Vorticity-preserving projection method for the Shallow Water Equations
Seminar Structure preserving discretizations of PDE

Sophie Haug

ETH Zurich

May 2011
1. The Shallow Water Equations
2. Conservation Laws and the FV-approach
3. Vorticity preservation
4. Vorticity preserving method for the wave equation
5. Vorticity preserving method for the Shallow Water Equations
Tsunami wave

Picture from: olc.spsd.sk.ca
The SWE describe shape and velocity of surface waves for shallow fluids, e.g.

- Tsunami waves near shores
- Rossby waves: large-scale waves important in climatology and oceanography
- on small scale: water waves in a bathtub

**assumption**

\[ h \ll L, \text{ where } L \text{ is the wavelength} \]
therefore: assume velocity in vertical direction to be constant
The Shallow Water Equations

The 1-D SWE are given by:

\[ h_t + (hu)_x = 0 \quad (1) \]
\[ (hu)_t + \left( \frac{1}{2}gh^2 \right)_x = 0 \quad (2) \]

The 2-D SWE are given by:

\[ h_t + (hu)_x + (hv)_y = 0 \quad (3) \]
\[ (hu)_t + (hu^2 + \frac{1}{2}gh^2)_x + (huv)_y = 0 \quad (4) \]
\[ (hv)_t + (huv)_x + (hv^2 + \frac{1}{2}gh^2)_y = 0 \quad (5) \]

\( h \): height of fluid column, \((u, v) \) (\( u \) resp.): velocity field
An illustration for the 1-D case
The SWE as a conservation law

- 1-D case: Let $U = [h, hu]^T$ and $f = [hu, \frac{1}{2}gh^2 + hu^2]^T$
- 2-D case: Let

$$U = [h, hu, hv]^T$$

$$f = [hu, hu^2 + \frac{1}{2}gh^2, huv]^T$$

$$g = [hv, huv, hv^2 + \frac{1}{2}gh^2]^T$$

The SWE can be written as:

$$U_t + f(U)_x = 0 \quad 1\text{-D case} \quad (6)$$

$$U_t + f(U)_x + g(U)_y = 0 \quad 2\text{-D case} \quad (7)$$

Eigenvalues of $f', g'$ are distinct and real $\rightarrow$ SWE represent hyperbolic conservation law
Remark: We recall the important concepts of theory of conservation laws looking at the 1-D case only

- consider:
  \[ u_t + f(u)_x = 0 \quad (8) \]
  where \( u(x, t) : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R} \)

- Recall: smooth solutions of (8) might not exist \(\rightarrow\) look for weak solutions

**Definition (weak solution of a conservation law)**

\( u \in L^1(\mathbb{R} \times \mathbb{R}^+) \) is a weak solution of (8) if \( \forall \phi \in C_c^1(\mathbb{R} \times \mathbb{R}^+) \), we have:

\[
\int_{\mathbb{R} \times \mathbb{R}^+} u\phi_t + f(u)\phi_x \, dx \, dt + \int_{\mathbb{R}} u(x, 0)\phi(x, 0) \, dx = 0.
\]
• problem: weak solutions for (8): in general not unique
• → impose entropy conditions

**Definition (entropy function, entropy flux)**

Let a scalar conservation law (8) be given. Let $E : \mathbb{R} \to \mathbb{R}$ be a strictly convex function and

$$Q(u) := \int_0^u f'(s)E(s)ds$$

Then $(E, Q)$ is an entropy pair for (8) $E$ is called *entropy function* and $Q$ the *corresponding entropy flux*. 
Definition (entropy solution)

A weak solution \(u\) of (8) is said to be an **entropy solution** for an entropy pair \((E, Q)\) if the **entropy inequality**:

\[
E(u)_t + Q(u)_x \leq 0 \tag{9}
\]

holds in the weak sense, i.e.

\[
\int_{\mathbb{R}^+} \int_{\mathbb{R}} E(u) \phi_t + Q(u) \phi_x \, dx \, dt + \int_{\mathbb{R}} E(x, 0) \phi(x, 0) \, dx \geq 0, \forall \phi \in C^1_c(\mathbb{R} \times \mathbb{R}^+) \text{ and } \text{Im}(\phi) \geq 0
\]

- It can be shown: for the SCL (8) \(\exists!\) weak solution that satisfies (9) for a specific class of entropy pairs \((E, Q)\).
- Forcing (9) to hold: ”physically correct” solution is approximated.
The entropy function we consider is the total energy.

- **1-D**

\[
E(h, hu) = \frac{1}{2}(hu^2 + gh^2)
\]

\[
Q(h, hu) = \frac{1}{2}(hu^3 + guh^2)
\]

- **2-D**

\[
E(h, hu, hv) = \frac{1}{2}(hu^2 + hv^2 + gh^2)
\]

\[
H(h, hu, hv) = \frac{1}{2}(hu^3 + hu^2v + gh^2u)
\]

\[
K(h, hu, hv) = \frac{1}{2}(hu^2v + hv^3 + gh^2v)
\]

with entropy fluxes \(H\) for \(f\) and \(K\) for \(g\).

Note: \(E\) is the sum of kinetic energy and potential energy. The corresponding entropy inequality for the 2-D case then reads:

\[
E(h, hu, hv)_t + H(h, hu, hv)_x + K(h, hu, hv)_y \leq 0. \tag{10}
\]
The Finite Volume approach I

2-D

• Assume a cartesian mesh to be given. Divide computational domain into grid cells:

\[ l_{i,j} = \left[ x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}} \right] \times \left[ y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}} \right], \quad x_{i+\frac{1}{2}} := x_i + \frac{\Delta x}{2} \]

• Aim: Approximate cell averages:

\[ U^n_{i,j} := \frac{1}{\Delta x \Delta y} \int_{l_{i,j}} U(x, t^n) \]

• The semidiscrete Finite Volume scheme:

\[ \frac{d}{dt} U_{i,j} = - \frac{1}{\Delta x} \left( F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j} \right) - \frac{1}{\Delta y} \left( G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}} \right) \]
\[
2y_j - \frac{1}{2} + 1 - \frac{1}{2} \leq 2y_{j+1} - \frac{1}{2} - \frac{1}{2} \\
\frac{1}{2} x_{i-1} + 1 + \frac{1}{2} \leq \frac{1}{2} x_{i+1} - \frac{1}{2} - \frac{1}{2} \\
U_{i,j} F_{i+\frac{1}{2},j} + G_{i,j+\frac{1}{2}} \\
x_i - \frac{1}{2} \leq x_i \leq x_i + \frac{1}{2} \\
y_j - \frac{1}{2} \leq y_j \leq y_j + \frac{1}{2}
\]
• Numerical fluxes \( F_{i+\frac{1}{2},j} = F(U_{i,j}, U_{i+1,j}) \) and 
\( G_{i,j+\frac{1}{2}} = G(U_{i,j}, U_{i,j+1}) \), which are approximate solvers of the Riemann problems at cell interfaces:

\[
\begin{align*}
U_t + f(U)_x &= 0 \\
U(x, 0) &= \begin{cases} 
U_{i,j} & \text{if } x < x_{i+\frac{1}{2}}, \\
U_{i+1,j} & \text{if } x > x_{i+\frac{1}{2}}.
\end{cases}
\end{align*}
\]

\[
\begin{align*}
U_t + g(U)_y &= 0 \\
U(x, 0) &= \begin{cases} 
U_{i,j} & \text{if } y < y_{j+\frac{1}{2}}, \\
U_{i,j+1} & \text{if } y > y_{j+\frac{1}{2}}.
\end{cases}
\end{align*}
\]

respectively.

• for time integration in (11), a suitable ODE-solver is used.
We apply a standard finite volume approach to solve the 2-D SWE using first the Rusanov flux and secondly the Roe flux. Compare the approximated height $h$ with the exact solution at $t=100$:

(a) Exact solution  
(b) Rusanov  
(c) Roe

Figure: height at $t=100$

→ obviously the numerical approximations are not satisfying  
→ what is the problem??

3U.Fjordholm and S.Mishra:Vorticity preserving finite volume schemes for the shallow water equations
**Definition (vorticity)**

Let $\vec{v}$ be the velocity field in a fluid. We define the vorticity:

$$\omega := \text{curl}(\vec{v})$$

for a 2-dimensional velocity field, we have: $\omega = (v_x - u_y)\vec{e}_3$, $\vec{v} = (u, v)$, where $\vec{e}_3$ is the standard vector orthogonal to $u, v$ in 3-D space.

**Theorem (vorticity transport equation)**

Let $h, (u, v)$ be a smooth solution of the SWE. Then the vorticity satisfies the following identity:

$$\omega_t + (u\omega)_x + (v\omega)_y = 0$$  \hspace{1cm} (11)

→ AIM: A new vorticity preserving scheme for the SWE
A vortex
Theorem (Hodge-Helmholtz decomposition)

Let \( \vec{v} \) be a vector field defined on a bounded domain \( \Omega \subset \mathbb{R}^3 \). Then

\[
\vec{v} = \text{curl}(\phi) + \nabla(\psi)
\]  

for a vector valued function \( \phi \) and a scalar function \( \psi \). This decomposition is called the Hodge (-Helmholtz) decomposition of \( \vec{v} \).

Note that we have:

\[
\text{curl}(\vec{v}) = \text{curl}(\text{curl}(\phi))
\]

\[
\text{div}(\vec{v}) = \Delta(\psi)
\]

as \( \text{curl}(\nabla) = 0 \), \( \text{div}(\text{curl}) = 0 \).
Consider a vector field $\vec{v}$ and its Hodge decomposition (12).
We first design a scheme for solving (12) for the rotation-free part $\nabla \psi$:

1. Applying the curl-operator on both sides of (12) gives:
   \[ \text{curl}(v) = \text{curl}(\text{curl}(\phi)) \]  
   (13)

2. Solve the equation (13) for $\phi$ (Remark: curl($v$) is assumed to be known!)

3. Obtain the rotation-free part $\nabla (\psi)$ using:
   \[ \nabla (\psi) = v - \text{curl}(\phi) \]
Consider a scalar conservation law:

\[ U_t + f(U)_x = 0 \]  \hspace{1cm} (14)

where \( U : \mathbb{R}^N \times \mathbb{R}^+ \to \mathbb{R} \).

Aim: we want the solution \( U \) to lie in \( M := \{ v | \text{curl}(v) = 0 \} \).

Algorithm:

1. Prediction step: Given a solution \( U^n \in M \) at \( t^n \), Compute a solution \( \tilde{U}^{n+1} \) at \( t^{n+1} \) of (14) using a suitable Finite Volume method.

2. Solve the (discretized) problem

\[ \text{curl} \left( \text{curl} \left( \phi \right) \right) = \text{curl}(\tilde{U}) \]

3. project \( \tilde{U}^{n+1} \) onto \( M \) and get the solution \( U^{n+1} \):

\[ U^{n+1} = \tilde{U}^{n+1} - \text{curl}(\phi) \]

Remark: To solve the \text{curlcurl} problem above, in the 2-D case: we can manipulate the problem so that it reduces to solving a \( \Delta \)-problem \( \to \) many efficient solvers known!
1. Applying one step of a Finite Volume solver

2. Project predictor solution, which no longer lies in $M_\Gamma$, back onto it

3. Obtain final approximation
Now: Apply this idea to get a vorticity preserving numerical scheme for the SWE
From last time, we know the system wave equation:

\[ \begin{bmatrix} \hbar \\ m_1 \\ m_2 \end{bmatrix}_t + \begin{bmatrix} 0 & c & 0 \\ c & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hbar \\ m_1 \\ m_2 \end{bmatrix}_x + \begin{bmatrix} 0 & 0 & c \\ 0 & 0 & 0 \\ c & 0 & 0 \end{bmatrix} \begin{bmatrix} \hbar \\ m_1 \\ m_2 \end{bmatrix}_y = 0 \]

this is indeed a special case of the 2-D SWE, \( c = \sqrt{gh_0}, \hbar = ch, m_1 = hu, m_2 = hv \) (cf. last time)

Measure of vorticity for the wave equation:

\[ \Gamma := ((m_2)_x - (m_1)_y) \vec{e}_3 \]

(\( \vec{e}_3 \) denoting the standard basis vector in direction orthogonal to \( m_1-/m_2\)-direction)

**Lemma**

*Let \( \Gamma \) defined as above. Then \( \Gamma_t = 0 \), i.e. \( \Gamma(t) = \Gamma_0 = \Gamma(0) \ \forall t > 0 \)*
We give a projection method for the wave equation, following the abstract idea introduced previously:

- **Step 0:** Discretize vorticity: Using some discrete vectorial curl-operator $\text{curl}_d$ define $\Gamma^n_{i,j} := \text{curl}_d((m_1)^n_{i,j}, (m_2)^n_{i,j}, 0)$.

- **Step 1:** Given $U^n$, compute a predictor solution $\tilde{U}^{n+1}$ using a suitable FV-method.

- **Step 2:** Solve problem:

  $$\text{curl}_d \text{curl}_d(\phi_{i,j}) = \tilde{\Gamma}^{n+1}_{i,j} - \Gamma^0_{i,j}$$

- **Step 3:** Projection. Let $\tilde{U}^{n+1}_{i,j} = [\tilde{h}^{n+1}_{i,j}, (\tilde{m}_1)^{n+1}_{i,j}, (\tilde{m}_2)^{n+1}_{i,j}]^T$. Define:

  $$U^{n+1}_{i,j} = [\tilde{h}^{n+1}_{i,j}, (m_1)^{n+1}_{i,j}, (m_2)^{n+1}_{i,j}]^T$$
where

\[ h_{i,j} = \tilde{h}_{i,j} \]

\[ (m_1)_{i,j}^{n+1} = (m_1)_{i,j}^{n+1} - \text{curl}_d(\phi)_{i,j}^1 \]

\[ (m_2)_{i,j}^{n+1} = (m_2)_{i,j}^{n+1} - \text{curl}_d(\phi)_{i,j}^2 \]

\begin{lemma}

The solution \( U_{i,j}^{n+1} = [h_{i,j}^{n+1}, (m_1)_{i,j}^{n+1}, (m_2)_{i,j}^{n+1}] \) computed with the projection method satisfies

\[ \Gamma_{i,j}^{n+1} = \Gamma_{i,j}^0 \]

\end{lemma}
Remark

The projection algorithm is energy stable, i.e. for

\[ E_{i,j} := \frac{1}{2}(\bar{h}_{i,j}^2 + (m_1)^2_{i,j} + (m_2)^2_{i,j}) \]

we have:

\[ \sum_{i,j} E_{i,j} \leq \sum_{i,j} \tilde{E}_{i,j} - 2 \sum_{i,j} \Gamma^0_{i,j} \phi_3, \]

where \( \phi_3 \) is the 3rd component of the vector valued function \( \phi \) in the Hodge decomposition (cf. (12)). Thus if \( \Gamma^0_{i,j} = 0 \), the energy dissipates in time.
Numerical experiment 2: Periodic wave

The method is tested on the following example for the wave equation:
\[ \dot{h} = 0 \quad m_1(x, y) = m_2(x, y) = \cos(\pi(x + y)) - \cos(\pi(x - y)). \]

The initial vorticity is \( \Gamma_0 = 2\pi \sin(\pi(x - y)) \).

As an example, we feature the momentum at \( t=2 \):

\[ (a) \text{ Exact solution} \quad (b) \text{ Rusanov} \quad (c) \text{ Rusanov + projection method} \]

**Figure**: momentum at \( t=2 \)

---

\(^4\)U.Fjordholm and S.Mishra: Vorticity preserving finite volume schemes for the shallow water equations
Extending this to the general 2-D SWE
Note: The velocity is not a conserved measure. However the momentum is a conserved variable in time. Therefore, instead of vorticity we look at the **pseudovorticity**, which is given by:

\[ \Omega := \text{curl}(hu, hv, 0) = \text{curl}(m_1, m_2, 0) = ((m_2)_x - (m_1)_y)e_3 \]

**Theorem (pseudovorticity transport equation)**

The pseudovorticity \( \Omega \) satisfies:

\[
\Omega_t + (u\Omega)_x + (v\Omega)_y + (m_2(u_x + v_y))_x - (m_1(u_x + v_y))_y \\
+ \left( \frac{u^2 + v^2}{2}h_y \right)_x - \left( \frac{u^2 + v^2}{2}h_x \right)_y = 0 \tag{15}
\]
• So far: wave equation, vorticity $\Gamma_t = 0$

• Now we are looking at a different paradigm: Pseudovorticity $\Omega$ must satisfy (15) → Getting the ”right part” of the solution requires more than just solving a curl-curl problem!
A vorticity preserving method for the SWE I

- Step 0: Discretize $\Omega$ using again a discrete vectorial curl operator $\text{curl}_d$ (e.g. based on central differences):

$$\text{curl}_d(((m_1^n)_{i,j}, (m_2^n)_{i,j}, 0)) =: \Omega^n_{i,j}$$

- Step 1: Prediction step: Given $U^n_{i,j}$, using a finite volume solver, compute $\tilde{U}^{n+1}_{i,j}$ and $\tilde{\Omega}^{n+1}_{i,j}$, i.e. the discrete pseudovorticity.

- Step 2: Vorticity estimate. Given $\Omega^n_{i,j}$ we need to obtain an estimate $\Omega^{n+1}_{i,j}$, for the pseudovorticity at $t^{n+1}$: should be an approximate solution of transport equation (15)

- Step 3: Knowing our estimate $\Omega^{n+1}_{i,j}$ from Step 2, use a projection method-step to force the vorticity of the solution computed with the FV-method to have that same value! This is done the same way as for the wave equation, i.e. we solve:

$$\text{curl}_d\text{curl}_d\phi_{i,j} = \tilde{\Omega}^{n+1}_{i,j} - \Omega^{n+1}_{i,j}$$
• Step 4: Projection. As for the wave equation,

\[ \tilde{h}^{n+1}_{i,j} = \widetilde{h}^{n+1}_{i,j} \]

\[ (m_1)^{n+1}_{i,j} = (\widetilde{m}_1)^{n+1}_{i,j} - \text{curl}_d(\phi_{i,j})^1 \]

\[ (m_2)^{n+1}_{i,j} = (\widetilde{m}_2)^{n+1}_{i,j} - \text{curl}_d(\phi_{i,j})^2 \]

**Lemma**

The pseudovorticity of the final approximation \( U^{n+1}_{i,j} \) equals the estimated pseudovorticity \( \Omega^{n+1}_{i,j} \) (computed in step 2).

**Proof.**

cf. wave equation, the proof is analogous.
How to do Step 2: The NT scheme I

We want: approximate the value of $\Omega$ at $t^{n+1}$ given data at $t^n$

- Rewrite (15) as:

$$\Omega_t - f(\Omega, U)_x - g(\Omega, U)_y = 0,$$

where

$$f(\Omega, U) = u\Omega + dm_2 + sh_y, \quad g(\Omega, U) = v\Omega + dm_1 + sh_x,$$

$$d = u_x + v_y, \quad s = \frac{u^2 + v^2}{2}.$$ 

- 2nd order central scheme. Look at the cells $J_{i,j} = [x_i, x_{i+1}) \times [y_j, y_{j+1}),$ centered at $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$(!). approximate:

$$\Omega_{i+\frac{1}{2}, j+\frac{1}{2}}^{n+1} \approx \frac{1}{\Delta x \Delta y} \int_{J_{i,j}} \Omega(x, y, t^{n+1}) dx dy,$$
• perform a piecewise linear reconstruction using as data $\Omega_{i+\frac{1}{2}, j+\frac{1}{2}}^{n+1}$ and slopes $\sigma_{i,j}, \gamma_{i,j}$, which are determined using the MINMOD limiter.

$$\Omega^{n+1}(x, y) := \Omega_{i+\frac{1}{2}, j+\frac{1}{2}}^{n+1} + \sigma_{i,j} (x - x_{i+\frac{1}{2}}) + \gamma_{i,j} (y - y_{j+\frac{1}{2}}) \quad (x, y) \in J_{i,j}$$

• obtain the approximation at the next time level by averaging the pw. linear reconstruction over $I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$:

$$\Omega_{i,j}^{n+1} = \frac{1}{\Delta x \Delta y} \int_{I_{i,j}} \Omega^{n+1}(x, y) dx dy$$
• How to obtain $\Omega_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1}$? Integrating the conservation law gives:

$$
\Omega_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \frac{1}{\Delta x \Delta y} \int_{J_{i,j}} \Omega^n(x, y) dx dy
$$

$$
- \frac{1}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{J_{i,j}} f(\Omega, U)_x + g(\Omega, U)_y dx dy dt
$$

$$
= \frac{1}{\Delta x \Delta y} \left( \int_{x_i}^{x_{i+\frac{1}{2}}} \int_{y_j}^{y_{j+\frac{1}{2}}} \Omega^n(x, y) dx dy 
+ \int_{x_i+\frac{1}{2}}^{x_{i+1}} \int_{y_j}^{y_{j+\frac{1}{2}}} \Omega^n(x, y) dx dy 
+ \int_{x_i+\frac{1}{2}}^{x_{i+1}} \int_{y_{j+\frac{1}{2}}}^{y_{j+1}} \Omega^n(x, y) dx dy 
- \int_{t^n}^{t^{n+1}} \int_{J_{i,j}} f(\Omega, U)_x + g(\Omega, U)_y dx dy dt \right)
$$
How to do Step 2: The NT scheme IV

\[ \approx \frac{1}{4} (\Omega_{n,i,j}^n + \Omega_{n,i+1,j}^n + \Omega_{n,i,j+1}^n + \Omega_{n,i+1,j+1}^n) \]

\[ - \frac{1}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{J_{i,j}} f(\Omega, U)_x + g(\Omega, U)_y \, dxdydt \]

- approximate the flux term:

\[ \frac{1}{\Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{J_{i,j}} f(\Omega, U)_x + g(\Omega, U)_y \, dxdydt \]

by a suitable quadrature rule.

- Note that using this kind of scheme we have:
  - numerical dissipation
  - + NO Riemann problems to solve (!)
We first show a numerical experiment where vorticity transport for a smooth solution is computed and show results for $t=100$. We see that without the projection method the Rusanov method loses almost all of the vortex.

Figure: vorticity at $t=100$
Next we consider the vortex advection equation with discontinuous initial data. We display the results for the height $h$ comparing Rusanov and Rusanov + projection methods. First look at the initial state:

![Image](image.png)

height at $t=0$

---

6U.Fjordholm and S.Mishra: Vorticity preserving finite volume schemes for the shallow water equations
We look at the numerical results at different times. The height-vortex should interact with the shock, transit it and continue to exist. We see that only by using the projection method we capture this behaviour.

7. U. Fjordholm and S. Mishra: Vorticity preserving finite volume schemes for the shallow water equations
Efficiency and Error analysis

- Runtime: Projection method takes about $2\times$ the runtime of the corresponding FV-solver used in the prediction step.
- Exactness: As seen, projection method leads to much higher accuracy. For the problem of Numerical Experiment 3, compare error vs. runtime over a sequence of meshes for the Rusanov, Roe and the projection method used with each of them (VPRus, VPRoe):
Pros
• very general method (can be extended to MHD equations, incompressible Navier-Stokes equations)
• not too hard to extend ideas to irregular meshes
• existing code can be used to apply it
• good and fast solvers for the curlcurl-Operator are at hand

Cons
• method can be expensive as curlcurl problem needs to be solved in each step
• need to work with pseudovorticity instead of vorticity
• as a consequence: need to deal with pseudovorticity transport equation which is complicated and raises some problematic issues.
The projection method was first developed in 1968 in solving the incompressible Navier Stokes equations. It can be extended also to MHD and, in particular, to a pseudovorticity-preserving scheme for the Euler equations.