Eidgenössische Technische Hochschule Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Vorticity preserving finite volume schemes for the shallow water equations

U.S. Fjordholm and S. Mishra

Research Report No. 2009-35
November 2009

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

VORTICITY PRESERVING FINITE VOLUME SCHEMES FOR THE SHALLOW WATER EQUATIONS.

U. S. FJORDHOLM AND S. MISHRA

Abstract. We consider the shallow water equations and propose a numerical algorithm that approximates the transport of vorticity accurately. The algorithm is based on a predictor-corrector type projection method. *Any* consistent finite volume scheme predicts the approximate solution. An elliptic equation is solved and the momentum field is corrected to obtain the correct evolution of vorticity. We describe this projection algorithm for the wave equation and the shallow water equations. The crucial role played by the *pseudo-vorticity* transport is highlighted. Numerical experiments demonstrating a considerable gain in computational efficiency with the vorticity projection algorithm are presented.

1. Introduction

In many interesting flows like those in rivers, in the near shore ocean and irrigation channels, the vertical scale (depth) of the flow is much smaller than the horizontal scales. This scale separation can be employed [23] to simplify the incompressible Euler equations and obtain the so-called *shallow water equations*:

$$\begin{aligned} h_t + (hu)_x + (hv)_y &= 0, \\ (hu)_t + \left(hu^2 + \frac{1}{2} g h^2 \right)_x + (huv)_y &= 0, \\ (hv)_t + (huv)_x + \left(hv^2 + \frac{1}{2} g h^2 \right)_y &= 0, \end{aligned} \tag{1.1}$$

where h is the height of the fluid column and (u, v) is the velocity field. The constant g is the acceleration due to gravity.

The shallow water equations are an example of a system of conservation laws:

$$U_t + f(U)_x + g(U)_y = 0, \tag{1.2}$$

where $U = [h, hu, hv]^\top$ is the vector of unknowns and $f = [hu, hu^2 + \frac{1}{2} g h^2, huv]^\top$ and $g = [hv, huv, hv^2 + \frac{1}{2} g h^2]^\top$ are the flux vectors. A straightforward calculation yields that the eigenvalues of the Jacobians $f'(U)$ and $g'(U)$, respectively, are

$$\begin{aligned} \lambda_1 &= u - \sqrt{gh}, & \lambda_2 &= u, & \lambda_3 &= u + \sqrt{gh}, \\ \mu_1 &= v - \sqrt{gh}, & \mu_2 &= v, & \mu_3 &= v + \sqrt{gh}. \end{aligned}$$

Hence, the shallow water system is a strictly hyperbolic system. Eigenvectors can be similarly calculated [15].

Hyperbolic conservation laws (1.2) are characterized by the formation of finite time discontinuities, even for smooth initial data. These discontinuities or *shock waves* are a very important object of study. The presence of shock waves implies that solutions of (1.2) are sought in the weak sense [6]. It is also well known that weak solutions are not necessarily unique. Additional admissibility criteria or *entropy conditions* need to be imposed in order to recover uniqueness. For the shallow water system (1.1), the total energy,

$$E = \frac{1}{2} \left(hu^2 + hv^2 + gh^2 \right)$$

Date: October 14, 2009.

1991 *Mathematics Subject Classification.* 65M06,35L65.

serves as the entropy function, with corresponding entropy flux functions

$$H(U) = \frac{1}{2}\left(hu^3 + huv^2\right) + gh^2u \qquad \text{and} \qquad K(U) = \frac{1}{2}\left(hu^2v + hv^3\right) + gh^2v.$$

Thus, the entropy condition imposed on weak solutions (in the sense of distributions) is

$$E(U)_t + H(U)_x + K(U)_y \leq 0. \tag{1.3}$$

The energy inequality leads to bounds on the solutions of the shallow water system. In addition to the total energy, another object of interest is the *vorticity*,

$$\omega = v_x - u_y. \tag{1.4}$$

This quantity measures the rotation of the flow and is very important in practical applications. A detailed account of the role played by vorticity in meteorological models is available in [1, 2] and other references therein. By a simple calculation, one finds that smooth solutions of the shallow water equations satisfy the *vorticity transport* equation:

$$\omega_t + (u\omega)_x + (v\omega)_y = 0. \tag{1.5}$$

Thus, vorticity is advected with the flow. In particular, the total circulation

$$C^*(t) = \int_{\mathbb{R}^2} \omega(x, y, t)dxdy$$

is conserved in time. However, this conservation is only valid for smooth solutions, and for discontinuous solutions, vorticity may be generated over shocks [12].

A related object of interest is the *potential vorticity*,

$$\zeta = \frac{\omega^2}{h}.$$

Smooth solutions of (1.1) also satisfy the potential vorticity transport,

$$\zeta_t + (u\zeta)_x + (v\zeta)_y = 0. \tag{1.6}$$

Integrating (1.6) in time and utilizing the positivity of height, we obtain bounds on the vorticity in L^2.

A similar situation is encountered with the Euler equations of gas dynamics [15]. The physical entropy satisfies an inequality of the form (1.3) and the vorticity of the flow is transported as in (1.5). Preserving quantities of interest like the energy (entropy) and vorticity in numerical approximations of hyperbolic conservation laws is a computational major challenge[18].

In the absence of explicit formulas for the solution of (1.1), numerical methods are the main tools for studying the shallow water equations. Among the most popular methods are the so-called finite volume methods [15]. For simplicity, we consider a uniform Cartesian mesh $\{(x_i, y_j)\}$ in $\mathbb{R}^2$ with mesh sizes Δx and Δy. The domain is partitioned into rectangular cells $I_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$. A standard cell-centered finite volume method consists of updating the cell averages

$$U_{i,j}(t) = \frac{1}{\Delta x \Delta y}\int_{I_{i,j}} U(x, y, t)dxdy,$$

at each time level. For simplicity, we drop the time dependence of every quantity and write a standard finite volume scheme for (1.2) in the semi-discrete form as

$$\frac{d}{dt}U_{i,j} = \mathcal{L}(U_{i,j}) = -\frac{1}{\Delta x}\left(F_{i+1/2,j} - F_{i-1/2,j}\right) - \frac{1}{\Delta y}\left(G_{i,j+1/2} - G_{i,j-1/2}\right), \tag{1.7}$$

where F and G are numerical fluxes at the cell-edges that are consistent with the fluxes f and g, respectively. The numerical fluxes are computed by using either exact or approximate solutions of Riemann problems across the cell-edges in the normal directions and take the form,

$$F_{i+1/2,j} = F(U_{i,j}, U_{i+1,j}), \quad G_{i,j+1/2} = G(U_{i,j}, U_{i,j+1}), \tag{1.8}$$

where F, G are determined in terms of either exact or approximate Riemann solvers.

The scheme (1.7), based on two point numerical fluxes (1.8) are first-order accurate and can be extended to higher order accuracy by employing numerical fluxes based on wider, $2p$-point stencils, $I_{i+1/2} := \{i' \,\big|\, |i' - i - 1/2| < p\}$ along the x-axis and $J_{j+1/2} := \{j' \,\big|\, |j' - j - 1/2| < p\}$ along the y-axis,

$$F_{i+1/2,j} = F\Big(\{U_{i',j}\}_{i' \in I_{i+1/2}}\Big), \quad G_{i,j+1/2} = G\Big(\{U_{i,j'}\}_{j' \in J_{j+1/2}}\Big). \tag{1.9}$$

The building blocks for such extensions are still the 2-point numerical fluxes, $F(\cdot,\cdot)$ and $G(\cdot,\cdot)$. As a prototype example, we recall the class of second-order schemes based on piecewise bilinear MUSCL reconstruction [14],

$$\mathbf{p}_{i,j}(x,y) := U_{i,j} + \frac{U'_{i,j}}{\Delta x}(x - x_i) + \frac{U^{\backprime}_{i,j}}{\Delta y}(y - y_j); \tag{1.10a}$$

Here, U' and $U^{\backprime}$ denote the *numerical derivatives*.

$$\begin{aligned} U'_{i,j} &= \operatorname{minmod}(U_{i+1,j} - U_{i,j}, \frac{1}{2}(U_{i+1,j} - U_{i-1,j}), U_{i,j} - U_{i-1,j}), \\ U^{\backprime}_{i,j} &= \operatorname{minmod}(U_{i,j+1} - U_{i,j}, \frac{1}{2}(U_{i,j+1} - U_{i,j-1}), U_{i,j} - U_{i,j-1}), \end{aligned} \tag{1.10b}$$

which utilize the minmod limiter

$$\operatorname{minmod}(a,b,c) = \begin{cases} sgn(a)\min\{|a|,|b|,|c|\}, & \text{if} \quad sgn(a) = sgn(b) = sgn(c), \\ 0, \qquad \text{otherwise.} \end{cases} \tag{1.10c}$$

In this manner, one can reconstruct in each cell $I_{i,j}$, the point values

$$U^E_{i,j} := \mathbf{p}_{i,j}(x_{i+\frac{1}{2}}, y_j), \quad U^W_{i,j} := \mathbf{p}_{i,j}(x_{i-\frac{1}{2}}, y_j), \quad U^N_{i,j} := \mathbf{p}_{i,j}(x_i, y_{j+\frac{1}{2}}), \quad U^S_{i,j} := \mathbf{p}_{i,j}(x_i, y_{j-\frac{1}{2}}), \tag{1.11a}$$

from the given neighboring cell averages $U_{i,j}, U_{i\pm1,j}$ and $U_{i,j}, U_{i,j\pm1}$. The resulting second-order fluxes are then given by

$$F_{i+\frac{1}{2},j} = F(U^E_{i,j}, U^W_{i+1,j}), \quad G_{i,j+\frac{1}{2}} = G(U^N_{i,j}, U^S_{i,j+1}). \tag{1.11b}$$

The use of minmod limiter ensures the non-oscillatory behavior of the second-order schemes (1.7),(1.10). Similar reconstructions together with upwind or central averaging yield a large class of high-resolution finite-volume semi-discrete schemes, e.g., [10, 20, 13],

For time integration, we use a standard forward Euler method for first order schemes, and for second order schemes the strong stability preserving (SSP) Runge-Kutta method of [9]. Given a solution $U^n_{i,j}$ at time step t_n, the solution $U^{n+1}_{i,j}$ is computed by

$$\begin{aligned} U^*_{i,j} &= U^n_{i,j} + \Delta t^n \mathcal{L}(U^n_{i,j}) \\ U^{**}_{i,j} &= U^*_{i,j} + \Delta t^n \mathcal{L}(U^*_{i,j}) \\ U^{n+1}_{i,j} &= \frac{1}{2}(U^n_{i,j} + U^{**}_{i,j}), \end{aligned} \tag{1.12}$$

where $\mathcal{L}$ is defined in (1.7). The time step Δt^n is determined by a standard CFL condition. In all simulations we use a CFL number of 0.45.

A wide variety of numerical fluxes (1.8) are available for the shallow water equations [15] and finite volume schemes of the form (1.7) are quite successful in approximating (1.1). However, standard numerical methods may not *faithfully* discretize objects of interest like energy and vorticity. In particular, the numerical approximations may not respect the energy inequality (1.3) or the vorticity transport (1.5). Energy errors are highly problematic for long time integration of the shallow water system [2]. Recent papers, [7, 8], present energy preserving and energy stable schemes for the shallow water system and demonstrate a major gain in accuracy as well as stability when the energy preserving numerical schemes are used.

Standard finite volume schemes may produce large errors in the vorticity transport [18, 12], which may result in incorrect qualitative behavior of, for instance, meteorological flows [2]. Similar issues arise in the simulation of the Euler equations of gas dynamics. It is well known [11] that standard finite volume schemes produce large vorticity errors, which results in incorrect approximations for strongly vortical flows.

Our goal is to design *vorticity preserving schemes* – numerical schemes that transport vorticity accurately – for the shallow water equations (1.1). Recent papers like [12, 18] have proposed a simple model problem, the *wave equation*

$$\begin{bmatrix} \hbar \\ m_1 \\ m_2 \end{bmatrix}_t + \begin{bmatrix} 0 & c & 0 \\ c & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hbar \\ m_1 \\ m_2 \end{bmatrix}_x + \begin{bmatrix} 0 & 0 & c \\ 0 & 0 & 0 \\ c & 0 & 0 \end{bmatrix} \begin{bmatrix} \hbar \\ m_1 \\ m_2 \end{bmatrix}_y = 0, \tag{1.13}$$

to act as a testbed for the design of such schemes. The wave equation may be viewed as a linearization of (1.1) around a stationary background state $h = h_0$, $u = v = 0$. Letting $c = \sqrt{gh_0}$ and $\hbar = ch$ and ignoring terms of order two or more in (1.1), we get precisely (1.13). The eigenvalues of the flux matrices in (1.13) are $-c$, 0 and c, and so the wave equation is strictly hyperbolic.

The relevant measure of vorticity for the wave equation is the curl of the momentum field,

$$\Gamma = (m_2)_x - (m_1)_y. \tag{1.14}$$

It is easily seen that this quantity is constant in time:

$$\begin{aligned} \Gamma_t &= -((c\hbar_y)_x - (c\hbar_x)_y) \\ &= 0. \end{aligned} \tag{1.15}$$

Recent papers like [12, 17, 18] have proposed finite volume schemes for (1.13) that preserve a discrete version of the vorticity. These schemes are based on rewriting the standard finite volume scheme (1.7) on staggered meshes [18], in terms of flux distributions [12], or in a genuinely multi-dimensional manner [17]. The task of extending some of these vorticity preserving schemes for the linear wave equation system (1.13) to obtain schemes that transport vorticity correctly (discrete version of (1.5)) for the shallow water equations is quite complicated. Non-linearity plays a leading role in (1.1) and it is not clear how the vorticity preserving schemes for (1.13) can be modified for this problem.

Another approach for designing schemes with correct vorticity transport was proposed recently by Ismail and Roe [11] in the context of the Euler equations of gas dynamics. Their algorithm is based on a modified projection method. The projection method relies upon the *Hodge decomposition* of any vector field into a divergence free part and a curl free part, and was first used for incompressible flow calculations by Chorin in [5]. Extensions of the projection method for incompressible flow were proposed in [3]. Another application of the projection method was in numerical methods for the magnetohydrodynamics (MHD) equations [22]. The MHD equations are a non-linear system of conservation laws, equipped with the constraint that the magnetic field remain divergence free during the evolution. Standard numerical schemes like the finite volume scheme (1.7) may not respect this constraint and may lead to oscillations and other instabilities [22]. In [4], the authors apply the projection method to clean divergence. This approach is quite popular in MHD codes.

The projection method leads to an elliptic equation at every time step. This equation can be solved by standard *fast* elliptic solvers and the resulting solution satisfies a constraint. The elliptic equation can be quite expensive to solve. The projection method of [5, 3, 4] cannot be directly used to design a scheme that transports vorticity correctly for the shallow water system (1.1). Instead as in [11], an estimate of the vorticity is provided by the *pseudo-vorticity*,

$$\Omega = (m_2)_x - (m_1)_y, \tag{1.16}$$

for momenta $(m_1, m_2) = (hu, hv)$. Note that the pseudo-vorticity is the curl of the momentum and the momenta are the conserved variables in (1.1). Smooth solutions of the shallow water system (1.1) satisfy the pseudo-vorticity transport:

$$\begin{aligned} &\Omega_t + (u\Omega)_x + (v\Omega)_y + (m_2(u_x + v_y))_x - (m_1(u_x + v_y))_y \\ &\quad + \left(\frac{u^2 + v^2}{2} h_y\right)_x - \left(\frac{u^2 + v^2}{2} h_x\right)_y = 0. \end{aligned} \tag{1.17}$$

The pseudo-vorticity transport is more complicated than the vorticity transport (1.5) as second derivative terms appear explicitly in the equation. Furthermore, the pseudo-vorticity transport (1.17) accounts for *compressibility*. Note that if the flow is incompressible, i.e, the velocity field is divergence free, and the height h is constant

in space, then (1.17) reduces to the vorticity transport equation (1.5). The non-zero divergence of the velocity and height variations contribute to the error terms in (1.17). The approach of [11] relied on controlling the pseudo-vorticity for the Euler equations. However, the corresponding pseudo-vorticity transport equation of [11] (eqn. 30) appears to be incorrect, particularly when the height (density for the Euler equations) varies in space. The projection method of [11] is based on a relaxation type elliptic solver.

In this paper we propose a projection method for the shallow water equations. Our *predictor-corrector method* involves a prediction step in which the solution is evolved by *any* consistent finite volume method (1.7). In particular, high-order accurate finite volume schemes (1.10) can be used in this step. The essential step involves controlling pseudo-vorticity by a suitable discretization of the transport equation (1.17). The correction step is based on a Hodge decomposition and results in an elliptic equation that needs to be solved at every time step. We utilize fast solvers to solve the elliptic equation: a direct method for Neumann boundary conditions and a suitable conjugate gradient method for periodic boundary conditions. Some stability estimates are derived for the projection method for the system wave equation. Numerical experiments are presented to demonstrate the robustness and computational efficiency of this modified projection method for both the linear wave equation (1.13) and the shallow water system (1.1).

The rest of the paper is organized as follows: We present the projection method for the system wave equation in Section 2 and for the shallow water equations in Section 3. Conclusions are drawn in Section 4.

2. Vorticity projection for the system wave equation

In this section we design a projection method for approximating the system wave equation (1.13) that preserves a discrete version of the vorticity (1.14). The wave equation system is a linearized form of the shallow water equations (1.1) and the vorticity is constant in time. We seek to modify the finite volume scheme (1.7) such that a discrete version of the vorticity remains constant in time. We do so by the following discrete projection algorithm.

Step 1: **Prediction.** Compute a candidate solution $\widetilde{U}_{i,j}^{n+1}$ for (1.13) at time t_{n+1} for all mesh points x_i, y_j with *any* consistent and conservative finite volume scheme (1.7). Let $\widetilde{\Gamma}_{i,j}^{n+1}$ be the discrete vorticity of $\widetilde{U}_{i,j}^{n+1}$, computed as

$$\widetilde{\Gamma}_{i,j}^{n+1} = D_x(\widetilde{m}_2)_{i,j}^{n+1} - D_y(\widetilde{m}_1)_{i,j}^{n+1}, \tag{2.1}$$

where D_x and D_y are the central differences

$$D_x U_{i,j} = \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta x} \qquad \text{and} \qquad D_y U_{i,j} = \frac{U_{i,j+1} - U_{i,j-1}}{2\Delta y}. \tag{2.2}$$

Step 2: **Elliptic Solve.** Find the solution $\psi = (\psi_{i,j})$ of the discrete Poisson problem:

$$\begin{cases} -\left(D_x^2 + D_y^2\right)\psi_{i,j} = \widetilde{\Gamma}_{i,j}^{n+1} - \Gamma_{i,j} & \text{in the interior,} \\ \psi_{i,j} = 0 & \text{at the boundary.} \end{cases} \tag{2.3}$$

Here, $\Gamma_{i,j} = \Gamma_{i,j}^0$ denotes the initial vorticity.

Step 3: **Projection.** Define the solution at the next time step $U_{i,j}^{n+1} = \left[\hbar_{i,j}^{n+1}, (m_1)_{i,j}^{n+1}, (m_2)_{i,j}^{n+1}\right]^\top$ as

$$\hbar_{i,j}^{n+1} = \widetilde{\hbar}_{i,j}^{n+1}, \qquad (m_1)_{i,j}^{n+1} = (\widetilde{m}_1)_{i,j}^{n+1} - D_y\psi_{i,j}, \qquad (m_2)_{i,j}^{n+1} = (\widetilde{m}_2)_{i,j}^{n+1} + D_x\psi_{i,j}. \tag{2.4}$$

Note that any finite volume scheme can be used in Step 1 of the above algorithm. Similarly, any linear system solver can be used to solve the discrete Poisson equation in Step 2. Hence, the above algorithm is very general. The key feature of the projection method is stated in the following lemma.

Lemma 2.1. *The solution $U^n = (\hbar^n, m_1^n, m_2^n)$ computed by the projection method at any time step level t^n satisfies*

$$\Gamma_{i,j}^n := D_x(m_2)_{i,j}^n - D_y(m_1)_{i,j}^n = \Gamma_{i,j},$$

with $\Gamma_{i,j} = \Gamma_{i,j}^0$ being the initial vorticity. Hence, the discrete vorticity remains constant in time.

Proof. Simply inserting the definition of m_1^n and m_2^n gives

$$\begin{aligned} D_x(m_2)_{i,j}^n - D_y(m_1)_{i,j}^n &= D_x(\widetilde{m}_2)_{i,j}^n - D_y(\widetilde{m}_1)_{i,j}^n + \left(D_x^2 + D_y^2\right)\psi_{i,j}^n \\ &= \Gamma_{i,j}. \end{aligned}$$

□

In addition to vorticity preservation, another desirable feature of the projection algorithm is its stability. The system wave equation (1.13) is equipped the energy

$$E = \frac{1}{2}\left(\hbar^2 + m_1^2 + m_2^2\right). \tag{2.5}$$

A straightforward calculation shows that the energy satisfies the identity

$$E_t + (c\hbar m_1)_x + (c\hbar m_2)_y = 0. \tag{2.6}$$

Hence, the total energy is conserved in time. Many standard finite volume schemes like the Rusanov scheme and the Roe scheme (detailed descriptions are provided later in this section) are energy stable (the total energy dissipates in time). Hence, by using such a scheme in Step 1 of the discrete projection algorithm results in a decrease in energy. We show that Steps 2 and 3 of the projection algorithm are also energy stable in the lemma below. We have dropped the time dependence of all quantities for notational convenience.

Lemma 2.2. *Let E be the energy of a solution computed by the projection method, and let $\widetilde{E} = \frac{1}{2}\left(\widetilde{\hbar}^2 + \widetilde{m}_1^2 + \widetilde{m}_2^2\right)$ be the energy of the solution obtained in the prediction step. We have*

$$\sum_{i,j} E_{i,j} \leq \sum_{i,j} \widetilde{E}_{i,j} - 2\sum_{i,j} \Gamma_{i,j}\psi_{i,j}.$$

In particular, if the initial vorticity is zero, then

$$\sum_{i,j} E_{i,j} \leq \sum_{i,j} \widetilde{E}_{i,j}.$$

Proof. We use summation by parts extensively. As $\psi = 0$ at the boundary, all boundary terms will drop out. Then

$$\begin{aligned} \sum_{i,j} E_{i,j} =& \sum_{i,j} \left(\hbar_{i,j}^2 + (\widetilde{m}_{1,i,j} - D_y\psi_{i,j})^2 + (\widetilde{m}_{2,i,j} + D_x\psi_{i,j})^2\right) \\ =& \sum_{i,j} \widetilde{E}_{i,j} - 2\sum_{i,j} \left(\widetilde{m}_{1,i,j} D_y\psi_{i,j} - \widetilde{m}_{2,i,j} D_x\psi_{i,j}\right) + \sum_{i,j} \left((D_x\psi_{i,j})^2 + (D_y\psi_{i,j})^2\right). \end{aligned}$$

The second term is

$$\begin{aligned} -2\sum_{i,j} \left(\widetilde{m}_{1,i,j} D_y\psi_{i,j} - \widetilde{m}_{2,i,j} D_x\psi_{i,j}\right) &= 2\sum_{i,j} \psi_{i,j}(D_y\widetilde{m}_{1,i,j} - D_x\widetilde{m}_{2,i,j}) \\ &= -2\sum_{i,j} \psi_{i,j}\left(\Gamma_{i,j} - (D_x^2 + D_y^2)\psi_{i,j}\right) \\ &= -2\sum_{i,j} \Gamma_{i,j}\psi_{i,j} - 2\sum_{i,j} \left((D_x\psi_{i,j})^2 + (D_y\psi_{i,j})^2\right). \end{aligned}$$

Hence,

$$\begin{aligned} \sum_{i,j} E =& \sum_{i,j} \widetilde{E}_{i,j} - 2\sum_{i,j} \Gamma_{i,j}\psi_{i,j} - \sum_{i,j} \left((D_x\psi_{i,j})^2 + (D_y\psi_{i,j})^2\right) \\ \leq& \sum_{i,j} \widetilde{E}_{i,j} - 2\sum_{i,j} \Gamma_{i,j}\psi_{i,j}. \end{aligned}$$

If $\Gamma_{i,j} \equiv 0$, then the last term drops out leading to energy dissipation. □

2.1. **Prediction Step.** In order to complete the description of the discrete projection algorithm, we need to specify the numerical fluxes for the prediction step (step 1 of the algorithm). As stated earlier, *any* consistent numerical flux can be used in the finite volume scheme (1.7). In particular, we choose the Rusanov flux, that takes the form:

$$\begin{aligned} F^{\mathrm{Rus}}_{i+1/2,j} &= \frac{1}{2}\big(f(U_{i,j}) + f(U_{i+1,j})\big) - \frac{c}{2}(U_{i+1,j} - U_{i,j}), \\ G^{\mathrm{Rus}}_{i,j+1/2} &= \frac{1}{2}\big(g(U_{i,j}) + g(U_{i,j+1})\big) - \frac{c}{2}(U_{i,j+1} - U_{i,j}), \end{aligned} \tag{2.7}$$

where f and g are fluxes for the system wave equation (1.13) in the x- and y- directions respectively. The Rusanov flux is very simple to implement and requires minimal characteristic information. However, it can be dissipative, particularly at discontinuities. A more accurate numerical flux is the Roe flux:

$$\begin{aligned} F^{\mathrm{Roe}}_{i+1/2,j} &= \frac{1}{2}\big(f(U_{i,j}) + f(U_{i+1,j})\big) - \frac{1}{2}R^x|\Lambda^x|R^{x,-1}(U_{i+1,j} - U_{i,j}), \\ G^{\mathrm{Roe}}_{i,j+1/2} &= \frac{1}{2}\big(g(U_{i,j}) + g(U_{i,j+1})\big) - \frac{1}{2}R^y|\Lambda^y|R^{y,-1}(U_{i,j+1} - U_{i,j}), \end{aligned} \tag{2.8}$$

where Λ^x, Λ^y are the diagonal matrices of eigenvalues and R^x, R^y the matrices of eigenvectors for the Jacobians of the wave equation system (1.13).

2.2. **Elliptic Solver.** The discrete Poisson equation (2.3) must be solved at every time step, so a fast and efficient solver is essential to the computational efficiency of the scheme. The subject of efficient linear solvers for the Poisson equation is treated in many text books. We describe the methods used in our computations very briefly. For simplicity, we compute on an $N \times M$ uniform Cartesian grid. If we write $C_{i,j} = \widetilde{\Gamma}^{n+1}_{i,j} - \Gamma_{i,j}$, then the matrix equation (2.3) can be rewritten as

$$\frac{1}{4\Delta x^2} T_N \Psi + \frac{1}{4\Delta y^2} \Psi T_M = C, \tag{2.9}$$

where $\Psi = (\psi_{i,j})$ and T_N is the $N \times N$ symmetric, positive definite matrix

$$T_N = \begin{bmatrix} 2 & 0 & -1 & & & & \\ 0 & 2 & 0 & -1 & & & \\ -1 & 0 & 2 & 0 & -1 & & \\ & \ddots & & \ddots & & \ddots & \\ & & -1 & 0 & 2 & 0 & -1 \\ & & & -1 & 0 & 2 & 0 \\ & & & & -1 & 0 & 2 \end{bmatrix}. \tag{2.10}$$

We solve this equation with a direct method from [16]. Let R_N be the matrix of eigenvectors and $D_N = \mathrm{diag}(\lambda^1_N, \ldots, \lambda^N_N)$ the matrix of eigenvalues of T_N. R_N may be chosen such that $R^2_N = I_N$, the identity matrix in $\mathbb{R}^{N\times N}$. Multiplying (2.9) by R_N on the left and R_M on the right, we find that

$$\frac{1}{4\Delta x^2} D_N X + \frac{1}{4\Delta y^2} X D_M = R_N C R_M,$$

where $X = R_N \Psi R_M$. The (i,j) entry of the left-hand side of this equation is

$$\frac{\lambda^i_N}{4\Delta x^2} X_{i,j} + \frac{\lambda^j_M}{4\Delta y^2} X_{i,j}.$$

Hence, $X = (R_N C R_M)./S$, where

$$S_{i,j} = \frac{\lambda^i_N}{4\Delta x^2} + \frac{\lambda^j_M}{4\Delta y^2}$$

and ./ denotes component-wise division. The solution of (2.9) is therefore

$$\Psi = R_N X R_M.$$

2.2.1. *Periodic boundary conditions.* The above discussion did not rely on the specifics of the boundary condition on U. In the special case of periodic boundary conditions:

$$U_{0,j} = U_{N,j}, \qquad U_{-1,j} = U_{N-1,j},$$

and similar conditions in the y-direction, we can improve the method by applying the same condition on ψ in (2.3). This will result in the matrix equation:

$$\frac{1}{4\Delta x^2} P_N \Psi + \frac{1}{4\Delta y^2} \Psi P_M = C, \tag{2.11}$$

with

$$P_N = \begin{bmatrix} 2 & 0 & -1 & 0 & \dots & -1 & 0 \\ 0 & 2 & 0 & -1 & \dots & 0 & -1 \\ -1 & 0 & \ddots & & & \vdots & \vdots \\ 0 & -1 & & \ddots & & -1 & 0 \\ \vdots & \vdots & & & \ddots & 0 & -1 \\ -1 & 0 & \dots & -1 & 0 & 2 & 0 \\ 0 & -1 & \dots & 0 & -1 & 0 & 2 \end{bmatrix}. \tag{2.12}$$

The direct method of the previous section is unsuitable for this case. Hence, we will employ the conjugate gradient method [16]. The conjugate gradient method searches for the solution of a matrix equation $Az = b$ along orthogonal search paths $s_k \in \mathbb{R}^{NM}$, and it finds the exact solution after at most NM iterations. As this number grows large quadratically, we will use the method as an iterative method, halting the process when $||s_k||_{\ell^2}$ is less than some $\varepsilon > 0$. Given a right-hand side b, we set $\varepsilon = \alpha||b||_{\ell^2}$ for an $\alpha > 0$, so that the allowed error in the solution is proportional to b. We chose $\alpha = 10^{-8}$ in this paper.

The matrix-matrix equation (2.11) can be rewritten as a matrix-vector equation $Az = b$, with

$$z = \mathrm{vec}(\Psi), \quad b = \mathrm{vec}(C) \quad \text{and} \quad A = \frac{1}{4\Delta x^2} P_N \otimes I_M + \frac{1}{4\Delta y^2} I_N \otimes P_M,$$

$\otimes$ denoting the Kroenecker product and $\mathrm{vec}(\Psi)$ the column-first vectorized version of Ψ [16]. Contrary to T_N, the matrix P_N is only positive semidefinite; its kernel is spanned by the vectors

$$r_N^1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \quad r_N^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix} \in \mathbb{R}^N$$

if N is even, and by

$$r_N^1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix} \in \mathbb{R}^N$$

if N is odd. Hence, the matrix A will be positive semidefinite with a kernel spanned by the vector(s)

$$r_N^k \otimes r_M^l.$$

As a consequence, the equation $Az = b$ does not have a solution whenever b has a nonzero component in $\ker A$. However, the equation $Az = \widehat{b}$, where

$$\widehat{b} = \mathrm{proj}_{(\ker A)^\perp}(b) \ \ (= \mathrm{proj}_{\mathrm{Im} A}(b)),$$

does have a unique solution in $(\ker A)^\perp$. The conjugate gradient method is well-defined and converges for this modified equation. Therefore, we propose using the solution Ψ of this modified equation. Note that the above

discussion provides a sort of preconditioner for the problem: We have modified the ill-conditioned problem $Az = b$ by multiplying on both sides by the projection matrix of $(\ker A)^\perp$, thus obtaining $Az = \widehat{b}$.

Remark 2.3. The linear solvers presented above are not necessarily optimal. We can employ more efficient *scalable* iterative solvers like variants of the preconditioned conjugate gradient and other Krylov space methods for solving the linear system (2.3) and increase the efficiency of our algorithm. However, we choose to keep the description of the projection algorithm simple and present the simplest linear solvers.

Remark 2.4. The projection method described in this section is based on a simple Cartesian mesh. In real life applications, one has to deal with unstructured meshes. It is relatively simple to extend the projection algorithm to unstructured meshes. Finite volume schemes on unstructured meshes are readily available. A suitable discrete form of vorticity can be easily defined ([18]). Solving the resulting discrete Poisson equation on an unstructured mesh is standard numerical linear algebra. All these ingredients need to combined to yield a vorticity projection algorithm on unstructured meshes. We will describe this method in detail in a forthcoming paper.

2.3. **Numerical experiments.** We test the vorticity preserving schemes presented in the last section on a couple of numerical experiments for the wave equation (1.13). The following schemes are tested:

Rus	First-order standard finite volume scheme (1.7) with Rusanov flux (2.7).
Roe	First-order standard finite volume scheme (1.7) with Roe flux (2.8).
VPRus	Vorticity preserving scheme with Rusanov flux (2.7) in the prediction step.
VPRoe	Vorticity preserving scheme with first-order Roe flux (2.8) in the prediction step.

Numerical experiment: Periodic waves. The first experiment features a periodic boundary and a nonzero initial vorticity. The initial conditions are given by

$$\hbar \equiv 0, \qquad m_1(x,y) = m_2(x,y) = \cos(\pi(x+y)) - \cos(\pi(x-y)).$$

It is readily checked that

$$\hbar(x,y,t) = \sqrt{2}\sin(\pi(x+y))\sin(\sqrt{2}\pi t),$$
$$m_1(x,y,t) = m_2(x,y,t) = \cos(\pi(x+y))\cos(\sqrt{2}\pi t) - \cos(\pi(x-y))$$

is the solution of this initial value problem. The corresponding vorticity is

$$\Gamma(x,y,t) = (m_2)_x - (m_1)_y = 2\pi\sin(\pi(x-y)).$$

As expected, the expression for Γ is constant in time.

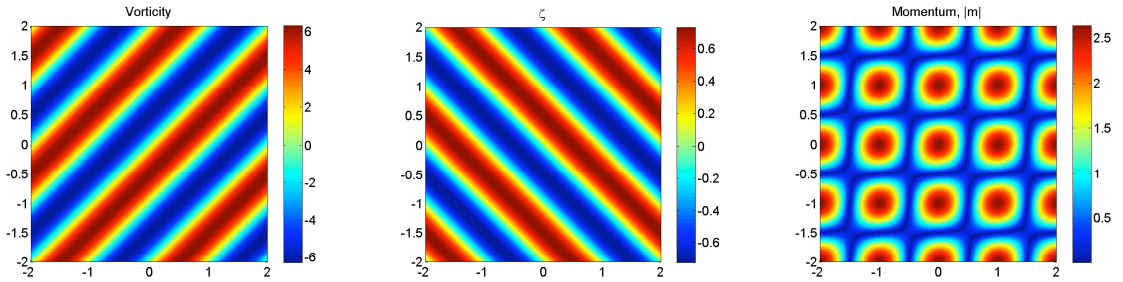
We compute for $(x,y) \in [-2,2]\times[-2,2]$ up to time $t = 2$. The solution at the final time step computed with the Rusanov, VPRus, Roe and VPRoe schemes are plotted in Figure 1, along with the exact solution. While $\hbar$ is left untouched by the VP schemes, the vorticity field is resolved much more sharply than in the predictor schemes. This is verified in Table 1 and 2, where we show relative errors

$$\frac{\|\Gamma - \Gamma_{\text{exact}}\|_{L^2}}{\|\Gamma_{\text{exact}}\|_{L^2}}$$

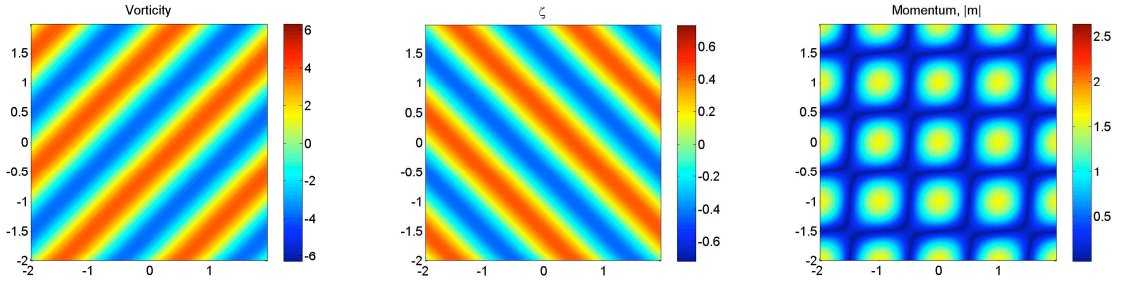
for Γ and $\hbar$ on a sequence of meshes. The VP schemes preserve vorticity up to machine precision, while the Rus and Roe schemes have errors in vorticity of the order of discretization error.

Momentum is shown in the third column of Figure 1. Clearly, the projection methods have a positive effect on the accuracy, preventing too much diffusion in m. Indeed, as shown in Table 3, the error in momentum is about $20-40$ per cent lower than in the prediction solvers.

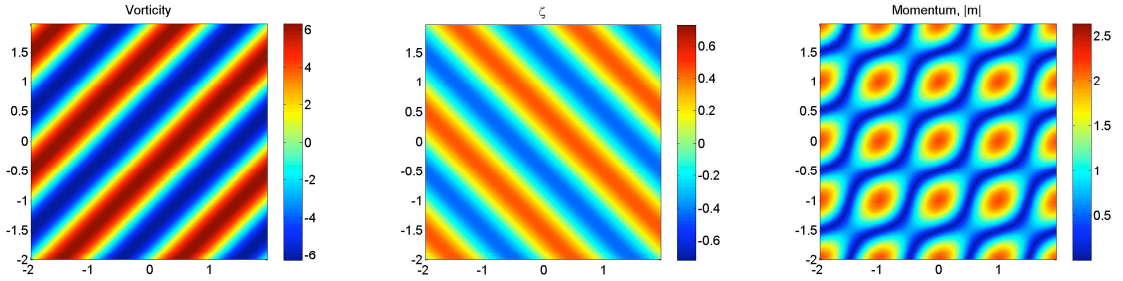
The conjugate gradient method used in the elliptic solver converges rapidly, with only 5 to 10 iterations needed to get below the error threshold. Thus, the overhead is low, and the VP schemes take only about 1.5 to three times more time to run than the standard schemes. The run time for the VP schemes can be improved by employing a more efficient iterative solver for (2.3).



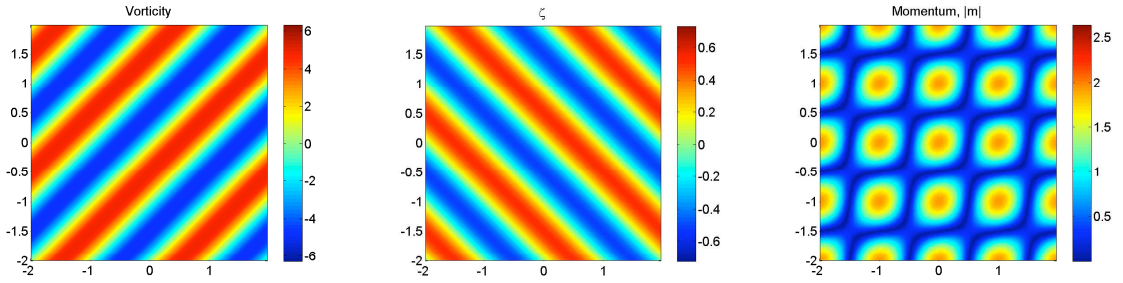
(a) Exact solution



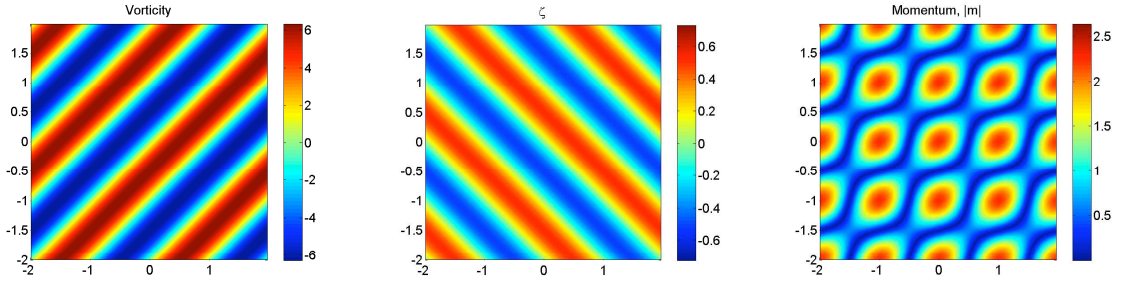
(b) Rus



(c) VPRus



(d) Roe



(e) VPRoe

FIGURE 1. Solutions at $t = 2$ computed by the four schemes on a mesh of 160×160 grid points.

	Rusanov	VPRus	Roe	VPRoe
40	8.65×10^{-1}	9.07×10^{-12}	6.28×10^{-1}	6.02×10^{-11}
80	6.29×10^{-1}	6.27×10^{-11}	3.90×10^{-1}	4.40×10^{-11}
160	3.90×10^{-1}	1.70×10^{-11}	2.19×10^{-1}	1.04×10^{-11}
320	2.19×10^{-1}	3.28×10^{-12}	1.16×10^{-1}	1.73×10^{-12}

TABLE 1. Relative error in Γ.

	Rusanov	VPRus	Roe	VPRoe
40	7.39×10^{-1}	7.39×10^{-1}	4.66×10^{-1}	4.66×10^{-1}
80	4.51×10^{-1}	4.51×10^{-1}	2.62×10^{-1}	2.62×10^{-1}
160	2.48×10^{-1}	2.48×10^{-1}	1.39×10^{-1}	1.39×10^{-1}
320	1.30×10^{-1}	1.30×10^{-1}	7.18×10^{-2}	7.18×10^{-2}

TABLE 2. Relative error in $\hbar$.

	Rusanov	VPRus	Roe	VPRoe
40	7.77×10^{-1}	4.16×10^{-1}	5.64×10^{-1}	3.02×10^{-1}
80	5.46×10^{-1}	2.66×10^{-1}	3.43×10^{-1}	1.74×10^{-1}
160	3.33×10^{-1}	1.52×10^{-1}	1.91×10^{-1}	9.41×10^{-2}
320	1.85×10^{-1}	8.20×10^{-2}	1.01×10^{-1}	4.89×10^{-2}

TABLE 3. Relative error in m, the momentum.

Numerical experiment: Expanding wave. This experiment features a smooth solution with an open (Neumann) type boundary condition. The initial data is given by

$$\hbar(x, y) = c \exp\left(-15(x^2 + y^2)\right), \qquad m_1 = m_2 = 0. \tag{2.13}$$

As the initial vorticity is zero, it should stay zero at all later times.

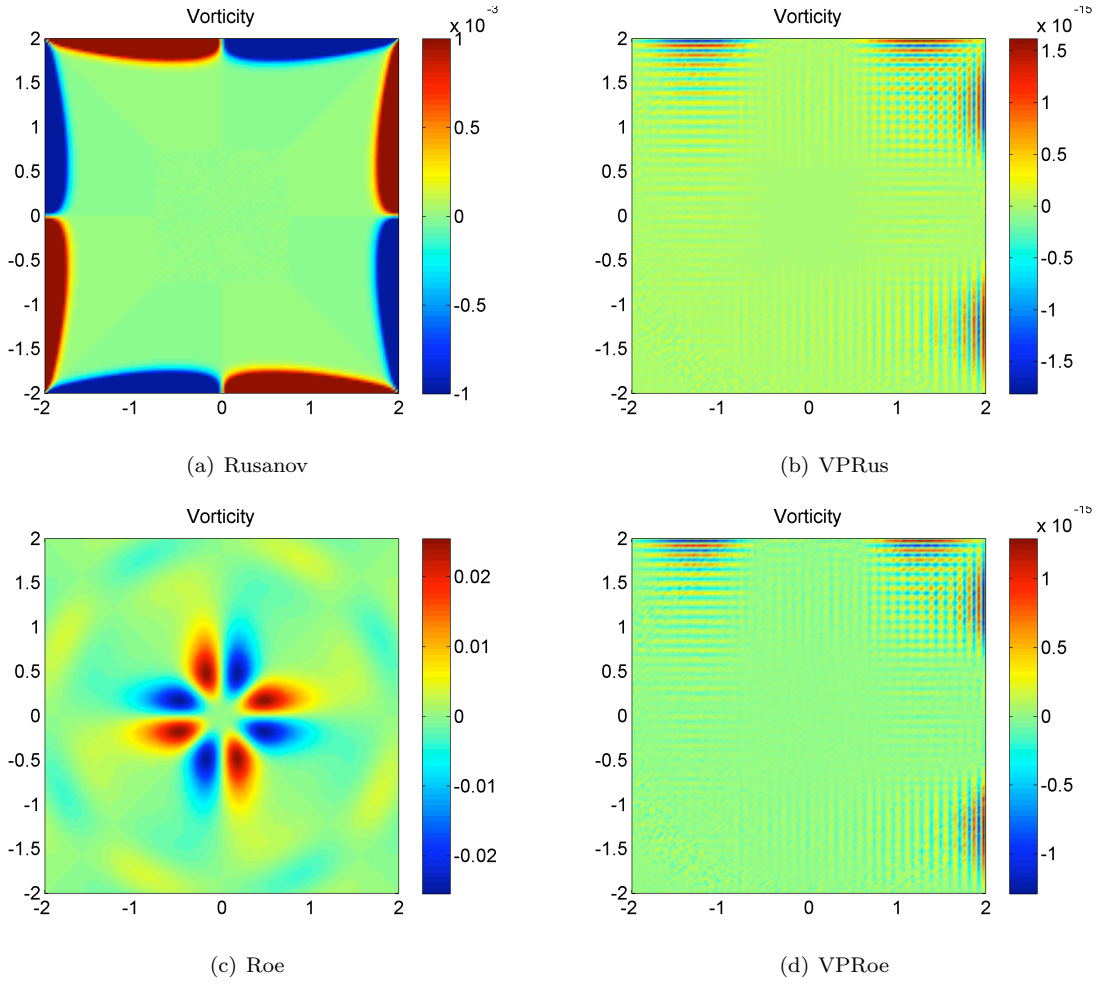
We solve for $(x, y) \in [-2, 2] \times [-2, 2]$ up to time $t = 2$. The spatial domain was discretized with $N = M = 50, 100, 150$ and 200 grid points in each direction. Vorticity at the final time step is shown in Figure 2. The Rus scheme preserves the initial vorticity exactly in the middle of the domain, but there is a large amount of vorticity production near the boundaries. This vorticity propagates into the domain at a speed of one grid cell per time step. As the ratio $\frac{\Delta t}{\Delta x}$ is kept constant, this speed is invariant with respect to grid size. This is clear in Table 4, where we show vorticity errors for the four schemes. The error for the Rusanov scheme is about $3 \cdot 10^{-2}$, irrespective of grid size. The VPRus scheme clears out these errors, and only noise of the order of machine precision is left (note the scaling of the figures).

The vorticity of the Roe scheme is shown in Figure 2(c). The figure shows that the initial constant vorticity is not preserved. Again, the projection method clears out these errors completely.

The projection method gives no gain in accuracy for the conserved variables in this experiment. As the run time of the method lies between 2 and 4 times that of the predictor solver, there is little incentive for using the vorticity projection when the main interest is in an accurate solution of conserved variables. However, the success of the method in clearing out vorticity errors motivates its use in computing vortex dominated flows.

	Rusanov	VPRus	Roe	VPRoe
50	$2.43 \cdot 10^{-2}$	$3.64 \cdot 10^{-16}$	$8.10 \cdot 10^{-2}$	$2.61 \cdot 10^{-16}$
100	$2.83 \cdot 10^{-2}$	$7.96 \cdot 10^{-16}$	$5.83 \cdot 10^{-2}$	$6.71 \cdot 10^{-16}$
150	$2.88 \cdot 10^{-2}$	$1.54 \cdot 10^{-15}$	$4.68 \cdot 10^{-2}$	$1.18 \cdot 10^{-15}$
200	$2.84 \cdot 10^{-2}$	$1.75 \cdot 10^{-15}$	$3.94 \cdot 10^{-2}$	$1.54 \cdot 10^{-15}$

TABLE 4. $\|\Gamma\|_{L^1}$ in the expanding wave problem.



(a) Rusanov (b) VPRus

(c) Roe (d) VPRoe

FIGURE 2. Vorticity at $t = 2$ for the four schemes, computed on a mesh of 150×150 grid points. Note the scaling of each figure.

3. Vorticity Projection for the Shallow water system.

The vorticity projection algorithm for the shallow water equations (1.1) is a modification of the vorticity projection algorithm for the system of wave equation. The prediction step is exactly the same: The solution at the next time step is estimated using any finite volume method. We use the Rusanov and Roe schemes, although any consistent numerical flux is applicable. Note that the scaling constant c in (2.7) should be replaced with estimates on the local wave speeds [15]. Furthermore, we can use high resolution schemes (1.10).

As stated earlier, we use pseudo-vorticity as a measure of vorticity. Unlike the wave equation, where pseudo-vorticity is constant in time, the pseudo-vorticity of the shallow water system exhibits the more complicated evolution equation (1.17). Hence, we must approximate the exact pseudo-vorticity Ω^{n+1} at the next time step as dictated by this equation. For this purpose we use the second-order central Nessyahu-Tadmor (NT) scheme [19]. As this is a central scheme, we avoid having to deal with the complicated wave structure of the evolution equation (1.17).

For completeness we include a description of the NT scheme. Given data $\Omega^n_{i,j}$ at time t_n, we solve for the staggered value $\Omega^{n+1}_{i+1/2,j+1/2}$ at time t_{n+1},

$$\Omega^{n+1}_{i+1/2,j+1/2} \approx \frac{1}{\Delta x \Delta y}\int_{J_{i,j}} \Omega(x,y,t_{n+1})\; dxdy \qquad \text{for } J_{i,j} = [x_i, x_{i+1}] \times [y_j, y_{j+1}].$$

To go from a vertex-centered solution $\Omega^{n+1}_{i+1/2,j+1/2}$ to a cell-centered solution $\Omega^{n+1}_{i,j}$, we perform a piecewise linear reconstruction of $\Omega^{n+1}_{i+1/2,j+1/2}$,

$$\Omega^{n+1}(x,y) = \Omega^{n+1}_{i+1/2,j+1/2} + \sigma_{i,j}(x - x_{i+1/2}) + \gamma_{i,j}(y - y_{j+1/2}) \qquad \text{for } (x,y) \in J_{i,j}.$$

This function is then averaged over $I_{i,j}$ to obtain the final solution

$$\Omega^{n+1}_{i,j} = \frac{1}{\Delta x \Delta y}\int_{I_{i,j}} \Omega^{n+1}(x,y)\; dxdy.$$

To solve for $\Omega^{n+1}_{i+1/2,j+1/2}$, we first write the vorticity flux functions as $f(\Omega, U) = u\Omega + dm_2 + sh_y$ and $g(\Omega, U) = v\Omega - dm_1 - sh_x$, where $d = u_x + v_y$ and $s = \frac{u^2+v^2}{2}$ (compare with (1.17)). We then have

$$\begin{aligned}
\Omega^{n+1}_{i+1/2,j+1/2} &= \frac{1}{\Delta x \Delta y}\int_{J_{i,j}} \Omega^n(x,y)\; dxdy - \frac{1}{\Delta x \Delta y}\int_{t_n}^{t_{n+1}}\int_{J_{i,j}} f(\Omega,U)_x + g(\Omega,U)_y\; dxdydt \\
&\approx \frac{1}{4}\left(\Omega^n_{i,j} + \Omega^n_{i+1,j} + \Omega^n_{i,j+1} + \Omega^n_{i+1,j+1}\right) \\
&\quad - \frac{\Delta t}{\Delta x \Delta y}\int_{J_{i,j}} f\left(\Omega^{n+\frac{1}{2}}, U^{n+\frac{1}{2}}\right)_x + g\left(\Omega^{n+\frac{1}{2}}, U^{n+\frac{1}{2}}\right)_y\; dxdy,
\end{aligned}$$

where we have used a quadrature rule for the time integral. $\Omega^{n+\frac{1}{2}}$ and $U^{n+\frac{1}{2}}$ are approximations of Ω and U at time $t_{n+\frac{1}{2}}$. We select $U^{n+\frac{1}{2}} = \frac{1}{2}(U^n + \widetilde{U}^{n+1})$ and $\Omega^{n+\frac{1}{2}} = \Omega^n - \frac{\Delta t}{2}(f^n_x + g^n_y)$, where f^n_x and g^n_y are the gradients of the flux at time t_n.

All gradients ($\sigma_{i,j}, \gamma_{i,j}, f^n_x$ and g^n_y) used in this method are obtained using the MC limiter

$$\Omega_{i,j} = \text{mm}\left(2\frac{\Omega_{i,j} - \Omega_{i-1,j}}{\Delta x},\; 2\frac{\Omega_{i+1,j} - \Omega_{i,j}}{\Delta x},\; \frac{\Omega_{i+1,j} - \Omega_{i-1,j}}{2\Delta x}\right),$$

where mm is the minmod function

$$\text{mm}(a,b,c) = \begin{cases} \max\{|a|,|b|,|c|\} & \text{if } \text{sign}(a) = \text{sign}(b) = \text{sign}(c), \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

We are now in a position to state the vorticity projection algorithm for the shallow water system.

Step 1: **Prediction.** Compute a candidate solution $\widetilde{U}^{n+1}_{i,j}$ of (1.1) at time t_{n+1} with *any* consistent finite volume scheme(1.7). Let $\widetilde{\Omega}^{n+1}_{i,j}$ be the discrete pseudo-vorticity computed by

$$\widetilde{\Omega}^{n+1}_{i,j} = D_x(\widetilde{m}_2)^{n+1}_{i,j} - D_y(\widetilde{m}_1)^{n+1}_{i,j}.$$

Step 2: **Vorticity Estimate.** Use the Nessyahu-Tadmor scheme to update the pseudo-vorticity and obtain an estimate of the pseudo-vorticity $\Omega^{n+1}_{i,j}$.

Step 3: **Elliptic Solve**. Find the solution $\psi = (\psi_{i,j})$ of the discrete Poisson problem

$$\begin{cases} -\left(D_x^2 + D_y^2\right)\psi_{i,j} = \widetilde{\Omega}^{n+1}_{i,j} - \Omega^{n+1}_{i,j} & \text{in the interior,} \\ \psi_{i,j} = 0 & \text{at the boundary,} \end{cases} \tag{3.2}$$

where D_x and D_y are the discrete derivatives (2.2) and Ω^{n+1} is the pseudo-vorticity estimate obtained in step 2.

Step 4: **Projection.** Define the solution at the next time step $U^{n+1}_{i,j} = \left[h^{n+1}_{i,j}, (m_1)^{n+1}_{i,j}, (m_2)^{n+1}_{i,j}\right]^\top$ as

$$h^{n+1}_{i,j} = \widetilde{h}^{n+1}_{i,j}, \qquad (m_1)^{n+1}_{i,j} = (\widetilde{m}_1)^{n+1}_{i,j} - D_y\psi_{i,j}, \qquad (m_2)^{n+1}_{i,j} = (\widetilde{m}_2)^{n+1}_{i,j} + D_x\psi_{i,j}. \tag{3.3}$$

The vorticity projection algorithm ensures that the pseudo-vorticity at any time level is equal to the estimated pseudo-vorticity Ω^{n+1}. The proof of this fact is identical to the proof of Lemma 2.1.

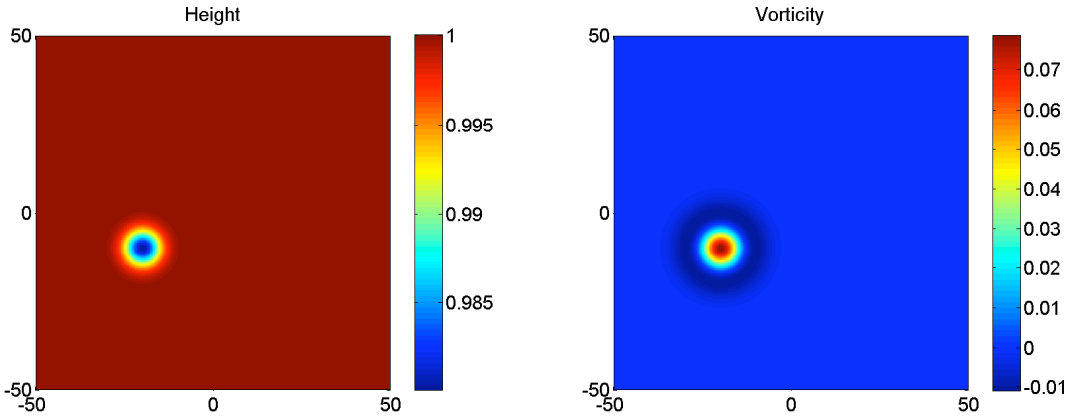
3.1. **Numerical experiment: Vorticity advection.** We test the projection method on a problem where the exact solution is known. It can be readily checked that

$$\begin{aligned} h(x,y,t) &= 1 - \frac{c_1^2}{4c_2 g} e^{2f} \\ u(x,y,t) &= M\cos(\alpha) + c_1(y - y_0 - Mt\sin(\alpha))e^f \\ v(x,y,t) &= M\sin(\alpha) - c_1(x - x_0 - Mt\cos(\alpha))e^f \end{aligned} \tag{3.4}$$

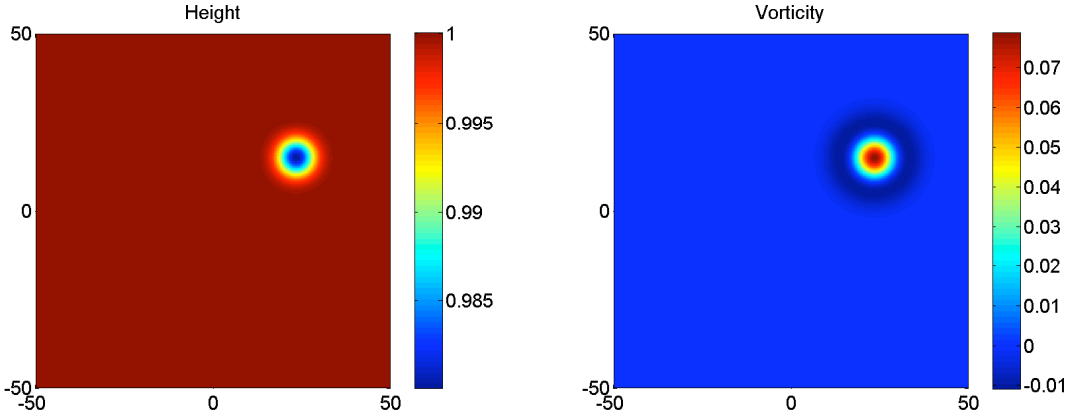
where

$$f = f(x,y,t) = -c_2\left((x - x_0 - Mt\cos(\alpha))^2 + (y - y_0 - Mt\sin(\alpha))^2\right),$$

gives a smooth solution $U = [h, hu, hv]^\top$ of the shallow water equations (1.1) for any choice of constants M, g, c_1, c_2, α, x_0 and y_0. The solution consists of a vortex traveling at a constant velocity M in a direction specified by the angle α. We let $M = 1/2$, $g = 1$, $c_1 = 0.04$, $c_2 = 0.02$ and $(x_0, y_0) = (-20, -10)$. To test the schemes'



(a) $t = 0$



(b) $t = 100$

FIGURE 3. Exact solution of the vorticity advection problem at $t = 0$ and $t = 100$.

ability of resolving flows that are not aligned with the computational grid, we let $\alpha = \frac{\pi}{6}$. We compute for $(x, y) \in [-50, 50] \times [-50, 50]$ up to time $t = 100$. The exact solution is shown in Figure 3.

The computed solutions with the Rus, Roe, VPRus and VPRoe schemes on a uniform 200×200 mesh are plotted in Figure 5. The Rus scheme dissipates the solution by a large amount, and the vortex is barely visible in the plot. The VPRus scheme, on the other hand, preserves both the magnitude and the symmetry of the vorticity quite well, and the solution resembles the exact solution closely. The Roe scheme solves the vortex advection problem poorly; the symmetry is destroyed. The projection method corrects this to some extent.

The numerical results show that in this vortex dominated problems, standard schemes are extremely inaccurate and fail to approximate the solution, even qualitatively. This experiment is representative of more complex vortex dominated flows. However, using the vorticity projection algorithm improves the accuracy to a considerable extent.

The VP schemes take about twice as long to run as the prediction solver on the same grid, and as such, a comparison of error versus grid size between the schemes would be unfair. Instead, we compute error versus run time over a sequence of meshes, from 50×50 to 250×250 grid points. These are plotted in Figure 4. Clearly, the VPRus scheme gives the best error per run time ratio of the schemes considered. The VPRoe scheme also performs well, but it seems like the non-symmetry of the Roe scheme pollutes the solution of the height variable and makes the solution look asymmetric. In both cases, the gain in computational efficiency with vorticity projection schemes is close to an order of magnitude. This experiment serves to illustrate the *considerable* gain in accuracy that results by using the vortex projection method.

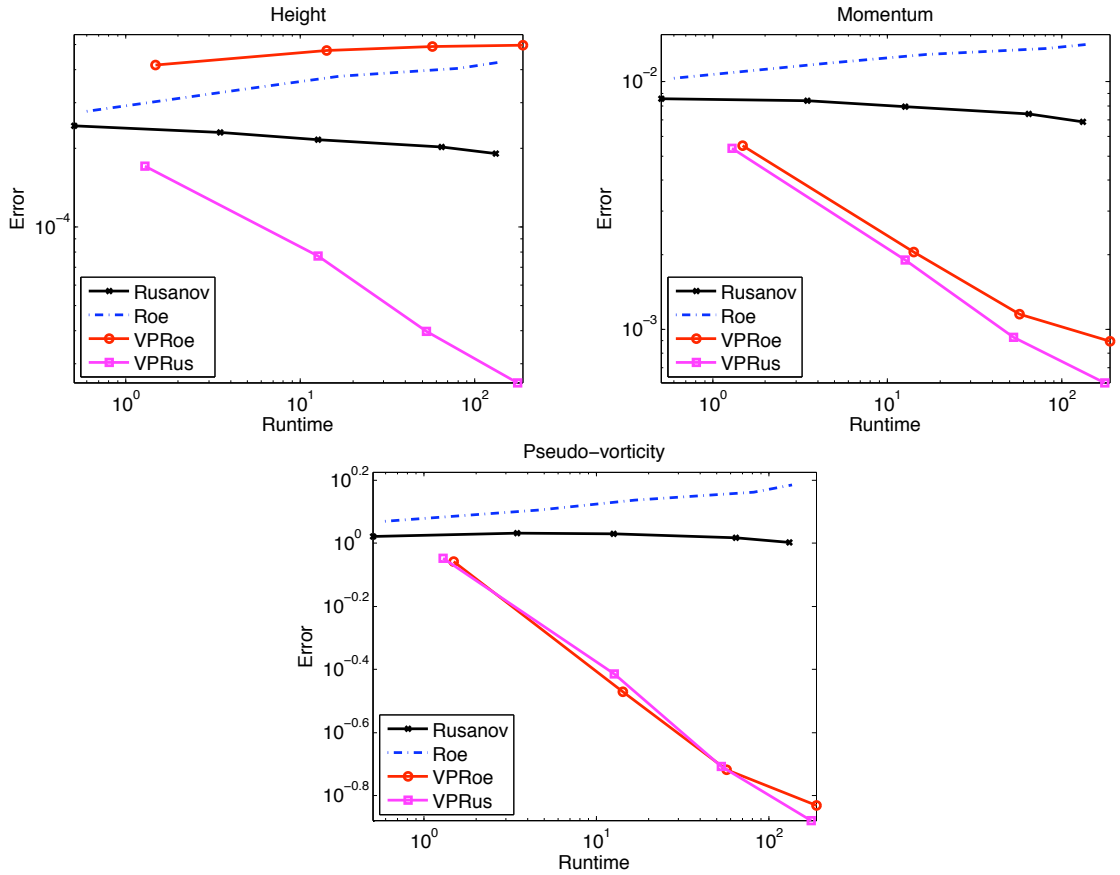
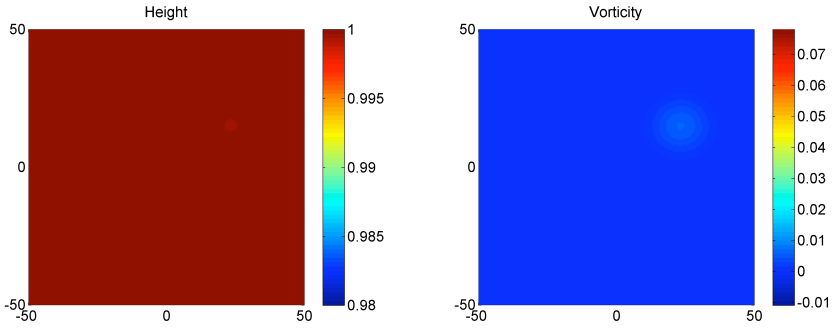
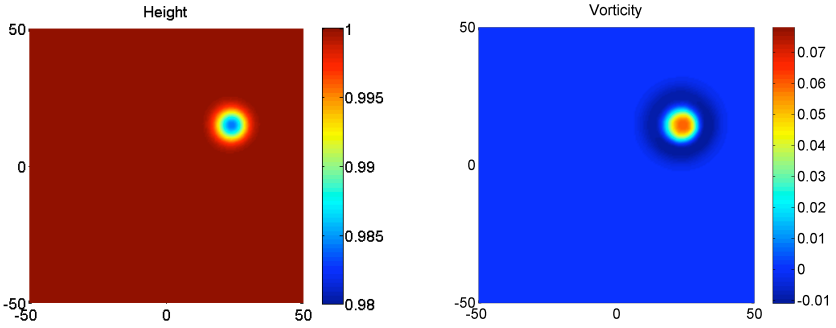


FIGURE 4. Runtime (x-axis) versus relative L^1 errors in height, momentum and vorticity (y-axis).

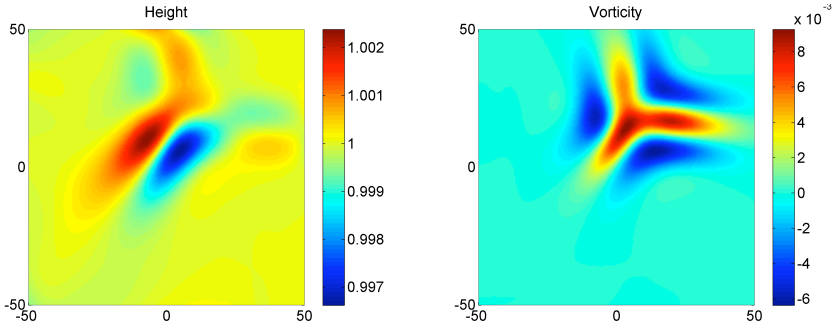
3.1.1. *Second-order schemes.* We repeat the vorticity advection experiment using second-order versions of the Rusanov (Rusanov2) and Roe (Roe2) schemes in the prediction step. The corresponding VP schemes are



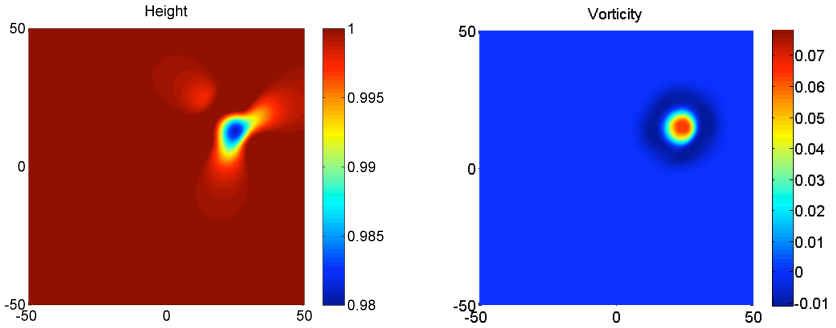
(a) Rusanov



(b) VPRus



(c) Roe



(d) VPRoe

FIGURE 5. Height and pseudo-vorticity at $t = 100$ computed by the four schemes on a mesh of 200×200 grid points.

termed VPRus2 and VPRoe2 respectively. The results are shown in Figure 6. Although the second-order Rusanov scheme performs much better than its first-order counterpart, the VPRus2 scheme is still significantly more accurate. The Roe scheme (even at second order) continues to perform extremely poorly on the vortex advection problem. In particular, the geometric structure of the vortex is destroyed by the Roe scheme. Some of the errors are corrected by the VPRoe2 scheme but it is still inferior in performance to the VPRus2 scheme. The computational efficiency is presented in Figure 7 and shows that the VPRus2 is the most efficient with accuracy-to-runtime ratios more than twice as high as in the Rusanov2 prediction solver. Interestingly, the Roe scheme shows no sign of convergence in this experiment.

3.2. **Shock-vortex interaction.** The previous numerical experiment considered a smooth solution. However, solutions of (1.1) contain discontinuities in the form of shock waves. It is interesting to study the performance of the VP schemes in the presence of shocks. Furthermore, the interaction between shocks and vortices is quite significant in many applications. Hence, we set the initial data to

$$h(x,y) = \begin{cases} \widetilde{h}(x,y) & \text{if } x < 20 \\ \widetilde{h}(x,y)(1+\xi) & \text{else} \end{cases}$$

$$u(x,y) = \begin{cases} \widetilde{u}(x,y) & \text{if } x < 20 \\ \widetilde{u}(x,y) - \sqrt{\frac{1}{2} g \widetilde{h}(x,y) \xi^2 \frac{2+\xi}{1+\xi}} & \text{else} \end{cases}$$

$$v(x,y) = \widetilde{v}(x,y),$$

where $\widetilde{h}, \widetilde{u}$ and $\widetilde{v}$ are the initial data (3.4) of the previous experiment at $t = 0$. The initial data is displayed in figure 8. The solution will consist of an advecting vortex, as in the previous example, moving over a left-going shock. The vortex interacts with the shock, distorts it and emerges out of it to continue its motion. The shock continues to move to the left. The parameter $\xi > 0$ determines the strength of the shock; we set $\xi = 10^{-2}$

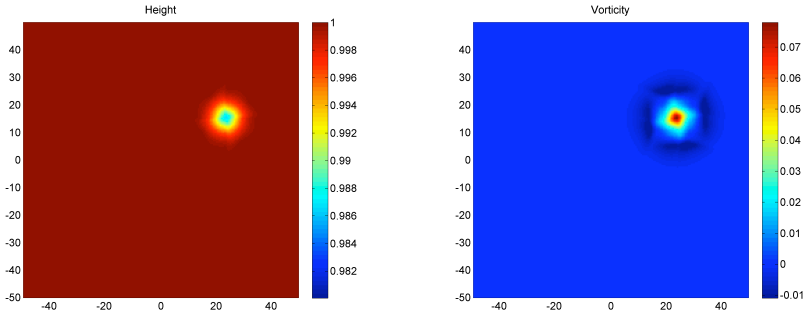
We run the Rusanov, VPRus, Rusanov and VPRus2 schemes on a 150×150 mesh and display the results in Figure 9. Clearly, the resolution of the vortex in the VPRus scheme is by far superior to the Rusanov scheme, while the shock is resolved up to the accuracy of the predictor scheme. Similarly, the VPRus2 scheme is more accurate than the Rusanov2 scheme. The vortex resolved to a greater degree of accuracy and the shock is captured as accurately as the underlying high resolution prediction scheme. This experiment clearly demonstrates that the vorticity preserving schemes perform quite well, even in the presence of discontinuities and complex shock-vortex interactions.

4. Conclusion

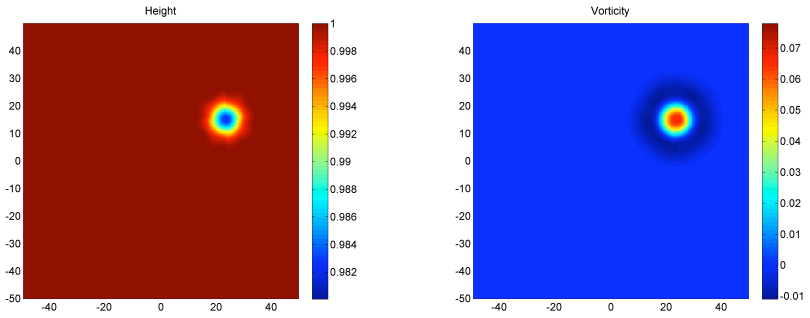
The shallow water equations (1.1) model a number of interesting flows in meteorology and oceanography. The system is non-linear and solutions contains discontinuities like shocks and interesting smooth regions like vortexes. In particular, the vorticity for smooth solutions is advected with the flow (1.5). Standard finite volume schemes may not transport vorticity in the correct manner and result in large vorticity errors. We address this issue and device vorticity *preserving* methods. These methods are based on a projection algorithm.

We illustrate the projection method by considering the system wave equation (1.13), a linearization of the shallow water system. The equation is linear and vorticity is constant in time. The projection method consists of an prediction step in which the solution is evolved by a standard finite volume scheme (1.7) at every time step. The projection consists of solving a Poisson equation (2.3) and a projection step where the field is corrected to obtain the right vorticity. The method preserves vorticity and some stability estimates are also shown. Numerical experiments for the wave equation demonstrates the computational efficiency of this projection method.

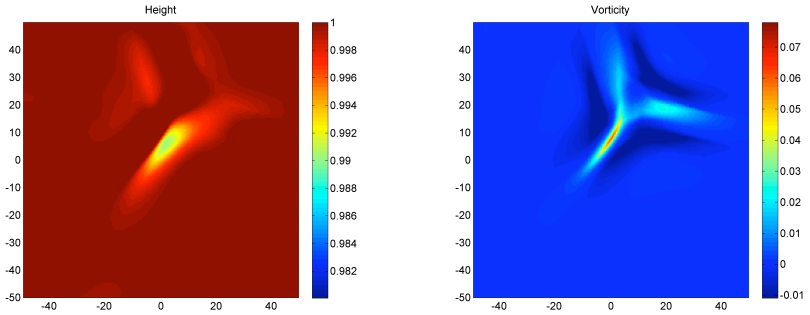
The vorticity projection method is extended to the shallow water equations. The prediction step involves computing approximate solutions with a finite volume scheme. However, the vorticity may no longer be constant in time but is advected with the flow. Furthermore, the momenta are conserved variables. Hence, we use the *pseudo-vorticity* (curl of the momentum) as a measure of the vorticity. The pseudo-vorticity is evolved via a complicated transport equation (1.17). This equation is solved numerically at every time step with a Nessyahu-Tadmor central scheme. The resulting quantity serves as a good estimate for the pseudo-vorticity at the next time step. The momentum field is corrected by an elliptic solver, and the resulting method yields the correct



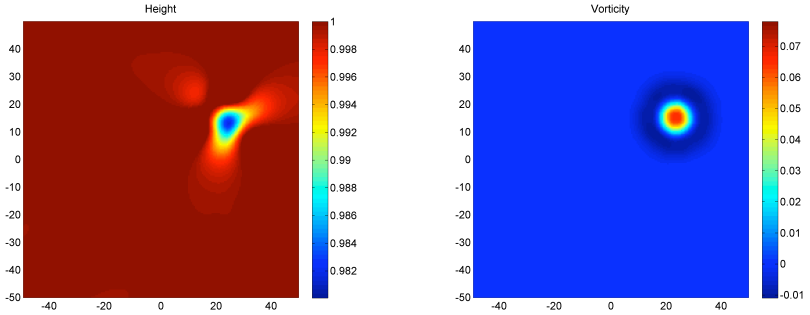
(a) Rusanov2



(b) VPRus2



(c) Roe2



(d) VPRoe2

FIGURE 6. Height and pseudo-vorticity at $t = 100$ computed by the four schemes on a mesh of 200×200 grid points.

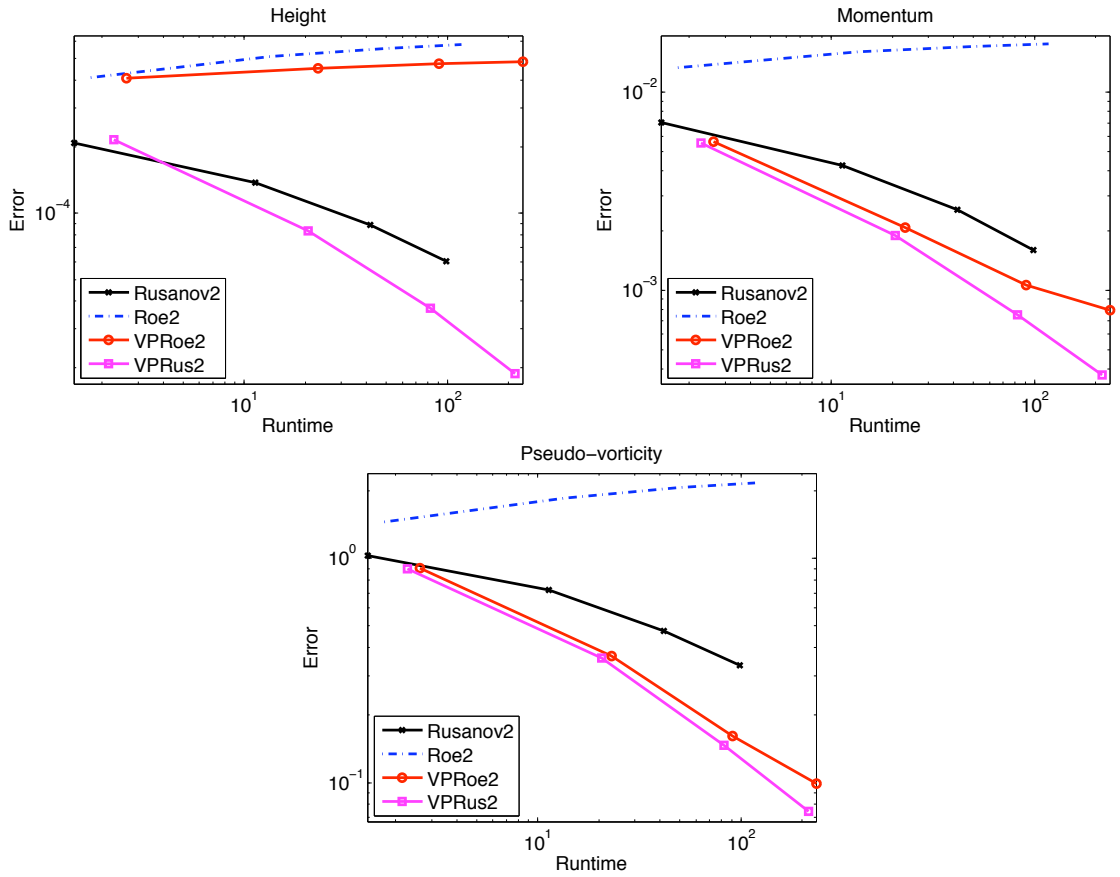


FIGURE 7. Runtime (x-axis) versus relative L^1 errors in height, momentum and vorticity (y-axis). Note the change of axes from Figure 4.

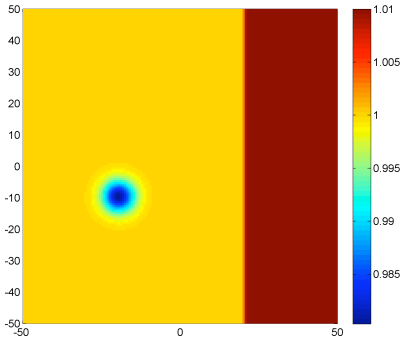


FIGURE 8. Initial height of the shock-vortex interaction problem.

pseudo-vorticity. The vorticity projection algorithm is tested on a challenging vortex advection and shock-vortex interaction problems. The numerical results demonstrate a *considerable* gain in accuracy and computational efficiency due to the vorticity projection method.

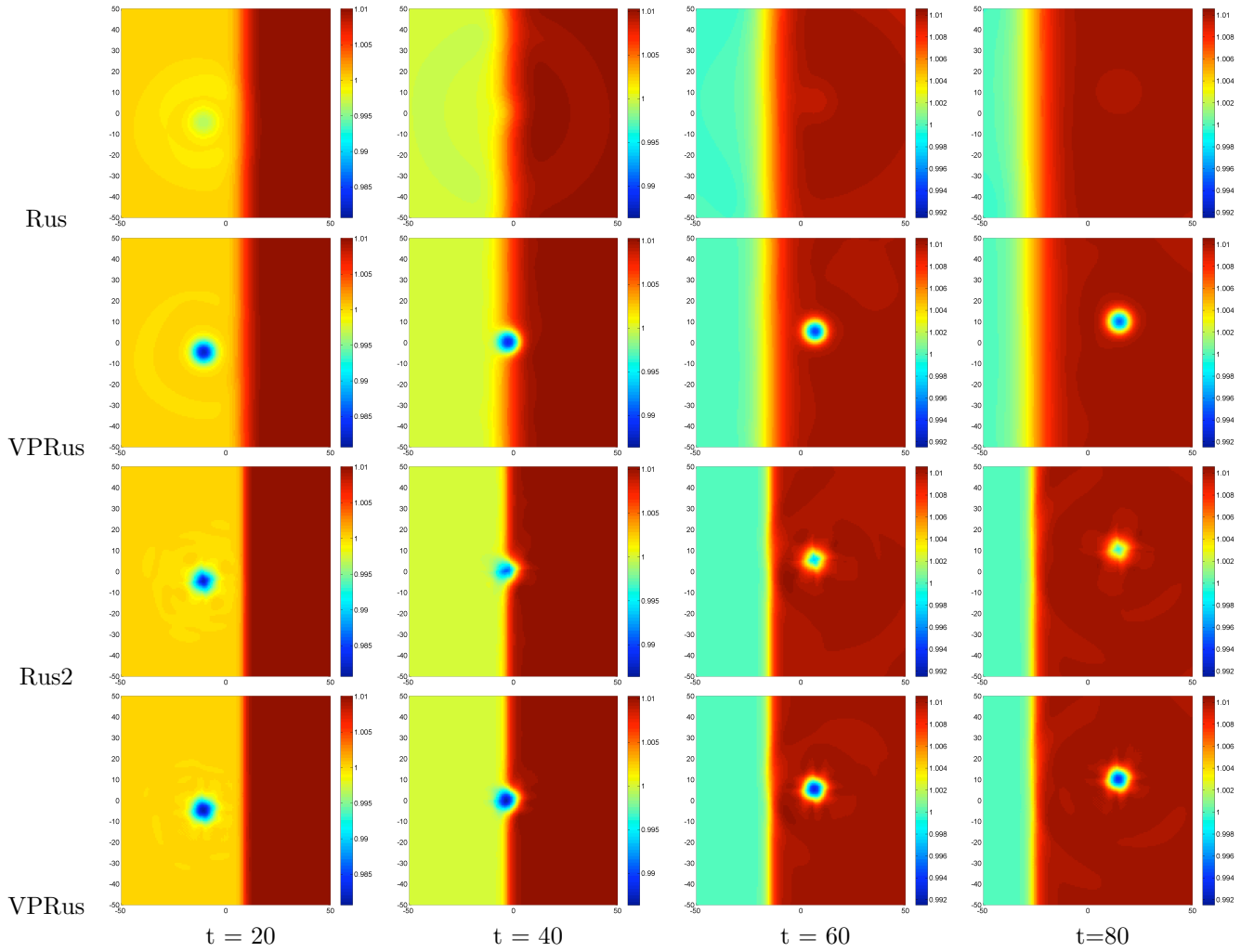


FIGURE 9. Shock-vortex interaction at times $t = 20, 40, 60, 80$.

This paper needs to be viewed as a first step in the design of vorticity preserving schemes for systems of conservation laws. The projection method seems quite promising, at least for the shallow water equations. Further investigation is required for analyzing the pseudo-vorticity transport equation (1.17), particularly on unstructured meshes. The vorticity projection method will be extended to the Euler equations of gas dynamics and to unstructured meshes in forthcoming papers.

REFERENCES

[1] A. Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. *J. Comput. Phys.,* 1 (1), 1966, 119 - 143.
[2] A. Arakawa and V. R. Lamb. Computational design of the basic dynamical process of the UCLA general circulation model. *Meth. Comput. Phys.,* 17, 1977, 173-265.
[3] J. B. Bell, P. Colella and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comp. Phys.*, 85, 257-283, 1989.
[4] J. U. Brackbill and D. C. Barnes. The effect of nonzero Div$B$ on the numerical solution of the magnetohydrodynamic equations. *J. Comp. Phys.*, 35:426-430, 1980.
[5] A. J. Chorin. Numerical solutions of the Navier-Stokes equations. *Math. Comp.*, 22, 745-762, 1968.
[6] C. Dafermos. *Hyperbolic conservation laws in continuum physics.* Springer, Berlin, 2000.

[7] U. S. Fjordholm, S. Mishra and E. Tadmor. Energy preserving and energy stable schemes for the shallow water equations. *"Foundations of Computational Mathematics"*, Proc. FoCM held in Hong Kong 2008 (F. Cucker, A. Pinkus and M. Todd, eds), London Math. Soc. Lecture Notes Ser. 363, pp. 93-139, 2009.
[8] U. S. Fjordholm, S. Mishra and E. Tadmor. Energy preserving and energy stable schemes for the shallow water equations with topography. *Preprint*, 2009.
[9] S. Gottlieb, C. W. Shu and E. Tadmor. High order time discretizations with strong stability property. *SIAM. Review,* 43, 2001, 89 - 112.
[10] A. Harten, B. Engquist, S. Osher and S. R. Chakravarty. Uniformly high order accurate essentially non-oscillatory schemes. *J. Comput. Phys.*, 1987, 231-303.
[11] F. Ismail and P. L. Roe. Towards a vorticity preserving second order finite volume scheme solving the Euler equations. *In proc. of 17th AIAA CFD conference.*
[12] R. Jeltsch and M. Torrilhon. On curl preserving finite volume discretizations of the shallow water equations. *BIT*, 46, 2006, suppl.
[13] A. Kurganov and E. Tadmor. New high resolution central schemes for non-linear conservation laws and convection-diffusion equations. *J. Comput. Phys*, 160(1), 241-282, 2000.
[14] B. van Leer Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method. *J. Com. Phys.,* 32, 101136, 1979.
[15] R. J. LeVeque. Finite volume methods for hyperbolic problems. *Cambridge university press,* Cambridge, 2002.
[16] T. Lyche. Lecture notes for Numerical Linear Algebra. *Downloaded from:* http://www.uio.no/studier/emner/matnat/ifi/INF-MAT4350/h08/undervisningmateriale/book2008.pdf
[17] S. Mishra and E. Tadmor, Constraint preserving schemes using potential-based fluxes. II. Genuinely multi-dimensional central schemes for systems of conservation laws. *Preprint,* 2009.
[18] K. W. Morton and P. L. Roe. Vorticity preserving Lax-Wendroff type schemes for the system wave equation. *SIAM. J. Sci. Comput.,* 23 (1), 2001, 170-192.
[19] H. Nessyahu and E. Tadmor. Non-oscillatory central difference schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 87, 1990, 408-463.
[20] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory schemes - II, *J. Comput. Phys.*, 83, 1989, 32 - 78.
[21] E. Tadmor. Approximate solutions of nonlinear conservation laws. *Advanced Numerical approximations of Nonlinear Hyperbolic equations, A. Quarteroni ed.*, Lecture notes in Mathematics, Springer Verlag (1998), 1-149.
[22] G. Toth. The Div$B = 0$ constraint in shock capturing magnetohydrodynamics codes. *J. Comp. Phys.*,161:605-652, 2000.
[23] G. B. Whitham. Linear and Nonlinear waves. *John Wiley and Sons.,* New York, 1999, 636 pp.

(Ulrik S.Fjordholm)
Seminar for Applied Mathematics (SAM)
D-Math, ETH
Rämistrasse 101, HG J 48,
Zurich - 8092 , Switzerland
E-mail address: ulriksf@gmail.com

(Siddhartha Mishra)
Seminar for Applied Mathematics (SAM)
D-Math, ETH
Rämistrasse 101, HG G 57.2,
Zurich - 8092 , Switzerland
E-mail address: smishra@sam.math.ethz.ch

Research Reports

No.	Authors/Title
09-35	*U.S. Fjordholm, S. Mishra* Vorticity preserving finite volume schemes for the shallow water equations
09-34	*S. Mishra, E. Tadmor* Potential based constraint preserving genuinely multi-dimensional schemes for systems of conservation laws
09-33	*S. Mishra, E. Tadmor* Constraint preserving schemes using potential-based fluxes. III. Genuinely multi-dimensional central schemes for for MHD equations
09-32	*S. Mishra, E. Tadmor* Constraint preserving schemes using potential-based fluxes. II. Genuinely multi-dimensional central schemes for systems of conservation laws
09-31	*S. Mishra, E. Tadmor* Constraint preserving schemes using potential-based fluxes. I. Multidimensional transport equations
09-30	*D. Braess, S. Sauter, C. Schwab* On the justification of plate models
09-29	*D. Schötzau, C. Schwab, T. Wihler* *hp*-dGFEM for second-order elliptic problems in polyhedra. II: Exponential convergence
09-28	*D. Schötzau, C. Schwab, T. Wihler* *hp*-dGFEM for second-order elliptic problems in polyhedra. I: Stability and quasioptimality on geometric meshes
09-27	*A. Moiola, R. Hiptmair, I. Perugia* Approximation by plane waves
09-26	*M. Karow, E. Kokiopoulou, D. Kressner* On the computation of structured singular values and pseudospectra
09-25	*M. Durán, M. Guarini, C.F. Jerez-Hanckes* Hybrid FEM/BEM modeling of finite-sized photonic crystals for semiconductor laser beams
09-24	*A. Bespalov, N. Heuer, R. Hiptmair* Convergence of the natural *hp*-BEM for the electric field integral equation on polyhedral surfaces
09-23	*R. Hiptmair, J. Li, J. Zou* Real interpolation of spaces of differential forms