

Re-evaluating efficiency of first-order numerical schemes for two-layer shallow water systems by considering different eigenvalue solutions

Nino Krvavica^{a,*}

^a *University of Rijeka, Faculty of Civil Engineering, Radmile Matejcic 3, 51000 Rijeka, Croatia*

Abstract

The efficiency of several first-order numerical schemes for two-layer shallow water equations (SWE) are evaluated here by considering different eigenvalue solutions. This study is a continuation of our previous work [25] in which we have proposed an efficient implementation of a Roe solver for two-layer SWE based on analytical expressions for eigenvalues and eigenvectors. In this work, the accuracy and computational cost of numerical, analytical, and approximated eigenvalue solvers are compared when implemented in Roe, Intermediate Field CaPturing (IFCP) and Polynomial Viscosity Matrix (PVM) schemes. Several numerical tests are performed to examine the overall efficiency of numerical schemes with different eigenvalue solvers when computing two-layer shallow-water flows. The results confirm that analytical eigenvalue solutions are much faster than numerical solvers, with a computational cost closer to approximate expressions. Consequently, the Roe scheme with analytical solutions to the eigenstructure is equally efficient as the IFCP scheme. On the other hand, IFCP and PVM schemes with analytical solutions to eigenvalues are found to be equally efficient as those with approximated expressions. Analytical eigenvalues show slightly better results when dealing with larger density differences between the layers.

*Corresponding author

Email address: nino.krvavica@uniri.hr (Nino Krvavica)

Keywords:

shallow-water equations, two-layer flow, eigenvalues, Roe scheme, IFCP scheme, PVM scheme, finite volume method

1. Introduction

Two-layer shallow-water equations (SWE) are widely used to simulate geophysical flows in stratified conditions. Some examples of a two-layer configuration include exchange flows in sea straits [15, 14], highly stratified estuaries [23, 24], as well as various types of gravity currents [27, 3], such as mudflows [7], debris flows [35, 30], submarine avalanches [20, 29, 36], and pyroclastic flows [19]. Although such processes can be described more accurately by 3D Navier-Stokes equations, two-layer shallow water models make a popular alternative because of their simplicity and a significantly lower computational cost.

Two-layer SWEs are mathematically defined as hyperbolic systems of coupled conservation laws with source terms, or so-called balance laws [15]. These equations are challenging to solve numerically because of the layer coupling and non-conservative source terms accounting for the variable geometry, friction, or entrainment. Over the last two decades, a numerical resolution of two-layer SWE has been an object of intense research [15, 14, 26, 16, 31, 6, 21, 28, 8, 18]. A popular choice for numerical resolution of two-layer SWEs are finite volume methods (FVM), and among them a family of path-conserving schemes [39, 5, 15, 34, 33].

Implementation of the path-conservative schemes involves a numerical viscosity matrix, which is usually derived from some or all eigenvalues of a corresponding Jacobian of the flux matrix. The choice of the numerical viscosity matrix determines the numerical diffusion and accuracy of the scheme. Since analytical expressions for eigenvalues of two-layer SWE systems were considered unavailable until recently [14, 35, 1, 21, 37], either approximate expressions [38, 1] or numerical algorithms were used instead. Unfortunately, numerical algorithms, such as root-finding and eigensolver methods, make schemes computa-

tionally (too) demanding. For example, several studies evaluating the efficiency of first-order schemes in solving two-layer flows [16, 10, 21], find Roe schemes, which are based on all eigenvalues, to be the most accurate, but far less efficient than some other first-order schemes, mainly because of excessive computational costs when performing a full spectral decomposition by a numerical eigensolver.

Recently, a new solution [25] to the efficiency problem for the Roe scheme was proposed. This new approach introduced a semi-analytical implementation of the Roe scheme based on simple closed-form solutions for the eigenvalues and eigenvectors [25]. New scheme, named A-Roe, was found to be much faster than the numerical implementation of the Roe scheme, while producing equally accurate results [25]. An additional advantage is that closed-form solutions enable a direct and accurate prediction of complex eigenvalues and implementation of a corrective algorithm for the loss of hyperbolicity [25].

Our previous paper [25] evaluated the performance of the A-Roe scheme based on analytical expressions for eigenvalues and eigenvectors, and compared its efficiency to Lax-Friedrichs (LF) and GFORCE scheme, as well as the Polynomial Viscosity Matrix redefinition of the Roe scheme (PVM-Roe) [10] and Intermediate Field CaPturing scheme (IFCP) [21]. The former two do not use any eigenvalue information, whereas the latter two use all four eigenvalues [16, 10, 21]. In the case of PVM-Roe and IFCP schemes, the classical implementation based on approximated eigenvalues was considered. Furthermore, our previous paper [25] focused on overall advantages of using analytical eigenvalues; primarily, increased computational speed and hyperbolicity correction algorithm.

The present study is a natural continuation of the previous paper [25]. One evident question emerged from the main conclusions of the previous paper - if analytical expressions for the eigenstructure significantly improved the efficiency of the Roe scheme, what effect will they have on the performance of other first-order schemes that are also based on some or all eigenvalues? Clearly, the efficiency of those numerical schemes should be re-evaluated.

The main aim of this study is to investigate the sensitivity of several other

numerical schemes, namely PVM-2U [10] and IFCP [21], on the choice of eigenvalues and to evaluate potential benefits of using analytical eigenvalues instead of recommended approximated expressions. For this purpose, the accuracy and computational speed of recently proposed closed-form eigenvalue solutions are carefully compared against two available alternatives - the numerical eigen-solvers and approximated expressions for eigenvalues. Next, the sensitivity of numerical schemes to the choice of an eigenvalue solver is assessed. And finally, the overall computational efficiency of Roe, IFCP, and PVM-2U schemes with different eigenvalues is evaluated by performing several numerical tests which consider different density differences between the layers and different channel geometries. Some significant remarks on the implementation of numerical schemes, which increase their efficiency, are also presented.

2. Two layer shallow-water flow: Theory, eigenvalues, and numerical schemes

2.1. Governing system of equations

A one-dimensional (1D) two-layer shallow-water flow in prismatic channels with rectangular cross-sections of constant width is considered for all tests. The governing system of equations written in a general vector form is repeated here in a more compact form for context and reproducibility [15]:

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{w})}{\partial x} = \mathbf{B}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x} + \mathbf{g}(\mathbf{w}), \quad (1)$$

where x refers to the axis of the channel and t is time. The vector of conserved quantities \mathbf{w} is defined as:

$$\mathbf{w} = \left\{ h_1 \quad q_1 \quad h_2 \quad q_2 \right\}^T, \quad (2)$$

where h_j is the layer thickness (or depth), $q_j = h_j u_j$ is the layer flow rate per unit width, and index $j = 1, 2$ denotes the respective upper and lower layer. The flux vector $\mathbf{f}(\mathbf{w})$ is:

$$\mathbf{f}(\mathbf{w}) = \left\{ q_1 \quad \frac{q_1^2}{h_1} + \frac{g}{2} h_1^2 \quad q_2 \quad \frac{q_2^2}{h_2} + \frac{g}{2} h_2^2 \right\}^T, \quad (3)$$

where g is acceleration of gravity. Matrix $\mathbf{B}(\mathbf{w})$ is a result of coupling the two-layer system, defined as [15]:

$$\mathbf{B}(\mathbf{w}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -c_1^2 & 0 \\ 0 & 0 & 0 & 0 \\ -rc_2^2 & 0 & 0 & 0 \end{bmatrix}, \quad (4)$$

where $r = \rho_1/\rho_2 < 1$ is the ratio between the upper layer density ρ_1 and the lower layer density ρ_2 , and $c_j^2 = gh_j$ is propagation celerity of internal and external perturbations (waves), for $j = 1, 2$. Finally, the bathymetry source term $\mathbf{g}(\mathbf{w})$ is defined as follows [15]:

$$\mathbf{g}(\mathbf{w}) = \left\{ 0 \quad -gh_1 \frac{db}{dx} \quad 0 \quad -gh_2 \frac{db}{dx} \right\}^T, \quad (5)$$

where b is the bed elevation.

2.2. Eigenvalues

The system given by Eq. (1) can be rewritten in the following quasi-linear form [15]:

$$\frac{\partial \mathbf{w}}{\partial t} + \mathcal{A}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x} = \mathbf{g}(\mathbf{w}), \quad (6)$$

where

$$\mathcal{A}(\mathbf{w}) = \frac{\partial \mathbf{f}(\mathbf{w})}{\partial \mathbf{w}} - \mathbf{B}(\mathbf{w}) = \mathbf{J}(\mathbf{w}) - \mathbf{B}(\mathbf{w}) \quad (7)$$

is the pseudo-Jacobian matrix that contains the flux gradient terms as well as the coupling terms:

$$\mathcal{A}(\mathbf{w}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ c_1^2 - u_1^2 & 2u_1 & c_1^2 & 0 \\ 0 & 0 & 0 & 1 \\ rc_2^2 & 0 & c_2^2 - u_2^2 & 2u_2 \end{bmatrix} \quad (8)$$

The four eigenvalues of $\mathcal{A}(\mathbf{w})$ define the propagation speeds of barotropic (external) and baroclinic (internal) perturbations. In most geophysical flows, one of two external eigenvalues is negative $\lambda_{ext}^- < 0$, while the other is positive $\lambda_{ext}^+ > 0$ [1]. Eigenvalues can be computed using numerical solvers, approximated expressions or analytical solutions.

2.2.1. Numerical eigenvalues

Eigenvalues and eigenvectors of $\mathcal{A}(\mathbf{w})$ can be numerically obtained by solving the following equation:

$$\mathcal{A}\mathbf{K} = \mathbf{K}\mathbf{\Lambda}. \quad (9)$$

where $\mathbf{\Lambda}$ is a 4×4 diagonal matrix whose coefficient are the eigenvalues $\lambda_k, k = 1, \dots, 4$, and \mathbf{K} is matrix whose columns are the corresponding right eigenvectors. Usually a QR algorithm is used for this purpose [4].

2.2.2. Approximated eigenvalues

The following approximation derived under the assumption of $r \approx 1$ and $u_1 \approx u_2$ are usually used for computing the internal and external eigenvalues [38]:

$$\lambda_{ext}^{\pm} = U_1 \pm \sqrt{g(h_1 + h_2)} \quad (10)$$

$$\lambda_{int}^{\pm} = U_2 \pm \sqrt{g(1-r) \frac{h_1 h_2}{h_1 + h_2} \left[1 - \frac{(u_1 - u_2)^2}{g(1-r)(h_1 + h_2)} \right]}, \quad (11)$$

with

$$U_1 = \frac{h_1 u_1 + h_2 u_2}{h_1 + h_2} \quad \text{and} \quad U_2 = \frac{h_1 u_2 + h_2 u_1}{h_1 + h_2}. \quad (12)$$

Note that Eqs. (10) and (11) are valid only when dealing with two layers of similar densities ($r = \rho_1/\rho_2 \approx 1$) and when velocities in both layers are comparable ($u_1 \approx u_2$). Those conditions are found in some stratified flows in nature, such as exchange flows through sea straits [14, 17] or some cases of highly stratified estuaries [24]. However, for geophysical flows characterized by a larger relative density difference, such as granular, debris or mud flows, the approximated values may significantly deviate from exact values and cannot accurately predict a possible hyperbolicity loss [1, 37, 25].

2.2.3. Analytical eigenvalues

Recently, a simple closed-form approach for computing real roots of the characteristic quartic Eq. (13) was proposed [25]. The solutions are based on Ferrari's formulas [2], they consist of eight simple evaluations, and are repeated here for consistency and clarity. A detailed derivation of these equations is

available in [25]. The proposed closed-form solutions to eigenvalues are given in terms of coefficients a, b, c and d of a characteristic polynomial of matrix $\mathcal{A}(\mathbf{w})$ [25]:

$$p(\lambda) = \lambda^4 + a\lambda^3 + b\lambda^2 + c\lambda + d \quad (13)$$

with:

$$a = -2(u_1 + u_2), \quad (14)$$

$$b = u_1^2 - c_1^2 + 4u_1u_2 + u_2^2 - c_2^2, \quad (15)$$

$$c = -2u_2(u_1^2 - c_1^2) - 2u_1(u_2^2 - c_2^2), \quad (16)$$

$$d = (u_1^2 - c_1^2)(u_2^2 - c_2^2) - rc_1^2c_2^2. \quad (17)$$

Real eigenvalues are then computed by the following expressions [25]:

$$\lambda_{ext}^{\pm} = \lambda_{4,1} = -\frac{a}{4} \pm \frac{\sqrt{Z} + \sqrt{-A - Z \mp \frac{B}{\sqrt{Z}}}}{2}, \quad (18)$$

$$\lambda_{int}^{\pm} = \lambda_{3,2} = -\frac{a}{4} \pm \frac{\sqrt{Z} - \sqrt{-A - Z \mp \frac{B}{\sqrt{Z}}}}{2}. \quad (19)$$

where

$$Z = \frac{1}{3} \left(2\sqrt{\Delta_0} \cos \frac{\phi}{3} - A \right), \quad (20)$$

$$\phi = \arccos \left(\frac{\Delta_1}{2\Delta_0\sqrt{\Delta_0}} \right), \quad (21)$$

with

$$A = 2b - \frac{3a^2}{4}, \quad (22)$$

$$B = 2c - ab + \frac{a^3}{4}. \quad (23)$$

and

$$\Delta_0 = b^2 + 12d - 3ac, \quad (24)$$

$$\Delta_1 = 27a^2d - 9abc + 2b^3 - 72bd + 27c^2. \quad (25)$$

Note, that it is possible to combine these equations into a single explicit solution in terms of conserved variables, but the resulting formula would certainly be too extensive to be presented in a journal format, and probably not optimized to be implemented in a computational algorithm. However, an example of such expanded formulation is available [40].

2.2.4. Nature of eigenvalues

Since the relative density difference r has a major influence on internal eigenvalues, they are usually smaller than the external ones. Therefore, the following indexing and order of eigenvalues will be used herein:

$$\lambda_1 = \lambda_{ext}^-, \quad \lambda_2 = \lambda_{int}^-, \quad \lambda_3 = \lambda_{int}^+, \quad \lambda_4 = \lambda_{ext}^+ \quad (26)$$

It is worth mentioning that the external eigenvalues λ_{ext}^\pm are always real [32, 1]; however, at sufficiently large relative velocities $\Delta u = |u_1 - u_2|$, as well as for very low or very high relative densities r , the internal eigenvalues λ_{int}^\pm may become complex and the governing system loses its hyperbolic character [32, 12].

Although analytical expressions given by Eqs. (18) and (19) are valid only for real roots, a simple hyperbolicity correction algorithm can be incorporated, which verifies whether the term in Eq. (21) satisfies the condition $|\frac{\Delta_1}{2\Delta_0\sqrt{\Delta_0}}| < 1$, and if not, iteratively corrects the velocities until a hyperbolic state is recovered. This algorithm is not a subject of the current study, but its description and details are available in [25].

2.3. Numerical schemes

A class of path-conservative schemes are considered here to approximate the governing equations for two-layer shallow water flow [33]. A first order accurate path-conservative scheme for Eq. (6) may be written as follows [33]:

$$\mathbf{w}_i^{n+1} = \mathbf{w}_i^n - \frac{\Delta t}{\Delta x} \left(\mathbf{D}_{i-1/2}^+ + \mathbf{D}_{i+1/2}^- \right) \quad (27)$$

where Δx and Δt are the respective spatial and time increment (considered constant here for simplicity), \mathbf{w}_i^n denotes the approximate cell-averaged values of the exact solution obtained by the numerical scheme at cell $I_i = [x_{i-1/2}, x_{i+1/2}]$ in time $t^n = n\Delta t$, and matrices $\mathbf{D}_{i+1/2}^\pm$ are continuous functions of conserved variables $\mathbf{D}^\pm(\mathbf{w}_i^n, \mathbf{w}_{i+1}^n)$.

For the governing system of equations, a generalized numerical scheme based

on Roe linearisation [39] may be written by Eq. (27), with:

$$\begin{aligned} \mathbf{D}_{i+1/2}^{\pm} = & \frac{1}{2} \left[\mathbf{f}(\mathbf{w}_{i+1}^n) - \mathbf{f}(\mathbf{w}_i^n) + \mathbf{B}_{i+1/2}(\mathbf{w}_{i+1}^n - \mathbf{w}_i^n) - \mathbf{g}_{i+1/2} \right. \\ & \left. \pm \mathbf{Q}_{i+1/2} \left(\mathbf{w}_{i+1}^n - \mathbf{w}_i^n - \mathcal{A}_{i+1/2}^{-1} \mathbf{g}_{i+1/2} \right) \right] \end{aligned} \quad (28)$$

where $\mathbf{Q}_{i+1/2}$ is a numerical viscosity matrix that determines the numerical diffusivity of the results, and whose choice depends on a particular scheme [21]. Matrices and vectors $\mathbf{B}_{i+1/2}$, $\mathbf{A}_{i+1/2}$, and $\mathbf{g}_{i+1/2}$ correspond to $\mathbf{B}(\mathbf{w}_{i+1}^n, \mathbf{w}_i^n)$, $\mathbf{A}(\mathbf{w}_{i+1}^n, \mathbf{w}_i^n)$, and $\mathbf{g}(\mathbf{w}_{i+1}^n, \mathbf{w}_i^n)$, respectively, evaluated at the cell interface after a suitable Roe linearization is performed (see [15] and [25] for details).

The viscosity matrix in Roe schemes coincides with the absolute pseudo-Jacobian matrix [15]:

$$\mathbf{Q}_{i+1/2} = |\mathcal{A}_{i+1/2}| \quad (29)$$

and the absolute value of $\mathcal{A}_{i+1/2}$ can be directly obtained from:

$$|\mathcal{A}_{i+1/2}| = \mathbf{K}_{i+1/2} |\mathbf{A}_{i+1/2}| \mathbf{K}_{i+1/2}^{-1}. \quad (30)$$

where $|\mathbf{A}_{i+1/2}|$ is a 4×4 diagonal matrix whose coefficient are the absolute eigenvalues $|\lambda_k|$, $k = 1, \dots, 4$.

When analytical eigenvalues are unavailable, Roe schemes require either approximation of eigenvalues and corresponding eigenvectors or a numerical decomposition to obtain the eigenstructure of matrix $\mathcal{A}_{i+1/2}$. The latter is computational expensive, while the former is less accurate. In both cases, an additional re-composition of the viscosity matrix from eigenstructure is required (see Eq. (30)), which imposes an additional computational cost. This drawback has motivated a development of numerical schemes in which the viscosity matrix $\mathbf{Q}_{i+1/2}$ is directly approximated from pseudo-Jacobian matrix.

One possible way to construct such a scheme is to approximate the viscosity matrix by a polynomial function. Those class of methods are called Polynomial Viscosity Matrix (PVM) [10]. PVM methods define the viscosity matrix through a general polynomial evaluation of the pseudo-Jacobian matrix, given by:

$$\mathbf{Q}_{i+1/2} = p_l(\mathcal{A}_{i+1/2}) \quad (31)$$

where $p_l(x)$ is a polynomial of degree l

$$p_l(x) = \sum_{j=0}^l \alpha_j x^j \quad (32)$$

The main advantage of PVM schemes is that a full spectral decomposition is not required.

In [10] several different PVM schemes were presented and tested, some of them were derived from other popular Riemann solvers redefined under the PVM formulation, and some of them were newly proposed. In our previous paper [25] we chose a PVM redefinition of the Roe scheme (PVM-Roe) that is based on all four eigenvalues, and here we choose a scheme called PVM-2U that is based only on two external eigenvalues. In [10], PVM-2U proved to be the most efficient throughout their numerical tests and is, therefore, selected here for further analysis and comparison regarding the effects of the eigenvalue solvers.

The viscosity matrix of PVM-2U may be computed as [10]:

$$\mathbf{Q}_{i+1/2} = \alpha_0 \mathbf{Id} + \alpha_1 \mathbf{A}_{i+1/2} + \alpha_2 \mathbf{A}_{i+1/2}^2, \quad (33)$$

where coefficients α_k , $k = 0, 1, 2$ are derived from two external eigenvalues (for more details see [10]).

A similar scheme called Intermediate Field Capturing Parabola (IFCP) [21] was also derived from the family of PVM schemes. In contrast to PVM-2U, the IFCP scheme uses both internal and external eigenvalue information, and should be more accurate than PVM-2U with a minor increase in the computational cost. The IFCP scheme is also defined by Eq. (33), where α_k , $k = 0, 1, 2$ are derived from two external and one internal eigenvalues (for more details see [21]).

It should be emphasized that all three schemes considered in this study (Roe, PVM-2U, IFCP) are well balanced for water at rest solutions and linearly stable under the CFL condition [15, 10]:

$$\frac{\Delta t}{\Delta x} \max |\lambda_{j,i+1/2}| = \gamma \leq 1 \quad (34)$$

Regardless of a particular scheme, the eigenvalues may be computed either by a numerical decomposition, analytical (closed-form) or approximated expressions. In most published studies [21, 10], approximated expressions are recommended for both schemes from the PVM family when dealing with two-layer SWEs, as a more efficient choice in comparison to numerical eigensolvers. This study examines the benefits of using the analytical solutions instead.

2.4. Remarks about the implementation of the schemes

There are some modifications of numerical schemes that can be made to optimize the algorithms and improve their computational performance.

The first optimization deals with the Roe scheme and re-composition of the viscosity matrix given by Eq. (30). Although analytical closed-form solutions to \mathbf{K}^{-1} are available in the Appendix of [25], it is computationally faster to rewrite Eq. (30) as

$$\mathbf{K}_{i+1/2}^T |\mathcal{A}_{i+1/2}|^T = (\mathbf{K}_{i+1/2} |\mathbf{\Lambda}_{i+1/2}|)^T, \quad (35)$$

which corresponds to a general matrix equation $\mathbf{Ax} = \mathbf{B}$, solve it numerically for $|\mathcal{A}_{i+1/2}|^T$ (for example, by a LAPACK routine *gesv* [4]), and then transpose it. This is about 2-3 times faster than finding the inverse of $\mathbf{K}_{i+1/2}$ and performing matrix multiplication to obtain the viscosity matrix as written in Eq. (30).

A second optimization is available for the family of PVM schemes. Using Eq. (7) and the usual Roe linearization of the pseudo-Jacobian matrix:

$$\mathbf{J}(\mathbf{w}_{i+1}, \mathbf{w}_i) \cdot (\mathbf{w}_{i+1} - \mathbf{w}_i) = \mathbf{f}(\mathbf{w}_{i+1}) - \mathbf{f}(\mathbf{w}_i) \quad (36)$$

term $\mathbf{Q}_{i+1/2} (\mathbf{w}_{i+1}^n - \mathbf{w}_i^n)$ given in Eq. (28) may be replaced by:

$$\mathbf{C}_{i+1/2} [\mathbf{f}(\mathbf{w}_{i+1}^n) - \mathbf{f}(\mathbf{w}_i^n) - \mathbf{B}_{i+1/2} (\mathbf{w}_{i+1}^n - \mathbf{w}_i^n)] \quad (37)$$

where $\mathbf{C}_{i+1/2} = \mathbf{Q}_{i+1/2} \mathcal{A}_{i+1/2}^{-1}$ lowers the order of a viscosity matrix by one, and is already needed for the source term discretization. In other words, full viscosity matrix $\mathbf{Q}_{i+1/2}$ is not required for the family of PVM schemes; instead, only $\mathbf{C}_{i+1/2}$ is computed. For both PVM-2U and IFCP, $\mathbf{C}_{i+1/2}$ is defined as:

$$\mathbf{C}_{i+1/2} = \alpha_0 \mathcal{A}_{i+1/2}^{-1} + \alpha_1 \mathbf{Id} + \alpha_2 \mathcal{A}_{i+1/2}, \quad (38)$$

In this way, computation of the square of $\mathcal{A}_{i+1/2}$ (see Eq. (33)) is avoided.

3. Results

Several numerical tests are presented to evaluate the efficiency of Roe, IFCP, and PVM-2U schemes with different eigenvalue solvers. First, the accuracy and computational speed of numerical, analytical and approximated eigenvalue solvers is analysed. Next, the performance of numerical schemes in computing the numerical viscosity matrix is examined, as well as their sensitivity to the choice of eigenvalues. Finally, five numerical tests are given to analyse the overall efficiency of different numerical schemes in computing two-layer shallow-water flows.

In particular, three eigensolver algorithms are examined:

- N-Eig uses a numerical eigenvalue solver which decomposes a general square matrix into a diagonal matrix $\mathbf{\Lambda}$ whose elements are eigenvalues, and matrix \mathbf{K} whose columns are right eigenvectors. This algorithm is implemented in Python using Numpy function `numpy.linalg.eig` which is based on the `geev` LAPACK routines written in FORTRAN [4].
- A-Eig is an analytical eigenvalue solver based on a closed-form solution to the roots of the characteristic quartic given by Eqs. (18) and (19)
- E-Eig only estimates eigenvalues based on the approximations given by Eqs. (10) and (11)

By combining different eigensolvers with Roe, IFCP, and PVM-2U schemes, the following numerical algorithms for computing the viscosity matrix and solving two-layer SWEs are chosen for the efficiency analysis: N-Roe, A-Roe, E-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 schemes.

The accuracy of algorithms and schemes are evaluated by using either absolute error AE, relative bias error RBE, or root relative square error RRSE, defined respectively as:

$$AE_{\Phi} = |\Phi - \Phi^{ref}| \quad (39)$$

$$RBE_{\Phi} = \frac{\Phi - \Phi^{ref}}{\Phi^{ref}} \quad (40)$$

$$RRSE_{\Phi} = \frac{\sqrt{\sum_{n=1}^M [\Phi(x_n, t_{end}) - \Phi^{ref}(x_n, t_{end})]^2}}{\sqrt{\sum_{n=1}^M \Phi^{ref}(x_n, t_{end})^2}}, \quad (41)$$

where Φ is the evaluated parameter (e.g. eigenvalue λ , numerical flux f , depth h or velocity u), Φ^{ref} are the corresponding reference values, and M is a number of spatial points.

All numerical algorithms have been implemented in Python 3.6 and vectorized using the Numpy package. The tests have been performed on 64-bit Windows 10 machine with Intel Core i7-3770 3.4 GHz processor. All algorithms for computing the eigenstructure and viscosity matrix are freely available on Github [22].

3.1. Computing eigenvalues: Accuracy and computational speed

This test examines the accuracy and computational speed (CPU time) of corresponding three different algorithms for computing eigenvalues N-Eig, A-Eig, and E-Eig. Although eigenvalues computed by the numerical eigensolver are not *exact*, their errors are of the order of round-off errors. Therefore, the numerical results are used as a reference when evaluating the accuracy of analytical and approximated eigenvalues.

Since the main idea is to evaluate algorithms for solving eigenvalues as a integral part of Roe, IFCP and PVM-2U schemes for two-layer SWEs, physically realistic flow parameters (which always produce real eigenvalues) are chosen for this test. Therefore, a large set of parameters ($N = 10^6$) is randomly generated from a given range: $1.0 < h_{1,2} < 2.0$ m and $-0.3 < u_{1,2} < 0.3$ m s⁻¹. Different density ratios are selected, namely $r = 0.98, 0.9, 0.7, 0.5$, and 0.3 , with g set to 9.8 m s⁻².

Fig. 1 illustrates the statistical representation of the absolute errors of A-Eig and E-Eig algorithms computed by Eq. (39) for one million sets of independent flow parameters. The average error of the analytic solver is around 10^{-15} and the maximum errors are below 10^{-14} which is close to a round-off error of

the numerical eigensolver. In other words, both these solvers produce almost identical results. The average errors of the approximated expressions are more noticeable, and they increase with density difference. For $r = 0.98$, average errors are around 10^{-3} , whereas for $r = 0.3$, average errors are larger than 10^{-1} . This is expected, since eigenvalue approximations are derived under the assumption of $r \approx 1$ and $u_1 \approx u_2$, and become less accurate as parameters deviate from these assumptions.

To examine the lack of accuracy of the eigenvalues approximations, Fig. 2 illustrates the relative bias errors of individual eigenvalues for small r . It seems that the eigenvalue approximations always overestimate external eigenvalues, and underestimate the internal ones. Contrary to approximated expressions given by Eq. 10 that suggest how r only affects the internal eigenvalues, it is clear from this analysis that r equally affects external and internal eigenvalues.

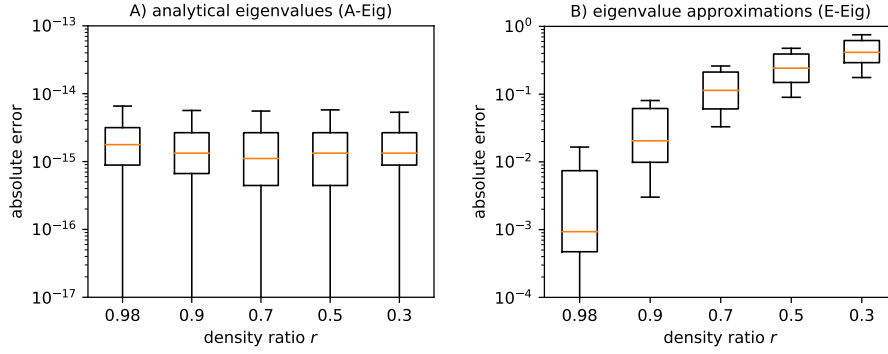


Figure 1: Boxplot of the absolute errors when computing eigenvalues by: A) the analytical closed-form solutions (A-Eig) and B) eigenvalue approximations (E-Eig). Boxes denote the interquartile range and median value, while whiskers denote min and max values.

Table 1 shows the computational speed of three eigenvalue algorithms. The fastest algorithm is the E-Eig (based on approximated eigenvalues) that needed 0.11 s, followed by the analytical solver A-Eig with 0.29 s, and finally the numerical solver N-Eig with 4.14 s. Notice that both approximated and closed-form eigenvalue solvers are one order of magnitude faster than the numerical solver. These results are in agreement with [25] and suggest that the prevailing opinion

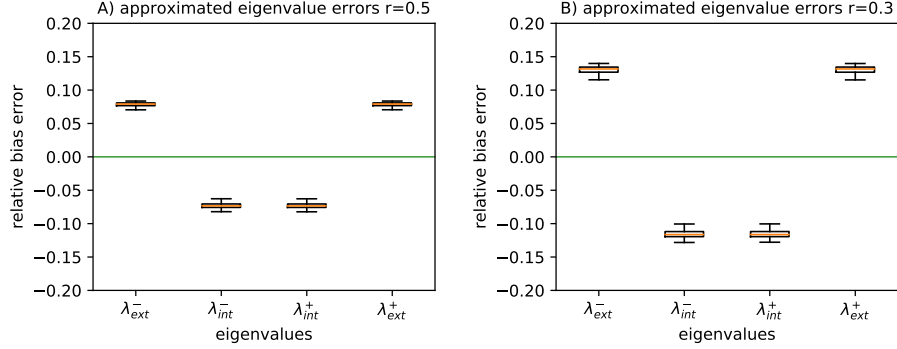


Figure 2: Boxplot of the relative bias errors when using approximated eigenvalues E-Eig for: A) $r = 0.5$ and B) $r = 0.3$. Boxes denote the interquartile range and median value, while whiskers denote min and max values.

in the scientific community about the "computational complexity" of the analytical solver is not justified. Although A-Eig needs double the time of E-Eig, it is considerably more accurate, as shown in Fig. 1.

Table 1: CPU times when computing one million sets of eigenvalues by numerical N-Eig, analytical A-Eig, and approximated E-Eig eigenvalue solvers (best of 5 runs).

Eigensolver	N-Eig	A-Eig	E-Eig
CPU time (ms)	4136.5	286.8	109.7

3.2. Computing numerical viscosity matrix: Accuracy and computational speed

The next test examines the accuracy and the computational speed of different implementations of Roe, IFCP, and PVM-2U schemes. The implementation differs in the choice of a particular eigenvalue algorithm, presented in the previous subsection. The main goal is to investigate how do eigenvalue algorithms affect the efficiency of Roe, IFCP, and PVM-2U schemes when computing the corresponding numerical viscosity matrix, and to analyse their sensitivity to a particular choice of the eigenvalue solver.

Same as in the previous example, physically realistic flow parameters are chosen. One million sets of parameters are randomly generated from a given

range: $1.0 < h_{1,2} < 2.0$ m and $-0.3 < u_{1,2} < 0.3$ m s⁻¹. Different density ratios are selected, namely $r = 0.98, 0.9, 0.7, 0.5$, and 0.3 , with g set to 9.8 m s⁻².

Note that Roe schemes compute \mathbf{Q} , whereas IFCP and PVM-2U schemes compute only \mathbf{C} , which is additionally multiplied by \mathbf{A} for comparison. The errors are estimated by a relative bias error given by Eq. (40), with N-Roe being the reference value.

Fig. 3 illustrates the statistical representation of relative bias errors for one million sets of independent flow parameters. Results obtained by A-Roe (Fig. 3a) and E-Roe (Fig. 3b) schemes correspond to the accuracy of the eigenvalue solvers they use. The average error of the A-Roe method is of the order 10^{-15} and the maximum errors are always below 10^{-14} . On the other hand, the average errors of the E-Roe scheme are much higher and increase with density difference. They amount to 10^{-2} for $r = 0.98$ and just under 1.0 for $r = 0.3$. Note that both positive and negative errors are observed, although in the case of E-Roe the positive errors (overestimation of the viscosity matrix) are more dominant.

The errors of IFCP and PVM-2U schemes are much higher than the A-Roe method, and comparable to E-Roe scheme. Between the two, PVM-2U (Fig. 3e,f) is one order of magnitude less accurate than the IFCP scheme (Fig. 3c,d). This emphasizes the importance of using both external and internal eigenvalue information. For A-IFCP schemes relative error is always under 10^{-2} , whereas for E-IFCP the relative error grows with decreasing r . It is interesting to note that E-IFCP, in contrast to E-Roe, underestimates the viscosity matrix. In PVM-2U scheme, the differences between the analytical and approximated eigenvalues are negligible, although the errors generated by approximated eigenvalues are somewhat higher for smaller values of r .

It is interesting to note that the errors for $r < 0.7$ are lower in IFCP and PVM-2U schemes (Fig. 3d,f) than in the E-Roe scheme (Fig. 3b). This is surprising, considering that they both use the same approximated eigenvalue solver, but the Roe scheme uses all eigenvalues, in contrast to IFCP which uses external and one internal eigenvalues, and PVM-2U which only uses minimum

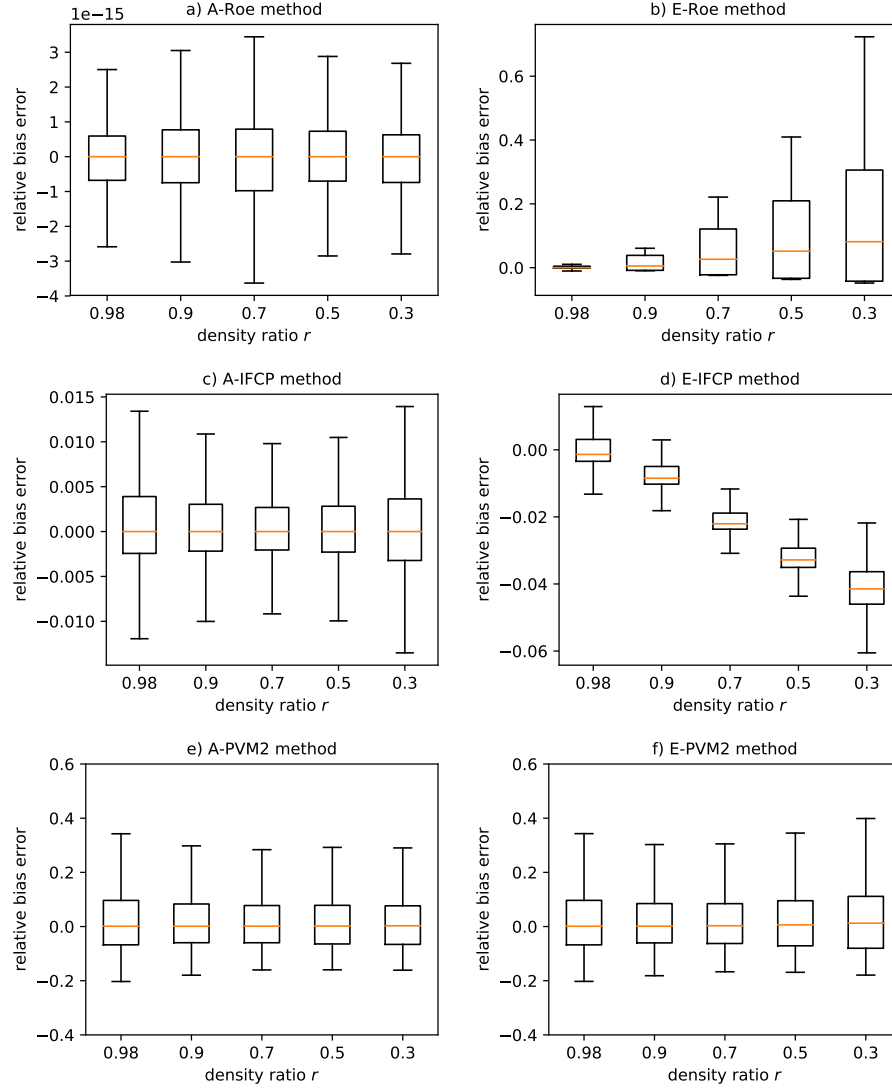


Figure 3: Boxplot of relative bias errors in computing one million numerical viscosity matrices by: a) A-Roe, b) E-Roe, c) A-IFCP, d) E-IFCP, e) A-PVM2, f) E-PVM2. Boxes denote the interquartile range and median value, while whiskers denote min and max values.

and maximum eigenvalues. It seems that the error produced by the E-Eig is somehow compensated by an approximation of the viscosity matrix in PVM-2U schemes. Considering the errors they produce, Roe schemes are very sensitive to the choice of eigenvalues, IFCP schemes are to some degree affected by the eigenvalues, whereas PVM-2U schemes are nearly unaffected by this choice.

The computational time of the schemes are shown in Tab. 2. As expected, the slowest scheme is N-Roe due to numerical eigen-decomposition. E-PVM2 and E-IFCP schemes are equally the fastest, followed by A-IFCP and A-PVM2. E-Roe and A-Roe are about 4-5 times faster than N-Roe, and about 35% slower than corresponding IFCP and PVM-2U schemes. When schemes based on A-Eig are computed, they are 0.14 - 0.19 s slower in comparison to the same schemes with E-Eig, which corresponds to the overhead due to analytical eigenstructure solver. Note that the computation of eigenvectors takes as much or even more time than the computation of eigenvalues. This is important because PVM-2U schemes do not require computation of eigenvectors, only eigenvalues.

The overall overhead from computing the analytical eigenvalues in comparison to approximated ones is only 8-13%, depending on the scheme. For N-Roe scheme, the eigendecomposition comprises over 80% of the total CPU time, whereas for A-Roe and E-Roe schemes, the computation of eigenvalues and eigenvectors comprises only 20-26% of the total CPU time. On the other hand, for IFCP and PVM-2U schemes, computation of approximated eigenvalues comprises about 8% of CPU time, and computational of closed-form eigenvalues about 18%. Therefore, the majority of time is used for computing viscosity matrices and not eigenvalues.

3.3. Numerical test I: Steady flow over smooth non-flat bed with zero flow rate

Since all considered schemes (N-Roe, A-Roe, E-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2) are theoretically stable only for exact eigenvalues [15, 21], the well-balance properties of these schemes with differently computed eigenvalues should be verified in practice. The first numerical test is designed to test the well-balanced properties of the schemes when simulating water at rest in a

Table 2: Total CPU time (including time required only for eigenvalues \mathbf{A} and eigenvectors \mathbf{K}) when computing one million numerical viscosity matrices by N-Roe, A-Roe, E-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2. (best of 5 runs)

Scheme	N-Roe	A-Roe	E-Roe	A-IFCP	E-IFCP	A-PVM2	E-PVM2
\mathbf{A} (s)	4.13	0.29	0.11	0.29	0.11	0.29	0.11
\mathbf{K} (s)	3.38	0.28	0.27	-	-	-	-
Total (s)	9.50	2.11	1.93	1.58	1.41	1.57	1.38

channel with smooth non-flat bed. To do so, the spatial domain is set to $[0, 1]$, and a bed function is defined by a single bump:

$$b(x) = \begin{cases} \frac{\cos(\pi(x - 0.5)/0.1) + 1}{4} \text{ m,} & \text{if } 0.4 < x < 0.6 \text{ m} \\ 0.0 \text{ m,} & \text{otherwise} \end{cases} \quad (42)$$

The initial condition is given by:

$$h_1(x, 0) = 0.4 \text{ m,} \quad h_2(x, 0) = 0.6 \text{ m} - b(x) \quad (43)$$

$$u_1(x, 0) = u_2(x, 0) = 0 \text{ m s}^{-1} \quad (44)$$

Non-reflective conditions are imposed at the boundaries, and the relative density ratio is set to $r = 0.98$. A single grid density of $\Delta x = 1/100$ m and a fixed time step $\Delta t = 0.002$ s is chosen.

Figure 4 shows the interface profile and the lower layer velocities obtained by N-Roe, A-Roe, E-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 numerical schemes at $t = 0.1$ s with $\Delta x = 1/100$ m. All schemes satisfy the well-balance properties for water at rest (the velocities are close to the order of computational precision), except E-Roe, where noticeable spurious oscillations appear at $0.4 < x < 0.6$ m. Similar results were obtained for other choices of r and Δx . Clearly, these oscillations appear because of eigenvalue approximations, and therefore the E-Roe scheme is excluded from further tests.

IFCP and PVM-2U schemes, however, are not prone to spurious oscillations, even when eigenvalues are approximated. Original studies that proposed these methods [21, 10] do not provide any theoretical proof of the stability of IFCP

and PVM-2U schemes for approximated eigenvalues, but they do confirm that no stability issues were observed in their extensive tests.

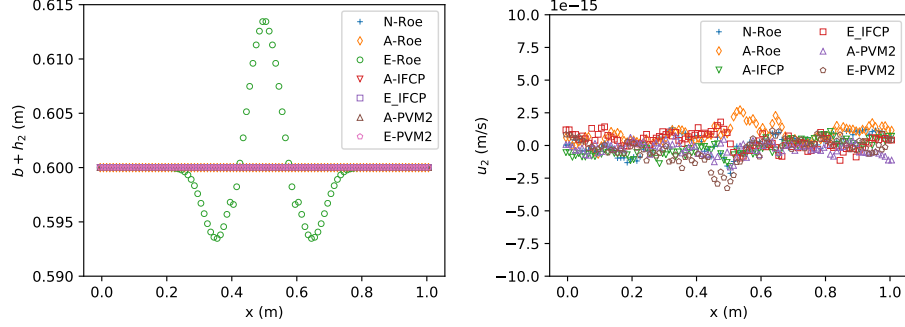


Figure 4: Test I: Computed interface and lower layer velocity obtained by N-Roe, A-Roe, E-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme at $t = 0.1$ s with $\Delta x = 1/100$ m

3.4. Numerical test II: Internal collision of two dam breaks with $r=0.98$

This test investigates the performance of numerical schemes when simulating a two-layer flow through a rectangular channel with a flat bottom topography. The flow structure is defined by two internal dam-breaks which eventually collide and produce a superimposed wave. A small density difference between the layers is considered, namely $r = 0.98$. The spatial domain is set to $[0, 100]$, and the initial condition is given by:

$$h_1(x, 0) = \begin{cases} 0.8 \text{ m,} & \text{if } 40 < x < 60 \text{ m} \\ 0.2 \text{ m,} & \text{otherwise} \end{cases} \quad h_2(x, 0) = 1.0 \text{ m} - h_1(x, 0) \quad (45)$$

$$u_1(x, 0) = u_2(x, 0) = 0 \text{ m s}^{-1} \quad (46)$$

Non-reflective conditions are imposed at the boundaries. Several mesh sizes are considered, namely $\Delta x = 1, 1/2, 1/4$, and $1/8$ m. A variable time step Δt is evaluated at each step to satisfy $CFL = 0.9$. The reference solution is computed using the A-Roe scheme and a dense grid with $\Delta x = 1/16$ m.

Figure 5 shows the temporal evolution of the interface and free-surface profiles for the reference solution. Solutions obtained by other methods are not

illustrated, because they are almost undistinguishable at this scale. Therefore, a detail of solutions for the interface depth and lower layer velocity are shown in Fig. 6 where N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 numerical schemes are compared against the reference solution at $t = 25$ s with $\Delta x = 1/2$ m. The results indicate that Roe and IFCP schemes are equally accurate, whereas PVM-2U schemes produce noticeably more diffused results. The implementation of the eigenvalue solver has no apparent influence on the accuracy of the results. This is expected, since approximated eigenvalues exhibit negligible errors for $r = 0.98$.

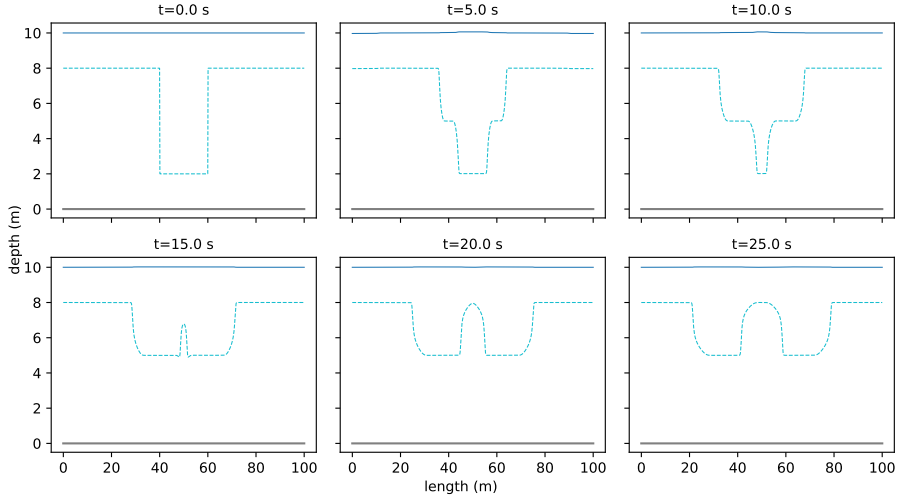


Figure 5: Test II: Temporal evolution of the interface and surface profile (the reference solution)

A CPU time vs. relative root square error E_Φ is presented in Fig. 7. The results suggest that the E-IFCP is the most efficient scheme, closely followed by A-IFCP and then A-Roe. In this test, the A-PVM2 and E-PVM2 are the least efficient schemes, even less than N-Roe. For all schemes, the differences in accuracy between the analytical and approximated eigenvalue solvers are insignificant. Approximated eigenvalues produce almost equal results as analytical ones, but need slightly less computational time. This can be explained by

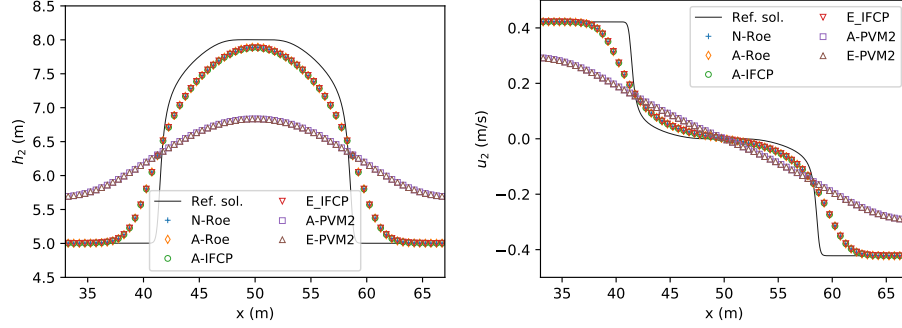


Figure 6: Test II: A detail of the interface depth and lower layer velocity obtained by N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution at $t = 25$ s with $\Delta x = 0.5$ m

satisfactory accuracy of approximated eigenvalues for r values close to one.

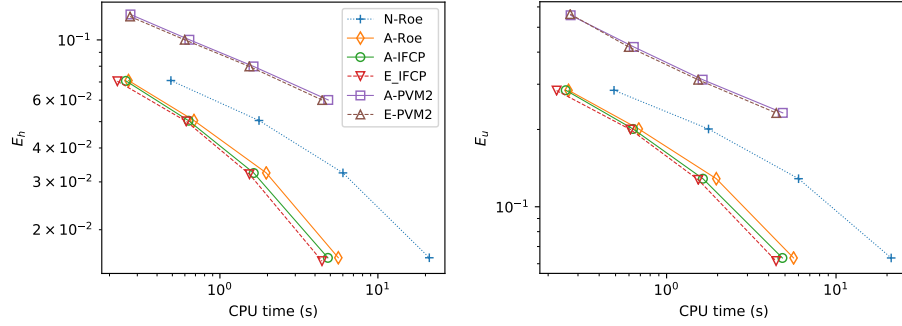


Figure 7: Test II: CPU time vs error for N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution

3.5. Numerical test III: Internal dam break over a sill with $r=0.98$

This test considers an internal dam-break over uneven bottom, which results in a transition from subcritical to supercritical flow. A small density difference between the layers is considered, namely $r = 0.98$. The spatial domain is set to $[0, 10]$, and the bottom topography is defined by a 0.5 m high sill located in the center of the channel, given by the function:

$$b(x) = 0.5 \exp(-(x-5)^2). \quad (47)$$

The initial condition is given by:

$$h_1(x, 0) = \begin{cases} 0.2 \text{ m}, & \text{if } x < 5 \text{ m} \\ 0.8 \text{ m}, & \text{otherwise} \end{cases} \quad h_2(x, 0) = 1.5 \text{ m} - b(x) - h_1(x, 0) \quad (48)$$

$$u_1(x, 0) = u_2(x, 0) = 0 \text{ m s}^{-1} \quad (49)$$

Non-reflective conditions are imposed at the boundaries. Several mesh sizes are considered, namely $\Delta x = 1/10, 1/20, 1/40$, and $1/80$ m. A variable time step Δt is evaluated at each step to satisfy $CFL = 0.9$. The reference solution is computed using the A-Roe scheme and a dense grid with $\Delta x = 1/160$ m.

Figure 8 shows the temporal evolution of the interface and free-surface profiles for the reference solution. A detail of solutions for the interface depth and lower layer velocity are shown in Fig. 9 where N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 numerical schemes are compared against the reference solution at $t = 15$ s with $\Delta x = 1/20$ m. The results indicate that Roe and IFCP schemes are equally accurate, whereas PVM-2U schemes produce noticeably more diffused results. Same as in the previous case, the implementation of the eigenvalue solver has no apparent influence on the accuracy of the results, since approximated eigenvalues exhibit negligible errors for $r = 0.98$.

A CPU time vs. relative root square error E_Φ is presented in Fig. 10. The results are very similar to the previous test case, shown in Fig. 7. The E-IFCP is the most efficient scheme, closely followed by A-IFCP and then A-Roe. The A-PVM2 and E-PVM2 are the least efficient schemes. The differences in efficiency between the analytical and approximated eigenvalue solvers are insignificant for IFCP and PVM-2U schemes.

3.6. Numerical test IV: Internal dam break with $r=0.4$

A two-layer flow through a rectangular channel with a flat bottom topography is considered again. In contrast to a similar internal dam-break scenario presented in [25], here we consider a much larger density difference between the layers, namely $r = 0.4$. The spatial domain is set to $[0, 50]$, and the initial

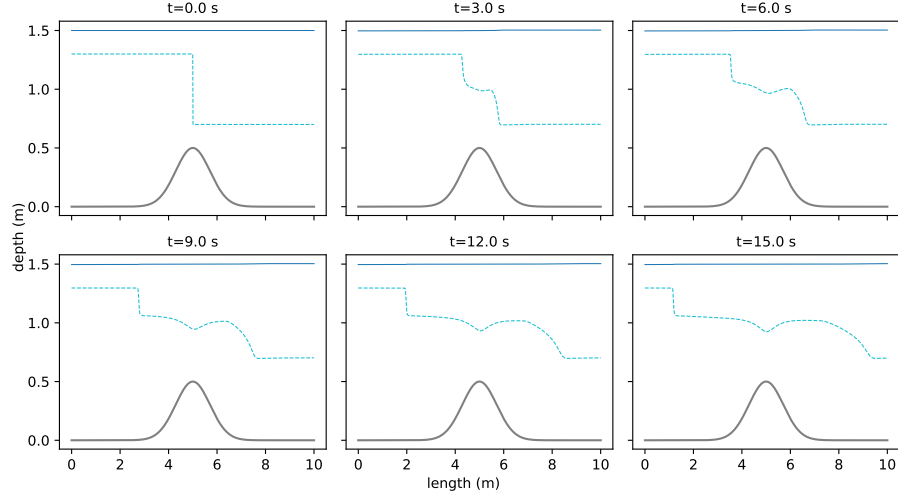


Figure 8: Test III: Temporal evolution of the interface and surface profile (the reference solution)

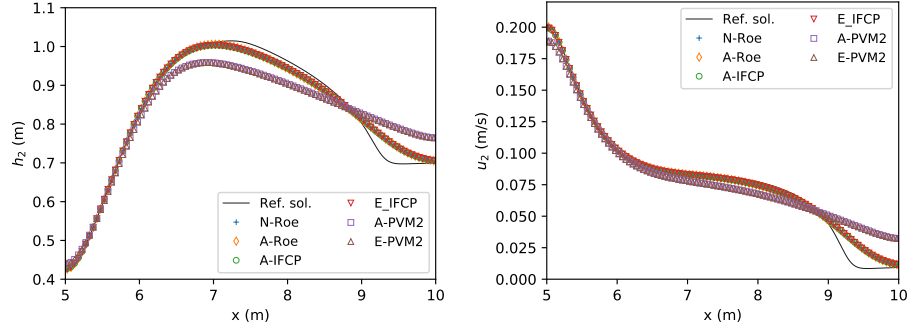


Figure 9: Test III: A detail of the interface depth and lower layer velocity obtained by N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution at $t = 15$ s with $\Delta x = 0.05$ m

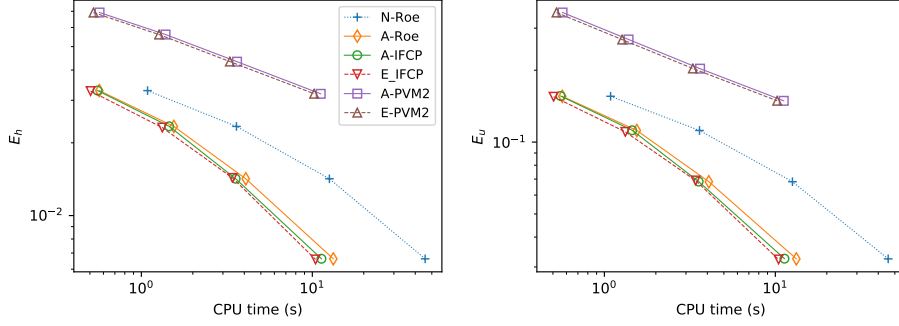


Figure 10: Test III: CPU time vs error for N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution

condition is given by:

$$h_1(x, 0) = \begin{cases} 0.2 \text{ m}, & \text{if } x < 25 \text{ m} \\ 0.8 \text{ m}, & \text{otherwise} \end{cases} \quad h_2(x, 0) = 1.0 \text{ m} - h_1(x, 0) \quad (50)$$

$$u_1(x, 0) = u_2(x, 0) = 0 \text{ m s}^{-1} \quad (51)$$

Non-reflective conditions are imposed at the boundaries. Several mesh sizes are considered, namely $\Delta x = 1/2, 1/4, 1/8$, and $1/16$ m. A variable time step Δt is evaluated at each step to satisfy $CFL = 0.9$. The reference solution is computed using the A-Roe scheme and a dense grid with $\Delta x = 1/32$ m.

Figure 11 shows the temporal evolution of the interface and free-surface profiles for the reference solution. A detail of solutions for the interface depth and lower layer velocity are shown in Fig. 12 where N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 numerical schemes are compared against the reference solution at $t = 5$ s with $\Delta x = 1/4$ m. The results indicate that all Roe and IFCP schemes are equally accurate, whereas PVM-2U schemes produce more diffused results. Analytical eigenvalues show slightly more accurate values in comparison to the approximated ones, which can be expected since the approximations are derived for $r \approx 1$, which is not the case here.

A CPU time vs. relative root square error E_Φ is presented in Fig. 13. In general, the results show that the A-IFCP, E-IFCP and A-Roe are the most effi-

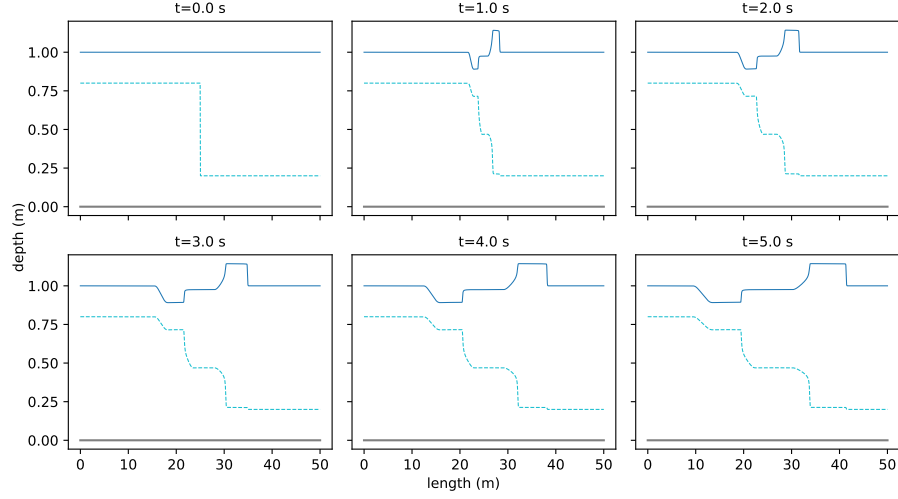


Figure 11: Test IV: Temporal evolution of the interface and surface profile (the reference solution)

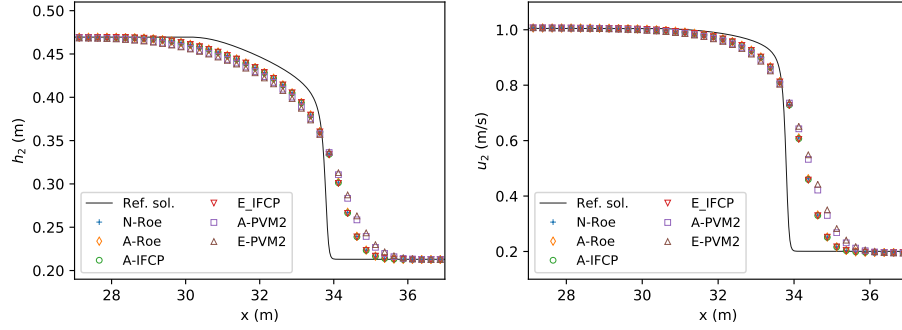


Figure 12: Test IV: A detail of the interface depth and lower layer velocity obtained by N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution at $t = 5$ s with $\Delta x = 0.25$ m

cient schemes. N-Roe and PVM-2U are noticeably less efficient. The differences in efficiency between the analytical and approximated eigenvalue solvers are not significant, but analytical eigensolvers seem to have a slight advantage over the approximated eigenvalues. Analytical eigenvalues produce more accurate results with a minor increase in the computational cost.

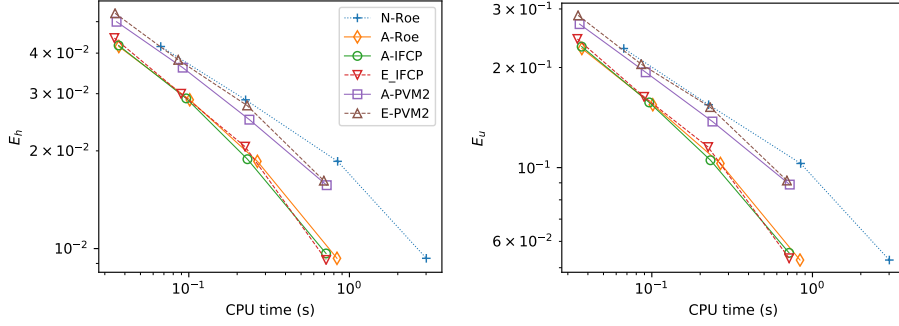


Figure 13: Test IV: CPU time vs error for N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution

3.7. Numerical test V: Internal column collapse on a slope with $r=0.4$

This test presents a lower-layer column collapse on a sloped bottom. Again a larger density difference between the layers is considered, namely $r = 0.4$. The spatial domain is set to $[0, 40]$, and the initial condition is given by:

$$h_1(x, 0) = \begin{cases} 0.2 \text{ m,} & \text{if } x < 25 \text{ m} \\ 0.8 \text{ m,} & \text{otherwise} \end{cases} \quad h_2(x, 0) = 1.0 \text{ m} - h_1(x, 0) \quad (52)$$

$$u_1(x, 0) = u_2(x, 0) = 0 \text{ m s}^{-1} \quad (53)$$

Non-reflective conditions are imposed at the boundaries. Several mesh sizes are considered, namely $\Delta x = 2/5, 1/5, 1/10$, and $1/20$ m. A variable time step Δt is evaluated at each step to satisfy $CFL = 0.9$. The reference solution is computed using the A-Roe scheme and a dense grid with $\Delta x = 1/40$ m.

Figure 14 shows the temporal evolution of the interface and free-surface profiles for the reference solution. A detail of solutions for the interface depth

and lower layer velocity are shown in Fig. 15 where N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 numerical schemes are compared against the reference solution at $t = 5$ s with $\Delta x = 1/5$ m. The results indicate that all Roe and IFCP schemes are equally accurate, whereas PVM-2U schemes produce more diffused results. As in the previous example, analytical eigenvalues show slightly more accurate values in comparison to the approximated ones.

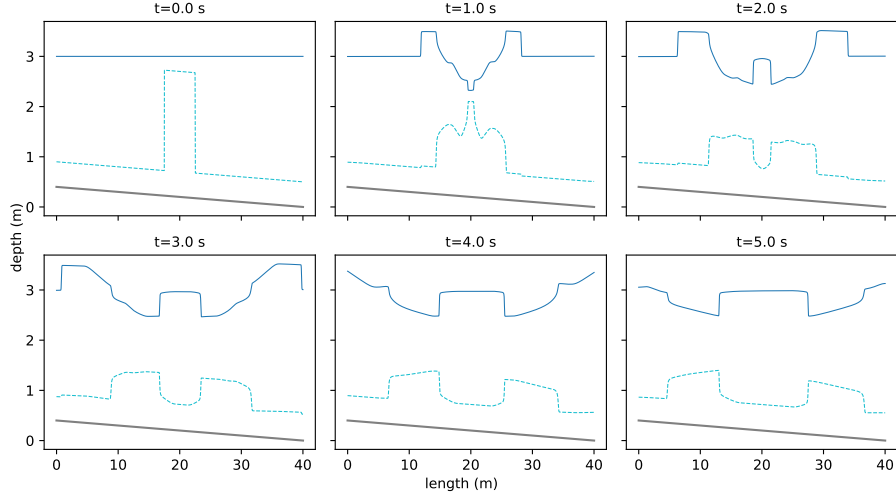


Figure 14: Test V: Temporal evolution of the interface and surface profile (the reference solution)

A CPU time vs. relative root square error E_Φ is presented in Fig. 13. Similar to the previous test, E-IFCP and A-IFCP are the most efficient schemes, closely followed by A-Roe scheme. Both PVM-2U schemes and N-Roe are noticeably less efficient. The differences in efficiency between the analytical and approximated eigenvalue solvers are not significant. For IFCP scheme they seem to be almost identical, but for PVM-2U scheme, the analytical implementation is more efficient.

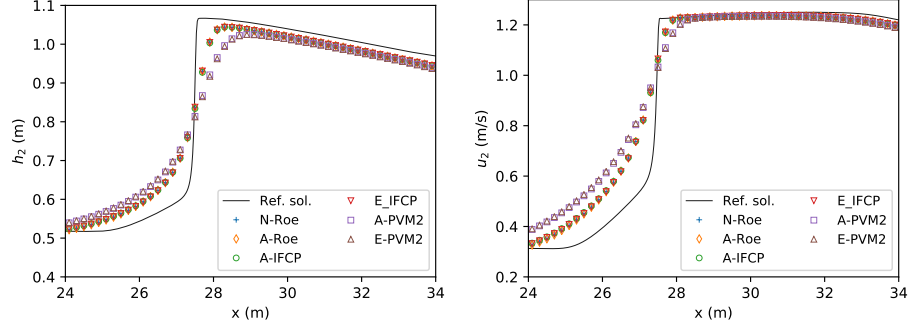


Figure 15: Test V: A detail of the interface depth and lower layer velocity obtained by N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution at $t = 5$ s with $\Delta x = 0.2$ m

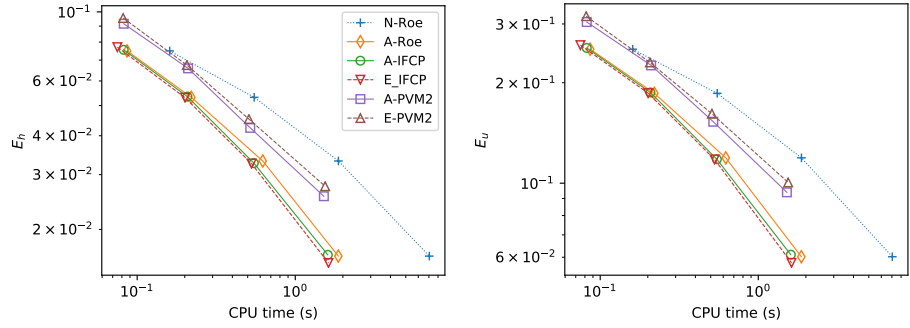


Figure 16: Test V: CPU time vs error for N-Roe, A-Roe, A-IFCP, E-IFCP, A-PVM2, and E-PVM2 scheme, compared to the reference solution

4. Discussion and Conclusion

This study re-evaluated the efficiency of Roe, IFCP, and PVM-2U schemes for two-layer shallow water systems with different solutions for eigenvalues. For this purpose, numerical, approximated, and recently-proposed analytical solutions for the eigenstructure were considered, combined with Roe, PVM-2U and IFCP numerical schemes. The choice of eigenvalues in numerical schemes for two-layer SWE models were evaluated in three stages.

First analysis focused only on the accuracy and computational time of different eigenvalue solvers. Numerical and analytical eigenvalue solvers produce almost identical results. Approximated expressions, on the other hand, are less accurate, and the errors grow with density difference between the layers. This increase in errors appears because approximate solutions deviate from the initial assumption that the density ratio is close to one. Regarding the computational time, analytical and approximated expressions are one order of magnitude faster than the numerical solver, with approximate expressions being two times faster than analytical ones. This additionally confirms our previous study [25] where analytical expressions were found to be equally accurate but one order of magnitude faster than numerical solvers (approximated expressions were not considered in the previous study).

The second analysis shifted the focus from eigensolvers to the numerical viscosity matrix, which was computed using different numerical schemes in combination with different eigenvalue solvers. The results revealed that the Roe method is highly sensitive to the choice of eigenvalues, IFCP method is somewhat sensitive, and PVM-2U method shows very little sensitivity to the choice of eigenvalue. In general, Roe method is the most accurate, followed by IFCP, and then PVM-2U method. This is expected since IFCP approximates the viscosity matrix using four eigenvalues, and PVM-2U using only two eigenvalues. Also, all schemes are more accurate when analytical eigenvalues are used in comparison to approximated values. It is interesting that the Roe method with approximated eigenvalues is the least accurate method overall for large density

differences. The computational time needed to obtain the numerical viscosity matrix is the longest for Roe scheme with numerical eigensolver. Other combinations of numerical schemes and eigensolvers are several times faster, and very close to each other. The rankings from the fastest to the slowest combination are: PVM-2U scheme with approximated eigenvalues, IFCP with approximated eigenvalues, PVM-2U with analytical eigenvalues, IFCP with analytical eigenvalues, Roe with approximated eigenvalues, and finally Roe with analytical eigenvalues. It is important to note that only 10-20% of total time needed to construct the viscosity matrix is spent on computing the eigenvalues.

In the third analysis several numerical tests were performed to investigate the overall performance of different numerical schemes in combination with different eigenvalue solvers. A total of five tests were designed to evaluate more realistic scenarios, including different density ratios between the layers and various channel geometries. The results revealed that the Roe scheme with approximated eigenvalues is not well-balanced, and should not be considered in two-layer modelling. Remaining four test showed that IFCP (with approximated and analytical eigenvalues) and Roe scheme (with analytical eigenvalues) are very close in performance, with IFCP being slightly better. PVM-2U was noticeably less efficient, regardless of the choice for the eigensolver.

These findings instil more confidence into findings from our previous study [25] that showed how Roe method with analytical eigenvalues is very close to IFCP scheme with approximated eigenvalues. Note that the previous study did not consider the modifications to the numerical schemes presented in Section 2.4, which may improve their computational speed. Also note that the conclusions from the previous study were based only on two numerical examples with the same (small) density difference between the layers and used a fixed time step. Furthermore, the previous study did not evaluate the impact of implementing analytical eigenvalues into PVM and IFCP schemes. Also, the PVM-2U is considered in the present study, whereas [25] evaluated PVM-Roe which is a redefinition of the Roe scheme under the PVM paradigm. PVM-2U, on the other hand, is a new PVM method based on two external eigenvalues.

Overall, when modelling layers of smaller density difference, it seems that approximated eigenvalues are a more efficient choice for IFCP and PVM-2U schemes. For larger density differences, however, the analytical eigenvalues are as efficient. It should be emphasized that analytical eigenvalues are more precise and can help with an accurate prediction of hyperbolicity losses for all density ratios, without producing any overhead in computational time.

The extension of all three considered schemes (Roe, IFCP, PVM-2U) to a higher order is straightforward, following a general approach presented in [13]. An extension to a two-dimensional case is also possible following the procedure from [11]. A high order extension of the family of PVM methods is similar to the Roe method [10]. For example, Castro et al. [9] presented the extension of IFCP method to solve a two-layer Savage-Hutter type model and simulate tsunamis generated by landslides. Since the eigenvalues affect only the viscosity matrix, the extension of each considered scheme to a higher order is the same regardless of the eigenvalue solver. On the other hand, some impact of the eigenvalues on the efficiency of higher order methods is expected and should be further analysed.

Although the efficiency of analytical solutions to the eigenstructure have been assessed here for two-layer shallow-water flows, these closed-form solutions to eigenvalues expressed in coefficient of a characteristic quartic, can directly be applied to some other non-conservative hyperbolic systems defined by four coupled partial differential equations, such as two-phase granular flows.

Acknowledgements

This work has been fully supported by the University of Rijeka under the project number 17.06.2.1.02 (River-Sea Interaction in the Context of Climate Change).

References

- [1] Abgrall, R., Karni, S., 2009. Two-layer shallow water system: a relaxation approach. *SIAM Journal on Scientific Computing* 31, 1603–1627. URL: <https://doi.org/10.1137/06067167X>.
- [2] Abramowitz, M., Stegun, I.A., 1965. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover Publications, New York.
- [3] Adduce, C., Sciortino, G., Proietti, S., 2011. Gravity currents produced by lock exchanges: Experiments and simulations with a two-layer shallow-water model with entrainment. *Journal of Hydraulic Engineering* 138, 111–121. URL: [https://doi.org/10.1061/\(ASCE\)HY.1943-7900.0000484](https://doi.org/10.1061/(ASCE)HY.1943-7900.0000484).
- [4] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D., 1999. *LAPACK Users' Guide*. Third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA.
- [5] Bermudez, A., Vazquez, M.E., 1994. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids* 23, 1049–1071. URL: [https://doi.org/10.1016/0045-7930\(94\)90004-3](https://doi.org/10.1016/0045-7930(94)90004-3).
- [6] Bouchut, F., Zeitlin, V., 2010. A robust well-balanced scheme for multi-layer shallow water equations. *Discrete and Continuous Dynamical Systems-Series B* 13, 739–758. URL: <http://dx.doi.org/10.3934/dcdsb.2010.13.739>.
- [7] Canestrelli, A., Fagherazzi, S., Lanzoni, S., 2012. A mass-conservative centered finite volume model for solving two-dimensional two-layer shallow water equations for fluid mud propagation over varying topography and dry areas. *Advances in Water Resources* 40, 54–70. URL: <https://doi.org/10.1016/j.advwatres.2012.01.009>.
- [8] Canestrelli, A., Toro, E.F., 2012. Restoration of the contact surface in FORCE-type centred schemes II: Non-conservative one-and two-layer two-dimensional shallow water equations. *Advances in Water Resources* 47, 76–87. URL: <https://doi.org/10.1016/j.advwatres.2012.03.018>.

- [9] Castro, M., de la Asunción, M., Macías, J., Parés, C., Fernández-Nieto, E., González-Vida, J., de Luna, T.M., 2012. Ifcp riemann solver: Application to tsunami modelling using gpus. *Numerical Methods for Hyperbolic Equations: Theory and Applications* , 237–244.
- [10] Castro, M.J., Fernández-Nieto, E.D., 2012. A class of computationally fast first order finite volume solvers: PVM methods. *SIAM Journal on Scientific Computing* 34, A2173–A2196. URL: <https://dx.doi.org/10.1137/100795280>.
- [11] Castro, M.J., Fernández-Nieto, E.D., Ferreiro, A., García-Rodríguez, J.A., Parés, C., 2009. High order extensions of roe schemes for two-dimensional nonconservative hyperbolic systems. *Journal of Scientific Computing* 39, 67–114.
- [12] Castro, M.J., Fernández-Nieto, E.D., González-Vida, J.M., Parés-Madronal, C., 2011. Numerical treatment of the loss of hyperbolicity of the two-layer shallow-water system. *Journal of Scientific Computing* 48, 16–40. URL: <https://doi.org/10.1007/s10915-010-9427-5>.
- [13] Castro, M.J., Gallardo, J., Parés, C., 2006. High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. applications to shallow-water systems. *Mathematics of computation* 75, 1103–1134.
- [14] Castro, M.J., Garcia-Rodriguez, J.A., González-Vida, J.M., Macías, J., Parés, C., Vázquez-Cendón, M.E., 2004. Numerical simulation of two-layer shallow water flows through channels with irregular geometry. *Journal of Computational Physics* 195, 202–235. URL: <https://doi.org/10.1016/j.jcp.2003.08.035>.
- [15] Castro, M.J., Macías, J., Parés, C., 2001. A Q-scheme for a class of systems of coupled conservation laws with source term. Application to a two-layer 1-D shallow water system. *ESAIM: Mathematical Modelling and Numerical Analysis* 35, 107–127. URL: <https://doi.org/10.1051/m2an:2001108>.
- [16] Castro, M.J., Pardo, A., Parés, C., Toro, E., 2010. On some fast well-balanced first order solvers for nonconservative systems. *Mathematics of Computation* 79, 1427–1472. URL: <http://doi.org/10.1090/S0025-5718-09-02317-5>.

- [17] Chakir, M., Ouazar, D., Taik, A., 2009. Roe scheme for two-layer shallow water equations: Application to the Strait of Gibraltar. *Mathematical Modelling of Natural Phenomena* 4, 114–127. URL: <https://doi.org/10.1051/mmnp/20094508>.
- [18] Chiapolino, A., Saurel, R., 2018. Models and methods for two-layer shallow water flows. *Journal of Computational Physics* 371, 1043–1066. URL: <https://doi.org/10.1016/j.jcp.2018.05.034>.
- [19] Doyle, E.E., Hogg, A.J., Mader, H.M., 2011. A two-layer approach to modelling the transformation of dilute pyroclastic currents into dense pyroclastic flows. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 467, 1348–1371. URL: <http://dx.doi.org/10.1098/rspa.2010.0402>.
- [20] Fernández-Nieto, E.D., Bouchut, F., Bresch, D., Castro, M.J., Mangeney, A., 2008. A new Savage–Hutter type model for submarine avalanches and generated tsunami. *Journal of Computational Physics* 227, 7720–7754. URL: <https://doi.org/10.1016/j.jcp.2008.04.039>.
- [21] Fernández-Nieto, E.D., Castro, M.J., Parés, C., 2011. On an intermediate field capturing Riemann solver based on a parabolic viscosity matrix for the two-layer shallow water system. *Journal of Scientific Computing* 48, 117–140. URL: <https://doi.org/10.1007/s10915-011-9465-7>.
- [22] Krvavica, N., 2019. Q-2L-SWE, Viscosity matrix for two-layer shallow water equations. URL: <https://github.com/NKrvavica/Q-2L-SWE>. [Online; accessed 25-February-2019].
- [23] Krvavica, N., Kožar, I., Travaš, V., Ožanić, N., 2017a. Numerical modelling of two-layer shallow water flow in microtidal salt-wedge estuaries: Finite volume solver and field validation. *Journal of Hydrology and Hydromechanics* 65, 49–59. URL: <https://doi.org/10.1515/johh-2016-0039>.
- [24] Krvavica, N., Travaš, V., Ožanić, N., 2017b. Salt-Wedge Response to Variable River Flow and Sea-Level Rise in the Microtidal Rječina River Estuary, Croatia. *Journal of Coastal Research* 33, 802–814. URL: <https://doi.org/10.2112/JCOASTRES-D-16-00053.1>.

- [25] Krvavica, N., Tuhtan, M., Jelenić, G., 2018. Analytical implementation of roe solver for two-layer shallow water equations with accurate treatment for loss of hyperbolicity. *Advances in Water Resources* 122, 187–205. URL: <https://doi.org/10.1016/j.advwatres.2018.10.017>.
- [26] Kurganov, A., Petrova, G., 2009. Central-upwind schemes for two-layer shallow water equations. *SIAM Journal on Scientific Computing* 31, 1742–1773. URL: <https://doi.org/10.1137/080719091>.
- [27] La Rocca, M., Adduce, C., Sciortino, G., Pinzon, A.B., Boniforti, M.A., 2012. A two-layer, shallow-water model for 3D gravity currents. *Journal of Hydraulic Research* 50, 208–217. URL: <https://doi.org/10.1080/00221686.2012.667680>.
- [28] Lee, W.K., Borthwick, A.G., Taylor, P.H., 2011. A fast adaptive quadtree scheme for a two-layer shallow water model. *Journal of Computational Physics* 230, 4848–4870. URL: <https://doi.org/10.1016/j.jcp.2011.03.007>.
- [29] Luca, I., Hutter, K., Kuo, C., Tai, Y., 2009. Two-layer models for shallow avalanche flows over arbitrary variable topography. *International Journal of Advances in Engineering Sciences and Applied Mathematics* 1, 99–121. URL: <https://doi.org/10.1007/s12572-010-0006-7>.
- [30] Majd, M.S., Sanders, B.F., 2014. The LHLLC scheme for two-layer and two-phase transcritical flows over a mobile bed with avalanching, wetting and drying. *Advances in water resources* 67, 16–31. URL: <https://doi.org/10.1016/j.advwatres.2014.02.002>.
- [31] Murillo, J., García-Navarro, P., 2010. An Exner-based coupled model for two-dimensional transient flow over erodible bed. *Journal of Computational Physics* 229, 8704–8732. URL: <https://doi.org/10.1016/j.jcp.2010.08.006>.
- [32] Ovsyannikov, L., 1979. Two-layer “shallow water” model. *Journal of Applied Mechanics and Technical Physics* 20, 127–135.
- [33] Parés, C., 2006. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis* 44, 300–321. URL: <https://doi.org/10.1137/050628052>.

- [34] Parés, C., Castro, M.J., 2004. On the well-balance property of roe’s method for nonconservative hyperbolic systems. applications to shallow-water systems. ESAIM: Mathematical Modelling and Numerical Analysis 38, 821–852. URL: <https://doi.org/10.1051/m2an:2004041>.
- [35] Pelanti, M., Bouchut, F., Mangeney, A., 2008. A Roe-type scheme for two-phase shallow granular flows over variable topography. ESAIM: Mathematical Modelling and Numerical Analysis 42, 851–885. URL: <https://doi.org/10.1051/m2an:2008029>.
- [36] Pitman, E.B., Le, L., 2005. A two-fluid model for avalanche and debris flows. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 363, 1573–1601. URL: <https://doi.org/10.1098/rsta.2005.1596>.
- [37] Sarno, L., Carravetta, A., Martino, R., Papa, M., Tai, Y.C., 2017. Some considerations on numerical schemes for treating hyperbolicity issues in two-layer models. Advances in Water Resources 100, 183–198. URL: <https://doi.org/10.1016/j.advwatres.2016.12.014>.
- [38] Schijf, J., Schönfeld, J., 1953. Theoretical considerations on the motion of salt and fresh water, in: Proceedings Minnesota International Hydraulic Convention, IAHR.
- [39] Toumi, I., 1992. A weak formulation of roe’s approximate riemann solver. Journal of computational physics 102, 360–373.
- [40] Wikipedia, 2018. Quartic function — wikipedia, the free encyclopedia. URL: https://upload.wikimedia.org/wikipedia/commons/9/99/Quartic_Formula.svg. [Online; accessed 19-February-2019].