Stolen with Pride

How Control Theory brings out the best in Numerical Analysis

Gustaf Söderlind

Numerical Analysis

Lund University

Lund, October 9, 2009

Contents

1. The problem

Time-stepping methods for initial value problem $\dot{y} = f(t, y)$

Given an approximation $y_n \approx y(t_n)$ compute

$$y_n \mapsto y_{n+1}$$

with time step size $h = t_{n+1} - t_n$

Work/accuracy trade-off

Don't use constant step size – put grid points where they really matter to accuracy

Adaptive methods

Most ODE solvers are *adaptive*

Given an *error tolerance* ε, they try to select the *time step* h_n so as to make each *local error* $r_n = \varepsilon$

Asymptotic step size – error model (as $h \to 0$)

$$r_n = \varphi_n h_n^k$$

If φ is constant, then $h_{n+1} = (\varepsilon/r_n)^{1/k} h_n$ will make $r_{n+1} \equiv \varepsilon$

Adaptive methods

Most ODE solvers are *adaptive*

Given an *error tolerance* ε, they try to select the *time step* h_n so as to make each *local error* $r_n = \varepsilon$

Asymptotic step size – error model (as $h \to 0$)

$$r_n = \varphi_n h_n^k$$

If φ is constant, then $h_{n+1} = (\varepsilon / r_n)^{1/k} h_n$ will make $r_{n+1} \equiv \varepsilon$

. . . and numerical analysis stopped thinking, right there. . . !

2. Control and signal processing

Classical time-step control

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{1/k} h_n$$

In logarithmic form

$$\log h_{n+1} - \log h_n = -\frac{1}{k}\left(\log r_n - \log \varepsilon\right)$$

This is plain integrating control

Linear difference equation, $\log r \mapsto \log h$

Linear control and signal processing

Signal processing

How to map observed error sequence $\log r$ to suitable step size sequence $\log h$ while keeping $r \approx \varepsilon$

General technique

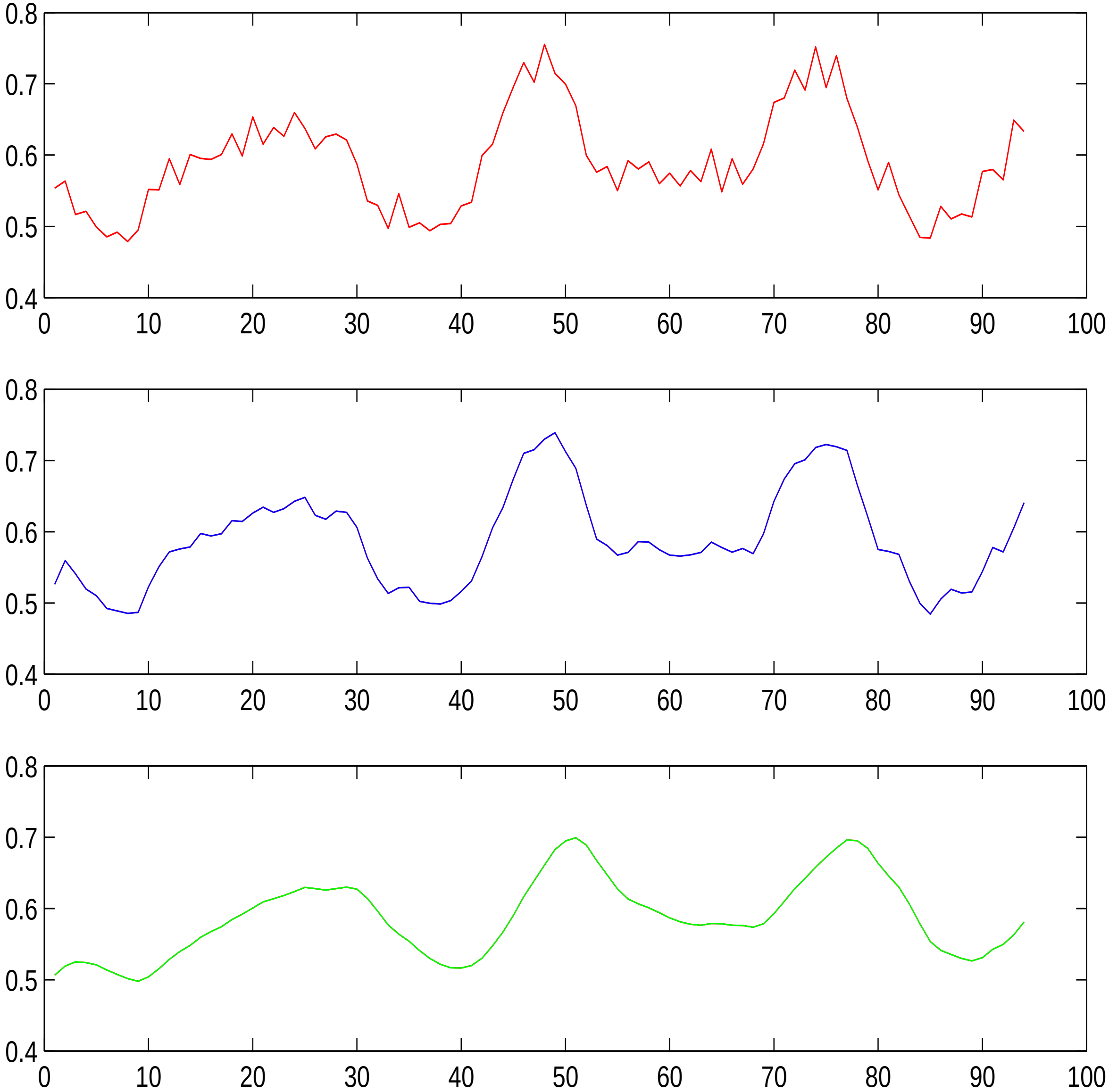
$$(q-1)Q(q)\log h = P(q)(\log r - \log \varepsilon)$$

P, Q polynomial operators in forward shift operator q

New possibilities

Q “autoregressive” part; P “moving average” part

What step size properties can be achieved?



Elementary control

FIR filter

Noise shaping filter

Will it have an impact on computations?

Standard, elementary time-step control

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{1/k} h_n$$

in logarithmic form is a *negative feedback control law*

$$\log \frac{h_{n+1}}{h_n} = -\frac{1}{k} \log \frac{r_n}{\varepsilon}$$

Actual implementations add safety nets and heuristics

Will it have an impact on computations?

Standard, elementary time-step control

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{1/k} h_n$$

in logarithmic form is a *negative feedback control law*

$$\log \frac{h_{n+1}}{h_n} = -\frac{1}{k} \log \frac{r_n}{\varepsilon}$$

Actual implementations add safety nets and heuristics

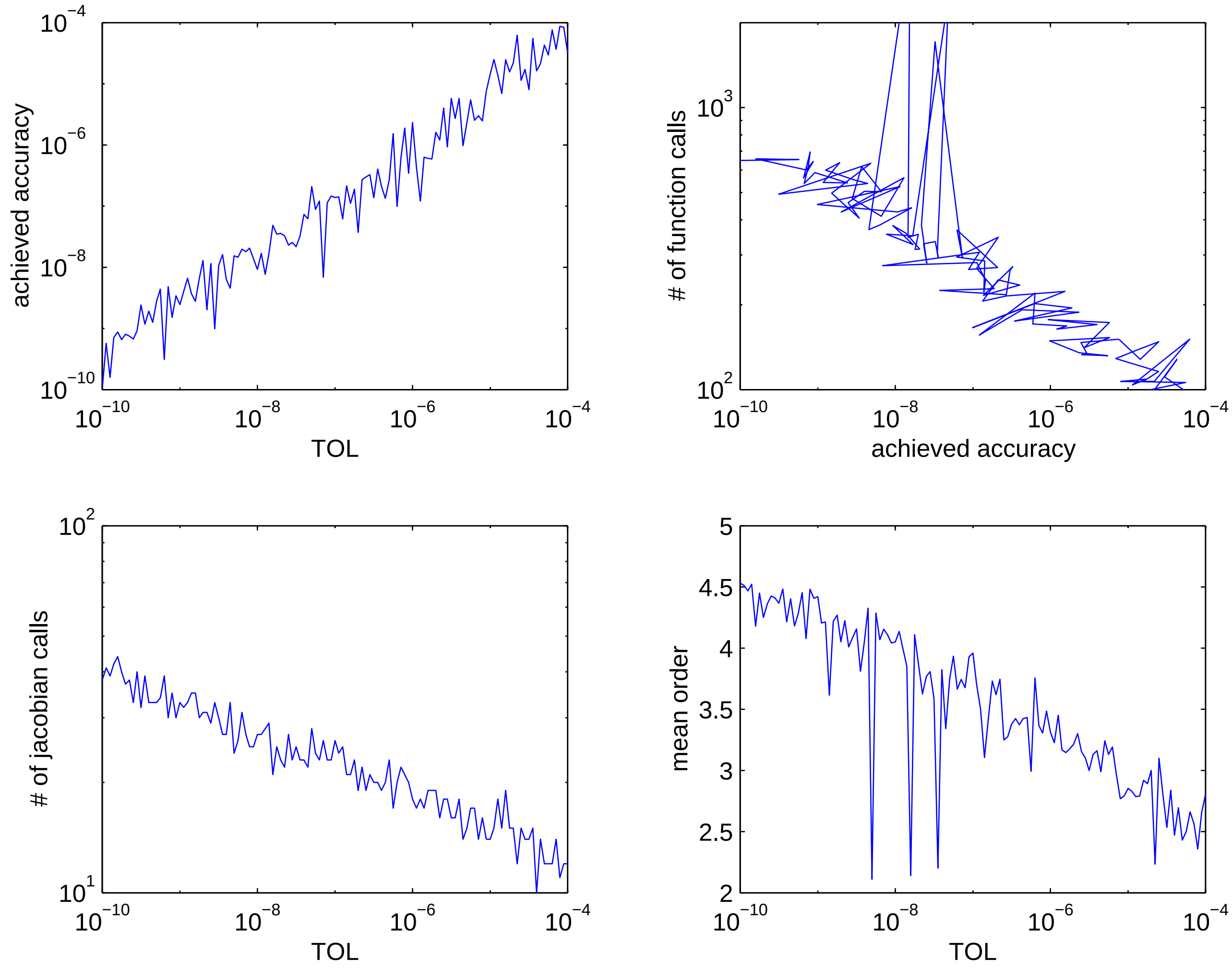
. . . so numerical analysis didn't quite stop thinking, right there!

Example of actual implementation with heuristics

Typical plot of $\log(h_{n+1}/h_n)$ *vs.* $\log(r_n/\varepsilon)$

Nonlinear, discontinuous and nonsymmetric!

How well does it work? *Chemakzo* problem



Poor computational stability

Small changes in ε have large effects on output

What is computational stability?

Continuous data dependence

$$c \cdot \varepsilon \leq \|e\| \leq C \cdot \varepsilon$$

with $\log_{10}(C/c) \ll 1$

What is computational stability?

Continuous data dependence

$$c \cdot \varepsilon \leq \|e\| \leq C \cdot \varepsilon$$

with $\log_{10}(C/c) \ll 1$

How can it be improved?

- *Digital filtering of error estimates*
- *Control theory for time–step selection*
- *Order selection controller*
- *Appropriate Newton iteration termination*

What is computational stability?

Continuous data dependence

$$c \cdot \varepsilon \leq \|e\| \leq C \cdot \varepsilon$$

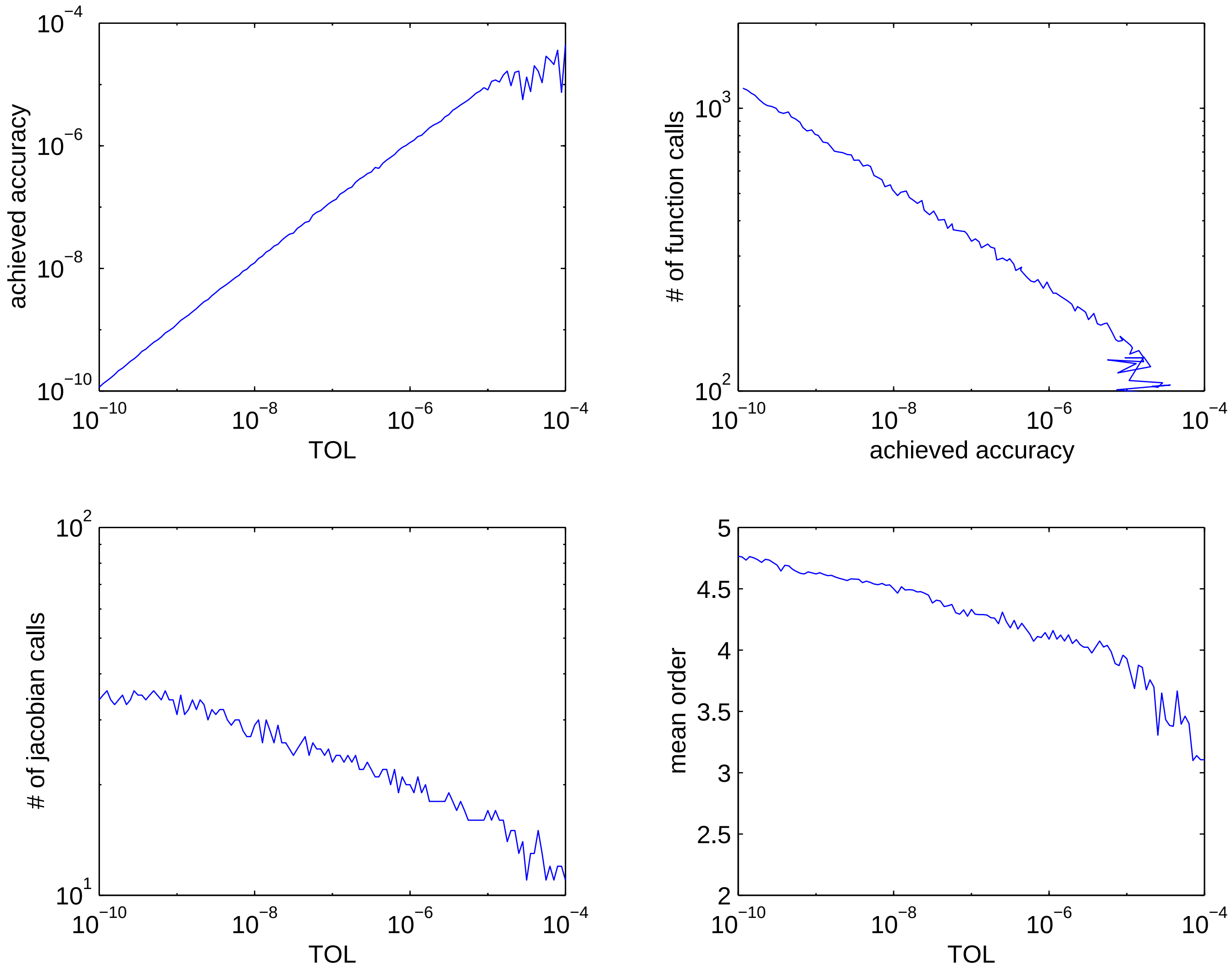
with $\log_{10}(C/c) \ll 1$

How can it be improved?

- *Digital filtering of error estimates*
- *Control theory for time–step selection*
- *Order selection controller*
- *Appropriate Newton iteration termination*

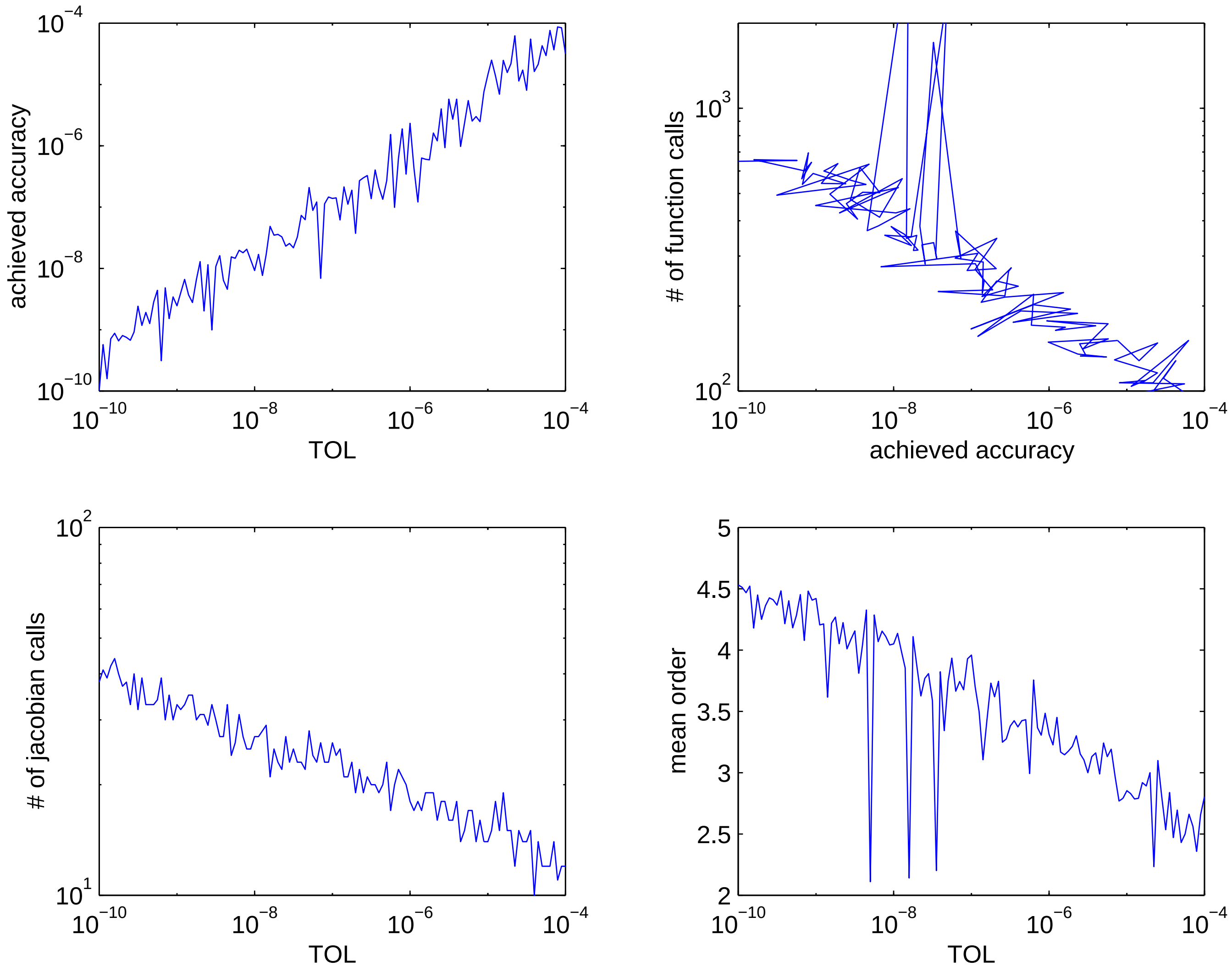
$\Rightarrow \log_{10}(C/c) \approx 0.05$ possible *at no extra cost!*

. . . and it works a lot better! *Chemakzo* problem



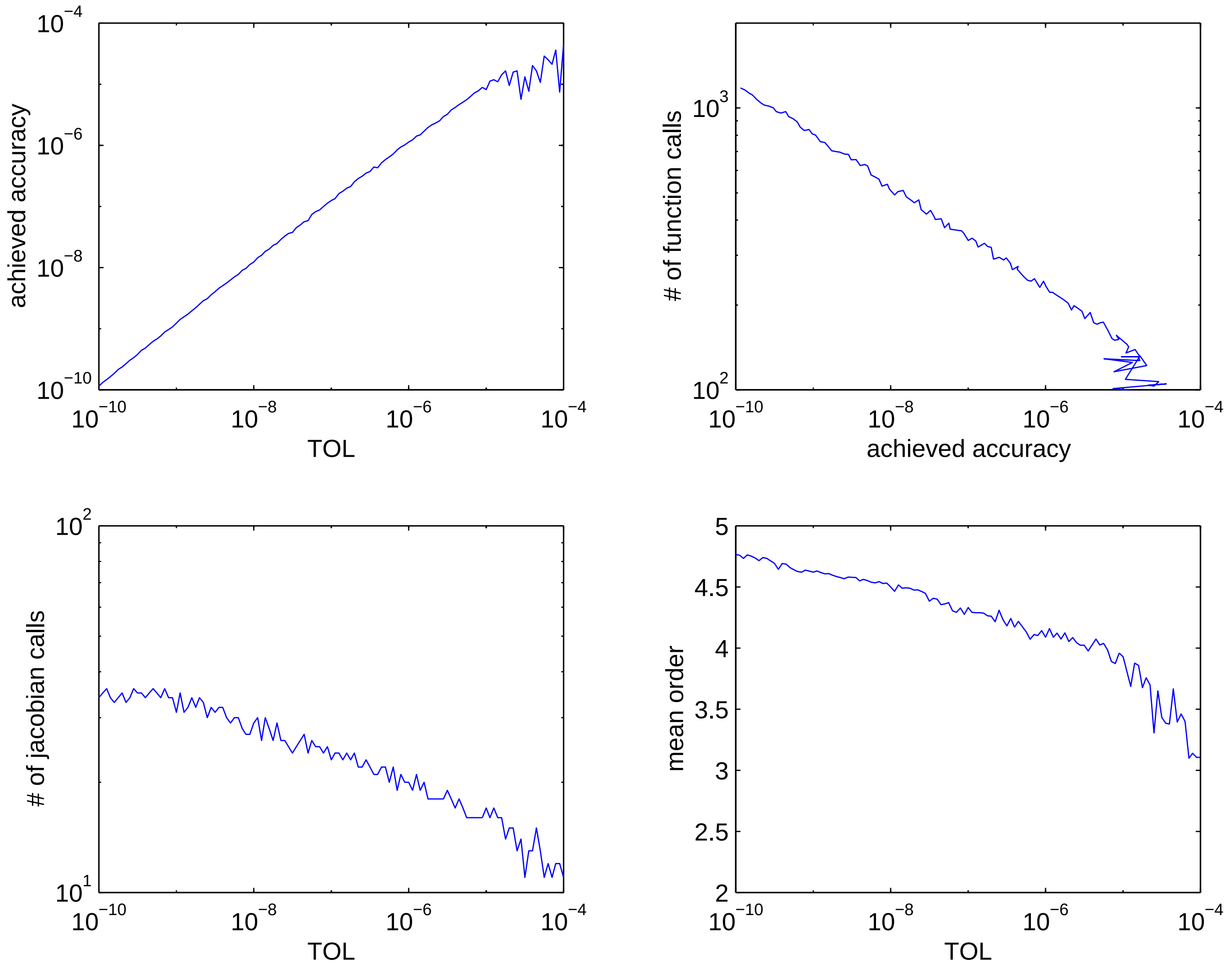
High computational stability

Compare to the standard implementation!



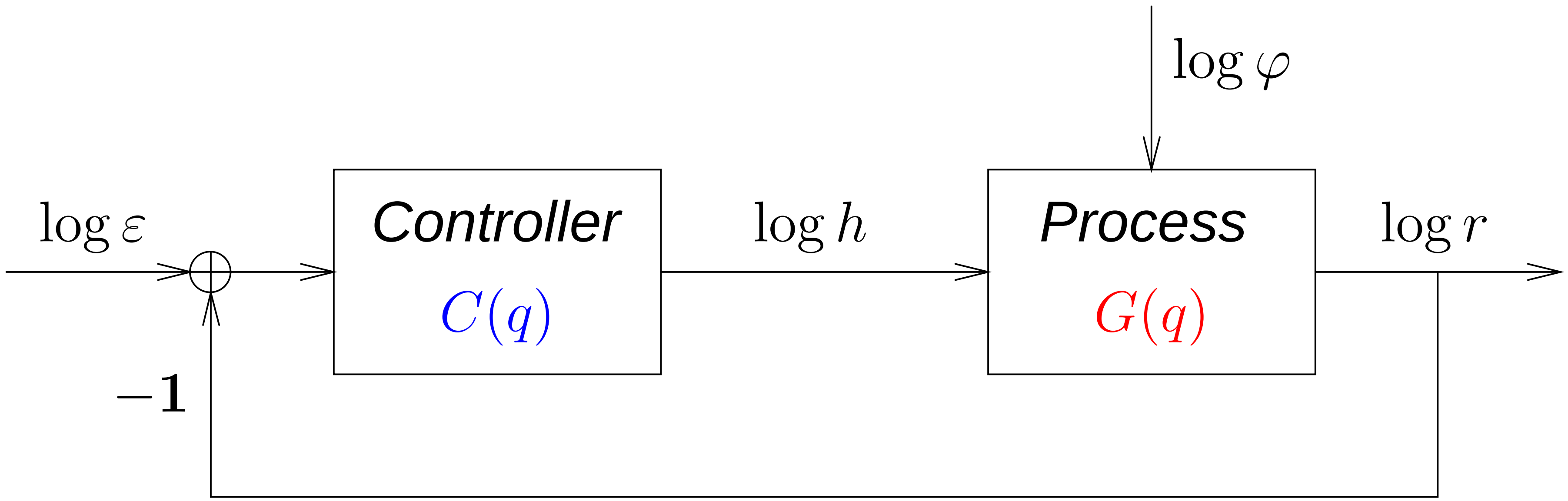
Poor computational stability

Revised implementation, *Chemakzo* problem



High computational stability

Control theoretic approach



Asymptotic process model

$$r = \varphi h^k \quad \Rightarrow \quad \log r = k \cdot \log h + \log \varphi\,; \qquad (G(q) = k)$$

Control law

$$\log h = C(q) \cdot (\log \varepsilon - \log r)$$

Closed loop response

Step size response $H_\varphi(q) : \log\varphi \mapsto \log h$

$$-kH_\varphi(q) = \frac{k \cdot C(q)}{1 + k \cdot C(q)}$$

Error response $R_\varphi(q) : \log\varphi \mapsto \log r$

$$R_\varphi(q) = \frac{1}{1 + k \cdot C(q)}$$

Control design *Low-pass* filter for h and *high-pass* filter for r

Step size response for $C(q) = P(q)/[(q-1)Q(q)]$

Linear recursion $\big((q-1)Q(q) + kP(q)\big)\log h = -P(q)\log\varphi$

Elementary control	$Q \equiv 1; \quad P \equiv 1/k$
Convolution filter	$Q \equiv 1; \quad P \equiv \gamma < 1/k$
I control	$Q \equiv 1; \quad \deg P = 0$
PI control	$Q \equiv 1; \quad \deg P = 1$
PID control	$Q \equiv 1; \quad \deg P = 2$
FIR filter	$(q-1)Q(q) + k \cdot P(q) = q^m$
Autoregressive (AR)	Q has zero(s) at $q = 1$
Moving average (MA)	P has zero(s) at $q = -1$

A digital filter for step size control

Step size low-pass filter $H211b$:

$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{1/(bk)} \left(\frac{\varepsilon}{r_{n-1}}\right)^{1/(bk)} \left(\frac{h_n}{h_{n-1}}\right)^{-1/b} h_n$$

The filter coefficients are determined by order conditions

A digital filter for step size control

Step size low-pass filter $H211b$:

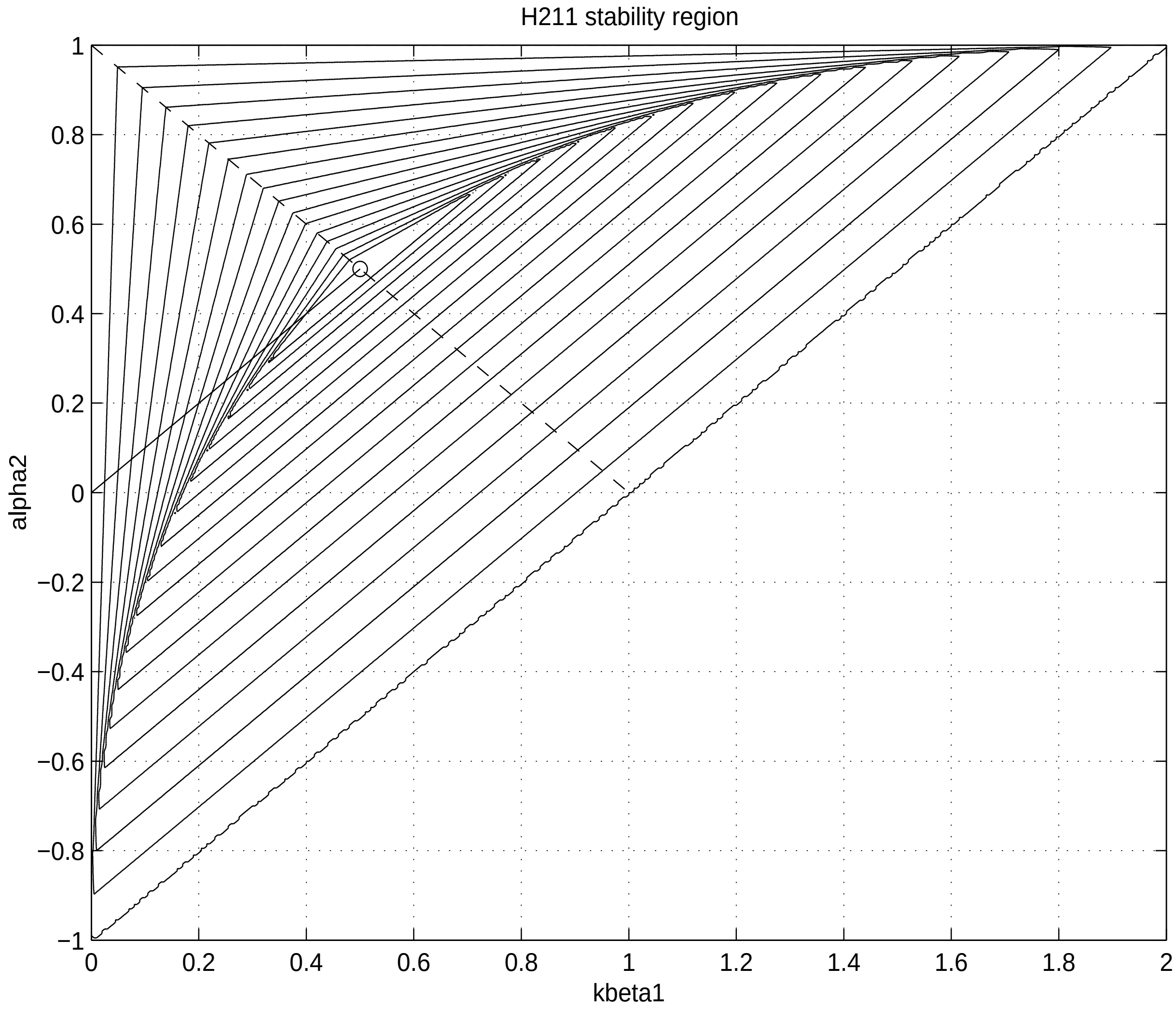
$$h_{n+1} = \left(\frac{\varepsilon}{r_n}\right)^{1/(bk)} \left(\frac{\varepsilon}{r_{n-1}}\right)^{1/(bk)} \left(\frac{h_n}{h_{n-1}}\right)^{-1/b} h_n$$

The filter coefficients are determined by order conditions

Properties

- Stable for $b \in [1, \infty)$ with poles at $q = 0,\, 1 - 2/b$
- 1st order low-pass FIR filter (deadbeat) at $b = 2$
- Increasing b increases noise suppression

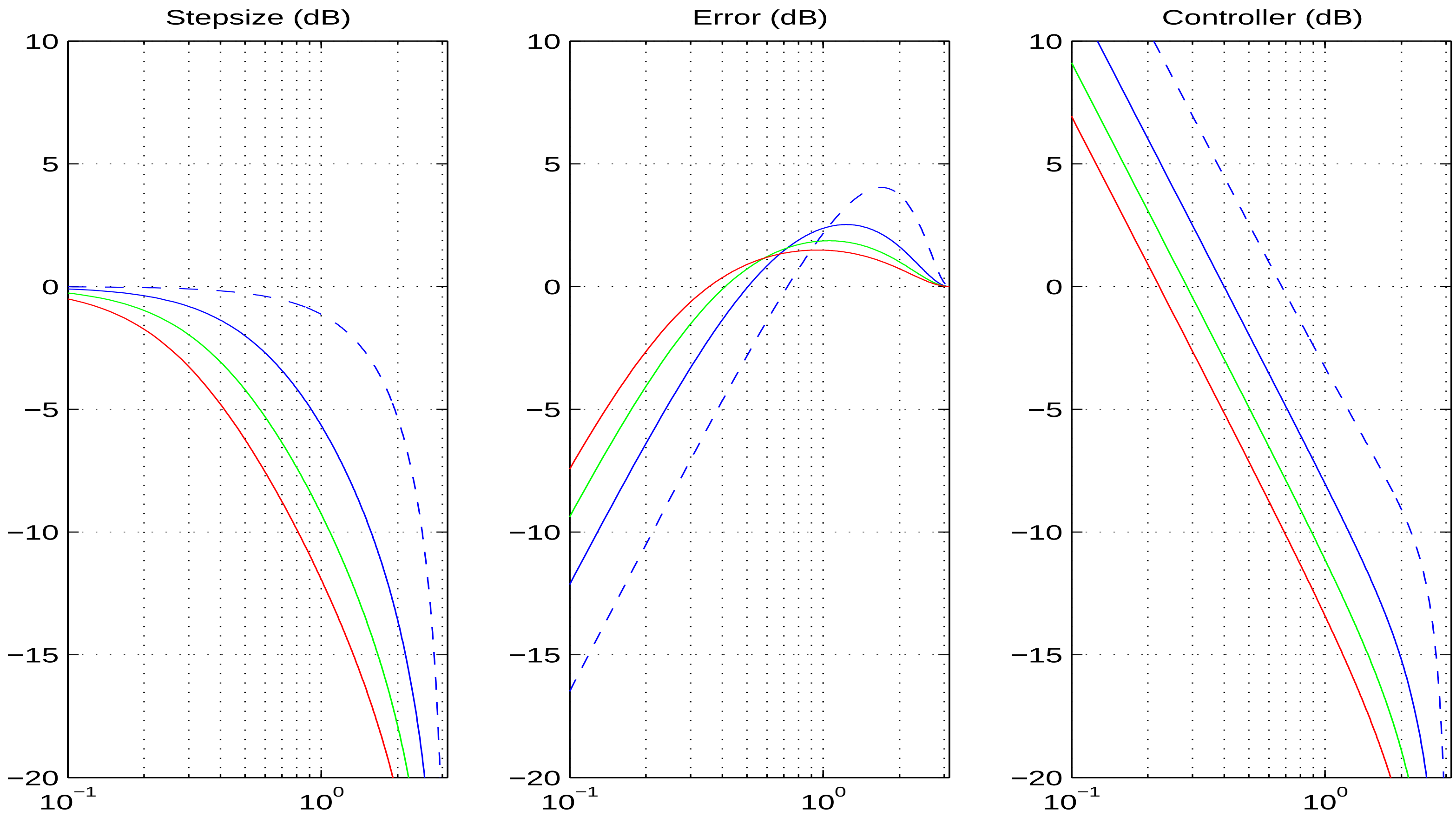
Pole placement



FIR filter at 'o', other $H211b$ on straight line segment

$H211b$ frequency response on $[0.1, \pi]$

$\log \varphi \mapsto k \log h$ $\qquad$ $\log \varphi \mapsto \log r$ $\qquad$ $\log r \mapsto \log h$



FIR filter (dashed), noise shaping: $b = 4$; 6 ; 8

On the boundary ∂S of the stability region

The model $G(q) = kq^{-1}$ is no longer valid

Instead

$$G^{\partial S}(q) = kq^{-1}\left(c_1 - \frac{c_2}{q-1}\right)$$

System identification determines coefficients c_1 and c_2

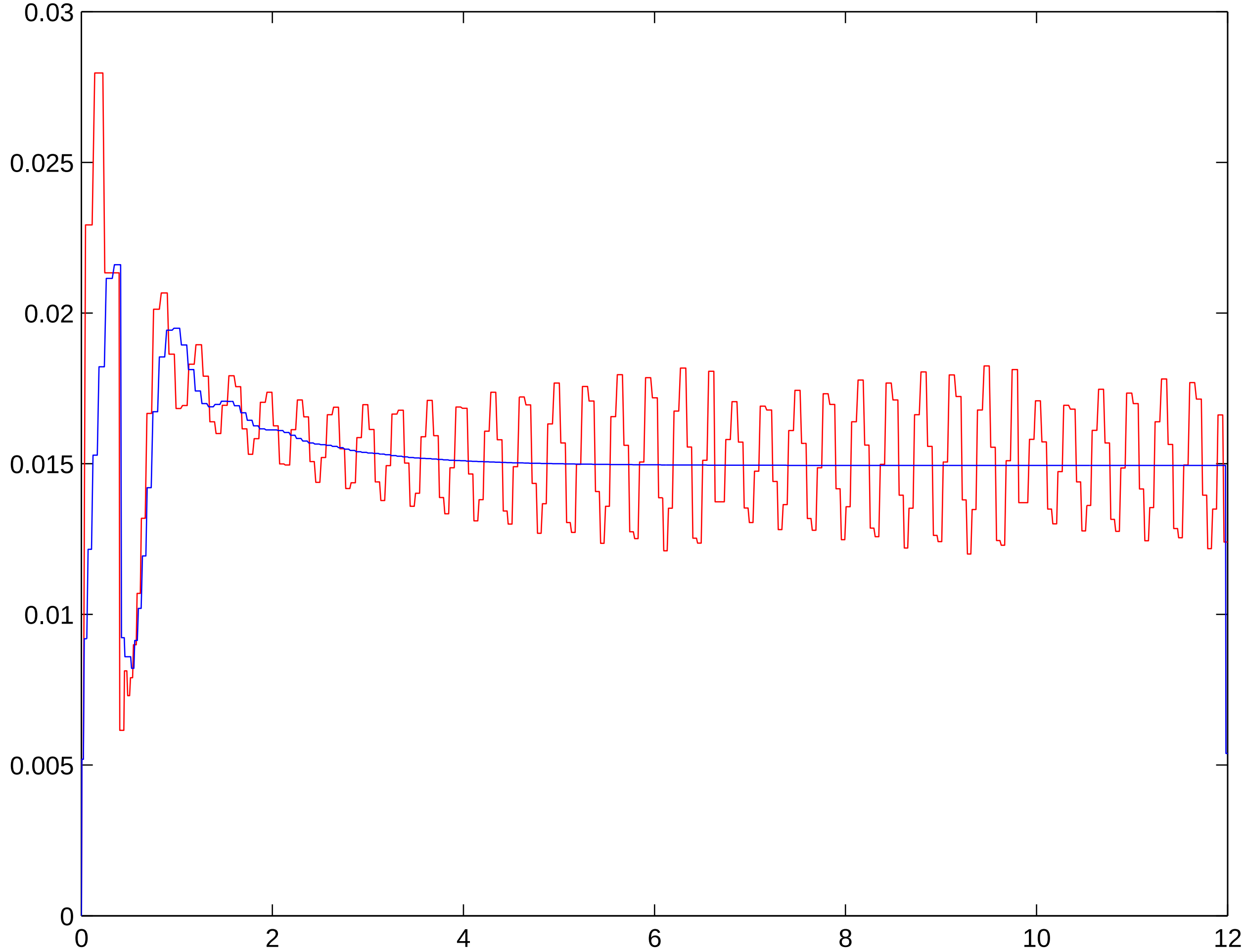
Coefficients in a narrow range for large classes of methods

Find controller that can control both $G(q)$ and $G^{\partial S}(q)$!

Proper PI controllers (positive P gain) solve this problem

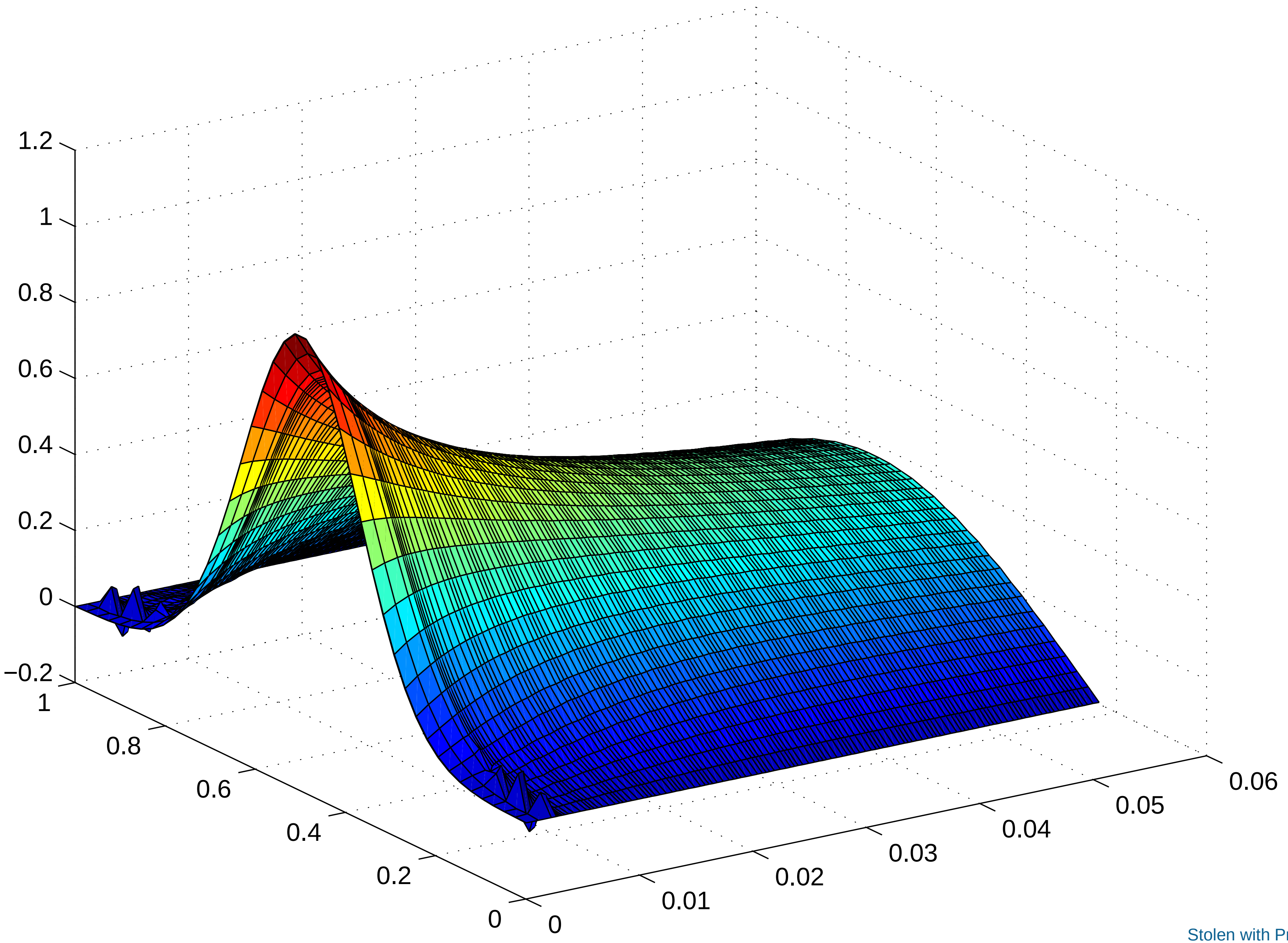
Example: modifying **ode45** in MATLAB

Step size sequences in chemotaxis problem

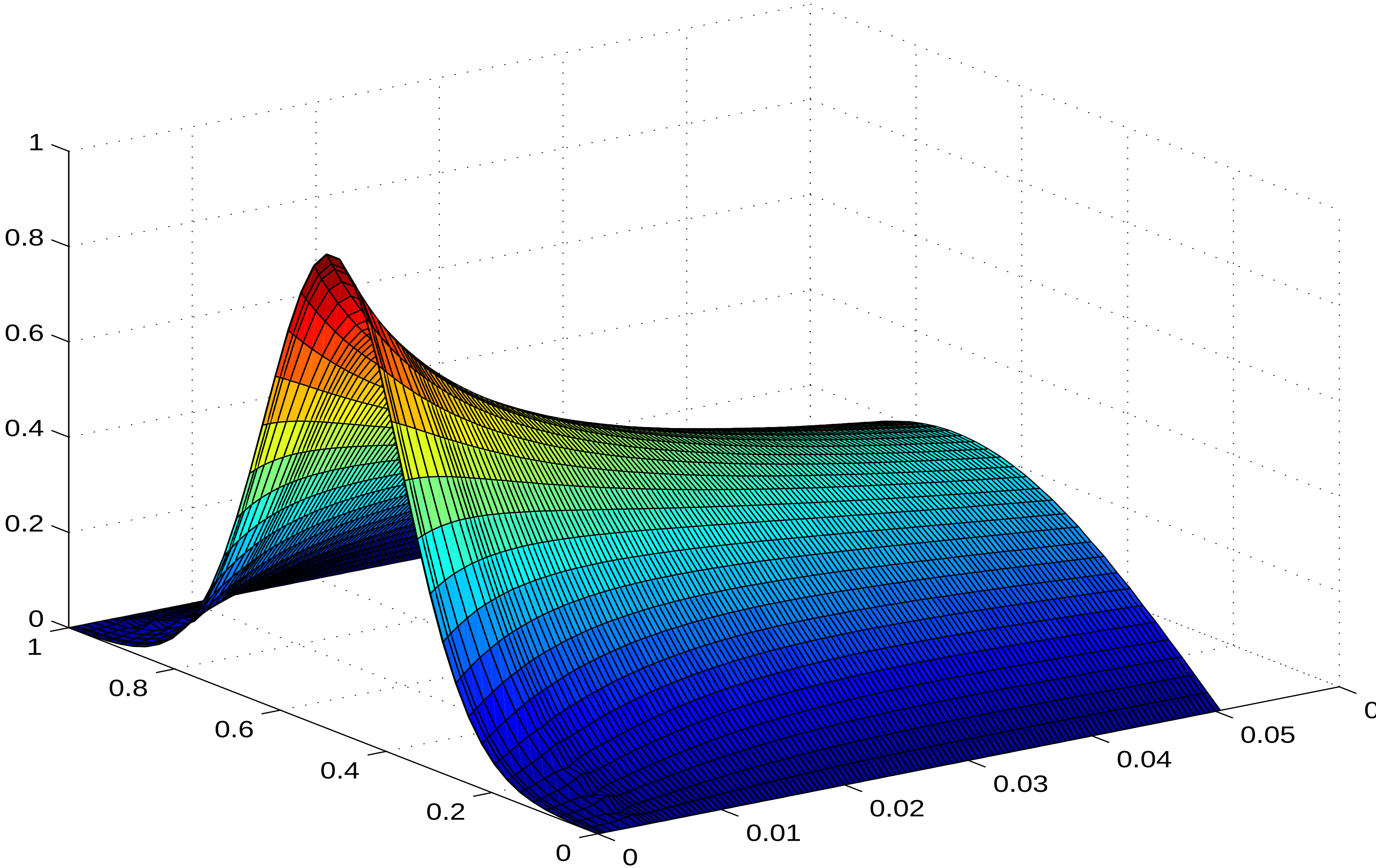


Original vs. modified code (PI controller)

Elementary deadbeat grid in diffusion problem



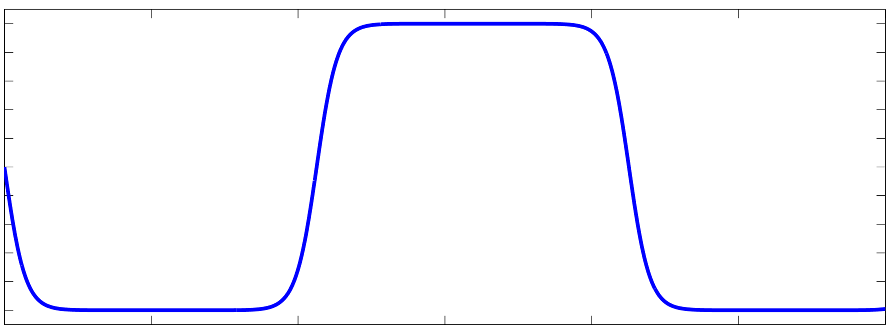
PI controlled grid in diffusion problem



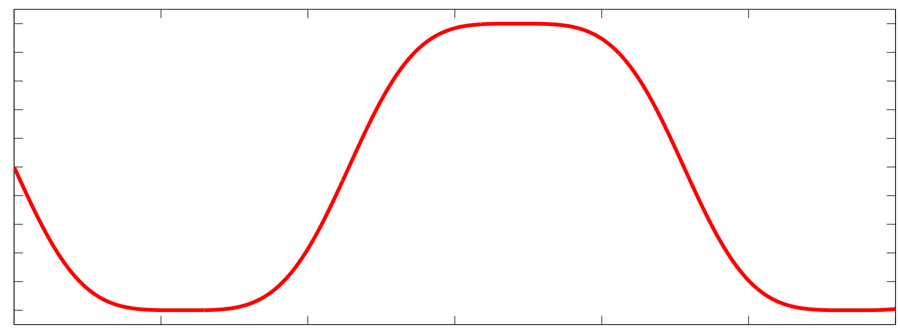
3. Time transformation adaptivity

Substitution $t \quad \longleftrightarrow \quad s$

Original solution



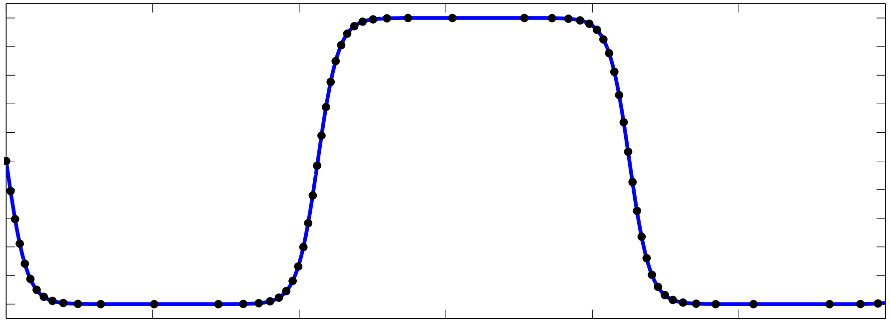
Transformed solution



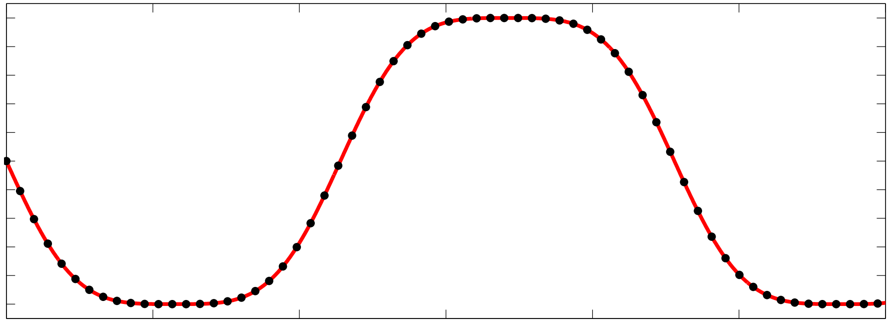
Variable steps via time transformations

Substitution $t \quad \longleftrightarrow \quad s$

Original solution



Transformed solution



Variable steps Δt correspond to constant step size Δs

Useful for "geometric integration" of Hamiltonian systems, problems with invariants, reversible dynamics

Grid density control

Time stretching/compression $t = \Theta(s)$ with derivative

$$\frac{\mathrm{d}\Theta}{\mathrm{d}s} = \Theta'(s) = \frac{1}{\rho(s)} \quad \Rightarrow \quad \mathrm{d}t = \mathrm{d}s/\rho$$

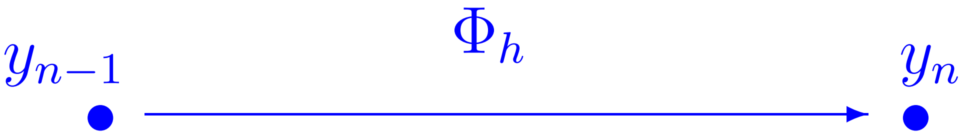
Transformed differential equation $\dot{y} = F(y) \;\Rightarrow\; y' = F(y)/\rho$

Sampling (step size) correspondence

$$\Delta t = h_{n+1/2} = t_{n+1} - t_n = \Theta(s_{n+1}) - \Theta(s_n) \approx \frac{\Delta s}{\rho(s_{n+1/2})}$$

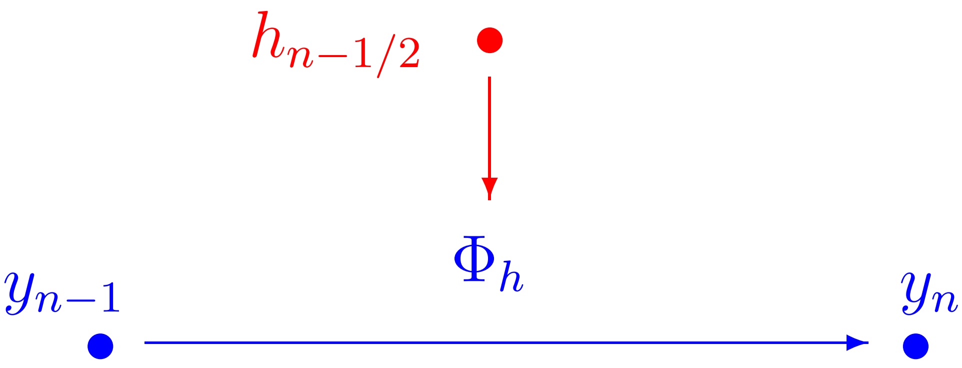
Constant steps Δs transform into variable steps $\Delta t = h_{n+1/2}$

Interaction of discrete flow and step size map



A single step forward: $y_n = \Phi_h(y_{n-1})$

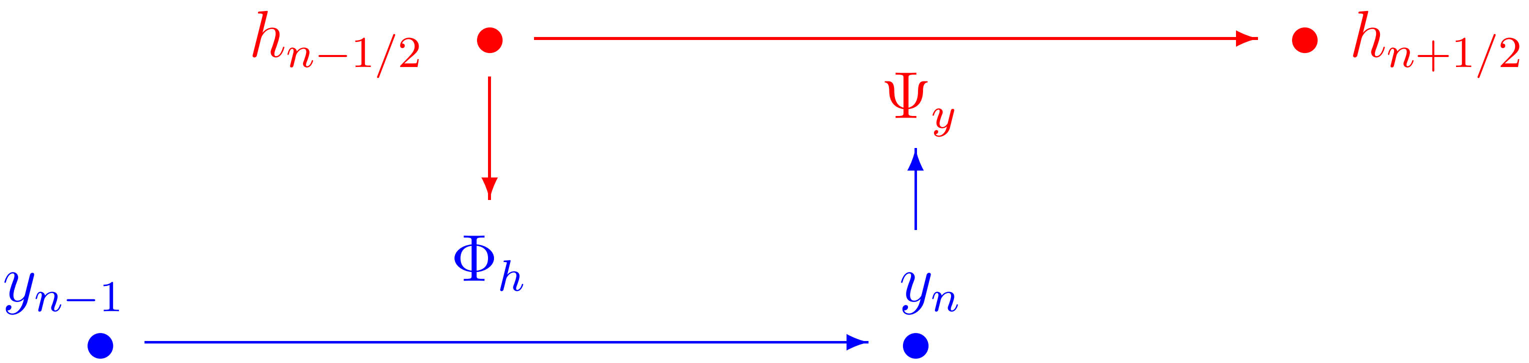
Interaction of discrete flow and step size map



A single step forward: $y_n = \Phi_{h_{n-1/2}}(y_{n-1})$

The discrete flow is parameterized by the actual step size

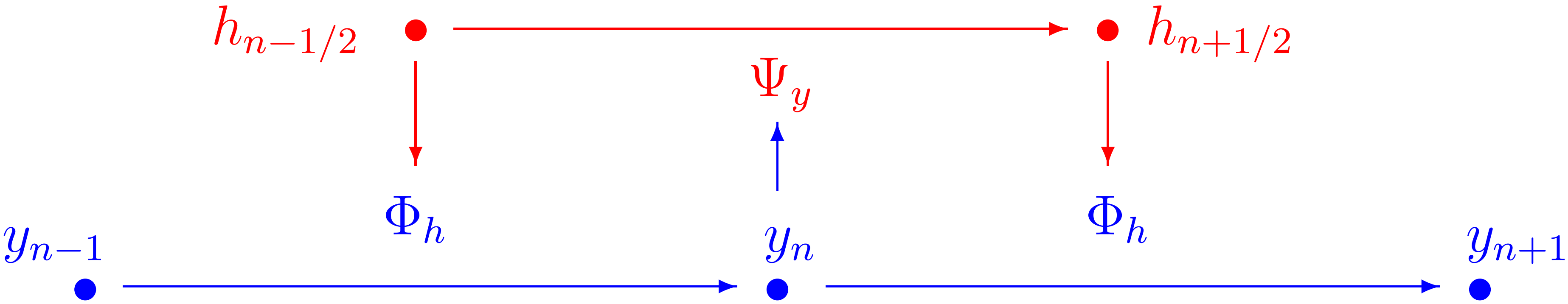
Interaction of discrete flow and step size map



Solution y_n *determines next step size:* $h_{n+1/2} = \Psi_{y_n}(h_{n-1/2})$

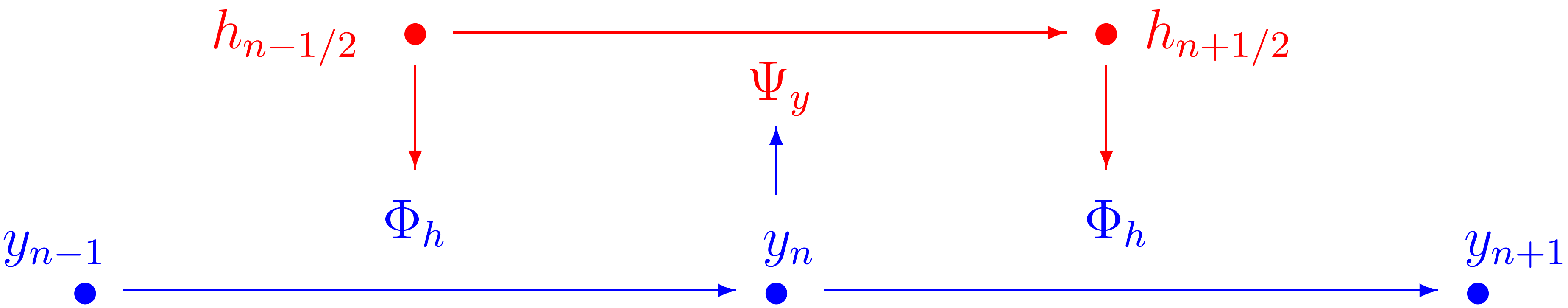
Step size map is parameterized by the solution

Interaction of discrete flow and step size map



New step size gives next step forward: $y_{n+1} = \Phi_{h_{n+1/2}}(y_n)$

Interaction of discrete flow and step size map



$$y_n = \Phi_h(y_{n-1})$$

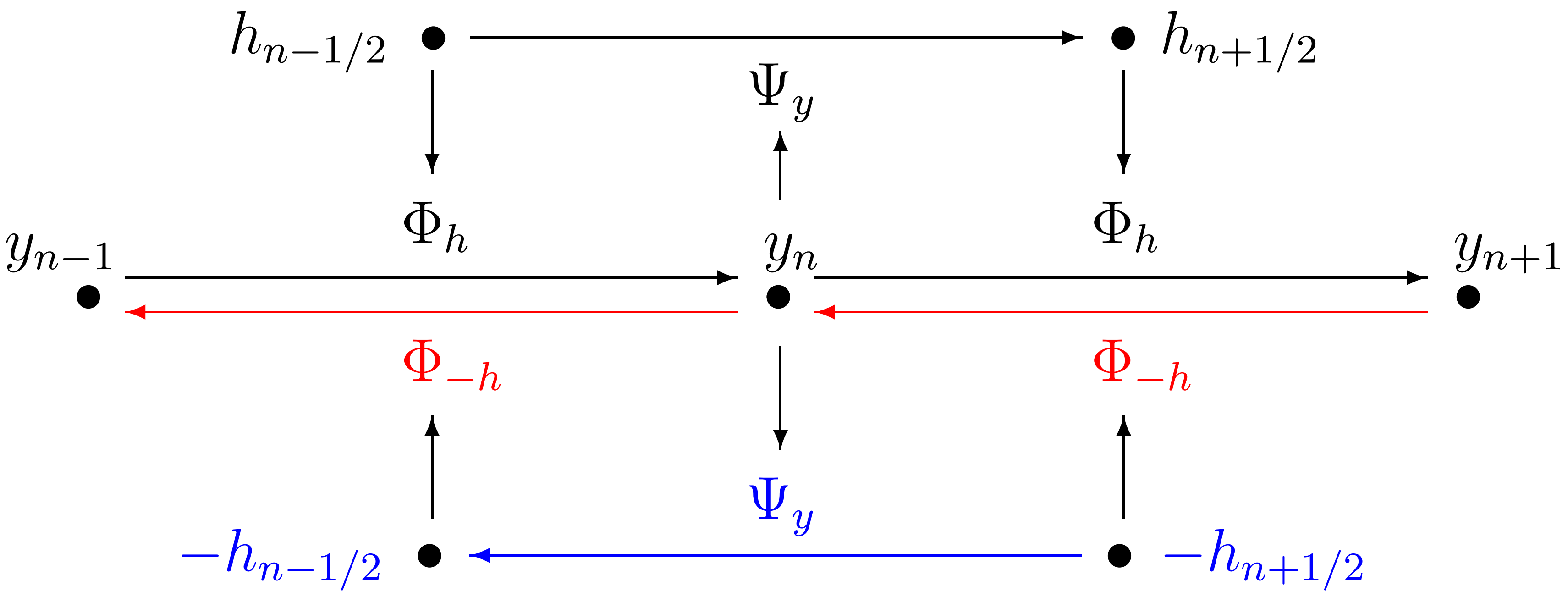
$$h_{n+1/2} = \Psi_y(h_{n-1/2})$$

Step size control adds dynamics interacting with method

Time reversal and time symmetric integration

Symmetric method: $\Phi_h^{-1} = \Phi_{-h}$

Symmetric control: $\Psi_y^{-1} = -\mathrm{id} \circ \Psi_y \circ (-\mathrm{id})$



Exact retracing of discrete orbit and step size sequence

Time symmetric step size control

Requirement $-\Psi$ is an *involution*

Note This precludes standard control (linear Ψ)

$$h_{n+1/2} = \theta \cdot h_{n-1/2}$$

as only $\theta = 1$ or $\theta = -1$ satisfy involution criterion

Need to seek nonlinear controllers!

Hamiltonian systems (with E. Hairer)

Find explicit step size controllers that are time symmetric, reversible, and preserve invariants

Consider step density objective $Q(y)/\rho = \mathrm{Const.}$

Note Do not implement this as an algebraic constraint!

$$\rho' = G(y)$$

Tracking Choose G so that ρ tracks Q

Hamiltonian (nonlinear) controllers

Let Q be symmetric w.r.t. time reversal $t \to -t$ and take

$$G(y) := \mathrm{D}_t(\log Q) = \frac{\mathrm{D}_t Q}{Q} = \frac{\nabla Q \cdot F(y)}{Q}$$

The step density controller

$$\rho' = \nabla Q \cdot F(y)/Q$$
$$t' = 1/\rho$$

is then *Hamiltonian* ($\rho' = \hbar_t \,;\ t' = -\hbar_\rho$) with first integral

$$\hbar(\rho, t) = \log[Q(y(t))/\rho] = \mathrm{Const.}$$

Discrete step density control

Discretize $\rho' = \nabla Q \cdot F(y)/Q$ with a *symplectic* method:

Explicit midpoint method

$$\rho_{n+1/2} = \rho_{n-1/2} + \frac{\varepsilon \nabla Q \cdot F(y)}{Q}$$

nearly preserves first integral $Q(y(t))/\rho$ "forever"!

Stable, non-oscillatory, integrating controller

Adaptive Störmer–Verlet method

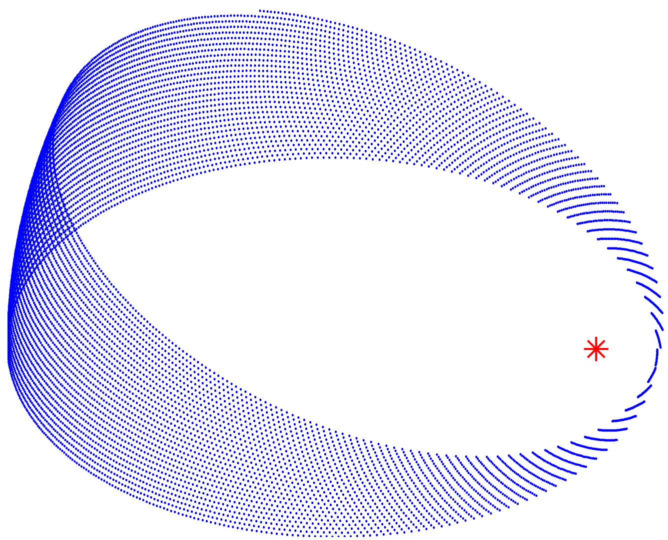
For Newtonian mechanics

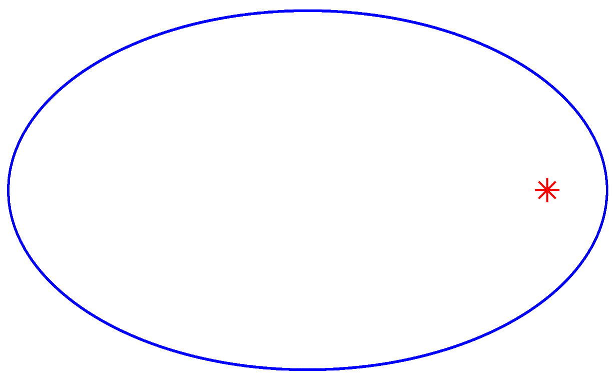
$$\begin{aligned} \dot{q} &= p \\ \dot{p} &= f(q) \end{aligned}$$

and step size control $hQ^\alpha = C$ with $Q = 1/\sqrt{q^{\mathrm{T}}q}$

$$\begin{aligned} \rho_{n+1/2} &= \rho_{n-1/2} + \varepsilon\,\alpha\,p_n^{\mathrm{T}}q_n/q_n^{\mathrm{T}}q_n \\ \Delta t &= \varepsilon/\rho_{n+1/2} \\ p_{n+1/2} &= p_n + \Delta t \cdot f(q_n)/2 \\ q_{n+1} &= q_n + \Delta t \cdot p_{n+1/2} \\ p_{n+1} &= p_{n+1/2} + \Delta t \cdot f(q_{n+1})/2 \\ t_{n+1} &= t_n + \Delta t \end{aligned}$$

Kepler problem, $e = 0.8$; 30 orbits; 10,000 steps

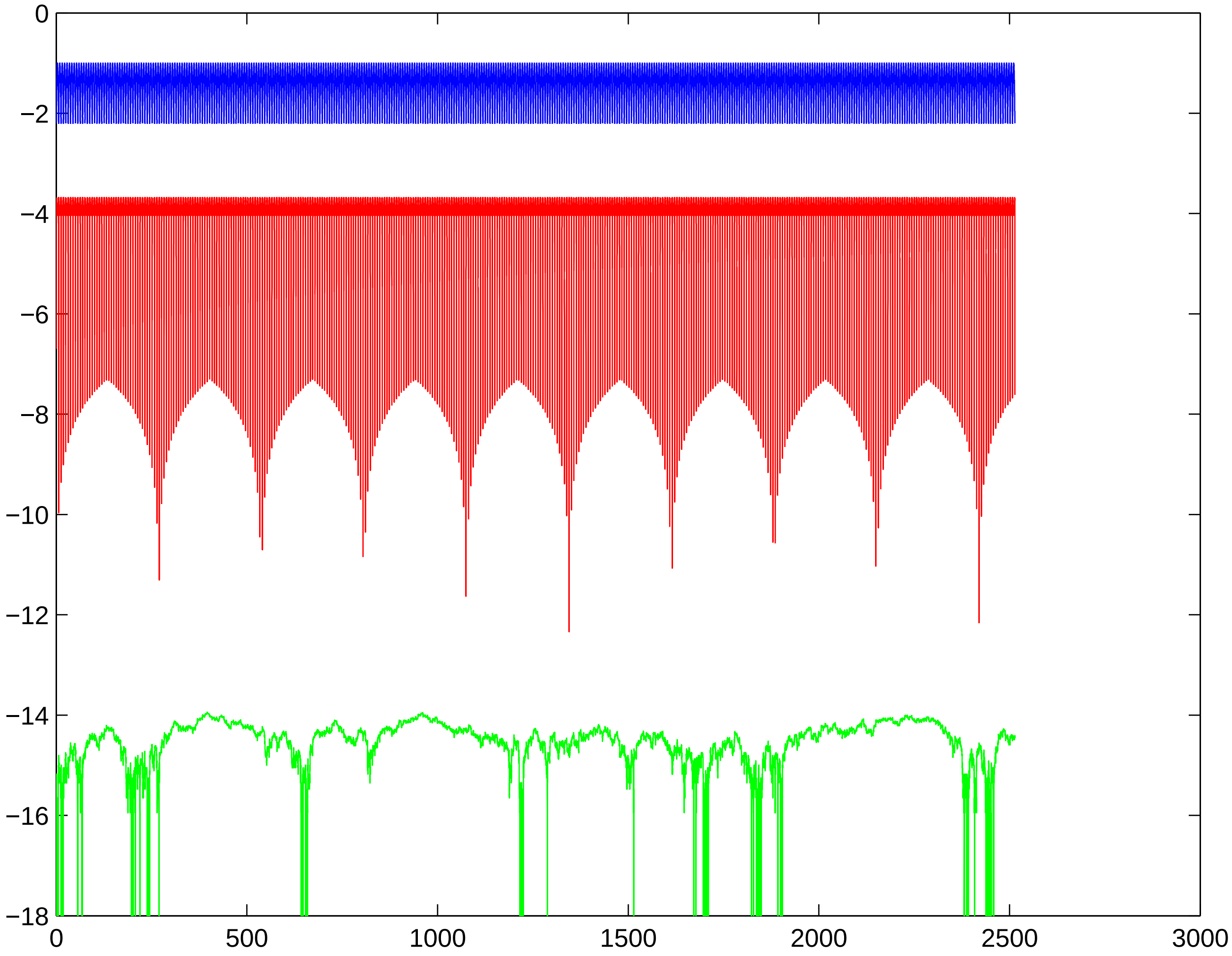




Constant step Störmer Verlet *Adaptive Störmer Verlet*

Precession suppressed, 30 times smaller errors, same cost

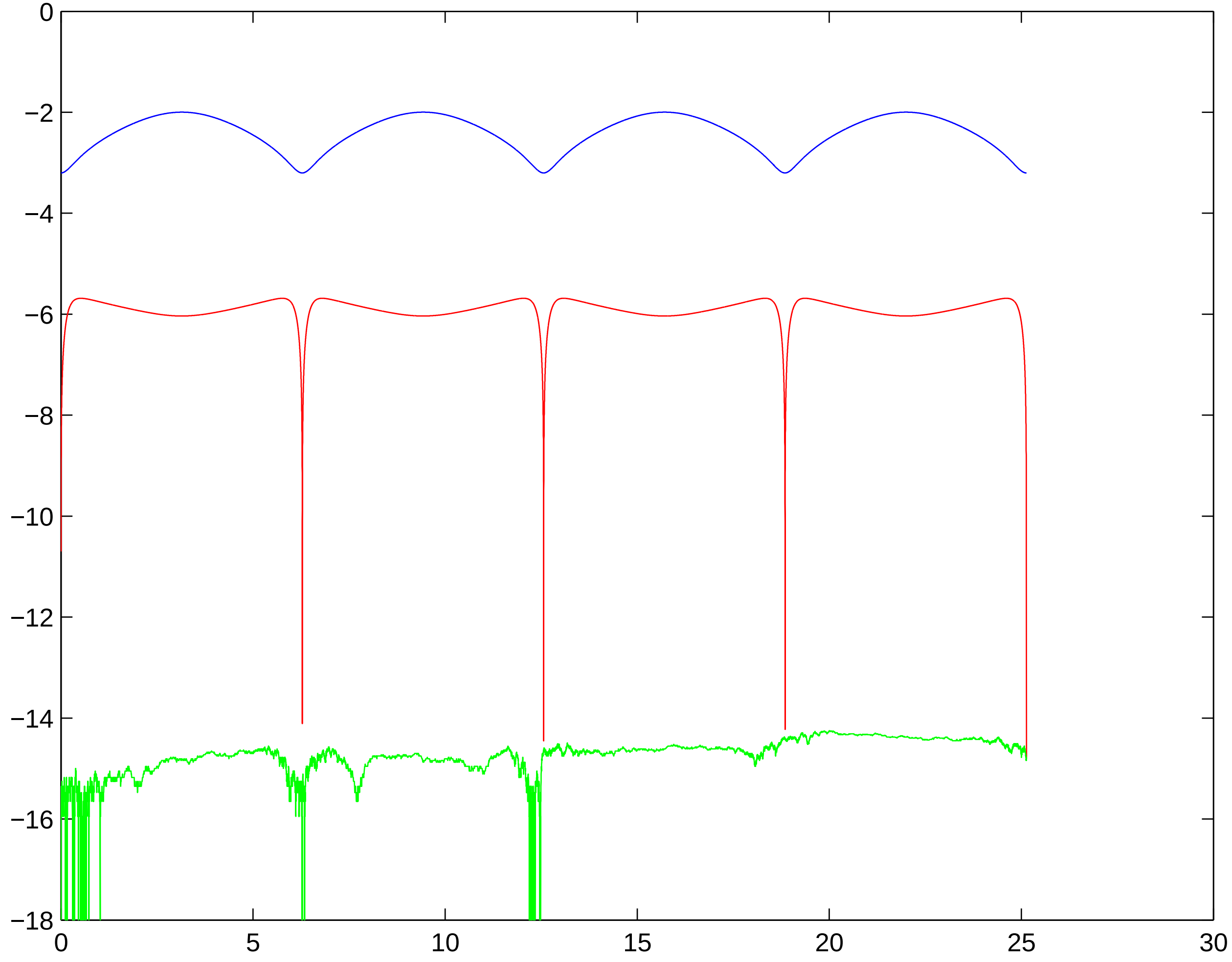
Energy conservation: 400 orbits; 100,000 steps



Average step size $2\pi/250$; variation by a factor 16

Hamiltonian error $< 2 \cdot 10^{-4}$; Angular momentum exact

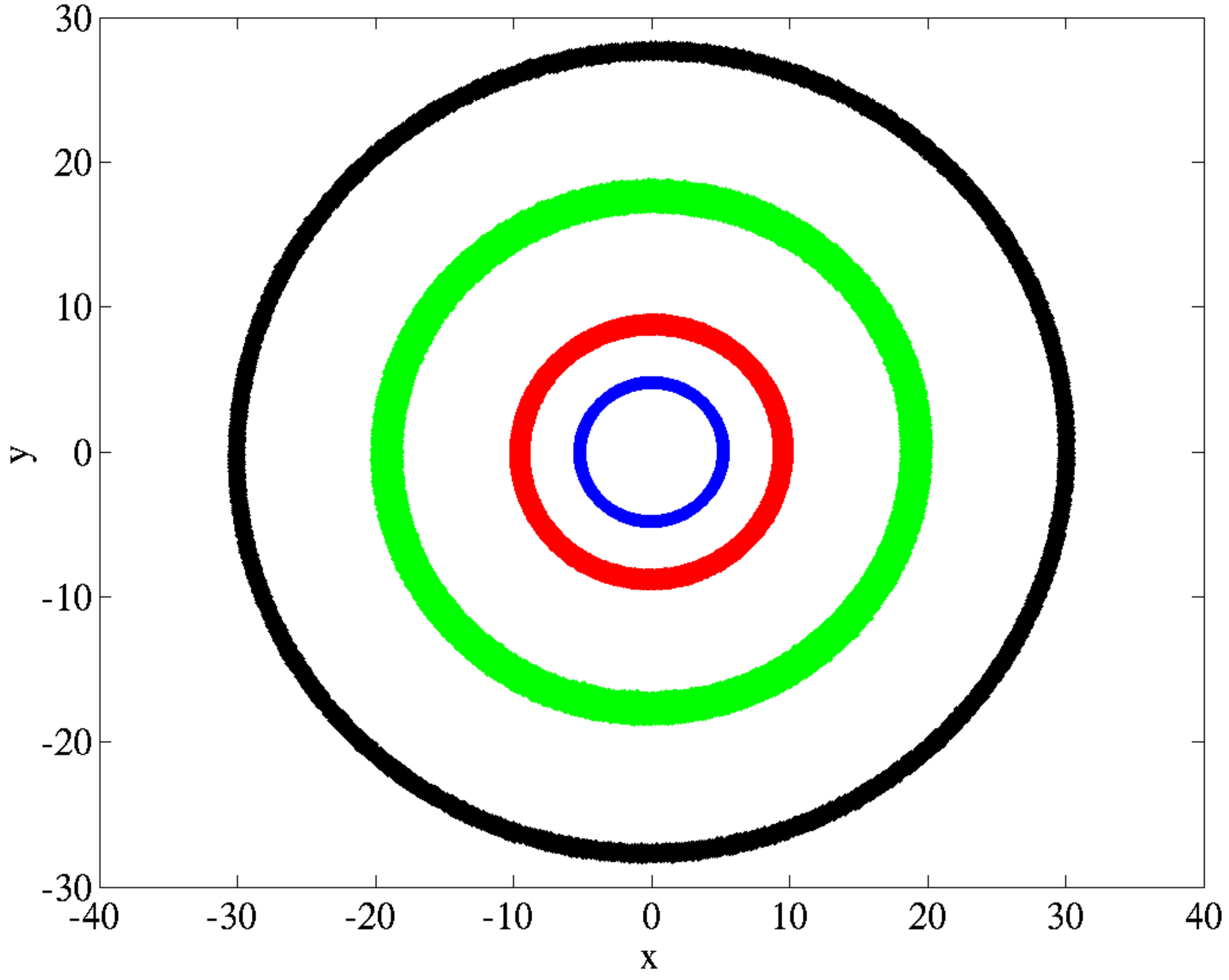
Energy conservation: 4 orbits; 10,000 steps



Average step size $2\pi/2500$; Angular momentum exact

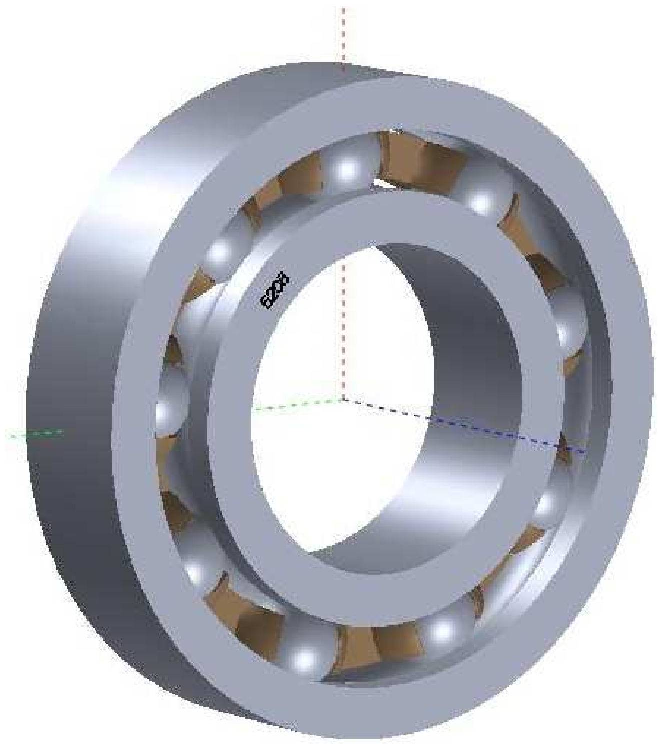
Hamiltonian error $< 2 \cdot 10^{-6}$; 2nd order accuracy

3D solar system, 1.38 billion years, 10^{10} steps



Symplectic control $G(p,q) = p^{\mathrm{T}}q/q^{\mathrm{T}}q$; integral gain $\alpha = 1$
10^5 steps/dot (with *Philip Sharp, Univ. of Auckland*)

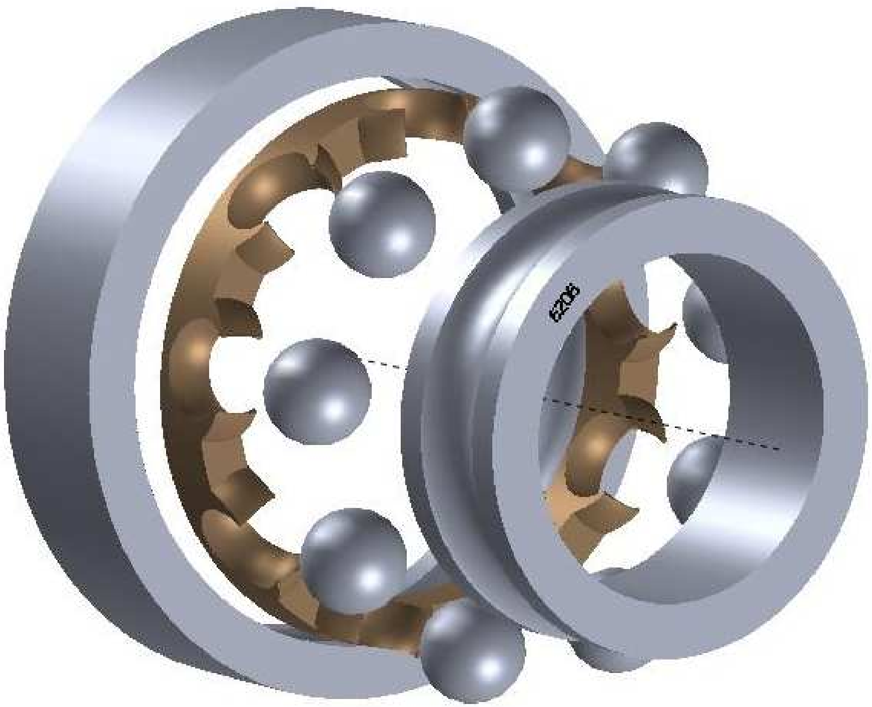
4. Rolling bearing dynamics



SKF software BEAST (BEAring Simulation Tool)

Used for R& D, and as virtual test rig

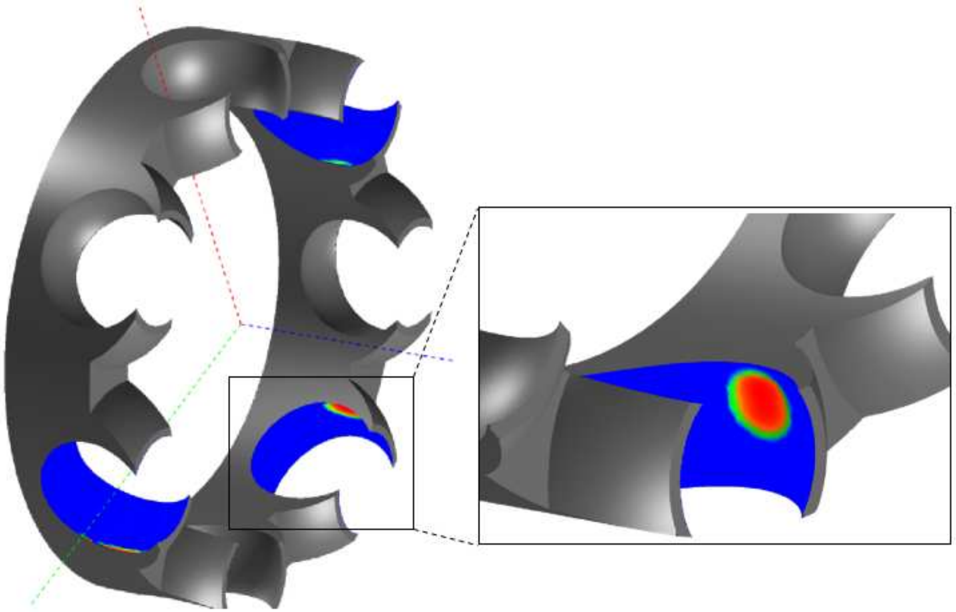
Rolling bearing dynamics



SKF software BEAST (BEAring Simulation Tool)

Used for R& D, and as virtual test rig

Rolling bearing dynamics



Dynamics governed by mechanical contacts between bodies

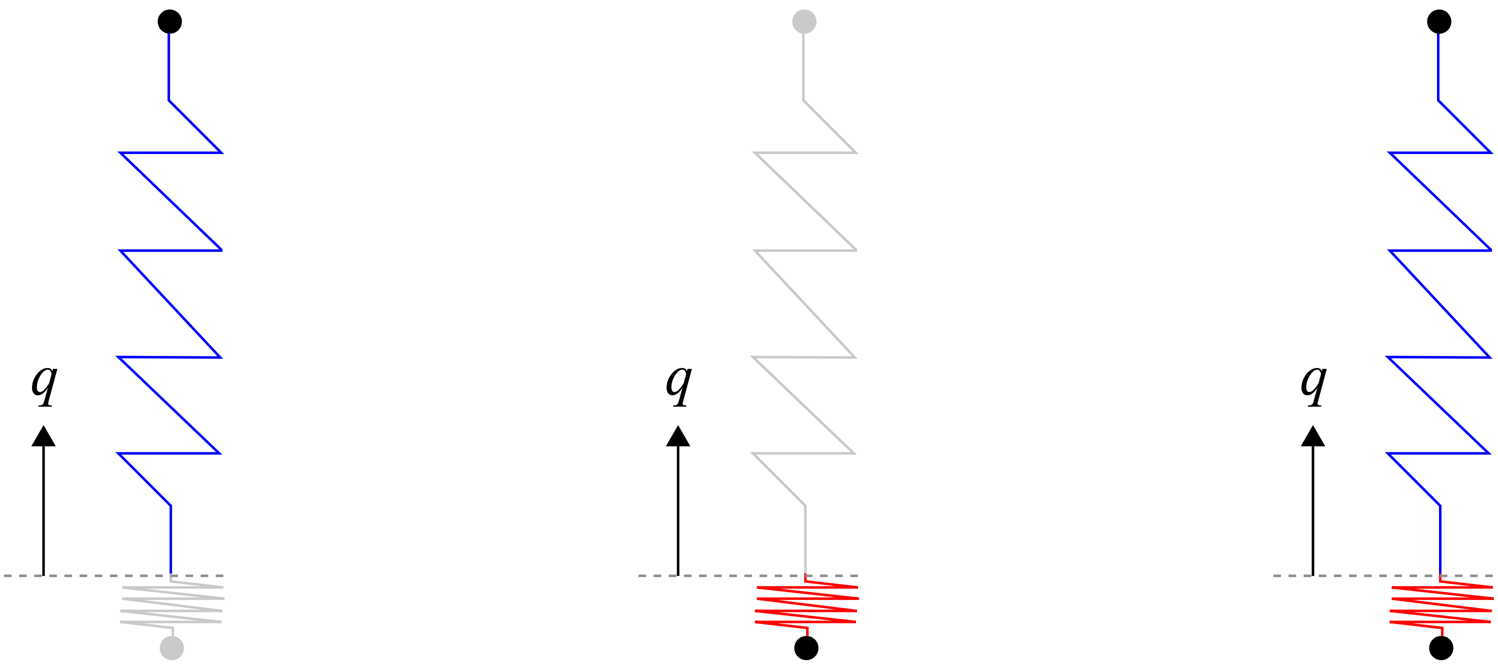
Described by complex force laws

Characteristics of governing equations

- Relatively few d.o.f., ~ 1000
- Computationally very expensive to evaluate forces
- Large rotations
- Some components in energy conservative motion
- Others subject to friction
- Only weak dissipation
- Geometric integrators (mechanical integrators) do well
- Time-step adaptivity required
- Exploit MBS structure with contacts

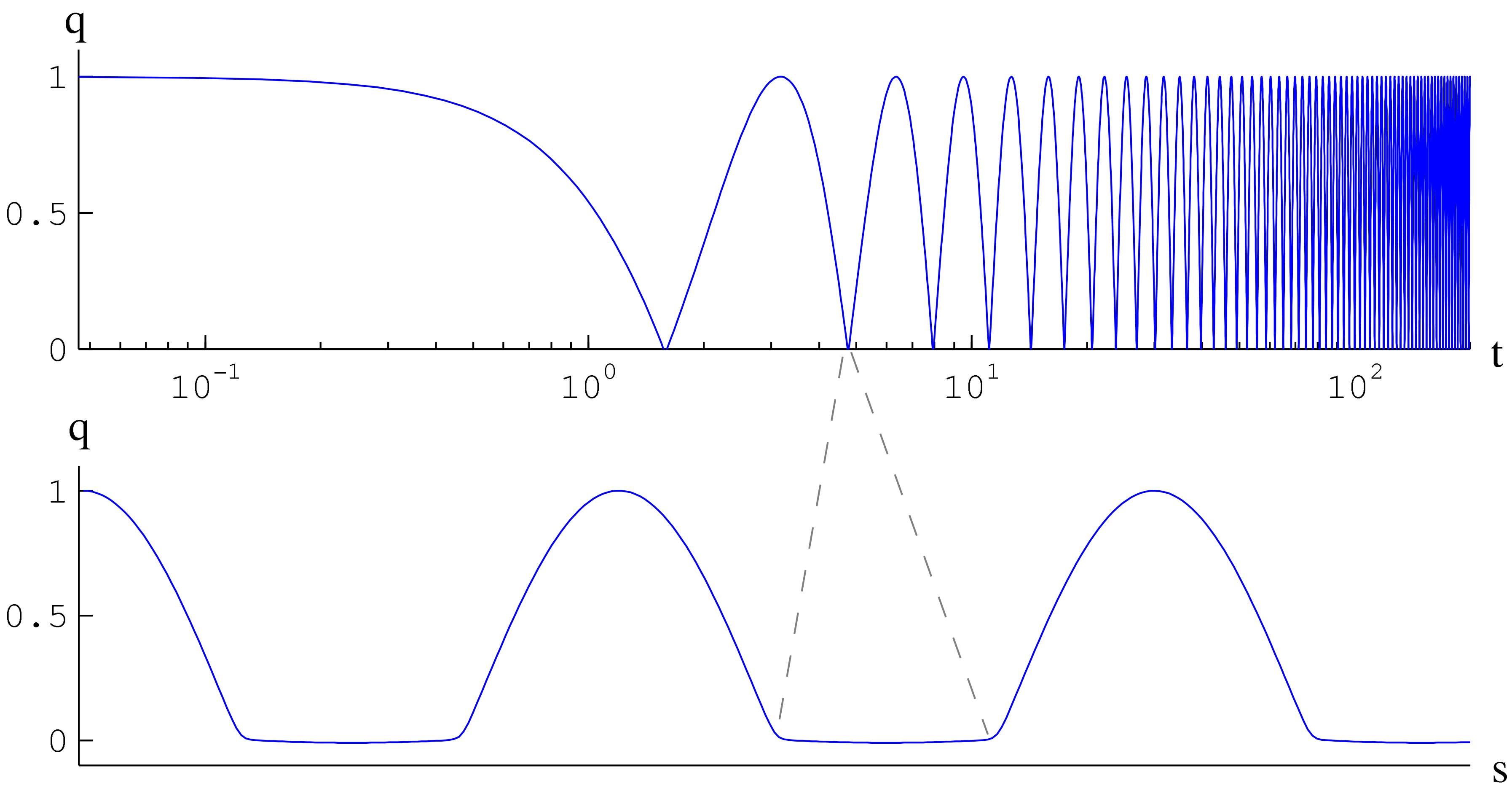
Time adaptivity test problem – stiff/loose spring

$$L(q,\dot{q}) = \frac{\dot{q}^2}{2} - V(q) \quad \text{with} \quad V(q) = \frac{1}{2}\max(0,q)^2 + \frac{10^4}{2}\min(0,q)^2$$



Scaling function $\sim V''(q)$ **NOTE** exact solution known

Effect of time transformation



Systems with weak Rayleigh damping

Phase space $\mathbb{R}^{2n}$ with coordinates $z = (q, p)$

Governing equations

$$\dot{q} = T_p$$
$$\dot{p} = -V_q - \varepsilon \mathsf{D}(q)\dot{q}$$

also written $\dot{z} = X_H(z) + \varepsilon Y(z)$

- Energy $H(q, p) = T(p) + V(q)$
- $\mathsf{D} : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ matrix-valued smooth function
- $\mathsf{D}(q)$ positive semi-definite
- Standard kinetic energy $T(p) = p^T \mathsf{M}^{-1} p/2$

Energy dissipation

Evolution of H

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \nabla H \cdot \dot{z} = \underbrace{\nabla H \cdot (X_H}_{=0} + \varepsilon Y) = \varepsilon \nabla H \cdot Y$$

$$= -\varepsilon(\mathsf{M}^{-1}p)^{\mathrm{T}}\mathsf{D}(q)\mathsf{M}^{-1}p \leq 0$$

- *Energy dissipation rate is* $\mathrm{O}(\varepsilon)$
- What is the discrete (numerical) dissipation rate?
- Use method that becomes conservative for $\varepsilon = 0$

Construction of explicit methods

- **Basic idea** Use splitting method

$$\dot{z} = X_H(z) + \varepsilon Y(z) = \underbrace{X_T(z)}_{X_1(z)} + \underbrace{X_V(z) + \varepsilon Y(z)}_{X_2(z)}$$

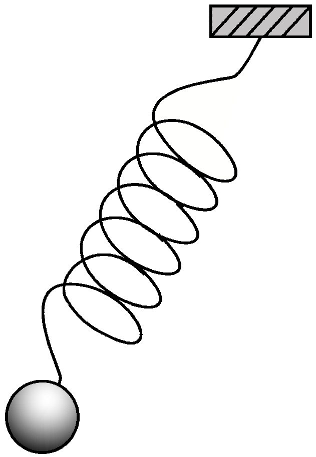
- Both X_1 and X_2 can be integrated by explicit methods
  - $X_1 = [\mathsf{M}^{-1}p, 0]^T$ by forward Euler
  - $X_2 = [0, -V'(q) - \varepsilon \mathsf{D}(q)\mathsf{M}^{-1}p]^T$ since it is linear in p

- **Suggestion** *Dissipative Störmer–Verlet (DSV)*

$$\varphi_{h,\varepsilon} = \exp(hX_1/2) \circ \exp(hX_2) \circ \exp(hX_1/2)$$

- $\varphi_{h,0}$ is classical Störmer–Verlet

Test problem. Damped elastic 3D–pendulum

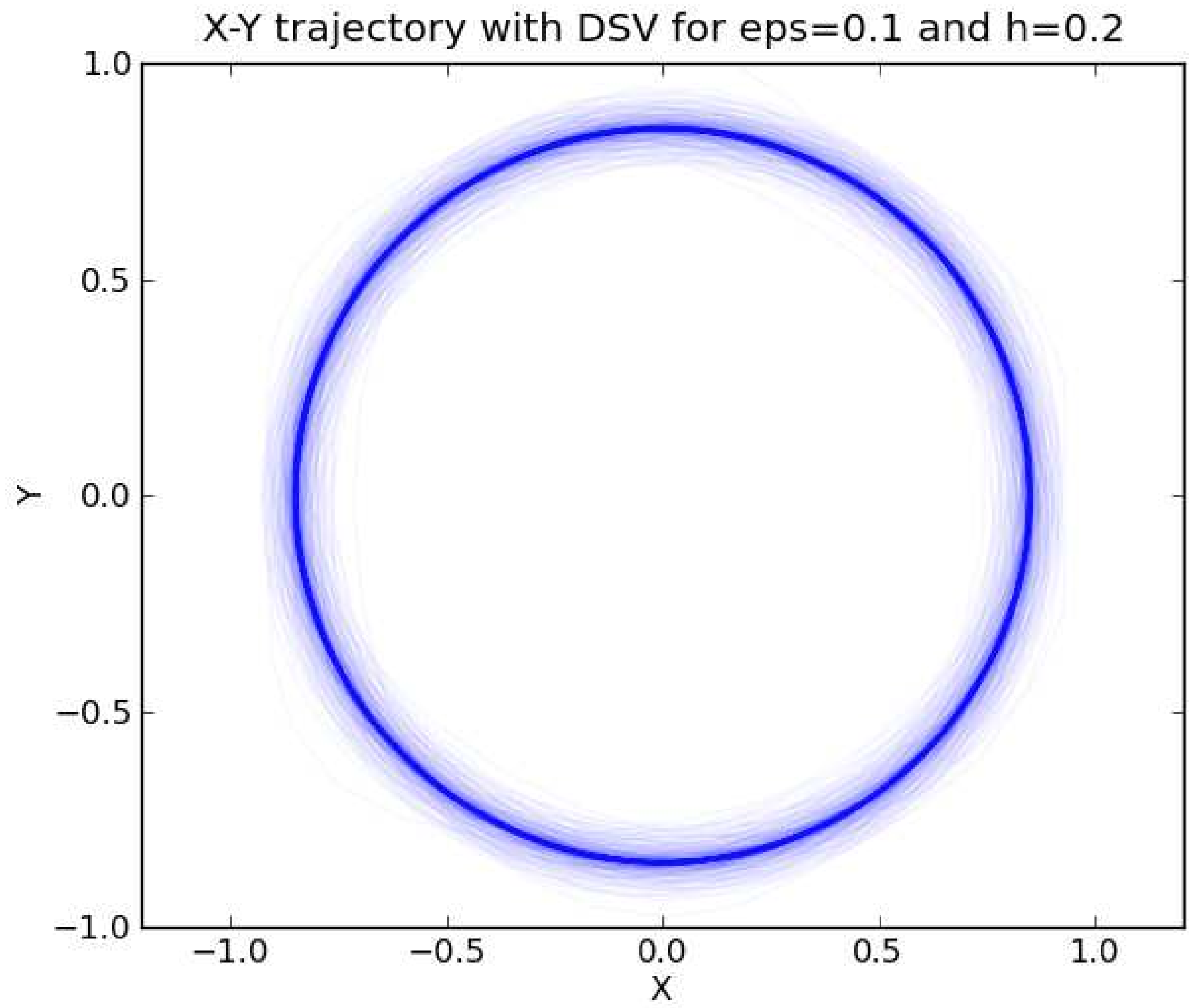


Governing equations for $q, p \in \mathbb{R}^3$

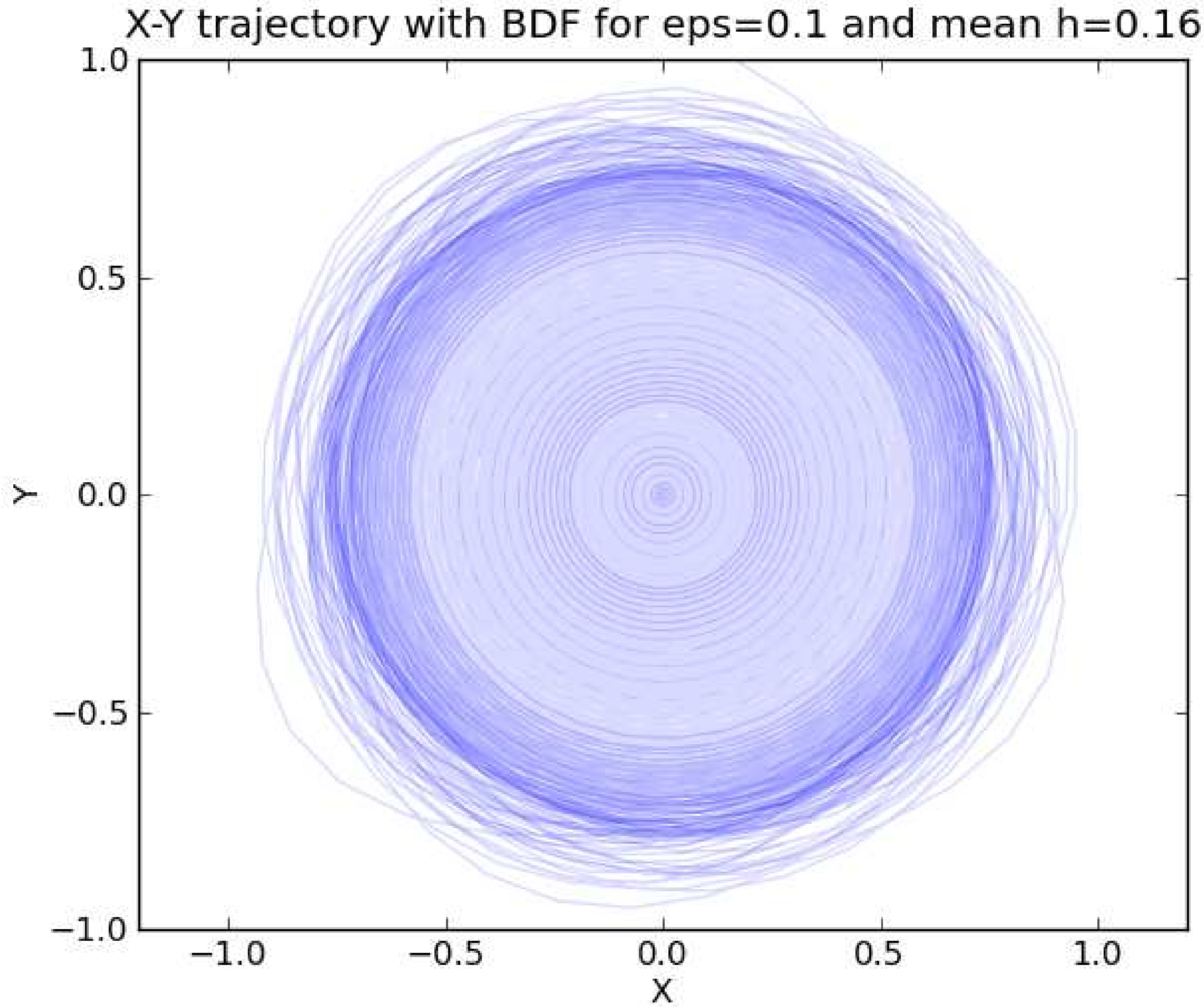
$$\dot{q} = p$$

$$\dot{p} = -\left(1 - \frac{1}{\|q\|}\right)q - g - \varepsilon \frac{qq^{\mathrm{T}}p}{\|q\|^2}$$

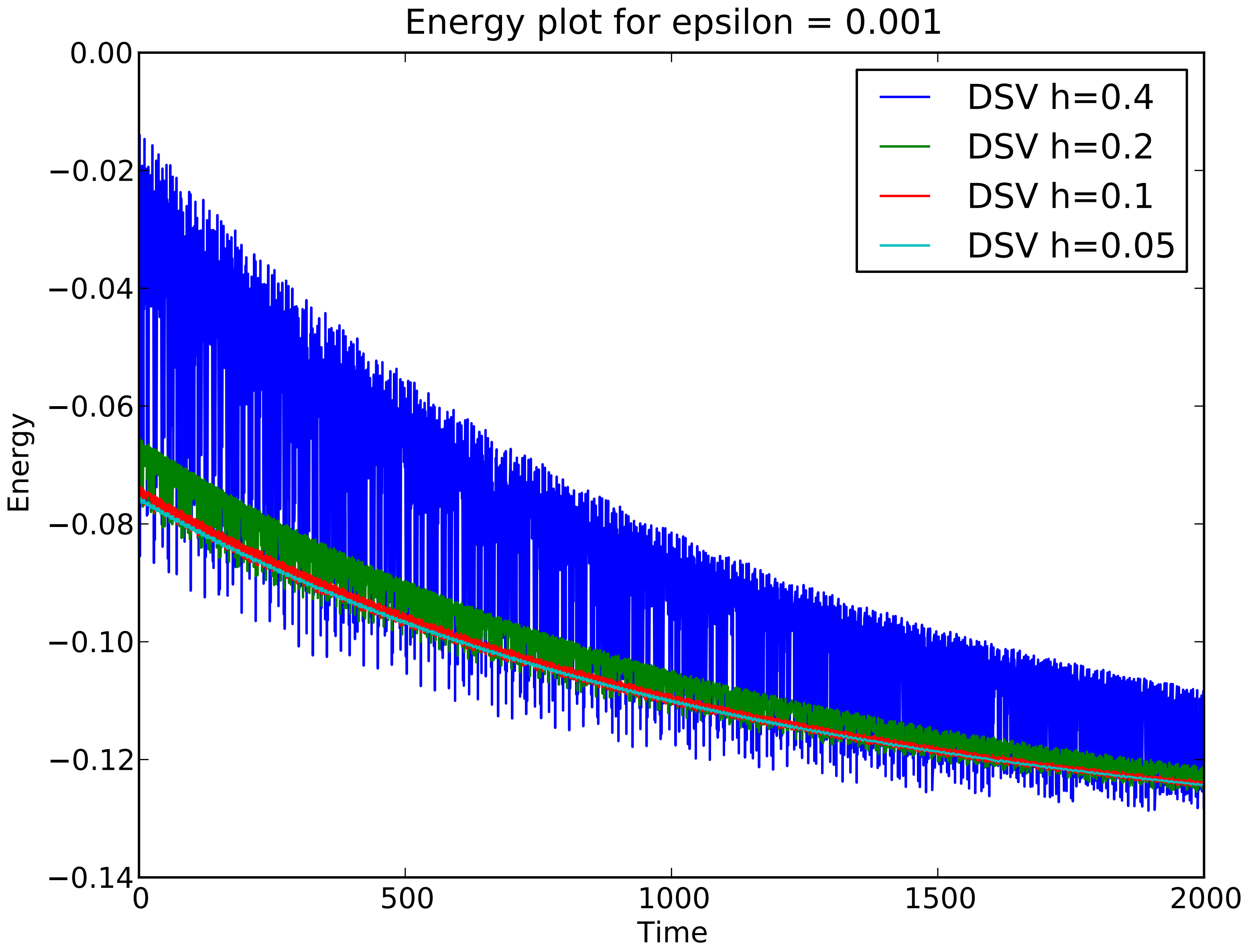
DSV xy–plane orbit projection



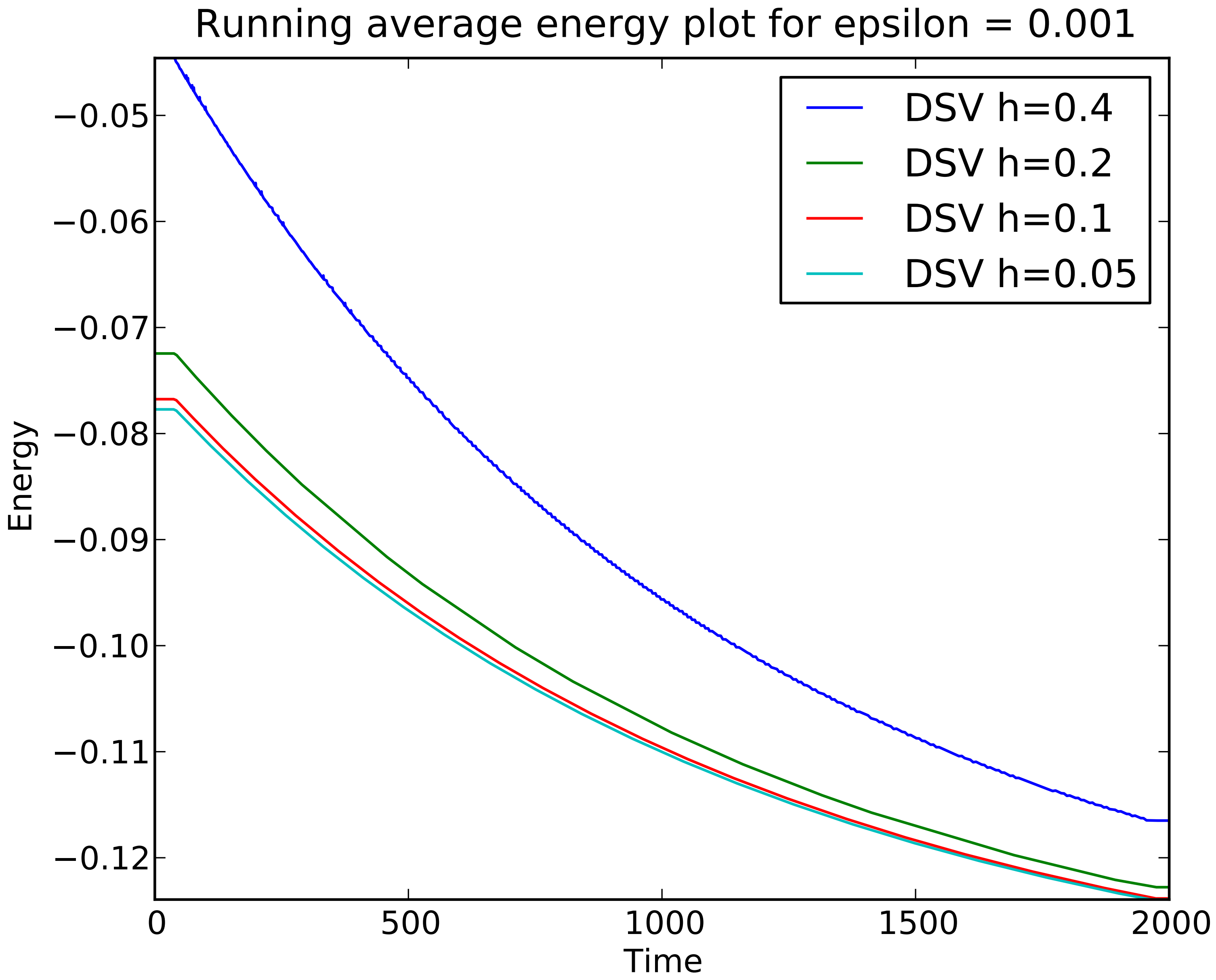
BDF xy–plane orbit projection



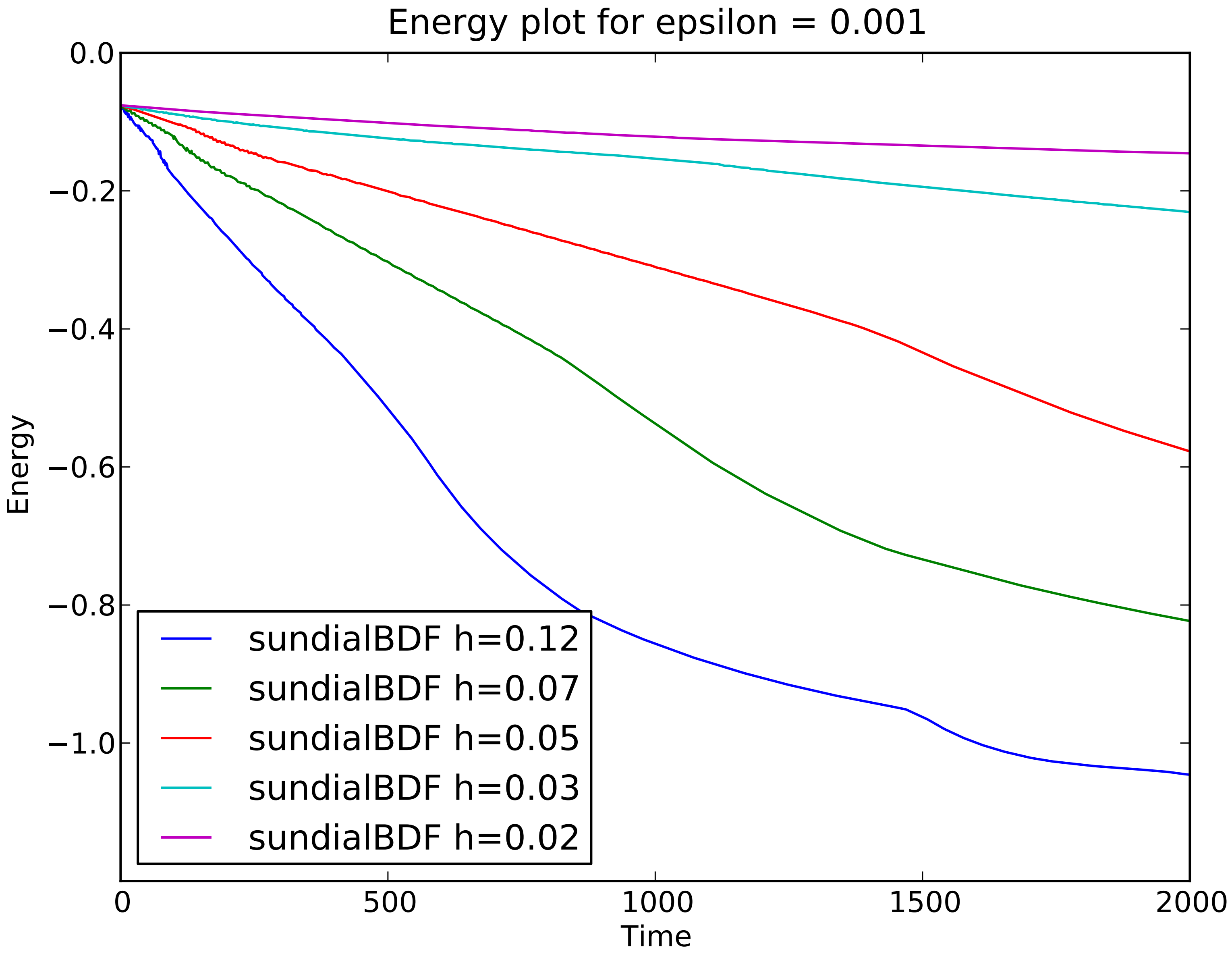
DSV energy dissipation as a function of h



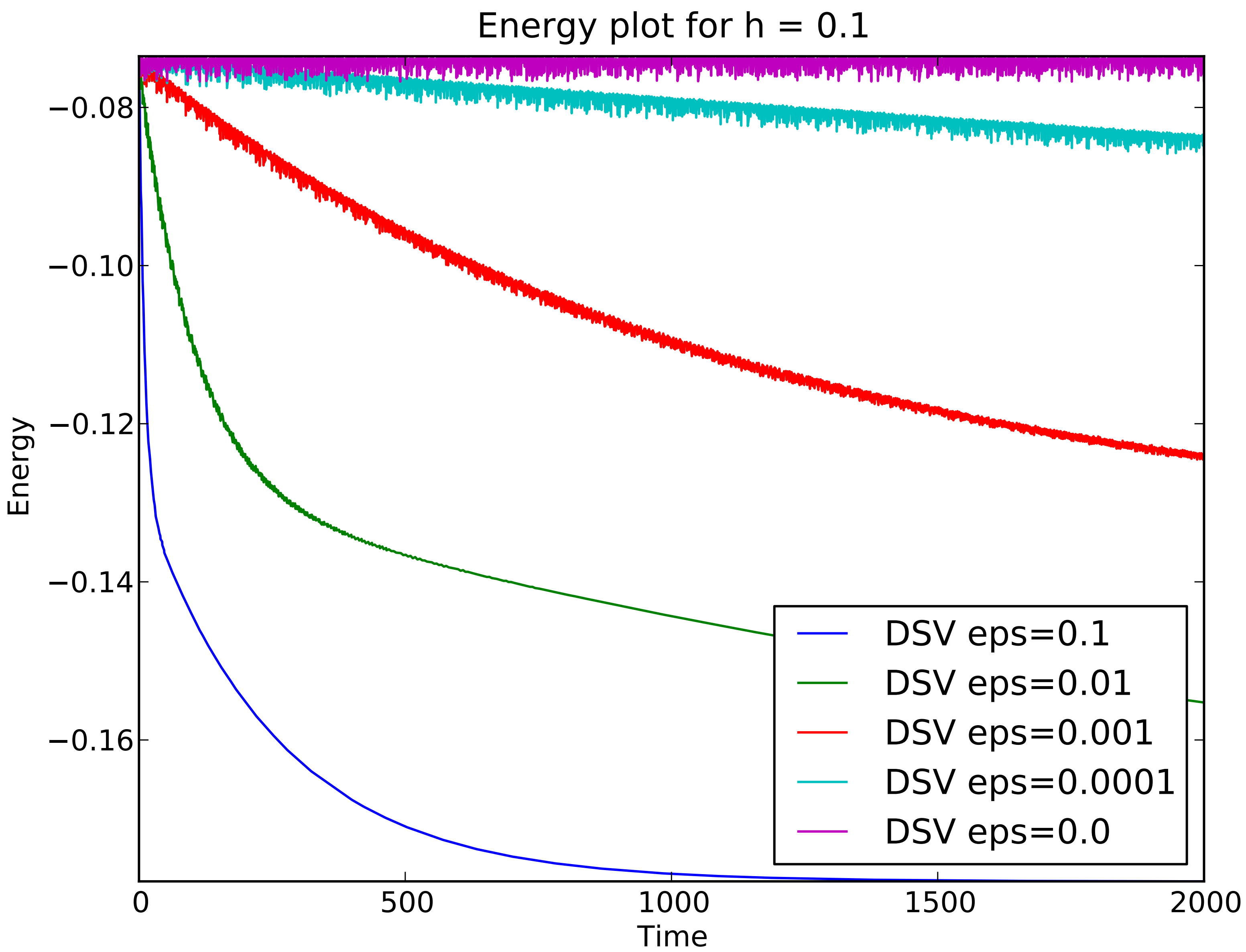
DSV averaged energy as a function of h



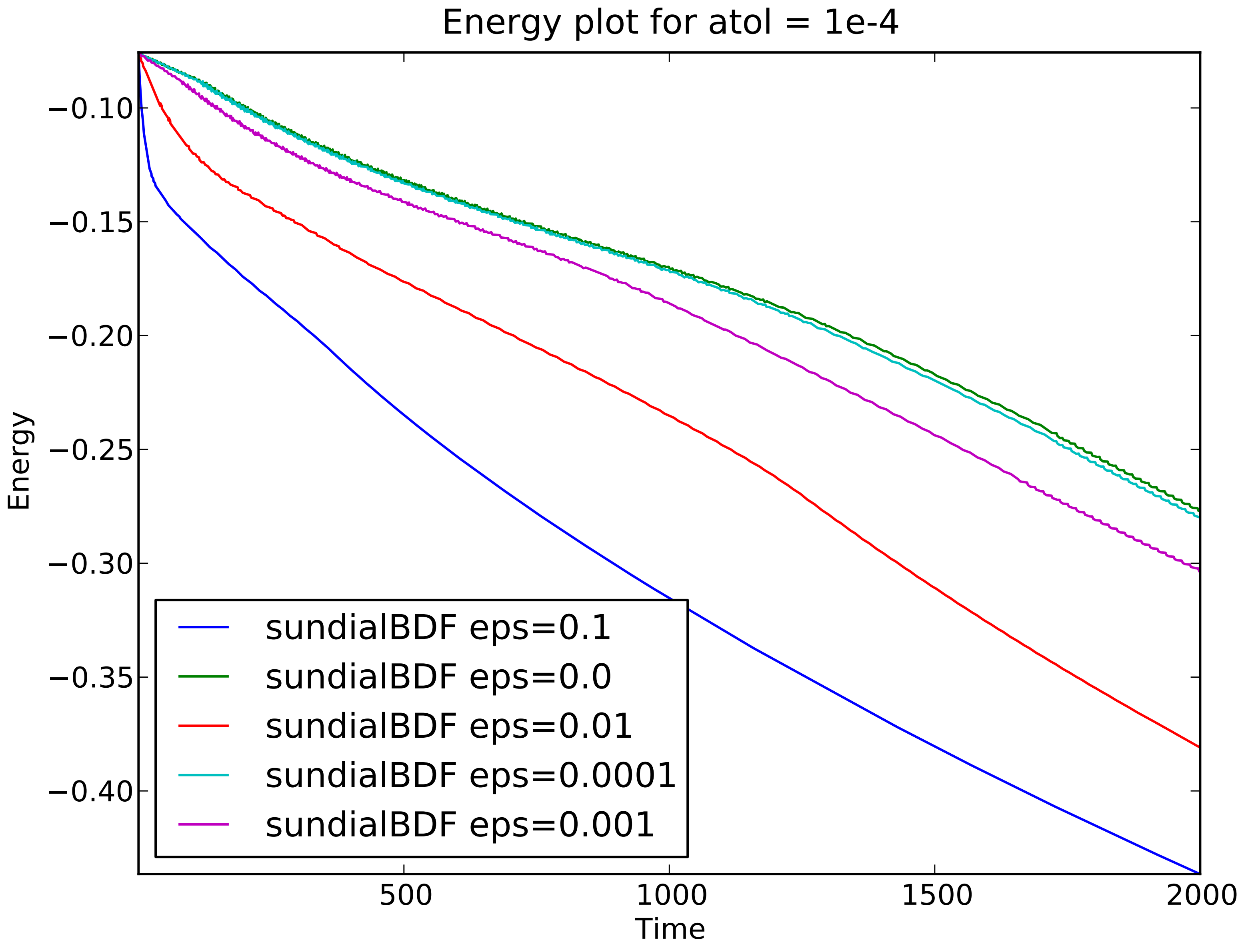
BDF energy dissipation as a function of average h



DSV energy dissipation as a function of ε



BDF energy dissipation as a function of ε



Conclusions

- Adaptivity based on

  Digital control

  Signal processing

  Density functions

  Symmetry/reversibility

  Energy dissipation

- Well-defined sets of controllers
- Theoretical backing and analysis
- Usually not too difficult to implement
- Improved computational stability
- No added computational expense
- Significant improvements for Hamiltonian systems

Thank you!