Active Flux for nonlinear balance laws – a 3rd order structure preserving method

Wasilij Barsukow

DFG postdoctoral fellow

University of Zurich

October 22, 2020

Outline

Outline:

1. Introduction to Active Flux
2. 1D versus multi-D
3. Evolution operators for **nonlinear problems**
   - Sufficiently high order of accuracy
   - Discontinuity formation?
   - Limiting
4. Well-balanced methods for balance laws

Introduction to Active Flux

Finite Volume schemes

Conservation law: $\partial_t q + \nabla \cdot \mathbf{f}(q) = 0$, $q : \mathbb{R}^+_0 \times \mathbb{R}^d \to \mathbb{R}^n$ with IVP $q(0, \mathbf{x}) = q_0(\mathbf{x})$.

Finite Volume schemes

Conservation law: $\partial_t q + \nabla \cdot \mathbf{f}(q) = 0$, $q : \mathbb{R}_0^+ \times \mathbb{R}^d \to \mathbb{R}^n$ with IVP $q(0, \mathbf{x}) = q_0(\mathbf{x})$.
The discrete degree of freedom $q_\mathcal{C}$ in computational cell $\mathcal{C}$ is given the interpretation $q_\mathcal{C} = \frac{1}{|\mathcal{C}|} \int_\mathcal{C} \mathrm{d}\mathbf{x}\, q(t, \mathbf{x})$. Its time update is, by Gauss law

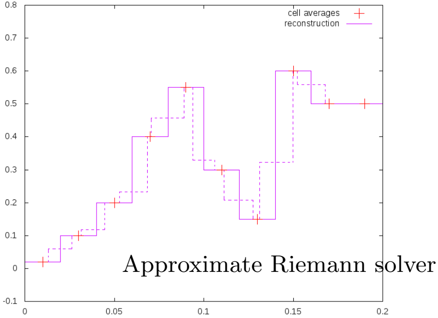
$$\partial_t q_\mathcal{C} + \frac{1}{|\mathcal{C}|} \int_{\partial\mathcal{C}} \mathrm{d}\mathbf{x}\mathbf{n} \cdot f(q) = 0 \tag{1}$$

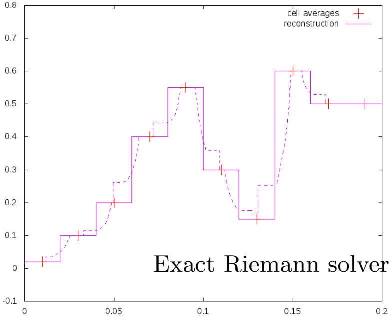
Finite Volume schemes

Conservation law: $\partial_t q + \nabla \cdot \mathbf{f}(q) = 0$, $q : \mathbb{R}^+_0 \times \mathbb{R}^d \to \mathbb{R}^n$ with IVP $q(0, \mathbf{x}) = q_0(\mathbf{x})$. The discrete degree of freedom $q_{\mathcal{C}}$ in computational cell $\mathcal{C}$ is given the interpretation $q_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \int_{\mathcal{C}} \mathrm{d}\mathbf{x}\, q(t, \mathbf{x})$. Its time update is, by Gauss law

$$\partial_t q_{\mathcal{C}} + \frac{1}{|\mathcal{C}|} \int_{\partial \mathcal{C}} \mathrm{d}\mathbf{x}\mathbf{n} \cdot f(q) = 0 \tag{1}$$

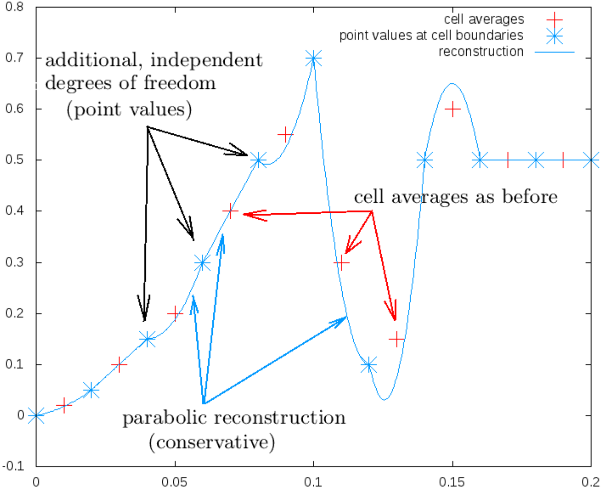
The numerical flux is obtained from a **conservative** and **piecewise continuous** reconstruction:





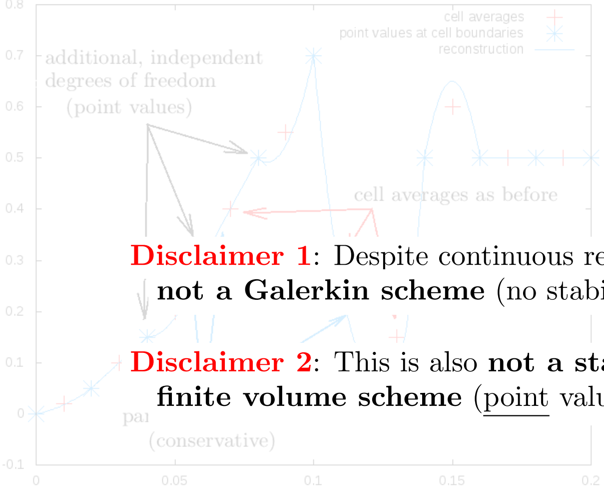
Active Flux schemes

In this talk the following reconstructions are considered instead:



Active Flux schemes

In this talk the following reconstructions are considered instead:

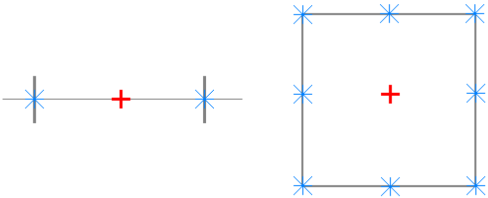


Disclaimer 1: Despite continuous reconstruction this is **not a Galerkin scheme** (no stability issues!)

Disclaimer 2: This is also **not a staggered finite volume scheme** (point values!)

Point values and cell averages

Design decision: Declare the **point values** at cell boundaries to be **independent degrees of freedom**.



A number of consequences:

- **higher order**
- Reconstruction has to interpolate the point values and match the average: **compact stencil**
- The flux needed for the cell average can be evaluated immediately

How to update the pointwise degrees of freedom?

Evolution operator

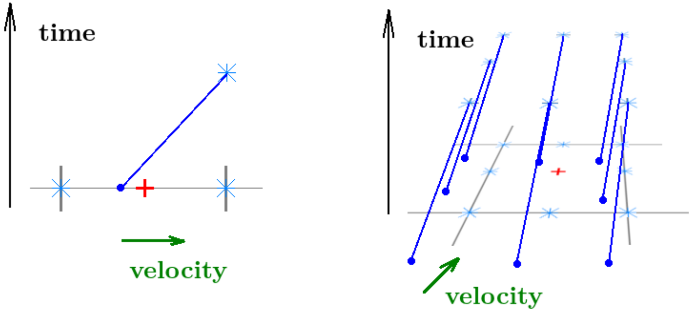
The reconstruction can be used as initial data for an IVP at the location of the pointwise degree of freedom.
The IVP can be solved exactly or approximately (compare: exact and approximate Riemann Solvers).

Evolution operator

The reconstruction can be used as initial data for an IVP at the location of the pointwise degree of freedom.
The IVP can be solved exactly or approximately (compare: exact and approximate Riemann Solvers).

Example: $\partial_t q + c\partial_x q = 0$ (one-dimensional linear advection)

$$q(t, x) = q_{\text{recon}}(x - ct)$$



History

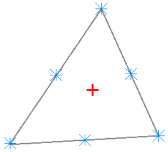
Published implementations of Active Flux:

- 1977: van Leer (one-dimensional advection)
- 2011: Eymann, Roe (Burgers' equation; one-dimensional nonlinear systems)

History

Published implementations of Active Flux:

- 1977: van Leer (one-dimensional advection)
- 2011: Eymann, Roe (Burgers' equation; one-dimensional nonlinear systems)
- 2013: Eymann, Roe (multi-dimensional acoustics on triangular grids)
- 2017: Fan, Maeng, Roe (p-system and pressureless Euler on triangular grids)
- 2019: Helzel, Kerkmann, Scandurra (approximate evolution operator)
- 2019: [WB et al., 2019] (multi-dimensional acoustics on Cartesian grids)
- 2020: [WB et al., 2020, WB and Berberich, 2020] (nonlinear balance laws in 1D)
- ...



Recap: Active Flux

General algorithm of any Active Flux method:

1. cell averages and point values given
2. compute conservative **reconstruction** that also interpolates the point values
3. use reconstruction as initial data for **point value update**
4. perform quadrature to obtain fluxes: **cell average update** as in finite volume methods
5. continue at 1.

Multi-d systems (very briefly)

Multi-dimensionality

The Active Flux method is particularly suited for **multiple spatial dimensions**.

Multi-dimensionality

The Active Flux method is particularly suited for **multiple spatial dimensions**.

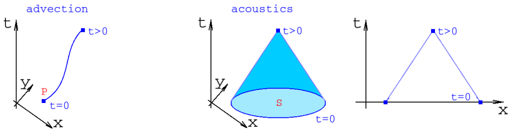
The **acoustic equations** are a prototypic hyperbolic system with non-trivial behaviour in multi-d. They are contained in the Euler equations:

$$\partial_t \varrho + \mathbf{v} \cdot \nabla \varrho + \varrho \nabla \cdot \mathbf{v} = 0$$

$$\partial_t \mathbf{v} + \nabla p = 0 \qquad\qquad \partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + \frac{\nabla p}{\varrho} = 0$$

$$\partial_t p + c^2 \nabla \cdot \mathbf{v} = 0 \qquad\qquad \partial_t p + \mathbf{v} \cdot \nabla p + \varrho c^2 \nabla \cdot \mathbf{v} = 0$$

Multi-dimensionality



The Active Flux method is particularly suited for **multiple spatial dimensions**.

The **acoustic equations** are a prototypic hyperbolic system with non-trivial behaviour in multi-d. They are contained in the Euler equations:

$$\partial_t \varrho + \mathbf{v} \cdot \nabla \varrho + \varrho \nabla \cdot \mathbf{v} = 0$$

$$\partial_t \mathbf{v} + \nabla p = 0 \qquad\qquad \partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + \frac{\nabla p}{\varrho} = 0$$

$$\partial_t p + c^2 \nabla \cdot \mathbf{v} = 0 \qquad\qquad \partial_t p + \mathbf{v} \cdot \nabla p + \varrho c^2 \nabla \cdot \mathbf{v} = 0$$

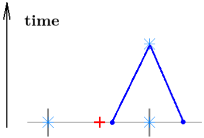
They capture the behaviour of acoustics and leave aside advection. They also govern the (**Lagrangian**) evolution of a fluid element. In particular, consider **linear acoustics** with $c =$ const. **Involution**: $\partial_t(\nabla \times \mathbf{v}) = 0$.

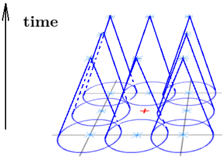
[Morton and Roe, 2001], [Lukacova-Medvidova et al., 2000], [Torrilhon and Fey, 2004], [Jeltsch and Torrilhon, 2006], [Mishra and Tadmor, 2009], [Dellacherie, 2010], [Lung and Roe, 2014], [Amadori and Gosse, 2015], [Franck and Gosse, 2018] and many others.

Evolution operator

linear acoustics

$$\partial_t \mathbf{v} + c\nabla p = 0$$
$$\partial_t p + c\nabla \cdot \mathbf{v} = 0$$





Theorem

$$p(t,\mathbf{x}) = p_0(\mathbf{x}) + \int_0^{ct} \mathrm{d}r\, r \cdot M\big[\operatorname{div}\operatorname{grad} p_0\big](\mathbf{x}, r) - ct \cdot M\big[\operatorname{div} \mathbf{v}_0\big](\mathbf{x}, ct)$$

$$\mathbf{v}(t,\mathbf{x}) = \mathbf{v}_0(\mathbf{x}) + \int_0^{ct} \mathrm{d}r\, r \cdot M\big[\operatorname{grad}\operatorname{div} \mathbf{v}_0\big](\mathbf{x}, r) - ct \cdot M\big[\operatorname{grad} p_0\big](\mathbf{x}, ct)$$

Spherical mean:

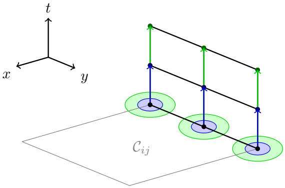
$$M\left[f\right](\mathbf{x}, r) := \frac{1}{4\pi} \oint_{S^2} \mathrm{d}\mathbf{y}\, f(\mathbf{x} + r\mathbf{y}) = \frac{1}{4\pi} \int_0^{2\pi} \mathrm{d}\phi \int_0^{\pi} \mathrm{d}\vartheta\, \sin\vartheta f\left(\mathbf{x} + r\cdot\mathbf{n}\right)$$

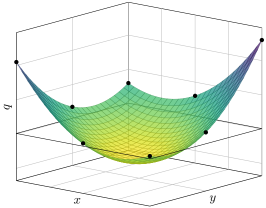
[WB and Klingenberg, 2018]

Active Flux for acoustics on Cartesian grids

A particular implementation of the general idea:

- **Acoustic equations**
- **Cartesian grid**; point values located at vertices and edge midpoints (9 free parameters)
- **Biparabolic reconstruction** (9 equations ✓)
- Exact evolution operator
- ⇒ **third order** (one substep in time, **fully explicit**, Simpson rule for flux quadrature)





WB, J. Hohm, C. Klingenberg and Ph.L. Roe: arXiv:1812.01612

Active Flux for linear acoustics

Theorem ([WB et al., 2019])

If the initial data fulfill the following discretizations of $\operatorname{div} \mathbf{v} = 0$

$$\frac{\{[u^N]_{i+\frac12}\}_{j+\frac12}}{\Delta x} + \frac{[\{v^N\}_{i+\frac12}]_{j+\frac12}}{\Delta y} = 0 \qquad \frac{\langle [u]_{i\pm1}\rangle_j^{(4)}}{\Delta x} + \frac{[\langle v\rangle_i^{(4)}]_{j\pm1}}{\Delta y} = 0 \tag{2}$$

$$\frac{\langle [u^{EH}]_{i+\frac12}\rangle_j^{(6)}}{\Delta x} + \frac{[\langle v^{EV}\rangle_i^{(6)}]_{j+\frac12}}{\Delta y} = 0 \qquad \frac{[u^{EV}]_{i-\frac12,j}}{\Delta x} + \frac{[v_i^{EH}]_{j-\frac12}}{\Delta y} = 0 \tag{3}$$

and if $p = const$ *then the numerical solution of the Active Flux method with the exact evolution operator remains stationary for all times.*

Corollary (Discrete involution)

There exists a discretization of $\nabla \times \mathbf{v}$ *which remains stationary for any discrete initial data.*

Active Flux for linear acoustics

Theorem ([WB et al., 2019])

If the initial data fulfill the following discretizations of $\operatorname{div} \mathbf{v} = 0$

$$\frac{\{[u^{N}]_{i+\frac12}\}_{j+\frac12}}{\Delta x} + \frac{[\{v^{N}\}_{i+\frac12}]_{j+\frac12}}{\Delta y} = 0 \qquad \frac{\langle [u]_{i\pm1}\rangle_j^{(4)}}{\Delta x} + \frac{[\langle v\rangle_i^{(4)}]_{j\pm1}}{\Delta y} = 0 \tag{2}$$

$$\frac{\langle [u^{EH}]_{i+\frac12}\rangle_j^{(6)}}{\Delta x} + \frac{[\langle v^{EV}\rangle_{i+\frac12}^{(6)}]_{j+\frac12}}{\Delta y} = 0 \qquad \frac{[u^{EV}]_{i-\frac12,j}}{\Delta x} + \frac{[v_i^{EH}]_{j-\frac12}}{\Delta y} = 0 \tag{3}$$

Finite difference discretizations of $\operatorname{div} \mathbf{v} = 0$

and if $p = const$ *then the numerical solution of the Active Flux method with the exact evolution operator remains stationary for all times.*

Corollary (Discrete involution)

There exists a discretization of $\nabla \times \mathbf{v}$ *which remains stationary for any discrete initial data.*

Active Flux is **vorticity preserving** for linear acoustics.

Nonlinear equations

Overview

When applying Active Flux to new systems of equations, an **approximate evolution operator** $\tilde{q}(t,x)$ is required, with at every x

$$\tilde{q}(t,x) = q(t,x) + \mathcal{O}(t^3) \tag{4}$$

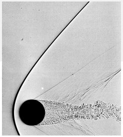
- Approximate evolution operator
  - scalar conservation laws (e.g. **Burgers' equation**)
  - systems of conservation laws in 1-d (e.g. **Euler equations**)
  - balance laws (e.g. **Shallow water equations**)
- What happens if characteristics cross?
- Entropy fix
- Limiting

Approximate evolution operator

- reconstruction globally continuous, but not globally C^1.
  - there is always some finite time interval before the first pair of characteristics will cross
  - short time evolution is often smooth (because often, $\Delta t_{\text{CFL}} < \Delta t_{\text{cross}}$)

but, sometimes, action is required (see later...)

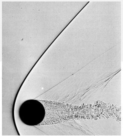
Approximate evolution operator



- reconstruction globally continuous, but not globally C^1.
    - there is always some finite time interval before the first pair of characteristics will cross
    - short time evolution is often smooth (because often, $\Delta t_{\mathrm{CFL}} < \Delta t_{\mathrm{cross}}$)

  but, sometimes, action is required (see later...)
- shocks are not everywhere
    - discontinuities are localized at countably many points/along lines
    - it is perfectly sensible to reconstruct continuously in almost the entire domain
    - almost everywhere in the domain an evolution operator that assumes smoothness of the solution will be right!

Approximate evolution operator



- reconstruction globally continuous, but not globally C^1.
  - there is always some finite time interval before the first pair of characteristics will cross
  - short time evolution is often smooth (because often, $\Delta t_{\mathrm{CFL}} < \Delta t_{\mathrm{cross}}$)

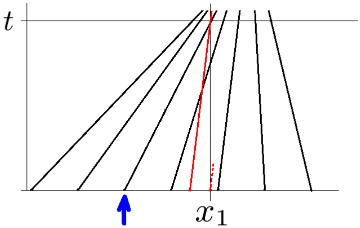
  but, sometimes, action is required (see later...)
- shocks are not everywhere
  - discontinuities are localized at countably many points/along lines
  - it is perfectly sensible to reconstruct continuously in almost the entire domain
  - almost everywhere in the domain an evolution operator that assumes smoothness of the solution will be right!
- how to choose approximate evolution operator?
  - Paradigm: Continuous reconstruction makes it possible *not* to use Riemann solvers
  - Usage of Riemann solvers might even be preventing structure preservation
  - LW/CK/ADER not suited because derivatives of non-differentiable data are required

Evolution operator (scalar equation)

Consider $\partial_t q + \partial_x \left(\frac{q^2}{2}\right) = 0$ (Burgers' equation) with initial data $q(0, x) = q_0(x)$.
Characteristics $x = \xi(t)$ are straight lines with slope $\xi'(t) = q(t, \xi(t))$.

Evolution operator (scalar equation)

Consider $\partial_t q + \partial_x \left(\frac{q^2}{2}\right) = 0$ (Burgers' equation) with initial data $q(0,x) = q_0(x)$.
Characteristics $x = \xi(t)$ are straight lines with slope $\xi'(t) = q(t, \xi(t))$.



Denote by x_1 some fixed location. (To distinguish it from the independent variable x.)

Approximate evolution at x_1 via local linearization means taking the slope $q_0(x_1)$ and tracing back the characteristic:

$$q_0(x_1 - q_0(x_1)t) = q(t, x_1) + \mathcal{O}(t^2) \tag{5}$$

Evolution operator (scalar equation)

Local linearization

$$q_0(x_1 - q_0(x_1)t) = q(t, x_1) + \mathcal{O}(t^2) \quad (6)$$

Higher order:

$$q_0\Big(x_1 - q_0(x_1 - q_0(x_1)t)t\Big) = q(t, x_1) + \mathcal{O}(t^3) \quad (7)$$

This is a **fixpoint iteration** on the characteristic equation [WB, 2019, subm.]

$$x^* = x_1 - q_0(x^*)t \quad (8)$$

Evolution operator (scalar equation)

Local linearization

$$q_0(x_1 - q_0(x_1)t) = q(t, x_1) + \mathcal{O}(t^2) \tag{6}$$

Higher order:

$$q_0\Big(x_1 - q_0(x_1 - q_0(x_1)t)t\Big) = q(t, x_1) + \mathcal{O}(t^3) \tag{7}$$

This is a **fixpoint iteration** on the characteristic equation [WB, 2019, subm.]

$$x^* = x_1 - q_0(x^*)t \tag{8}$$

Evolution operator (scalar equation)

Local linearization

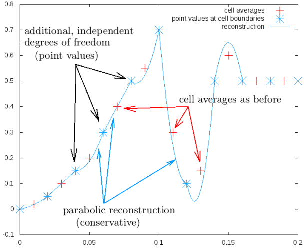
$$q_0(x_1 - q_0(x_1)t) = q(t, x_1) + \mathcal{O}(t^2) \quad (6)$$

Higher order:

$$q_0\Big(x_1 - q_0(x_1 - q_0(x_1)t)t\Big) = q(t, x_1) + \mathcal{O}(t^3) \quad (7)$$

This is a **fixpoint iteration** on the characteristic equation [WB, 2019, subm.]

$$x^* = x_1 - q_0(x^*)t \quad (8)$$



Further approximate solution operators that yield the correct order:

- [Helzel et al., 2019], via LW/CK/ADER
- [Roe, 2017]

However, both involve **derivatives** $q_0'(x_1)$.

Approximate evolution operator

Counterparts to ODE evolution operators:
When solving $y' = f(y)$ for $y(t)$ with $y(0) = y_0$ we have:

- **Linearization**:

$$y(t) = y_0 + tf(y_0) + \mathcal{O}(t^2) \tag{9}$$

Approximate evolution operator

Counterparts to ODE evolution operators:
When solving $y' = f(y)$ for $y(t)$ with $y(0) = y_0$ we have:

- **Linearization**:

$$y(t) = y_0 + tf(y_0) + \mathcal{O}(t^2) \tag{9}$$

- using **LW/CK/ADER**: $y'' = f'(y)y' = f'(y)f(y)$ and thus

$$y(t) = y(0) + ty'(0) + \frac{1}{2}t^2 y''(0) + \mathcal{O}(t^3) \tag{10}$$

$$= y_0 + tf(y_0) + \frac{1}{2}t^2 f'(y_0)f(y_0) + \mathcal{O}(t^3) \tag{11}$$

Approximate evolution operator

Counterparts to ODE evolution operators:
When solving $y' = f(y)$ for $y(t)$ with $y(0) = y_0$ we have:

- **Linearization**:

$$y(t) = y_0 + tf(y_0) + \mathcal{O}(t^2) \tag{9}$$

- using **LW/CK/ADER**: $y'' = f'(y)y' = f'(y)f(y)$ and thus

$$y(t) = y(0) + ty'(0) + \frac{1}{2}t^2 y''(0) + \mathcal{O}(t^3) \tag{10}$$

$$= y_0 + tf(y_0) + \frac{1}{2}t^2 f'(y_0)f(y_0) + \mathcal{O}(t^3) \tag{11}$$

- using **Runge-Kutta**:

$$\text{predictor: } \tilde{y} = y_0 + tf(y_0) \tag{12}$$

$$y(t) = y_0 + t\frac{f(y_0) + f(\tilde{y})}{2} = y_0 + t\frac{f(y_0) + f(y_0 + tf(y_0))}{2} \tag{13}$$

The evolution operators that are presented here are rather "like Runge-Kutta", than LW/CK/ADER: They generate the higher order terms in the Taylor series through convolution / predictor-corrector steps.

Approximate evolution operator

Counterparts to ODE evolution operators:
When solving $y' = f(y)$ for $y(t)$ with $y(0) = y_0$ we have:

- **Linearization**:

$$y(t) = y_0 + tf(y_0) + \mathcal{O}(t^2) \tag{9}$$

- using **LW/CK/ADER**: $y'' = f'(y)y' = f'(y)f(y)$ and thus

$$y(t) = y(0) + ty'(0) + \frac{1}{2}t^2 y''(0) + \mathcal{O}(t^3) \tag{10}$$

$$= y_0 + tf(y_0) + \frac{1}{2}t^2 f'(y_0)f(y_0) + \mathcal{O}(t^3) \tag{11}$$

- using **Runge-Kutta**:

predictor: $\tilde{y} = y_0 + tf(y_0)$ (12)

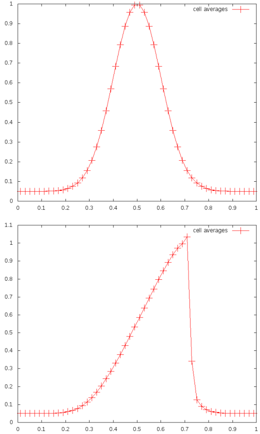
$$y(t) = y_0 + t\frac{f(y_0) + f(\tilde{y})}{2} = y_0 + t\frac{f(y_0) + f(y_0 + tf(y_0))}{2} \tag{13}$$

No derivative required!

The evolution operators that are presented here are rather "like Runge-Kutta", than LW/CK/ADER: They generate the higher order terms in the Taylor series through convolution / predictor-corrector steps.

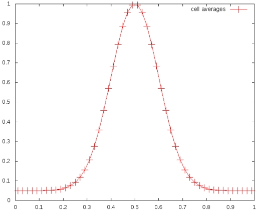
Approximate evolution operator

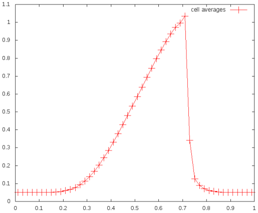
Burgers' equation



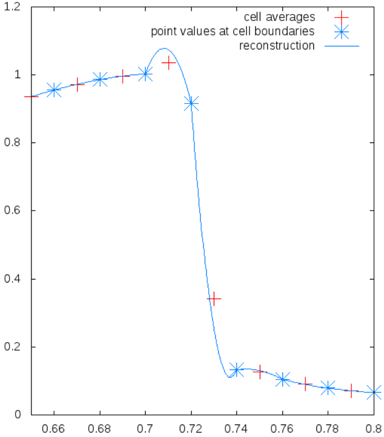
Approximate evolution operator

Burgers' equation





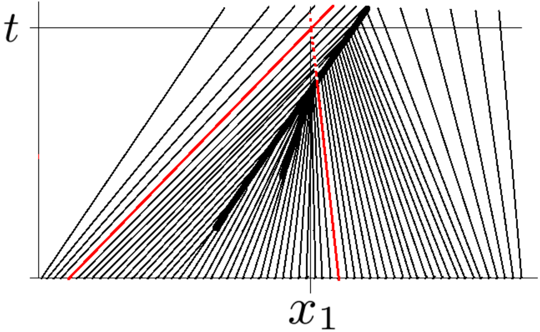
[WB, 2019]



Mind that this scheme is **third order** and we do not use any limiter here!

Approximate evolution operator

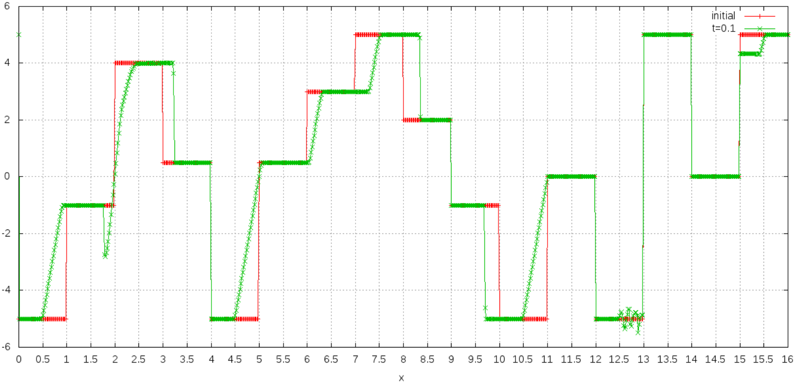
What happens if characteristics cross?



For scalar equations the selection of the correct characteristic can be achieved using the Lax-Hopf formula (see [Qiu & Shu, 2008]).
Here, it is suggested to use *the quickest characteristic*, out of the two obtained by initializing the fixpoint iteration at $x_1 \pm \Delta x$.

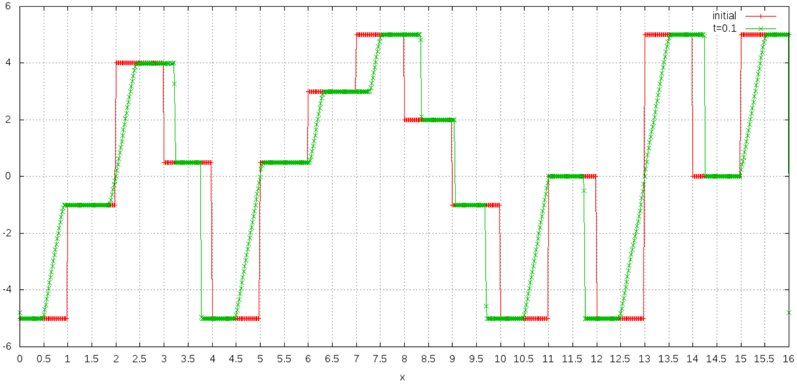
Approximate evolution operator

No modification concerning crossing characteristics



Approximate evolution operator

Modification from [WB, 2019]



Approximate evolution operator (systems)

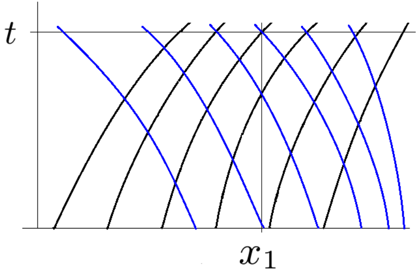
How to generalize to **systems**? Consider characteristics.
Consider a nonlinear system which admits characteristic variables (e.g. for a 3×3 system):

$$\partial_t Q_1 + \lambda_1(Q_1, Q_2, Q_3)\partial_x Q_1 = 0 \tag{14}$$

$$\partial_t Q_2 + \lambda_2(Q_1, Q_2, Q_3)\partial_x Q_2 = 0 \tag{15}$$

$$\partial_t Q_3 + \lambda_3(Q_1, Q_2, Q_3)\partial_x Q_3 = 0 \tag{16}$$

Here, $\{\lambda_1, \dots, \lambda_M\}$ are the eigenvalues of the Jacobian f' in $\partial_t q + f(q) = 0$.



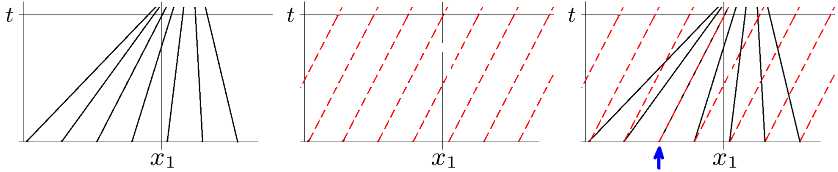
Approximate evolution operator (systems)

Philosophy: When looking for $q(t, x_1)$, find an **equivalent linear problem** (different for each x_1). Observe that the last step of the evolution operator is

$$q(t, x_1) \simeq q_0(x_1 - \lambda^* t) \tag{17}$$

which is the solution to the IVP at x_1 of

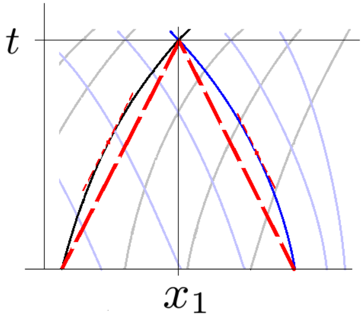
$$\begin{cases} \partial_t q + \lambda^* \partial_x q &= 0 \\ q(0, x) &= q_0(x) \end{cases} \tag{18}$$



$\partial_t q + \partial_x f(q) = 0$ $\quad$ $\partial_t q + \lambda^* \partial_x q = 0$

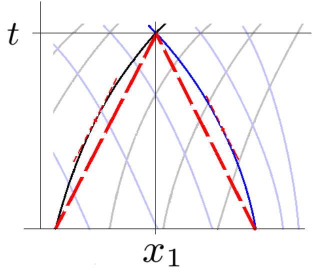
Approximate evolution operator (systems)

Systems case:



Observe that neither the speed of the characteristic at the foot point, nor at the top is correct. We need **"average" speed of the characteristic** – but only to sufficient accuracy.

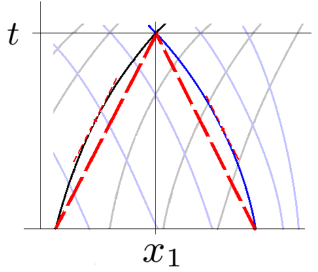
Approximate evolution operator (sys



Consider the characteristic ξ_i associated with λ_i:

$$\xi_i'(t) = \lambda_i(\xi_i(t))$$

Approximate evolution operator (sys



Consider the characteristic ξ_i associated with λ_i:

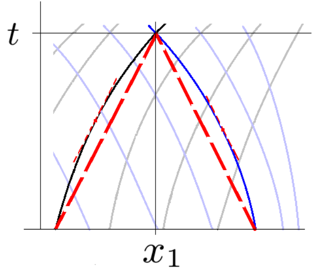
$$\xi_i'(t) = \lambda_i(\xi_i(t))$$

If $Q_i(t, x_1) = Q_{i,0}(x_1 - \lambda_i^* t)$ then we must have

$$\lambda_i^* = \frac{\xi(t) - \xi(0)}{t} = \frac{1}{t}\int_0^t \mathrm{d}\tau\, \xi'(\tau) = \frac{1}{t}\int_0^t \mathrm{d}\tau\, \lambda_i\Big(\xi_i(\tau)\Big) \tag{20}$$

Of course, $\lambda_i(\xi_i(\tau)) \equiv \lambda_i\Big(Q_1(\tau, \xi_i(\tau), \ldots, Q_M(\tau, \xi_i(\tau)))\Big)$.

Approximate evolution operator (sys



Consider the characteristic ξ_i associated with λ_i:

$$\xi_i'(t) = \lambda_i(\xi_i(t))$$

If $Q_i(t, x_1) = Q_{i,0}(x_1 - \lambda_i^* t)$ then we must have

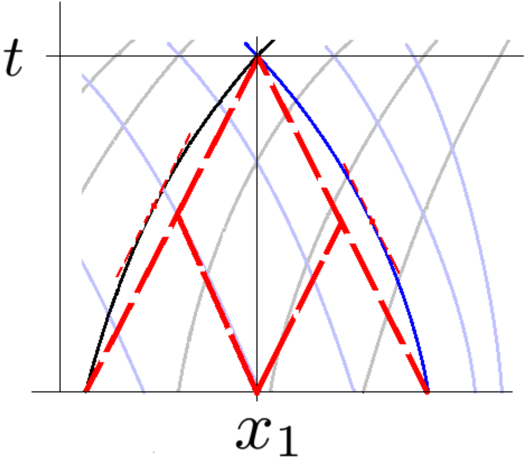
$$\lambda_i^* = \frac{\xi(t) - \xi(0)}{t} = \frac{1}{t}\int_0^t \mathrm{d}\tau\, \xi'(\tau) = \frac{1}{t}\int_0^t \mathrm{d}\tau\, \lambda_i\Big(\xi_i(\tau)\Big) \tag{20}$$

Of course, $\lambda_i(\xi_i(\tau)) \equiv \lambda_i\Big(Q_1(\tau, \xi_i(\tau)), \ldots, Q_M(\tau, \xi_i(\tau))\Big)$.
This expression now needs to be approximated to first order only. Thus, it is natural to use

$$\lambda_i^* \simeq \lambda_i\Big(\xi_i(t/2)\Big) \tag{21}$$

We have $Q_j\left(\frac{t}{2}, \xi_i\left(\frac{t}{2}\right)\right) \simeq Q_j\left(\frac{t}{2}, x_1 - \frac{t}{2}\lambda_i(x_1)\right) \simeq Q_{j,0}\left(x_1 - \frac{\lambda_i(x_1) + \lambda_j(x_1)}{2} t\right)$.

Approximate evolution operator (systems)



Approximate evolution operator (systems)

Characteristic variables exist e.g. for the isentropic Euler equations (= shallow water equations). In general (e.g. for the full Euler equations) they do not. Then both the eigenvalues and the transformation matrix R in $f' = R\Lambda R^{-1}$ need to be predicted. In particular:

$$q_\beta^{(i)} := \sum_{k,\alpha=1}^{m} F_{\beta\alpha}^{(k)}(x) q_{\alpha,0}\left(x - t\frac{\lambda_i(x) + \lambda_k(x)}{2}\right) \tag{22}$$

where $F^{(k)}$ is the projector onto the k-th eigenspace.
Then use

$$\lambda_i^* := \lambda_i(q^{(i)}) \qquad\qquad R_{ij}^* := R(q^{(i)}) \tag{23}$$

The algorithm cannot be given the simple geometric interpretation any more, but the idea remains the same.

[WB, 2019]

Approximate evolution operator

For linear problems,

- the evolution operators presented here are exact after one step for **any** initial data
- LW/CK/ADER is exact after one step for **linear** initial data

Philosophy: We do not need to construct approximations for linear problems, because they can be solved exactly straight away.

Euler equations

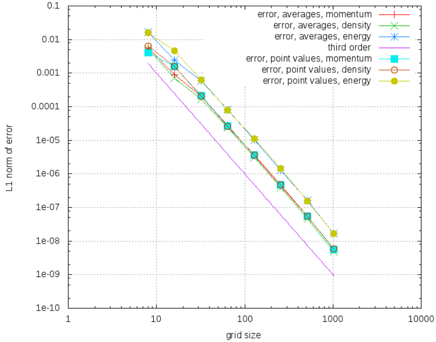


Figure: *Left*: **Third order convergence** of the numerical solution on both point values and averages, for momentum ϱv, density ϱ and energy e. The lines virtually lie on top of each other indicating comparable error. *Right*: Setup and numerical solution for $\Delta x = 1/100$ showing point values.

Euler equations

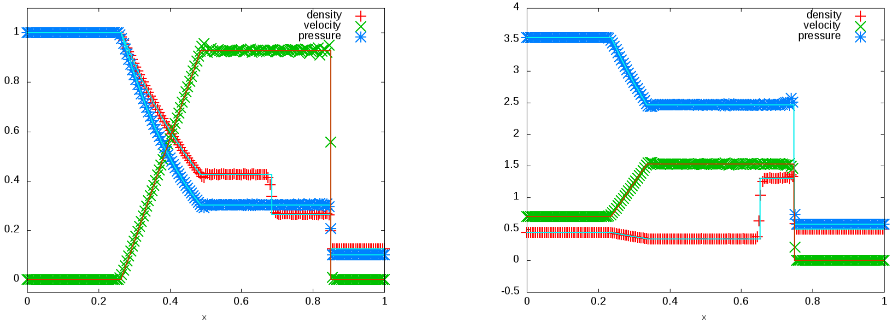


Figure: Riemann problem setups for the full Euler equations. *Left*: **Sod's test problem** [Sod, 1978]. *Right*: **Lax's test problem** [Lax, 1954]. Solid lines show the exact solution.

Euler equations

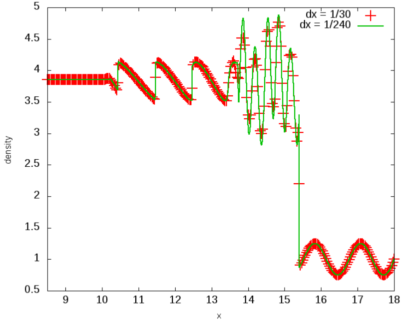


Figure: Interaction between **shock and sound wave** [Shu & Osher, 1989] on grids with $\Delta x = 1/30$ (crosses) and 1/240 (solid line). *Left*: Density. *Right*: Pressure.

Approximate evolution operator (balance laws)

Consider

$$\partial_t Q_i + \lambda_i(Q_1, \ldots, Q_M)\partial_x Q_i = S_i(x; Q_1, \ldots, Q_M) \qquad i = 1, \ldots, M \qquad (24)$$

Approximate evolution operator (balance laws)

Consider

$$\partial_t Q_i + \lambda_i(Q_1, \ldots, Q_M)\partial_x Q_i = S_i(x; Q_1, \ldots, Q_M) \qquad i = 1, \ldots, M \tag{24}$$

Then, similarly, an approximate evolution operator yielding a third order Active Flux method is given by

$$Q_i(t, x) \simeq Q_{i,0}(x - \lambda_i^* t) + tS_i^* \tag{25}$$

with

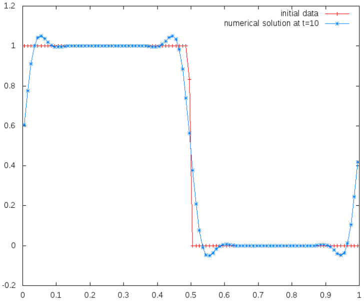
$$\lambda_i^* := \lambda_i \left(Q_1 \left(\frac{t}{2}, x - \lambda_i \frac{t}{2} \right), \ldots, Q_M \left(\frac{t}{2}, x - \lambda_i \frac{t}{2} \right) \right) + \mathcal{O}(t^2) \tag{26}$$

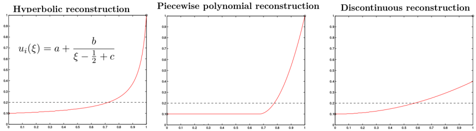
$$S_i^* := S_i \left(x - \lambda_i \frac{t}{2}; Q_1 \left(\frac{t}{2}, x - \lambda_i \frac{t}{2} \right), \ldots, Q_M \left(\frac{t}{2}, x - \lambda_i \frac{t}{2} \right) \right) + \mathcal{O}(t^2) \tag{27}$$

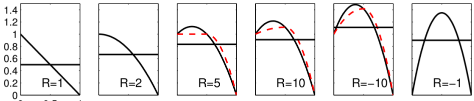
and

$$Q_j \left(\frac{t}{2}, x - \lambda_i \frac{t}{2} \right) \simeq Q_{j,0} \left(x - \frac{\lambda_i + \lambda_j}{2} t \right) + \frac{\Delta t}{2} S_j \left(x; Q_{1,0}(x), \ldots, Q_{M,0}(x) \right)$$

Limiting





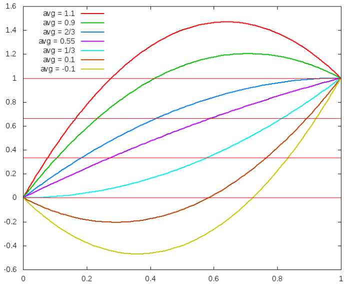


Several suggestions exist in the literature:

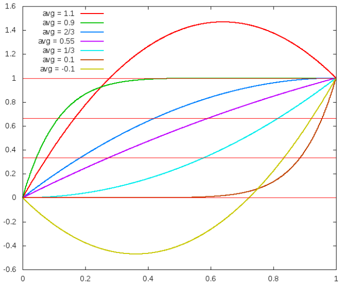
- Modifying the point values/introducing discontinuities:
  - Eymann, Roe, 2011
  - Eymann, 2013: modify point values (Burgers' equation)
  - Helzel et al., 2019: extremum at cell boundary
  - ...
- Continuous reconstructions:
  - Roe et al. 2015/Maeng 2017: joining several parabolas
  - Helzel et al. 2019: hyperbola
  - ...

Limiting

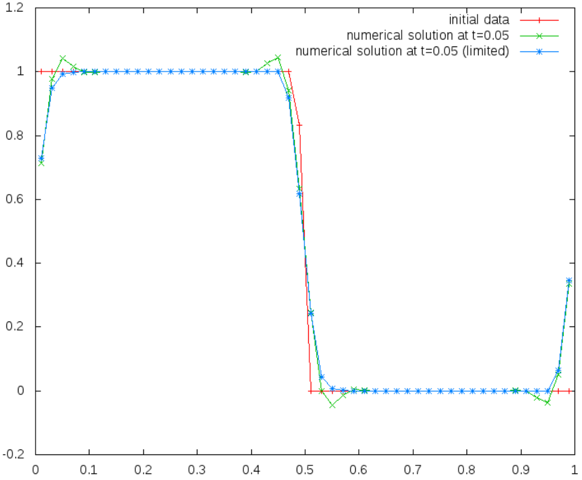
Parabolic:



Power law limiting: [WB, 2019]

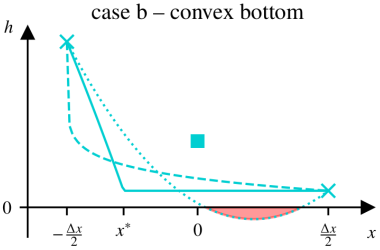


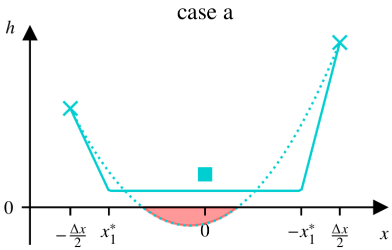
Limiting

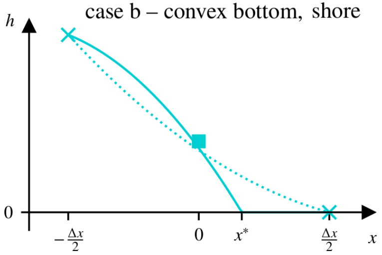

initial data
numerical solution at t=0.05
numerical solution at t=0.05 (limited)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Non-negativity preserving reconstruction

Preserving **non-negative water height** for the shallow water equations:







[WB and Berberich, 2020]

Well-balanced methods

Acoustics with gravity

$$\partial_t \varrho + \partial_x v = 0 \tag{28}$$

$$\partial_t v + \partial_x p = \varrho g \qquad g \in \mathbb{R} \tag{29}$$

$$\partial_t p + c^2 \partial_x v = 0 \qquad c \in \mathbb{R}^+ \tag{30}$$

Acoustics with gravity

$$\partial_t \varrho + \partial_x v = 0 \tag{28}$$

$$\partial_t v + \partial_x p = \varrho g \qquad g \in \mathbb{R} \tag{29}$$

$$\partial_t p + c^2 \partial_x v = 0 \qquad c \in \mathbb{R}^+ \tag{30}$$

Theorem (Stationarity preservation with exact evolution)

If the discrete data fulfill

$$\bar\varrho_i = \frac{\varrho_{i+\frac12} + \varrho_{i-\frac12}}{2} \qquad \frac{p_{i+\frac12} - p_{i-\frac12}}{\Delta x} = g\frac{\varrho_{i-\frac12} + \varrho_{i+\frac12}}{2} \tag{31}$$

$$\frac{\bar p_{i+\frac32} - \bar p_{i+\frac12}}{\Delta x} = g\frac{\varrho_{i+\frac32} + 4\varrho_{i+\frac12} + \varrho_{i-\frac12}}{6} \tag{32}$$

and Active Flux with the exact evolution operator for (28)–(30) *is used, then the numerical solution remains stationary.*

The proof involves the discrete Fourier transform for showing the stationarity of point values. Then, stationarity of averages is checked.

[WB et al., 2020]

Acoustics with gravity

Make sure that, if the data fulfill those discrete relations, the point values remain stationary for the **approximate evolution operator** as well!

Theorem (Stationarity preservation with approximate evolution)

If the approximate evolution operator is modified by adding the term

$$\frac{\alpha g^2}{4} \frac{\varrho_{i+\frac{1}{2}} - \varrho_{i-\frac{1}{2}}}{\Delta x} t^3 \tag{33}$$

to the velocity evolution, then

i) *its accuracy is not changed*

ii) *it becomes stationarity preserving / well-balanced with the same discrete stationary states as the exact evolution operator.*

[WB et al., 2020]

Acoustics with gravity

Make sure that, if the data fulfill those discrete relations, the point values remain stationary for the **approximate evolution operator** as well!

Theorem (Stationarity preservation with approximate evolution)

If the approximate evolution operator is modified by adding the term

$$\frac{\alpha g^2}{4} \frac{\varrho_{i+\frac{1}{2}} - \varrho_{i-\frac{1}{2}}}{\Delta x} t^3 \tag{33}$$

to the velocity evolution, then

1. *its accuracy is not changed*
2. *it becomes stationarity preserving / well-balanced with the same discrete stationary states as the exact evolution operator.*

[WB et al., 2020]

Shallow water equations

$$\partial_t h + \partial_x m = 0 \qquad h : \mathbb{R}_0^+ \times \mathbb{R} \to \mathbb{R}^+ \tag{34}$$

$$\partial_t m + \partial_x \left(\frac{m^2}{h} + \frac{1}{2} g h^2 \right) = -gh\partial_x b \qquad m : \mathbb{R}_0^+ \times \mathbb{R} \to \mathbb{R},\ g \in \mathbb{R} \tag{35}$$

The **exact evolution operator is well-balanced** if the lake at rest ($h + b =$ const) is reconstructed exactly.

Shallow water equations

Well-balancing strategy for an approximate solution operator: Assume an approximate evolution at x_1 to be $q_1(t, x_1) = (h_1(t, x_1), m_1(t, x_1))$.

1. Compute

$$W := h(0, x_1) + b(x_1) \tag{36}$$

(If the data are actually a lake at rest, W is the constant water level.)

Shallow water equations

Well-balancing strategy for an approximate solution operator: Assume an approximate evolution at x_1 to be $q_1(t, x_1) = (h_1(t, x_1), m_1(t, x_1))$.

1. Compute

$$W := h(0, x_1) + b(x_1) \tag{36}$$

(If the data are actually a lake at rest, W is the constant water level.)

2. Apply the approximate evolution operator to initial data $h_0(x) = W - b(x)$, $v_0(x) = 0$ and denote the solution at x_1 by $\tilde{h}(t, x_1)$, $\tilde{m}(t, x_1)$. (Clearly, any actual time evolution is entirely spurious.)

Shallow water equations

Well-balancing strategy for an approximate solution operator: Assume an approximate evolution at x_1 to be $q_1(t, x_1) = (h_1(t, x_1), m_1(t, x_1))$.

1. Compute

$$W := h(0, x_1) + b(x_1) \tag{36}$$

(If the data are actually a lake at rest, W is the constant water level.)

2. Apply the approximate evolution operator to initial data $h_0(x) = W - b(x)$, $v_0(x) = 0$ and denote the solution at x_1 by $\tilde{h}(t, x_1)$, $\tilde{m}(t, x_1)$. (Clearly, any actual time evolution is entirely spurious.)

3. The well-balanced approximate solution $(h^{\mathrm{wb}}(t, x_1), m^{\mathrm{wb}}(t, x_1))$ at x_1 is obtained by subtracting the spurious evolution

$$m^{\mathrm{wb}}(t, x_1) := m_1(t, x_1) - h_1(t, x_1)\frac{\tilde{m}(t, x_1)}{\tilde{h}(t, x_1)} \tag{37}$$

$$h^{\mathrm{wb}}(t, x_1) := h_1(t, x_1) - \left(\tilde{h}(t, x_1) - h(0, x_1)\right) \tag{38}$$

[WB and Berberich, 2020]

Shallow water equations

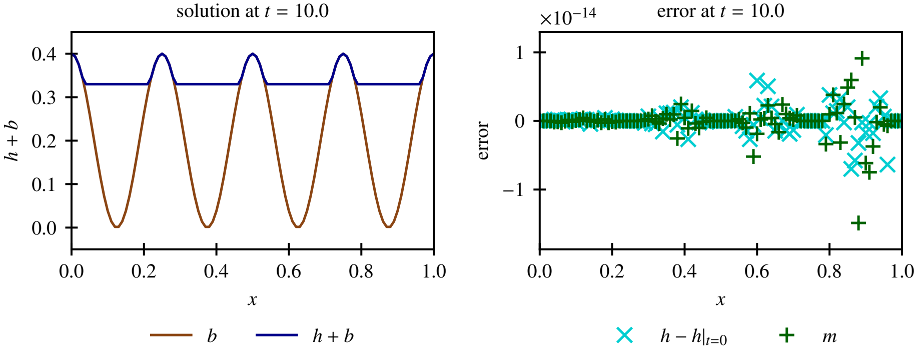


Figure: Demonstration of the **well-balanced property in presence of partially dry cells**. *Left*: Setup with four lakes at rest. Point values of $h + b$ are shown. *Right*: Errors of the point values of the numerical solution at $t = 10$.

Shallow water equations

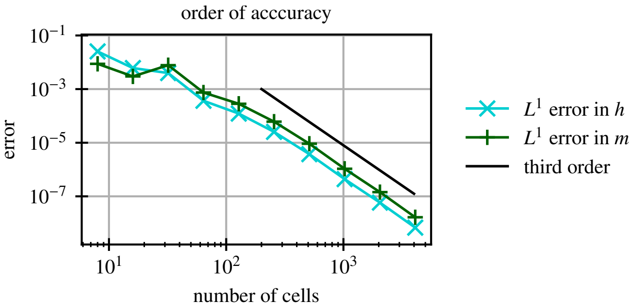
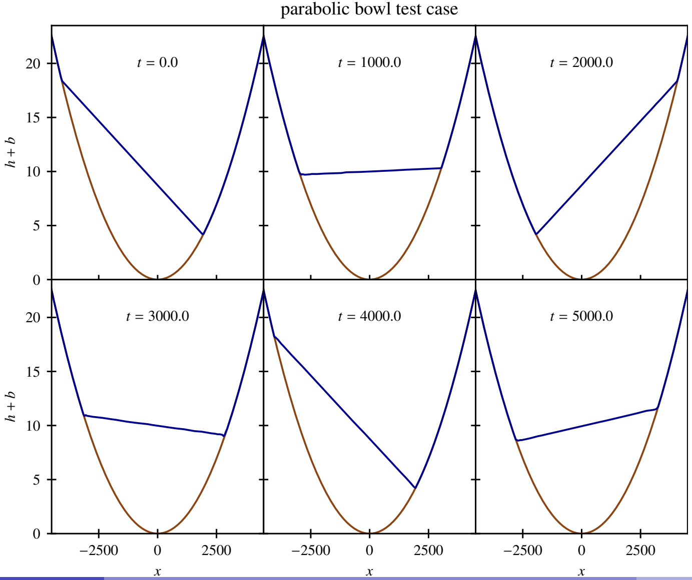


Figure: **Convergence** of a Gaussian wave on cosine-shaped bottom. The L^1 error of the point values is shown.

Shallow water equations



Shallow water equations

Tsunami run-up onto a plane beach

(benchmark problem 1 from the 3rd International Workshop on Long-Wave Runup Models, 2004, Wrigley Marine Science Center

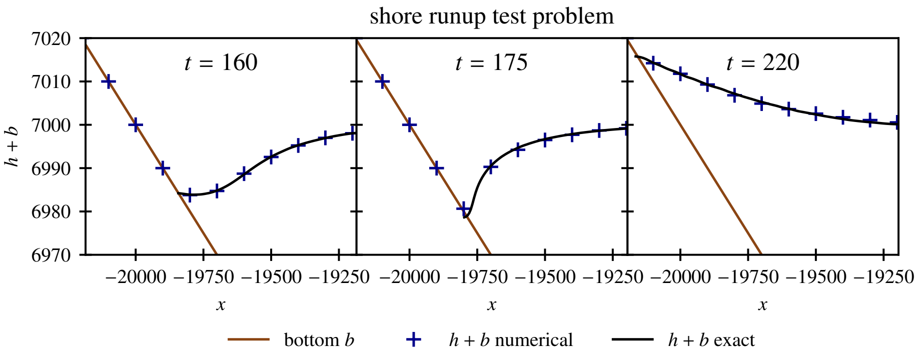


Figure: Point values of $h + b$ are shown together with the analytical solution (solid line).

Summary

- Active Flux:
  - **high order**
  - **compact stencil**
  - **continuous reconstruction**
  - **structure preserving** in many cases

Summary

- Active Flux:
  - **high order**
  - **compact stencil**
  - **continuous reconstruction**
  - **structure preserving** in many cases
- Approximate evolution operators for nonlinear problems with characteristics
  - high order achieved **without using derivatives** of initial data
  - "tree" structure to approximate **curved characteristics**
  - accounting for **crossing characteristics**, entropy fix

Summary

- Active Flux:
  - **high order**
  - **compact stencil**
  - **continuous reconstruction**
  - **structure preserving** in many cases
- Approximate evolution operators for nonlinear problems with characteristics
  - high order achieved **without using derivatives** of initial data
  - "tree" structure to approximate **curved characteristics**
  - accounting for **crossing characteristics**, entropy fix
- Applications
  - linear acoustics (vorticity/stationarity preserving)
  - linear acoustics with gravity (well-balanced)
  - **full Euler equations** (1D)
  - **Shallow water equations** (with wetting/drying, well-balanced, 1D)

Summary

- Active Flux:
  - **high order**
  - **compact stencil**
  - **continuous reconstruction**
  - **structure preserving** in many cases
- Approximate evolution operators for nonlinear problems with characteristics
  - high order achieved **without using derivatives** of initial data
  - "tree" structure to approximate **curved characteristics**
  - accounting for **crossing characteristics**, entropy fix
- Applications
  - linear acoustics (vorticity/stationarity preserving)
  - linear acoustics with gravity (well-balanced)
  - **full Euler equations** (1D)
  - **Shallow water equations** (with wetting/drying, well-balanced, 1D)
- Future work:
  - extension to multi-d systems of nonlinear equations (characteristic cones!)
  - and further study of structure preservation

Thank You!

wasilij.barsukow@math.uzh.ch
http://user.math.uzh.ch/barsukow

WB, Jonathan Hohm, Christian Klingenberg, Philip L. Roe: *The active flux scheme on Cartesian grids and its low Mach number limit*, 2019, J. Sci. Comp. 81(1): 594-622 (arXiv:1812.01612)

WB: *Stationarity preserving schemes for multi-dimensional linear systems*, Math.Comp. (2019) 88(318): 1621-1645 (arXiv:1811.11766)

WB: *The active flux scheme for nonlinear problems, 2019 submitted*

WB, Jonas P. Berberich, Christian Klingenberg: *On the active flux scheme for hyperbolic PDEs with source terms*, 2020 submitted

WB, Jonas P. Berberich: *A well-balanced Active Flux scheme for the shallow water equations with wetting and drying*, 2020 submitted

Amadori, D. and Gosse, L. (2015).

Error Estimates for Well-Balanced Schemes on Simple Balance Laws: One-Dimensional Position-Dependent Models.

BCAM Springer Briefs in Mathematics, Springer.

Dellacherie, S. (2010).

Analysis of Godunov type schemes applied to the compressible Euler system at low Mach number.

Journal of Computational Physics, 229(4):978–1016.

Franck, E. and Gosse, L. (2018).

Stability of a Kirchhoff–Roe scheme for two-dimensional linearized Euler systems.

ANNALI DELL'UNIVERSITA' DI FERRARA, 64(2):335–360.

Helzel, C., Kerkmann, D., and Scandurra, L. (2019).

A new ADER method inspired by the active flux method.

Journal of Scientific Computing, 80(3):1463–1497.

Jeltsch, R. and Torrilhon, M. (2006).

On curl-preserving finite volume discretizations for shallow water equations.

BIT Numerical Mathematics, 46(1):35–53.

Lukacova-Medvidova, M., Morton, K., and Warnecke, G. (2000).

Evolution Galerkin methods for hyperbolic systems in two space dimensions.

Mathematics of Computation of the American Mathematical Society, 69(232):1355–1384.

Lung, T. and Roe, P. (2014).

Toward a reduction of mesh imprinting.

International Journal for Numerical Methods in Fluids, 76(7):450–470.

Mishra, S. and Tadmor, E. (2009).
Constraint preserving schemes using potential-based fluxes II. genuinely multi-dimensional central schemes for systems of conservation laws.
ETH preprint, (2009-32).

Morton, K. W. and Roe, P. L. (2001).
Vorticity-preserving Lax-Wendroff-type schemes for the system wave equation.
SIAM Journal on Scientific Computing, 23(1):170–192.

Roe, P. (2017).
Is discontinuous reconstruction really a good idea?
Journal of Scientific Computing, 73(2-3):1094–1114.

Torrilhon, M. and Fey, M. (2004).
Constraint-preserving upwind methods for multidimensional advection equations.
SIAM journal on numerical analysis, 42(4):1694–1728.

WB (2019).
The active flux scheme for nonlinear problems.
subm.

WB and Berberich, J. (2020).
A well-balanced active flux scheme for the shallow water equations with wetting and drying.
submitted.

WB, Berberich, J., and Klingenberg, C. (2020).
On the active flux scheme for hyperbolic PDEs with source terms.
submitted.

WB, Hohm, J., Klingenberg, C., and Roe, P. L. (2019).
The active flux scheme on Cartesian grids and its low Mach number limit.
Journal of Scientific Computing, 81(1):594–622.

WB and Klingenberg, C. (2018).
Exact solution and a truly multidimensional Godunov scheme for the acoustic equations.
submitted.