric products gives

$$\|d_1\|^2 - \|d_0\|^2 = 2h \sum_i b_i \langle D_i, g_i \rangle - h^2 \sum_{ij} (b_i a_{ij} + b_j a_{ji} - b_i b_j) \langle g_i, g_j \rangle.$$

If $b_i \ge 0$ and the displayed symmetric coefficient matrix is positive semidefinite, the step cannot increase distances, assuming its nonlinear stage equations have a solution. This sufficient algebraic stability condition applies to nonlinear dissipative fields. For one-stage midpoint $a = 1/2, b = 1$ the matrix is zero. Backward Euler $a = b = 1$ gives one. Both are contractive, but their scalar amplification limits as $h\lambda \to -\infty$ are -1 and 0 respectively. Only the latter damps an unresolved fast mode completely.

Recover the derivatives needed to see the multiplier

For index one, $y' = f(y, z),\ 0 = g(y, z)$ with g_z invertible, the implicit function theorem eliminates z. Perturbing the algebraic constraint at fixed y by ϵ gives $\delta z = O(\epsilon / \sigma_{\min}(g_z))$. A mechanical position constraint has a different structure:

$$\begin{aligned} q' &= v, \\ M v' &= F(q, v) - G(q)^T \lambda, \\ g(q) &= 0, \\ G &= g_q. \end{aligned}$$

Differentiating once gives $Gv = 0$. Differentiating again gives $Gv' + g_{qq}[v, v] = 0$. For positive definite mass and independent constraints the multiplier matrix $G M^{-1} G^T$ is invertible and

$$(G M^{-1} G^T) \lambda = G M^{-1} F + g_{qq}[v, v].$$

The position formulation is a differentiation-index-three system in first-order form. Consistent initial data require both position and velocity constraints. A small position residual alone places no bound on multiplier error without regularity of its time dependence.

Derive a concrete order loss before interpreting an index

A manufactured constraint $q = \sin t$ with $q' = v,\ v' = \lambda$ has $v = \cos t, \lambda = -\sin t$. Suppose a position solver introduces an alternating error $e_n = (-1)^n h^p$. A backward difference for velocity amplifies it to $2(-1)^n h^{p-1}$, and a second difference for the multiplier amplifies it to $4(-1)^n h^{p-2}$. A smooth error series instead can gain cancellations. This distinction prevents assigning every index-three method a universal loss of two orders. Consistent stage order, constraint enforcement, regularity, initial conditions and the reconstruction used for the multiplier all affect the theorem.

Consider a converged Radau collocation step for the simpler index-two constraint $q' = z,\ q = g(t)$. Its stages enforce the exact sampled $g(t_n + c_i h)$. Starting from exact q_n, the derivative stages equal those of the degree-s interpolant through the start and the s stage nodes. Stiff accuracy identifies the final stage as the output. Interpolant differentiation gives endpoint derivative error $O(h^s)$ for smooth g. The constrained position is exact here, illustrating precisely how classical differential-variable order fails to predict a multiplier measurement.

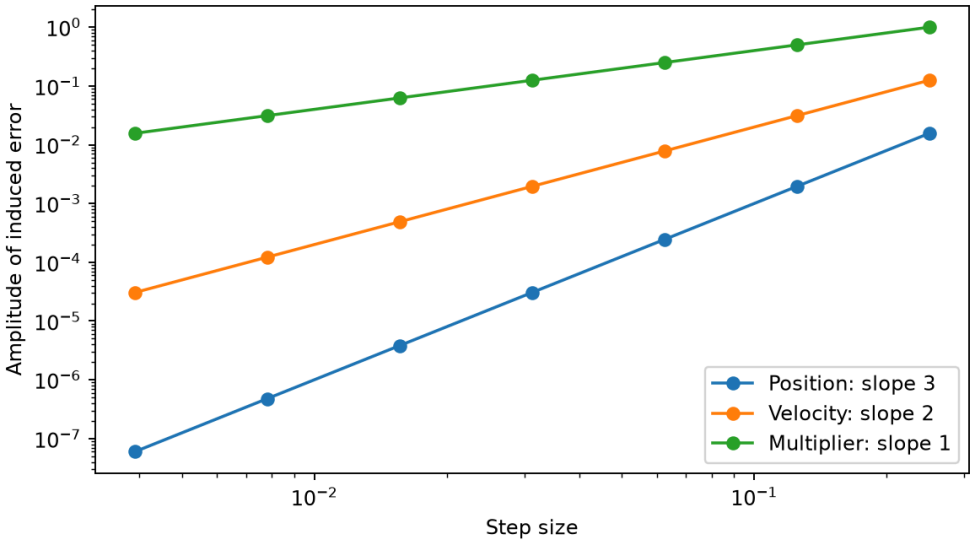
Derive an index-dependent rate in the constant-coefficient case

For a regular linear differential-algebraic pencil with constant coefficients, the algebraic part can be written as $Nz' = z - g(t)$, where $N^\nu = 0$ and $N^{\nu-1} \neq 0$. Repeated substitution gives $z = g + Ng' + \cdots + N^{\nu-1} g^{(\nu-1)}$. For a single Jordan chain, its last component specifies the input directly while the first can depend on its derivative of order $\nu - 1$. The highest derivative actually seen depends on the input direction and the reported component.

Interpolate g through $s + 1$ distinct nodes at fixed relative positions on an interval of length h. Taylor expansion at the endpoint through degree s is exact for the interpolant. The residual nodal values are $O(h^{s+1})$. Differentiating the fixed reference cardinal polynomials ℓ times introduces $h^{-\ell}$, giving $O(h^{s+1-\ell})$ for derivative order $\ell \leq s$. Smoothness through degree $s + 1$ is required. If nodal observations themselves carry $O(h^p)$ disturbances, their differentiated contribution is $O(h^{p-\ell})$. The worst reconstructed component on an index-ν chain therefore has order $\min(s + 1, p) - (\nu - 1)$, when $\nu - 1 \leq s$ and the forcing couples to that last derivative. This is an error bound for this specified reconstruction. A different integrator's stage order and enforcement of the constraints need a separate analysis.

What does a nonlinear stage tolerance control?

A stage solve tolerance on position has the units of position. When converting residuals to a velocity or acceleration tolerance, budget the factors $1/h$ and $1/h^2$ and the conditioning of the constraint matrix. Plot force error on a manufactured trajectory as well as the position-constraint residual.



Differencing an alternating position disturbance of amplitude h³ yields 2h² for velocity and 4h for the multiplier. These are the contributions from the disturbance alone. Smooth truncation errors need not follow these slopes.

Tutorial on contractivity and constraint data

Check the algebraic stability test on forward Euler

▼ **Work through the solution**

With $a = 0, b = 1$, the matrix coefficient is -1. For $y' = -y$, the amplification $1 - h$ has magnitude above one for $h > 2$, confirming that increasing the step may violate contractivity.

What force holds a unit mass on the unit circle at speed three?

▼ **Work through the solution**

Choose $g = (q^T q - 1)/2$, so $G = q^T$, $g_{qq} = I$, and $F = 0$. With $\|q\| = 1,\ q^T v = 0$, the multiplier is $\|v\|^2 = 9$ and the constraint force is $-9q$. An outward initial velocity violates the hidden constraint even if the position is exact.

Hairer and Wanner · stiff and differential-algebraic equations

Change parameters · Why an accurate position can hide an inaccurate constraint force

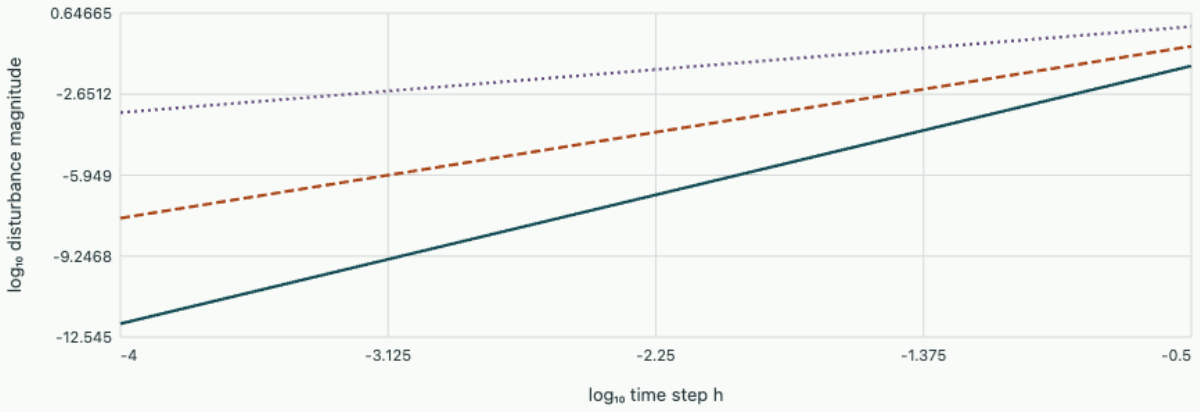
Change one input at a time and compare the recalculated curves with the argument above. The browser evaluates this small model locally; the accompanying numerical values state its assumptions and reference checks. Reset restores the reference inputs, and the data can be downloaded as CSV.

Open this calculation with adjustable inputs

Try the calculation

Position disturbance order p **3**

Disturbance coefficient **1**



An alternating disturbance loses one power of h under each discrete derivative. For p = 3, the induced multiplier contribution scales as h^1.