PROCESSING FLOW DATA BY ORTHOGONAL PROJECTION FOR TRANSPORT COMPUTATION

ANTONY SIAHAAN
1189778

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and
Computer Science

October 2005

PROCESSING FLOW DATA BY ORTHOGONAL PROJECTION FOR TRANSPORT COMPUTATION

Antony Siahaan

Delft, October 2005

Committee :

Prof. Dr. Ir. A.W. Heemink
Dr. P. Wilders
Dr. Ir. M. Zijlema

Table of Contents

Abstract

Satisfying the constancy condition is required in advective solvers. A violation to this condition in a numerical tranport scheme can create instabilities. The constancy condition is satisfied when flow coefficients are conservative and there is consistency between the discretization of the flow equations and that of the transport equation.

The inputs obtained from a solver or other sources don't necessarily satisfy the flow equation which is consistent with the underlying transport numerical scheme. Orthogonal projection methods are developed to correct these inputs so as to be conservative. These methods are first developed in the context of the finite volume scheme of the divergence free velocity equation. Then they are adopted in the context of shallow water flow. A major difference now is that several choices of variables to be corrected are possible.

Simulations for some test problems of shallow water flow show that the constancy condition can be achieved by means of these projection methods, even under for large Courant numbers. Despite that, there are some important aspects that need concern :

1. The positivity of water depth cannot always be ensured
2. Choice of variables to be corrected seems to depend on their original value

Acknowledgements

I first thank my supervisor Dr. Peter Wilders for his guidance during the work of this thesis. He has also given me constant support by being patience in serving my very basic questions. Many thanks also go to Prof. A.W. Heemink and Dr. M. Zijlema as the members of the thesis committee.

I would also like to express my gratitude to Prof. Roger Cooke and Dr. Dorota Kurowicka for their encouragement and patience during my two years hard time in catching up the study in the Risk and Environmental Modeling program.

1. Introduction

Advective transport, in the absence of sources or sinks, will satisfy the property of constancy condition, i.e. a constant solution at $t = 0$ will reproduce the similar solution in next instances. A violation to this condition in the numerical scheme can create instabilities [1]. The constancy condition is satisfied when flow coefficients are conservative and there is a consistency between the discretization of continuity equation and that of transport equation [2]. In the finite volume scheme of advection transport within incompressible flow, the normal velocity needs to be numerically divergence free to ensure this constancy condition.

A given velocity field, however, doesn't always meet the divergence free condition. Such inputs might not be conservative. Our goal is to determine a small correction such that the new input meets the divergence free condition. In the implementation of finite volume in advective transport, computation can be done using the normal velocity or flux at faces instead of the full velocity field. To achieve zero divergence it is necessary to preserve zero net flux within cells. An iterative procedure to modify the flux in order to achieve the divergence freedom has been implemented in [3] and [4]. It attempts to make a slight modification such that the new normal velocity can be maintained as close as possible to the original normal velocity. It turns out that this procedure is in fact an example of an orthogonal projection method. In this thesis some orthogonal projection methods are developed in a general sense and applied to obtain a nearby divergence free velocity field. Similar to above, iterative procedures are necessary to solve the projection equations.

Shallow water flow is a special case of incompressible flow and also here the constancy condition needs to be satisfied in its transport computation. Beside velocity, the corresponding continuity equation now contains the water depth and it changes dynamically in time. So the velocity is no longer the only variable that can be corrected. The correction can be done to either velocity only, water depth only or both of them. Accordingly, for each choice a corresponding orthogonal projection method can be developed.

The outline of this report is as follows. Chapter 2 gives the construction of finite volume scheme of the advection transport and describes the requirement of a numerical divergence free condition as a result of the necessity of constancy condition. In Chapter 3, the concept of orthogonal projection methods is developed to cope with the numerical divergence free condition. Iterative methods for solving the correction equations are presented and some examples are discussed. Chapter 4 is devoted to a more specific case of incompressible flow, i.e. shallow water flow.

Orthogonal projection methods are developed for this case. Also in this chapter, two test problems are introduced to be attacked by the orthogonal projection methods. The corresponding numerical results are given in Chapter 5. The final section is Chapter 6 which gives some conclusions regarding the results of this investigation and some recommendation for further research.

2. Finite Volume Scheme of Transport Equation & Constancy Condition

This chapter discusses the construction of a finite volume scheme for advective transport and describes the requirement of a numerical divergence free condition in its transport computation in order to satisfy constancy condition. Afterward an iterative procedure of flux correction to recover the divergence free condition is given.

2.1 Geometry of Face

A finite volume mesh divides a domain into a partition of cells and the boundary of a cell is set by its faces. The convention used in the geometry of mesh is essential in the formalization of a finite volume scheme.

In 2D domain, a face is determined by its end points which are in an order arranged in the mesh description. The unit normal n_e on the face e is taken such that the normal points to the right if the face is looked from the end point 1 to the end point 2.

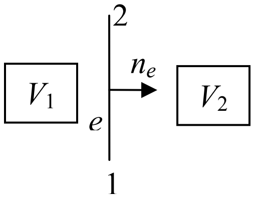


Figure 2.1: The convention of the vector normal of a face

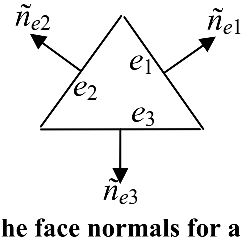


Figure 2.2: The face normals for a cell

The face normals of a cell are always taken outward as seen in Figure 2.2. Consequently, two cells sharing the same face will have different signs for their face normals. As for the face e in Figure 2.1, there holds $\tilde{n}_e^{V_1} = n_e$ and $\tilde{n}_e^{V_2} = -n_e$ for which $\tilde{n}_e^{V_1}$ and $\tilde{n}_e^{V_2}$ are the face normal of face e for cell V_1 and V_2 respectively.

For any cell V sharing the face $e_,$ we define

$$\alpha(V,e)=\begin{cases}0, & \text{if } \tilde{n}_e^V = n_e \\ 1, & \text{if } \tilde{n}_e^V = -n_e\end{cases}$$

2.2 Advection Equation

The equation governing the advection transport in incompressible flow, with the absence of sources and sinks, is given in the following general form

$$\frac{\partial c}{\partial t}+\nabla.(\tilde{u}c)=0 \tag{2.1}$$

where c is the concentration of substance contained in the fluid (per unit volume) and $\tilde{u}$ denotes the velocity field of the flow.

2.3 Consistency of Discretization

Integrating the equation (2.1) over a volume grid cell V and applying Green's divergence theorem leads to the next relation

$$\frac{\partial}{\partial t}\int_V cdV = -\int_S c(\tilde{u}.\tilde{n}^V)dS \tag{2.2}$$

where S is the boundary enclosing V and $\tilde{n}^V$ is the outward unit normal vector to S.
In a grid cell, the right hand side is the sum of integrals over all faces of the cell,

$$\int_S c(\tilde{u}.\tilde{n}^V)dS = \sum_{e\in V}\int_{S_e} cu_n^V dS \tag{2.3}$$

where $u_n^V = \tilde{u}.\tilde{n}^V$ is the normal velocity to faces in the cell V. The integral is then approximated by $c_e u_e^V l_e$ for which c_e, u_e^V, *and* l_e are respectively the concentration at face e, the normal velocity at face *e* outward the cell *V* and the area of face *e*. The normal velocity u_e^V is taken as $u_n^V = \tilde{u}.\tilde{n}_e^V$ at the face midpoint.

The average concentration of the cell *V* is $c_V = \frac{\int_V cdV}{\Delta V}$, ΔV being the volume of cell *V*. The semi-discrete of (2.2) now turns into

$$\Delta V \frac{dc_V}{dt} = -\sum_{e \in V} c_e u_e^V l_e \tag{2.4}$$

The discrete formula (2.4) needs to be completed by presenting a formula expressing c_e as a function of surrounding cell center values c_V. In the present context, it is not necessary to present further details at this point.

The definition of consistency of discretization is given by [1] and the references therein : *a discretization of the advection equation is consistent with continuity if, given a spatially uniform scalar field as an initial datum, and a general flow field, the discretized scalar advection equation reduces to the discretized continuity equation.*

By that definition, the simplified form of discretized continuity equation with which (2.4) is consistent can be found by applying a uniform scalar field. Substituting a constant solution $c = 1$ in (2.4) yields

$$\sum_{e \in V} u_e^V l_e = 0 \tag{2.5}$$

This equation is also the representation of the finite volume discretization of divergence-free velocity, $\nabla.\vec{u} = 0$. Suppose that the discretized conservative flow equation leads to the numerical divergence free of (2.5), the discretized advection (2.4) is consistent with continuity.

2.3 Constancy Condition

The constancy condition is satisfied when the flow coefficients are conservative and the consistency between flow and transport equation is guaranteed [2]. The equation (2.4) has been shown to be consistent with (2.5). Furthermore as the term $u_e^V l_e$ is equivalent to the flux through the face *e*, the flow coefficients can also be regarded as flux instead of normal velocity. Accordingly, (2.4) can meet the constancy condition when the fluxes or normal velocities at faces satisfy the zero divergence equation (2.5).

In many cases, the given velocity field doesn't necessarily achieve zero divergence when applied to a developed numerical scheme, thus it is of importance to do a slight modification to the flow coefficients so as to satisfy (2.5). In the next section, an iterative procedure is presented to perform the flux modification. This method will be the main concern in the rest of this report.

2.4 Endlich's Iterative Procedure of Flux Correction

In this section a procedure is described to make the flux satisfying the zero divergence equation. The idea is based on Endlich method [3,4], but now with flux modification instead of velocity modification. This choice is made to comply with the requirement of the finite volume scheme.

Instead of using u_e^V, (2.5) will be expressed in term of the normal velocity on the face *e* which is approximated by $u_e = \tilde{u}.n_e$ at the midpoint of face *e*. The equation (2.5) can now be written as

$$\sum_{e \in V} (-1)^{\alpha(V,e)} u_e l_e = 0$$

Defining $f_e = u_e l_e$ as the flux through the face *e*, the last equation becomes

$$\sum_{e \in V} (-1)^{\alpha(V,e)} f_e = 0 \tag{2.6}$$

The left hand side of (2.6) which will be later denoted as FT^V, is the net outflow flux from the cell *V*. The iterative procedure will be applied to modify f_e until the net flux $FT^V = 0$ is accomplished up to a certain tolerance.

It is often required to retain the value of flux on the boundary faces. For this reason, it is necessary to have a boundary indicator for every face *e,*

$$\beta_e = \begin{cases} 0, & \text{if } e \text{ is a boundary face} \\ 1, & \text{otherwise} \end{cases}$$

The procedure can be performed with one of the following two different iterative methods, namely the sequential or parallel method. Let *ncells* be the number of cells in the domain and nfb^V be the number of boundary faces on the cell *V*. The sequential method will lead the iterative procedure with the following two steps for $V = 1,\ldots,ncells$:

1. Compute FT^V
2. Compute for every face $e \in V$ the new values of flux

$$f_e = f_e - (-1)^{\alpha(V,e)} \frac{FT^V}{nfaces^V - nfb^V} \beta_e$$

This method computes the net outflow flux using the newest values of the flux whereas the parallel method does it using the flux in the previous iteration. With the parallel method, the iterative procedure consists of the following steps :

1. Compute FT^V for $V = 1,\ldots,ncells$
2. Compute the new values of flux for $V = 1,\ldots,ncells$ and for every face $e \in V$

$$f_e = f_e - (-1)^{\alpha(V,e)} \frac{FT^V}{nfaces^V - nfb^V} \beta_e$$

For any method, applying the two steps for a single cell only will result in (2.6). In general, the flux is modified more than once while looping over all cells, so the procedure seeks to reduce the value of *FT*. The procedure is stopped when the maximum value of *FT* over all cells is smaller than a tolerance *ε*.

3. Projection Formulation of Divergence Free Velocity Equation

The concept of orthogonal projection methods is developed in this chapter to cope with the numerical divergence free condition. Iterative methods for solving the correction equations in projection methods are presented and some examples are discussed.

3.1 *Finite Volume representation of velocity correction*

The Finite Volume scheme of divergence free velocity (2.6), which is expressed in terms of fluxes at faces, can be arranged in the matrix equation

$$LF = 0 \tag{3.1}$$

where $L : \Re^{nface} \to \Re^{nvol}$ is the discrete divergence operator and F is the vector of fluxes at faces. In general there holds *nface* > *nvol* and (3.1) presents an undetermined system.

It can also be written using normal velocity at faces since flux is equal to the normal velocity times the area of the face. In 2D grid, the area is given by the length of the face. If we store the information of the face length in a diagonal matrix *D*, (3.1) can be written as

$$LDq = 0 \tag{3.2}$$

where q is the vector of normal velocities at faces.

Due to some possibilities such as different discretization schemes, a given normal velocity $\tilde{q}$ at faces might not satisfy (3.2). The main objective now is therefore, to recover the discrete equation (3.2) by presenting a new normal velocity q which is obtained from the correction of the given normal velocity. Taking q' as the correction applied to $\tilde{q}$, the following relation follows :

$$q = \tilde{q} + q' \tag{3.3}$$

The velocity correction must take into account the boundary conditions, which requires the values of some fluxes or normal velocities to be fixed. Let q_1 corresponds to the faces on which the flux correction is applied and q_2 be the fixed normal velocity ($q_2 = \tilde{q}_2$), (3.2) can be decomposed as

$$L_1 D_1 q_1 + L_2 D_2 q_2 = 0 \tag{3.4}$$

with $L = [L_1 \quad L_2]$, $D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$, and $q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$

Combining (3.3) & (3.4) and knowing that $q_2 = \widetilde{q}_2$ leads to

$$L_1 D_1 (\widetilde{q}_1 + q_1') + L_2 D_2 \widetilde{q}_2 = 0$$
$$L_1 D_1 q_1' + (L_1 D_1 \widetilde{q}_1 + L_2 D_2 \widetilde{q}_2) = 0$$
$$L_1 D_1 q_1' = -(L_1 D_1 \widetilde{q}_1 + L_2 D_2 \widetilde{q}_2)$$

$$L_1 D_1 q_1' = r_{\widetilde{q}} \quad ; \quad r_{\widetilde{q}} = -LD\widetilde{q} \tag{3.5}$$

3.2. *Scalar Field Formulation*

Let E denotes the diagonal matrix containing the distance between two centers of the cells sharing the same face. We define a vector of quantities S as

$$S = Eq$$

Setting $E = \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix}$ where E_1 and E_2 correspond to the faces of corrected and fixed fluxes respectively, we then have

$$S_1 = E_1 q_1$$

The correction of quantity $S_1' = E_1 q_1'$ is assumed to depend upon a scalar field φ via

$$S_1' = M^T \varphi$$

Here, $\varphi \in \Re^{nvol}$ is a scalar field related to volumes and $M^T : \Re^{nvol} \to \Re^{nface}$ is an operator transferring volume information to face information. It follows that

$$q_1' = E_1^{-1} M^T \varphi \tag{3.6}$$

Of course, a question arising is how to choose the matrix *M*. We will serve this question later on.

Remark :

If we choose *M* = *L*, then (3.6) reads

$$q_1' = G\varphi \quad , \quad G = E^{-1}L^T$$

It can be seen that *G* presents a discretization of the gradient. Against this background assuming $S_1' = M^T \varphi$ is related to the Helmholtz decomposition of vector fields.

3.3 *System of Equations*

The equations (3.3), (3.5) and (3.6) can be brought together into a system of matrix equation

$$\begin{bmatrix} 0 & L_1 D_1 & 0 \\ M^T & -E_1 & 0 \\ 0 & -I & I \end{bmatrix} \begin{bmatrix} \varphi \\ q_1' \\ q_1 \end{bmatrix} = \begin{bmatrix} r_{\dddot{q}} \\ 0 \\ \tilde{q}_1 \end{bmatrix} \tag{3.7}$$

where

$$r_{\tilde{q}} = -LD\tilde{q} \; .$$

An equivalent system is

$$\begin{bmatrix} A & 0 & 0 \\ M^T & -E_1 & 0 \\ 0 & -I & I \end{bmatrix} \begin{bmatrix} \varphi \\ q_1' \\ q_1 \end{bmatrix} = \begin{bmatrix} r_{\dddot{q}} \\ 0 \\ \tilde{q}_1 \end{bmatrix} \tag{3.8}$$

with

$$A = L_1 D_1 E_1^{-1} M^T$$

We can now rewrite the system of equations into a single equation :

$$q_1 = \tilde{q}_1 + E_1^{-1} M^T A^{-1} r_{\tilde{q}} \tag{3.9}$$

or in terms of flux as

$$F_1 = \tilde{F}_1 + D_1 E_1^{-1} M^T A^{-1} r_{\tilde{q}} \tag{3.10}$$

3.4. Orthogonal Projection

The specific form of (3.9) or (3.10) depends on the choice of matrix *M*. There are four special choices of matrix *M*, each having its own objective.

3.4.1 Minimal flux correction

Choosing $M = L_1 D_1^{-1} E_1$ turns (3.10) into

$$F_1 = \tilde{F}_1 + L_1^T A^{-1} r_{\tilde{q}} \qquad ; A = L_1 L_1^T \tag{3.11}$$

$$F_1' = L_1^T A^{-1} r_{\tilde{q}} = L_1^T (L_1 L_1^T)^{-1} r_{\tilde{q}} \tag{3.12}$$

Recall equation (3.5) in flux form :

$$L_1 F_1' = r_{\tilde{q}} \tag{3.13}$$

The orthogonal projections $F_{1,R}'$ and $F_{1,N}'$ of F_1' into the row space $\Re(L_1^{\ T})$ and the null space $N(L_1)$ respectively are represented by :

$$F'_{1,R} = L_1^T (L_1 L_1^T)^{-1} L_1 F_1' == L_1^T (L_1 L_1^T)^{-1} r_{\tilde{q}}$$
$$F'_{1,N} = \left(\mathrm{I} - L_1^T (L_1 L_1^T)^{-1} L_1\right) F_1'$$

It is obvious that with this choice of M, the correction F_1' is represented by its projection on the row space only. Since we have $F'_{1,N}$ = 0, the length of flux correction $\left\| F_1' \right\|$ is minimized.

3.4.2 Minimal velocity correction

From (3.9), the choice of $M = L_1 D_1 E_1$ results in

$$\begin{aligned} q_1 &= \tilde{q}_1 + D_1 L_1^T A^{-1} r_{\tilde{q}} \\ &= \tilde{q}_1 + (L_1 D_1)^T A^{-1} r_{\tilde{q}}; \quad A = (L_1 D_1)(L_1 D_1)^T \end{aligned} \tag{3.14}$$

$$q_1' = (L_1 D_1)^T \left((L_1 D_1)(L_1 D_1)^T\right)^{-1} r_{\tilde{q}} \tag{3.15}$$

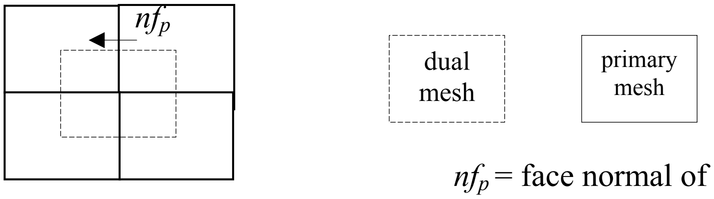
q_1' in (3.15) is the orthogonal projection of normal velocity correction on the row space $\Re\left((L_1 D_1)^T\right)$ and according to (3.5), it represents the minimal length of normal velocity correction $\left\| q_1' \right\|$.

Writing (3.14) in terms of flux :

$$F_1 = \tilde{F}_1 + D_1 \left(L_1 D_1\right)^T A^{-1} r_{\tilde{q}}; \quad A = (L_1 D_1)(L_1 D_1)^T \tag{3.16}$$

3.4.3 Minimal shear correction

We have defined the correction of a quantity in a mesh as $S_1' = E_1 q_1'$. If we assume that its faces are perpendicular to faces of its dual mesh and its cell centers coincide with vertices of the dual mesh, then S_1' contains the correction of shears on the dual mesh. See figure 3.1.



nf_p = face normal of primary face

Figure 3.1: Primary & dual mesh

The equation (3.5) can also be written as

$$L_1 D_1 E_1^{-1}\left(E_1 q_1'\right) = r_{\tilde{q}} \quad ; \quad r_{\tilde{q}} = -LD\tilde{q}$$
$$L_s S_1' = r_{\tilde{q}} \tag{3.17}$$

with $L_s = L_1 D_1 E_1^{-1}$

Now let us choose $M = L_s = L_1 D_1 E_1^{-1}$. Left-multiplying (3.9) by E_1:

$$S_1 = \tilde{S}_1 + L_s^{T} A^{-1} r_{\tilde{q}} \quad ; \quad A = L_1 D_1 E_1^{-1}\left(L_1 D_1 E_1^{-1}\right)^T = L_s L_s^{T}$$
$$S_1' = L_s^{T}\left(L_s L_s^{T}\right)^{-1} r_{\tilde{q}} \tag{3.18}$$

The expression (3.18) represents the projection of solution of (3.17) on the row space $\Re\left(L_s^{T}\right)$, hence resulting in the minimal length of correction $\left\|S_1'\right\|$.

The normal velocity and flux with this projection are given by

$$\begin{aligned} q_1 &= \tilde{q}_1 + E_1^{-1} L_s^T A^{-1} r_{\tilde{q}} \\ F_1 &= \tilde{F}_1 + D_1 E_1^{-1} L_s^T A^{-1} r_{\tilde{q}} \quad ; \quad A = L_s L_s^T ; L_s = L_1 D_1 E_1^{-1} \end{aligned} \tag{3.19}$$

3.4.4 Correction of zero curl

Let us define $\left(L_1^T \varphi\right)_e$ as the row equation in $L_1^T \varphi$ which corresponds with the face *e*. Now consider Fig. 3.2. For the face *e*, we have

$$\left(L_1^T \varphi\right)_e = \varphi_1 - \varphi_2 \tag{3.20}$$

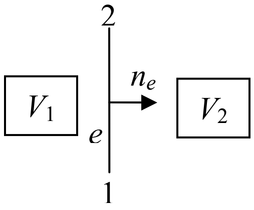
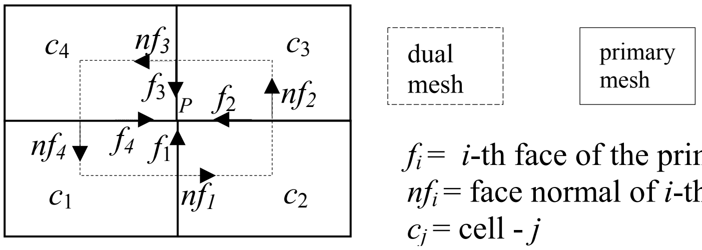


Figure 3.2: Face *e*

Next see Fig. 3.3. Consider all faces that meet in the vertex *P* of the primary mesh :



f_i = *i*-th face of the primary mesh
nf_i = face normal of *i*-th face of the primary mesh
c_j = cell - *j*

Figure 3.3: Mesh around point *P*

There holds

$$\sum_{i=1}^{4} \left(L_1^T \varphi\right)_{f_i} = \varphi_1 - \varphi_2 + \varphi_2 - \varphi_3 + \varphi_3 - \varphi_4 + \varphi_4 - \varphi_1 = 0 \tag{3.21}$$

We assume there exists a mesh like in Figure 3.3 where vertices of the dual mesh coincide with cell centers of the primary mesh and faces of the dual mesh are perpendicular to those of primary mesh. We denote the cell of the dual mesh with V' and the face of the dual mesh with f_i'. We see that the tangent to the face f_i' are identical to the normal nf_i of the primary mesh (or only different in the sign). Let $S_1' = E_1 q_1'$ as before. The curl discretization (see Appendix) of velocity correction within the cell V' which surrounds *P* then reads :

$$(CS'_1)_{V'} = S'_{1f'_1} + S'_{1f'_2} + S'_{1f'_3} + S'_{1f'_4} \tag{3.22}$$

Taking $S_1 = L_1^{\ T} \varphi$, it follows from (3.22) and (3.21) that

$$CL_1^{\ T} \varphi = 0$$

This is valid for arbitrary φ, thus $CL_1^{\ T} = 0$.

It is clear now that choosing $M = L_1$ leads to zero curl in the dual mesh with correction equations

$$\begin{aligned} q_1 &= \widetilde{q}_1 + E_1^{-1} L_1^T A^{-1} r_{\widetilde{q}} \quad ; \quad A = L_1 D_1 E_1^{-1} L_1^{\ T} \\ F_1 &= \widetilde{F}_1 + D_1 E_1^{-1} L_1^T A^{-1} r_{\widetilde{q}} \end{aligned} \tag{3.23}$$

The projection solutions presented by (3.11), (3.16), (3.19), and (3.23) require the solution of $A\varphi = r_{\ddot{q}}$. The matrix A is symmetric for the first three projection methods. In the zero curl correction, A is symmetric only if a regular grid is applied to the domain of problem.

When we deal with a regular grid or a problem in 1D, all those projection methods reduce to the minimal flux correction. In the later discussion, we focus on this method only. If we don't mention a specific type of projection method, then the term 'the projection method' or 'the projection technique' will refer to the method of minimal flux correction.

If our grid has a huge amount of cells, a direct method for the inversion might not be feasible. The next sections of this chapter are related to some iterative methods implemented for solving this φ equation.

3.5. *The Endlich's Iterative Procedure Revisited*

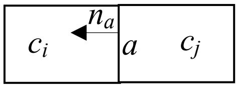
The solution of $A\varphi = r_{\widetilde{F}}$ can be iteratively computed with Jacobi method as follows :

$$\varphi_i(k+1) = \frac{\left[r_{\tilde{F}}^{\;i} - \sum_{l \neq i} a_{il}\varphi_l(k) \right]}{a_{ii}} \tag{3.24}$$

where a_{ij} (i = 1,2,…., *nvol* and j = 1,2,…, *nvol*) is an element of matrix A.

We rewrite (3.12) :

$$F_1' = L_1^T A^{-1} r_{\tilde{F}} = L_1^T \varphi; \qquad r_{\tilde{F}} = -L\tilde{F} \tag{3.25}$$



c_i = cell-*i*
c_j = cell-*j*
n_a = unit normal of face-*a*

Figure 3.4: Face *a*

Equation (3.25) constitutes a set of equations related to all faces of the grid, and every single equation corresponds to a specific face. Accordingly, by considering Figure 3.4, we can determine the equation for the face a as follows :

$$f_a'(k+1) = \left(L_1^T \varphi(k+1)\right)_a = -\varphi_i(k+1) + \varphi_j(k+1)$$

Substituting every corresponding φ with (3.24):

$$f_a'(k+1) = -\frac{\left[r_{\tilde{F}}^{\;i} - \sum_{l \neq i} a_{il}\varphi_l(k) \right]}{a_{ii}} + \frac{\left[r_{\tilde{F}}^{\;j} - \sum_{l \neq j} a_{jl}\varphi_l(k) \right]}{a_{jj}} \tag{3.26}$$

The corrected flux at iteration k and k+1 are obtained after the corrections   :

$$f_a\,(k+1) = \tilde{f}_a + f_a'(k+1)$$
$$f_a\,(k) = \tilde{f}_a + f_a'(k)$$

from which we can get the relation

$$f_a\,(k+1) = f_a(k) - f_a^{'}(k) + f_a^{'}(k+1) = f_a(k) + (f_a^{'}(k+1) - f_a^{'}(k)) \qquad (3.27)$$

The last term in the right hand side can be elaborated using (3.26) :

$$f_a^{'}(k+1) - f_a^{'}(k) = \quad -\varphi_i(k+1) + \varphi_j(k+1) + \varphi_i(k) - \varphi_j(k)$$

$$= -\frac{\left[r_{\widetilde{F}}^{\ i} - \sum_{l\neq i} a_{il}\varphi_l(k)\right]}{a_{ii}} + \frac{\left[r_{\widetilde{F}}^{\ j} - \sum_{l\neq j} a_{jl}\varphi_l(k)\right]}{a_{jj}} + \left\{\varphi_i(k) - \varphi_j(k)\right\}$$

$$= -\frac{r_{\widetilde{F}}^{\ i}}{a_{ii}} + \frac{\left[\sum_l a_{il}\varphi_l(k)\right]}{a_{ii}} - \varphi_i(k) + \frac{r_{\widetilde{F}}^{\ j}}{a_{ii}} - \frac{\left[\sum_l a_{jl}\varphi_l(k)\right]}{a_{jj}} + \varphi_j(k) + \left\{\varphi_i(k) - \varphi_j(k)\right\}$$

$$= \frac{\left[\sum_l a_{il}\varphi_l(k)\right]}{a_{ii}} - \frac{\left[\sum_l a_{jl}\varphi_l(k)\right]}{a_{jj}} - \frac{r_{\widetilde{F}}^{\ i}}{a_{ii}} + \frac{r_{\widetilde{F}}^{\ j}}{a_{ii}}$$

$$= \frac{\left(L_1F_1'(k)\right)_i}{a_{ii}} - \frac{\left(L_1F_1'(k)\right)_j}{a_{jj}} + \frac{\left(L_1\widetilde{F}_1(k) + L_2\widetilde{F}_2\right)_i}{a_{ii}} - \frac{\left(L_1\widetilde{F}_1(k) + L_2\widetilde{F}_2\right)_j}{a_{ii}}$$

$$= \frac{\left(LF(k)\right)_i}{a_{ii}} - \frac{\left(LF'(k)\right)_j}{a_{jj}} = \frac{\left(FT(k)\right)_i}{a_{ii}} - \frac{\left(FT(k)\right)_j}{a_{jj}}$$

where $FT_i(k)$ and $FT_j(k)$ are respectively the net flux in the cell-*i* and cell-*j* at iteration *k*. The corrected flux (3.27) is now expressed as

$$f_a\,(k+1) = f_a(k) + \frac{FT_i(k)}{a_{ii}} - \frac{FT_j(k)}{a_{jj}}$$

or in the following form with $\alpha(i,a) = 0$ and $\alpha(j,a) = 1$,

$$f_a\,(k+1) = f_a(k) + (-1)^{\alpha(i,a)}\,\frac{FT_i}{a_{ii}} + (-1)^{\alpha(j,a)}\,\frac{FT_j}{a_{jj}} \qquad (3.28)$$

Elements a_{ii} and a_{jj} of A = $L_1L_1^T$ denote the number of corrected fluxes corresponding to cell i and j respectively. It is now evident that the Jacobi method of minimal flux correction is equivalent to the Endlich's parallel iterative procedure.

One can also demonstrate that the Gauss Seidel method of minimal flux correction is equivalent to the Endlich's sequential iterative procedure

3.6. Some numerical results by Endlich's Procedure

In this section, we rewrite some results obtained in Internship [5] that concern the projection method. Two test problems are included here, each having non-zero numerical velocity divergence for the given initial data of velocity. The recovery of the divergence free condition was performed using the Endlich's parallel and sequential procedures, which were just mentioned to be equivalent to the orthogonal projection technique using Jacobi and Gauss Seidel method respectively.

The first test problem is presented as a close relevance to the nature and solved with the sequential procedure. It will be shown that this Gauss Seidel-type projection method is not fast enough in its convergence aspect. The second test problem is given to study the limit solution of the Endlich's parallel procedure in connection to its 1D analysis.

3.6.1 Sand Layer

The domain of sand layer is presented in the Figure 3.5. The flow is considered as quasi-1D in this layer. Thus a good approximation of the velocity can be found by assuming the vertical component velocity to be zero and by replacing the horizontal component of the velocity by its vertical average. This is commonly approximated by applying the integrated mass balance over the full vertical in the sand layer which yields velocity field $q = (\overline{u}(x),0)$

Yet that approximation doesn't satisfy the boundary conditions in the sand layer where there is a need of zero normal velocity on the boundary. Another approximation is found by setting

$q = \left(\overline{u}(x), -y \frac{\partial \overline{u}(x)}{\partial x} \right)$. This velocity satisfies both the local mass balance as well as the boundary condition at the upper and lower boundary.

Based on the last approximation, the velocity field (u,v) in the sand layer reads

$$
\begin{aligned}
&u = 20, && v = 0 &&; \quad x < -5 \,\&\, x \geq 5 \\
&u = \frac{200}{6 - 0.8 \cdot x}, && v = \frac{-0.8 \cdot 200 \cdot y}{(6 - 0.8 \cdot x)^2}; && \quad x < 0 \,\&\, x \geq -5 \\
&u = \frac{200}{6 + 0.8 \cdot x}, && v = \frac{0.8 \cdot 200 \cdot y}{(6 + 0.8 \cdot x)^2}; && \quad x < 5 \,\&\, x \geq 0
\end{aligned}
$$

The profile of this velocity is displayed in the Figure 3.6. In this test example, we retain the value on the boundaries and consequently no correction on the fluxes is allowed on the boundaries. The value taken on the left and the right boundary is therefore (u,v) = (20,0). On the top and bottom boundary, the zero normal velocity is zero. It can be verified that, analytically, the divergence of the velocity is zero. But some numerical schemes, when using the value of these analytical functions, will present a nonzero divergence around discontinuity lines x = -5, x = 0 and x = 5.

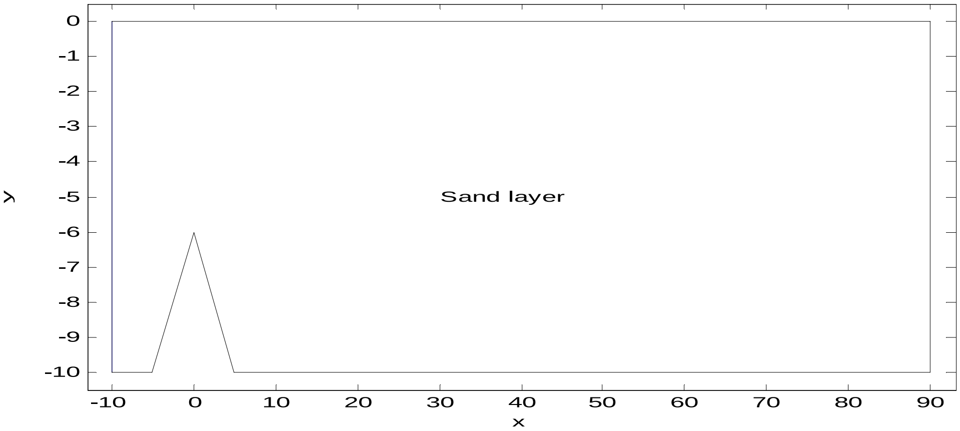


Figure 3.5: Domain of sand layer

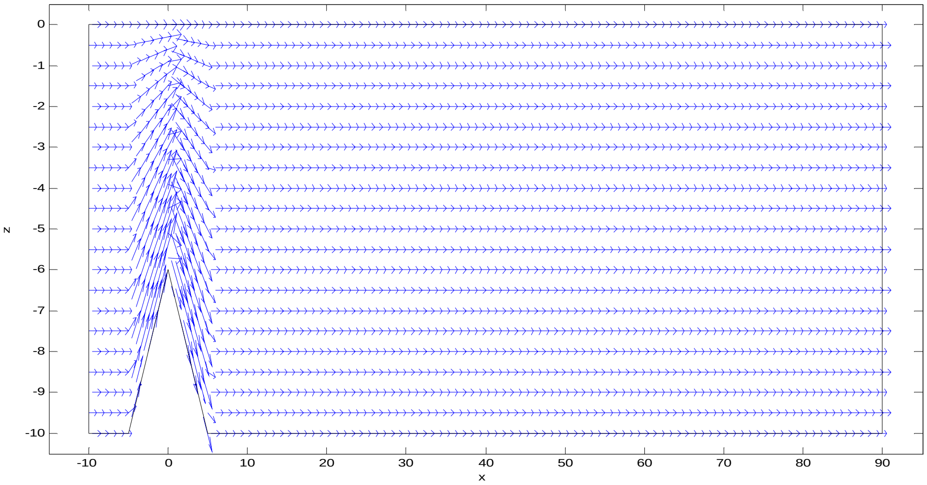


Figure 3.6: Velocity field in the sand layer

3.6.2 Results

For this sand layer problem, a triangular mesh of 2244 cells and 3488 faces is applied to the problem domain. When the original velocity field is used in the finite volume scheme, the numerical divergence at cells of discontinuity is far from zero. Figure 3.7 depicts this.

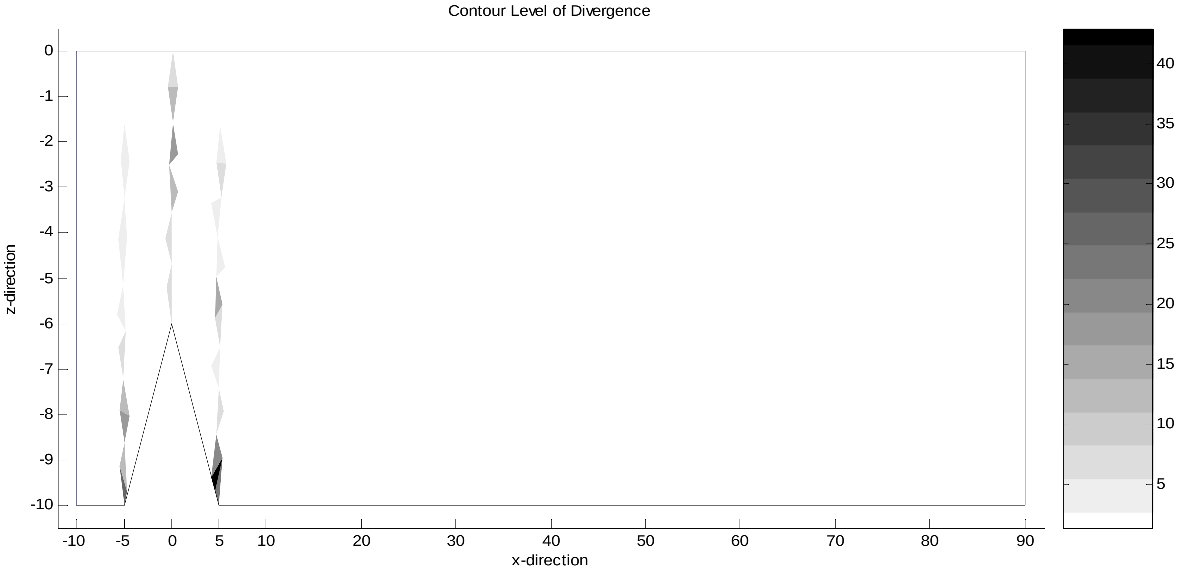


Figure 3.7: Numerical divergence of original velocity in the sand layer

The cells coinciding with the discontinuity points (x = -5, x = 0, x = 5) give high divergence which is caused by the flux difference between faces having different velocity.

As mentioned before, for this problem the computational procedure of flux modification is implemented using the sequential method. Figure 3.8 illustrates the convergence of the computational procedure and presents the logarithmic plot of L_∞-norm of divergence.

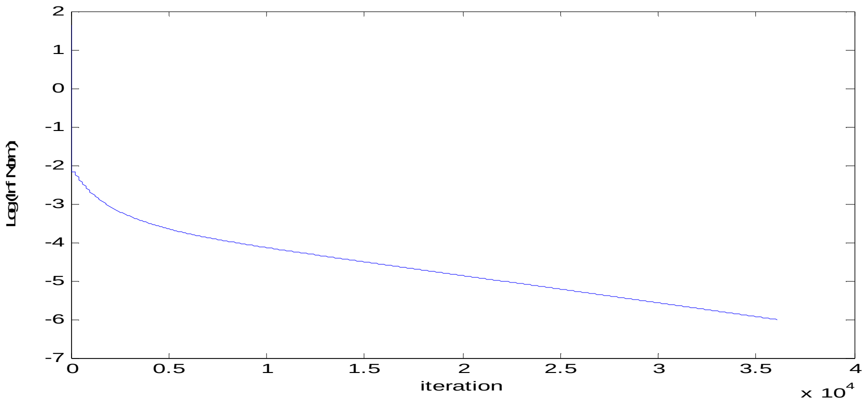


Figure 3.8: Log L_∞-norm of divergence

This iterative procedure can recover the numerical divergence free up to the tolerance of 10^{-6} within 36127 iterations, but this performance is slow with respect to the convergence. Starting from iteration k = 71, the convergence ratio is never again below 0.998. Figure 3.9 gives us an obvious outlook of this.

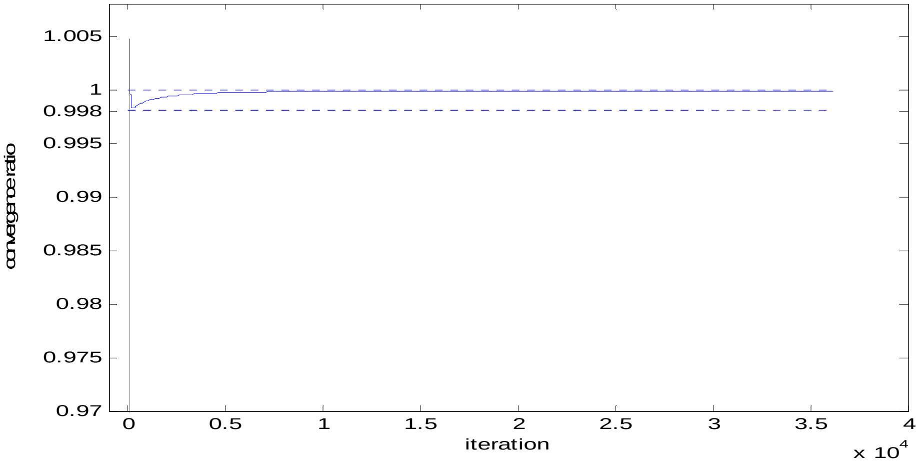


Figure 3.9: convergence speed

Large corrections to the flux are made around the discontinuity points. This is shown in the contour of the correction of the normal velocity at faces, given in Figure 3.10.

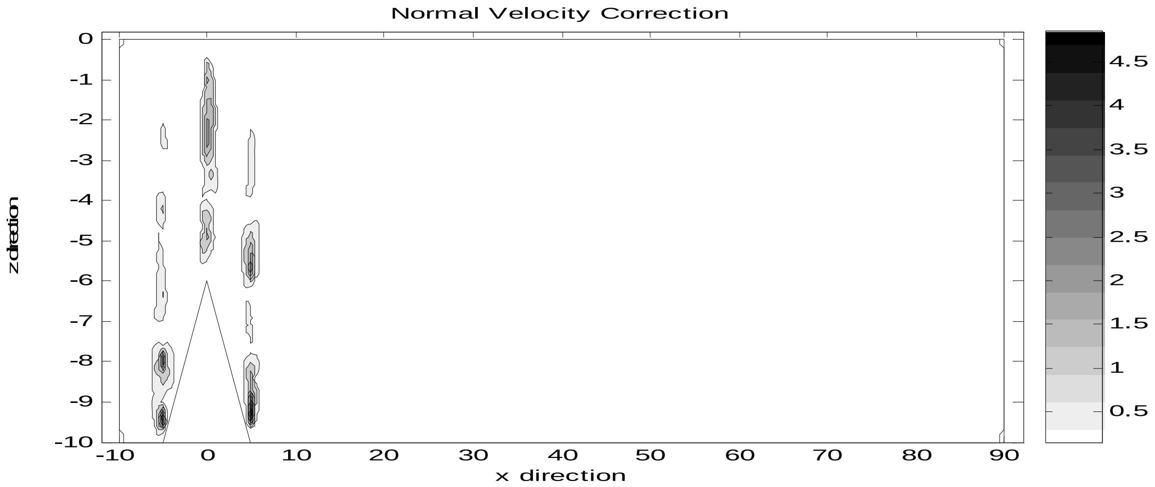


Figure 3.10: Contour of normal velocity correction

The procedure deals with normal velocity components only. Having the corrected normal components, a full velocity field can be recovered using the technique in [6], by which the velocity field can be uniquely determined once the total flux is zero. The corrected full velocity field resulting from this technique is presented in Figure 3.11. Figure 3.12 gives the velocity field correction, which is the difference between the original and corrected velocity field, in a substantial part of domain. Considerably large corrections to the velocity field are made around the discontinuity points.

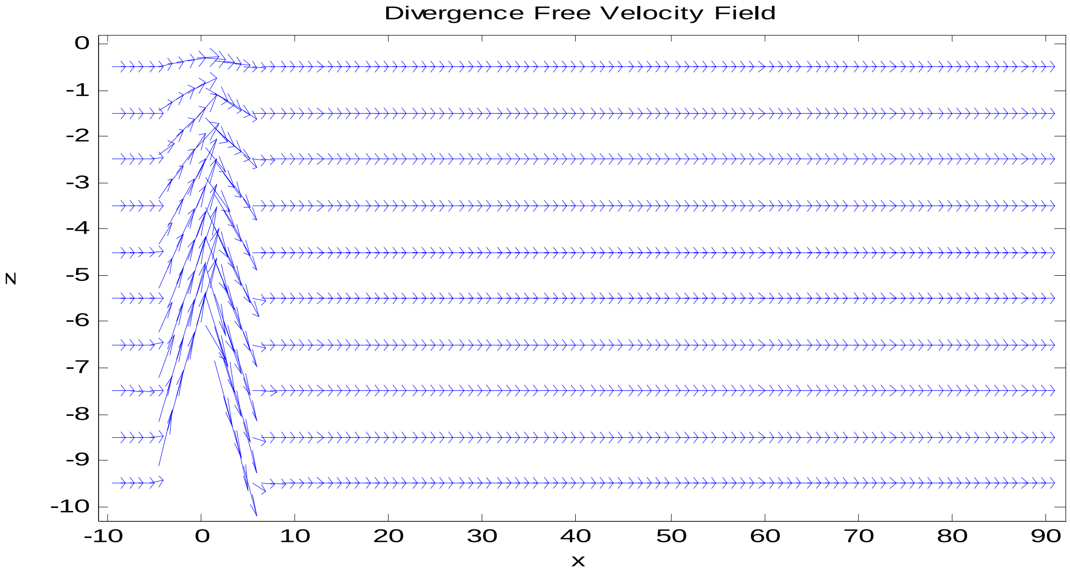


Figure 3.11: Numerically divergence free velocity field

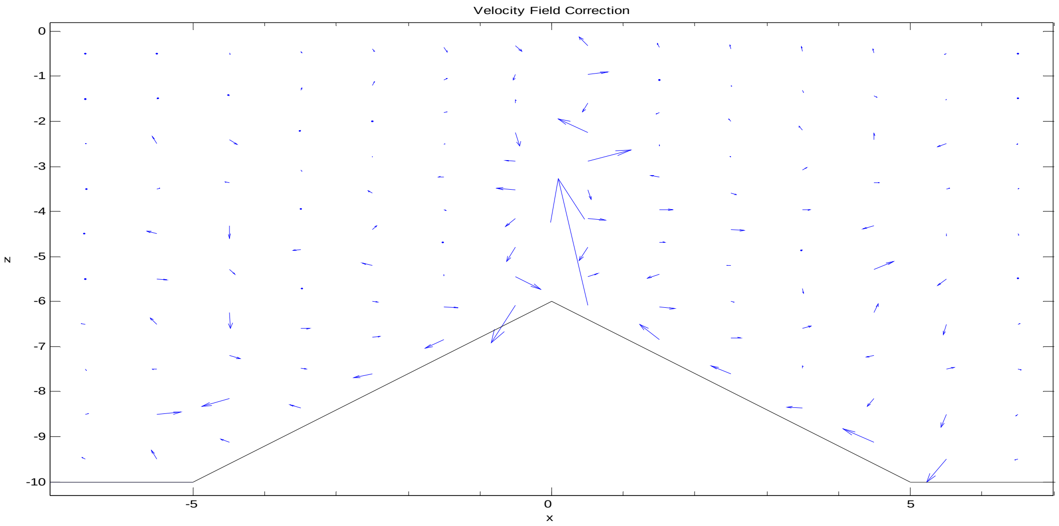


Figure 3.12: Correction of velocity field (small domain, around discontinuity points)

3.6.3 One Dimensional Problem

In the second test problem, the focus is on the limit solution. The domain is given in Figure 3.13, while the velocity profile is as follows :

$$u = 1; \quad 0 \le x \le 0.5, \quad 0 \le y \le 1$$
$$u = 0; \quad 0.5 < x \le 1, \quad 0 \le y \le 1$$

$$v = 0; \quad everywhere$$

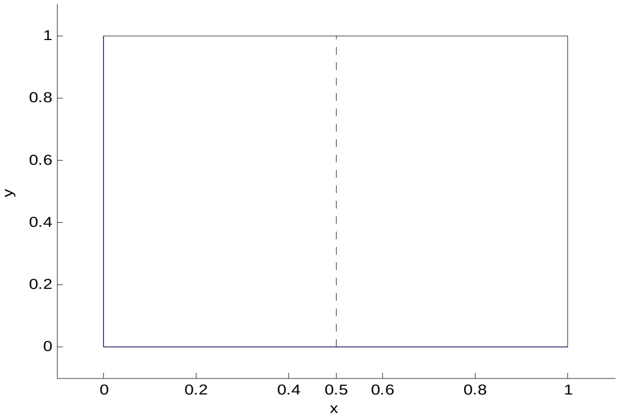


Figure 3.13: Problem Domain

An equidistant square grid is used for the computational purpose. The numerical computation using the original velocity profile results in nonzero divergence around discontinuity points. This is given in Figure 3.14.

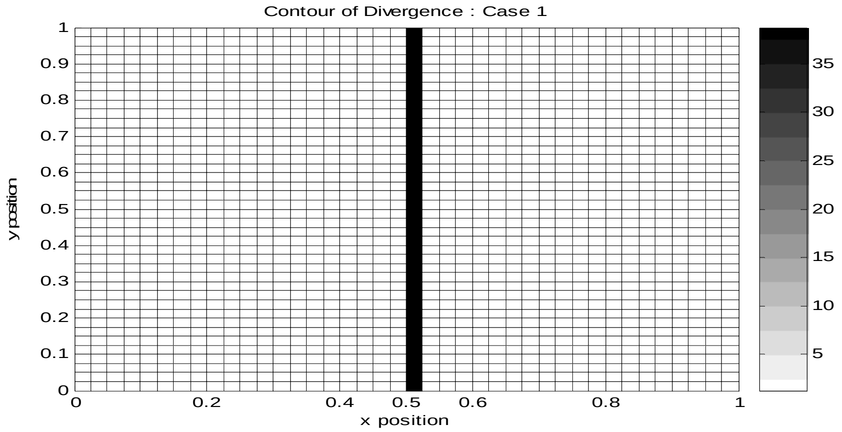


Figure 3.14: Numerical divergence within cells before any correction

Choosing the parallel method as the iterative procedure will not give any change on the flux at horizontal faces because the net flux in the neighbouring cells will cancel each other. In Figure 3.15, the horizontal face *e* is shared by two cells in the same column, namely cell-1and cell-2 with net fluxes FT_1 and FT_2 respectively. Because of residing on the same column, there holds $FT_1 = FT_2$. At the iteration - *k*, the flux at face *e* will be corrected according to the parallel procedure as follows

$$\begin{aligned} flux_e(k+1) &= flux_e(k) + \frac{1}{4}FT_1(k) - \frac{1}{4}FT_2(k) \\ &= flux_e(k) + \frac{1}{4}\left(FT_1(k) - FT_2(k)\right) \qquad ; FT_1(k) = FT_2(k) \\ &= flux_e(k) \end{aligned}$$

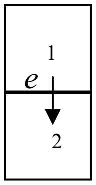


Figure 3.15

Beside that, there is no flux through the face *e* ($flux_e = 0$) as the consequence of zero normal velocity. Therefore the net flux within a cell is only affected by fluxes through the vertical faces only. The numerical scheme in these cases can then be considered as resulting from 1D numerical computation.

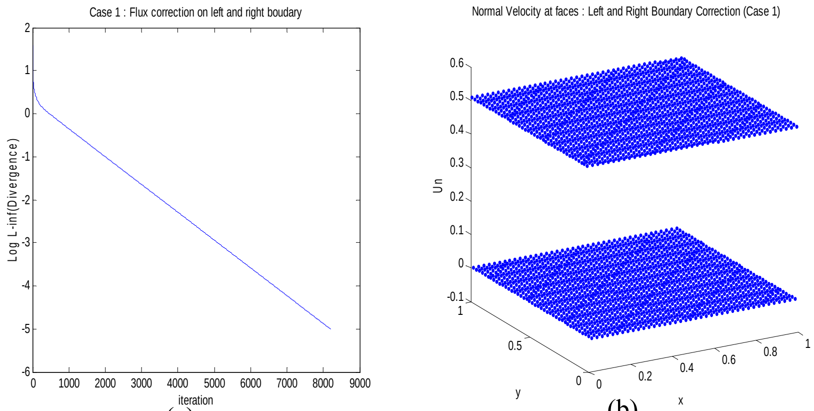
The next subsection will give the implementation results of the parallel iterative procedure. In this experiment, the flux correction is allowed to take place on the left and right boundary.

3.6.4 Results

In [5], a 1 D analysis has been done on this test problem. A zero velocity divergence in 1D implies a solution of constant velocity. When flux correction is allowed to take place on the left and right boundary of the domain, the solution that minimizes the L_2-norm of the velocity correction (Δu) is given by

$$u(x) = \frac{1}{2}, x \epsilon [0,1] \tag{3.29}$$

The numerical results using the parallel procedure are given in two plots of Figure 3.16. Figure 3.16(a) shows that the L_∞ of the divergence of velocity converges to zero. Figure 3.16(b) presents the plot of normal velocity at faces when a divergence free condition with tolerance 10^{-5} is achieved. It has been discussed earlier that at horizontal faces, the normal velocity will remain zero. This is why we found many points with exactly zero normal velocity in that figure. The normal velocity at vertical faces, which is the main interest in our concern, goes to a constant limit solution of 0.5.



(a) (b)

Figure 3.16: (a) Log L_∞-norm of divergence (b) Normal Velocity at faces

For the vertical faces, the horizontal velocity equals the normal velocity. Thus the limit solution given by the numerical computation satisfies the 1D analysis. This implies the procedure converges to a limit solution that minimizes the L_2-norm of $\Delta \boldsymbol{u}$.

3.7. Comparison of Iterative Methods

In the last section, the orthogonal projection technique using Gauss Seidel suffered from a slow convergence. Then it is a good idea to check the performance of other iterative methods. In this section, the performance of four iterative methods in solving the projection equation will be compared. These methods are Jacobi (Endlich's parallel procedure), Gauss Seidel (Endlich's sequential procedure), Symmetric Successive Overrelaxation (SSOR), and Conjugate Gradient (CG). For the implementation, we rely upon MATLAB. Moreover, for CG, we use the MATLAB standard routine *pcg*. With respect to SSOR, we have chosen to start off with 20 Gauss Seidel iterations. This enables an estimation of the spectral radius of the Gauss Seidel iteration matrix and, next, to determine an estimation of ω_{opt} [7].

For this purpose, the second test problem from the previous section is used. The stopping criterion for all the iterative methods is chosen such that the residual $\leq$ 1.5e-7. This residual is chosen instead of the maximum divergence over cells because we want to focus on the solution of the projection equation which delivers φ as the output. However, for checking we compute the final max-div as well.

The results appear in Table 3-1 & 3-2. The Conjugate Gradient (CG) method is better than the rest in the sense of computational time and number of iteration. At this point, it is reasonable to apply a preconditioner to this best method and see whether it can give considerable effect. This method is well known as Preconditioning Conjugate Gradient (PCG).

Table 3-3 shows the performance of PCG and it shows less iterations but poorer computational time. A refinement on the grid reveals a significant impact to the computational time. This could be a major drawback in time-dependent computations. For this reason, we will use the Conjugate Gradient Method in the time dependent computation in later sections.

Table 3-1 Jacobi & Gauss Seidel

Grid	Jacobi			Gauss Seidel		
	n.iter	Time	Max(Div)	n.iter	Time	Max(Div)
100x100	56571	116.516	2.1236e-6	Out	of	Memory
60x60	20943	16.016	1.6457e-6	10474	573.25	1.6596e-6
40x 40	9555	3.6090	1.3434e-6	4779	72.735	1.3628e-6
20x20	2532	0.2970	9.5085e-7	1268	1.4220	9.7065e-7
10x10	691	0.0630	6.6154e-7	347	0.0620	6.9103e-7

Table 3-2 SSOR & Conjugate Gradient (CG)

Grid	SSOR			Conjugate Gradient		
	n.iter	Time	Max(Div)	n.iter	Time	Max(Div)
100x100	Out	Of	Memory	144	0.8910	3.4448e-6
60x60	830	263.719	9.5318e-6	86	0.2180	1.3318e-6
40x 40	448	29.203	6.4521e-6	56	0.0780	7.9987e-7
20x20	184	0.9380	3.1254e-6	24	0.0310	1.4252e-7
10x10	87	0.0470	1.2421e-6	10	0.0150	1.9469e-9

Table 3-3 Preconditioned Conjugate Gradient (PCG)

Grid	PCG		
	n.iter	Time	Max(Div)
100x100	6	1.7030	3.4448e-6
60x60	4	0.3600	1.4797e-6
40x 40	3	0.1250	1.7325e-6
20x 20	2	0.0470	1.0038e-6
10x10	2	0.0310	9.0099e-9

In the above implementation, the MATLAB routine *pcg* is run by first building the matrix *A* of the projection equation. MATLAB also provides another means to run this routine, i.e. the matrix free technique. Table 3-4 presents the performance of this matrix free Conjugate Gradient method and it appears that the conventional technique (building the matrix *A*) is somewhat faster. So we will choose the conventional technique while running the Conjugate Gradient.

Table 3-4 Matrix-free Conjugate Gradient

Grid	Matrix free Conjugate Gradient		
	n.iter	Time	Max(Div)
100x100	144	1.0000	3.4479e-6
60x60	86	0.2340	1.3303e-6
40x 40	56	0.1410	8.0060e-7
20x 20	24	0.0320	1.4269e-7
10x10	10	0.0150	1.9377e-9

4. Projection Schemes For The Depth-Averaged Transport Equation

In the last chapter, the concept of orthogonal projection methods has been developed under the realm of a divergence free velocity equation. In this chapter, the idea is expanded to the case of shallow water flow. Then two test problems of shallow water flow are introduced in the last section.

4.1 Introduction

The continuity equation of depth-averaged shallow water flow can be derived from the divergence free velocity equation, $\nabla . u = 0$ by averaging over the vertical.

The derivation can be found in many references such as [8]. The equation reads

$$\frac{\partial h}{\partial t} + \nabla\left(UH\right) = 0 \tag{4.1}$$

where *U* is depth averaged velocity, *h* is the water elevation measured from the reference level while *H*=*h*+*d* is the total water depth.

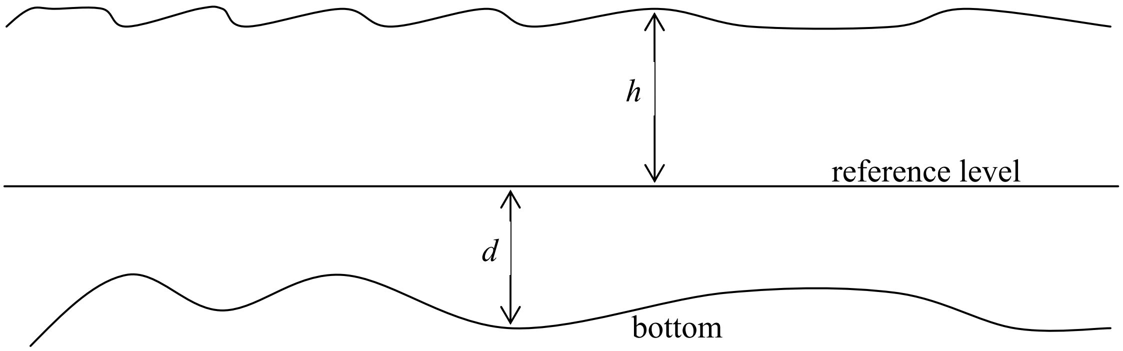


Figure 4.1 Shallow water flow

It is expressed in 2D flow by

$$\frac{\partial h}{\partial t} + \frac{\partial\left[u\left(d+h\right)\right]}{\partial x} + \frac{\partial\left[v\left(d+h\right)\right]}{\partial y} = 0 \tag{4.2}$$

whereas in 1D flow :

$$\frac{\partial h}{\partial t} + \frac{\partial [u(d+h)]}{\partial x} = 0 \tag{4.3}$$

In the case where the width is constant, the 1D shallow water equation can be expressed in terms of water discharge,

$$B\frac{\partial h}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{4.4}$$

where B is the width and $Q = uA_x = u(d+h)B$

4.2 Projection Scheme for Shallow Water Equation

The generic transport equation of shallow water flow associated with (4.1) reads :

$$\frac{\partial Hc}{\partial t} + \nabla.(UHc) = 0 \tag{4.5}$$

where *c* is the concentration of the transported substance in the water.

The finite volume semi-discretization of the transport equation is

$$A\left.\frac{d(Hc)}{dt}\right|_{cell} = \sum_{e \in cell} (-1)^{\alpha(cell,e)} u_e H_e c_e l_e \tag{4.6}$$

where *A* is the area of the cell, *H*$_e$ is the water depth at face *e* and the other terms remain the same as in the previous chapter.

Using the Crank Nicholson method to approximate the time derivative, (4.6) turns into

$$A_{cell}\frac{(H_{cell}c_{cell})^{t+1} - (H_{cell}c_{cell})^{t}}{\Delta t} + \frac{1}{2}\left(\sum_{e \in cell}(-1)^{\alpha(cell,e)} u_e H_e c_e l_e\right)^{t+1} + \frac{1}{2}\left(\sum_{e \in cell}(-1)^{\alpha(cell,e)} u_e^V H_e c_e l_e\right)^{t} = 0$$

For simplicity, we will use superscript 0 to indicate the old time level and no superscript for the new time level :

$$A_{cell}\frac{\left(H_{cell}c_{cell}\right)-\left(H_{cell}c_{cell}\right)^{0}}{\Delta t}+\frac{1}{2}\left(\sum_{e\in cell}\left(-1\right)^{\alpha(cell,e)}u_{e}H_{e}c_{e}l_{e}\right)+\frac{1}{2}\left(\sum_{e\in cell}\left(-1\right)^{\alpha(cell,e)}u_{e}H_{e}c_{e}l_{e}\right)^{0}=0 \qquad (4.7)$$

Substituting c =1 reduces (4.7) into

$$A_{cell}\frac{H_{cell}-H_{cell}^{0}}{\Delta t}+\frac{1}{2}\left(\sum_{e\in cell}\left(-1\right)^{\alpha(cell,e)}u_{e}H_{e}l_{e}\right)+\frac{1}{2}\left(\sum_{e\in cell}\left(-1\right)^{\alpha(cell,e)}u_{e}^{V}H_{e}l_{e}\right)^{0}=0$$

The depth of reference level d doesn't change in time, hence ΔH = Δh and the last equation becomes

$$A_{cell}\frac{h_{cell}-h_{cell}^{0}}{\Delta t}+\frac{1}{2}\sum_{e\in cell}\left(-1\right)^{\alpha(cell,e)}u_{e}H_{e}l_{e}+\frac{1}{2}\sum_{e\in cell}\left(-1\right)^{\alpha(cell,e)}u_{e}^{0}H_{e}^{0}l_{e}=0 \qquad (4.8)$$

One can verify that this is the equation resulting from the finite volume discretization of (4.1) using the implicit trapezoidal rule. Now, choosing Equation (4.8) as the discrete continuity equation automatically determines the consistency between the flow and transport computation. Now it remains to provide the conservative inputs u_e and H_e that satisfy (4.8).

The role of projection method, as described in the previous chapter, is to correct the given coefficients in the staggered grid so as to make them conservative. The requirement data in a control volume is at least having the water elevation (h_{cell}) at cell center and water depth (H_e) and velocity (u_e) at the face centers. From the solver one usually obtains data of water elevation at the cell center. The water depth at face centers can then be obtained from the interpolation of water elevation at cell centers. But this is beyond the scope of our investigation. We just assume that, by any means, the water depth (or water discharge) at the face center are known.

We write (4.8) by arranging the coefficients of new time level in the left hand side :

$$\frac{A_{cell}h_{cell}}{\Delta t}+\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}u_eH_el_e=\frac{A_{cell}h_{cell}^0}{\Delta t}-\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}u_e^0H_e^0l_e \tag{4.9}$$

Because this is a time dependent computation, we propose to do the correction on the coefficients of new time level and keep the value of old time level. In fact, on the old time level the original values are always taken. The alternative is to take the previously corrected data but this has the danger of drifting away too far as time goes on.

While in the divergence free velocity equation the correction is performed only on the normal velocity, there are more choices for the continuity equation (4.9). Three choices of correction are possible in every cell, i.e. :

1. Correcting the value $\left(A\frac{h}{\Delta t}\right)_{cell}$ at the cell center and the discharge flux ($Q_e = u_eH_el_e$) at the face.
2. Correcting the normal velocity u_e or discharge Q_e at the face and retaining the value of water depth.
3. Correcting the water elevation h_{cell} at cell center and total water depth H_e at face while retaining the value of normal velocity.

For the first choice, we cannot use the primary variables as the corrected coefficients because otherwise we will deal with a problem in a nonlinear system. If this is the case, then our projection methods can not be applied.

There might be a situation when the available data at the face center is discharge Q_e instead of u_e and $H_{e.}$ Given this condition, the second choice becomes correcting the water discharge Q_e and the third does correcting the water elevation h_{cell} only.

4.2.1 Correcting the coefficients at cell centers and face centers

To formulate the correction scheme based on the first choice, we use Q_e to represent $u_eH_el_e$ in the discrete continuity equation :

$$\frac{A_{cell}h_{cell}}{\Delta t}+\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}Q_e=\frac{A_{cell}h^0_{cell}}{\Delta t}-\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}Q^0_e$$

$$\left(\frac{A}{\Delta t}h\right)_{cell}+\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}Q_e=\left(\frac{A}{\Delta t}h^0\right)_{cell}-\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}Q^0_e \qquad (4.10)$$

Based on the Equation (4.10) for each cell, we formulate a matrix equation

$$LF=g \qquad (4.11)$$

where $F=\left[\left(\frac{A}{\Delta t}h\right)_{cell\_1}\ .\ .\ .\left(\frac{A}{\Delta t}h\right)_{cell\_nc}\quad Q_{e\_1}.\ .\ .Q_{e\_nf}\right]^T$ is the vector of coefficients at cell centers and face centers, and g is the vector of right hand side of (4.9). The size of matrix L is therefore nc x (nc+nf).

Having (4.10), we can adopt the technique used in the former chapter (basic idea of orthogonal projection methods) to formulate the correction equation. The constraint of boundary conditions entails no correction on some fluxes at the boundary. Let's denote F_1 as the vector of coefficients that needs correction and F_2 as the vector of coefficients whose values have been fixed as the original data. If nfb is the number of coefficient that is not corrected, then we can arrange a matrix L_1 of the size ncc x (ncc + nfc -nfb) and a matrix L_2 of the size ncc x nfb such that (4.11) can be decomposed into

$$L_1F_1+L_2F_2=g \qquad (4.12)$$

Let $\widetilde{F}_1$ and F_1' be the original data and the correction respectively so that $F_1=\widetilde{F}_1+F_1'$, it follows that

$$L_1\left(\widetilde{F}_1+F_1'\right)+L_2F_2=g$$

$$L_1F_1'=g-L_2F_2-L_1\widetilde{F}_1$$

$$L_1F_1'=r_F\quad;\quad r_F=g-L_2F_2-L_1\widetilde{F}_1 \qquad (4.13)$$

Following the first projection method, we then obtain the projected conservative inputs :

$$F_1 = \widetilde{F}_1 + L_1^T A^{-1} r_F \qquad ; A = L_1 L_1^T$$

$$F_1' = L_1^T A^{-1} r_F = L_1^T (L_1 L_1^T)^{-1} r_F$$

4.2.2 Correcting the normal velocity

Fixing the water depth leaves us only with the normal velocity. Thus the second correction scheme can give us freedom in applying all the types of orthogonal projection described in Chapter 3. Rearrange (4.8) by putting the normal velocity of new time level on the left hand side :

$$\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)} u_e H_e l_e = \frac{A_{cell}}{\Delta t}\left(h_{cell}^0 - h_{cell}\right) - \frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)} u_e^0 H_e^0 l_e \qquad (4.14)$$

The set of equations of (4.14) for every cell can be organized in the matrix equation

$$LDq = g \qquad (4.15)$$

where $q = \left[u_{e\_1} . \quad . \quad . u_{e\_nf}\right]^T$ is the vector of normal velocities at faces, D is a diagonal matrix containing coefficients $H_e l_e$ of every face, and g is the right hand side of (4.14).

Using the same procedure taken for the derivation of (3.2) until (3.5), we will arrive at

$$L_1 D_1 q_1' = r_{\widetilde{q}} \quad ; \quad r_{\widetilde{q}} = g - LD\widetilde{q} \qquad (4.16)$$

where $\widetilde{q}$ is the vector of original normal velocity value and q_1' is the vector of correction associated with the normal velocity that needs to be corrected.

From (4.16), we can proceed with the projection result given by (3.11), (3.16), (3.19) and (3.23).

4.2.3 Correcting the water depth

Preserving the value of normal velocity at faces, the equation (4.9) is rearranged as follow

$$\frac{A_{cell}}{\Delta t}h_{cell}+\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}u_e l_e H_e=\frac{A_{cell}h^0_{cell}}{\Delta t}-\frac{1}{2}\sum_{e\in cell}(-1)^{\alpha(cell,e)}u^0_e H^0_e l_e \tag{4.17}$$

We set the matrix equation based on (4.17) as :

$$LF = g$$

where $F=\left[h_{cell\_1}.\quad .\quad .h_{cell\_nc}\quad H_{e\_1}.\quad .\quad .H_{e\_nf}\right]^T$ is the vector of water elevation on cell centers and total water depth on face centers.

This has the same form as (4.11), hence by means of the procedures pacing from (4.11) to (4.13), we obtain the projection result on the corrected water depth *H*

$$F_1'=L_1^T(L_1L_1^T)^{-1}r_F \qquad ; \quad r_F=g-L_2F_2-L_1\widetilde{F}_1$$

with $L_1, F_1', \widetilde{F}_1, L_2$ and F_2 being derived from the analogous idea in the Section 4.2.1.

4.3 Test Problems

Two test problems are addressed to the idea of projection schemes for shallow water equation. Their purposes are to demonstrate how an orthogonal projection method recovers the constancy condition for a specified numerical scheme by correcting the available input. The first test problem is taken from [9] while the second is from [10]. Both tests only enable us to use the first projection method (minimal flux correction) because the velocity (or discharge) of our concern is one-dimensional. In these test problems there holds that the mass conservation equation of the flow is satisfied analytically for the input data. Deviations occur because of discretization.

4.3.1 First Test Problem : Spreading of a Water Drop

- **Problem Description**

This test problem is within the framework of the one-dimensional shallow water equation. In dimensionless notation, the governing equation, in absence of surface friction and background rotation can be expressed as

$$\frac{\partial H}{\partial t}+\frac{\partial (uH)}{\partial x}=0$$

$$\frac{\partial Hu}{\partial t}+\frac{\partial (uHu)}{\partial x}=-H\frac{\partial H}{\partial x}$$

where *u* is the velocity in the *x*-direction, and *H* is the depth of the fluid layer. We consider the spreading of a parabolic-shaped drop of shallow water on a horizontal plane. The drop is initially confined to $|x|\le 1$ according to

$$H(x,t=0)=\begin{cases}1-x^2, |x|\le 1\\ 0, \qquad |x|\ge 1\end{cases}$$

Upon releasing the drop, it spreads under the effect of gravity. The temporal evolution of this system has been analytically investigated by Frei [11]. He noted that the parabola shape is always retained and that the velocity across the drop is a linear function of *x.* In our case, the dimensionless solution can be represented by

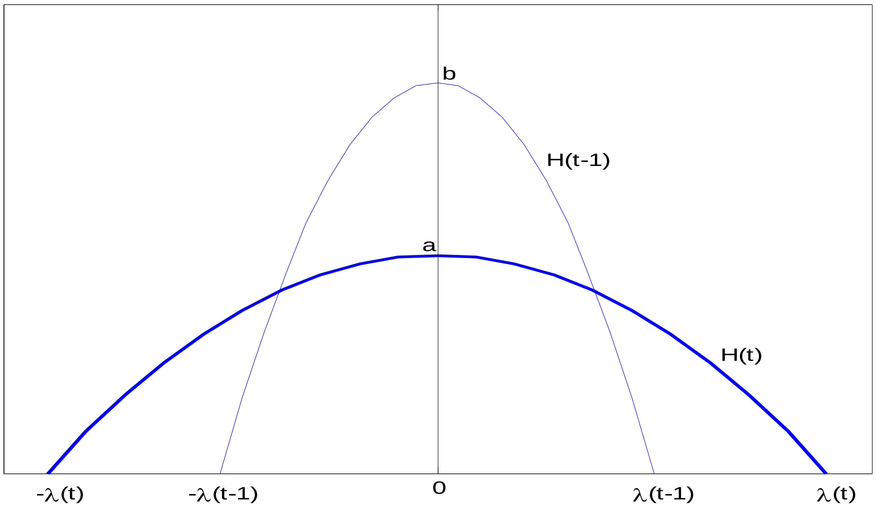
$$\begin{aligned}&H(x,t)=\lambda^{-1}\left[1-(x/\lambda)^2\right]\\&u(x,t)=x(\lambda_t/\lambda),\end{aligned} \qquad (4.18)$$

where λ describes the half-width of the drop, and $\lambda_t=\frac{\partial\lambda}{\partial t}$ is the velocity of the leading edge. Following [11], the function $\lambda(t)$ is obtained numerically as the root of the equation

$$t=\tfrac{1}{2}\left[\sqrt{\lambda(\lambda-1)}+\ln\left(\sqrt{\lambda-1}+\sqrt{\lambda}\right)\right]$$

and

$$\lambda_t=2\sqrt{1-\lambda^{-1}}$$



$a = 1/\lambda(t\text{-}1)$
$b = 1/\lambda(t)$

Figure 4.2 The drop of shallow water

The Figure **Error! Reference source not found.** illustrates how the water drop spreads with the width $2\lambda(t)$ at time t and fills up further area which was still dry at time t-1. A full numerical treatment covering the entire domain has to deal with a moving boundary mesh [12] and this is not a part of our investigation. We confine the test of the constancy condition to the domain of interval $-1 \leq x \leq 1$ which is the initial spread of the water drop.

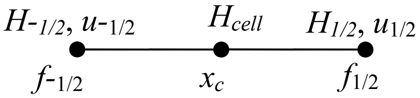


Figure 4.3 The available data in one grid cell for test problem-1

To perform the test on this problem, the domain will be meshed with an equidistant grid and we assume that the input data is available as given in Figure 4.3, i.e. the water depth H_{cell} at cell centers (x_c), the water depth $H_{1/2}$ & $H_{-1/2}$ and velocity $u_{1/2}$ & $u_{-1/2}$ at faces ($f_{1/2}$ & $f_{-1/2}$). These data are provided by the values given by the analytical solutions (4.18) at the corresponding points.

Given this setting, the objective is to satisfy the constancy condition of the discretized version of the 1D shallow water transport equation,

$$\frac{\partial Hc}{\partial t}+\frac{\partial(uHc)}{\partial x}=0\,.$$

which is approximated by the 1D version of transport equation (4.7). For an equidistant grid cell with inputs defined in Fig.(4.3), by placing the old time level at the right hand side, it reads

$$\frac{\Delta x}{\Delta t}H_{cell}c_{cell}+\frac{1}{2}H_{1/2}u_{1/2}c_{1/2}-\frac{1}{2}H_{-1/2}u_{-1/2}c_{-1/2}=\frac{\Delta x}{\Delta t}H^{0}_{cell}c^{0}_{cell}-\frac{1}{2}\left(H^{0}_{1/2}u^{0}_{1/2}c^{0}_{1/2}-H^{0}_{-1/2}u^{0}_{-1/2}c^{0}_{-1/2}\right)$$

where $c_{1/2}$ and $c_{-1/2}$ stand for the concentration at face $f_{1/2}$ and $f_{-1/2}$ respectively.

The flow equation for consistency with this transport equation is then given by :

$$\frac{\Delta x}{\Delta t}H_{cell}+\frac{1}{2}H_{1/2}u_{1/2}-\frac{1}{2}H_{-1/2}u_{-1/2}=\frac{\Delta x}{\Delta t}H^{0}_{cell}-\frac{1}{2}\left(H^{0}_{1/2}u^{0}_{1/2}-H^{0}_{-1/2}u^{0}_{-1/2}\right)$$

One can also check that this is the 1D version of (4.9) by taking $H_{cell} = h_{cell}$.

Given the last equation and the available data, the following presents the choices of correction in the projection method :

1. Correcting the value $\left(\frac{\Delta x}{\Delta t}H\right)_{cell}$ at the cell centers and the flux $H_e u_e$. at the faces.
2. Correcting the normal velocity u_e at the face and retaining the value of water depth.
3. Correcting the water depth H_{cell} at cell center and total water depth H_e at face while retaining the value of normal velocity.

On both boundaries, the variables are corrected since there is no boundary condition that has to be fulfilled in this test problem.

4.3.1 Second Test Problem : Water Wave

The second test problem concerns a shallow water wave governed by the following continuity and momentum equations :

$$B\frac{\partial h}{\partial t}+\frac{\partial Q}{\partial x}=0 \tag{4.19}$$

$$\frac{\partial Q}{\partial t}+gA_s\frac{\partial h}{\partial x}+\kappa Q=0 \tag{4.20}$$

In these equations, *B* is the width of the channel, *Q* is the water discharge, *g* is the gravity constant, A_s is approximated by $A_s = Bd$ where *d* is the depth of reference level, and $\kappa=\frac{8}{3\pi}c_f\frac{\hat{U}}{R}$. Here, $\hat{U}$ is a reference velocity, $R=\frac{A_s}{B+2d}$ and $c_f=\frac{g}{C^2}$ where *C* is the Chezy constant.

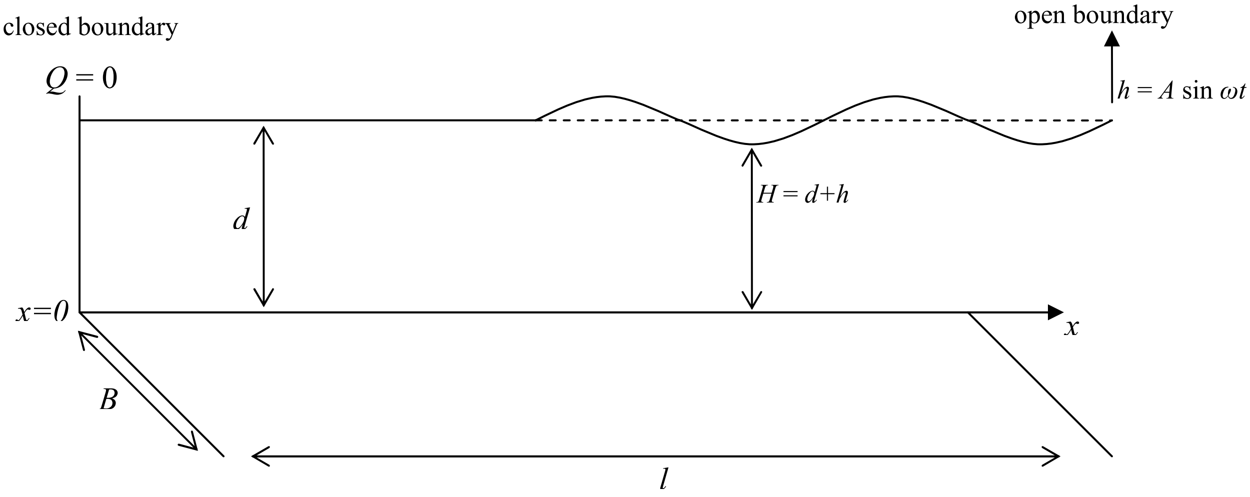


Figure 4.4 A shallow water wave moving in x-direction

The solutions to (4.19) and (4.20)are given by

$$h(x,t)=\mathrm{Re}\{\tilde{h}(x)\exp(j\omega t)\} \tag{4.21}$$

$$Q(x,t)=\mathrm{Re}\{\tilde{Q}(x)\exp(j\omega t)\} \tag{4.22}$$

where

$$\tilde{h}(x)=\frac{\cosh(px)}{\cosh(pl)}.\tilde{h}(l)$$

$$\widetilde{Q}(x) = -\frac{i\omega B}{p}\frac{\sinh(px)}{\cosh(pl)}.\widetilde{h}(l)$$

$$p = i\frac{\sqrt{\omega^2 - i\omega\kappa}}{c_0} \qquad ; \qquad c_0 = \sqrt{gA_s / B}$$

For the numerical simulation, we use the following values of all parameters :

$A = 0.5$ m, $\omega = 2\pi / T$ with $T = 5400$ s.

$l = 247000$ m, $B = 300$ m, $d = 20$ m

$\hat{U} = 0.721$ m/s, $\widetilde{h}(l) = 0.5$ m, $C = 70$

The domain is covered by an equidistant grid and we assume that the input data is available such as highlighted in Figure 4.5, i.e. the water elevation h_{cell} at cell centers (x_c), and water discharge $Q_{1/2}$ & $Q_{-1/2}$ at faces ($f_{1/2}$ & $f_{-1/2}$). These data are determined from the analytical solutions (4.21) and (4.22)

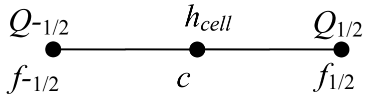


Figure 4.5 The available data in one grid cell for test problem-2

The transport equation in this context is

$$B\frac{\partial(Hc)}{\partial t} + \frac{\partial Qc}{\partial x} = 0 \qquad (4.23)$$

Its finite volume discretization using the implicit trapezoidal rule results in :

$$\frac{B\Delta x}{\Delta t}\left(H_{cell}c_{cell} - H^0_{cell}c^0_{cell}\right) + \left(\frac{1}{2}\left(Q_{1/2}c_{1/2} - Q_{-1/2}c_{-1/2}\right) + \frac{1}{2}\left(Q^0_{1/2}c^0_{1/2} - Q^0_{-1/2}c^0_{-1/2}\right)\right) = 0$$

$$B\frac{\Delta x}{\Delta t}H_{cell}c_{cell} + \frac{1}{2}Q_{1/2}c_{1/2} - \frac{1}{2}Q_{-1/2}c_{-1/2} = B\frac{\Delta x}{\Delta t}H^0_{cell}c^0_{cell} - \frac{1}{2}Q^0_{1/2}c^0_{1/2} + \frac{1}{2}Q^0_{-1/2}c^0_{-1/2} \qquad (4.24)$$

Substituting $c = 1$ gives rise to the discrete continuity equation which is consistent with the last transport equation :

$$B\frac{\Delta x}{\Delta t}h_{cell}+\frac{1}{2}Q_{1/2}-\frac{1}{2}Q_{-1/2}c_{-1/2}=B\frac{\Delta x}{\Delta t}h_{cell}^{0}-\frac{1}{2}Q_{1/2}^{0}+\frac{1}{2}Q_{-1/2}^{0} \tag{4.25}$$

To take care that the input becomes conservative, the first projection method will be applied with the following choices of the correction :

1. Correcting the value $\left(\frac{B\Delta x}{\Delta t}h\right)_{cell}$ at cell centers and the discharge flux $Q_{e.}$at faces.
2. Correcting the discharge Q_e at the face and retaining the value of water depth.
3. Correcting the water elevation h_{cell} cell centers while retaining the value of water discharge.

We keep the condition on the closed boundary, therefore no correction is imposed here. Since the value of water elevation at the closed boundary face is not given (see Figure 4.5), we preserve its value at the nearest cell center to the closed boundary.

5. Numerical Results

This chapter presents the numerical results associated with the two test problems described in the previous chapter. Simulations were done with respect to the model that has been developed. Accordingly, the procedure was carried out by first projecting the model input which makes use of the Conjugate Gradient as the iterative method, and next by computing the concentration (transport quantity) on the basis of the projected data. For this, we use (4.24) with $c_{1/2} = ½ (c_o + c_1)$. In the present context such a simple implementation of the transport solver suffices. Our only goal is to check for constancy.

5.1 First Test

The simulations for the first test are performed until $T = 50$ while varying the grid size and the time step. The initial concentration is $c = 1$at every grid point.

5.1.1 Correction of water depth (*H*) at cell centers and discharge (*UH*) at faces

The results of accuracy are given in Table 5-1 in term of the maximum deviation from concentration = 1 over the entire domain and time interval. The transport computation using the projected data is compared with the transport computation using the original data only.

Table 5-1 Accuracy of constancy condition

Δt	Δx	Max(|concentration – 1|)		Max (Courant number)
		Original data	Projected Data	Original data
0.01	0.05	4.1920e-004	3.3501e-011	0.1501
0.1	0.05	0.0191	1.1617e-009	1.4993
1	0.05	0.6336	5.1922e-007	14.4539
1	0.01	0.4552	1.0272e-006	73.7519
1	0.001	0.4557	1.5806e-006	741.2250

The results show that the numerical scheme is stable, with a very high Courant number (η=741.2250) the concentration is still bounded. There is a significant difference of accuracy where the transport computation using the projection method is much better. Computation with the original data produces a maximum of deviation of 0.4557 for the highest Courant number while the accuracy of almost 10^{-6} is obtained by means of the projection method.

Table 5-2 explains why inaccuracy occurs when using the original data. The tolerance of residual is not satisfactory, thus contributing to the error of constancy condition. Evidently from that table, using the projected data can recover the conservativeness of the input.

Table 5-2 Data Conservativeness

Δt	Δx	Max(| Residual |)	
		Original data	Projected Data
0.01	0.05	1.50 e-5	8.42 e-12
0.1	0.05	6.36 e-4	8.60 e-11
1	0.05	1.84 e-2	3.39 e-9
1	0.01	3.90 e-3	7.04 e-10
1	0.001	3.98 e-4	4.22 e-10

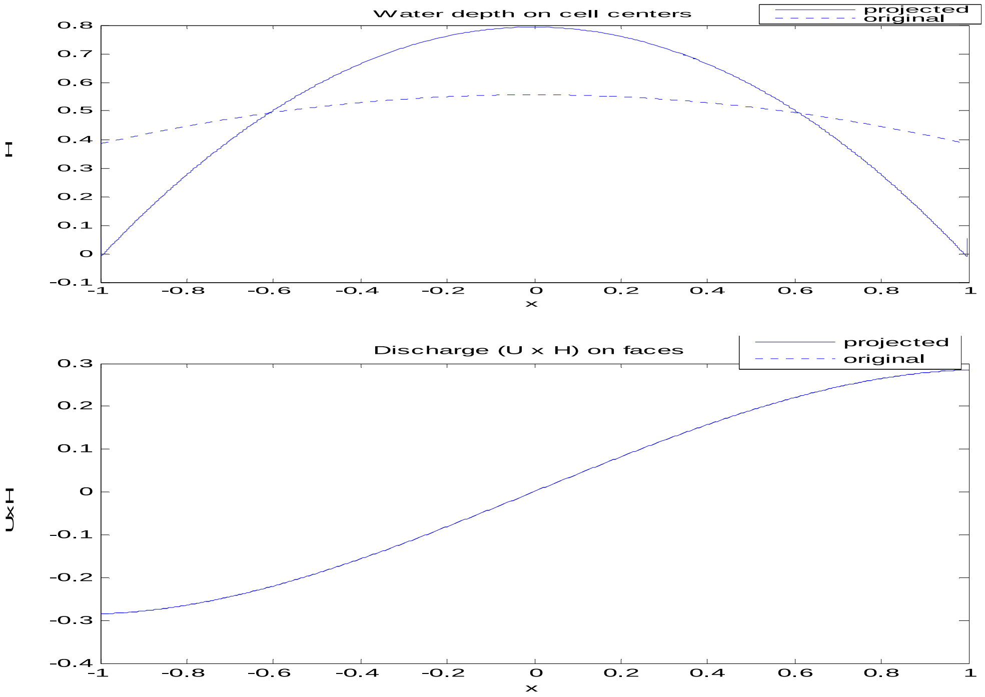


Figure 5.1 Water depth (*H*) and discharge (*UH*) at T = 1 (Δt = 1; Δx = 0.001)

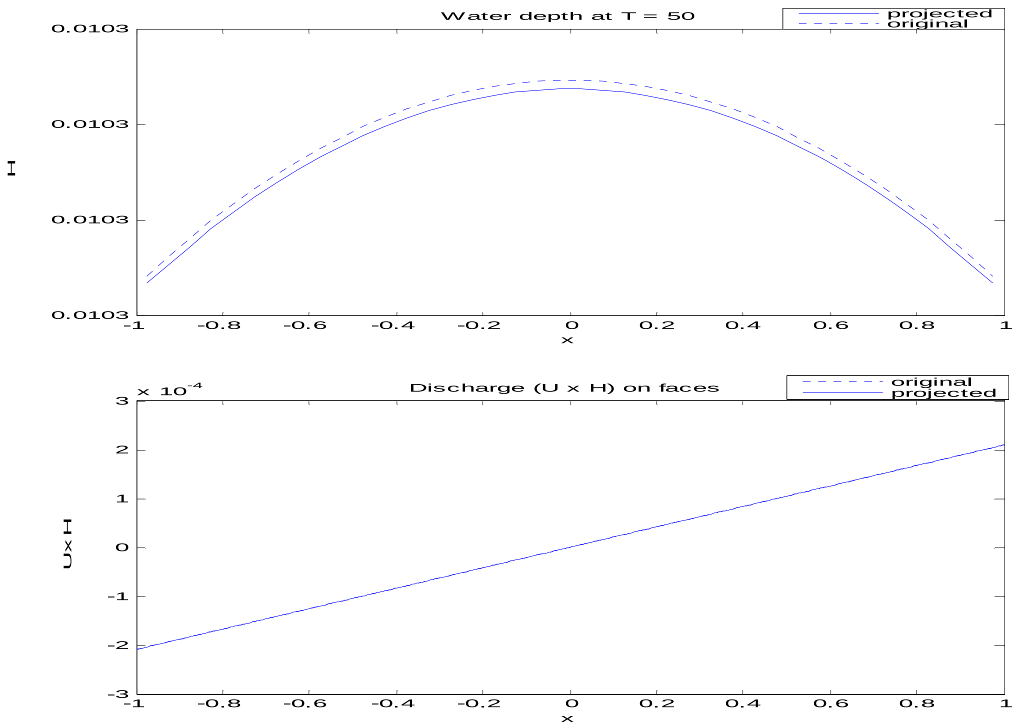


Figure 5.2 Water depth (*H*) and discharge (*UH*) at *T* = 50 (Δ*t* = 1; Δ*x* = 0.001)

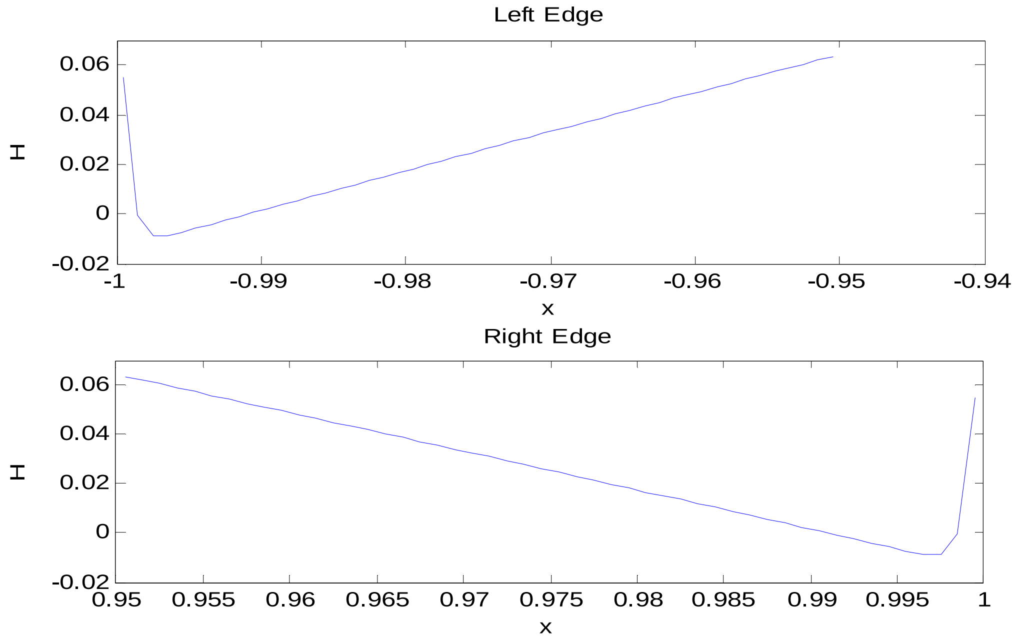


Figure 5.3 Negative water depth near the left and right edge at *T* = 1 (Δ*t* = 1; Δ*x* = 0.001)

In Figure 5.2 (at *T* = 50), the water depth *H* and discharge *UH* are small and everything seems normal. The projected data (both *H* and *UH*) don't differ significantly from the original data.

A contrast can be observed at $T = 1$, as depicted in Figure 5.1. A large correction of water depth is needed to satisfy the constancy condition. Moreover, there is a negative value of total water depth near the left and the right edge, which is indicated in Figure 5.3. This implies some dry regions occur in the interval [-1,1].

5.1.2 Correction of velocity at faces

By fixing the value of water depth, the positivity of water depth can be preserved. Thus correcting the velocity appears to be more accepted than the former. The accuracy of the constancy condition can be achieved with a tolerance below 10^{-5} for the maximum Courant number of 961.1514. Only a slight compensation is the decreasing slope of velocity at the left and right edge at T=1 (Figure 5.4).

Table 5-3 Accuracy of constancy condition for velocity correction

Δt	Δx	Max(|concentration – 1|)		Max (Courant number)	
		Original data	Projected Data	Original data	Projected Data
0.01	0.05	4.1920e-004	2.7982e-012	0.1501	0.1541
0.1	0.05	0.0191	4.8504e-010	1.4993	1.5522
1	0.05	0.6336	1.7741e-010	14.4539	19.2296
1	0.01	0.4552	3.9777e-006	73.7519	96.1174
1	0.001	0.4557	7.6988e-006	741.2250	961.1514

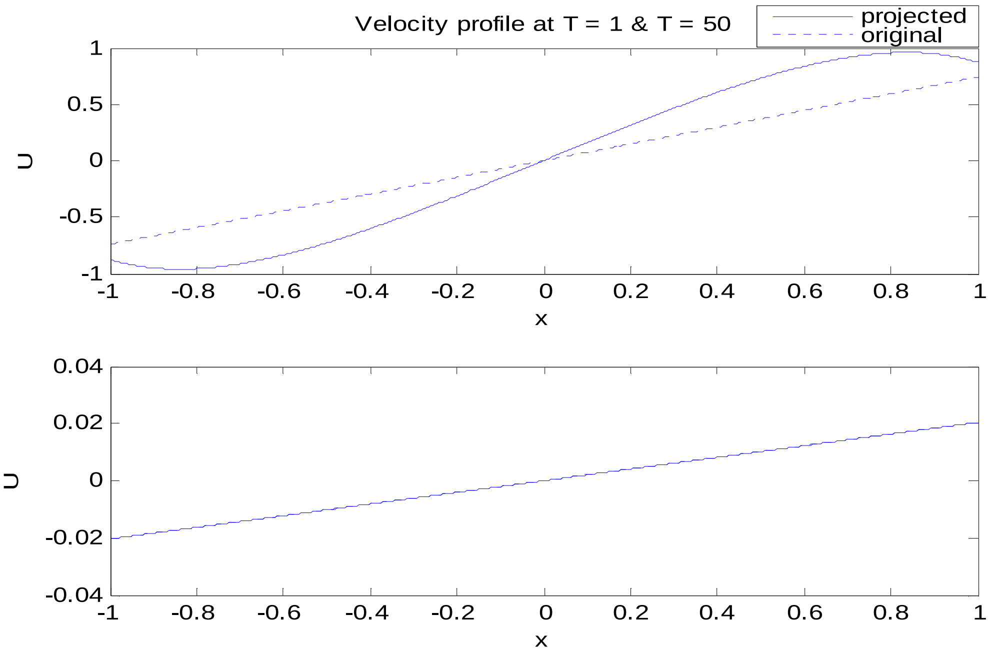


Figure 5.4 Velocity profile at T = 1 and T = 50 (Δt = 1; Δx = 0.001)

The velocity profiles of projected and original data appear to coincide at T = 50, but Figure 5.5 demonstrates that the transport computation using original data cannot recover the constancy condition anymore (at T=50) because of the inaccuracy produced in early timesteps. Figure 5.6 shows the maximum error generated over the entire domain when using the original data.

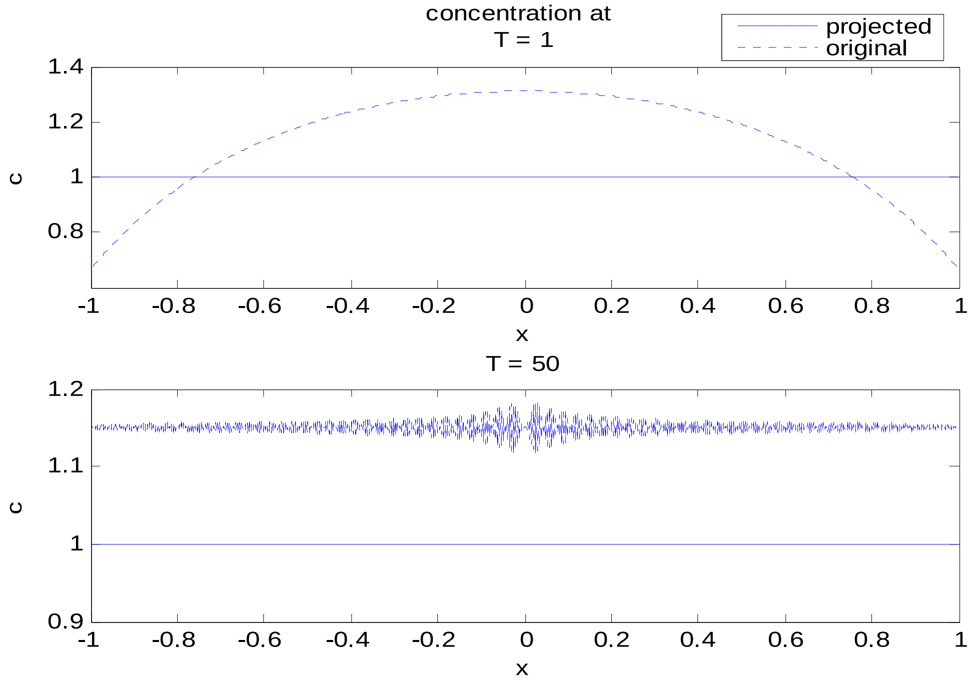


Figure 5.5 Concentration profile at T = 1and T = 50 (Δt = 1; Δx = 0.001)

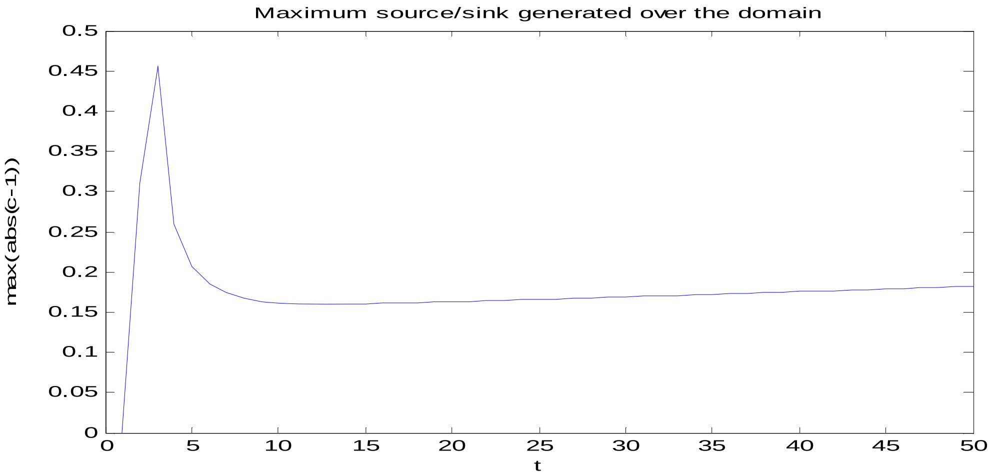


Figure 5.6 Maximum source/sink generated when using the original data

(Δt = 1; Δx = 0.001)

5.1.3 Correction of the water depth

The projection method by correcting the water depth and maintaining the value of velocity satisfies the constancy condition with a tolerance below 10^{-5}. The result of the projection of the water depth is ensured its positivity. However, as shown in Figure 5.7, the mass of water mounted up in the interval [-1,1] is considerably bigger than given by the original data.

Table 5-4 Accuracy of constancy condition for the correction on water depth

Δt	Δx	Max(|concentration – 1|)		Max (Courant number)
		Original data	Projected Data	
0.01	0.05	4.1920e-004	2.4094e-011	0.1501
0.1	0.05	0.0191	3.5984e-010	1.4993
1	0.05	0.6336	7.0691e-008	14.4539
1	0.01	0.4552	2.9810e-006	73.7519
1	0.001	0.4557	3.4720e-006	741.2250

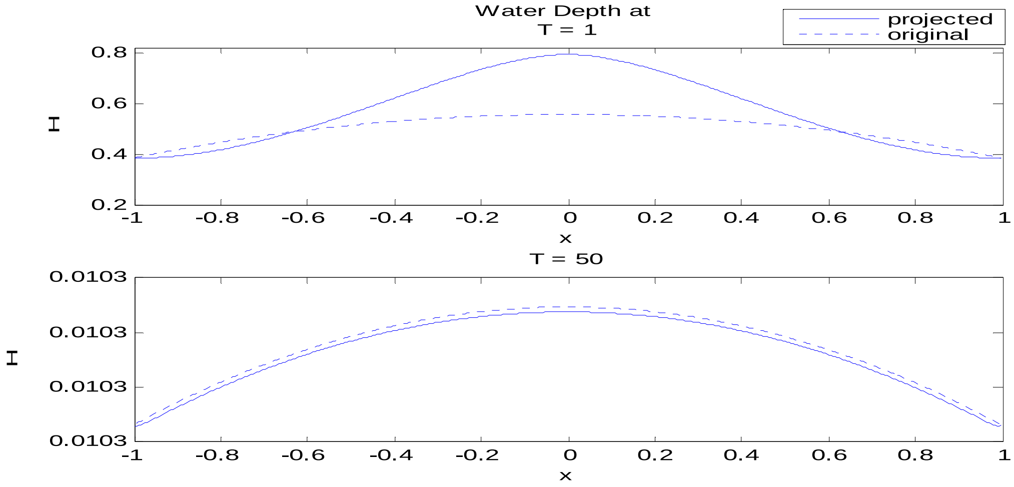


Figure 5.7 Water depth at *T* = 1 & *T* = 50 (Δ*t* = 1; Δ*x* = 0.001)

5.2 Second Test

The simulations for the second test are performed until T = 108000 while varying the grid size and the time step. The initial concentration is c = 1at every grid point. The elapsed time in every table of accuracy refers to the total computational time when using the projected data.

5.2.1 Correction of water elevation (*h*) at cell centers and discharge (*Q*) at faces

Using the original data only, a significant deviation takes place as can be seen in Table 5-5. Figure 5.9 shows that the deviation is present most of the time. The projected method provides input that can give a tolerance lower than 10^{-7} for Δt=1000 and Δx=10. The Courant number for this specification is 63.1743.

Table 5-5 Accuracy of constancy condition

Δt	Δx	Max(|concentration – 1|)		Max (Courant number)	Elapsed Time
		Original data	Projected Data		
100	100	1.7631e-004	1.9409e-010	0.6317	359.39 s
1000	1000	0.0182	1.7954e-009	0.6316	2.75 s
1000	100	0.0182	1.8897e-008	6.3174	24.56 s
1000	50	0.0182	2.2467e-008	12.6348	63.53 s
1000	10	0.0182	2.3029e-008	63.1743	1119.2 s

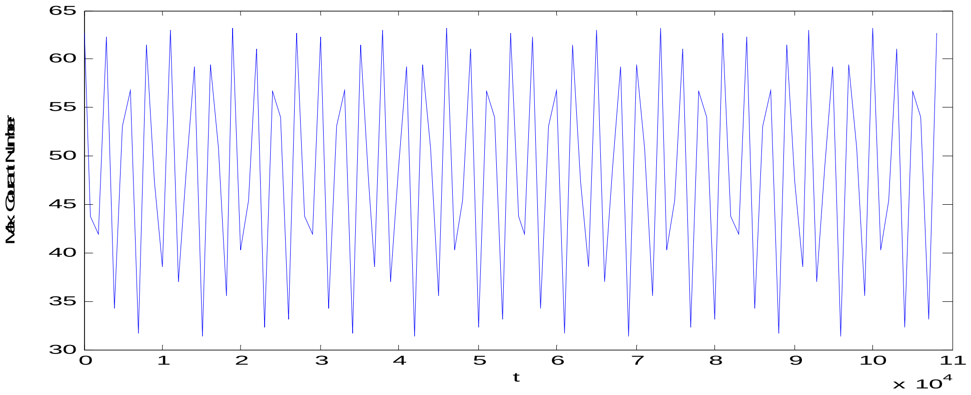


Figure 5.8 Maximum Courant number for Δt = 1000 and Δx = 10

In Figure 5.10, the projected input of water elevation is able to follow the trajectory pattern of the original input. But that doesn't take place on the entire domain. Shown in Figure 5.11, there happens to be a very steep slope near the closed boundary.

Figure 5.12 gives a picture of the coinciding plots of projected and original *Q*. The scale of the quantity of discharge is very large such that the discrepancy is not visible in the figure. Therefore

Figure 5.13 is presented to show the difference between them whereas Figure 5.14 gives the picture in smaller domain in order to show the steep slope near the left boundary.

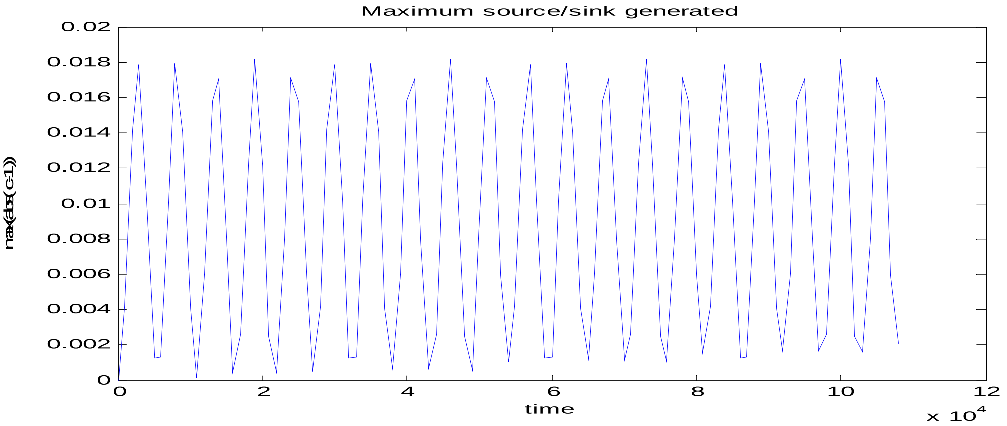


Figure 5.9 Source/sink generated when using original data (Δ*t* = 1000 and Δ*x* = 10)

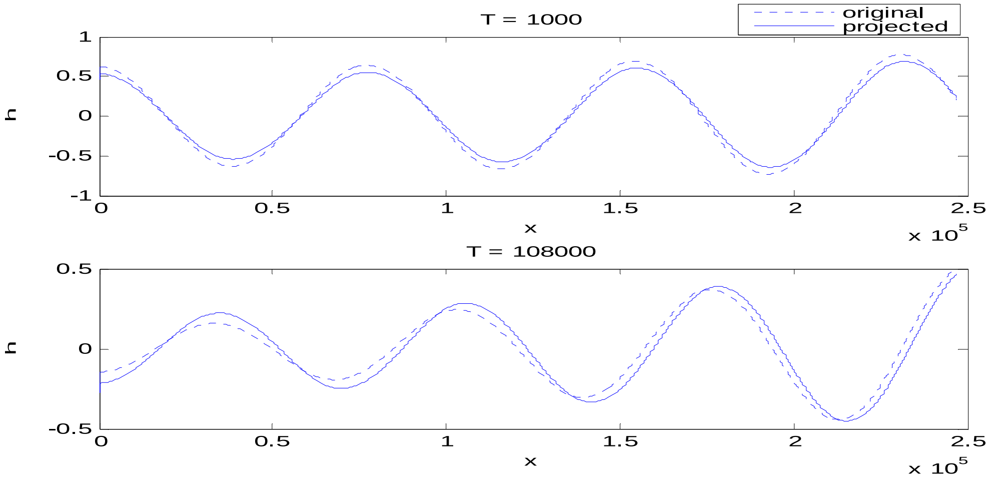


Figure 5.10 Water elevation at *T* = 1000 & *T* = 108000 (Δ*t* = 1000; Δ*x* =10)

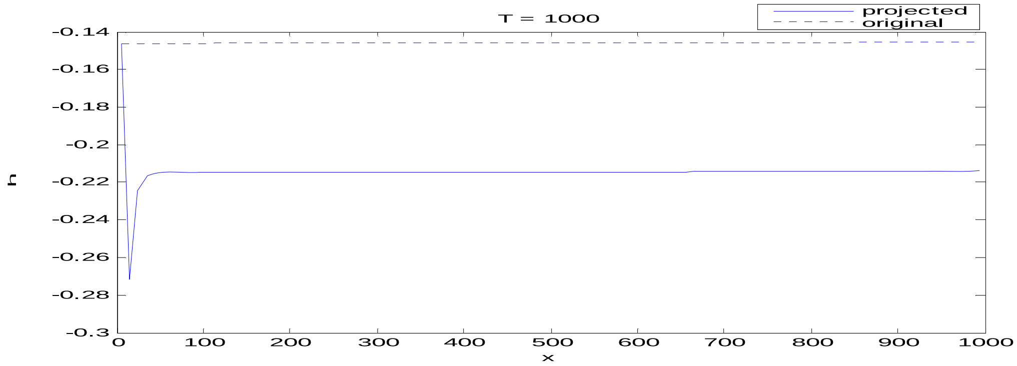


Figure 5.11 Water elevation on the interval [0,1000] at *T*=1000 (Δ*t*=1000; Δ*x*=10)

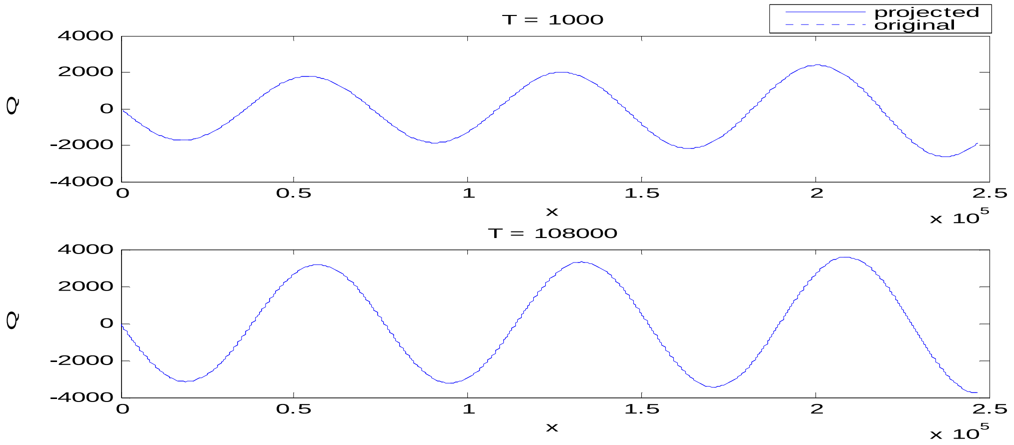


Figure 5.12 Discharge at T=1000 and T=108000 (Δt=1000; Δx =10)

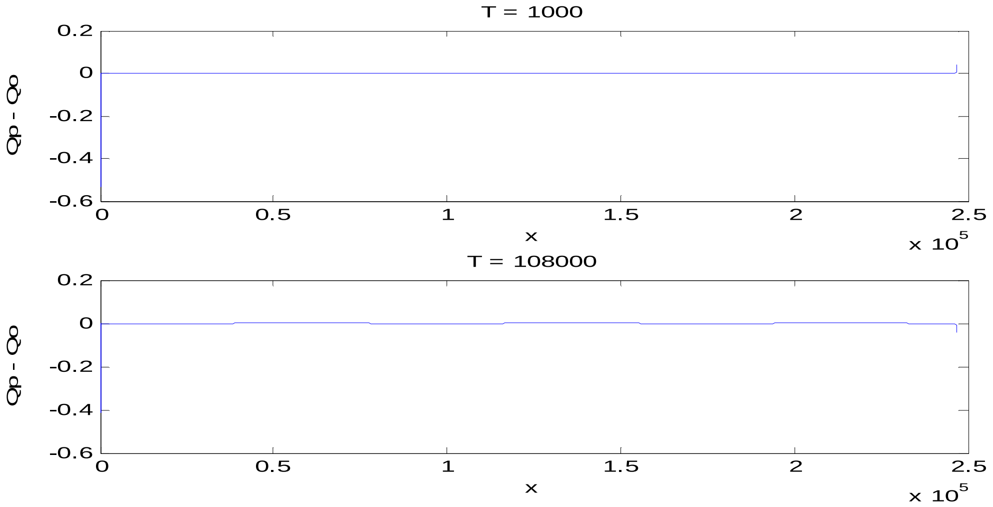


Figure 5.13 Difference between the projected discharge (Q_p) and original discharge (Q_p) at T=1000 and T=108000 (Δt=1000; Δx =10)

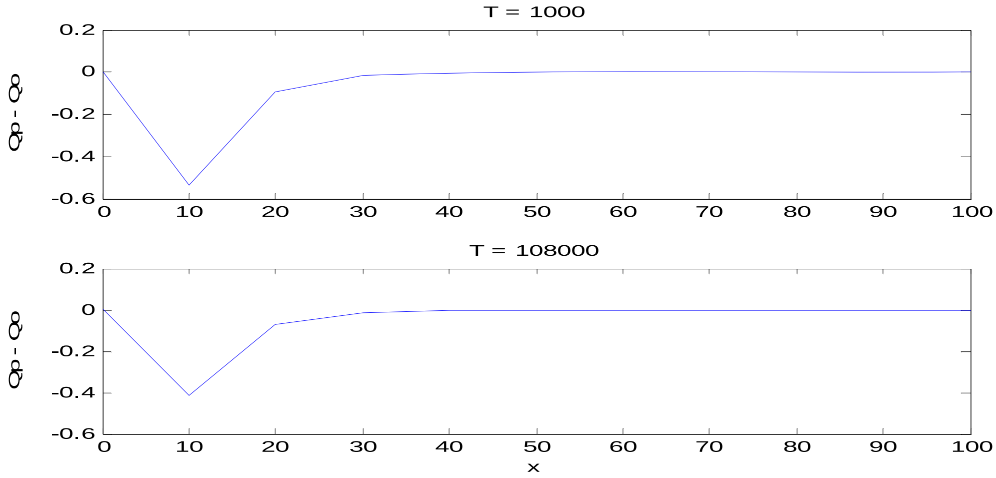


Figure 5.14 Difference between the projected discharge (Q_p) and original discharge (Q_p) on the interval [0,100] at T=1000 and T=108000 (Δt=1000; Δx =10)

5.2.2 Correction of discharge (*Q*) at faces

The accuracy of constancy condition obtained by correcting the discharge at faces is remarkably fascinating as presented in Table 5-6. With the computational scheme of Δ*t*=1000 and Δ*x*=50, the accuracy is achieved below 10^{-10}, much better than correcting discharge and water elevation altogether. But actually this is not really feasible, it needs an extremely lot of iterations to find the solution of projection equation using the Conjugate Gradient iterative method. When the grid size is decreased to Δ*x*=10, it needs 956 seconds for 3 time steps. This is awfully time consuming compared to the correction on both discharge and water elevation which only needs roughly 31 seconds within 3 time steps.

Table 5-6 Accuracy of constancy condition

Δ*t*	Δ*x*	Max(|concentration – 1|)		Max (Courant number)	Elapsed time
		Original data	Projected Data		
1000	1000	0.0182	2.7423e-014	0.7512	9.45 s
1000	100	0.0182	8.7568e-012	7.5091	377.77 s
1000	50	0.0182	4.2377e-011	15.0182	1511.39 s
1000	10	0.0182	Not feasible		956 s (only for 3 time steps)

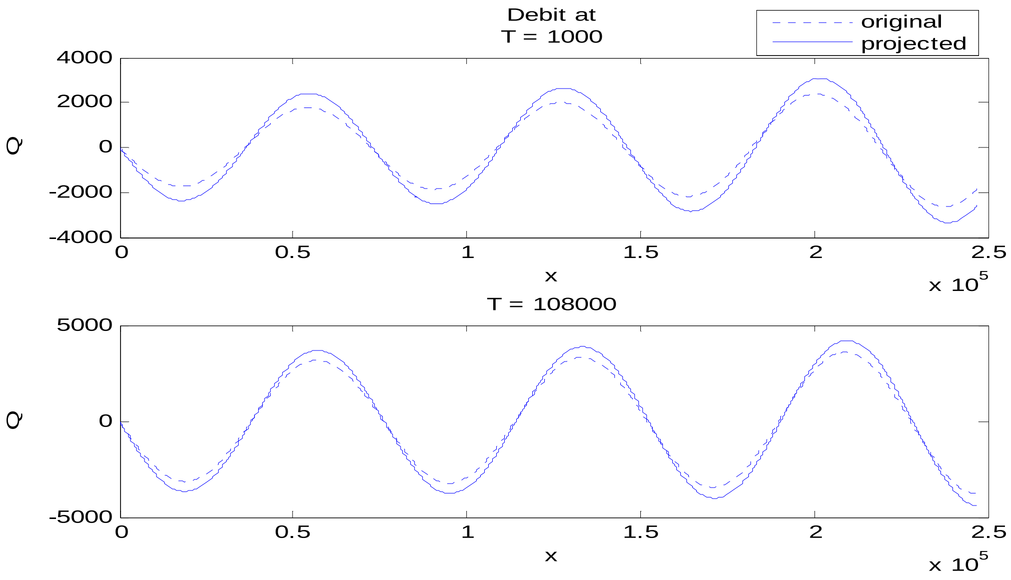


Figure 5.15 Discharge at *T*=1000 and *T*=108000 (Δ*t*=1000; Δ*x* =50)

5.2.3 Correction of water elevation (*h*) at cell centers

Correction on water elevation only doesn't seem to be a good option for this test. It doesn't give good accuracy with respect to the constancy condition as presented in Table 5-7. Although Figure 5.16 shows that the source/sink generated is bounded (the scheme is stable), refining the grid size when Δ*t* = 1000 doesn't improve the accuracy. It looks as if the correction on water discharge has more significant impact in improving accuracy.

Table 5-7 Accuracy of constancy condition

Δ*t*	Δ*x*	Max(|concentration – 1|)		Max (Courant number)	Elapsed Time
		Original data	Projected Data		
100	100	1.7631e-004	4.9825e-005	0.6317	126.64 s
1000	1000	0.0182	0.0051	0.6316	2.09 s
1000	100	0.0182	0.0050	6.3174	14.56 s
1000	50	0.0182	0.0050	12.6348	45.42 s
1000	10	0.0182	0.0050	63.1743	335.38 s

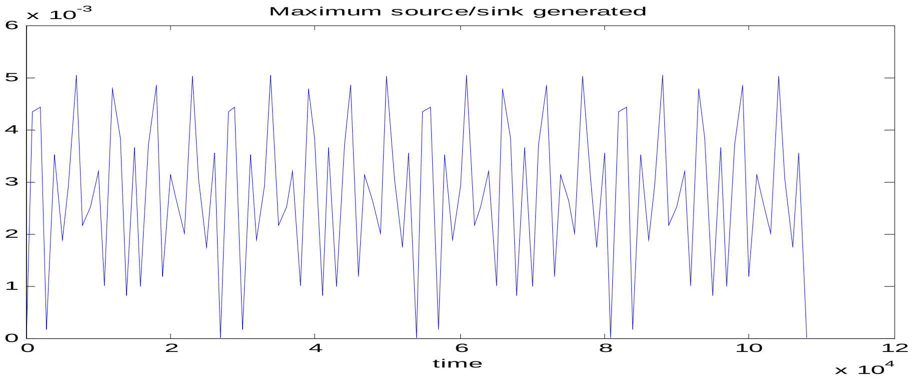


Figure 5.16 Source/sink generated when using the projected data of water elevation (Δ*t* = 1000 and Δ*x* = 10)

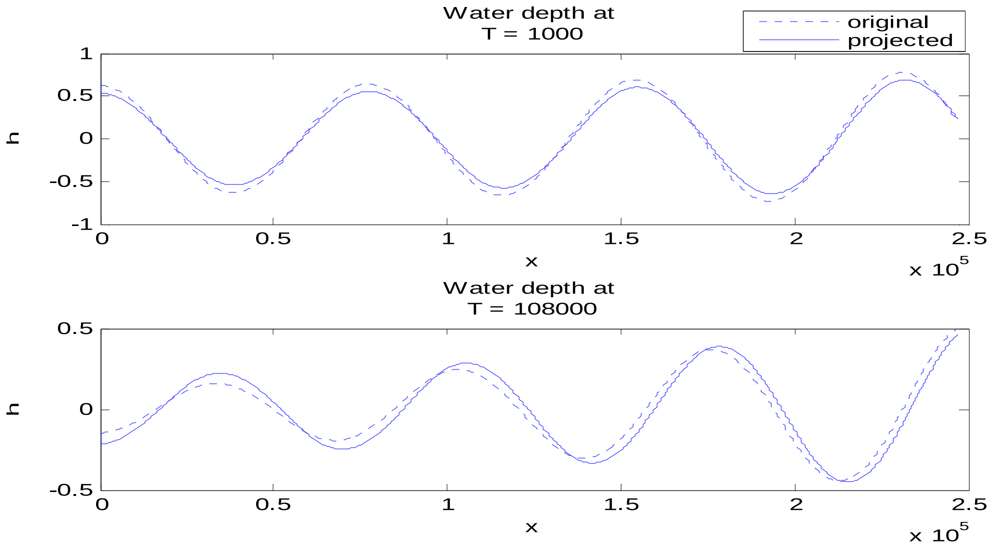


Figure 5.17 Water elevation at *T* = 1000 & *T* = 108000 (Δ*t* = 1000; Δ*x* =50)

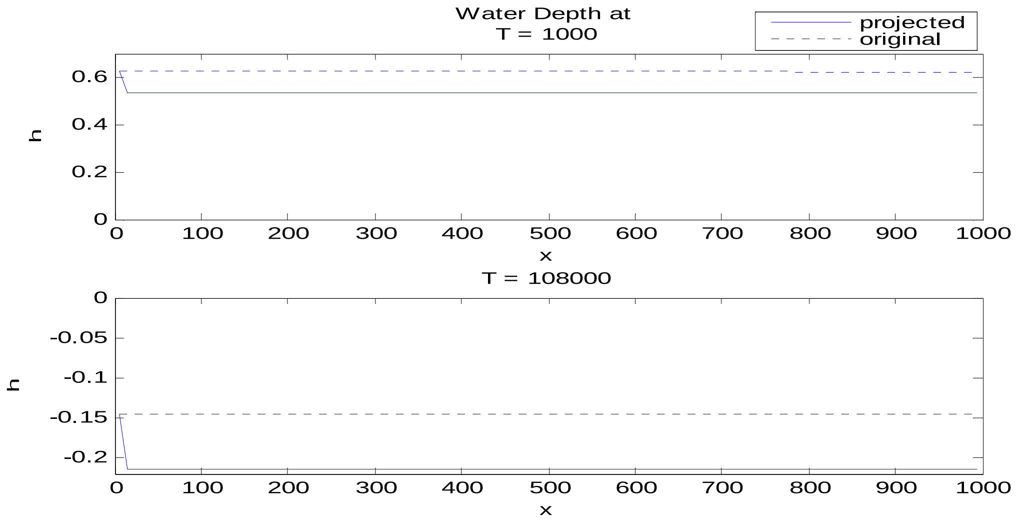


Figure 5.18 Water elevation on domain [0,1000] at *T* = 1000 & *T* = 108000 (Δ*t* = 1000; Δ*x* =50)

6. Conclusion

In this study, four orthogonal projection methods to correct velocity inputs have been developed under the circumstance of a divergence free velocity equation. Each method has its own objective on the projected normal velocity/flux, i.e. either minimal flux correction, minimal velocity correction, minimal shear correction, or zero curl correction. It has been verified that the Endlich's iterative procedure is equivalent to the projection method of minimal flux correction. Two examples have been presented and iterative procedures for solving the correction equation have been discussed.

The projection method is then adopted to derive projection methods in the specific case of shallow water flow. Here, the corrected variables can be velocity, or water elevation or both. It is needed to correct these inputs to satisfy the constancy condition in the transport computation of shallow water flow.

The projection method of minimal flux correction is addressed in two test problems. They are the spread of a water drop and the evolution of a water wave. In the first test, the constancy condition can be achieved with good accuracy even with a high Courant number. This works for any choice of corrected variables. However the scheme suffers from some drawbacks as a result of the correction. When the corrected inputs are water depth and velocity, a large correction of water depth is found and at some points the value of total water depth becomes negative. When the correction is done on velocity only, a slight change to the monotonicity of velocity takes place, while choosing the water depth as the corrected input can significantly affect the total mass of fluid. It is concluded that correcting the velocity only is the best compromise for this test problem.

In the second test, a boundary condition is set on the closed boundary, i.e. no correction is allowed for. Choice of water elevation and discharge as the corrected inputs tops the other choices. Its disadvantage is that the data smoothness near the closed boundary becomes disturbed. On the other hand, correction on discharge gives excellent constancy condition, but the computational time is miserable. Choosing water elevation as the only corrected input doesn't yield a good accuracy. In this case, it looks as if the value of discharge is more dominant in the improvement of accuracy.

References

[1] E.S. Gross, L. Bonaventura and G. Rosatti. Consistency with continuity in conservative advection schemes for free-surface models. International Journal for Numerical Methods in Fluids. 38:307-327, 2002

[2] F.Naifar and P.Wilders. Towards a conservative and positive computation of pollutant transport.European Congress on Computational Methods in Applied Sciences and Engineering. ECCOMAS 2004, 2004.

[3] M. van Loon. Numerical Methods in Smog Prediction. CWI, Amsterdam, 1996.

[4] R.M. Endlich. An iterative method for altering the kinematic properties of wind fields. Journal of Applied Meteorology, 6: 837-844, 1967.

[5] A.L. Siahaan, An Iterative Procedure To Recover The Zero Divergence of Velocity Field. Internship Report. TU Delft, 2005.

[6] R.A. Nicolaides. Direct Discretization of Planar Div-Curl Problems. SIAM Journal of Numerical Analysis, 29: 32-56, 1992.

[7] J.W. Thomas. Numerical Partial Differential Equations : Conservation Laws and Elliptic Equations. Springer-Verlag New York, 1999.

[8] B.L.Mehaute. An Introduction to Hydrodynamics & Water Waves. Springer-Verlag New York, 1976.

[9] C.Schar and P.K. Smolarkiewicz. A Synchronous and Iterative Flux Correction Formalism for Coupled Transport Equations. *Journal of Computational Physics,* 128: 101-120, 1996.

[10] J.A. Battjes. Stroming in Waterlopen. *Faculteit Civiele Techniek en Geowetenschappen*, *TU Delft*, 2001

[11] C.Frei. Dynamics of a two-dimensional ribbon of shallow water on an *f*-plane. *Tellus A*, 45: 44-53, 1993.

[12] C.B. Vreugdenhil. Numerical Methods For Shallow-Water Flow. *Kluwer Academic Publishers*, 1994.

Appendix

Curl Discretization

The volume integral of curl $\boldsymbol{q}$ (denoted by $\nabla \times \boldsymbol{q}$) in 2D, according to Stokes Theorem can be written as

$$\int_V (\nabla \times \boldsymbol{q}).\boldsymbol{dV} = \oint_{\partial V} \boldsymbol{q}.\boldsymbol{t}\, ds \tag{1}$$

In that equation, $\boldsymbol{t}$ denotes the unit tangential vector to ∂V.

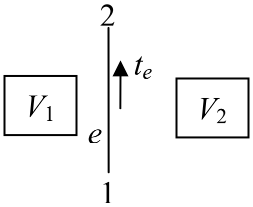


Figure A.1 tangential vector t_e

In 2D domain, the unit tangent t_e of the face e is determined by the end points of the face which has been arranged in the mesh description. In Figure A.1 above, t_e points from end point 1 to end point 2.

For an individual cell V, the tangents over its faces are taken such that they form a counterclockwise direction.

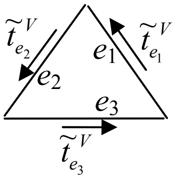


Figure A.2 tangents of cell V over its faces

Based on that, we can define an indicator function for every face e in any volume V :

$$\beta(V,e) = \begin{cases} 0, & \text{if } \tilde{t}^V_e = t_e \\ 1, & \text{if } \tilde{t}^V_e = -t_e \end{cases}$$

Referring to Figure A.2, $\beta(V_1, e) = 0$ and $\beta(V_2, e) = 1$.

Discretization of the right hand side of (1) requires us to introduce the integrated velocity, also known as shear

$$S_e = q_e^t |e|$$

where $q_e^t = q.t_e$ is the tangential velocity on the face *e* and $|e|$ is the length of the face.

Thus the discretization of (1) for any volume *V* can be written as follows

$$\sum_{e \in V} (-1)^{\beta(V,e)} S_e = g_v, \quad \forall V \tag{2}$$

where g_v is the magnitude of numerical curl *q* in the volume *V*.

From (2) can be derived the resulting system of equations :

$$CS = g \tag{3}$$

Denote the number of volumes and faces as *nV* and *nf* respectively, the matrix *C* is the operator from $\Re^{nf}$ to $\Re^{nV}$. Every row of *C* corresponds to a specific volume and all nonzero elements in that row correspond to faces of that volume.