Surprising Computations

Uri Ascher

Department of Computer Science
University of British Columbia
ascher@cs.ubc.ca
www.cs.ubc.ca/∼ascher/

Surprising computations

- Numerical simulations often involve sophisticated algorithms for challenging problems. The process of deriving such algorithms as well as showing that they are robust, stable, accurate and efficient, usually involves a lot of insight and subtle work, but often without great surprises: this does not make such work trivial or unimportant.
- In the course of preparing the text I did nonetheless bump several times into method derivations and computations that surprised me. Some such surprises are briefly described here. It's also an opportunity to consider, or refer to, a bunch of case studies.
- The ones about Hamiltonian systems, splitting NLS and WENO are further developed in my paper "Surprising computations" (2012), please see my home page.

List of surprises

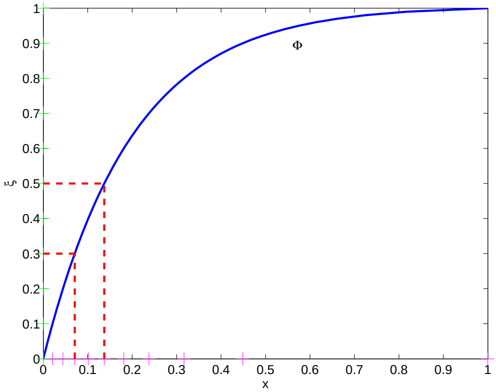
- Quadtrees and Octrees (Chapter 3)
- Cloth simulation (Chapter 2)
- Matlab's ode45 for Hamiltonian systems (Chapters 6 & 2)
- KdV instability (Chapters 5 & 7)
- Splitting cubic NLS (Chapter 9)
- Artificial boundary waves (Chapter 8)
- ENO, WENO and SSP (Chapter 10)
- MEMS device (Chapter 11)

Outline

- Quadtrees and Octrees (Chapter 3)
- Cloth simulation
- Matlab's ode45 for Hamiltonian systems
- KdV instability
- Splitting cubic NLS
- Artificial boundary waves
- ENO, WENO and SSP
- MEMS device

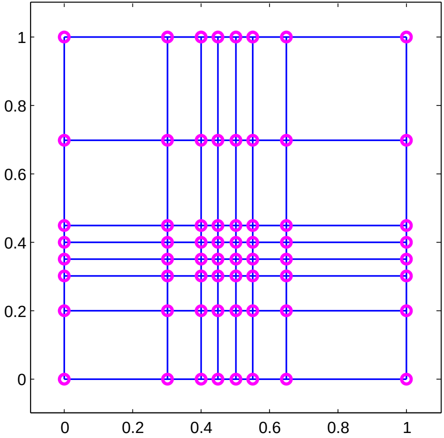
Nonuniform mesh in 1D

Handling a nonuniform mesh is relatively straightforward in 1D



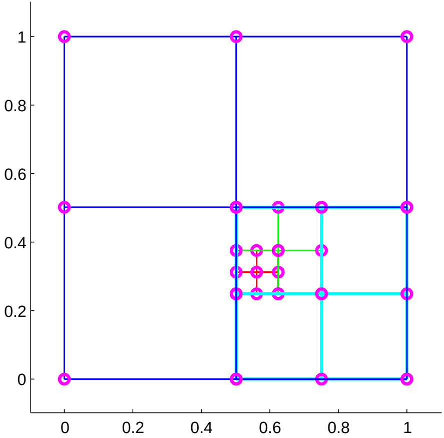
Nonuniform mesh in 2D

Significant additional difficulties in several space dimensions, even on a square domain



Quadtrees

Quadtree mesh (insisting on not using finite elements) can localize refinement in 2D, Octree mesh likewise in 3D



Example: Poisson problem

- $$\min_u I(u) = \int_\Omega \left[|\nabla u|^2 - 2uq \right] \, dxdy,$$

  on the unit square with BC $u|_{\partial\Omega} = 0$.
- Euler-Lagrange necessary and sufficient

$$\begin{aligned} -\Delta u &= q, \\ u|_{\partial\Omega} &= 0. \end{aligned}$$

- Choose $q(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$, then $u(x, y) = \sin(\pi x) \sin(\pi y)$ smooth.

Discretize then optimize on Quadtree mesh

Obtain $I_h(u) = I(u) + O(h^2)$ by adding cell (finite volume) contributions

$$\int_{c_l} \left[|\nabla u|^2 - 2uq\right] \; dxdy =$$
$$\frac{1}{2}\left[(u_l^{NE} - u_l^{NW})^2 + (u_l^{SE} - u_l^{SW})^2\right] +$$
$$\frac{1}{2}\left[(u_l^{NE} - u_l^{SE})^2 + (u_l^{NW} - u_l^{SW})^2\right] -$$
$$\frac{h_l^2}{2}\left[u_l^{NE} q_l^{NE} + u_l^{NW} q_l^{NW} + u_l^{SE} u_l^{SE} + u_l^{SW} q_l^{SW}\right] + O(h_l^4).$$

Expect 2nd order accuracy in solution as well.

Results

- Obtain $O(h^2)$ error if mesh is uniform ...
- ... **surprise:** but only $O(h)$ otherwise!
- In fact, this yields an artificial interface (boundary) across which homogeneous Neumann BC hold.

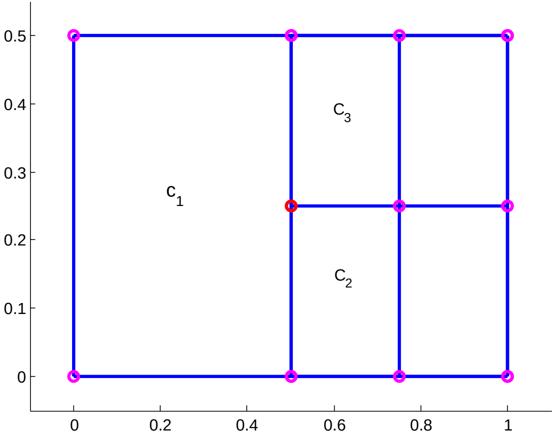
Results

- Obtain $O(h^2)$ error if mesh is uniform ...
- ... **surprise:** but only $O(h)$ otherwise!
- In fact, this yields an artificial interface (boundary) across which homogeneous Neumann BC hold.

Results

- Obtain $O(h^2)$ error if mesh is uniform ...
- ... **surprise:** but only $O(h)$ otherwise!
- In fact, this yields an artificial interface (boundary) across which homogeneous Neumann BC hold.

Typical neighboring cells



Results and explanation

- Obtain $O(h^2)$ error if mesh is uniform but only $O(h)$ otherwise!
- **Reason**: because of the **dangling node *** at middle of eastern face of cell c_1

$$4v_* - v_2^{SW} - v_3^{NW} - (v_2^{NE} + v_3^{SE}) = h^2 q_*.$$

- This yields an artificial interface (boundary) across which homogeneous Neumann BC hold.

What to do? How to fix?

Three options:

1. Set value of v_* as interpolation of its closest neighbors.
   However, this generates a nonsymmetric matrix to invert.
2. Switch to FEM, replacing c_1 by three triangles, adding edges from red point to opposite corners.
   Quadtree is still useful in keeping track of activities on this mesh.
3. Do nothing and live with the reduced order.

Outline

- Quadtrees and Octrees
- Cloth simulation (Chapter 2)
- Matlab's ode45 for Hamiltonian systems
- KdV instability
- Splitting cubic NLS
- Artificial boundary waves
- ENO, WENO and SSP
- MEMS device

Cloth simulation

This is reserved for another talk...

Outline

- Quadtrees and Octrees
- Cloth simulation
- Matlab's `ode45` for Hamiltonian systems (Chapters 6 & 2)
- KdV instability
- Splitting cubic NLS
- Artificial boundary waves
- ENO, WENO and SSP
- MEMS device

Hamiltonian systems

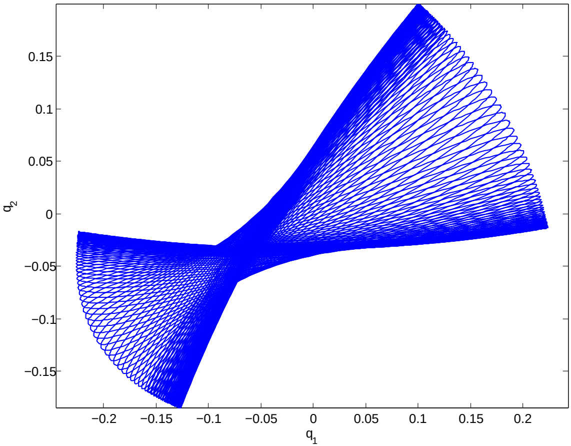
$$\frac{d\mathbf{q}}{dt} = \nabla_{\mathbf{p}} H(\mathbf{q}, \mathbf{p}), \quad \frac{d\mathbf{p}}{dt} = -\nabla_{\mathbf{q}} H(\mathbf{q}, \mathbf{p}).$$

Example: Henon-Heiles

$$H = \frac{1}{2}\left(p_1^2 + p_2^2\right) + \frac{1}{2}\left(q_1^2 + q_2^2\right) + q_1^2 q_2 - \frac{1}{3} q_2^3,$$

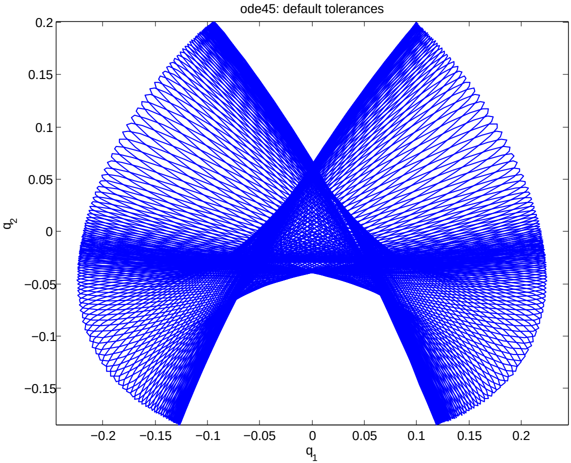
Henon-Heiles

Quasi-periodic orbit [McLachlan & Quispel ('06)]



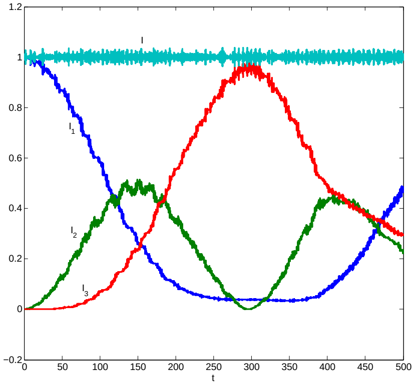
Henon-Heiles

Same phase portrait by `ode45` default tolerances [McLachlan] **surprise:**



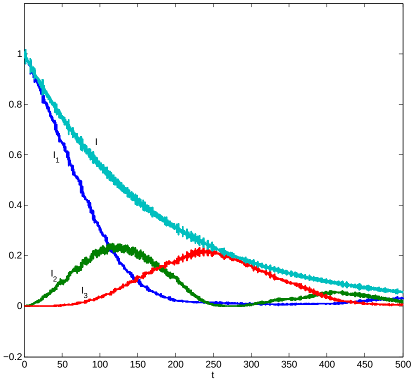
Another example: Fermi-Pasta-Ulam

$\omega = 100$; adiabatic invariant I



Fermi-Pasta-Ulam

Same adiabatic invariant I by ode45 default tolerances **surprise:**



Cause of poor results

Is this because:

- The method implemented in ode45 (which is a Dormand-Prince pair of orders 4 and 5) is not symplectic?
- The method becomes unstable for imaginary eigenvalues of the Jacobian matrix?
- The default tolerances are too loose?
- Something else?

Cause of poor results

Is this because:

- The method implemented in `ode45` (which is a Dormand-Prince pair of orders 4 and 5) is not symplectic?
- The method becomes unstable for imaginary eigenvalues of the Jacobian matrix?
- The default tolerances are too loose?
- Something else?

Cause of poor results

Is this because:

- The method implemented in `ode45` (which is a Dormand-Prince pair of orders 4 and 5) is not symplectic?
- The method becomes unstable for imaginary eigenvalues of the Jacobian matrix?
- The default tolerances are too loose?
- Something else?

Cause of poor results

Is this because:

- The method implemented in `ode45` (which is a Dormand-Prince pair of orders 4 and 5) is not symplectic?
- The method becomes unstable for imaginary eigenvalues of the Jacobian matrix?
- The default tolerances are too loose?
- Something else?

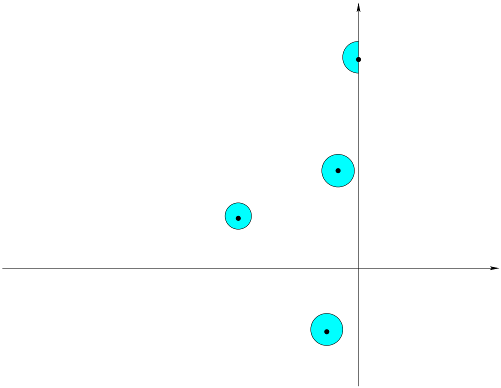
Cause of poor results

Is this because:

- The method implemented in ode45 (which is a Dormand-Prince pair of orders 4 and 5) is not symplectic?
- The method becomes unstable for imaginary eigenvalues of the Jacobian matrix?
- The default tolerances are too loose?
- Something else?

To dissipate or not to dissipate?

Consider integrating a hyperbolic-type PDE over a long time.



Common wisdom

- Common wisdom I: apply a little dissipation (almost conservative)
- Common wisdom II: do not apply dissipation (conservative, symplectic, multisymplectic)

Common wisdom

- Common wisdom I: apply a little dissipation (almost conservative)
- Common wisdom II: do not apply dissipation (conservative, symplectic, multisymplectic)

Recall: ODE absolute stability region

$$\frac{du}{dt} = \lambda u,$$

λ complex scalar (eigenvalue). Numerical method

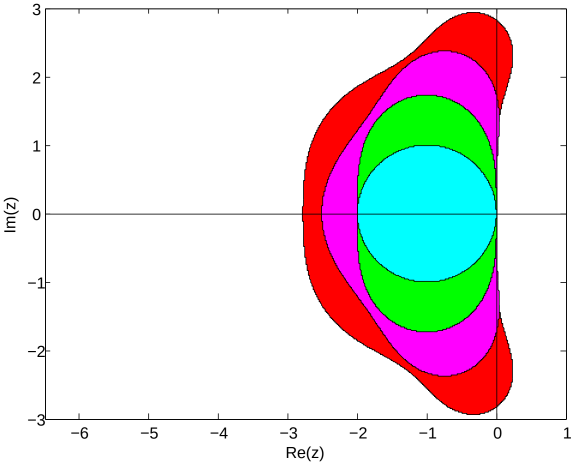
$$u_{n+1} = R(z) u_n, \quad z = \lambda \Delta t.$$

Region of absolute stability in complex plane of z is where

$$|u_{n+1}| \leq |u_n| \quad i.e. \; |R(z)| \leq 1.$$

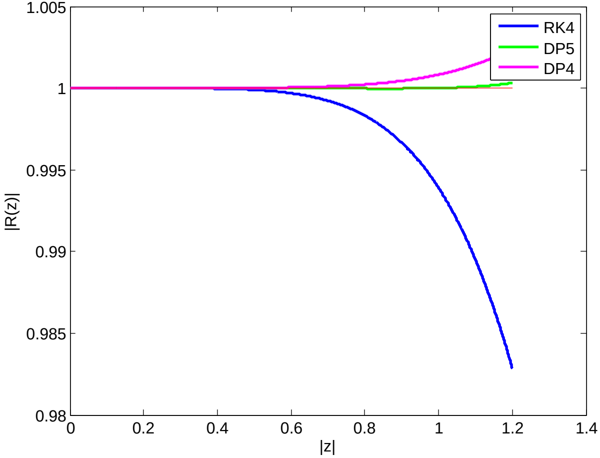
s-stage Runge-Kutta methods of order s

for $s = 1, 2, 3, 4$



Amplification factors along imaginary axis

RK4, DP5 and DP4

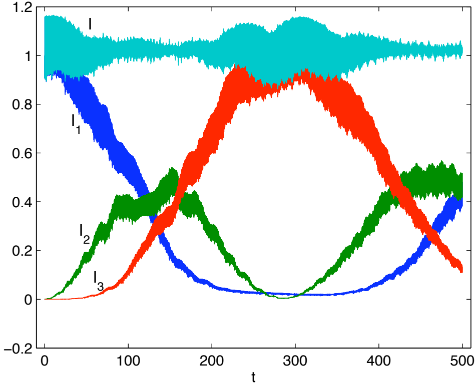


Time steps and quality results

Problem	Method	Steps	Result good?
HeHe	ode45 def.	5,961	No
	ode45 10^{-5}	8,737	Yes
	RK4	20,000	Yes
	RK4	10,000	No
	DP5	10,000	Yes
FPU	ode45 def.	112,085	No
	ode45 10^{-5}	253,369	No
	ode45 10^{-6}	402,045	Yes
	RK4	1,000,000	Yes
	RK4	200,000	No
	DP5	500,000	Yes
	DP5	100,000	No
	Verlet	200,000	Yes
	Verlet	50,000	No

Symplectic Verlet

50,000 steps



Outline

- Quadtrees and Octrees
- Cloth simulation
- Matlab's ode45 for Hamiltonian systems
- KdV instability (Chapters 5 & 7)
- Splitting cubic NLS
- Artificial boundary waves
- ENO, WENO and SSP
- MEMS device

Korteweg-de Vries (KdV) numerical instability

$$\begin{aligned} u_t &= \alpha(u^2)_x + \rho u_x + \nu u_{xxx} \\ &= V'(u)_x + \nu u_{xxx}, \quad V(u) = \frac{\alpha}{3}u^3 + \frac{\rho}{2}u^2. \end{aligned}$$

Initial conditions $u(x,0) = u_0(x)$
Boundary conditions: periodic
Set $\rho = 0$. Consider Eulerian finite volume/difference discretizations: on a fixed grid with step sizes $\Delta x,\ \Delta t$.

Explicit numerical method

[Zabusky & Kruskal ('65)]: Leap-frog – an explicit scheme. With $\mu = \frac{\Delta t}{\Delta x}$

$$
\begin{aligned}
v_j^{n+1} &= v_j^{n-1} + \frac{2\alpha\mu}{3}(v_{j-1}^n + v_j^n + v_{j+1}^n)(v_{j+1}^n - v_{j-1}^n) \\
&+ \frac{\nu\mu}{\Delta x^2}(v_{j+2}^n - 2v_{j+1}^n + 2v_{j-1}^n - v_{j-2}^n)
\end{aligned}
$$

Constant coefficient **stability analysis**: restrict time step to

$$
\Delta t < \Delta x / \left[\frac{|\nu|}{\Delta x^2} + 2|\alpha u_{\max}| \right]
$$

Numerical example

- [Zhao & Qin ('00), Ascher & McLachlan ('04,'05)]: Take

$$u_0(x) = \cos(\pi x), \quad u(t,0) = u(t,2),$$
$$\nu = -0.022^2, \ \alpha = -0.5.$$

- Try various Δx, Δt satisfying linear stability bound.
- **surprise:** Obtain blowup for $t > 21/\pi$ (!)
  The instability takes time to develop, so results at $t = 1$ (say) do not indicate the trouble at a later time.

Numerical example

- [Zhao & Qin ('00), Ascher & McLachlan ('04,'05)]: Take

$$u_0(x) = \cos(\pi x), \quad u(t,0) = u(t,2),$$
$$\nu = -0.022^2, \ \alpha = -0.5.$$

- Try various Δx, Δt satisfying linear stability bound.
- **surprise:** Obtain blowup for $t > 21/\pi$ (!)
  The instability takes time to develop, so results at $t = 1$ (say) do not indicate the trouble at a later time.

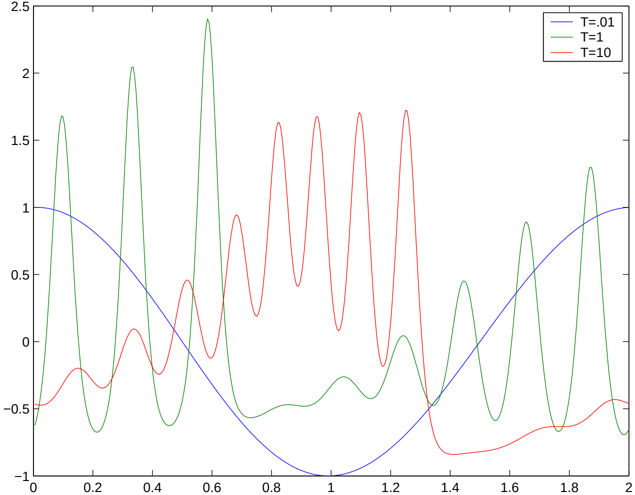
Numerical example

- [Zhao & Qin ('00), Ascher & McLachlan ('04,'05)]: Take

$$u_0(x) = \cos(\pi x), \quad u(t,0) = u(t,2),$$
$$\nu = -0.022^2, \ \alpha = -0.5.$$

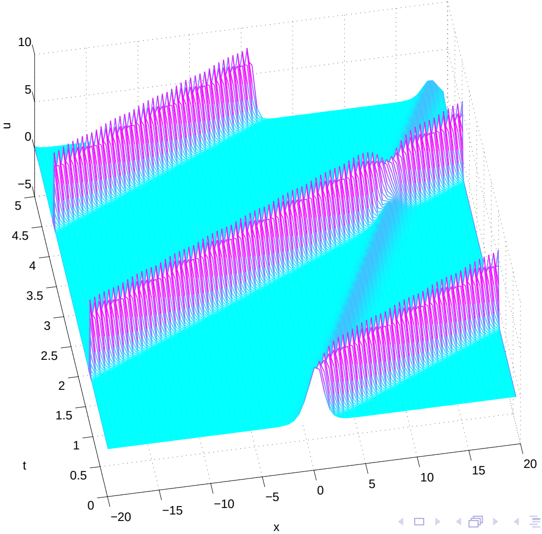
- Try various Δx, Δt satisfying linear stability bound.
- **surprise:** Obtain blowup for $t > 21/\pi$ (!)
  The instability takes time to develop, so results at $t = 1$ (say) do not indicate the trouble at a later time.

Solution components



KdV soliton

Solution progress in time for another set of parameters displaying two solitons, using a better method



Outline

- Quadtrees and Octrees
- Cloth simulation
- Matlab's ode45 for Hamiltonian systems
- KdV instability
- Splitting cubic NLS (Chapter 9)
- Artificial boundary waves
- ENO, WENO and SSP
- MEMS device

Splitting cubic NLS

Nonlinear Schrödinger equation with a cubic nonlinearity

$$\psi_t = \imath(\psi_{xx} + |\psi|^2\psi).$$

- Norm preservation $\|\psi(t)\|^2 = \|\psi(0)\|^2$
- Hamiltonian PDE $H(\psi, \bar{\psi}) = \psi_x \bar{\psi}_x - \frac{1}{2}\psi^2\bar{\psi}^2$

Splitting cubic NLS

Nonlinear Schrödinger equation with a cubic nonlinearity

$$\psi_t = \imath(\psi_{xx} + |\psi|^2\psi).$$

- Norm preservation $\|\psi(t)\|^2 = \|\psi(0)\|^2$
- Hamiltonian PDE $H(\psi, \bar{\psi}) = \psi_x \bar{\psi}_x - \frac{1}{2}\psi^2\bar{\psi}^2$

Splitting cubic NLS

Nonlinear Schrödinger equation with a cubic nonlinearity

$$\psi_t = \imath(\psi_{xx} + |\psi|^2\psi).$$

- Norm preservation $\|\psi(t)\|^2 = \|\psi(0)\|^2$
- Hamiltonian PDE $H(\psi, \bar{\psi}) = \psi_x \bar{\psi}_x - \frac{1}{2}\psi^2\bar{\psi}^2$

An obvious (staggered) splitting

1. $$u_t = \imath u_{xx}$$

   Apply standard 3-point discretization in space and **implicit midpoint** in time: symplectic and norm-preserving.

2. $$u_t = \imath |u|^2 u$$

   Exact ODE solution for each x

   $$u(t + \Delta t) = u(t)\ e^{\imath \Delta t |u|^2}.$$

Composition yields a symplectic, conservative method.

An obvious (staggered) splitting

1. $$u_t = \imath u_{xx}$$

   Apply standard 3-point discretization in space and **implicit midpoint** in time: symplectic and norm-preserving.

2. $$u_t = \imath |u|^2 u$$

   Exact ODE solution for each x

   $$u(t + \Delta t) = u(t)\ e^{\imath \Delta t |u|^2}.$$

Composition yields a symplectic, conservative method.

An obvious (staggered) splitting

1. $$u_t = \imath u_{xx}$$

   Apply standard 3-point discretization in space and **implicit midpoint** in time: symplectic and norm-preserving.

2. $$u_t = \imath |u|^2 u$$

   Exact ODE solution for each x

   $$u(t + \Delta t) = u(t)\ e^{\imath \Delta t |u|^2}.$$

Composition yields a symplectic, conservative method.

Example: solitons

Periodic BC on $[-20, 80]$. IC

$$\psi(0,x) = e^{\imath x/2}\mathrm{sech}(x/\sqrt{2}) + e^{\imath(x-25)/20}\mathrm{sech}((x-25)/\sqrt{2}).$$

[Hundsdorfer & Verwer (03')]

t	Δt	Δx	Error-Ham	Error-norm
200	.1	.1	3.7e-5	4.3e-13
	.01	.01	3.9e-9	1.5e-11

Example: solitons

Periodic BC on $[-20, 80]$. IC

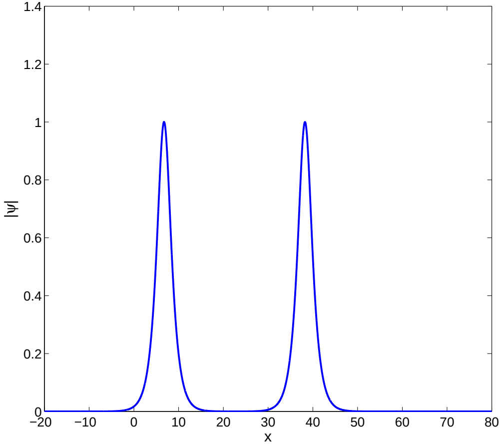
$$\psi(0,x) = e^{\imath x/2}\mathrm{sech}(x/\sqrt{2}) + e^{\imath(x-25)/20}\mathrm{sech}((x-25)/\sqrt{2}).$$

[Hundsdorfer & Verwer (03')]

t	Δt	Δx	Error-Ham	Error-norm
200	.1	.1	3.7e-5	4.3e-13
	.01	.01	3.9e-9	1.5e-11

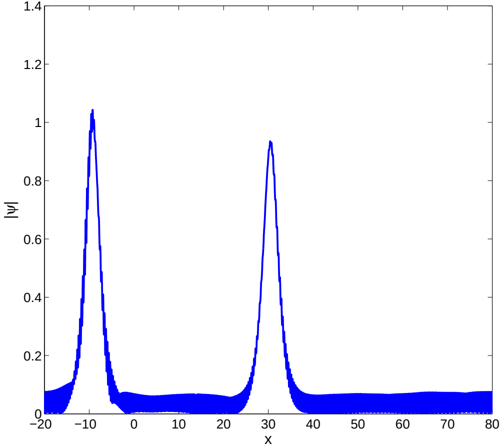
Soliton solution

Solution at time $t = 200$



Soliton solution

surprise: Solution at time $t = 1000$ displays instability in derivative



Error indicators

t	Δt	Δx	Error-Ham	Error-norm
200	.1	.1	3.7e-5	4.3e-13
	.01	.01	3.9e-9	1.5e-11
1000	.1	.1	5.2e+2	2.9e-12
	.01	.01	3.4e+2	7.7e-11

Recall splitting method

1.
$$u_t = \imath u_{xx}$$

Apply standard 3-point discretization in space and **implicit midpoint** in time: symplectic and norm-preserving.

2.
$$u_t = \imath |u|^2 u$$

Exact ODE solution for each x

$$u(t + \Delta t) = u(t)\ e^{\imath \Delta t |u|^2}.$$

Composition yields a symplectic method.

The full error table

t	Δt	Δx	Error-Ham	Error-norm
200	.1	.1	3.7e-5	4.3e-13
	.01	.01	3.9e-9	1.5e-11
1000	.1	.1	5.2e+2	2.9e-12
	.01	.1	3.3e-7	4.2e-12
	.01	.01	3.4e+2	7.7e-11

[Ascher & Reich ('99)]

Spectral splitting method

Same splitting, but solve $u_t = \imath u_{xx}$ using a spectral method in both space and time:

$$u(t + \Delta t) = \mathcal{F}^{-1}\left(e^{-\imath\xi^2\Delta t}\mathcal{F}(u(t))\right).$$

Discretize: $u(t) \equiv u^n = (u_1^n, \ldots, u_J^n)$, $u_j^n \approx u(j\Delta x, n\Delta t)$, $u(t + \Delta t) \equiv u^{n+1}$, and $\mathcal{F}$ is the fast Fourier transform.

Spectral splitting method results

With same number J of Fourier modes as spatial mesh points before, results are

- more accurate before instability sets in;
- however, instability sets in even earlier, and results then are even less accurate.

For ensured stability, take

$$\Delta t < \frac{\Delta x^2}{\pi}$$

[Weideman & Herbst, '86]

Spectral splitting method results

With same number J of Fourier modes as spatial mesh points before, results are

- more accurate before instability sets in;
- however, instability sets in even earlier, and results then are even less accurate.

For ensured stability, take

$$\Delta t < \frac{\Delta x^2}{\pi}$$

[Weideman & Herbst, '86]

Spectral splitting method results

With same number J of Fourier modes as spatial mesh points before, results are

- more accurate before instability sets in;
- however, instability sets in even earlier, and results then are even less accurate.

For ensured stability, take

$$\Delta t < \frac{\Delta x^2}{\pi}$$

[Weideman & Herbst, '86]

Spectral splitting method results

With same number J of Fourier modes as spatial mesh points before, results are

- more accurate before instability sets in;
- however, instability sets in even earlier, and results then are even less accurate.

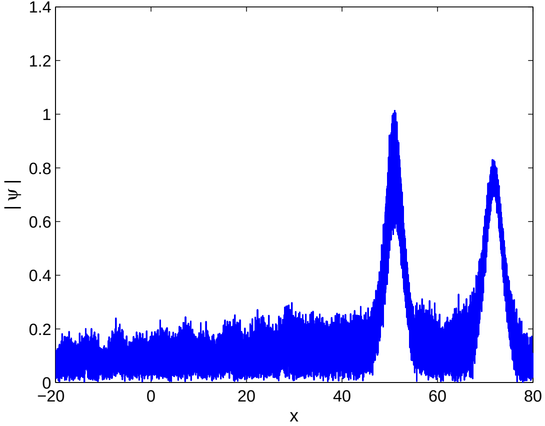
For ensured stability, take

$$\Delta t < \frac{\Delta x^2}{\pi}$$

[Weideman & Herbst, '86]

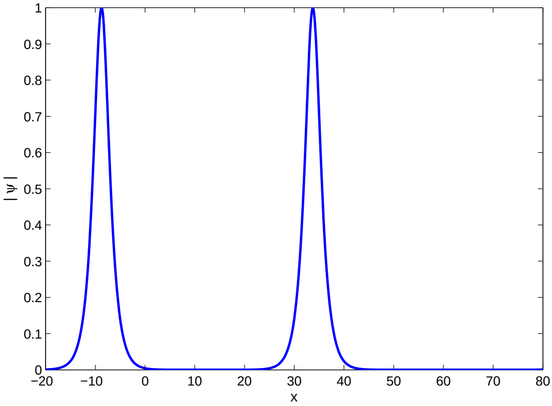
Soliton solution with spectral method

Solution at time $t = 1000,\ \Delta x = .01,\ \Delta t = .01$



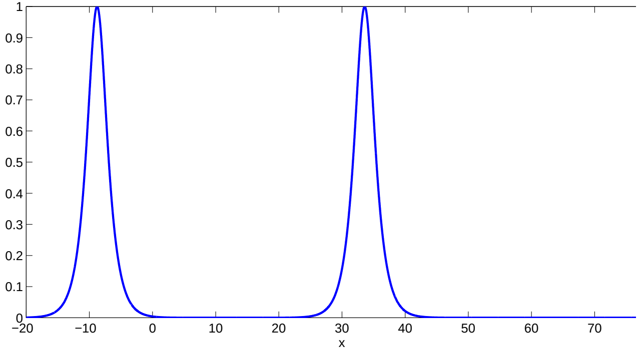
Soliton solution with spectral method

Solution at time $t = 1000$, $\Delta x = .1$, $\Delta t = .0025$



Soliton solution with attenuated midpoint method

Solution at time $t = 1000,\ \Delta x = .01,\ \Delta t = .01,\ \varepsilon = h^2$



Outline

- Quadtrees and Octrees
- Cloth simulation
- Matlab's ode45 for Hamiltonian systems
- KdV instability
- Splitting cubic NLS
- Artificial boundary waves (Chapter 8)
- ENO, WENO and SSP
- MEMS device

Simple advection with special IC

$$u_t - u_x = 0, \quad 0 \le x \le 1, \; t \ge 0,$$
$$u_0(x) = e^{-100(x-.5)^2}, \qquad u(t,1) = 0.$$

Note: problem requires only BC at $x = 1$, not at $x = 0$.

- **Upwind**: need nothing additional
- **Lax-Wendroff** and **leap-frog**: require numerical BC at $x = 0$.
- For the latter two choose, reasonably, simple extrapolation

$$v_0^n = v_1^n, \quad \forall n.$$

Simple advection with special IC

$$u_t - u_x = 0, \quad 0 \le x \le 1, \; t \ge 0,$$
$$u_0(x) = e^{-100(x-.5)^2}, \qquad u(t,1) = 0.$$

Note: problem requires only BC at $x = 1$, not at $x = 0$.

- **Upwind**: need nothing additional
- **Lax-Wendroff** and **leap-frog**: require numerical BC at $x = 0$.
- For the latter two choose, reasonably, simple extrapolation

$$v_0^n = v_1^n, \quad \forall n.$$

Simple advection with special IC

$$
\begin{aligned}
u_t - u_x &= 0, \quad 0 \le x \le 1, \; t \ge 0, \\
u_0(x) = e^{-100(x-.5)^2}, &\qquad u(t,1) = 0.
\end{aligned}
$$

Note: problem requires only BC at $x = 1$, not at $x = 0$.

- **Upwind**: need nothing additional
- **Lax-Wendroff** and **leap-frog**: require numerical BC at $x = 0$.
- For the latter two choose, reasonably, simple extrapolation

$$v_0^n = v_1^n, \quad \forall n.$$

Simple advection with special IC

$$\begin{aligned} u_t - u_x &= 0, \quad 0 \le x \le 1, \ t \ge 0, \\ u_0(x) = e^{-100(x-.5)^2}, & \qquad u(t,1) = 0. \end{aligned}$$

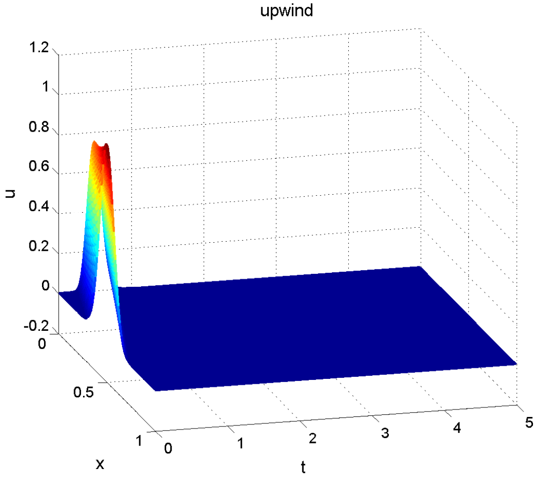
Note: problem requires only BC at $x = 1$, not at $x = 0$.

- **Upwind**: need nothing additional
- **Lax-Wendroff** and **leap-frog**: require numerical BC at $x = 0$.
- For the latter two choose, reasonably, simple extrapolation

$$v_0^n = v_1^n, \quad \forall n.$$

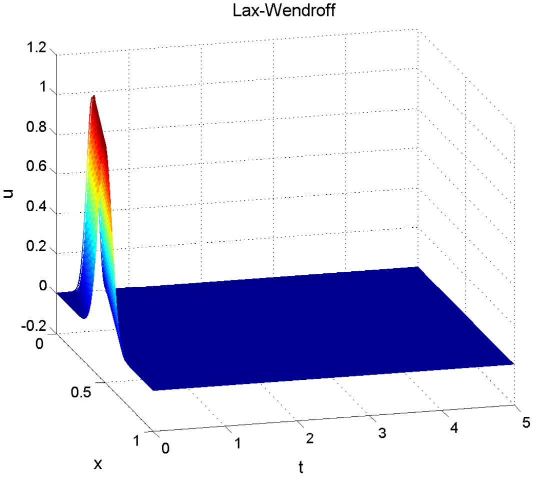
Upwind performance

Upwind: no artificial wave but inaccurate



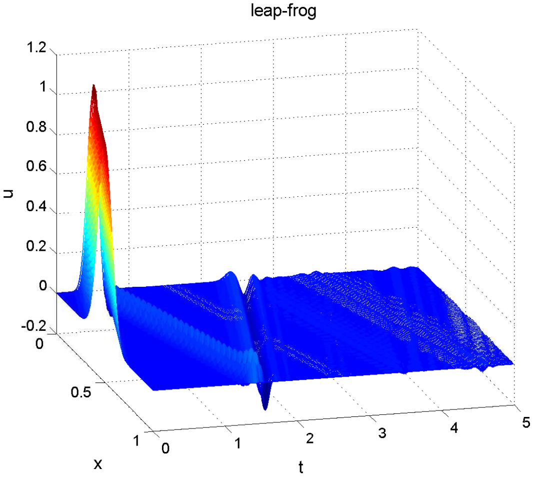
Lax-Wendroff performance

Lax-Wendroff: a little dissipativity helps; results more accurate



Leap-frog performance

Leap-frog: **surprise:** bad artificial waves; a conservative disaster



Outline

- Quadtrees and Octrees
- Cloth simulation
- Matlab's ode45 for Hamiltonian systems
- KdV instability
- Splitting cubic NLS
- Artificial boundary waves
- ENO, WENO and SSP (Chapter 10)
- MEMS device

Conservation laws

System of conservation laws

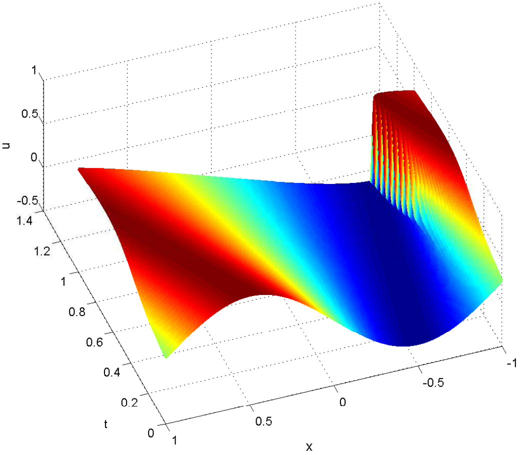
$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}.$$

Instance: the inviscid Burgers equation

$$u_t + \frac{1}{2}(u^2)_x = 0.$$

Inviscid Burgers

Shock in the inviscid Burgers equation: discontinuous solution develops from smooth initial data and periodic boundary conditions



Upwind differences

With $\mu = \Delta t / \Delta x$

$$v_j^{n+1} = v_j^n - \frac{\mu}{2} \begin{cases} [(v_{j+1}^n)^2 - (v_j^n)^2] & \text{if } v_j^n < 0 \\ [(v_j^n)^2 - (v_{j-1}^n)^2] & \text{if } v_j^n \geq 0 \end{cases}.$$

Can view as:

1. Semi-Lagrangian approach, integrating along characteristics;
2. Forward Euler applied to a one-sided semi-discretization.

Upwind differences

With $\mu = \Delta t / \Delta x$

$$v_j^{n+1} = v_j^n - \frac{\mu}{2} \begin{cases} [(v_{j+1}^n)^2 - (v_j^n)^2] & \text{if } v_j^n < 0 \\ [(v_j^n)^2 - (v_{j-1}^n)^2] & \text{if } v_j^n \geq 0 \end{cases} .$$

Can view as:

1. Semi-Lagrangian approach, integrating along characteristics;
2. Forward Euler applied to a one-sided semi-discretization.

Upwind differences

With $\mu = \Delta t / \Delta x$

$$v_j^{n+1} = v_j^n - \frac{\mu}{2} \begin{cases} [(v_{j+1}^n)^2 - (v_j^n)^2] & \text{if } v_j^n < 0 \\ [(v_j^n)^2 - (v_{j-1}^n)^2] & \text{if } v_j^n \geq 0 \end{cases}.$$

Can view as:

1. Semi-Lagrangian approach, integrating along characteristics;
2. Forward Euler applied to a one-sided semi-discretization.

Essentially non-oscillatory (ENO)

[Harten et al. ('87)] Higher order one-sided semi-discretization (finite volume or finite difference). Construct divided differences for polynomial interpolation and choose points that minimize size of high divided difference, thus not crossing discontinuity. e.g. for a cubic choose from

$$x_{j-5/2}\ x_{j-3/2}\ x_{j-1/2}\ x_{j+1/2} \qquad x_{j-1/2}\ x_{j+1/2}\ x_{j+3/2}\ x_{j+5/2}$$

$$x_{j-3/2}\ x_{j-1/2}\ x_{j+1/2}\ x_{j+3/2}$$

to obtain 3rd order.
Combine with higher order discretization in time: **which one?**

Strong stability preserving (SSP)

[Gottlieb, Shu & Tadmor ('01), Ruuth & Spiteri ('02), Higueras ('04), Gottlieb, Ketcheson & Shu ('09)]

Assuming that forward Euler satisfies strong stability

$$\|v^{n+1}\| \leq \|v^n\| \quad \forall n,$$

require higher order ODE method to satisfy this too. Popular 3rd order method:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{array}$$

Strong stability preserving (SSP)

[Gottlieb, Shu & Tadmor ('01), Ruuth & Spiteri ('02), Higueras ('04), Gottlieb, Ketcheson & Shu ('09)]

Assuming that forward Euler satisfies strong stability

$$\|v^{n+1}\| \le \|v^n\| \quad \forall n,$$

require higher order ODE method to satisfy this too. Popular 3rd order method:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{array}$$

Weighted essentially non-oscillatory (WENO)

[Shu ('98), Wang & Spiteri ('07)]
e.g. for a cubic, **instead of** choosing one from

$x_{j-5/2}\ x_{j-3/2}\ x_{j-1/2}\ x_{j+1/2}$ $\qquad$ $x_{j-1/2}\ x_{j+1/2}\ x_{j+3/2}\ x_{j+5/2}$

$x_{j-3/2}\ x_{j-1/2}\ x_{j+1/2}\ x_{j+3/2}$

select a **weighted** combination to obtain one-sided near discontinuity and **5**th order where solution is smooth. Choose weights dynamically.
Combine with higher order discretization in time: **which one?**

Time discretization for WENO

- Note that in smooth areas WENO is essentially (close to) being a centred discretization in space. So eigenvalues of semi-discretization are close to being imaginary: forward Euler will not work well!
- (Positive) **surprise:** Can forget about the SSP restrictions in the WENO context.
- Can apply the classical Runge-Kutta RK4, or DP5.

Time discretization for WENO

- Note that in smooth areas WENO is essentially (close to) being a centred discretization in space. So eigenvalues of semi-discretization are close to being imaginary: forward Euler will not work well!
- (Positive) **surprise:** Can forget about the SSP restrictions in the WENO context.
- Can apply the classical Runge-Kutta RK4, or DP5.

Outline

- Quadtrees and Octrees
- Cloth simulation
- Matlab's ode45 for Hamiltonian systems
- KdV instability
- Splitting cubic NLS
- Artificial boundary waves
- ENO, WENO and SSP
- MEMS device (Chapter 11)

MEMS device

[Guo, Pan & Ward ('06)]

$$u_t = \Delta u - \frac{\lambda}{(1+u)^2}, \quad (x,y) \in \Omega,$$

subject to homogeneous Dirichlet BC and zero initial data.

If $\lambda > \lambda_*$ then u will reach value -1 at some point in finite touchdown time T_*.

MEMS DEVICE

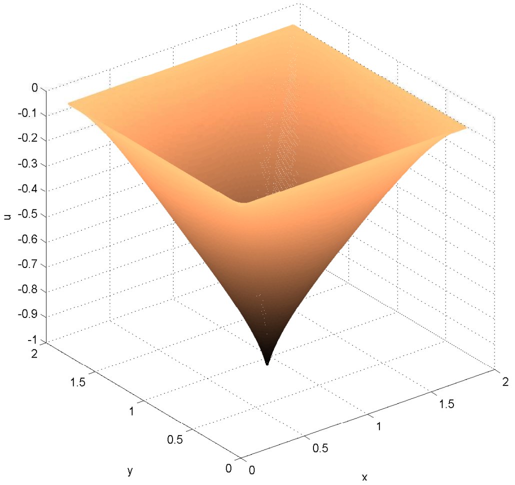
[Guo, Pan & Ward ('06)]

$$u_t = \Delta u - \frac{\lambda}{(1+u)^2}, \quad (x,y) \in \Omega,$$

subject to homogeneous Dirichlet BC and zero initial data.
If $\lambda > \lambda_*$ then u will reach value -1 at some point in finite touchdown time T_*.

MEMS TOUCHDOWN

Solution $u(T_*, x, y)$, $\lambda = 2$, $T_* = .1975$



MEMS device transformation

- Obviously, obtaining this solution by discretizing the given PDE directly will require at least a very fine (or highly nonuniform) mesh.
- (Positive) **surprise:** Can transform the PDE first!

$$w = \frac{1}{3\lambda}(1+u)^3,$$

yields the PDE

$$w_t = \Delta w - \frac{2}{3w}|\nabla w|^2 - 1.$$

- Note that w has touchdown value 0, and importantly, unlike u it varies gently in x everywhere. Hence there is no serious numerical difficulty in solving this problem anymore.
- The figure was obtained by solving for w and transforming pointwise back to u.

MEMS device transformation

- Obviously, obtaining this solution by discretizing the given PDE directly will require at least a very fine (or highly nonuniform) mesh.
- (Positive) **surprise:** Can transform the PDE first!

$$w = \frac{1}{3\lambda}(1+u)^3,$$

yields the PDE

$$w_t = \Delta w - \frac{2}{3w}|\nabla w|^2 - 1.$$

- Note that w has touchdown value 0, and importantly, unlike u it varies gently in x everywhere. Hence there is no serious numerical difficulty in solving this problem anymore.
- The figure was obtained by solving for w and transforming pointwise back to u.

MEMS DEVICE TRANSFORMATION

- Obviously, obtaining this solution by discretizing the given PDE directly will require at least a very fine (or highly nonuniform) mesh.
- (Positive) **surprise:** Can transform the PDE first!

$$w = \frac{1}{3\lambda}(1+u)^3,$$

yields the PDE

$$w_t = \Delta w - \frac{2}{3w}|\nabla w|^2 - 1.$$

- Note that w has touchdown value 0, and importantly, unlike u it varies gently in x everywhere. Hence there is no serious numerical difficulty in solving this problem anymore.
- The figure was obtained by solving for w and transforming pointwise back to u.

Conclusion

- Strange computations can lead to interesting observations.
- In theory, practice and theory are close. In practice, they may not be.