V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

A FINITE VOLUME MULTILEVEL APPROXIMATION OF THE SHALLOW-WATER EQUATIONS

A. Bousquet*, R. Temam*

*Institute for Scientific Computing and Applied Mathematics,
Indiana University, Rawles Hall, Bloomington, IN 47405-5701

Key words: Multilevel methods, Shallow-water equations, Finite Volumes, Central-upwind

Abstract. *We present a multilevel method which is based on a central-upwind finite volume deiscretization and uses new incremental unknowns which enable to preserve the numerical conservation of the scheme. The method is applied to the nonviscid two-dimensional shallow-water equations, it is tested and analyzed on two and three levels of discretization on different test cases and turns out to furnish a good solution of the problems while saving CPU time.*

1 Introduction

A class of multilevel methods, called Incremental Unknowns methods was introduced to improve calculation speed in the simulation of complex physical phenomena while maintaining an accurate solution of the problems. They were originally developed for the study of turbulent flows ([11], [12], [13], [24], [25]) but can be of interest as well in other types of problems encompassing many different scales.
In this lecture partly based on the article [1] we are concerned with the solution of the two dimensional nonlinear shallow-water equations by a multilevel method using finite volume discretization. This work is intended at exploring the implementation of such methods for this system. Incremental unknowns were introduced in the context of finite differences in [24] for the approximation of inertial manifolds. For references on parabolic or elliptic problems treated with multilevel methods in the context of finite differences or finite elements see [5]-[8], [25], in the context of pseudo-spectral methods or wavelets, see [9]-[13]. The implementation of finite volume multilevel schemes for the solution of the Burgers equations with a diffusive term was made in [15]. The general principle is to split the unknowns in two (or more) terms: a "large-scale" component Y and one (or several) "small-scale" component(s) Z and to treat differently Y and Z. The decomposition of the unknowns that we employ in this article is purely algebraic but it enables to preserve the numerical conservation of the scheme. The decomposition of the variables is here done globally but it can be done locally in certain parts of the domain only, based on physical

motivations (or on information on the flow).

The shallow-water equations describe the propagation of surface waves of long wavelength and of relatively large amplitude, which give rise to strongly nonlinear flows. Multilevel methods for the shallow-water equations supplemented with a hyper-dissipative operator were studied in [13] in the context of spectral methods for the simulation of turbulence. Our study covers a general framework but we are particularly interested in the modeling of oceanic or atmospheric flows in the presence of mild turbulence. Therefore, unlike some other situations of physical interest (like e.g. the breakdown of a dam), the height is not meant to vanish, as in e.g. [2], [3]. From this perspective we present simulations based on initial conditions taken from [13]. Our multilevel method allows to resolve accurately the problems studied while reducing the CPU time and preserving the numerical conservation of the scheme.

For the spatial finite volume discretization, the hyperbolic nature of the system requires that we consider schemes that are well adapted to such problems. Recently, several finite volume schemes have been developed for the simulation of the shallow-water equations ([2], [3], [17], [21]) in order to study some particular properties (e.g. preservation of steady states, positivity of the height of the water). Here we apply a multilevel method built on central-upwind type schemes which were constructed to solve numerically nonlinear conservation laws, [19]-[21]. These Godunov-type schemes are based on exact evolution and averaging over Riemann fans and do not need the use of Riemann solvers and characteristic decomposition, which render them simple and efficient; moreover they can be reduced to a very simple semi-discrete form. More particularly we will work with the schemes presented in [20], [21]: they are based on the one-sided local speeds of propagation and constitute less dissipative generalizations of the semi-discrete central-upwind schemes; they also allow to work on non staggered grids. However any finite volume scheme which can be written in a semi-discrete form can be used as well , provided that it is adapted to the hyperbolic nature of the system. Our method has been tested with central leap frog fluxes and the results were not satisfactory. For the time discretization we need to use a TVD method which preserves the spatial accuracy: in the simulations we employ a Runge Kutta method of order two or four. As stated above and as explained in more details below, our multilevel method is based on a different treatment of the large-scale and the small-scale components of the flow, the small-scale components being small in magnitude. For instances small-scale component of the variables can be determined through a simple time scheme; in our experimentations we chose to freeze them. For the reader not familiar with these schemes and the puzzling terminology "central-upwind", see [1] and the references therein.
We conclude our presentation by showing some numerical simulations for the Equatorial Rossby soliton using our multilevel algorithm (more simulations are described in [1] and will be presented in the oral communication). In fact we present two versions of the

simulations for the Equatorial Rossby soliton; the first one exactly as in the standard description of this test problem - see below - uses a Dirichlet boundary condition at the boundary of the domain. This is acceptable as long as the soliton does not reach the boundary, but when the soliton reaches the boundary it is reflected in a non physically realistic way as shown in our calculations (see Figure 12). So we also implemented our method for the Equatorial Rossby soliton, using a nonlinear boundary condition inspired by some theoretical work in progress, [23], [4]). This boundary condition appears to be transparent (non reflective at the boundary), and this shows very clearly in our numerical simulations (see Figure 13).

2 Presentation of the problem

We are interested in implementing a finite volume multilevel scheme for the discretization of the nonlinear two-dimensional shallow-water system on a rectangular domain $\mathcal{M} = (0, L_x) \times (0, L_y)$, with periodic or Dirichlet boundary condition; see Figure 1. This

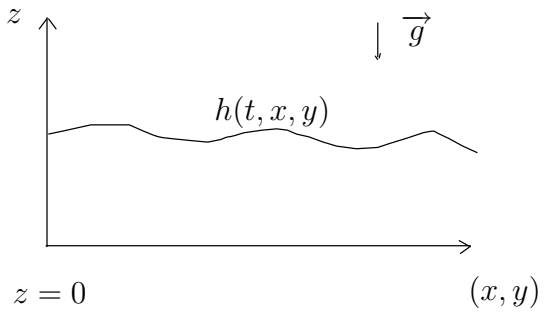

Figure 1: Vertical section of the domain

system is the following:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial uh}{\partial x} + \dfrac{\partial vh}{\partial y} = S_h, \\ \dfrac{\partial uh}{\partial t} + \dfrac{\partial hu^2}{\partial x} + \dfrac{\partial huv}{\partial y} + \dfrac{g}{2}\dfrac{\partial h^2}{\partial x} = S_U, \\ \dfrac{\partial vh}{\partial t} + \dfrac{\partial huv}{\partial x} + \dfrac{\partial hv^2}{\partial y} + \dfrac{g}{2}\dfrac{\partial h^2}{\partial y} = S_V. \end{cases} \tag{1}$$

Here h is the fluid depth above the bottom which is supposed flat, u and v are the x and y components of the velocity, and g denotes the gravity constant; $S = (S_h, S_U, S_V)^t$ represents a source term which usually vanishes but is introduced here at no cost for "mathematical" generality. All quantities are non dimensional. We will write $U = uh$ and $V = vh$ and:

$$Q = \begin{pmatrix} h \\ U \\ V \end{pmatrix}, \; F(Q) = \begin{pmatrix} U \\ \frac{U^2}{h} + \frac{gh^2}{2} \\ \frac{UV}{h} \end{pmatrix}, \; G(Q) = \begin{pmatrix} V \\ \frac{UV}{h} \\ \frac{V^2}{h} + \frac{gh^2}{2} \end{pmatrix},$$

which allows us to write the system in the conservative form

$$\begin{cases} \dfrac{\partial h}{\partial t} + \nabla.(U, V) = S_h, \\ \dfrac{\partial U}{\partial t} + \nabla.(uU, uV) + \dfrac{g}{2}\nabla_x(h^2) = S_U, \\ \dfrac{\partial V}{\partial t} + \nabla.(vU, vV) + \dfrac{g}{2}\nabla_y(h^2) = S_V, \end{cases} \tag{2}$$

or equivalently:

$$\frac{\partial Q}{\partial t} + \frac{\partial F(Q)}{\partial x} + \frac{\partial G(Q)}{\partial y} = S. \tag{3}$$

The discretization of $\mathcal{M}$ is done using rectangular finite volumes $K_m = [x_{m/w}, x_{m/e}] \times [y_{m/s}, y_{m/n}]$ of centers (x_m, y_m), see Figure 2 and of dimensions $\Delta x \times \Delta y$, with $N_x \Delta x = L_x$ and $N_y \Delta y = L_y$:

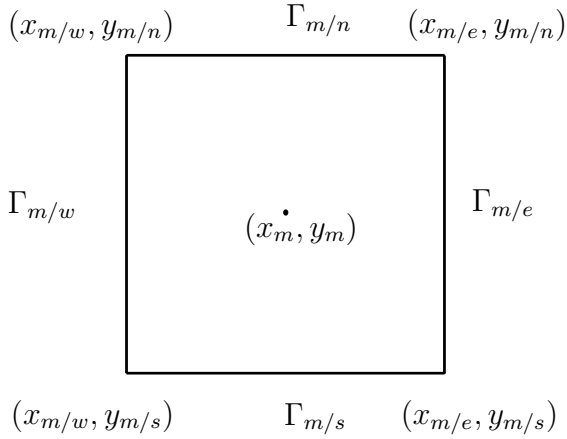

Figure 2: A cell K_m

We use a NSWE (North-South-West-East) stencil which is presented in Figure 3 to identify the unknowns.

The unknowns will be approximations of the cell averages:

$$Q_m(t) = \frac{1}{\Delta x \Delta y} \int_{K_m} Q(t, x, y) dx dy,$$

where $Q_m(t) = (h_m(t), U_m(t), V_m(t))^T$ and similarly for the source terms.

To derive the space discretized equations, we integrate the system (2) on each cell K_m, divide by its area $\Delta x \Delta y$, and we obtain:

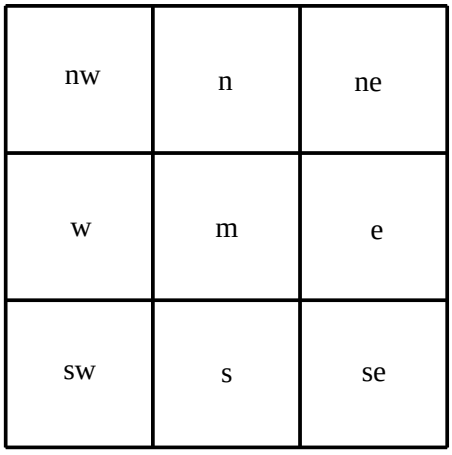

Figure 3: The NSWE stencil

$$\frac{d}{dt}Q_m(t) = -\frac{H^x_{m/e}(t) - H^x_{w/m}(t)}{\Delta x} - \frac{H^y_{m/n}(t) - H^y_{s/m}(t)}{\Delta y} + S_m(t), \tag{4}$$

S_m representing the contribution of the source term.
Here $H^x_{m/e}(t)$ and $H^y_{m/n}(t)$, for example, are respectively the east flux (along the x axis) and the north flux (along the y axis) on the edges between K_m and K_e, and between K_m and K_n, and similarly for the other terms (see Figure 2); for example:

$$H^x_{m/e}(t) = \frac{1}{\Delta x}\int_{\Gamma_{m/e}} F(Q(t,x,y))dy.$$

These approximations of the fluxes depend on the method employed; we will consider central-upwind fluxes, which are made explicit in Section 3.3, see [18]-[21], but *our multilevel method below can also be based on other fluxes.*

3 Presentation of the multilevel method

The domain is discretized by two-levels of rectangular finite volume meshes: the fine mesh $\mathcal{F}_1$ counts $N_x \times N_y$ control volumes of dimensions $\Delta x \times \Delta y$, with $N_x\Delta x = L_x$, $N_y\Delta y = L_y$; and the coarse mesh $\mathcal{F}_2$ has $\dfrac{N_xN_y}{9}$ control volumes of dimensions $3\Delta x \times 3\Delta y$. Although all quantities are not used above and below we have, available on each coarse cell M and each fine cell m, the average values h_M, h_m, of h. Here we use small letters for the fine mesh and capital letters for the coarse mesh: we denote by K_m a control volume of the fine mesh and by K_M a control volume of the coarse mesh (see Figure 4).

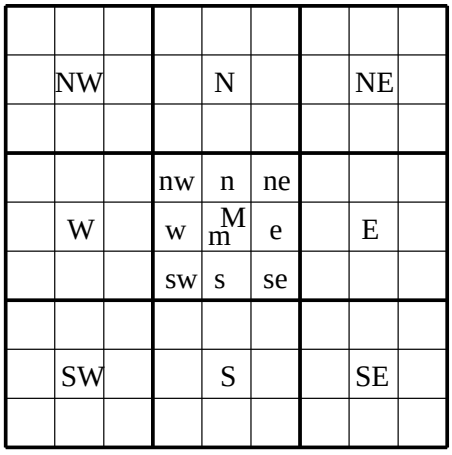

Figure 4: The NSWE stencil with coarse and fine cells

3.1 Incremental unknowns

We define the incremental unknowns for the conservative variables of the shallow-water system, that is the three components of Q. We split each of the unknowns in a large-scale component Y and a small-scale component Z, which is meant to be frozen during a certain number of time steps. By large-scale and small-scale, we mean that Y contains the major information on the solution and that Z represents a correcting term which is comparatively small, as explained in Lemma 3.1 below. For incremental unknowns defined by spectral decompositions like Fourier or wavelets, see [5]-[13], [24]-[25].

Definition 3.1. *Suppose that $Q = (h, U, V)^T$ is known on the fine mesh $\mathcal{F}_1$.*
Then on the control volume K_M, the large-scale component $Y_M = (Y_{h,M}, Y_{U,M}, Y_{V,M})^T$ and the small-scale components Z_e, Z_w, Z_n, Z_s, Z_{ne}, Z_{se}, Z_{nw}, Z_{sw} are defined as follows (see Figure 5):

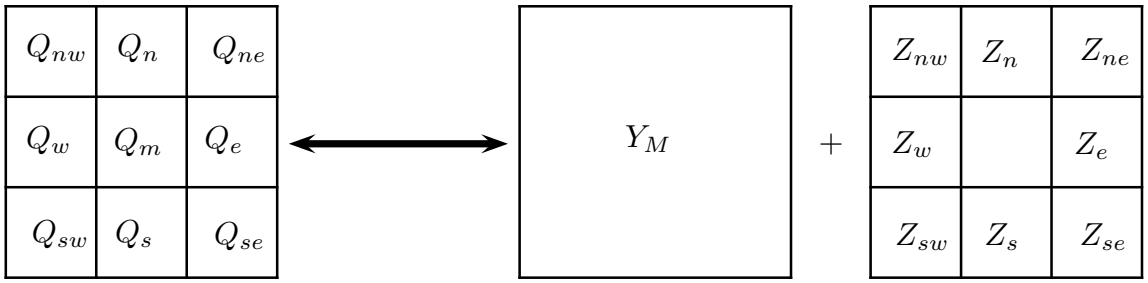

Figure 5: The Finite Volume Incremental Unknowns

$$
\begin{aligned}
&Y_M = \frac{1}{9}(Q_m + Q_e + Q_w + Q_n + Q_s + Q_{ne} + Q_{nw} + Q_{se} + Q_{sw}),\\
&Z_e = Q_e - \frac{1}{3}(Y_E + 2Y_M), \quad Z_w = Q_w - \frac{1}{3}(Y_W + 2Y_M),\\
&Z_n = Q_n - \frac{1}{3}(Y_N + 2Y_M), \quad Z_s = Q_s - \frac{1}{3}(Y_S + 2Y_M),\\
&Z_{ne} = Q_{ne} - \frac{1}{3}(Y_E + Y_M + Y_N), \quad Z_{se} = Q_{se} - \frac{1}{3}(Y_S + Y_M + Y_E),\\
&Z_{nw} = Q_{nw} - \frac{1}{3}(Y_W + Y_M + Y_N), \quad Z_{sw} = Q_{sw} - \frac{1}{3}(Y_S + Y_M + Y_W),
\end{aligned}
\tag{5}
$$

Remark 3.1. The definition of the Z in (5) is at our disposal. We chose them so that the Z are of order $\Delta x^2 + \Delta y^2$

Remark 3.2. It is important to observe that this definition of the incremental unknowns is recursive: once the Y and Z corresponding to the second level coarse grid have been calculated, we can split Y by the same means to find the Y and Z corresponding to a finer level of discretization, that is to say the Y of the second level play then the role of large-scale variable for the third level.

□

3.2 The multilevel scheme

3.2.1 Scheme on the coarse grid

We split each component of $Q = (h, U, V)^T$ into its large-scale component $Y = (Y_h, Y_U, Y_V)^T$ and its small-scale component $(Z_h, Z_U, Z_V)^T$. To obtain the scheme on the coarse grid of level 2, we write (4) on each fine cell $K_m, K_e, K_w, K_n, K_s, K_{ne}, K_{se}, K_{nw}, K_{sw}$ of the coarse cell K_M (see Figure 4), and we take the mean value by summing all these equations and dividing by 9. This results in:

$$
\begin{aligned}
\frac{d}{dt}Y_M(t) = \frac{1}{9\Delta x}\Big[&H^x_{m/w} - H^x_{m/e} + H^x_{n/nw} - H^x_{n/ne} + H^x_{s/sw} - H^x_{s/se}\\
&+ H^x_{We/w} - H^x_{m/w} + H^x_{nw/Wne} - H^x_{nw/n} + H^x_{sw/Wse} - H^x_{sw/s}\\
&+ H^x_{n/ne} - H^x_{Enw/ne} + H^x_{m/e} - H^x_{e/Ew} - H^x_{se/Esw} + H^x_{s/se}\Big]\\
&+ \frac{1}{9\Delta y}\Big[H^y_{m/n} - H^y_{n/Ns} + H^y_{s/m} - H^y_{n/m} + H^y_{s/Sn} - H^y_{s/m}\\
&+ H^y_{ne/e} - H^y_{ne/Nse} + H^y_{e/se} - H^y_{ne/e} + H^y_{se/Snw} - H^y_{se/e}\\
&+ H^y_{nw/w} - H^y_{nw/Nsw} + H^y_{sw/w} - H^y_{w/nw} + H^y_{sw/Sne} - H^y_{sw/w}\Big] + S_M(t),
\end{aligned}
\tag{6}
$$

with :

$$S_M(t) = \frac{1}{9}(S_{nw}(t) + S_n(t) + S_{ne}(t) + S_w(t) + S_n(t) + S_e(t) + S_{sw}(t) + S_s(t) + S_{se}(t)). \tag{7}$$

Also in (6) the definition of the fluxes such as $H^x_{m/w}$, $H^x_{m/nw}$ is obvious and the less obvious fluxes such as $H^x_{We/w}$ etc, relate to the edges shown in Figure 6. Equation (6) gives after simplifications the following semi-discrete scheme to be applied on the coarse grid of level 2:

$$\frac{d}{dt}Y_M(t) = \frac{1}{9\Delta x}\Big[(H^x_{nw/Wne} + H^x_{We/w} + H^x_{sw/Wse}) - (H^x_{Enw/ne} + H^x_{e/Ew} + H^x_{se/Esw})\Big] + \frac{1}{9\Delta y}\Big[(H^y_{sw/Sne} + H^y_{s/Sn} + H^y_{se/Snw}) - (H^y_{nw/Nsw} + H^y_{n/Ns} + H^y_{ne/Nse})\Big] + S_M(t). \tag{8}$$

Remark 3.3. Note that formula (8) corresponds just to the averaging of equations (4) on the 9 cells corresponding to the coarse cell M. Of course this process is completely recursive and can be repeated for simulations on three or more levels of grids.

We can now conceive different numerical schemes depending on the definition (choice) of the fluxes and the solution on the coarse level can be done locally in certain parts of the domain. We ought also to choose the time discretization for dY_M/dt. Concerning the spatial discretization we chose to freeze the Z components during the iterations on the coarse grid, while the large-scale components Y are computed through this scheme.

3.2.2 Multilevel algorithm

The small-scale components Z have an important effect on the size of the error. To explain this, let us describe the multilevel algorithm in details.
For a general multilevel situation, let us fix the number N_{max} of levels of grids on which we are going to compute. From $t = 0$ until the final time T, we repeat N_{it}/L times (where N_{it} is the total number of iterations) the cycle $n_1 n_2 ... n_L$ where for $1 \leq i \leq L$, n_i stands for an iteration on the level n_i, $1 \leq n_i \leq N_{max}$. For example for a simulation on two-levels, as considered here we chose to repeat cycles of the form: 111122221111, where 1 corresponds to the fine grid and 2 to the coarse one. Therefore at the n^{th} iteration, we compute :

- At level 1 we work on the fine mesh $\mathcal{F}_1$ and compute Q^{n+1} with the classical scheme (described in our case in (10) and (11) below).

- At level 2
  - we calculate explicitly the fluxes needed by the scheme (8),
  - we split Q^n into its large-scale Y^n and small-scale Z^n components,
  - we compute Y^{n+1} with (8),
  - we recompose Q^{n+1} from Y^{n+1} and Z^n.

We freeze the small-scale components Z during each iteration at level 2. This induces an error on Z of the order $\Delta t \times$ magnitude of $Z = \Delta t(\Delta x^2 + \Delta y^2)$. This error committed in freezing Z is, and this adds up to the classical error; this time variation thus needs to be controlled during the simulations.

3.2.3 Gain of CPU time

It is important to notice that implementing the scheme on the coarse grid requires to calculate the fluxes on the fine grid only on the exterior edges of the coarse cell, as indicated in Figure 6.

When implementing such schemes in the context of shallow-water equations, the most

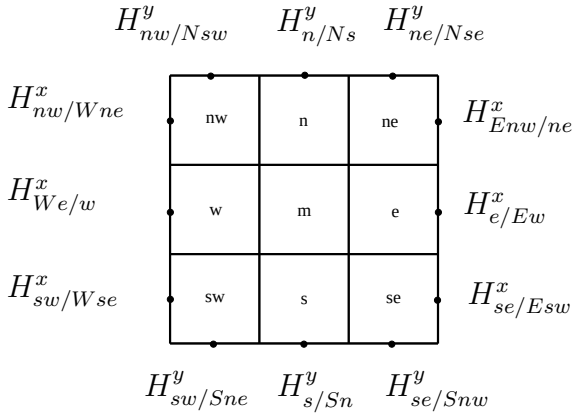

Figure 6: Fluxes needed for an iteration of the scheme on the coarse grid to calculate y^{n+1} on K_M

time consuming step during one iteration is the calculation of the fluxes; therefore the multilevel method is expected to reduce significantly the CPU time.

Indeed, if we implement the classical finite volume scheme using only on one-level namely the fine mesh $\mathcal{F}_1$ which has $N_x \times N_y$ control volumes, for each time iteration, we need to calculate the fluxes on $2N_xN_y + N_x + N_y$ edges.

Now with the multilevel method implemented on two-levels of grids $\mathcal{M}_1$ and $\mathcal{M}_2$, during an iteration on the coarse grid, the fluxes need to be evaluated on $2N_xN_y/3 + N_x + N_y$ edges. This means that for this iteration, we gain $\mathcal{G}$ computations of fluxes, where

$$\mathcal{G} = \frac{4N_xN_y}{3}.$$

In the particular case of a square mesh (that we will consider in the numerical experiments),

$$\mathcal{G} = 4N_x^2/3,$$

and this corresponds to a computational saving of

$$\left(100 \times \frac{2}{3}\frac{N_x}{N_x+1}\right)\%.$$

For example if we work on two-levels with two grids of 300×300 and 100×100, this means a gain of 66.4% for each iteration on the coarse grid. We notice that the maximum gain in percentage that we can expect is less than 66.66%, and that when the number of cells of the fine mesh ($N_x \times N_y$) increases, the gain gets closer to this maximal value.

The behavior of the multilevel method then depends on the number of iterations performed on the coarse level: as this number increases, the CPU-time decreases, whereas the error increases. Nevertheless, in order to be sure that we obtain a good approximation of the solution, we need to check that the error when using the multilevel method ranges between the error made when calculating on the fine level and that made when calculating on the coarse level. This will ensure that the multilevel method enables us to get a better solution than when calculating on the coarse level, while being faster than the classical one-level method on the fine grid. We have to make a compromise between these two aspects. In our case we obtained a gain of 15.6% of CPU time for a given accuracy.

3.3 The multilevel method with central-upwind schemes

The space discretization is done using a semi-discrete central-upwind scheme (as in [20], [21]); we describe here in detail the expression of these central-upwind fluxes. These types of schemes have the advantage of being perfectly adapted to the discretization of hyperbolic systems of conservation laws due to their upwind nature while being robust and simple since they do not require to solve any Riemann problem; moreover they are non staggered schemes. The starting point of the construction of this type of schemes is the equivalent integral formulation of the system. They are based on integration over Riemann fans using the one-sided local speeds of propagation.

Recall that the semi-discrete form of the scheme reads:

$$\frac{d}{dt}Q_m(t) = -\frac{H^x_{m/e}(t) - H^x_{w/m}(t)}{\Delta x} - \frac{H^y_{m/n}(t) - H^y_{s/m}(t)}{\Delta y} + S_m(t). \tag{9}$$

Equations (9) is exact. The approximation procedure starts with the approximation definitions of the fluxes that we choose. We first use a second order version following [20], [21], and the corresponding numerical fluxes are:

$$H^x_{m/e} \cong \frac{a^+_{m/e}F(Q^E_m) - a^-_{m/e}F(Q^W_e)}{a^+_{m/e} - a^-_{m/e}} + \frac{a^+_{m/e}a^-_{m/e}(Q^W_e - Q^E_m)}{a^+_{m/e} - a^-_{m/e}} \tag{10}$$

and

$$H^y_{m/n} \cong \frac{b^+_{m/n}G(Q^N_m) - b^-_{m/n}G(Q^S_n)}{b^+_{m/n} - b^-_{m/n}} + \frac{b^+_{m/n}b^-_{m/n}(Q^S_n - Q^N_m)}{b^+_{m/n} - b^-_{m/n}}. \tag{11}$$

With F,G as in (3), we use a non-oscillatory linear polynomial reconstruction to evaluate the following point values which are present in (10), (11):

$$Q^E_m = p_m(t, x_{m/e}, y_m),\ Q^W_m = p_m(t, x_{m/w}, y_m),$$
$$Q^N_m = p_m(t, x_m, y_{m/n}),\ Q^S_m = p_m(t, x_m, y_{m/s}).$$

where $p_m(t, x_m, y_m) = Q_m(t) + s^x_m(t)(x - x_m) + s^y_m(t)(y - y_m)$.
We use a piecewise linear reconstruction in order to obtain a second order scheme. The order of the scheme also relates to the order of the quadrature formula used to approximate the flux integrals coming from the integral formulation.
The slopes of this linear approximation are calculated using a minmod limiter:

$$\begin{aligned} s^x_m(t) &= \text{minmod}(\theta\frac{Q_m(t) - Q_w(t)}{\Delta x}; \frac{Q_e(t) - Q_w(t)}{2\Delta x}; \theta\frac{Q_e(t) - Q_m(t)}{\Delta x}), \\ s^y_m(t) &= \text{minmod}(\theta\frac{Q_m(t) - Q_s(t)}{\Delta y}; \frac{Q_n(t) - Q_s(t)}{2\Delta y}; \theta\frac{Q_n(t) - Q_m(t)}{\Delta y}), \end{aligned} \tag{12}$$

with

$$\text{minmod}(x_1, x_2, ..) := \begin{cases} \min(x_i), \text{ if } x_i > 0\ \forall i \\ \max(x_i), \text{ if } x_i < 0\ \forall i \\ 0, \text{ otherwise.} \end{cases}$$

where $\theta \in [1, 2]$.
An appropriate choice of these approximate derivatives is crucial to ensure that the above reconstruction is non oscillatory in the sense of preventing appearance of new extrema in the solution; it can be shown that with such approximate derivatives the scheme satisfies the scalar total-variation-diminishing (TVD) property (see [18], [20]). The parameter $\theta \in [1, 2]$ has to be chosen in an empirical optimal way in order to obtain good results.
The one-sided local speeds of propagation are given by:

$$\begin{aligned} a^+_{m/e} &= \max[\lambda_{max}(\frac{\partial F}{\partial Q}(Q^W_e)), \lambda_{max}(\frac{\partial F}{\partial Q}(Q^E_m)), 0], \\ a^-_{m/e} &= \min[\lambda_{min}(\frac{\partial F}{\partial Q}(Q^W_e)), \lambda_{min}(\frac{\partial F}{\partial Q}(Q^E_m)), 0], \\ b^+_{m/n} &= \max[\lambda_{max}(\frac{\partial G}{\partial Q}(Q^S_n)), \lambda_{max}(\frac{\partial G}{\partial Q}(Q^N_m)), 0], \\ b^-_{m/n} &= \min[\lambda_{min}(\frac{\partial G}{\partial Q}(Q^S_n)), \lambda_{min}(\frac{\partial G}{\partial Q}(Q^N_m)), 0], \end{aligned} \tag{13}$$

where $\lambda_{max}(\frac{\partial F}{\partial Q}(\tilde{Q}))$ and $\lambda_{min}(\frac{\partial F}{\partial Q}(\tilde{Q}))$ (resp. $\lambda_{max}(\frac{\partial G}{\partial Q}(\tilde{Q}))$ and $\lambda_{min}(\frac{\partial G}{\partial Q}(\tilde{W}))$) are respectively the largest and the smallest eigenvalue of the Jacobian matrix of F, $\frac{\partial F}{\partial Q}$ (resp. of G, $\frac{\partial G}{\partial Q}$) at the point $\tilde{Q}$.

For the time discretization, we use a second order Runge-Kutta (or Heun) and fourth order Runge-Kutta method. Let $T > 0$ be fixed, denote the time step by $\Delta t = T/N_{it}$, where N_{it} is an integer representing the total number of time iterations; for $n = 0, .., N_{it}$ we define Q^n as the approximate value of Q at time $t_n = n\Delta t$.

4 Numerical simulations

4.1 The Equatorial Rossby Soliton

We tested our multilevel method on the classical Equatorial Rossby soliton, (see http://marine.rutgers.edu/po/tests/rossby/index.html). The equations are slightly different as we have to add the Coriolis force and there is no more source term. Furthermore we now consider the Dirichlet boundary conditions. The equations and all quantities being non-dimensional we take $\{\mathcal{M} = (-24, 24) \times (-8, 8)\} \times \{0 < t < T\}$, as shown in Figure 4.1, with Dirichlet boundary conditions. Based on the data given by the website (metric section), the Rossby number is $R_o \approx 0.067$ and the Froude number is $F_r \approx 3.16$.

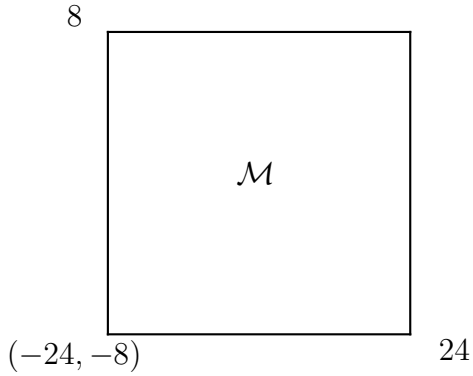

Figure 7: Domain for the equatorial Rossby Soliton.

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial uh}{\partial x} + \dfrac{\partial vh}{\partial y} = 0, \\ \dfrac{\partial uh}{\partial t} + \dfrac{\partial hu^2}{\partial x} + \dfrac{\partial huv}{\partial y} + \dfrac{g}{2}\dfrac{\partial h^2}{\partial x} - fv = 0, \\ \dfrac{\partial vh}{\partial t} + \dfrac{\partial huv}{\partial x} + \dfrac{\partial hv^2}{\partial y} + \dfrac{g}{2}\dfrac{\partial h^2}{\partial y} + fu = 0. \end{cases} \tag{14}$$

Here :

- h is the fluid depth above the bottom which is supposed flat
- u and v are the x and y components of the velocity
- g denotes the gravity, in this case $g = 1$
- f is the Coriolis force, which is equals to $f_0 + \beta y$, in this case $f_0 = 0$ and $\beta = 1$

The boundary conditions are of Dirichlet type:

$$\begin{cases} h_{\partial\mathcal{M}} &= 1 \\ u_{\partial\mathcal{M}} &= 0 \\ u_{\partial\mathcal{M}} &= 0 \end{cases} \tag{15}$$

Initial solution :

$$\begin{cases} u(x,y,0) = \phi(x)\dfrac{(-9+6y^2)}{4}e^{-\frac{y^2}{2}}, \\ v(x,y,0) = \dfrac{\partial\phi(x)}{\partial x}(2y)e^{-\frac{y^2}{2}}, \\ h(x,y,0) = \phi(x)\dfrac{(3+6y^2)}{4}e^{-\frac{y^2}{2}} + 1, \end{cases} \tag{16}$$

with :

$$\begin{aligned} &B = 0.395 \\ &A = 0.7771B^2 \\ &\phi(x) = Asech^2Bx \\ &\tfrac{\partial\phi(x)}{\partial x} = -2Btanh(Bx)\phi. \end{aligned}$$

Figure 8 shows the solution that we obtained by solving the equations using the Runge Kutta 4 method, with a time step of 10^{-3}.

We are going to compare the results using the Runge Kutta 4 methods, with a time step of 10^{-3}, a fine mesh of 900x300 squares, and a coarse mesh of 300x100 squares. The multilevel method will use a cycle of 1111122222211111, where 1 is a computation on the fine mesh and 2 on the coarse mesh, we will denote it MM37%.

Figure 9 shows that the relative error for the mass N_h (defined on top of Figure 9) is constant at 10^{-13}. This is confirmed by both calculations, where the mass $= \rho\int_{\mathcal{M}} hdxdy$, and ρ is the volumic mass of the fluid. :

$$\rho\frac{\partial}{\partial t}\int_{\mathcal{M}} hdxdy = 0$$

Figure 10 shows the potential enstrosphy on 1-level on the fin grid and on 2-levels (MM37%).

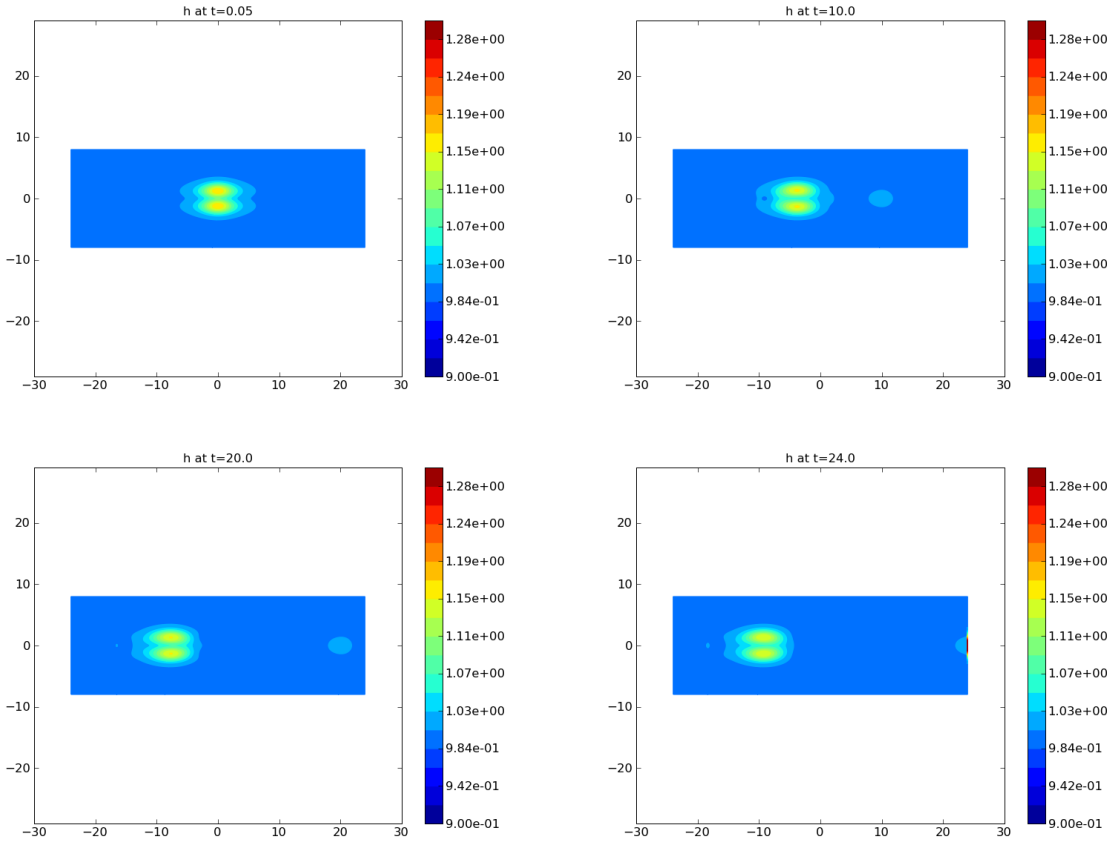

Figure 8: Rossby Soliton, the value of h(x,y) at t=0.05, 10, 20, 24.

Then we compare the maximal difference between the h,u and v of the two computation (one on the coarse and one using the multilevel method). Figure 11 shows that h,u,v are the same at 10^{-3} until the wave touch the border on x=0, where there is a condition of Dirichlet.

For the time saving, we compare the one-level method on the fine mesh and the one using two-levels for 16 iterations in time which are the length of a cycle for the multilevel one. It takes 55 seconds for the one-level method and 40 seconds for the two-levels method. We save 15 seconds for only 16 iterations in time. This means we gain 21.4% of CPU time.

For the boundary conditions we first used the Dirichlet boundary condition as proposed on the reference web site. The behavior of the soliton is satisfactory as long as the soliton does not reach the boundary (see Figure 8). However when the soliton reaches the boundary it is reflected in a non physically realistic way, see Figure 12. On the contrary in Figure 13 we used some other boundary conditions being proposed in a theoretical

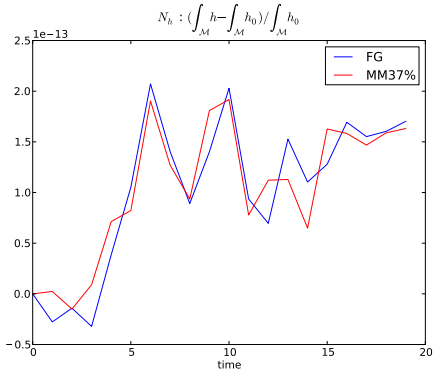

Figure 9: Computation of the relative error of the mass over time for each computations (on using 1-level on the fine grid, the other with MM37% using 2-level), which equals $\left(\int_M h dxdy - \int_M h_{t=0} dxdy\right) / \int_M h_{t=0} dxdy$, for the Equatorial Rossby Soliton

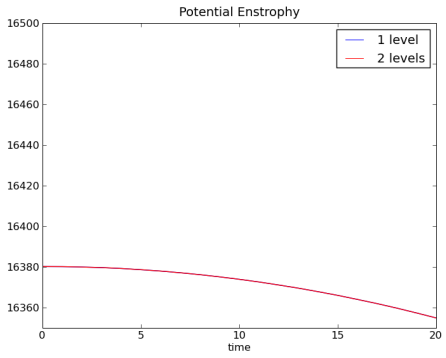

Figure 10: Computation of the potential enstrophy for the Equatorial Rossby Soliton

work in progress [23], [4]. With these boundary conditions, the boundary appears to be transparent (non reflective) in our simulations based on the same multilevel method. More details on our transparent boundary conditions and more simulations using these boundary conditions will appear in a work in a progress. Another work in progress will further develop the multilevel algorithms.

4.2 Concluding remarks

In this article we have implemented and studied a multilevel method to approximate the solution of a hyperbolic system of conservation laws: the shallow-water equations. We have introduced new incremental unknowns which enabled us to preserve the conservation property of the schemes. The numerical simulations show that the method remains accurate while enabling to decrease the time of computation although a certain number of iterations are made on the coarse grid. The method shows its best performance for simulations requiring a small space step and very fine meshes. It could be applied with improved efficiency by using an adaptative criterion which would depend on the problem

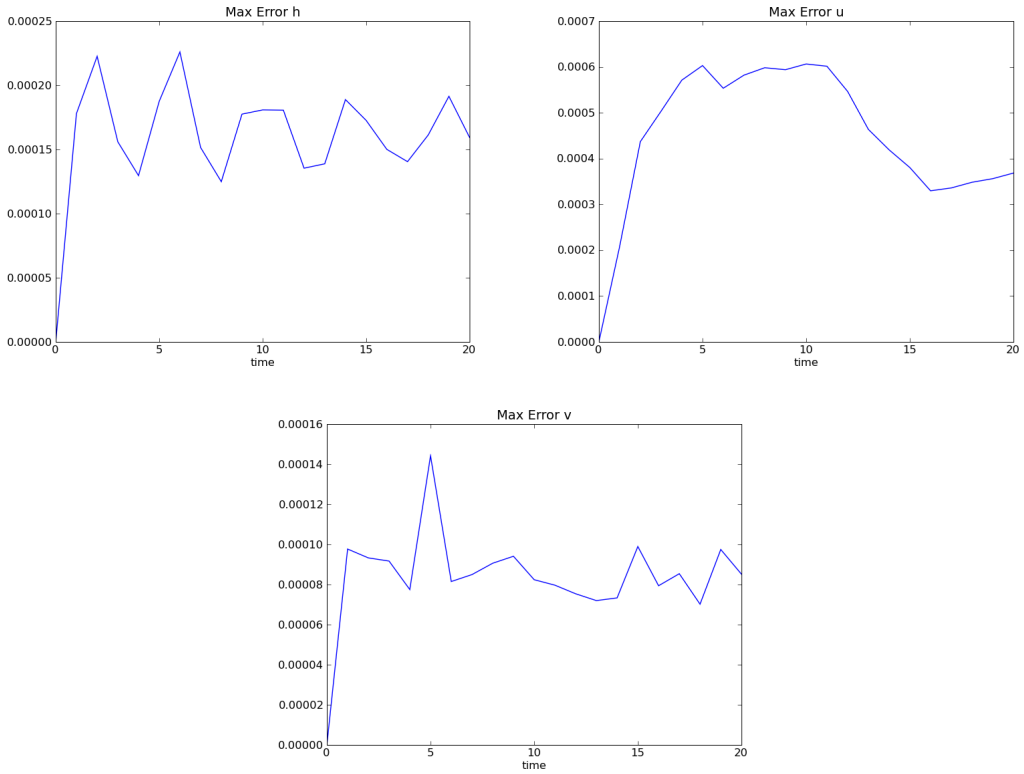

Figure 11: Computation of the L^∞ error between Q on 1-level and Q on 2-levels, for the Equatorial Rossby Soliton

to be solved An improvement of this method could also be done by applying it locally and selectively in the spatial domain. These ideas are left to future work.

5 Acknowledgements

This work was supported in part by NSF Grant DMS0906440, and by the Research Fund of Indiana University.

REFERENCES

[1] K. Adamy, A. Bousquet, S. Faure, J. Laminie, R. Temam, 2010, A multilevel method for finite volume discretization of the two-dimensional nonlinear shallow-water equations *Ocean Modelling 33(3-4), p. 235-256* .

[2] E. Audusse, F. Bouchut, M-O. Bristeau, R. Klein, B. Perthame, 2004, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow-water flows *SIAM J. Sci. Comput. 25, No. 6, p. 2050-2065.*

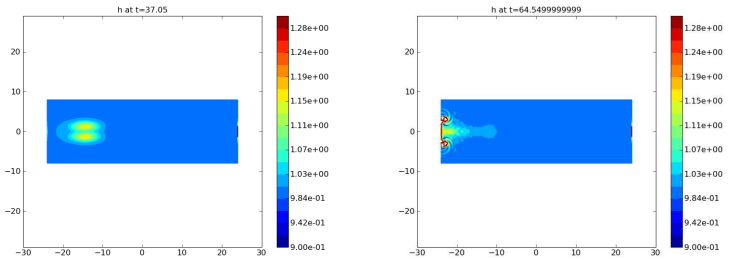

Figure 12: Behavior (reflection) of the Rossby soliton when it reaches the boundary, when using the Dirichlet boundary condition.

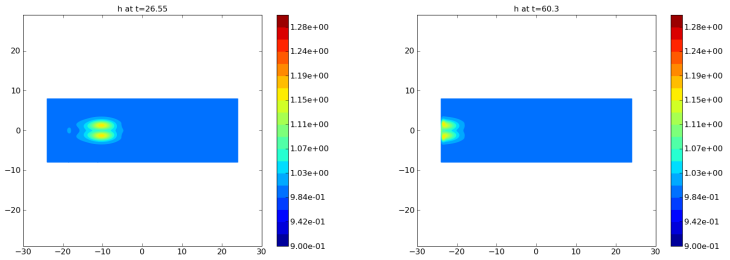

Figure 13: Behavior of the Rossby soliton when it reaches the boundary, when using the transparent boundary condition. The soliton freely moves out of the domain without any backward deformation.

[3] E. Audusse, M-O. Bristeau, R. Klein, B. Perthame, 2000, Kinetic schemes for Saint-Venant equations sith source terms on unstructured grids *INRIA Report RR-3989.*

[4] Benzoni-Gavage, Sylvie, 2007, Serre, Denis, Multidimensional hyperbolic partial differential equations. First-order systems and applications *Oxford Mathematical Monographs. The Clarendon Press, Oxford University Press, Oxford, xxvi, pp. 508.*

[5] C. Calgaro, J. Laminie, R. Temam, 1997, Dynamical multilevel schemes for the solution of evolution equations by hierarchical finite element discretization *Appl. Numer. Math. 23(4), p. 403-442.*

[6] J. P. Chehab, 1998, Incremental unknowns method and compact schemes *RAIRO Model. Math. Anal. Num. 32(1), p. 51-83.*

[7] M. Chen, R. Temam, 1991, Incremental unknowns for solving partial differential equations *Numer. Math. 59(3), p. 255-271.*

[8] M. Chen, R. Temam, 1993, Incremental unknowns in finite differences: condition number of the matrix *SIAM J. Matrix Anal. Appl. 14(2), p. 432-455.*

[9] A. Debussche, J. Laminie, E. Zahrouni, 2005, A dynamical multi-level scheme for the Burgers equation: wavelet and hierarchical finite element *J. Sci. Comput. 25(3), p. 445-497.*

[10] L. Debreu, E. Blayo, 2008, Two-way embedding algorithms: a review *Ocean Dynamics 58, p. 415-428.*

[11] T. Dubois, F. Jauberteau, R. Temam, 1999, Dynamic multilevel methods and the numerical simulation of turbulence *Cambridge University Press, Cambridge.*

[12] T. Dubois, F. Jauberteau, R. Temam, 1998, Incremental unknowns, multilevel methods and the numerical simulation of turbulence *Comput. Methods Appl. Mech. Engrg. 159(1-2), p. 123-189.*

[13] T. Dubois, F. Jauberteau, R. Temam, J. Tribbia, 2005, Multilevel schemes for the shallow-water equations *J. Comput. Phys. 207(2), p. 660-694.*

[14] R. Eymard, T. Gallouet, R. Herbin, 2000, Finite volume methods *in Handbook of numerical analysis VII, p.713-1020, North Holland, Amsterdam.*

[15] S. Faure, J. Laminie, R. Temam, 2005, Finite volume discretization and multilevel methods in flow problems, *J. Sci. Comput., 25(1-2), p. 231-261.*

[16] S. Faure, 2003, Méthodes de volumes finis et multiniveaux pour les équations de Navier-Stokes, de Burgers et de la chaleur, *Thèse de Doctorat.*

[17] T. Gallouët, J.M. Hérard, N. Seguin, 2003, Some approximate Godunov schemes to compute shallow-water equations with topography *Comput. Fluids 32(4), p. 479-513.*

[18] A. Kurganov, E. Tadmor, 2000, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, *J. Comput. Phys. 160(2), p. 720-742.*

[19] A. Kurganov, G. Petrova, 2001, A third-order semi-discrete genuinely multidimensionnal central scheme for hyperbolic conservation laws and related problems *Numer. Math. 88, 683-729.*

[20] A. Kurganov, S. Noelle, G. Petrova, 2001, Semidiscrete central upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations, *SIAM J. Sci. Comput. 23(3), 707-740.*

[21] A. Kurganov, D. Levy, 2002, Central-upwind schemes for the Saint-Venant system, *M2AN, 36(3), 397-425.*

[22] M. Pectu, R. Temam, 2010, The shallow water equations with Dirichlet boundary conditions, Discrete and Continuous Dynamical Systems B, *to appear.*

[23] M. Pectu, R. Temam, 2010, article in preparation.

[24] R. Temam, 1990, Inertial manifolds and multigrid methods *SIAM J. Math. Anal., 21(1), p. 154-178.*

[25] R. Temam, 1996, Multilevel methods for the simulation of turbulence. A simple model, *J. Comput. Phys., 127(2), p. 309-315.*