Large time step finite volume evolution Galerkin methods [1]

A. Hundertmark-Zaušková[2], M. Lukáčová - Medviďová[2] and F. Prill[2]

Abstract

We present two new large time step methods within the framework of the well-balanced finite volume evolution Galerkin (FVEG) schemes. The methodology will be illustrated for low Froude number shallow water flows with source terms modeling the bottom topography and Coriolis forces, but results can be generalized to more complex systems of balance laws. The FVEG methods couple a finite volume formulation with approximate evolution operators. The latter are constructed using the bicharacteristics of multidimensional hyperbolic systems, such that all of the infinitely many directions of wave propagation are taken into account explicitly. We present two variants of large time step FVEG method: a semi-implicit time approximation and an explicit time approximation using several evolution steps along bicharacteristic cones.

Key words: well-balanced schemes, steady states, systems of hyperbolic balance laws, shallow water equations, large time step, semi-implicit approximation, evolution Galerkin schemes

AMS Subject Classification: 65L05, 65M06, 35L45, 35L65, 65M25, 65M15

1 Introduction

Many problems arising in geophysics, engineering or natural sciences lead to hyperbolic balance laws. An interesting example are the shallow water equations with the source term modeling the bottom topography and/or Coriolis forces, which arise in oceanography and atmospheric sciences

$$\boldsymbol{u}_t + \boldsymbol{f}_1(\boldsymbol{u})_x + \boldsymbol{f}_2(\boldsymbol{u})_y = \boldsymbol{b}(\boldsymbol{u}), \tag{1.1}$$

where $\boldsymbol{u} = (h, hu, hv)^T$ are conservative variables. Fluxes and source term are given as

$$\boldsymbol{f}_1(\boldsymbol{u}) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \boldsymbol{f}_2(\boldsymbol{u}) = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}, \ \boldsymbol{b}(\boldsymbol{u}) = \begin{pmatrix} 0 \\ -ghb_x + fhv \\ -ghb_y - fhu \end{pmatrix}. \tag{1.2}$$

[1] This research has been supported by the German Research Foundation DFG under the grant LU 1470/2-1.

[2] Institute of Mathematics, University of Mainz, Staudingerweg 9, 55099 Mainz Germany, emails: {hundertm,lukacova,prill}@uni-mainz.de

Here h denotes the water height, u, v are vertically averaged velocity components in x- and y-direction, g stands for the gravitational constant, f is the Coriolis parameter, and $b(x, y)$ denotes the bottom topography. Shallow flows are characterized by the Froude number $\mathbf{Fr} = \sqrt{u^2 + v^2}/c$, where $c = \sqrt{gh}$ denotes the wave speed. For $\mathbf{Fr} < 1$, $\mathbf{Fr} = 1$ and $\mathbf{Fr} > 1$ the flow is called subcritical, critical and supercritical, respectively. We are interested in problems when bores or hydraulic jumps may develop and thus a suitable numerical method has to take the hyperbolic character of the system into account.

Since many geophysical flows are close to some fundamental equilibrium states, a suitable numerical method has to preserve these states, too. We consider states which are both stationary, $(h, u, v)_t = 0$, and constant along streamlines, $(\dot{h}, \dot{u}, \dot{v}) = 0$. Then the desired solution has to satisfy several conditions

$$u = 0,\ v_y = 0,\ vh_y = 0,\ g(h + b)_x = fv,\ g(h + b)_y = 0.$$

In the region $\{(x, y)|\, v(x, y) = 0\}$ we obtain the *lake at rest solution*, where the water level $h + b$ is flat. When $v(x, y) \neq 0$ we get a balance between a raise in the water level $g(h + b)_x$ and the sidewards pressure due to the earth rotation fv. In meteorological literature this state is called the *geostrophic equilibrium*.

In this paper we will deal with the well-balanced finite volume evolution Galerkin (FVEG) method developed in [4] by Lukáčová, Noelle and Kraft. The FVEG method couples a finite volume formulation with approximate evolution operators which are based on the theory of bicharacteristics for first order systems [3]. As a result exact integral representations for solutions of linear or linearized hyperbolic conservation laws can be derived, which take into account all of the infinitely many directions of wave propagation. The well-balanced FVEG methods preserve exactly the lake at rest solution as well as geostrophic equilibrium [4].

Another typical characteristic of geophysical flows is their multiscale behavior with wave speeds differing by orders of magnitude. If the Froude number is small the gravitational waves are much faster than advection waves. The main goal of the present paper is to improve well-balanced FVEG scheme by alleviating the severe CFL time step restriction for the case of small Froude number problems, i.e. $\mathbf{Fr} << 1$. Indeed, the CFL stability condition for explicit schemes reads

$$\frac{\max(|u| + c, |v| + c)\Delta t}{\hbar} \leq 1, \tag{1.3}$$

here Δt is the time step and $\hbar$ denotes the mesh size. Inserting the Froude number, we obtain the sufficient condition

$$\max c\,(1 + \mathbf{Fr})\frac{\Delta t}{\hbar} \leq 1. \tag{1.4}$$

Now if the Froude number is small then the time step is dictated essentially by the velocity c of gravitational waves. This leads to a classical computational challenge: If one is interested in the effect of advection, which moves with the flow velocity (u, v), then a wave requires $(1 + 1/\mathbf{Fr})$ time steps to pass a single cell. For example, if the Froude number is $\mathcal{O}(10^{-2})$, a vortex would need about 10^2 time steps to pass a single cell. This can be seen by rewriting the CFL condition as

$$\max\left(\left(1 + \frac{1}{\mathbf{Fr}}\right)\sqrt{u^2 + v^2}\right)\frac{\Delta t}{\hbar} \leq 1. \tag{1.5}$$

As a consequence, the numerical solution is very dissipative and computationally expensive. The present paper, as many previous works on low Froude number flows, tries to reduce the adverse effect of the factor $1 + 1/\mathbf{Fr}$ upon the time step Δt: one would like to choose a time step which is as large as that for a purely advective flow, with no gravitational waves present.

In the next section we explain briefly a basic tool of any FVEG scheme, the so-called evolution operators. In Section 3 we introduce an explicit large time step FVEG method. The semi-implicit FVEG method is described in Section 4. Finally, in Section 5 we present numerical experiments comparing accuracy and stability of both large time step FVEG methods with the standard explicit FVEG scheme.

2 Evolution operators

In the predictor step of finite volume evolution Galerkin methods we need firstly to determine point values on cell interfaces. It is here that the classical bicharacteristic theory comes into play. Taking all infinitely many directions of wave propagation into account it provides exact integral formulae for point values of solutions to multidimensional hyperbolic systems.

Time evolution of a solution of a multidimensional system of hyperbolic balance laws takes place along the so-called bicharacteristics. In general, these are curves that evolve in time according to the ray velocities determined by the eigenvalues of hyperbolic systems. If the system is linear or locally linearized, the bicharacteristics are just straight lines. Rewriting the shallow water system (1.1) for characteristic variables and integrating each equation along the corresponding bicharacteristic yields the exact integral representation, see, e.g. [4] for a detailed derivation.

$$
\begin{aligned}
h(P) &= \frac{1}{2\pi}\int_0^{2\pi} h(Q) - \frac{\tilde{c}}{g}\left(u(Q)\cos\theta + v(Q)\sin\theta\right)\mathrm{d}\theta \\
&\quad -\frac{1}{2\pi}\int_{t_n}^{t_n+\tau}\frac{1}{t_n+\tau-\tilde{t}}\int_0^{2\pi}\frac{\tilde{c}}{g}\Big(u(\tilde{Q})\cos\theta + v(\tilde{Q})\sin\theta\Big)\mathrm{d}\theta\mathrm{d}\tilde{t} \\
&\quad +\frac{1}{2\pi}\tilde{c}\int_{t_n}^{t_n+\tau}\int_0^{2\pi}\Big(b_x(\tilde{Q})\cos\theta + b_y(\tilde{Q})\sin\theta\Big)\mathrm{d}\theta\mathrm{d}\tilde{t} \\
&\quad -\frac{1}{2\pi}\frac{\tilde{c}f}{g}\int_{t_n}^{t_n+\tau}\int_0^{2\pi}\Big(v(\tilde{Q})\cos\theta - u(\tilde{Q})\sin\theta\Big)\mathrm{d}\theta\mathrm{d}\tilde{t}, \\
u(P) &= \frac{1}{2}u(Q_0) + \frac{1}{2\pi}\int_0^{2\pi} -\frac{g}{\tilde{c}}h(Q)\cos\theta + u(Q)\cos^2\theta + v(Q)\sin\theta\cos\theta\,\mathrm{d}\theta \\
&\quad -\frac{g}{2}\int_{t_n}^{t_n+\tau}\Big(h_x(\tilde{Q}_0) + b_x(\tilde{Q}_0)\Big)\mathrm{d}\tilde{t} \\
&\quad -\frac{g}{2\pi}\int_{t_n}^{t_n+\tau}\int_0^{2\pi}\Big(b_x(\tilde{Q})\cos^2\theta + b_y(\tilde{Q})\sin\theta\cos\theta\Big)\mathrm{d}\theta\mathrm{d}\tilde{t} \\
&\quad +\frac{1}{2\pi}\int_{t_n}^{t_n+\tau}\frac{1}{t_n+\tau-\tilde{t}}\int_0^{2\pi}\Big(u(\tilde{Q})\cos 2\theta + v(\tilde{Q})\sin 2\theta\Big)\mathrm{d}\theta\mathrm{d}\tilde{t} \\
&\quad +\frac{f}{2}\int_{t_n}^{t_n+\tau}v(\tilde{Q}_0)\mathrm{d}\tilde{t} + \frac{f}{2\pi}\int_{t_n}^{t_n+\tau}\int_0^{2\pi}\Big(v(\tilde{Q})\cos^2\theta - u(\tilde{Q})\sin\theta\cos\theta\Big)\mathrm{d}\theta\mathrm{d}\tilde{t}.
\end{aligned}
\tag{2.1}
$$

The equation for the velocity v is analogous to that for u. In fact, evolution takes place along the bicharacteristic cone, see Fig. 1 for a locally linearized case, where $P = (x, y, t_n + \tau)$ is the peak of bicharacteristic cone, $Q_0 = (x - \tilde{u}\tau, y - \tilde{v}\tau, t_n)$ denotes the center of the sonic circle at time t_n, $\tilde{Q}_0 = (x - \tilde{u}(t_n + \tau - \tilde{t}), y - \tilde{v}(t_n + \tau - \tilde{t}), \tilde{t})$, $\tilde{Q} = (x - \tilde{u}(t_n + \tau - \tilde{t}) + c(t_n + \tau - \tilde{t}) \cos\theta, y - \tilde{v}(t_n + \tau - \tilde{t}) + c(t_n + \tau - \tilde{t}) \sin\theta, \tilde{t})$ stands for an arbitrary point on the mantle, $\tilde{u}, \tilde{v}, \tilde{c}$ are locally fixed flow and wave velocities at (x, y, t_n), and $Q = Q(\tilde{t})\Big|_{\tilde{t}=t_n}$ denotes a point at the perimeter of the sonic circle at time t_n.

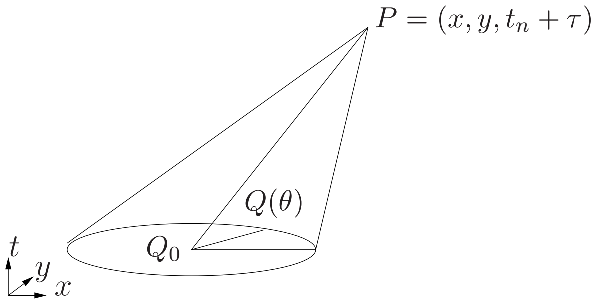

Figure 1: Bicharacteristic cone.

In [4] we have derived a well-balanced time explicit approximation of exact evolution equations (2.1). For the sake of completeness we recall here the *approximate evolution operator* E_τ; for piecewise constant data $\boldsymbol{U} = (h, hu, hv)$ we denote $\hat{\boldsymbol{U}} = E_\tau \boldsymbol{U}$.

$$\begin{aligned}
\hat{h}\,(P) &= -b(P) + \frac{1}{2\pi}\int_0^{2\pi} \left[(h\,(Q) + b(Q)) - \frac{\tilde{c}}{g}\,(u\,(Q)\,\mathrm{sgn}(\cos\theta) + v\,(Q)\,\mathrm{sgn}(\sin\theta)) \right] \mathrm{d}\theta \\
&\quad + \frac{\tau}{2\pi}\int_0^{2\pi} (\tilde{u} b_x(Q) + \tilde{v} b_y(Q))\,\mathrm{d}\theta, \qquad (2.2) \\
\hat{u}\,(P) &= \frac{1}{2\pi}\int_0^{2\pi} \left[-\frac{1}{\tilde{c}} K\,(Q)\,\mathrm{sgn}(\cos\theta) + u\,(Q) \left(\cos^2\theta + \frac{1}{2} \right) + v\,(Q) \sin\theta\cos\theta \right] d\theta,
\end{aligned}$$

here $K := g(h + b) - f \int v \mathrm{d}x$ is the potential energy in the x-direction. An analogous operator has been derived for bilinear data.

Further, in [8] the so-called *local approximate operator* has been derived by Sun and Ren. The main idea of the method is to apply the approximate evolution operator in the limiting case as $\tau \to 0$. Hence only the integrals along the base of the bicharacteristic cones are taken into account, mantle integrals in (2.1) vanish, see [8] for a detailed derivation. The approximate evolution operator E_0 reads

$$\begin{aligned}
\hat{h}(P) &= -b(P) + \frac{1}{2\pi}\int_0^{2\pi} \left[h(Q) + b(Q) - \frac{\tilde{c}}{g}\,(u(Q)\cos\theta + v(Q)\sin\theta) \right] \mathrm{d}\theta, \\
\hat{u}(P) &= \frac{1}{\pi}\int_0^{2\pi} \left[-\frac{1}{\tilde{c}} K(Q)\cos\theta + u(Q)\cos^2\theta + v(Q)\sin\theta\cos\theta \right] \mathrm{d}\theta. \qquad (2.3)
\end{aligned}$$

The main advantage of the local evolution is a decoupling of time evolution and spatial integration. Thus, the local evolution operator may be easily used in semi-discrete FV schemes.

In what follows we will use the above approximate evolution operators in the large time step FVEG method. In Section 3 the approximate evolution operator (2.2) will be used in order to derive *the explicit large time step FVEG method.* Note that the evolution may take place in the positive as well as negative temporal direction, i.e. from t_n to $t_n+\Delta t$ as well as from $t_n+\Delta t$ to t_n. Applying the approximate evolution operators $E^+_{\Delta t}$ and $E^-_{\Delta t}$ we can obtain predicted data at cell interfaces at different time instances, i.e. $\hat{\boldsymbol{U}}^{n+1} := E^+_{\Delta t}\boldsymbol{U}^n$, $\hat{\boldsymbol{U}}^{n} := E^-_{\Delta t}\boldsymbol{U}^{n+1}$. These can be used in the semi-implicit finite volume update in order to evaluate fluxes along cell interfaces, see Section 4. Unfortunately, this approach yields an unstable scheme. Indeed, even for the scalar Burgers equation a semi-implicit characteristics based finite volume method is unstable for large time steps. This is the reason to construct *the semi-implicit FVEG scheme* by means of the local evolution operator (2.3), see Section 4, (4.3).

3 Explicit large time step FVEG method

Let us divide a computational domain Ω into a finite number of regular finite volumes $\Omega_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]\times[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] = [x_i-\hbar/2, x_i+\hbar/2]\times[y_j-\hbar/2, y_j+\hbar/2]$, $i,j\in\{1,\ldots,N\}$, where $\hbar$ is the mesh size. Denote by $\boldsymbol{U}^n_{ij}$ the piecewise constant approximate solution on a mesh cell Ω_{ij} at time t_n and start with initial approximations obtained by the integral averages $\boldsymbol{U}^0_{ij} = \frac{1}{|\Omega_{ij}|}\int_{\Omega_{ij}}\boldsymbol{U}(\cdot,0)$.

Further, let us divide the time interval $[0,T]$ into M large time steps $[t_n,t_{n+1}]$ with length Δt, $t_{n+1}=t_n+\Delta t$. In order to construct a method that approximates advection waves explicitly we enforce a stability CFL condition $\max(|u|,|v|)\Delta t/\hbar \le 1$. Additionally, we divide a large time step$[t_n,t_{n+1}]$ into L small evolution time steps with length $\tau := \Delta t/L$. Here the substep τ is chosen in such a way that a classical CFL stability condition is fulfilled, i.e. $\max(|u|+c,|v|+c)\tau/\hbar\le 1$. Consequently, we have the following intermediate time steps: $t_n, t_n+\tau, t_n+2\tau,\ldots,t_n+L\tau\equiv t_{n+1}$.

The FVEG method is a predictor-corrector method. In the predictor step the approximate evolution operators are used to evolve the solution along the cell interfaces up to intermediate time levels. In the corrector step the finite volume update is done over a large time step

$$\boldsymbol{U}^{n+1}_{ij} = \boldsymbol{U}^n_{ij} - \frac{\Delta t}{\hbar}\sum_{k=1}^{2}\delta^{ij}_{x_k}\bar{\boldsymbol{f}}^{n+1/2}_k + \frac{\Delta t}{\hbar}\bar{\boldsymbol{B}}^{n+1/2}_{ij}. \tag{3.1}$$

Here $\delta^{ij}_{x_k}$ stands for the central difference operator in the x_k-direction, $k=1,2$ and $\bar{\boldsymbol{f}}^{n+1/2}_k$ represents a time average of the edge flux over $[t_n,t_{n+1}]$, cf. (3.3). Further, $\bar{\boldsymbol{B}}^{n+1/2}_{ij}$ stands for the approximation of the source term multiplied by the mesh size, $\hbar\boldsymbol{b}$.

To approximate a nonlinear evolution from t_n to t_{n+1} in the predictor step we apply L small evolution steps along the corresponding bicharacteristic cones from $t_n+\ell\tau$ to $t_n+(\ell+1)\tau$, $\ell=0,1,\ldots,L-1$. The cell interface fluxes $\boldsymbol{f}^{n+(2\ell-1)/2L}_k$ are evolved using an approximate evolution operator denoted by E_τ from the old small time step into a new one, i.e. from t_n to $t_n+\tau/2$, from $t_n+\tau/2$ to $t_n+3\tau/2$, etc. In order to increase the accuracy of the method we apply a recovery step after each small evolution time step, e.g. we have for $\ell=2,3,\ldots,L$ and a suitable bilinear recovery R_h

$$\hat{\boldsymbol{U}}^{n+(2\ell-1)/2L} := E_\tau R_h\boldsymbol{U}^{n+(2\ell-3)/2L} \quad\text{and}\quad \hat{\boldsymbol{U}}^{n+1/2L} := E_{\tau/2}R_h\boldsymbol{U}^n.$$

Now the fluxes are averaged along the cell interface edge,

$$\boldsymbol{f}_k^{n+(2\ell-1)/2L} \quad := \quad \sum_j \omega_j \boldsymbol{f}_k(\hat{\boldsymbol{U}}^{n+(2\ell-1)/2L}(\boldsymbol{x}^j)), \qquad \ell = 1, 2, \ldots, L. \tag{3.2}$$

Here $\boldsymbol{x}^j$ are the nodes and ω_j the weights of the quadrature for the flux integration along the edges. Finally, applying the composite midpoint rule for time integration we obtain the time average of cell interface fluxes

$$\bar{\boldsymbol{f}}_k^{n+1/2} := \frac{\tau}{\Delta t} \sum_{\ell=1}^{L} \boldsymbol{f}_k^{n+(2\ell-1)/2L}. \tag{3.3}$$

Further, in order to describe the well-balanced approximation of sources we introduce the following notation. Along the edges we have quadrature nodes $(x_{i\pm\frac{1}{2}}, y_{j+j'})$ resp. $(x_{i+i'}, y_{j\pm\frac{1}{2}})$, where $i', j' \in \{0, \pm\frac{1}{2}\}$. These nodes are already sufficient for the midpoint, the trapezoidal and Simpson's rule. In order to discretize the source term in a well-balanced manner we apply again the same quadrature rule for edge integration as for the flux interface integration

$$\boldsymbol{B}_{ij}^{n+(2\ell-1)/2L} = -\, g \begin{pmatrix} 0 \\ \sum_{j'} \omega_{j'} \, (\mu_{x_1}^{i,j+j'} \, \hat{h}_{i,j+j'}^{(2\ell-1)/2L}) \, (\delta_{x_1}^{i,j+j'} \, (\hat{b} - \hat{V})_{i,j+j'}^{(2\ell-1)/2L}) \\ \sum_{i'} \omega_{i'} \, (\mu_{x_2}^{i+i',j} \, \hat{h}_{i+i',j}^{(2\ell-1)/2L}) \, (\delta_{x_2}^{i+i',j} \, (\hat{b} + \hat{U})_{i+i',j}^{(2\ell-1)/2L}) \end{pmatrix}. \tag{3.4}$$

Here $\hat{U}$ and $\hat{V}$ are the discrete primitives of the Coriolis forces, defined by

$$\delta_{x_1}^{i,j+j'} \hat{V}_{i,j+j'}^{(2\ell-1)/2L} = \frac{f}{g} \, \mu_{x_1}^{i,j+j'} \hat{v}_{i,j+j'}^{(2\ell-1)/2L} \qquad \delta_{x_2}^{i+i',j} \hat{U}_{i+i',j}^{(2\ell-1)/2L} = \frac{f}{g} \, \mu_{x_2}^{i+i',j} \hat{u}_{i+i',j}^{(2\ell-1)/2L}$$

and μ_{x_1}, μ_{x_2} the average operators $\mu_{x_1}^{ij} a = (a_{i+1/2,j} + a_{i-1/2,j})/2$, $\mu_{x_2}^{ij} a = (a_{i,j+1/2} + a_{i,j-1/2})/2$. Consequently, the time average of the source term is defined analogously to (3.3) as

$$\bar{\boldsymbol{B}}_{ij}^{n+1/2} := \frac{\tau}{\Delta t} \sum_{\ell=1}^{L} \boldsymbol{B}_{ij}^{n+(2\ell-1)/2L}. \tag{3.5}$$

Let us note that if $L = 1$ the large time step explicit FVEG method reduces to the standard FVEG developed in [4].

4 Semi-implicit large time step FVEG method

Another way to overcome a restrictive CFL condition (1.5) dictated by fast gravitational waves is to use semi-implicit time discretization. In the semi-implicit FVEG method we apply the trapezoidal rule for time integration of cell interface fluxes and sources

$$\boldsymbol{U}_{ij}^{n+1} = \boldsymbol{U}_{ij}^{n} - \frac{\Delta t}{2\hbar} \sum_{k=1}^{2} \left(\delta_{x_k}^{ij} \boldsymbol{f}_k^{n} + \delta_{x_k}^{ij} \boldsymbol{f}_k^{n+1} \right) + \frac{\Delta t}{2\hbar} \left(\boldsymbol{B}_{ij}^{n} + \boldsymbol{B}_{ij}^{n+1} \right). \tag{4.1}$$

Integrals along cell interfaces are approximated by a suitable quadrature rule; for $\ell = 0, 1$ and quadrature weights ω_j we have

$$\boldsymbol{f}_k^{n+\ell} := \sum_j \omega_j \boldsymbol{f}_k(\hat{\boldsymbol{U}}^{n+\ell}(\boldsymbol{x}^j)). \tag{4.2}$$

Using the local evolution operator E_0 we predict data at cell interfaces at the corresponding quadrature points

$$\hat{\boldsymbol{U}}^n := E_0 R_h \boldsymbol{U}^n \quad \text{and} \quad \hat{\boldsymbol{U}}^{n+1} := E_0 R_h \boldsymbol{U}^{n+1}. \tag{4.3}$$

Note that the source terms $\boldsymbol{B}_{ij}^n$ and $\boldsymbol{B}_{ij}^{n+1}$, cf. (3.4), are also evaluated using $\hat{\boldsymbol{U}}^n$ and $\hat{\boldsymbol{U}}^{n+1}$, respectively.

The fully nonlinear semi-implicit equation (4.1) will now be solved iteratively by the Newton method. For this let us rewrite (4.1) in the following way

$$\mathcal{F}(\boldsymbol{U}^{n+1}) = \mathcal{G}(\boldsymbol{U}^n), \qquad \boldsymbol{U} = (\boldsymbol{U}_{1,1}, \dots \boldsymbol{U}_{N,N})^T$$

with

$$\begin{aligned} \mathcal{F}(\boldsymbol{U}_{i,j}^{n+1}) &= \boldsymbol{U}_{i,j}^{n+1} + \frac{\Delta t}{2\hbar}\left[\sum_{k=1}^{2} \delta_{x_k}^{ij} \boldsymbol{f}_k^{n+1} - \boldsymbol{B}_{ij}^{n+1}\right] \\ \mathcal{G}(\boldsymbol{U}_{i,j}^{n}) &= \boldsymbol{U}_{i,j}^{n} - \frac{\Delta t}{2\hbar}\left[\sum_{k=1}^{2} \delta_{x_k}^{ij} \boldsymbol{f}_k^{n} - \boldsymbol{B}_{ij}^{n}\right]. \end{aligned} \tag{4.4}$$

We are looking for a solution $\boldsymbol{U}^{n+1}$ satisfying

$$\mathcal{T}(\mathbf{U}^{n+1}) := \mathcal{F}(\mathbf{U}^{n+1}) - \mathcal{G}(\mathbf{U}^n) = 0. \tag{4.5}$$

Starting with $\boldsymbol{U}^{n+1,1} := \boldsymbol{U}^n$ and applying the Newton method for (4.5) we obtain

$$\frac{d\mathcal{T}(\boldsymbol{U}^{n+1,k})}{d\boldsymbol{U}}(\boldsymbol{U}^{n+1,k+1} - \boldsymbol{U}^{n+1,k}) = -\mathcal{T}(\boldsymbol{U}^{n+1,k}), \quad k = 1, 2, \dots,$$

This leads to a linear system $\boldsymbol{AX} = \boldsymbol{R}$

$$\underbrace{\frac{d\mathcal{F}(\boldsymbol{U}^{n+1,k})}{d\boldsymbol{U}}}_{\boldsymbol{A}} \underbrace{(\boldsymbol{U}^{n+1,k+1} - \boldsymbol{U}^{n+1,k})}_{\boldsymbol{X}} = \underbrace{\mathcal{G}(\boldsymbol{U}^n) - \mathcal{F}(\boldsymbol{U}^{n+1,k})}_{\boldsymbol{R}}. \tag{4.6}$$

The Jacobian matrix $\boldsymbol{A} \in R^{3N^2} \times R^{3N^2}$ consists of blocks corresponding to the update equations for conservative variables h, hu, hv.

$$\frac{d\,\mathcal{F}(\boldsymbol{U}^{n+1,k})}{d\,\boldsymbol{U}} = \begin{bmatrix} I + \frac{d\,\mathcal{F}^1\left(\boldsymbol{U}^{n+1,k}\right)}{d\,h} & \frac{d\,\mathcal{F}^1\left(\boldsymbol{U}^{n+1,k}\right)}{d\ \ (hu)} & \frac{d\,\mathcal{F}^1\left(\boldsymbol{U}^{n+1,k}\right)}{d\,(hv)} \\ \frac{d\,\mathcal{F}^2\left(\boldsymbol{U}^{n+1,k}\right)}{d\,h} & I + \frac{d\,\mathcal{F}^2\left(\boldsymbol{U}^{n+1,k}\right)}{d\,(hu)} & \frac{d\,\mathcal{F}^2\left(\boldsymbol{U}^{n+1,k}\right)}{d\,(hv)} \\ \frac{d\,\mathcal{F}^3\left(\boldsymbol{U}^{n+1,k}\right)}{d\,h} & \frac{d\,\mathcal{F}^3\left(\boldsymbol{U}^{n+1,k}\right)}{d\,(hu)} & I + \frac{d\,\mathcal{F}^3\left(\boldsymbol{U}^{n+1,k}\right)}{d\,(hv)} \end{bmatrix},$$

here $\mathcal{F}^m$, $m = 1, 2, 3$, are the respective parts of the update operator $\frac{\Delta t}{2\hbar} \left[\sum_{k=1}^{2} \delta_{x_k}^{ij} \boldsymbol{f}_k - \boldsymbol{B}_{ij} \right]$, cf. (4.4), corresponding to the update equations for h, hu and hv.

The structure of the system matrix has a crucial influence on the performance of the overall iteration. In the numerical experiments the number of Newton steps depends on the time step size Δt and thus on the CFL number. Furthermore, we should point out that the low Froude number problems yield a badly conditioned matrix $\boldsymbol{A}$ and special preconditioning techniques have to be used to solve (4.6) iteratively, see, e.g., techniques developed in [5] for the case of low Mach number flows. On the other hand, if the size of discrete problem is moderate, i. e. the computational mesh contains less than $\mathcal{O}(10^5 - 10^6)$ cells, efficient direct solvers can be applied as well. In the numerical experiments presented in this paper the direct solver UMFPACK [2] has been successfully to the linear problems (4.6). The abort criterion for the Newton method is given by $\|\mathcal{T}(\boldsymbol{U}^{n+1,k+1})\| \le \varepsilon$ with a tolerance $\varepsilon := 10^{-4}$ in all computations presented below. Typically the Newton iteration terminates after a small number of steps and we get $\boldsymbol{U}^{n+1} := \boldsymbol{U}^{n+1,k+1}$ as an approximate solution of (4.5).

The computation of the Jacobian matrix $\boldsymbol{A}$ was based on forward mode automatic differentiation. More precisely, the C++ library *Sacado* has been employed, which is part of the *Trilinos* package [7]. For an efficient assembly of (4.6), the derivatives of the separable sum (4.1) were computed cell-wise, with the exception of the non-local Coriolis force terms. On structured cartesian meshes the FVEG method results in compact 3 × 3 and 5 × 5 stencils for the first and second order method, respectively. In order to attain cache efficiency, the unknowns of the state vector have been reordered in a pointwise manner and the differentiation of the stencils was distributed over multiple cores of an Intel Xeon Quad Core architecture. The whole approach makes use of template directives and operator overloading.

5 Numerical experiments

In what follows we present behavior of both large time steps FVEG schemes on a set of numerical experiments. The time step is controlled just by the CFL condition dictated by the advection velocity

$$CFL_{adv} := \frac{\max(|u|, |v|)\Delta t}{\hbar}, \qquad CFL_{adv} \le 1. \tag{5.1}$$

Numerical results presented below demonstrate stability and robustness of the large time step FVEG schemes; in particular also in the case of small Froude numbers when

$$CFL_{total} \gg 1, \quad CFL_{total} := \frac{\max(|u| + c, |v| + c)\Delta t}{\hbar}. \tag{5.2}$$

Example 1

We the choose bottom topography and the initial data as

$$\begin{aligned} b(x, y) &= \sin(2\pi x) + \cos(2\pi y), \quad h(x, y, 0) = 10 + \exp(\sin(2\pi x)) \cos(2\pi y), \\ hu(x, y, 0) &= \sin(\cos(2\pi x)) \sin(2\pi y), \quad hv(x, y, 0) = \cos(2\pi x) \cos(\sin(2\pi y)). \end{aligned}$$

The gravitational constant is set to $g = 9.812$ and the Coriolis parameter is $f = 10$. The computational domain $[0, 1] \times [0, 1]$ was divided into 200 × 200 mesh cells and periodic boundary conditions have been applied. In this experiment small Froude number flow develops with the Froude number varying from about 10^{-2} to 0.36.

In Figure 2 results obtained by the first and second order large time step semi-implicit FVEG schemes at time $t = 0.05$ are presented. In the second order method a bilinear recovery with minmod limiter has been applied. At the beginning of the simulation fast dynamics take place with large velocities. In order to approximate appropriately rapidly changing quantities we set $CFL_{adv} = 0.01$ in the first time step and allow a new time step to be at most twice larger as a previous one at the beginning, afterwards $CFL_{adv} = 0.8$. The reference solution has been computed by the standard second order explicit FVEG scheme [4] using small time steps, $CFL_{total} = 0.5$.

In Table 1 variations of the Froude number **Fr**, CFL_{total}, number of Newton's iterations as well as the corresponding time steps during the evolution process are illustrated for the second order semi-implicit FVEG method using a mesh with 100 × 100 cells and $CFL_{adv} = 0.6$. Table 2 illustrates the performance of semi-implicit FVEG scheme on different meshes. Additionally, this table contains the wall clock time required by the UMFPACK linear solver.

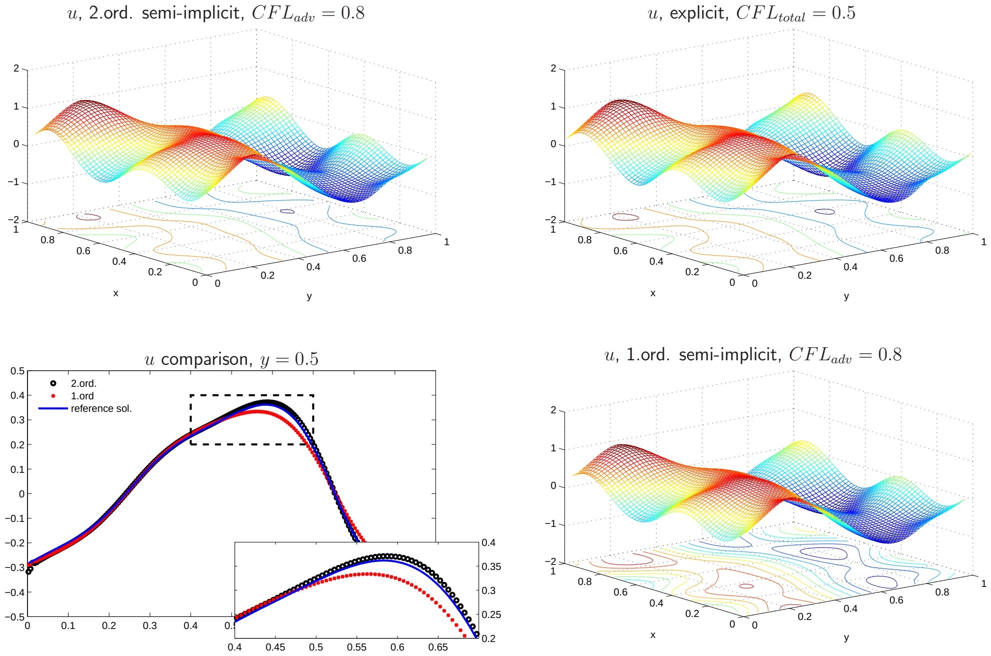

Figure 2: Solution obtained on a mesh with 200×200 cells by the first (bottom right) and second order (top left) semi-implicit FVEG method, u-velocity component. Reference solution (top right), graph of cross section at $y = 0.5$ (bottom left).

Figure 3 compares results obtained by the first and second order large time step explicit FVEG method. We set $CFL_{adv} = 0.9$ for the second order method, for the first order method it has to be reduced to $CFL_{adv} = 0.5$. The reference solution is the same as in Figure 2. Setting $CFL_{adv} = 0.9$ we have $CFL_{total} \in (5.46, 89.89)$. Thus, the number of $\tau-$substeps may vary from 6 up to 90. Despite of this fact the results presented in Figure 3 demonstrate robustness of the explicit large time step FVEG method even for flows with strongly varying Froude numbers.

time step	Δt	# Newton's iter.	**Fr**	CFL_{total}	CPU
1.	0.0009	10	0.0120	0.99	46 s
2.	0.0018	14	0.0233	2.00	1 min 49 s
3.	0.0036	23	0.0627	4.03	3 min 32 s
20.	0.0035	19	0.2248	4.42	23 min 55 s
21. (final)	0.0006	8	0.2084	0.81	24 min 31 s

Table 1: Variation of time steps, Froude numbers, CFL_{total} and CPU time during time evolution. Results obtained for the second order semi-implicit FVEG method on a mesh with 100 × 100 mesh cells, $CFL_{adv} = 0.6$.

problem size	max(CFL_{total})	CPU	CPU UMFPACK
20 × 20	2.987	11.6 s	5.07 s
40 × 40	4.184	1 min 24 s	46.2 s
80 × 80	4.236	11 min 54 s	7 min 41 s
100 × 100	5.078	24 min 31 s	16 min 57 s
160 × 160	7.726	1 h 44 min 4 s	1 h 15 min 32 s
200 × 200	8.082	3 h 35 min 7 s	2 h 40 min 39 s

Table 2: Performance of the semi-implicit FVEG scheme for different problem sizes, $CFL_{adv} = 0.6$.

Table 3 illustrates variations of **Fr**, CFL_{total} and the corresponding time steps for the second order large time step explicit FVEG method on a grid with 100 × 100 cells, $CFL_{adv} = 0.6$. Clearly, the explicit large time step FVEG method is computationally more efficient, the total CPU time needed for the second order FVEG method is just about 2 seconds (about 22 seconds on a grid with 200 × 200 cells).

time step	Δt	# τ-steps	**Fr**	CFL_{total}	CPU
1.	0.0269	30	0.0120	29.993	0.72 s
2.	0.0021	3	0.3140	2.5945	0.87 s
8.	0.0030	4	0.2420	3.7711	1.84 s
9.	0.0034	5	0.2209	4.2947	2.03 s
10.(final)	0.002632	4	0.1952	3.3019	2.10 s

Table 3: Variation of time steps, Froude numbers and CFL_{total} and CPU during time evolution. Results obtained for the second order large time step explicit FVEG method on a mesh with 100 × 100 mesh cells, $CFL_{adv} = 0.6$.

Figure 4 illustrates the comparison of both second order large time step FVEG methods, semi-implicit and explicit one. The CFL number $CFL_{adv} = 0.8$ has been used for both methods. Numerical results demonstrate clearly that both large time steps FVEG methods approximate solution in a very good manner. The explicit large time step method is faster than the semi-implicit FVEG scheme. This is mainly due to the linear solver and due to the computation of Jacobian matrices by means of the automatic differentiation that is computationally expensive. In order to decrease computational time further optimization and parallelization of the semi-implicit code is required in the future. On the other hand, the semi-implicit scheme yields slightly more accurate results. In particular,

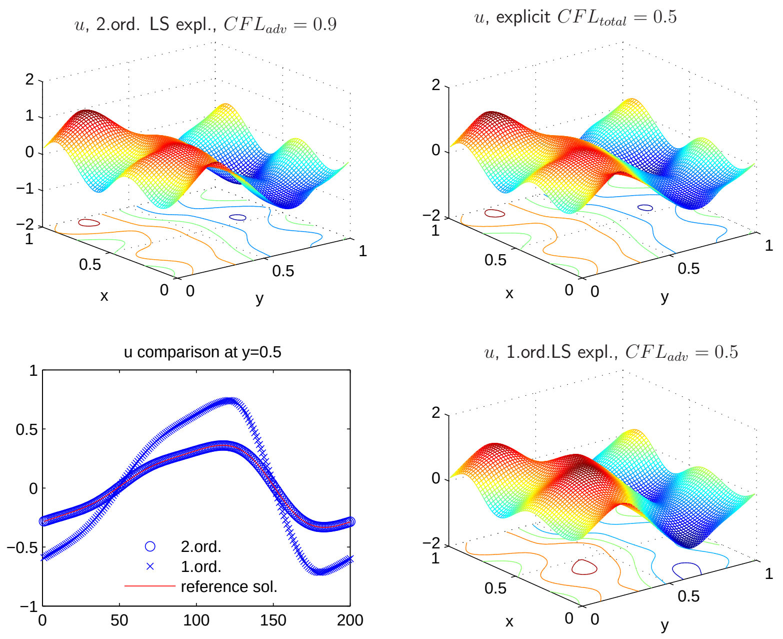

Figure 3: Solution obtained on a mesh with 200 × 200 cells by the first (bottom right) and second order (top left) large time step explicit FVEG method, u-velocity component. Reference solution (top right), graph of cross section at $y = 0.5$ (bottom left).

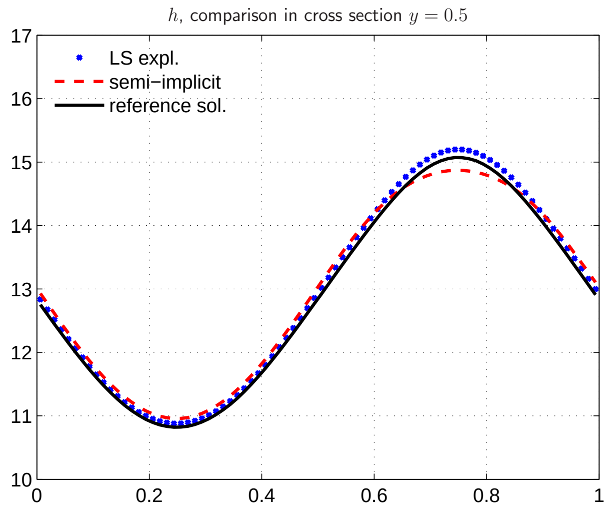

Figure 4: Comparison of the second order large time step explicit and semi-implicit FVEG methods on a mesh with 80 × 80 cells, water height h. Graph of cross section at $y = 0.5$.

the accuracy and stability of the first order explicit FVEG scheme is less favorable. This is due to many nonconservative evolution steps with constant approximate data.

Example 2

On a computational domain $[0, 1] \times [0, 1]$ we simulate a vortex with center starting at $x_C = (0.5, 0.5)$ and moving from left to right with the velocity $(u_\infty, v_\infty) = (0.6, 0)$. We apply to the left and right periodic boundary conditions whereas at the top and bottom the boundary is set to weak farfield conditions. The initial data are the following

$$h(r_C) = 110 + \begin{cases} \frac{1}{g}(\frac{1.5}{\omega})^2(H(\omega r_C) - H(\pi)) & \text{if } \omega\, r_C \le \pi \\ 0 & \text{otherwise} \end{cases}$$

with

$$H(x) = 2\cos(x) + 2x\sin(x) + \frac{1}{8}\cos(2x) + \frac{x}{4}\sin(2x) + \frac{12}{16}x^2$$

and

$$(u,v)(r_C) = (u_\infty, v_\infty) + \begin{cases} 1.5(1+\cos(\omega r_C))(0.5 - x_2, x_1 - 0.5) & \text{if } \omega\, r_C \le \pi \\ 0 & \text{otherwise,} \end{cases}$$

Here r_C denotes the distance from x_C, ω is an angular wave frequency, which we set to $\omega = 4\pi$ and the gravitational constant is chosen as $g = 1$ for this case. This is a variant of a classical example, see e.g. [6], where we have chosen parameters of initial data in such a way that small Froude number flow develops, **Fr** ≈ 0.0782.

In what follows we present results of numerical experiments obtained on a grid with 200×200 cells at final time $t = 0.1$. In Figure 5 solutions obtained by a standard second order explicit FVEG method [4], the second order explicit large time step FVEG method and the second order semi-implicit FVEG method are compared. We use a bilinear recovery without limiter in this case. In order to obtain a reference solution by the standard explicit FVEG scheme CFL_{total} is set to 0.5. For the large time step methods (both explicit and semi-implicit) the CFL condition (5.1) is applied with $CFL_{adv} = 0.6$ Additionally, for the large time step explicit FVEG method small τ-substeps are chosen in such a way, that $\max(|u| + c, |v| + c)\tau/\hbar = 0.5$. Correspondingly, we have $CFL_{total} \approx 8.5$, cf. (5.2). Numerical results presented in Figure 5 clearly show that both large time step FVEG schemes yield very good approximation of the reference solution, see also Figure 6, where a solution on a mesh with 300×300 cells has been computed for a whole period up to $t = 5/3$.

In Table 4 experimental order of convergence as well as global L^1 errors are presented. We can notice that the second order semi-implicit method converges slightly faster and yields smaller global errors on finer meshes than the large time step explicit scheme. As already mentioned in the previous experiment the semi-implicit scheme is quite expensive computationally: in this experiment we need about two hours to compute results on a mesh with 200×200 cells, whereas the large time step explicit method just needs about 32 seconds.

6 Conclusions

In this paper we have derived two new large time step FVEG methods. In the *explicit large time step FVEG method* the finite volume update is realized over a large time step Δt, whereas several small evolution steps (τ-substeps) are applied in order to approximate nonlinear evolution along Δt. The *semi-implicit method* is based on the Crank-Nicolson scheme for time discretization and the use of local evolution operators at the old and new time levels. The resulting fully nonlinear system is solved iteratively by the Newton method. The direct solver UMFPACK has been used for linear systems. The time step is controlled by the CFL condition (5.1) dictated just by the flow velocities with no fast

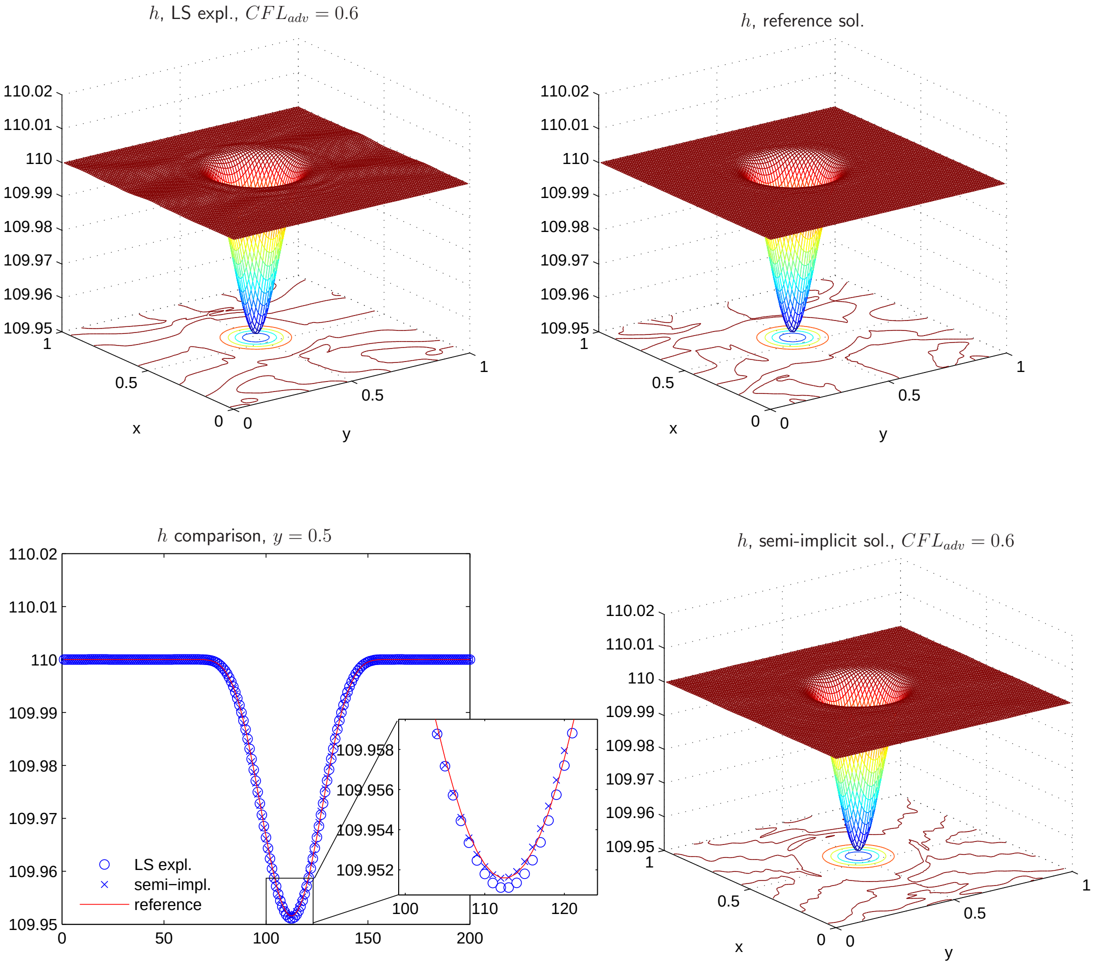

Figure 5: Water height h at $t = 0.1$ computed on a mesh with 200×200 cells by the explicit large time step FVEG method (top left) and the semi-implicit FVEG method (bottom right). Reference solution is obtained by the standard FVEG method (top right), graph of cross section at $y = 0.5$ (bottom left).

gravitational waves present. Since the evolution step as well as the finite volume update are done in the well-balanced way, cf. [4], both large time step FVEG methods are well-balanced and preserve lake at rest state and the geostrophic equilibrium in an analogous way as the standard FVEG scheme with small time steps.

Numerical experiments demonstrate that both large time step schemes yield good resolution of low Froude number flows. With respect to CPU time the explicit large time step FVEG method outperforms clearly the semi-implicit scheme. In our future work further development of the semi-implicit approach is necessary in order to improve the efficiency of the large time step semi-implicit FVEG method.

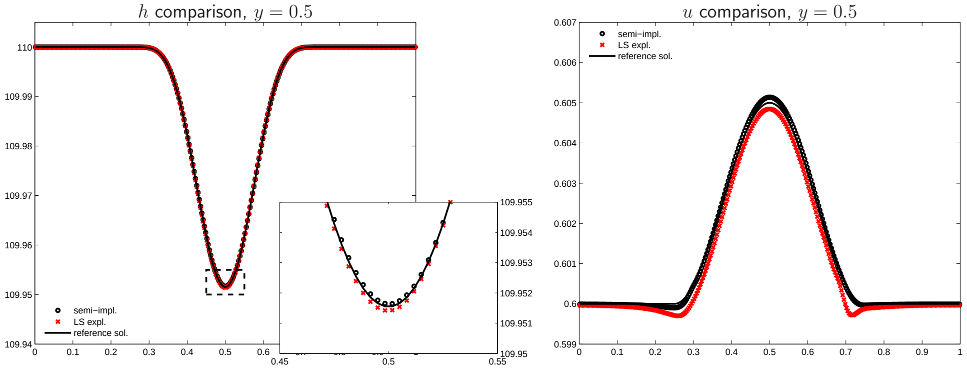

Figure 6: Comparison of the large time step explicit and semi-implicit FVEG methods and the reference solution (initial data) on a mesh with 300 × 300 cells, $t = \frac{5}{3}$.

Large time step explicit FVEG scheme

mesh: $N/2N$	L^1 error in u	EOC	L^1 error in v	EOC	L^1 error in h	EOC
20/40	$3.000\ 10^{-3}$		$3.5020\ 10^{-3}$		$2.3010\ 10^{-3}$	
40/80	$7.8794\ 10^{-4}$	1.9761	$8.6328\ 10^{-4}$	2.0195	$6.3103\ 10^{-4}$	1.8659
80/160	$1.6530\ 10^{-4}$	2.2530	$1.8246\ 10^{-4}$	2.2422	$2.1625\ 10^{-4}$	1.5450

Semi-implicit FVEG scheme

mesh: $N/2N$	L^1 error in u	EOC	L^1 error in v	EOC	L^1 error in h	EOC
20/40	$3.4144\ 10^{-3}$		$3.2199\ 10^{-3}$		$3.2449\ 10^{-3}$	
40/80	$8.9582\ 10^{-4}$	1.9304	$7.4733\ 10^{-4}$	2.1072	$8.5369\ 10^{-4}$	1.9264
80/160	$1.6375\ 10^{-4}$	2.4517	$1.3772\ 10^{-4}$	2.4400	$1.5049\ 10^{-4}$	2.5041

Table 4: Experimental order of convergence for large time step explicit (above) and semi-implicit (below) schemes $CFL_{adv} = 0.3,\ CFL_{real} \approx 4.4$.

Acknowledgment:

The authors would like to thank Sebastian Noelle and Andreas Bollermann (Aachen) as well as Gabriella Puppo (Torino) for fruitful discussions on the topic.

References

[1] Bollermann A., Lukáčová-Medviďová M., Noelle S. Well-balanced finite volume Evolution Galerkin methods for the 2D shallow water equations on adaptive grids, *Proceedings of ALGORITMY*, 2009:81-90.

[2] Algorithm 832: Davis T.A. UMFPACK, an unsymmetric-pattern multifrontal method, *ACM Transactions on Mathematical Software* 2004 **30(2)**:196-199.

[3] Lukáčová-Medviďová M., Morton K. W., Warnecke G. Evolution Galerkin methods for hyperbolic systems in two space dimensions. *MathComp.* 2000 **69**:1355–1384.

[4] Lukáčová-Medviďová M., Noelle S., Kraft M. Well-balanced finite volume evolution Galerkin methods for the shallow water equations. *J. Comp. Phys.* 2007 **221**:122-147.

[5] Meister A. Asymptotic based preconditioning technique for low Mach number flows *ZAMM* 2003 **83(1)**:3-25.

[6] Ricchiuto M., Bollermann A. Stabilized residual distribution for shallow water simulations, *J. Comp. Phys* 2009 **228**:1071-1115.

[7] SACADO:
http://trilinos.sandia.gov/packages/docs/dev/packages/sacado/doc/html/index.html

[8] Sun Y., Ren Y.- X. The finite volume local evolution Galerkin method for solving the hyperbolic conservation laws, *J. Comp. Phys.* 2009 **228(13)**:4945-4960.

[9] Xing Y., Shu C.-W. High order finite difference WENO schemes with the exact conservation property for the shallow water equations, *J. Comput. Phys.* 2005 **208**:206-227.