The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

HIGH-ORDER WELL-BALANCED NUMERICAL SCHEMES FOR SHALLOW-WATER SYSTEMS WITH CORIOLIS TERMS

M. J. CASTRO-DÍAZ[1], J. A. GARCÍA-RODRÍGUEZ[2], V. GONZÁLEZ-TABERNERO[2], J. G. LÓPEZ-SALAS[2] AND J. PÉREZ-VILLARINO[2]

[1] Dpto. Análisis Matemático, Estadística e Investigación Operativa y Matemática Aplicada,
Universidad de Málaga,
Bulevar Louis Pasteur, 31, 29010 Málaga, Spain.
e-mail: mjcastro@uma.es

[2] Dpto. de matemáticas, CITIC y Facultad de Informática
Universidad de A Coruña, A Coruña, Spain
e-mail: jose.garcia.rodriguez@udc.es, v.gonzalez.tabernero@udc.es, jose.lsalas@udc.es, joel.perez.villarino@udc.es,

Key words: Hyperbolic Problems, Numerical Methods, Well-Balanced, Deep Learning

Summary. The goal of this work is to develop LDG high-order well-balanced schemes for the shallow-water equations with Coriolis terms. We will follow the strategy presented in [3,8] for building well-balanced methods by considering well-balanced reconstruction operators. The critical step of this procedure is solving a stationary PDE problem for each cell and time step. In this work, we combine local and global solvers to determine stationary solutions for the system. Once the local solution is obtained at the volume, it is extended to the whole considered stencil. We focus on the case where it is not possible to obtain the exact solutions for the stationary system. In this case, the exact stationary solution is replaced by a suitable approximation. Following [3], in this case, the semi-discrete finite volume numerical scheme is well-balanced, if the sequence of cell averages computed from the approximation is an equilibrium of the system of the ODE system given by the semi-discrete scheme. In this work, we propose techniques based on nonlinear optimization and Deep Learning to compute the stationary discrete solutions with the desired properties.

1 INTRODUCTION

The main objective of the current work is to develop LDG high-order well-balanced schemes for the two-dimensional shallow-water equations with Coriolis terms. The main novelty of this work is the development of numerical methods that allow the capture of high-order approximations of the stationary solutions for this system. The approximations will be computed with optimization processes and deep learning. To the best of our knowledge, this is the first time presented in the literature. This problem is fitted in the more general framework of high-order schemes for two-dimensional balance laws [12]

$$\frac{d}{dt}\mathbf{u} = -\nabla \mathbf{f}(\mathbf{u}) + \mathbf{S}(\mathbf{u}).$$

Where $\mathbf{u} \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^2$ is the solution, $\mathbf{f} \in C^1(\mathbb{R}^2, \mathbb{R}^2)$ is the physical flux, and $s \in C^1(\mathbb{R}^2, \mathbb{R}^2)$ is the source term.

We are interested in the study of the elliptic PDEs that describe the stationary solutions:

$$\nabla \mathbf{f}(\mathbf{u}^*)(\mathbf{x}) = \mathbf{S}(\mathbf{u}^*)(\mathbf{x}),$$

where $\mathbf{u}^*$ is the so-called stationary solution.

The main difficulty in this problem is the construction of the stationary solution of the system. This issue has been of great interest during the last decades, giving rise to a large number of publications [3, 5, 8]. In particular, the shallow-water system of equations is of great interest in this field due to its complexity [1, 4, 6].

Our approach consists in applying optimization processes to surpass the main complexities of these two-dimensional equations: uniqueness of solutions and solving a non-linear elliptic equation. Also, a new method is proposed using new Deep Learning techniques for this problem.

2 WELL-BALANCED LOCAL DISCONTINUOUS GALERKIN METHOD

2.1 The LDG method

The Local Discontinuous Galerkin (LDG) methods are used to obtain numerical solutions of balance laws. For the sake of simplicity, we will explain the method in the one-dimensional case.

First, we consider Finite Element-like mesh. It consists of a set of $N+1$ cells

$$\mathcal{M} = \{\Omega_i = [x_{i-1/2}, x_{i+1/2}) \mid i = 0, \dots, N\}.$$

On each cell, we consider a compact support basis $\phi_p^i \in P^k(\Omega_i)$ where P^k is a family of polynomials and ϕ_p^i stands for the p polynomial in the basis for the i-th cell.

We project the solution on that basis

$$\mathbf{u}(x,t) \approx \mathbf{u}_h(x,t) = \sum_{i,p} \mathbf{u}_{i,p}(t) \phi_p^i(x).$$

So, in each cell, we will have p values of $\mathbf{u}$. In particular, the polynomial basis considered are the Lagrange polynomials associated with the quadrature points x_p of Gauss-Legendre on each cell. This means, that $\mathbf{u}_{i,p}$ is the point value of the solution in the quadrature point x_p on the i-th cell.

The weak formulation of the balance law leads to a semi-implicit formulation by multiplying by a general test function of our basis

$$\begin{aligned} \frac{d}{dt}\mathbf{u}_{i,p}(t) \int_{\Omega_i} \phi_p^i(x)^2 = &\phi_p^i(x_{i-1/2})\mathbf{F}_{i-1/2} - \phi_p^i(x_{i+1/2})\mathbf{F}_{i+1/2} \\ &+ \int_{\Omega_i} \mathbf{F}(\mathbf{u}_h)\partial_x \phi_p^i(x) + \int_{\Omega_i} \mathbf{S}(\mathbf{u}_h)\phi_p^i(x). \end{aligned} \tag{1}$$

Here, the numerical fluxes $\mathbf{F}_{i+1/2}$ are computed by solving the Riemann problem at each intercell with a consistent numerical flux $\mathbf{F}_{i+1/2} = \mathrm{F}(\mathbf{u}_{h,i+1/2}^-, \mathbf{u}_{h,i+1/2}^+)$. We choose the HLL Riemann solver (see [11]). Also, the integrals are computed using the Gauss-Legendre quadrature formulas and the derivatives of the basis functions can be computed analytically. Finally, the time integration will be accomplished with a Strong Stability Preserving Runge-Kutta (SSP- RK) [9] with an accuracy according to the spatial integration order.

For our problem, we consider structured meshes for the 2d problems. Also, we will focus on two-point quadratures and linear polynomials for the cell basis.

2.2 Well-balanced schemes

The well-balancing procedure is introduced in numerical schemes to preserve the stationary solutions of the problem (see [3, 8]).

For hyperbolic problems, the LDG scheme must be able to preserve the stationary solutions described by

$$\nabla \mathbf{f}(\mathbf{u}^*)(\mathbf{x}) = \mathbf{S}(\mathbf{u}^*)(\mathbf{x}). \tag{2}$$

We introduce the following definitions

Definition 2.1. An LDG scheme (1) is said to be exactly well-balanced if it can preserve the exact stationary solutions (2). This means the analytical solutions.

Definition 2.2. An LDG scheme (1) is said to be well-balanced if it can preserve a high-order approximation of the exact stationary solutions (2). This approximation must be specified and consistent with the numerical method.

In both cases, the discrete numerical schemes will have stationary solutions to be preserved.

Following [8], the well-balanced procedure can be accomplished following the next steps for each cell Ω_i:

1. Compute the stationary solution on the ith-cell ($\mathbf{u}_i^*$):

$$\nabla \mathbf{f}(\mathbf{u}_i^*)(\mathbf{x}) = \mathbf{S}(\mathbf{u}_{\mathbf{i}}^*)(\mathbf{x}) \; x \in \Omega_i.$$

   Which is usually constrained with the preservation of the mean value and minimizes the L_2 norm on Ω_i.

2. Compute the DG fluctuations concerning the stationary solution in the cell

$$\mathbf{v}_{h,i}(x) = \mathbf{u}_h(x)|_{\Omega_i} - \mathbf{u}_i^*(x).$$

3. Redefinition of the numerical method (1)

$$\left(\int_{\Omega_i} \phi_p^i(x)^2\right) \frac{d}{dt}\mathbf{u}_{i,p}(t) = \phi_p^i(x_{i-1/2}) \left(\hat{\mathbf{F}}_{i-1/2} - \mathbf{F}^*_{i-1/2}\right) - \phi_p^i(x_{i+1/2}) \left(\hat{\mathbf{F}}_{i+1/2} - \mathbf{F}^*_{i+1/2}\right) + \int_{\Omega_i} \left(\mathbf{F}(\mathbf{u}_h) - \mathbf{F}(\mathbf{u}_i^*)\right) \partial_x \phi_p^i(x) + \int_{\Omega_i} \left(\mathbf{S}(\mathbf{u}_h) - \mathbf{S}(\mathbf{u}_i^*)\right) \phi_p^i(x).$$

   Where $\hat{\mathbf{F}}_{i+1/2} = \mathrm{F}\left(\mathbf{u}_i^*(x_{i+1/2}) + \mathbf{v}_{h,i}(x_{i+1/2}), \mathbf{u}_{i+1}^*(x_{i+1/2}) + \mathbf{v}_{h,i+1}(x_{i+1/2})\right)$ and $\mathbf{F}^*_{i-1/2} = \mathbf{F}(\mathbf{u}_i^*(x_{i+1/2}))$ is the evaluation of the physical flux with the stationary solution. This redefinition is valid since u_i^* is a stationary solution of the problem.

If the system is locally in a stationary state, on the i-th cell, the fluctuations $v_{h,i}(x) = 0$. If the system is globally in a stationary state, then it is locally stationary $\forall \Omega_i$, in this case, the stationary solution is preserved.

To preserve the stationary solution, the critical step lies in the first. We need to compute the stationary solution on the ith-cell only using local information.

2.3 The 2d shallow-water system with Coriolis forces

The shallow-water equations in two dimensions [1], are given by

$$\partial_t \begin{bmatrix} h \\ q_1 \\ q_2 \end{bmatrix} + \partial_x \begin{bmatrix} q_1 \\ \frac{q_1^2}{h} + \frac{gh^2}{2} \\ \frac{q_1 q_2}{h} \end{bmatrix} + \partial_y \begin{bmatrix} q_2 \\ \frac{q_1 q_2}{h} \\ \frac{q_2^2}{h} + \frac{gh^2}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ f q_2 \\ -f q_1 \end{bmatrix}.$$

Determining stationary solutions for this problem has been studied widely in the literature [4]. The main difficulties of the two-dimensional problem are the following:

1. Stationary solutions are determined by a first-order PDE system in two dimensions that needs to be solved locally.
2. A divergence-free equation is part of the system which complicates the resolution of the problem.
3. The PDE system is non-linear. In most cases, proving the uniqueness of the solution is challenging.

3 WELL-BALANCED PROCEDURE FOR THE 2D SHALLOW-WATER EQUATIONS WITH CORIOLIS FORCES

For this problem, we seek for a general solver that computes a general global solution of the stationary solutions system. The global solution is an approximation of the exact solution which converges as the number of cells grows. This will be a preprocessed initial condition for the LDG method.

Additionally, we develop a local solver which, using only information in the neighborhood of each cell, calculates the same solution as the global solver. So, in this case, the scheme is well-balanced as it preserves a high-order approximation of the stationary solution obtained from the global solver.

In 1d, initializing with the continuous solution, the local solver can determine the exact solution with the neighborhood information [6]. It is an **exactly well-balanced scheme**.

3.1 Global solver

We consider a Q_1 basis for the LDG solver. For the global stationary solution solver we consider the dual mesh of the LDG mesh. This is the set of vertices of each cell. On this dual mesh, the basis $\Psi_{I,J}(x,y) \in P_1 \times P_1$ has compact support on the cells containing the vertex I, J and $\Psi_{I,J}(x_P, y_Q) = \delta_{I,P}\delta_{J,Q}$ with P, Q being the set of vertices. This choice forces the global continuity of the solution.

However, to make computations cell-wise, we can define the coordinates of the vertex (x_m, y_n) on the cell (i,j) as (i,j,m,n). In this case, defining the base function $\psi_{i,j}^{k,l}(x,y)$ as the Lagrange polynomials on the cell (i,j) associated with the vertex (k,l) have compact support on the cell (i,j). This is related to the global basis as follows

$$\Psi_{I,J}(x,y) = \sum_{i,j,k,l} \psi_{i,j}^{k,l}(x,y) \tag{3}$$

for $(i,j,k,l) \in [(I,J)]$. Where $[(I,J)]$ is the equivalence relation of the nodes (i,j,k,l) that correspond to the same vertex.

We project the solution in this space:

$$\mathbf{u}(x) = \sum_{I,J} \mathbf{u}_{(I,J)} \Psi_{I,J}(\mathbf{x}).$$

With $\mathbf{u}_{(I,J)} = (u^0, u^1, u^2)_{(I,J)} = (h, q_1, q_2)_{(I,J)}$.

Now, we multiply the stationary system by a test function on this space and integrate over the domain, denoting, this last operation as $\langle \cdot, \cdot \rangle_\Omega$.

$$\begin{aligned}
&\langle \partial_x (q_1), \psi_{I,J} \rangle_\Omega &&+ \langle \partial_y (q_2), \psi_{I,J} \rangle_\Omega &&=0,\\
&\langle \partial_x \left(\frac{q_1^2}{h} + g\frac{h^2}{2}\right), \psi_{I,J} \rangle_\Omega &&+ \langle \partial_y \left(\frac{q_1 q_2}{h}\right), \psi_{I,J} \rangle_\Omega - \langle f q_2, \psi_{I,J} \rangle_\Omega &&=0,\\
&\langle \partial_x \left(\frac{q_1 q_2}{h}\right), \psi_{I,J} \rangle_\Omega &&+ \langle \partial_y \left(\frac{q_2^2}{h} + g\frac{h^2}{2}\right), \psi_{I,J} \rangle_\Omega - \langle -f q_1, \psi_{I,J} \rangle_\Omega &&=0.
\end{aligned}$$

However, taking into account (3) the previous equations can be written, in a more convenient way, as

$$\begin{aligned}
&\langle \partial_x (q_1), \psi_{i,j}^{k,l} \rangle_\Omega &&+ \langle \partial_y (q_2), \psi_{i,j}^{k,l} \rangle_\Omega &&=0,\\
&\langle \partial_x \left(\frac{q_1^2}{h} + g\frac{h^2}{2}\right), \psi_{i,j}^{k,l} \rangle_\Omega &&+ \langle \partial_y \left(\frac{q_1 q_2}{h}\right), \psi_{i,j}^{k,l} \rangle_\Omega - \langle f q_2, \psi_{i,j}^{k,l} \rangle_\Omega &&=0,\\
&\langle \partial_x \left(\frac{q_1 q_2}{h}\right), \psi_{i,j}^{k,l} \rangle_\Omega &&+ \langle \partial_y \left(\frac{q_2^2}{h} + g\frac{h^2}{2}\right), \psi_{i,j}^{k,l} \rangle_\Omega - \langle -f q_1, \psi_{i,j}^{k,l} \rangle_\Omega &&=0.
\end{aligned}$$

Which leads to a nonlinear system on each cell (i,j). We have used the property that each of the functions $\psi_{i,j}^{k,l}$ has compact support on cell (i,j), and $\psi_{I,J}$ is decomposed as a sum of these functions. This is useful for computing the loss function in the LDG mesh.

Now, for each cell, (i,j) we compute the Finite Element approximation for the vertex (m,n) according to these matrices, it is.

$$\begin{aligned}
PDE_{(0,i,j,m,n)} &= \sum_{k,l} -\langle \psi_{i,j}^{k,l}(\mathbf{x}), \partial_x \psi_{i,j}^{m,n}(\mathbf{x}) \rangle (q_1)_{(i,j,k,l)} - \langle \psi_{i,j}^{k,l}(\mathbf{x}), \partial_y \psi_{i,j}^{m,n}(\mathbf{x}) \rangle (q_2)_{(i,j,k,l)},\\
PDE_{(1,i,j,m,n)} &= \sum_{k,l} -\langle \psi_{i,j}^{k,l}(\mathbf{x}), \partial_x \psi_{i,j}^{m,n}(\mathbf{x}) \rangle \left(\frac{q_1^2}{h} + g\frac{h^2}{2}\right)_{(i,j,k,l)} - \langle \psi_{i,j}^{k,l}(\mathbf{x}), \partial_y \psi_{i,j}^{m,n}(\mathbf{x}) \rangle \left(\frac{q_1 q_2}{h}\right)_{(i,j,k,l)}\\
&\quad - \langle \psi_{i,j}^{k,l}(\mathbf{x}), \psi_{i,j}^{m,n}(\mathbf{x}) \rangle f\, (q_2)_{(i,j,k,l)},\\
PDE_{(2,i,j,m,n)} &= \sum_{k,l} -\langle \psi_{i,j}^{k,l}(\mathbf{x}), \partial_x \psi_{i,j}^{m,n}(\mathbf{x}) \rangle \left(\frac{q_1 q_2}{h}\right)_{(i,j,k,l)} - \langle \psi_{i,j}^{k,l}(\mathbf{x}), \partial_y \psi_{i,j}^{m,n}(\mathbf{x}) \rangle \left(\frac{q_2^2}{h} + g\frac{h^2}{2}\right)_{(i,j,k,l)}\\
&\quad - \langle \psi_{i,j}^{k,l}(\mathbf{x}), \psi_{i,j}^{m,n}(\mathbf{x}) \rangle f\, (-q_1)_{(i,j,k,l)},
\end{aligned}$$

In the case (i,j,m,n) is a boundary node we set the $PDE_{(a,i,j,m,n)}$ to the Dirichlet boundary condition. Finally, using (3) we can compute the $PDE_{(I,J)}$ terms for each vertex (I,J) of the mesh.

$$PDE_{(eq,I,J)} = \sum_{i,j,k,l} PDE_{(eq,i,j,k,l)} \; if \; (i,j,k,l) \in [(I,J)].$$

Notice that the $PDE_{(eq,I,J)}$ terms are the same as the variables $u_{(I,J)}$ in the LDG mesh. This is a centered approximation of the equations. The PDE terms an interior node will have contributions of (and contribute to) 9 vertices from 4 cells. A boundary node will contribute to the equations of 4 vertices from 2 cells and a corner node contribute to 2 vertices from 1 cell.

Once the PDE terms are computed, we assemble the loss function

$$F_{loss}(U) = \sum_{eq,I,J} \mathrm{PDE}^2_{(eq,I,J)}.$$

Noticing that an interior node P has contributions of 4 cells. When P is a boundary node PDE_P^k is a Dirichlet boundary condition based on the exact value on the boundary.

The solution of the global solver is obtained by minimizing the F_{loss} using its analytical ∇F_{loss}. The minimum value should be 0. The minimization process is performed with an L-BFGS solver initialized with the exact solution at each node. When the number of nodes is large enough the minimization stops at the starting value, granting convergence.

Once the solution of the global solver is computed, de vertices values are interpolated to the interior nodes of LDG and used as initial conditions for the LDG solver.

3.2 Local solver

For each cell (i,h), the local solver procedure is:

1. Build a 5×5 patch of the center cell and its neighbors (stencil).

2. Extrapolate the values to the dual mesh of the stencil. A mean of the extrapolation is used in each vertex. We obtain a representation of the global solution in a patch.

3. Compute $PDE_{(eq,I,J)}$ for each (I,J) being a vertex of the center cell. Compute $F_{loss} = \sum_{(eq,I,J)}(PDE_{(eq,I,J)})^2$. Notice that we only optimize the F_{loss} for the center cell values, the rest of the values of the stencil are fixed. This is, we only need the values of the PDE for the four vertices and the three equations, leading to a total of 12 terms.

    - If the local values are the same as the global solution calculated previously, F_{loss} is already at its minimum and it does not need optimization.
    - Otherwise, it optimizes for the vertices values.

4. Interpolate the vertices values to the center cell LDG nodes and use this projection as the local stationary solution.

If the optimization process returns the global stationary solution, then our scheme is well-balanced.

3.3 Numerical results

The initial condition considered is

$$\begin{aligned}
h(x,y) =& 1 - \frac{1}{g}\left(\frac{f\varepsilon}{2b} exp(-br^2) + \frac{\varepsilon^2}{4b} exp(-2br^2)\right),\\
\Psi(x,y) =& \varepsilon exp(-br^2),\\
q_1(x,y) =& -y\Psi(x,y),\\
q_2(x,y) =& x\Psi(x,y),
\end{aligned}$$

for $r = \sqrt{x^2+y^2}$, which is a radial solution of the stationary problem. The values are $\varepsilon = 0.05$, $f = g = 1$ and $b = 100$. $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$.

First, in table 1 we have the L_2 norm errors between the global solver results and the exact stationary solution. We can observe that when the number of nodes grows the global solver solution coincides with the exact solution. This error is computed acording to:

$$L_2(N) = \int_\Omega (\mathbf{u}_N - \mathbf{u}^*)^2 d\Omega,$$

where $\mathbf{u}^*$ is the exact solution and $\mathbf{u}_N$ is the global solver approximation.

In figure 3.3 we represent the error for the water depth between the preprocessed initial condition and the numerical solution of the well-balanced scheme for a time $T = 5$ under the previous parameters. The errors committed are of the order of the machine errors concluding that our scheme is well-balanced for this family of preprocessed initial conditions.

N	$L_2(N)$
10	1.1e-4
20	4.1e-6
40	3.1e-7
60	0
80	0

Table 1: L_2 norm errors of the global solver against the exact solution. N^2 is the total number of cells in the mesh.

4 CALCULATING THE STATIONARY SOLUTION WITH DEEP LEARNING

The ongoing work consists in developing a PINN (Physical Informed Neural Network) which approximates the stationary solutions. This is an extension of the previous work. The goal is to develop a more efficient and faster local solver which does not require to perform the optimization on each time step. We will use a feed-forward network. A map that transforms an input $y \in \mathbb{R}^{\hat{d}}$ into an output $z \in \mathbb{R}^m$ using the composition of a variable number, L, of vector-valued functions called layers. These consist of units (neurons), which are the composition of affine-linear maps with scalar non-linear activation functions, [7]. Thus, assuming a L-layer network with β_l neurons per layer, it admits the representation

$$h(y;\theta) := h_L(\,\cdot\,,\theta^L) \circ h_{L-1}(\,\cdot\,,\theta^{L-1}) \circ \cdots \circ h_1(\,\cdot\,,\theta^1)(y), \tag{4}$$

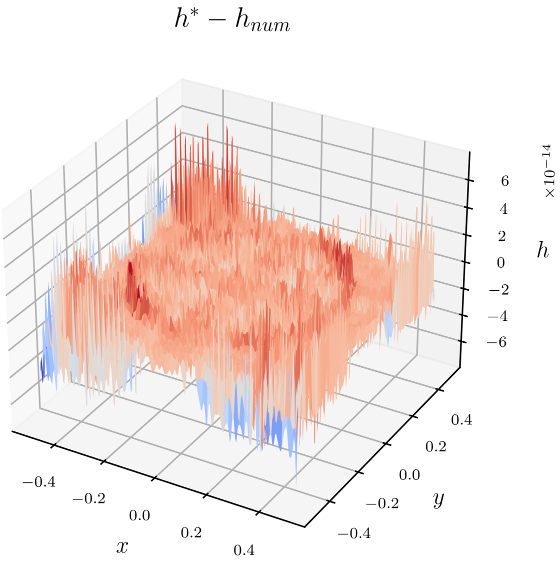

Figure 1: Point-wise difference between the preprocessed initial condition and the numerical solution for the variable h at a time $T = 5$.

where, for any $1 \leq l \leq L$,

$$h_l(z_l;\theta^l) = \sigma_l(W_l z_l + b_l), \quad W_l \in \mathbb{R}^{\beta_{l+1}\times\beta_l}, z_l \in \mathbb{R}^{\beta_l}, b_l \in \mathbb{R}^{\beta_{l+1}}, \tag{5}$$

with $z_1 = y$, $\beta_1 = \hat{d}$ and $\beta_L = m$. Usually (and this is taken as a guideline in this paper) the activation functions are assumed to be the same in all layers except in the last one, where we consider the identity map, $\sigma_L(\,\cdot\,) = Id(\,\cdot\,)$. In addition, taking into account the nature of the problem, it is required that the neural network fulfills the differentiability conditions imposed by the stationary solutions (2), in the unsupervised task, requiring sufficiently smooth activation functions such as the sigmoid or the hyperbolic tangent. The weights of the linear maps are updated with a training procedure. This training consists on an optimization process where the network's weights are updated depending on a metric commonly known as loss function.

In our research, the global solution will be represented using a fully connected neural network, using *tanh* in the hidden layers and linear functions in the output layer as the activation functions, trained with a stochastic gradient-based descent algorithm. The input is the water depth and the energy at a given spatial point (x, y). The output would be the values of the stationary solution (h, q_1, q_2) at that point.

Locally, the stationary values are determined with the evaluation of the neural network. The problem is divided into two stages:

1. The PINN would be trained, initially, against the exact solution with different coefficients. This training set is unlimited. $Loss = \frac{1}{N}\sum_j (z_j - \hat{z}_j)^2 = MSE(z, \hat{z})$ for z being (h, q_1, q_2) from the exact solution and $\hat{z}$ being the proxy from the NN.

2. Finally, the PINN will be trained using the PDE information. $Loss = \sum_{(eq,I,J)}(PDE_{(eq,I,J)})^2$ as it was done in the previous example.

5 CONCLUSIONS

In this work, we can conclude that using optimization processes can surpass the unicity of the problem leading to a unique numerical solution close to the analytical. Also, we are able to preserve the stationary solutions of this family by using the same procedure in the local solver as the one used in the global solver.

However, this procedure has a high computational cost due to the optimization process on each cell at every time step. This is the motivation to use PINNs to substitute the optimization process by an evaluation of the neural network on each time step.

6 ACKNOWLEDGMENTS

The first author's research has been funded by FEDER and the Spanish Government through the coordinated Research project RTI2018-096064-B-C1. Also, this research has been partially funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR" through the grant PDC2022-133663-C21 and by MCIN/AEI/10.13039/50110001103 and by "ERDF A way of making Europe", by the "European Union" through the grant PID2022-137637NB-C21. The other authors' research has been funded by the Spanish MINECO under research project number PID2022-141058OB-I00 and by the grant ED431G 2019/01 of CITIC, funded by Consellería de Educación, Universidade e Formación Profesional of Xunta de Galicia and FEDER and CITIC. The latter as a center accredited for excellence within the Galician University System and a member of the CIGUS Network, receives subsidies from the Department of Education, Science, Universities, and Vocational Training of the Xunta de Galicia. Additionally, it is co-financed by the EU through the FEDER Galicia 2021-27 operational program (Ref. ED431G 2023/01).

Finally, the third author thanks Conselleria de Cultura, Educacion, FP y Uni. y de Conselleria de Economia, Ind. e Inno. for the predoctoral funding ED481A/2023-110.

REFERENCES

[1] Audusse, E., Dubos, V., Duran, A., Gaveau, N., Nasseri, Y., et al.. "Numerical approximation of the shallow water equations with Coriolis source term". ESAIM: Proceedings, 2021, 70, pp.31-44. ⟨10.1051/proc/202107003⟩.⟨hal-03182659v2⟩.

[2] Castro, M. J. , López, J. A., and Parés, C.. "Finite volume simulation of the geostrophic adjustment in a rotating shallow-water system. SIAM Journal on Scientific Computing", 31(1):444–477, 2008.

[3] Castro, M. J., and Parés, C.. "Well-balanced high-order finite volume methods for systems of balance laws". J Sci Comput, 82(48):939–973, 2020.

[4] Chertock, A., Dudzinski, M., Kurganov, A., et al. "Well-balanced schemes for the shallow water equations with Coriolis forces", Numer. Math. 138, 939–973 (2018).

[5] Gómez-Bueno, I., Castro, M. J., Parés, C., and Russo, G.. "Collocation methods for high-order well-balanced methods for systems of balance laws". Mathematics, 9(15), 2021.

[6] González Tabernero, V., Castro, M. J., García-Rodríguez, J. A.. "High-order well-balanced numerical schemes for one-dimensional shallow-water systems with Coriolis terms", App. Math. and Comp. 469, ISSN 0096-3003.

[7] Goodfellow, I., Bengio, Y., and Courville, A.. "Deep learning". MIT Press, 2016.

[8] Guerrero Fernández, E.; Escalante, C.; Castro Díaz, M.J. "Well-Balanced High-Order Discontinuous Galerkin Methods for Systems of Balance Laws". Mathematics 2022, 10, 15. https://doi.org/10.3390/math10010015.

[9] Isherwood, Leah and Grant, Zachary J. and Gottlieb, Sigal, "Strong Stability Preserving Integrating Factor Runge–Kutta Methods", SIAM Journal on Numerical Analysis, vol 56, N 6, pp. 3276-3307, 2018, 10.1137/17M1143290.

[10] Raissi, M., Perdikaris, P., and Karniadakis, G. E.. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". Journal of Computational Physics, 378:686–707, 2019.

[11] Toro, E. F.. "The HLL and HLLC Riemann Solvers. In: Riemann Solvers and Numerical Methods for Fluid Dynamics". Springer, Berlin, Heidelberg (1997).

[12] Toro, E. F.. "Riemann Solvers and Numerical Methods for Fluid Dynamics,. A Practical Introduction", Springer-Verlag Berlin Heidelberg, 2009, ISBN: 978-3-540-25202-3.