Numerics of Dynamics

Tony Humphries

Tony.Humphries@mcgill.ca

McGill

Symposium on:

Numerical Algorithms for Differential Equations and Dynamical Systems

CMS/CAIMS Summer 2004 Meeting

June 13-15

Dalhousie University

Schedule for Numerical Algorithms for Differential Equations and Dynamical Systems

Sunday June 13

1:30-2:20	Tony Humphries	Numerics and Dynamics
2:30-3:20	Raymond Spiteri	A Comparison of Stiffness Detection Methods for Initial-Value ODEs
3:45-4:35	Wayne Hayes	A Practical Shadowing-based Timestep Criterion for Galaxy Simulations
4:45-5:35	Eusebius Doedel	Bifurcation of periodic orbits in the Circular Restricted 3-Body Problem

Monday June 14

1:30-2:20	Ned Nedialkov	Solving Differential-Algebraic Equations by Taylor Series
2:30-2:55	David Cottrell	A backward analysis of simple collisions
3:00-3:25	Martin Gander	Moving Mesh Methods and Energy Conservation

Numerics of Dynamics

In dynamical systems it is often the asymptotic (large time) behaviour for general initial conditions that is of interest. However traditional numerical analysis typically focuses on the solution of a given initial value problem over a finite time interval, usually providing error bounds which grow exponentially in time, which are not directly useful in a dynamical systems context. Over the last two decades there has been an explosion of work in the "numerics of dynamics" providing techniques for numerically studying dynamical systems and rigorous meanings for the pretty pictures obtained. In this session we will see some of these techniques including direct methods for computing special trajectories (eg periodic orbits), backward error analysis, stiffness and adaptive time-stepping. In this overview talk we introduce some issues that arise, and the techniques used to tackle them. Along the way we will show that the backward Euler method is a very bad method.

ODEs and Dynamical Systems

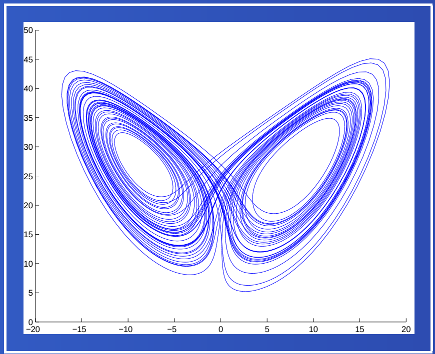
Consider autonomous ODE

$$\dot{u} = f(u) \in \mathbb{R}^p$$

Existence and uniqueness of solutions allows us to define an evolution operator

$$S(t) : u(0) \rightarrow u(t)$$

and plot solutions as curves (parameterized by t) in phase space $\mathbb{R}^p$.

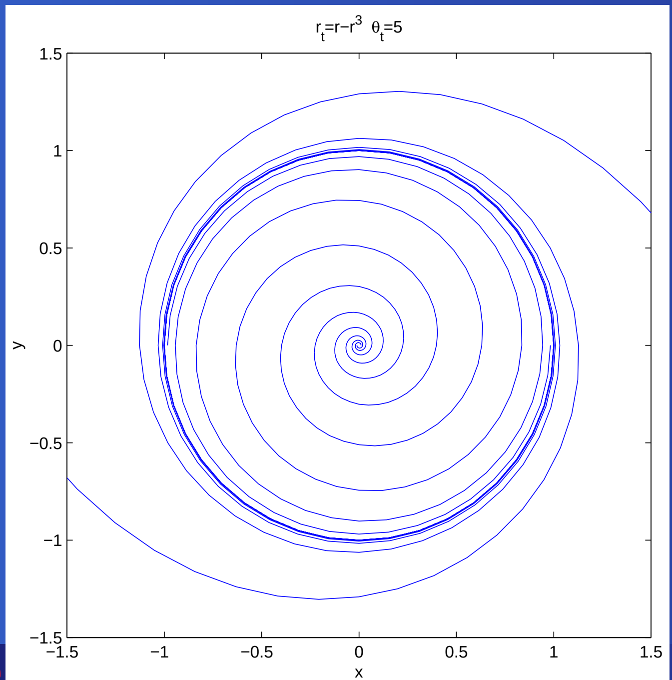


Invariant Sets and Attractors

- Asymptotic behaviour defined by ω-limit sets which are invariant under $S(t)$.
- Invariant sets include fixed points, periodic orbits, invariant tori, heteroclinic and homoclinic connections, strange attractors.

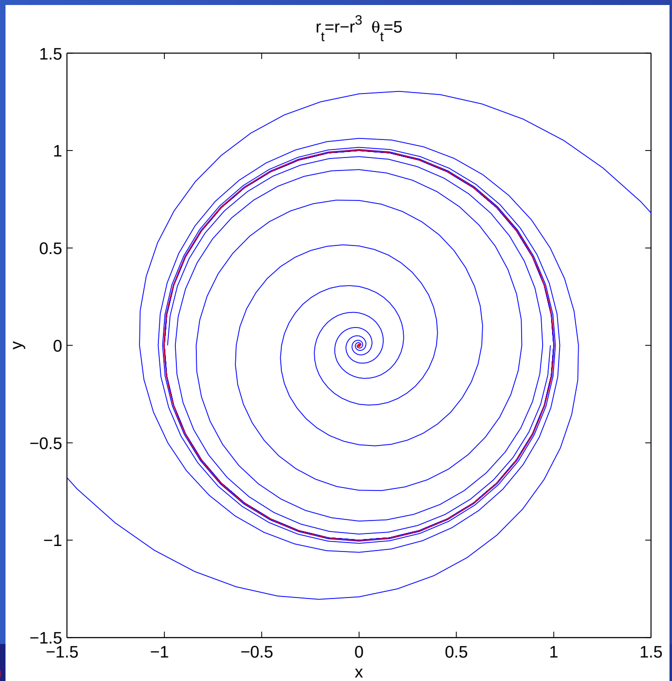
Invariant Sets and Attractors

- Asymptotic behaviour defined by ω-limit sets which are invariant under $S(t)$.
- Invariant sets include fixed points, periodic orbits, invariant tori, heteroclinic and homoclinic connections, strange attractors.



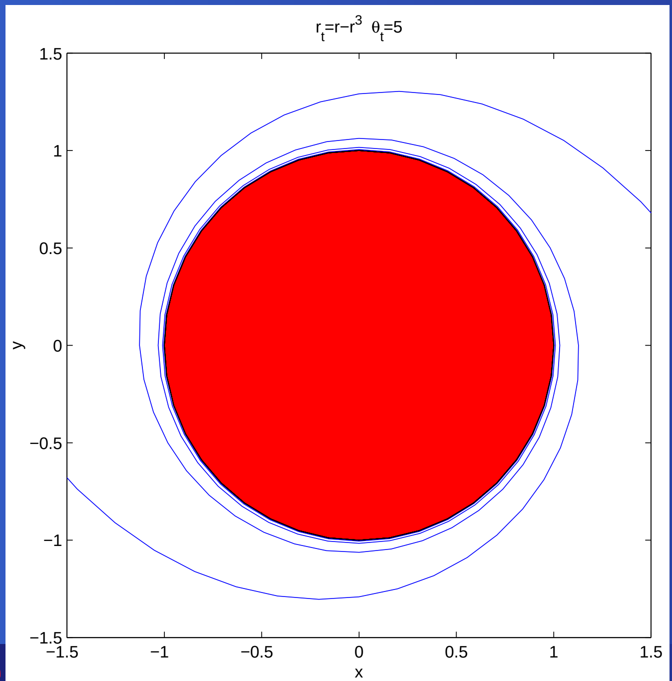
Invariant Sets and Attractors

- Asymptotic behaviour defined by ω-limit sets which are invariant under $S(t)$.
- Invariant sets include fixed points, periodic orbits, invariant tori, heteroclinic and homoclinic connections, strange attractors.



Invariant Sets and Attractors

- Asymptotic behaviour defined by ω-limit sets which are invariant under $S(t)$.
- Invariant sets include fixed points, periodic orbits, invariant tori, heteroclinic and homoclinic connections, strange attractors.



Numerical Approximation

u_n is approx to $u(nh)$ which solves $\dot{u} = f(u)$, $u(0) = U$.

- Forward Euler (order 1): $u_{n+1} = u_n + hf(u_n)$
- Backward Euler (order 1): $u_{n+1} = u_n + hf(u_{n+1})$
- General order p order implicit and explicit Runge-Kutta methods

Classical error bound:

$$\|u(t_n) - u_n\| \leq Ch^{p+1}(e^{Lt_n} - 1).$$

- “Traditional” numerical analysis fixes finite time interval $[0, T]$ and initial condition $u(0) = U$ and considers $h \to 0$.
- Error bound grows exponentially in time, so what do long-time numerics mean ?

Taylor Series Methods

Runge-Kutta methods are not the only numerical methods for ODEs...

- Runge-Kutta methods approximate the Taylor series of the exact solution just evaluations of $f(u)$.
- This avoids symbolic differentiation of f which would be required to evaluate the Taylor series directly.
- With modern computer power and techniques symbolic differentiation is not so expensive as it once seemed, and Taylor series solutions can be competitive for some applications.
- N. Nedialkov will present Taylor Series methods for solving high-index differential algebraic equations (DAEs).

The Direct and Indirect Approach

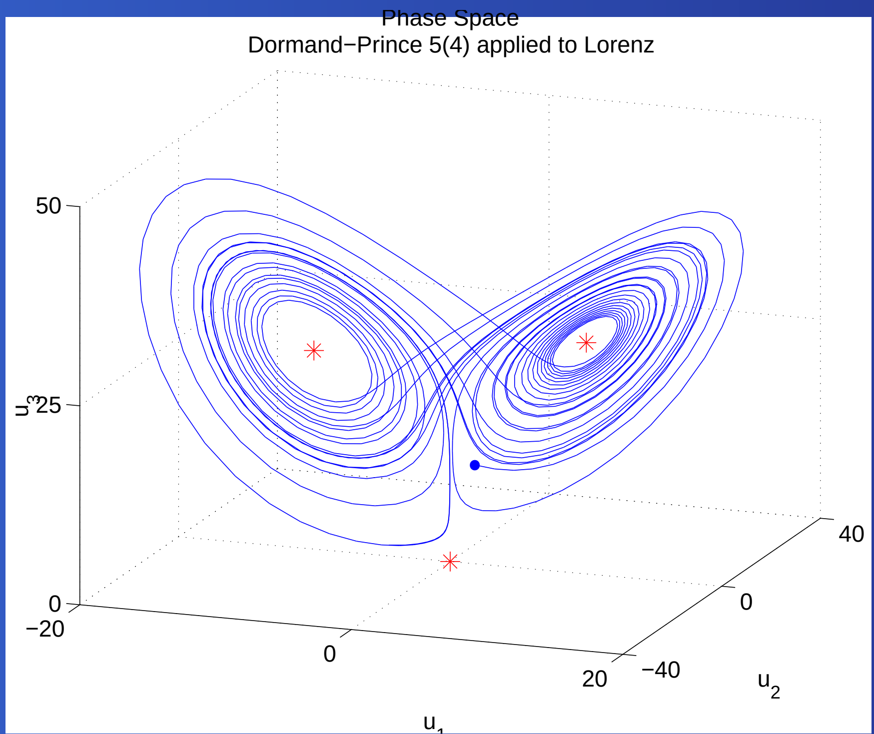
Two approaches to

$$\dot{u}(t) = f(u(t)), \qquad u(0) = U \in \mathbb{R}^d,$$

1. Direct Approach
   Set up equations to directly solve for interesting invariant sets.
2. Indirect Approach
   Simulate dynamical system numerically and invariant sets are observed indirectly in the flow.

Approaches are complementary. Akin to zoom or wide-angle camera lens.

Organisation of flow on chaotic attractors

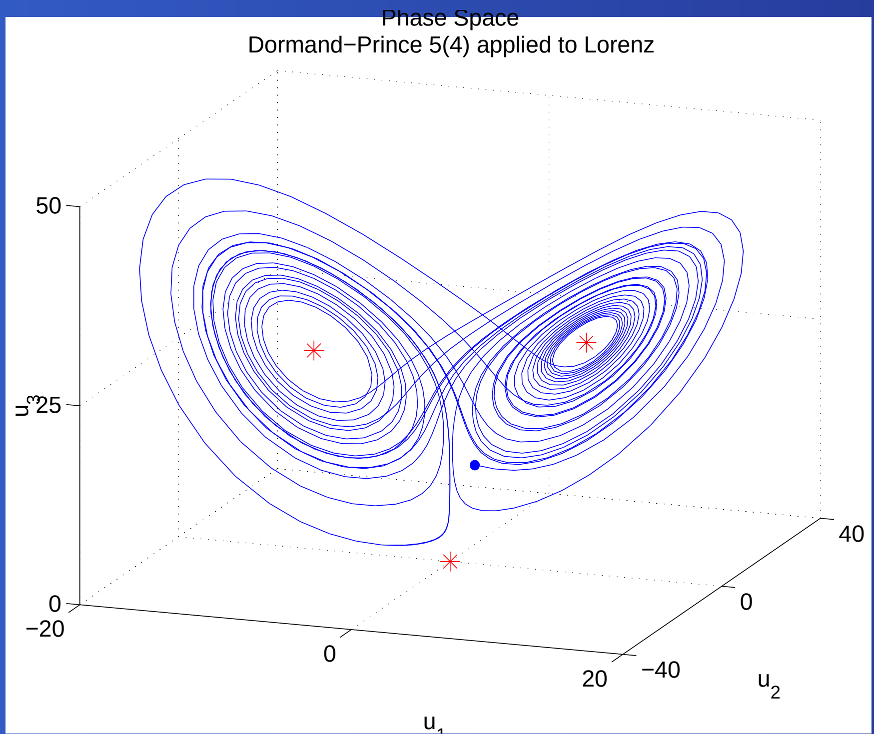


Lorenz Equations

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = rx - y - xz$$
$$\dot{z} = xy - bz$$

Flow in forward time organised by fixed points and unstable manifolds.

Organisation of flow on chaotic attractors



Lorenz Equations

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = rx - y - xz$$
$$\dot{z} = xy - bz$$

Flow in forward time organised by fixed points and unstable manifolds.

1. Seek to reproduce fixed point & local unstable manifold structure
2. Requires (much more than) good solutions near fixed points

The Indirect Approach

Backward Error Analysis: Shadowing

- Numerical trajectory is a poor approximation to solution with given initial value, due to growing error.
- Can we find a perturbed initial condition $\widetilde{u}(0)$ such that the numerical solution with initial condition $u(0)$ stays close to the exact solution with initial condition $\widetilde{u}(0)$ for long or infinite time?
- Infinite time shadowing usually not possible in practical applications.
- W. Hayes will present work on shadowing in Galaxy Simulations.

The Indirect Approach

Backward Error Analysis: Modified Equations

- Rather than perturb initial condition perturb the vector field.
- Can find a hierarchy of vector fields $f_q(u) = f(u) + \mathcal{O}(h^q)$ such that u_n is $\mathcal{O}(h^{q+1})$ close to solution of $\dot{u} = f_q(u)$.
- Usually sequence does not converge, but truncating optimally can find small perturbation of original differential equation which numerical solution solves nearly exactly.

This sort of backward error analysis particularly successful for Hamiltonian systems.

- The symplectic Runge-Kutta methods have numerical solutions which define a symplectic map when applied to a Hamiltonian ODE, which are the exact solution of a Hamiltonian perturbation of the original Hamiltonian system.

The Indirect Approach

Backward Error Analysis: Modified Equations

- Rather than perturb initial condition perturb the vector field.
- Can find a hierarchy of vector fields $f_q(u) = f(u) + \mathcal{O}(h^q)$ such that u_n is $\mathcal{O}(h^{q+1})$ close to solution of $\dot{u} = f_q(u)$.
- Usually sequence does not converge, but truncating optimally can find small perturbation of original differential equation which numerical solution solves nearly exactly.

This sort of backward error analysis particularly successful for Hamiltonian systems.

D. Cottrell will present a backward error analysis of the symplectic Euler method applied to a particle collision model.

Analysis of The Indirect Approach

- Let $S(t) : u(0) \rightarrow u(t)$ be evolution operator for dynamical system $\dot{u} = f(u)$.
- The numerical method defines a map $S_h : u_n \rightarrow u_{n+1}$. Eg for Forward Euler $u_{n+1} = u_n + hf(u_n)$ so

$$S_h u = u + hf(u),$$

- For implicit methods S_h defined implicitly. Eg for Backward Euler $u_{n+1} = u_n + hf(u_{n+1})$ so

$$S_h u = u + hf(S_h u).$$

Analysis of The Indirect Approach

- Let $S(t) : u(0) \rightarrow u(t)$ be evolution operator for dynamical system $\dot{u} = f(u)$.
- The numerical method defines a map $S_h : u_n \rightarrow u_{n+1}$. Eg for Forward Euler $u_{n+1} = u_n + hf(u_n)$ so

$$S_h u = u + hf(u),$$

- So numerical methods define discrete dynamical systems.
- We can use dynamical systems techniques to compare behaviour of the dynamical system defined by $S(t)$ and the one parameter family of dynamical systems defined by S_h, especially their invariant sets.
- This approach well established, many results.....

Parameterized Families of Dynamical Systems

- We treat step-size h as a bifurcation parameter.
- But if original dynamical system has parameter(s) we get multiple bifurcation parameters. Eg we will consider Lorenz equations with r varying.
- Can have spurious bifurcations of co-dimension 1 or higher.

McGill

LaTeX

Fixed Point Bifurcations

- For the forward Euler method $S_h u = u + hf(u)$ so if $f(u) = 0$ then $S_h u = u$. Thus the method preserves all fixed points of dynamical system.
- [Iserles]: all Runge-Kutta and linear multistep methods preserve all fixed points of underlying dynamical system
- For parameterised family of dynamical systems this implies that all fixed point bifurcations are reproduced exactly with exact parameter values.
- However Runge-Kutta methods may admit additional spurious fixed points and fixed point bifurcations.

Spurious Solutions and Stability

Let λ be eigenvalue of Jacobian of f at a fixed point. Then numerical stability in direction of corresponding eigenvector determined by solution of numerical method applied to

$$\dot{u} = \lambda u.$$

For general Runge-Kutta method

$$u_{n+1} = R(h\lambda)u_n, \qquad u_n = \Big\{R(h\lambda)\Big\}^n u_0,$$

For Forward Euler

$$u_{n+1} = u_n + h\lambda u_n, \text{ so } R(z) = 1 + z.$$

Region on which $|R(z)| < 1$ is called stability region.
When this boundary is crossed stability changes and in general bifurcation occurs.

Saddle Point Example

Backward Euler $\equiv$ Forward Euler

Consider simple linear saddle point

$$\dot{u}_1 = -u_1, \qquad \dot{u}_2 = u_2$$

Forward Euler

$$u_1^{n+1} = u_1^n - h u_1^n$$

stability $h < 2$, mono $h < 1$

Backward Euler

$$u_1^{n+1} = u_1^n - h u_1^{n+1}$$

mono stability $\forall h > 0$

Saddle Point Example

Backward Euler $\equiv$ Forward Euler

Consider simple linear saddle point

$$\dot{u}_1 = -u_1, \qquad \dot{u}_2 = u_2$$

Forward Euler

$$u_1^{n+1} = u_1^n - hu_1^n$$

stability $h < 2$, mono $h < 1$

$$u_2^{n+1} = u_2^n + hu_2^n$$

mono instability $\forall h > 0$

Backward Euler

$$u_1^{n+1} = u_1^n - hu_1^{n+1}$$

mono stability $\forall h > 0$

$$u_2^{n+1} = u_2^n + hu_2^{n+1}$$

instability $h < 2$, mono $h < 1$.

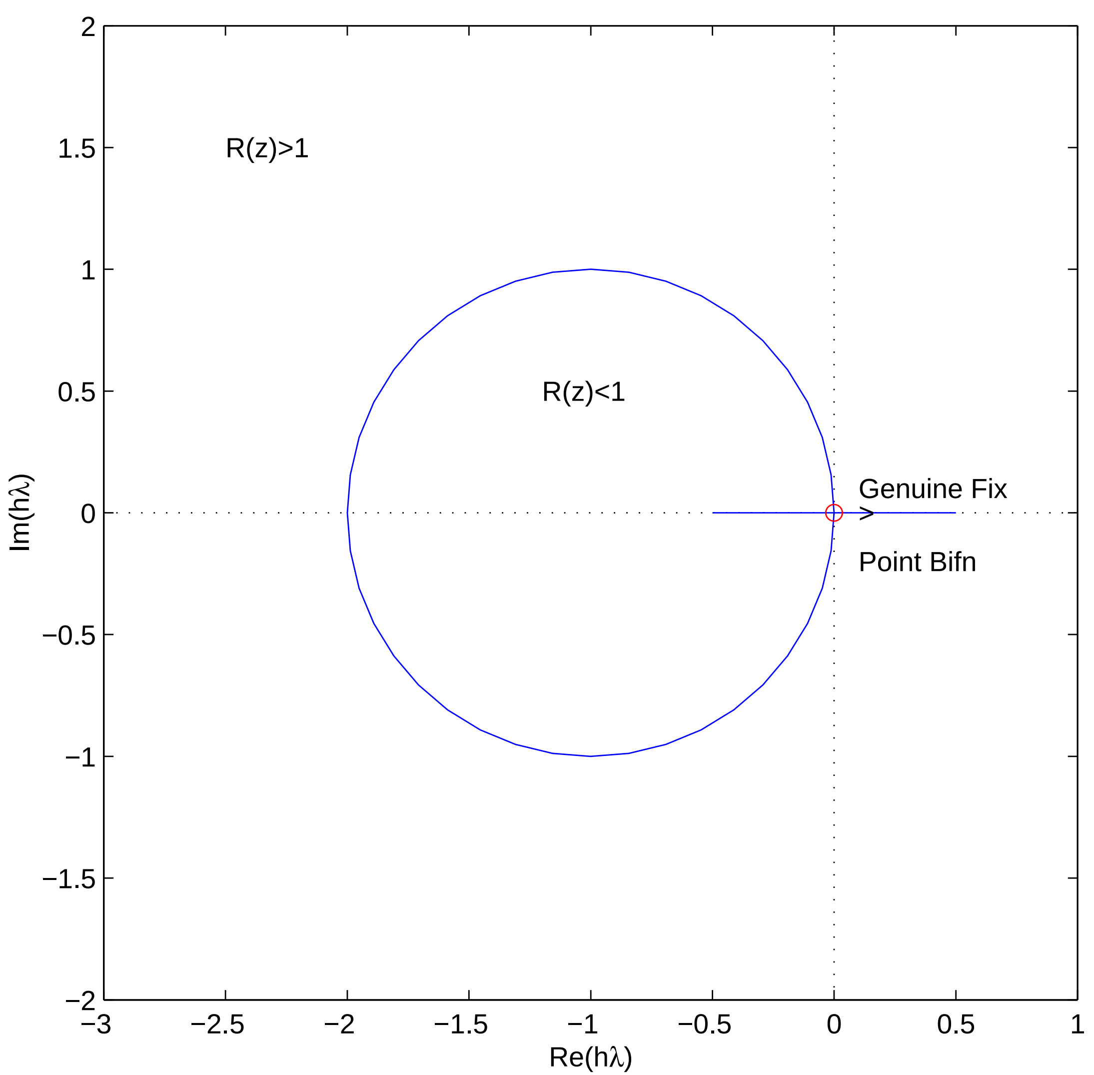
The Indirect Approach

The Problem of Stiffness

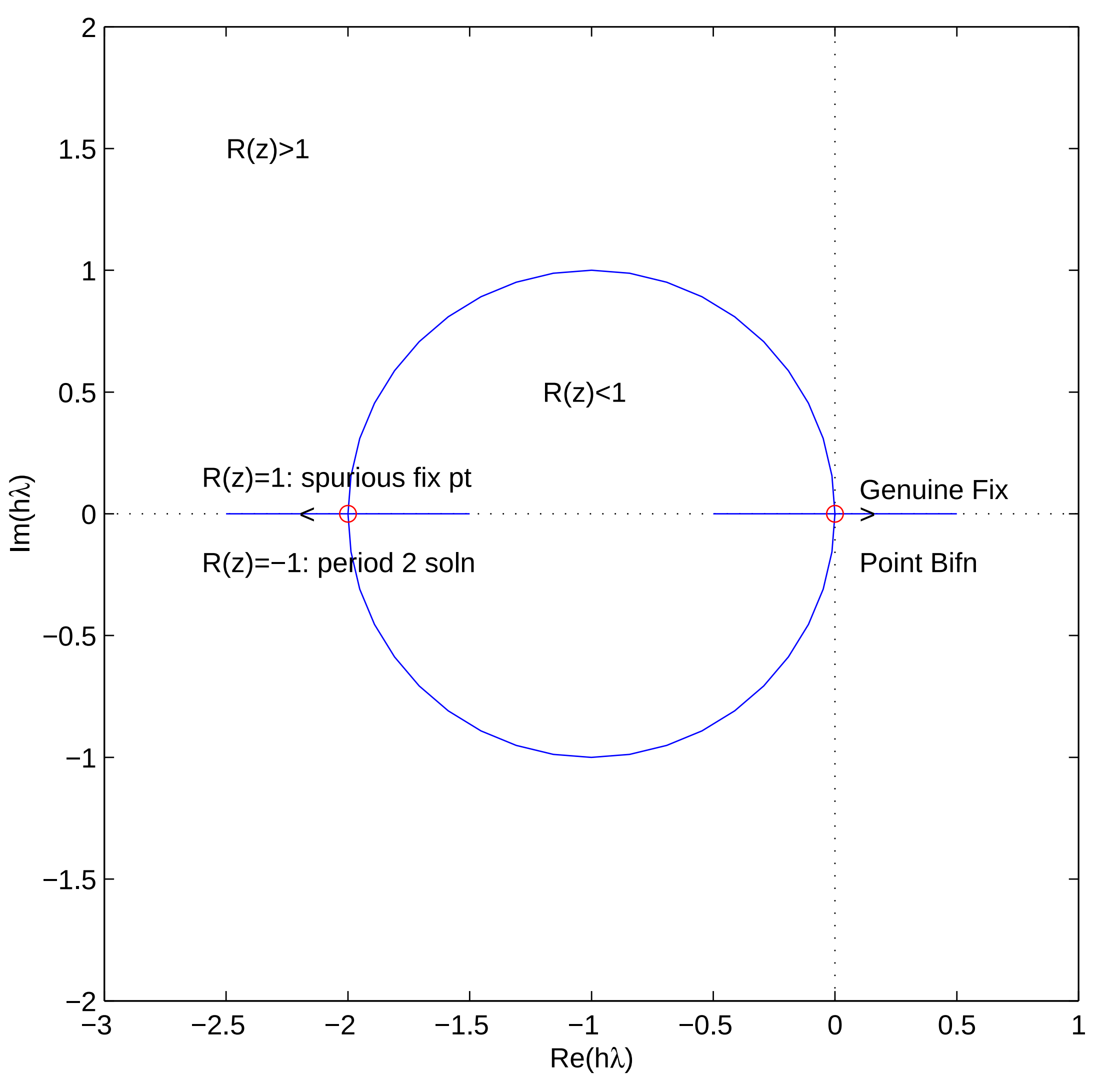
- Spurious invariant limit sets typically bifurcate from linear stability (or instability) limit.
- So keep step-size below stability limit (and instability limit) if possible.
- Large negative eigenvalues present a problem. Require stiff L-stable methods with $R(z) \to 0$ as $|z| \to \infty$ to avoid having to use very small step-sizes. All such methods implicit.
- Stiff methods are not a cure all. Stiffness theory typically ignores saddle points and problem of preservation of instability.
- Problem of stiffness detection important: R. Spiteri will consider this.

Spurious Solutions and Stability Limit

Forward Euler Method

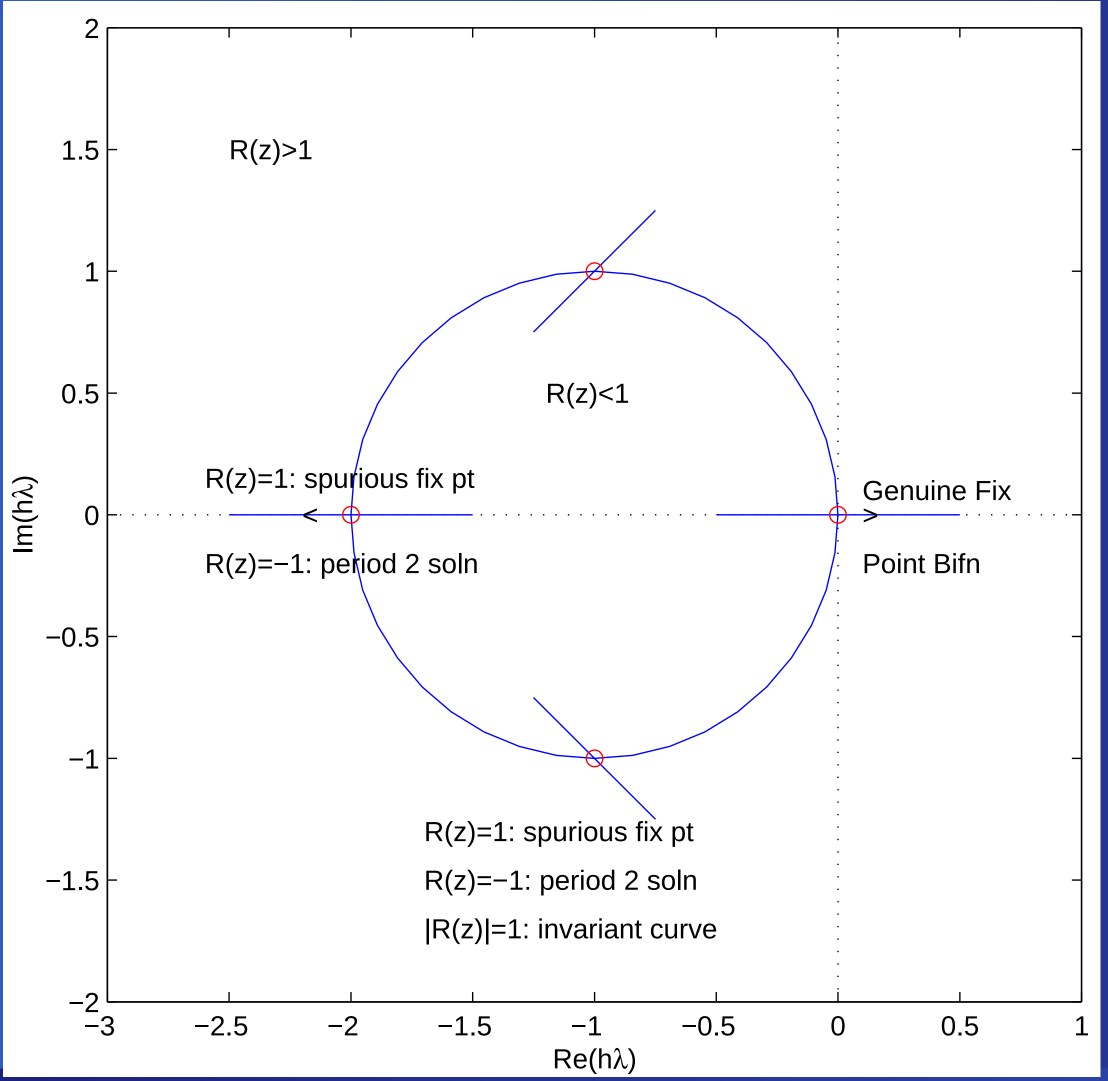


Spurious Solutions and Stability Limit
Forward Euler Method



Spurious Solutions and Stability Limit

Forward Euler Method

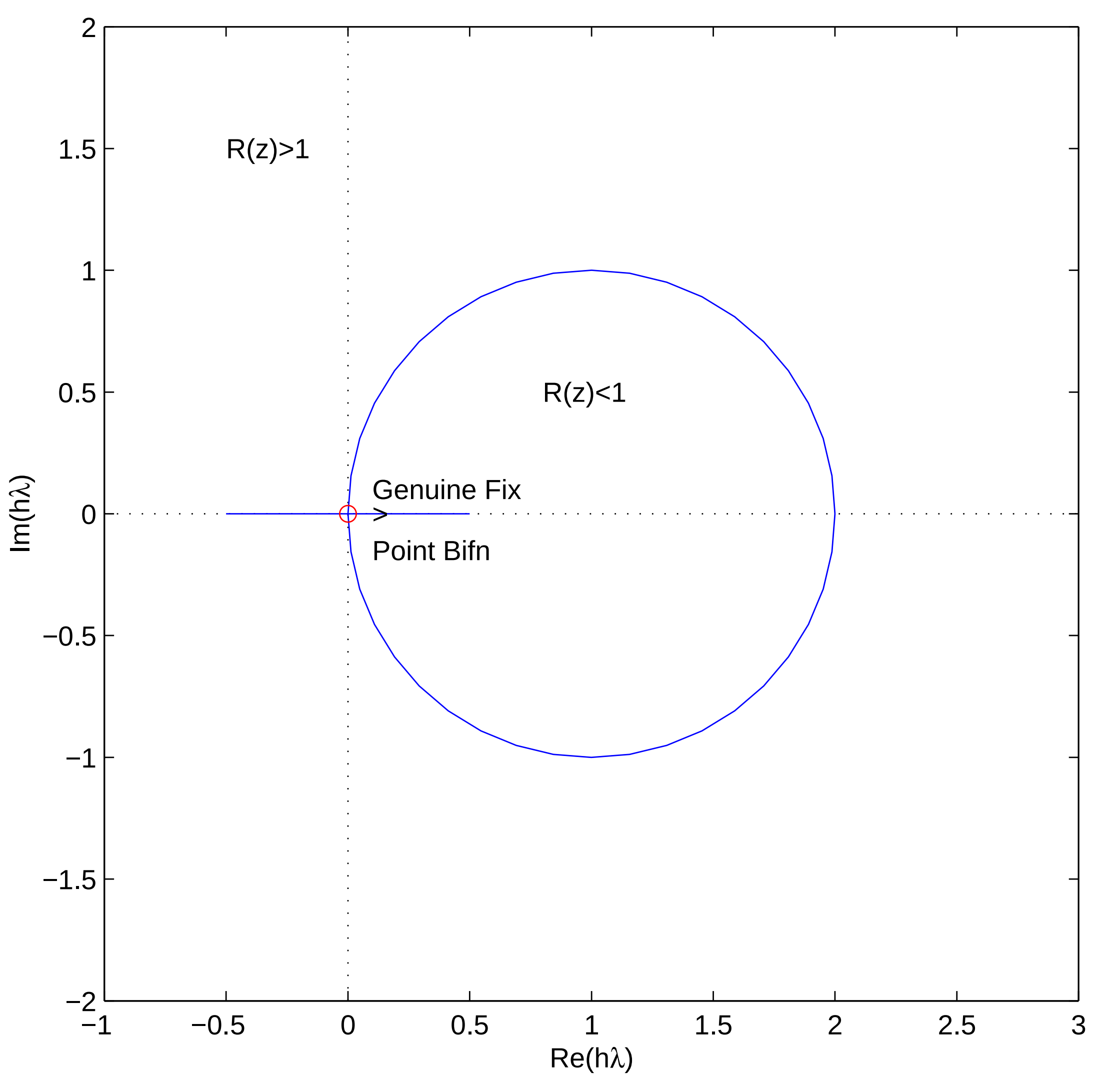


McGill

LATEX

Spurious Solutions and Stability Limit

Backward Euler Method

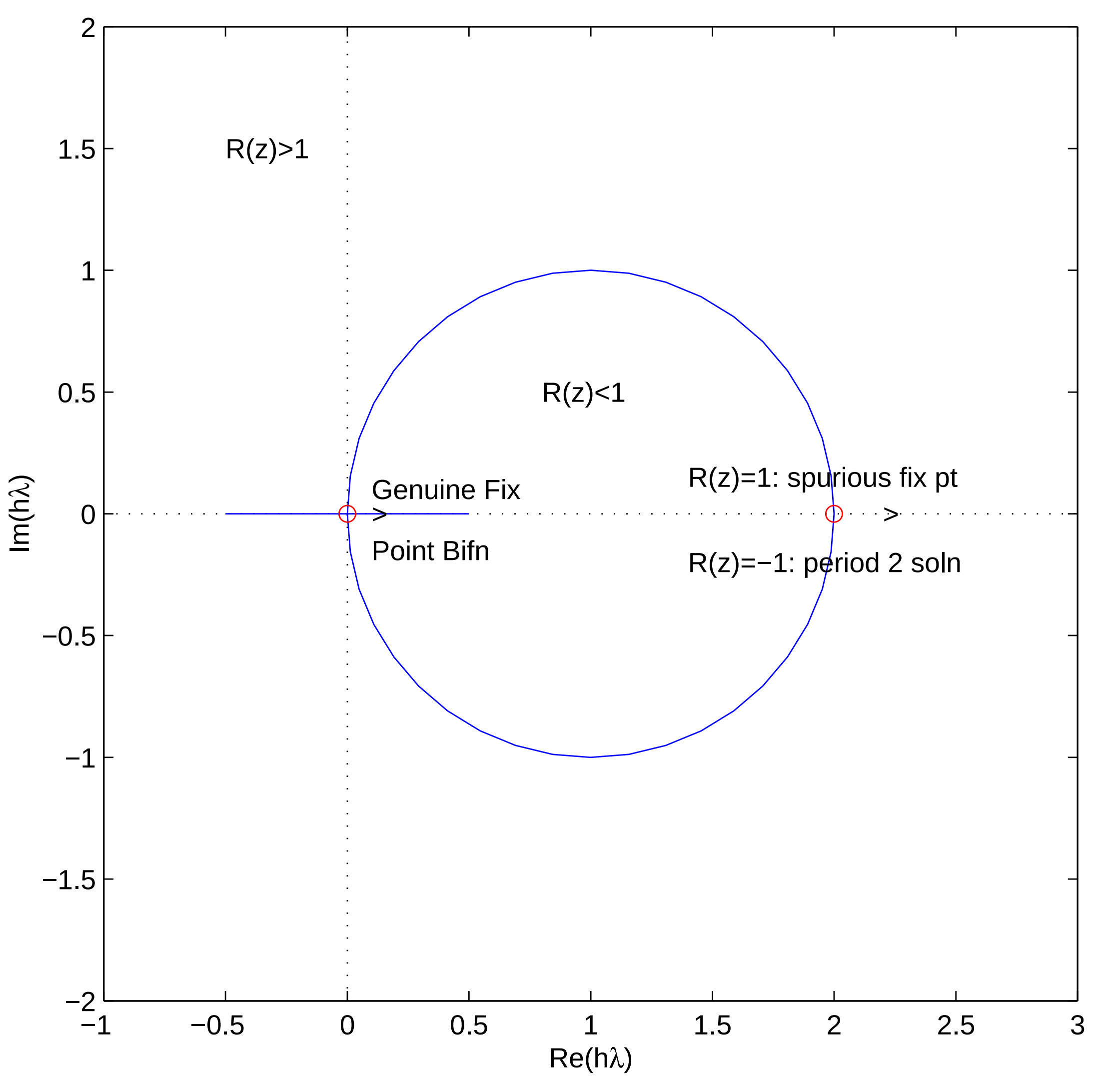


McGill

LaTeX

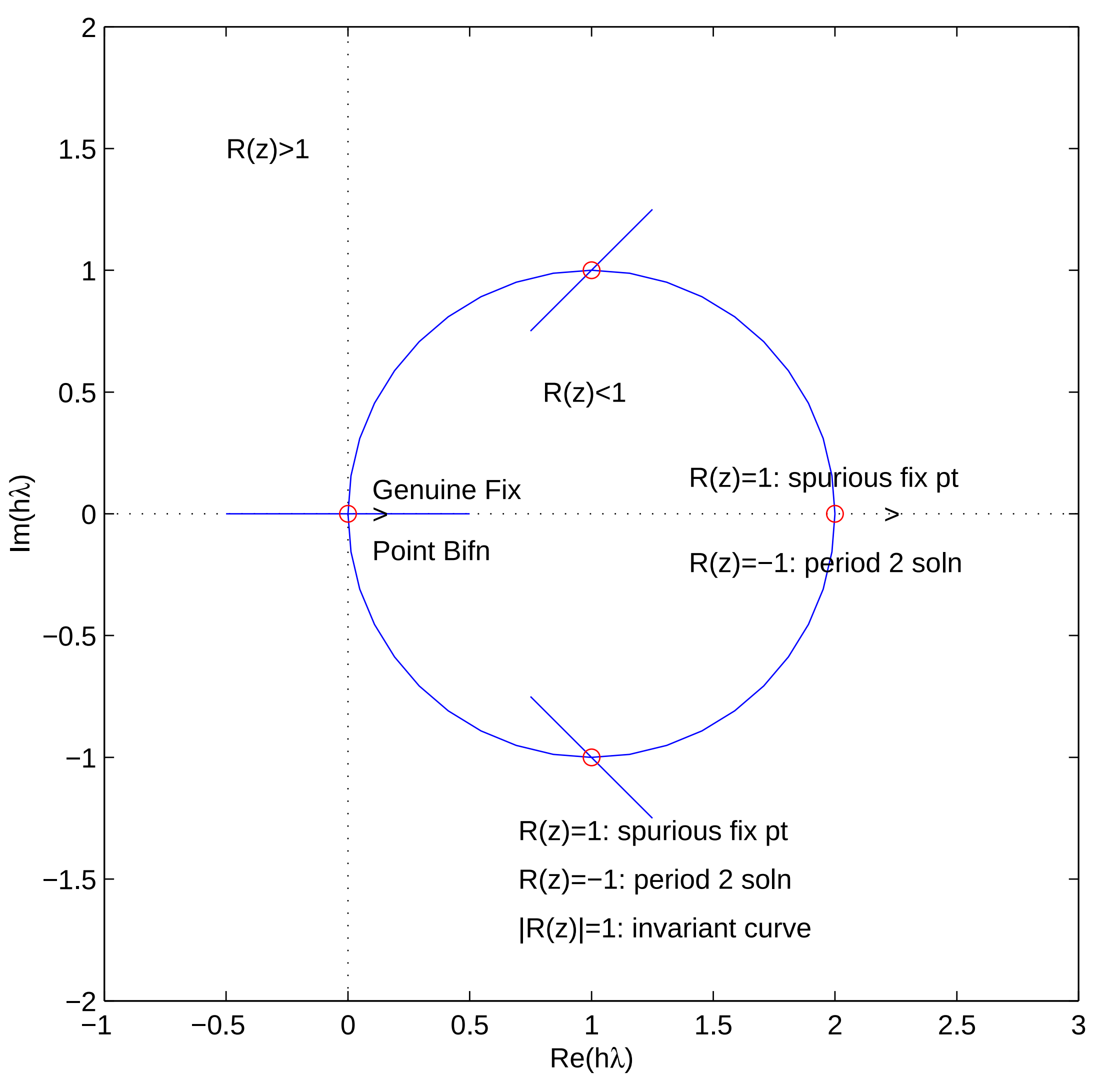
Spurious Solutions and Stability Limit

Backward Euler Method

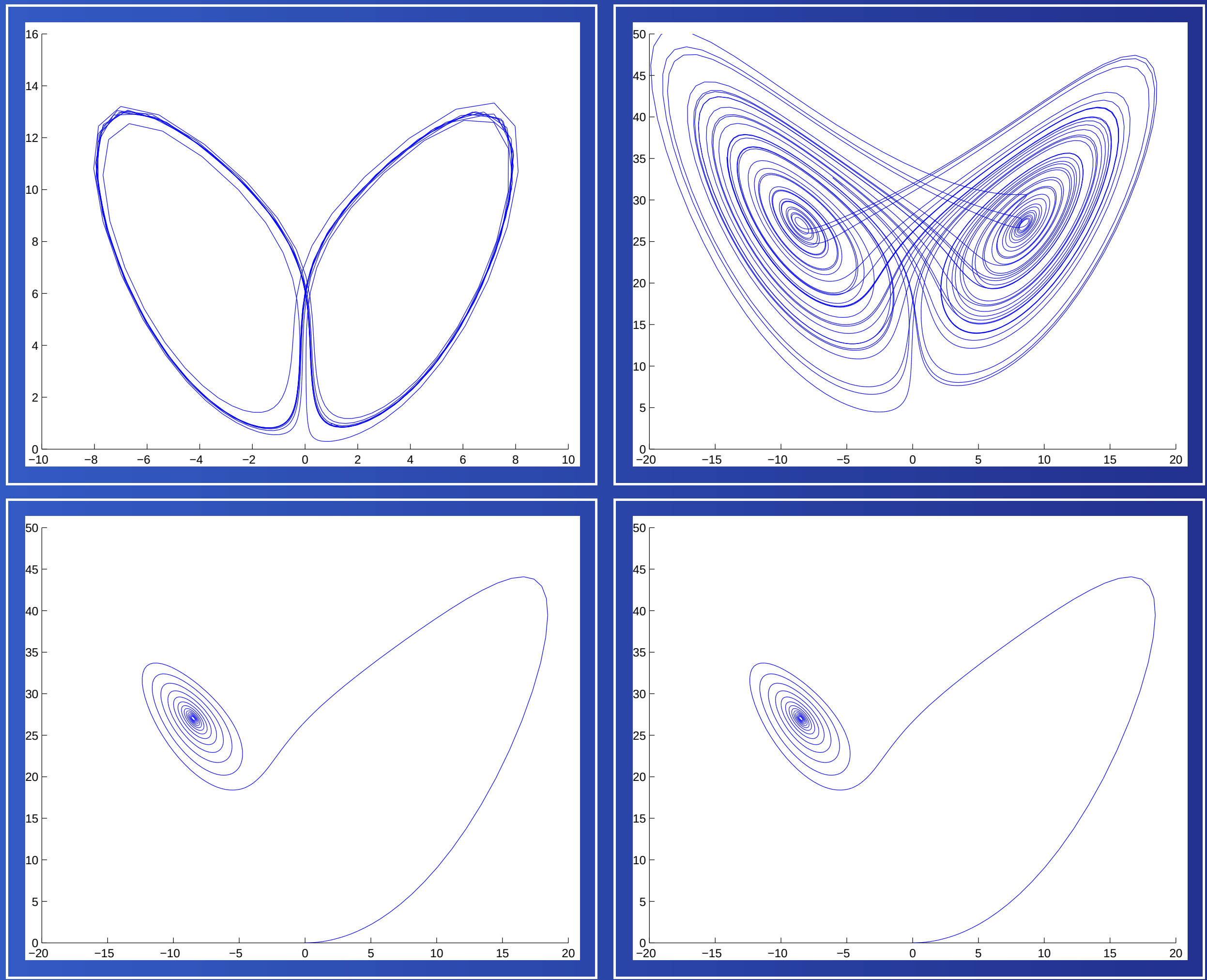


Spurious Solutions and Stability Limit

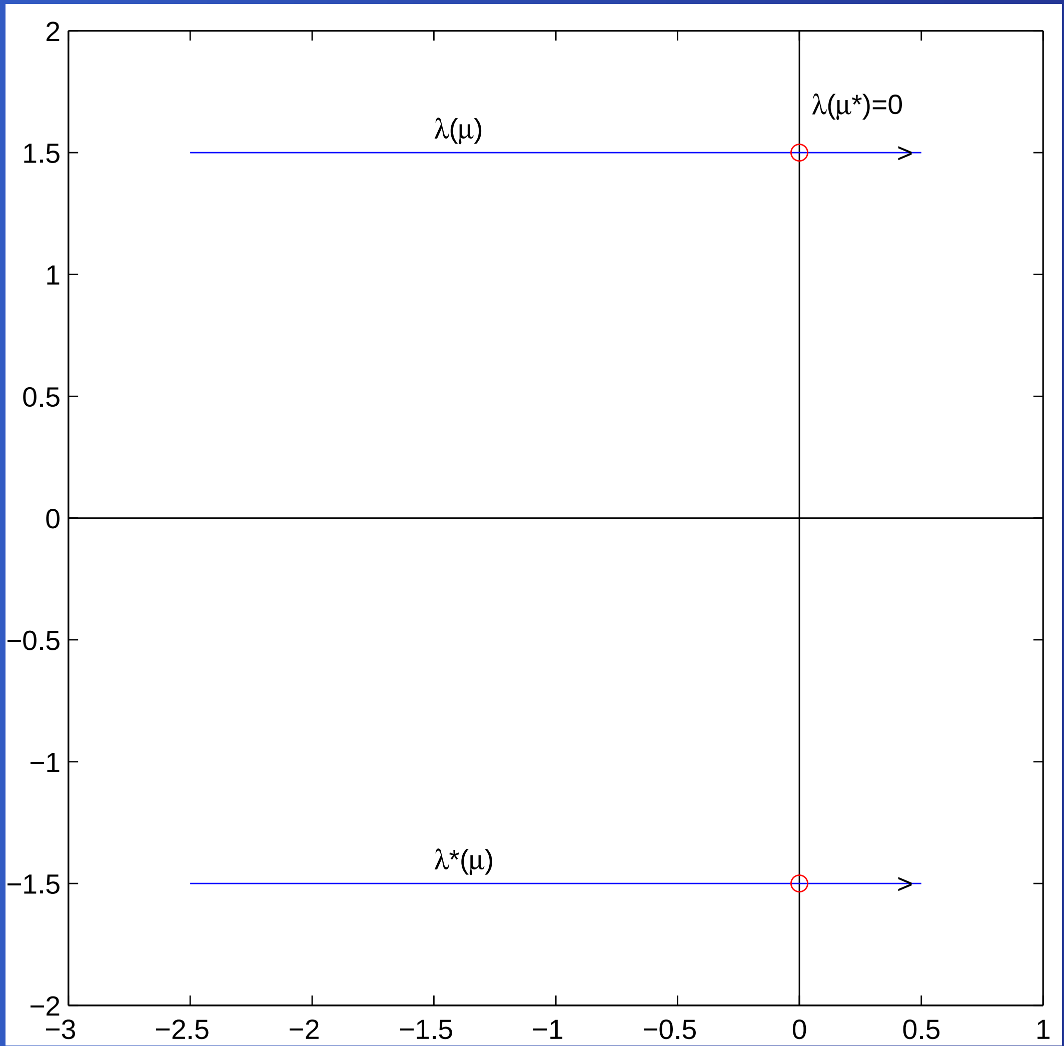
Backward Euler Method



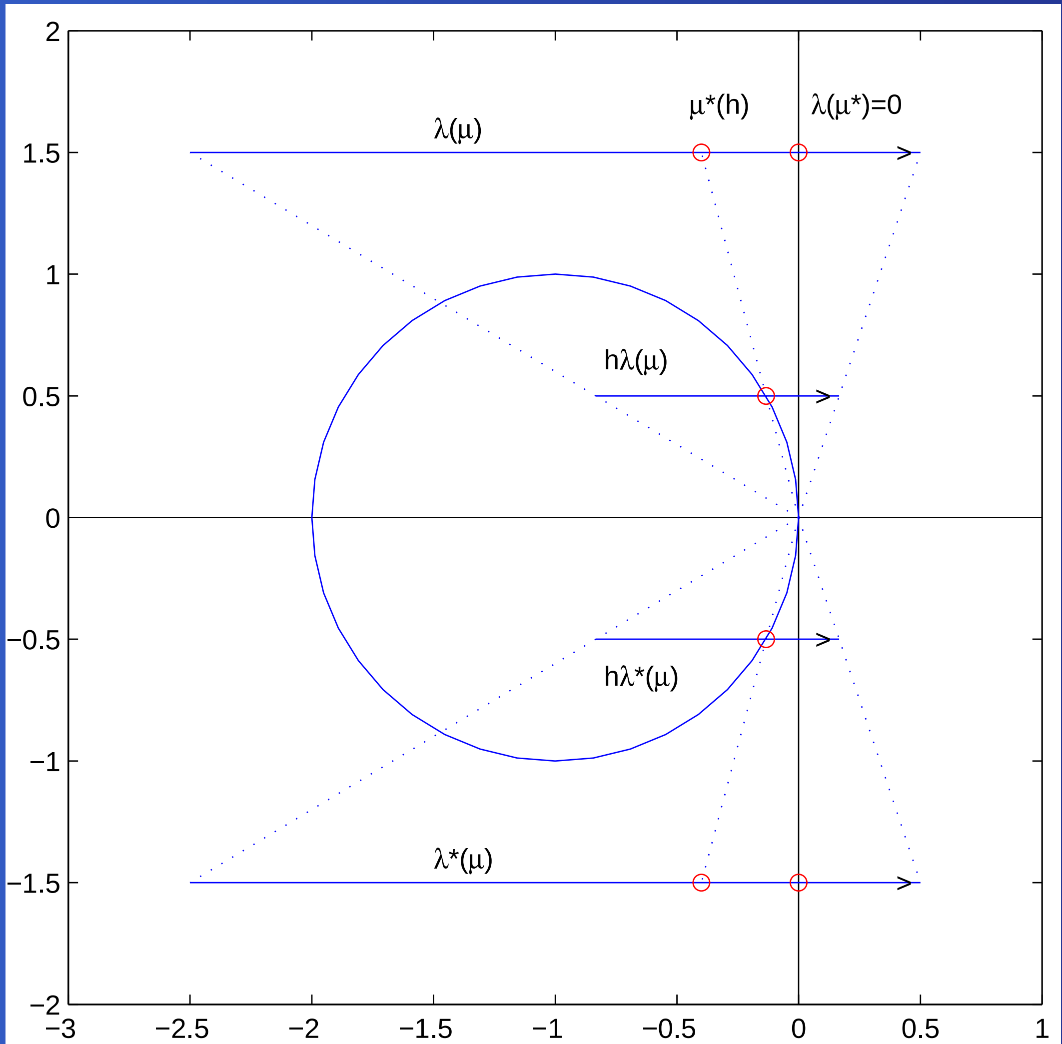
Simulating Systems with Hopf Bifurcations



Perturbation of Hopf Bifurcation

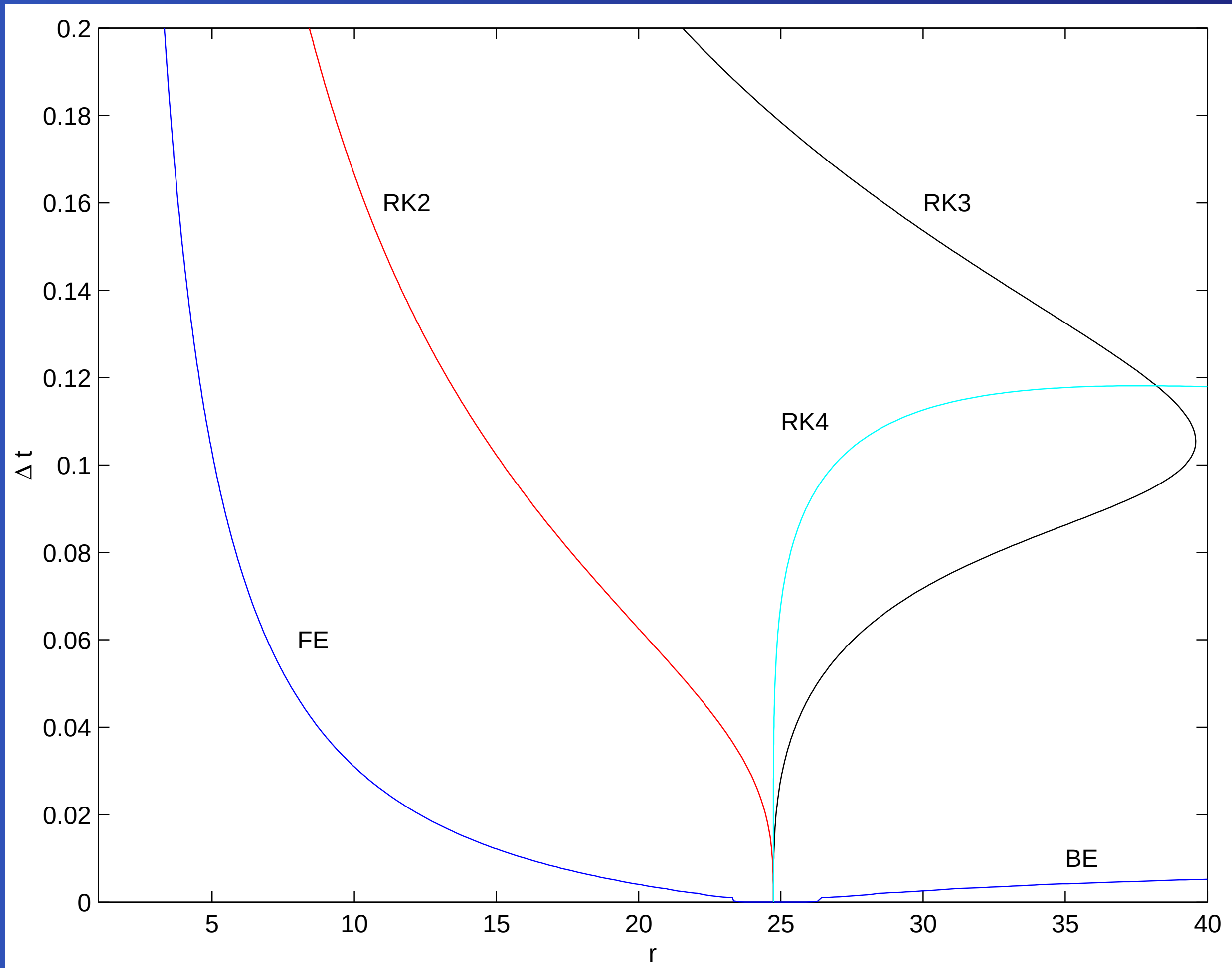


Perturbation of Hopf Bifurcation

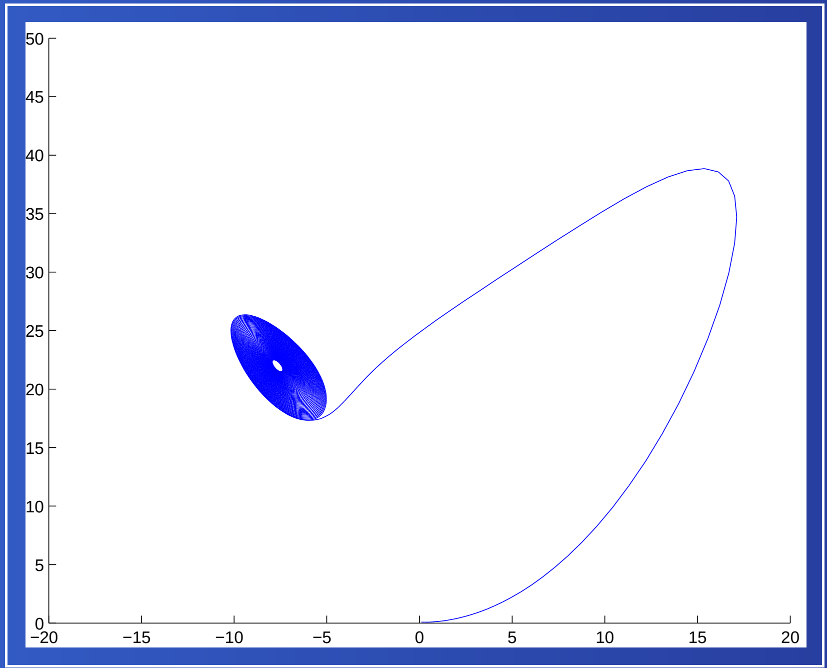


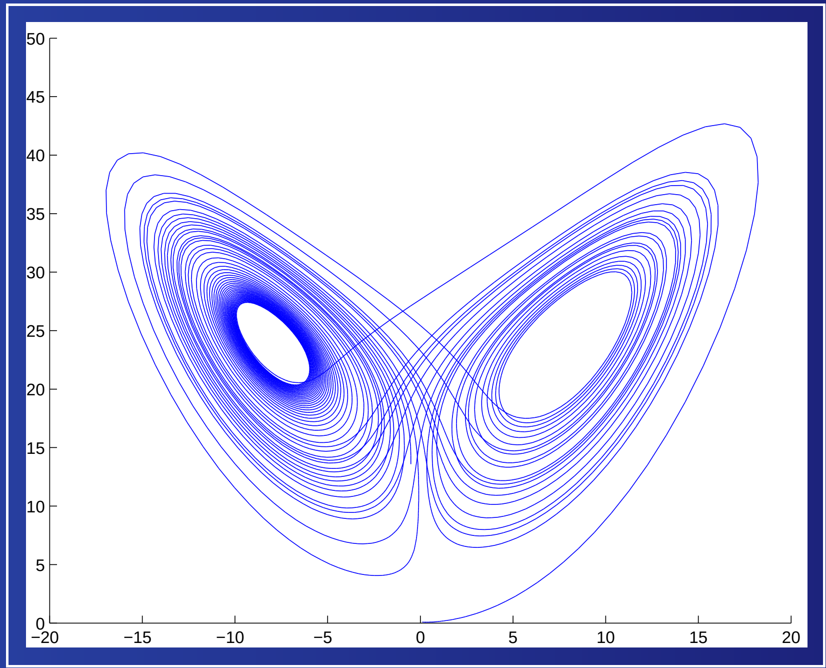
Lorenz Hopf Bifurcation

Location in r-h parameter space



Lorenz Hopf Bifurcation
RK2 simulations





(i) $r = 23$

(ii) $r = 25$

Hopf Bifurcation Theorem

Theorem For a Runge-Kutta method of linear order $2p$ for $p \geqslant 0$

$$\mu^* - \mu_h^* = \mathcal{O}(h^{2p+1})$$

where μ^* is parameter value of Hopf bifurcation of dynamical system and μ_h^* is perturbed bifurcation for numerical method with step-size h.

[Humphries & Christodoulou, in prep]

The Direct Approach
Finding Invariant Sets

How do we find periodic orbits implied by Hopf bifurcation theorem, and other invariant sets ?

- Set up equations defining required invariant set and solve.
- For $\dot{u} = f(u)$ fixed points given by $f(u) = 0$.
- Periodic orbits, heteroclinic connections etc not so easy.
- Strange attractor: not possible.

The Direct Approach
Periodic Orbits

- Subject of Doedel's talk.

Need to solve for orbit and period T. Let $v(t) = u(tT) \in \mathbb{R}^p$ then

$$\dot{v} = Tf(v), \quad t \in [0,1]$$

$$v(0) = v(1).$$

- One more unknown than equations. One parameter family of solutions, starting anywhere on orbit.
- Add phase condition to fix particular orbit. Use Doedel's AUTO code to solve.
- More difficult with Hamiltonian systems with families of periodic orbits. Need to break Hamiltonian structure.
- Parameter continuation, bifurcation detection all possible.

The Indirect Approach
Variable Time-Stepping

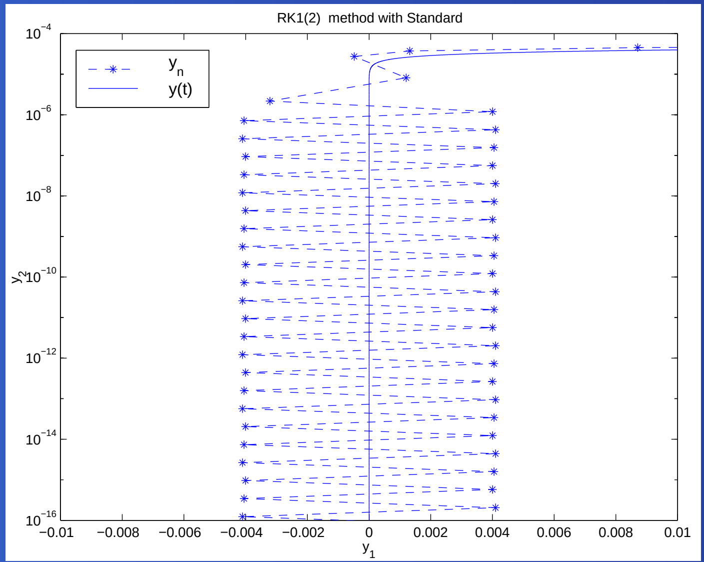
Now allow step-size h_n to vary step to step, for efficiency.

- Dynamical systems techniques still appropriate but harder
- Evolution $(u_n, h_n) \to (u_{n+1}, h_{n+1})$ includes step-size so is different dimension to underlying dynamical system
- Map is discontinuous due to step-size rejections.
- Traditional time-stepping based on local error control
- Analysis of this relied on tolerance proportionality assumption which is false.
- [Stuart],[Lamba & Stuart]: rigorous finite time convergence for some cases.

Stable Fixed Point Example

Consider the method RK1(2) applied to the linear system

$$\dot{u} = \begin{bmatrix} -5 & 0 \\ 0 & -1 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad u(0) = [1, 10^{-4}]^T.$$



$(0,0)$ – stable fixed point.

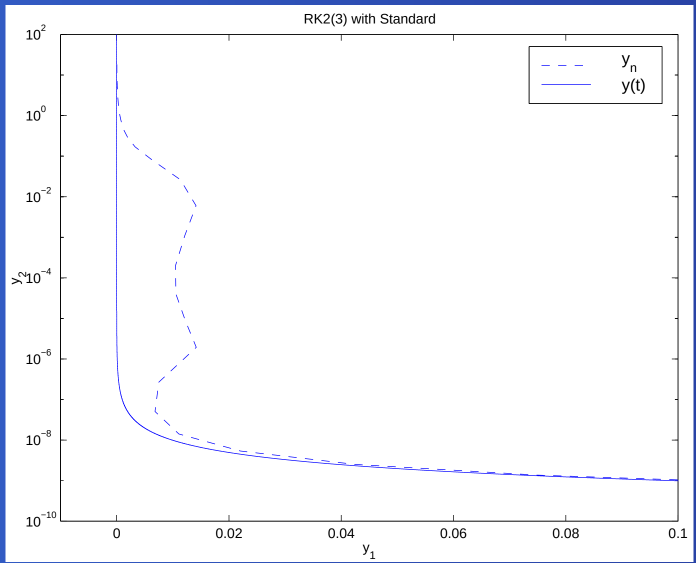
For this method, the numerical solution gives persistent spurious oscillations and the y_1 component has $\mathcal{O}(\tau)$ oscillation about the fixed point.

RK2(3) & RK4(5) Saddle Point Example

$$\dot{u} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} u, \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad u(0) = \left(0.99, 10^{-10}\right)^T.$$

RK2(3) & RK4(5) Saddle Point Example

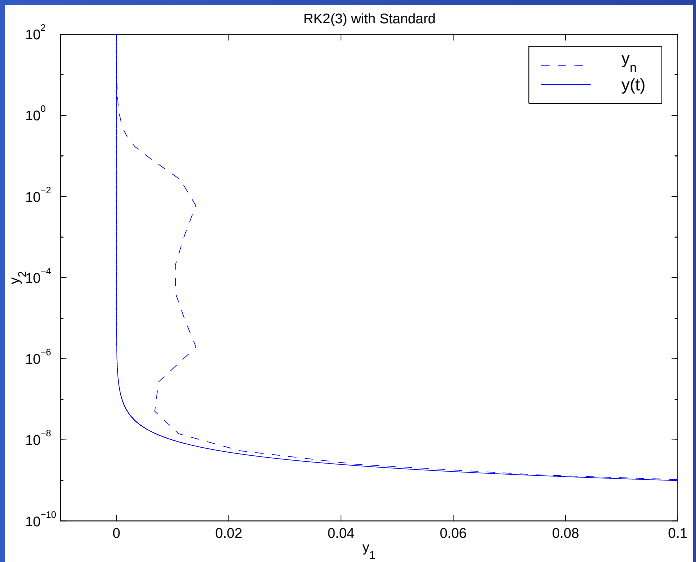
$$\dot{u} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} u, \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad u(0) = (0.99, 10^{-10})^T.$$



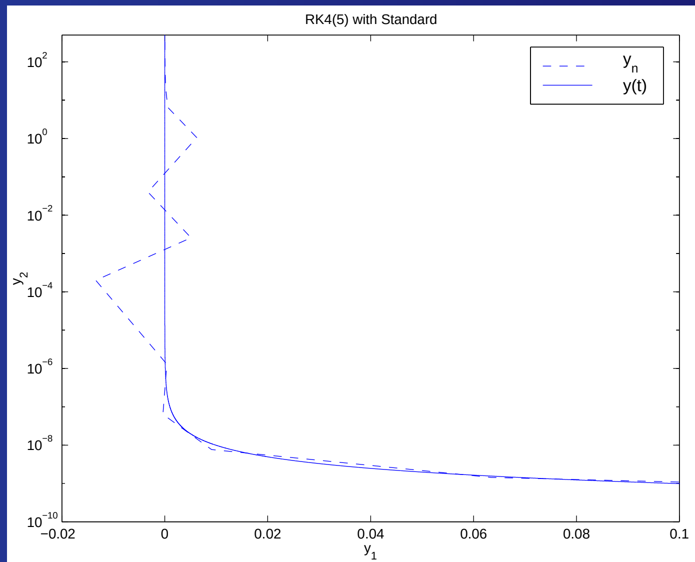
RK2(3) numerical solution does not pass close to fixed point or the local unstable manifold.

RK2(3) & RK4(5) Saddle Point Example

$$\dot{u} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} u, \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad u(0) = (0.99, 10^{-10})^T.$$



RK2(3) numerical solution does not pass close to fixed point or the local unstable manifold.



RK4(5) has spurious oscillations about the unstable manifold. Numerical solution can ultimately end up either side of the unstable manifold.

Local error approximation

With user-defined tolerance, $0 < \tau \ll 1$, step h_n chosen by

$$\|E(u_n, h_n)\| \leqslant \tau, \quad \text{where} \quad E(u_n, h_n) = \frac{1}{h_n^\rho}(u_{n+1} - \widetilde{u}_{n+1}).$$

with $\rho = 0$ error per step (EPS) or $\rho = 1$ error per unit step (EPUS).

McGill

LaTeX

Local error approximation

With user-defined tolerance, $0 < \tau \ll 1$, step h_n chosen by

$$\|E(u_n, h_n)\| \leqslant \tau, \quad \text{where} \quad E(u_n, h_n) = \frac{1}{h_n^\rho}(u_{n+1} - \widetilde{u}_{n+1}).$$

with $\rho = 0$ error per step (EPS) or $\rho = 1$ error per unit step (EPUS).

Algorithm attempts to ensure

$$E(u_n, h_n) \approx \gamma\tau, \quad \gamma \in (0, 1) \text{ safety factor}$$

Leads to trouble near fixed points since $f(u_n) = 0$ implies

$$E(u_n, h_n) = 0.$$

McGill

LaTeX

Variable Time-stepping

Summary

- Standard algorithm must fail near (un)stable manifolds; seeks to control error over infinite time interval
- [Hall 1986], [Hall & Higham 1988] showed step-size driven to linear stability limit near fixed point
- Tolerance proportionality fails; maximum step-size independent of tolerance
- Stiff Implicit methods unsuitable; typically severe step-size restrictions required to preserve the genuine instabilities which drive chaos
- No better time-stepping algorithm. Gustaffson's PI-controller based algorithm addresses stable fixed point problem, ensures step-size driven to stability limit in stable manner; does not address underlying dynamics.

We need a Different Time-Stepping Approach

- Control theory step-size controllers [Gustaffson & Soderlind] address stability of step-size sequence, but use same error control so dynamics not resolved
- M. Gander will present an idea based on a moving mesh formulation.
- We consider a alternative approach devising a phase space based error control which can be applied in consort with the traditional error control.

Phase Space (PS_θ) Error Control

- We demand at each step the phase space (PS_θ) error control

$$\begin{aligned} \|u_{n+1} - u_n - h_n[(1-\theta)f(u_n) + \theta f(u_{n+1})]\| \\ \leq \quad \varphi h_n \|(1-\theta)f(u_n) + \theta f(u_{n+1})\|, \quad \varphi \in (0,1). \end{aligned}$$

Phase Space (PS$_\theta$) Error Control

- We demand at each step the phase space (PS$_\theta$) error control

$$\begin{aligned} &\|u_{n+1} - u_n - h_n[(1-\theta)f(u_n) + \theta f(u_{n+1})]\| \\ &\qquad \leq \varphi h_n \|(1-\theta)f(u_n) + \theta f(u_{n+1})\|, \quad \varphi \in (0,1). \end{aligned}$$

- $h_n\|(1-\theta)f(u_n) + \theta f(u_{n+1})\|$ is approximation to arc length evolved over step.

McGill

LaTeX

Phase Space (PS_θ) Error Control

- We demand at each step the phase space (PS_θ) error control

$$\begin{aligned} \|u_{n+1} - u_n - h_n[(1-\theta)f(u_n) + \theta f(u_{n+1})]\| \\ \leq \quad \varphi h_n \|(1-\theta)f(u_n) + \theta f(u_{n+1})\|, \quad \varphi \in (0,1). \end{aligned}$$

- $h_n\|(1-\theta)f(u_n) + \theta f(u_{n+1})\|$ is approximation to arc length evolved over step.
- So PS_θ error control bounds an approximation to local error by a fraction φ of an approximation to solution arc length in phase space. So is a phase space error control.

Phase Space (PS_θ) Error Control

- We demand at each step the phase space (PS_θ) error control

$$\begin{aligned}\|u_{n+1} - u_n - h_n[(1-\theta)f(u_n) + \theta f(u_{n+1})]\| \\ \leq \quad \varphi h_n \|(1-\theta)f(u_n) + \theta f(u_{n+1})\|, \quad \varphi \in (0,1).\end{aligned}$$

- $h_n\|(1-\theta)f(u_n) + \theta f(u_{n+1})\|$ is approximation to arc length evolved over step.
- So PS_θ error control bounds an approximation to local error by a fraction φ of an approximation to solution arc length in phase space. So is a phase space error control.
- Will show it also acts as a stability control.

McGill

LaTeX

Phase Space (PS_θ) Error Control

- We demand at each step the phase space (PS_θ) error control

$$\|u_{n+1} - u_n - h_n[(1-\theta)f(u_n) + \theta f(u_{n+1})]\| \\ \leq \varphi h_n \|(1-\theta)f(u_n) + \theta f(u_{n+1})\|, \quad \varphi \in (0,1).$$

- $h_n\|(1-\theta)f(u_n) + \theta f(u_{n+1})\|$ is approximation to arc length evolved over step.
- So PS_θ error control bounds an approximation to local error by a fraction φ of an approximation to solution arc length in phase space. So is a phase space error control.
- Will show it also acts as a stability control.
- Will combine this error control with standard error control; and demand both are satisfied at every step.

Key Properties of Phase Space Error Control

- Negligible additional computation is needed;

Key Properties of Phase Space Error Control

- Negligible additional computation is needed;
- Away from fixed points the standard error control is sufficient to ensure that the PS_θ condition is satisfied.

Key Properties of Phase Space Error Control

- Negligible additional computation is needed;
- Away from fixed points the standard error control is sufficient to ensure that the PS_θ condition is satisfied.
- Prevents spurious fixed points;

Key Properties of Phase Space Error Control

- Negligible additional computation is needed;
- Away from fixed points the standard error control is sufficient to ensure that the PS_θ condition is satisfied.
- Prevents spurious fixed points;
- Forces convergence to stable fixed points;

Key Properties of Phase Space Error Control

- Negligible additional computation is needed;
- Away from fixed points the standard error control is sufficient to ensure that the PS_θ condition is satisfied.
- Prevents spurious fixed points;
- Forces convergence to stable fixed points;
- Gives stable step-size sequence with suitable step-size selection mechanism

Key Properties of Phase Space Error Control

- Negligible additional computation is needed;
- Away from fixed points the standard error control is sufficient to ensure that the PS_θ condition is satisfied.
- Prevents spurious fixed points;
- Forces convergence to stable fixed points;
- Gives stable step-size sequence with suitable step-size selection mechanism
- Good behaviour near saddle points

Key Properties of Phase Space Error Control

- Negligible additional computation is needed;
- Away from fixed points the standard error control is sufficient to ensure that the PS_θ condition is satisfied.
- Prevents spurious fixed points;
- Forces convergence to stable fixed points;
- Gives stable step-size sequence with suitable step-size selection mechanism
- Good behaviour near saddle points
- Satisfies tolerance proportionality condition which implies most of above.

McGill

LaTeX

Properties

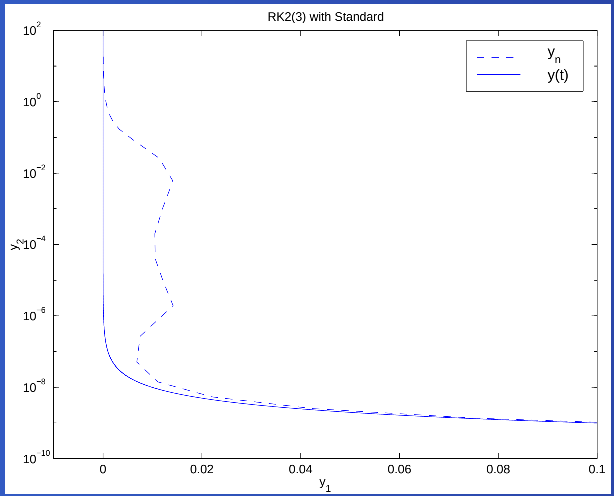
Non-stiff hyperbolic fixed point for φ suff small

- Stable: numerical convergence to fixed point, typical solutions tangential to slowest direction
- Saddle: numerical manifolds exist tangential to exact manifolds at fixed point

Stiff hyperbolic fixed point for φ suff small indep of stiffness

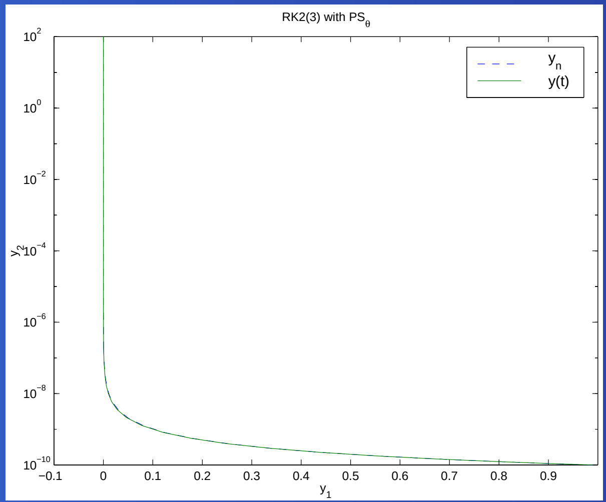
- Stable: numerical convergence to fixed point, max angle between typical solutions and slowest direction decreases with φ
- Saddle: orbits entering close to stable manifold exit close to unstable manifold at max angle which decreases with φ

Saddle Point Example (Revisited)



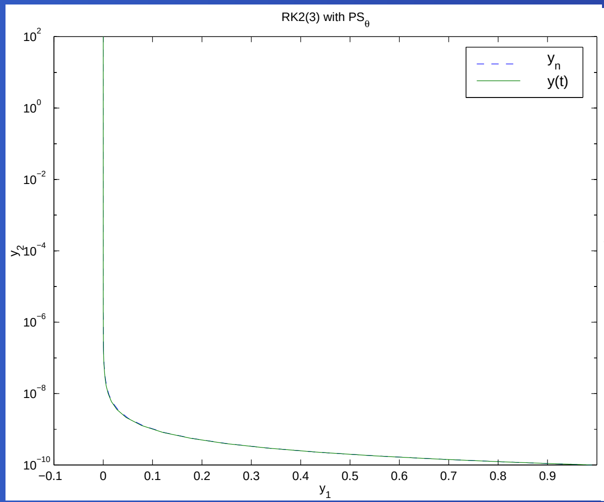
Recall solution with RK2(3) standard algorithm

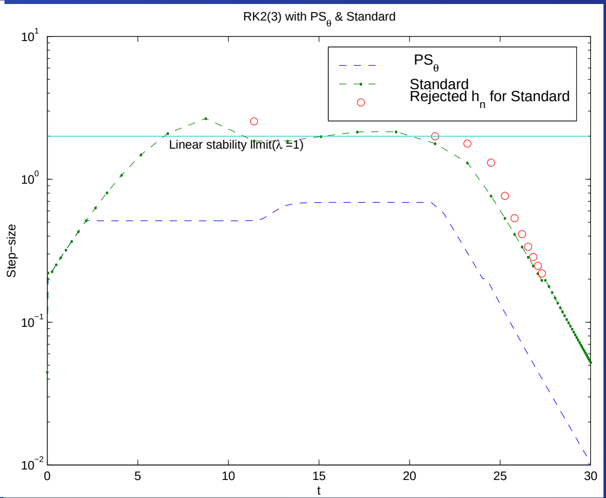
Saddle Point Example (Revisited)



With PS_θ spurious oscillation is removed

Saddle Point Example (Revisited)



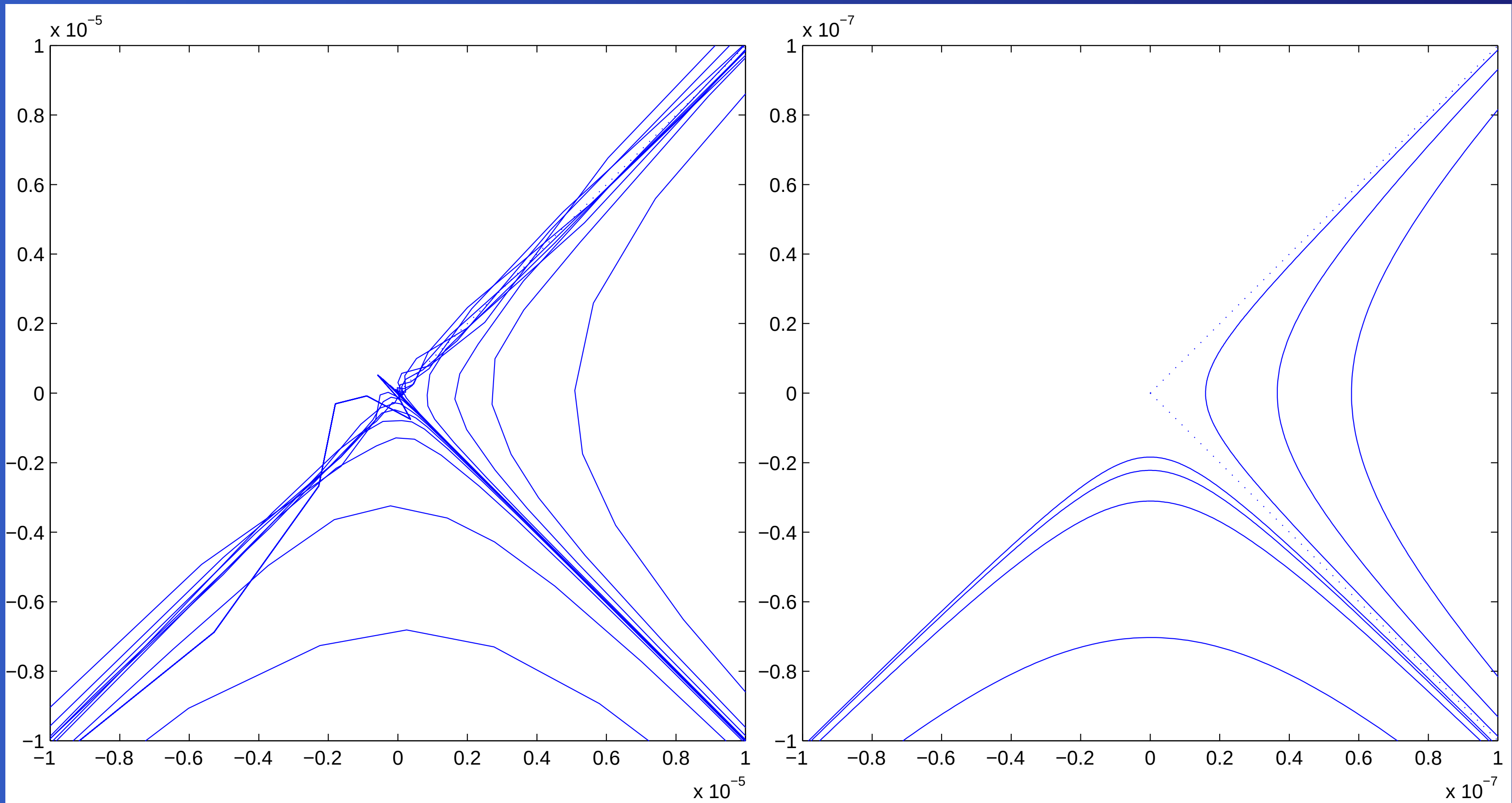


With PS$_\theta$ spurious oscillation is removed
Step-size is kept below stability limit.
Step-sizes bounded near fixed point.
PS$_\theta$ only determines step-size near fixed point.

McGill

LaTeX

Nonlinear Saddle Point Example



- ode45.m applied to problem from Hale & Kocak

McGill

LaTeX

Summary

- Numerical methods have been used to simulate dynamical systems for a long time.
- To give meaning to these solutions its fruitful to use dynamical systems techniques to study these numerical solutions.
- There are many unresolved issues especially interaction of stiffness and dynamics and adaptive time-stepping
- Much current research is concerned with similar issues applied to DAEs, Stochastic DEs, Functional Differential Equations.