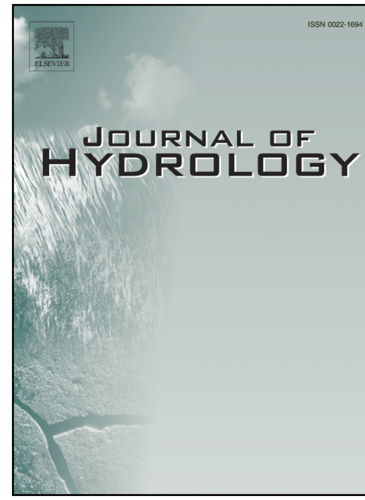
Accepted Manuscript



A 2D Parallel Diffusive Wave Model for floodplain inundation with variable time step (P-DWave)

J. Leandro, A.S. Chen, A. Schumann

PII: S0022-1694(14)00384-9
DOI: http://dx.doi.org/10.1016/j.jhydrol.2014.05.020
Reference: HYDROL 19619

To appear in: *Journal of Hydrology*

Received Date: 14 March 2014
Revised Date: 22 April 2014
Accepted Date: 5 May 2014

Please cite this article as: Leandro, J., Chen, A.S., Schumann, A., A 2D Parallel Diffusive Wave Model for floodplain inundation with variable time step (P-DWave), *Journal of Hydrology* (2014), doi: http://dx.doi.org/10.1016/j.jhydrol.2014.05.020

A 2D Parallel Diffusive Wave Model for floodplain inundation with variable time step (P-DWave)

Leandro J. [1,3], Chen, A. S. [2], and Schumann, A. [1]

[1] Institute of Hydrology, Water Management and Environmental Techniques, Ruhr-University Bochum, Germany. E-mail: Jorge.leandro@ruhr-uni-bochum.de, Andreas.Schumann@ruhr-uni-bochum.de

[2] Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK. E-mail: a.s.chen@exeter.ac.uk

[3] IMAR, Department of Civil Engineering , University of Coimbra, R. Luís Reis Santos, Polo 2, 3030-788 Coimbra, Portugal.

Abstract

Growing advances in remote sensing technologies together with the widespread availability of Digital terrain models (DTM) have intensified the research into two-dimensional (2D) models. Supported by detailed DTM, 2D models can become very accurate tools yet not without an added cost on the computational effort. Floodplain inundation is characterized by a slow varying phenomenon which can last hours, days or even weeks. In this paper we aim to develop a specific parallel diffusive wave model with variable time step suitable for flood inundation. Taking advantage of up to 12 processors, speed-up times ranging from 1.7 to 5.2 and 1.2 to 1.7 are achieved with the Matlab parallel computing toolbox and Fortran OpenMP Application Programming, respectively. The variable time method and the process devised to represent Wet-Dry fronts kept the solution stable and preserved absolute mass conservation. P-DWave performed well against known analytical solutions and dynamic and diffusive models in a total of seven tests.

Keywords: parallelization, diffusive wave model, Wet-Dry fronts, variable time step, absolute mass conservation

1. Introduction

The growing advances in remote sensing technologies led to the widespread use and development of two-dimensional (2D) inundation models (Bradbrook *et al.*, 2013). Digital terrain models (DTM) obtained from light detection and ranging (LiDAR) data enable large areas of terrain to be precisely characterised, therefore rendering almost unlimited accuracy possibilities in numerical modelling. Unpleasantly, the use of high resolution grids (i.e. 10 m or less) for large areas can lead to unacceptable run times for 2D Models simulations, while simultaneously demanding a more complex treatment of the model flow resistance (Dottori et al., 2013). This can nonetheless be improved by coarsening the DTM's resolution with penalty on the accuracy of the results. Herein we aim to develop a 2D Parallel Diffusive Wave Model (P-DWave) with variable time step to improve the computational efficiency of 2D diffusive wave models.

Except for catastrophic scenarios of dam break where the full dynamic equations must be applied, flooding over plain areas (often termed inundation) is characterized by a slow moving phenomena whereby the inundation can be modelled by the diffusive equations (Chen *et al.*, 2005). The diffusive wave simplification neglects the inertial terms allowing, therefore, a simplified set of equations to be solved. In general terms using a simplified set of equations leads to faster computational times, however, due to stability criterion some authors have verified that at coarser resolutions (10m or more) diffusive models were computationally less effective than dynamic models (Hunter et al., 2008; and Neal et al., 2012). In urban areas, Maksimovic and Prodanović (2001) suggest values between 1 and 2m and Mark et al. (2004) between 1 and 5m as optimal grid sizes to capture all the main topographic features.

Improvements on model performance have been the focus of recent research, particularly for explicit 2D diffusive wave models (2D DWM) it has been driven by two main reasons: the higher performance gain by explicit models because they use smaller time steps than their implicit counterparts due to stability considerations (Hirsch, 2007), and the fact that dynamic models require more complex numerical schemes than diffusive models (Prestininzi, 2008). Dottori & Todini (2011) enhanced the model performance of a 2D DWM by including an inertial formulation to compute discharges developed by Bates et al. (2010) and an adaptive time step developed by Zhang et al. (1994). Hunter et al. (2005) upgraded the raster-based cell model LISPFLOOD-FP using and adaptive time step (Bates and Roo, 2000). Liang and Borthwick (2009) improved a 2D full dynamic model by using dynamically adaptive grids. Burger and Rauch (2012) implemented a parallel version of the open-source implicit 1D full dynamic EPA SWMM model (Rossman, 2005). Despite published improvements (Chen *et al.*, 2012), the original overland flow models such as the explicit 2D DWM LISFLOOD-FP and SIRIPLAN have proven their usefulness particular for calibration purposes (Di Baldassarre *et al.*, 2009; and Horritt, 2006) or modelling of large scale sites (da Paz *et al.*, 2013).

If a meaningful cross-comparison of model's accuracy should use similar benchmark cases (Pender and Néelz, 2010), a meaningful cross-comparison of model performance by parallelization of algorithms ideally should use the same programming language and computational architecture as these may affect the overall performance. Judi et al. (2011) described reductions in computational time due to desktop parallelization varying from 14 to 300 times using a 16 core-processors computer while Ceyhan et al. (2007) reported 1.3 to 2 times for the same number of core-processors. Apart from the different languages and speed of processors used, computer acceleration times are limited by Amdahl's law i.e. the speedup is limited by the time needed for the sequential part of the program. It is thus clear that the former extraordinary increase was not only due to the parallelization but also due to changes in the algorithm (as rightfully acknowledged by the Authors). Yu (2010) and Neal et al. (2009) reported speedups in line with Amdahl's law ranging from 1.6 to 1.8 and from 3.4 to 5.2 using 2 and 8 core-processors respectively.

In this paper we develop a parallelized diffusive wave (P-DWave) model with an adaptive time step. The aim is to develop an accurate flood inundation model whose efficiency can be scalable through the use of multi-processor computer architecture. The Model is developed in Matlab^{TM} and

Fortran™ languages and tested in an AMD Opteron™ Processor 6276 with 12 cores 2.3GHz CPU and 192GB of RAM available at the RUHR University of Bochum. Next section describes the Parallel Diffusive Wave Model (P-DWave)

2. Parallel Diffusive Wave Model (P-DWave)

2.1. The Diffusive Wave Model Equations

The 2D Shallow Water Equations (SWE) can be written in the conservative form as:

$$\frac{dh}{dt} + \nabla(\boldsymbol{u}h) = R \tag{1}$$

$$\frac{d\boldsymbol{u}}{dt} + (\boldsymbol{u}\nabla)\boldsymbol{u} + \frac{\nu_t}{h}(h\nabla\boldsymbol{u}) + g\nabla(h+z) = g\boldsymbol{S_f} \tag{2}$$

h is the water depth, $\boldsymbol{u} = [u_x \quad u_y]^T$ is the depth-averaged flow velocity vector, g is the acceleration due to gravity, z is the bed elevation, ν_t is the turbulent eddy viscosity, R is the source/sink term (e.g. rainfall or inflow) and $\boldsymbol{S_f} = [S_{fx} \quad S_{fy}]^T$ is the bed friction vector.

Diffusive wave model neglects all the forces in the momentum equations except for the gravity term $g\nabla(h+z)$ and bed friction $\boldsymbol{S_f}$. The momentum equation Eq. (2) simplifies to:

$$g\nabla(h+z) = g\boldsymbol{S_f} \tag{3}$$

The bed friction can be approximated using Manning's formula:

$$\begin{bmatrix} S_{fx} \\ S_{fy} \end{bmatrix} = \begin{bmatrix} \dfrac{n^2|\boldsymbol{u}|u_x}{h^{4/3}} \\ \dfrac{n^2|\boldsymbol{u}|u_y}{h^{4/3}} \end{bmatrix} \tag{4}$$

$\nabla(h+z) = [S_{wx} \quad S_{wy}]^T$ is the water-level surface-gradient vector, where $S_{wx} = d(h+z)/dx$. The modulus of the depth-averaged flow velocity vector is given by:

$$|\boldsymbol{u}| = \frac{h^{2/3}\sqrt{I_m}}{n} \tag{5}$$

$$I_m{}^2 = S_{wx}{}^2 + S_{wy}{}^2 \tag{6}$$

2.2. Discretisation of the P-DWave and variable time step

The continuity equation Eq. (1) is solved using an explicit first order finite volume discretization on a regular grid. The spatial domain of P-DWave is discretised in cell-centered control volumes:

$$\frac{h_i^{t+1} - h_i^t}{\Delta t} + \frac{1}{A_i}\sum_{j=1}^{4} h_{ij}u_{ij}L_{ij} = R \tag{7}$$

For the sake of simplicity all variables without the time index are evaluated at the current time (t). A_i is the cell-area, L_{ij} is the contact face between cells, u_{ij} and h_{ij} are the water velocity and water-depth at each of the four cell faces evaluated as following:

$$h_{ij} = \frac{h_i + h_j}{2} \tag{8}$$

$$u_{ij} = \frac{h_{ij}{}^{4/3}}{n_{ij}{}^2\left|\boldsymbol{u}_{ij}\right|} I_{n,ij} \tag{9}$$

u_{ij} is the velocity in the direction perpendicular to each cell face. $I_{n,ij} = \left(S_{wx}\tilde{n}_x + S_{wy}\tilde{n}_y\right)_{ij}$ is the water-level surface-gradient vector multiplied with the face unit normal vector $\tilde{\boldsymbol{n}} = [\tilde{n}_x \quad \tilde{n}_y]^T$.

Central schemes are based on local fluxes estimations. The fluxes estimation shown in Eq. (7) requires a lower number of flux evaluations compared with other schemes because the fluxes are averaged at the faces according to Eq. (8) and (9).

Explicit schemes must have the time step limited in order to ensure stability. In order to study the stability of the proposed numerical scheme Eq. (7) is re-written as (for the sake of simplicity R will be set to zero):

$$h_i^{t+1} = h_i^t \left(1 - \frac{\Delta t}{2A_i} \sum_{j=1}^{4} a_{ij} \right) + \frac{\Delta t}{2A_i} \sum_{j=1}^{4} a_{ij} h_j^t \tag{10}$$

Whereby $a_{ij} = u_{ij} L_{ij}$, and after u_{ij} replacement:

$$a_{ij} = \frac{{h_{ij}}^{2/3}}{n_{ij}} \frac{I_{n,ij}}{\sqrt{I_{m,ij}}} L_{ij} \tag{11}$$

All coefficients in Eq. (10) must be positive in order to ensure that the scheme remains stable and monotonic:

$$1 - \frac{\Delta t}{2A_i} \sum_{i=1}^{4} a_{ij} > 0 \tag{12}$$

For a regular grid $A_i = \Delta x^2$ the final expression for the variable time step for the x direction can be obtained by replacing the water-level surface gradient vector in Eq. (11):

$$\Delta t < ArgMax \left(ArgMin \left(2\Delta x^2 n_{ij} \frac{\sqrt{S_{wx,ij}}}{{h_{ij}}^{5/3}} \right), \Delta t_{min} \right) \; for \; all \; i,j \tag{13}$$

A similar expression exists for the y direction. The minimum of both is taken as the final Δt. Eq. (13) is similar to the expressions found in other diffusive models (Hunter *et al.*, 2005; and Cea *et al.*, 2010), however herein the smallest allowable time step is twice of those models; the gain comes from the fluxes discretization at the faces (Eq. (8)). Comparing with dynamic models (SWE), the P-DWave time step is proportional to Δx^2 instead of Δx, that means that as long as Δx remains larger than 1 m the P-DWave should be more efficient than the SWE (i.e. having less computational time and increasing quadratically with the cell size). Δt_{min} is justified in order to avoid a too lengthily computational time when $\lim_{S_w \to 0}\left(2\Delta x^2 n \frac{\sqrt{S_w}}{h^{5/3}}\right) = 0$ and still avoid instabilities when the water is at near rest ($S_w \approx 0$). This solution is similar to the use of tolerance parameters as discussed by Hunter *et al.*(2005); and Cea *et al.* (2010); as such this value should also be decided based on the case study in order not to compromise accuracy. Despite the setting of a Δt_{min} the presented scheme is fully conservative up to machine precision as further discussed in the next sub-section

2.3. Process representation of Wet-Dry fronts

During a flooding event there will be inevitably initially dry cells that will switch to a wet state, whereas others will switch from wet to dry as the flood passes. This means that in many situations one must deal with a moving boundary condition. An often found solution when dealing with fixed computational meshes is the use of a depth-threshold or also called wet/dry parameter (Hubbard and Dodd, 2002). This procedure unfortunately adds/removes water to the global system that can either be redistributed to the surrounding cells (Nikolos and Delis, 2009) or negative mass-balance checks need to be incorporate to ensure mass conservation (Liang and Borthwick, 2009).

Herein a different approach is presented whereby a φ parameter is introduced into the P-DWave continuity equation Eq. (7):

$$\frac{h_i^{t+1} - h_i^t}{\Delta t} + \frac{1}{A_i} \sum_{j=1}^{4} \varphi_j h_{ij} u_{ij} L_{ij} = R \tag{14}$$

In Eq. (14) φ is always set to 1 unless the water depth in the next time step falls below zero ($h_i^{t+1} < 0$), in that case φ will take values between $0 < \varphi < 1$ in order to prevent the water-depths from becoming negative. The following condition is applied:

$$\varphi = \varphi_{1\ldots4} = \begin{cases} 1 & for \quad h_i^{t+1} > 0 \\ \dfrac{\Delta x^2}{\Delta t} \dfrac{h_i^t + R\Delta t}{\sum_{j=1}^{4} h_{ij} u_{ij} L_{ij}} & for \quad h_i^{t+1} < 0 \end{cases} \tag{15}$$

Eq. (15) allows the model to remain fully conservative up to machine precision. This is clear since φ parameter is updated for all 4 faces belonging to the cell ($\varphi_{1\ldots4}$) enabling neighbouring cells to use the corrected φ, avoiding negative water-depths and water gains or losses. Bradbrook et al. (2013) presented a similar approach but the scaling of the fluxes was still dependent on a minimum depth greater than 0 and absolute mass conservation was not attained.

2.4. Parallel implementation of the code

The code is implemented in both Matlab and Fortran environment. Fortan parallelization is achieved by implementing OpenMP Application Programming Interface (API) directives. OpenMP API is preferred to an MPI approach due to its ease of implementation with minimal changes to the non-parallel version. Regarding Matlab , vectorised operations are used whenever possible to improve modelling efficiency. Vectorised code is more efficient than the traditional do-loop iterations; however not all computing steps are able to be vectorised. In the latter case, we take advantage of modern multi-CPUs and multicores and adopted the built-in parallel computing toolbox in the Matlab to accelerate the computation.

The Matlab parallel computing toolbox provides several functions to use multicore processors, including parfor loop, GPU computing, spmd (single program multiple data), etc. When applying parfor loops, 2D arrays need to be sliced into multiple arrays such that each WORKER[1] can update the variable to the sliced array without causing problems in the shared memory. After each iteration the sliced arrays are gathered back into the original 2D array such that all WORKERs can compute with the correct updated array in the next iteration.

It should be noted that we also tested spmd approach (Matlab) on a multicore desktop but found the benefit to be limited on the shared memory computer. The application of spmd performs better on a distributed memory framework, which requires more attention on domain decomposition to ensure the optimum balance of load among computing nodes, and the minimum data to be synchronised. In the algorithm, the calculation of a cell requires information from its neighbour cells such that addition information of cells surrounding the decomposed domain is needed, which makes the domain decomposition a more complex task. Therefore, we leave the smpd implementation for a future stage when simulations with large scale data on distributed machines are required.

3. Model Testing: results and discussion

To assess the model performance seven tests are selected which enable studying specific flooding aspects and verify the model accuracy. The first and second tests were first presented in Hunter et al. (2005) and allow testing the model accuracy in propagating an inundation front. Third, fourth and fifth tests are taken from the Benchmark tests carried out by the UK Environmental Agency (EA) and allow a direct comparison with existing diffusive and dynamic models (Pender and Néelz, 2010).

[1] WORKER is the terminology used in Matlab to refer to a thread, i.e. the maximum number of processes that can be run simultaneously during a parallel session.

Computational times are additionally given to allow the reader to compare them with the various models ran in the EA. The sixth test was first presented by Wasantha Lal (1998). This test aims to quantitatively assess the model's accuracy in propagating an inundation front in a 2D space. In the original paper the Authors offered a way to obtain a solution against which numerical models can be compared to. In the seventh and last test we recover the Wasantha Lal (1998) test to verify the efficiency of the Parallel coding. Herein computational times are also disclosed.

3.1. Horizontal plane wetting

The horizontal plane wetting test performed in a horizontal 5km long rectangular channel (slope=0) aims to test the model accuracy in propagating an inundation front. Hunter et al. (2005) showed that by considering a constant inflow at the left boundary it is possible to simplify the SWE and obtain an analytical solution. All boundary conditions are defined as closed except for the left boundary. The left boundary condition is obtained by setting the horizontal coordinate x equal to 0 in the analytical solution. The final expression for the boundary condition is presented in Eq. (16) :

$$
\begin{aligned}
h_1^t &= \left[\frac{7}{3}\left(0.07 + n^2 u_1^3 t\right)\right]^{3/7} (m) \\
u_1^t &= 1\ (m/s)
\end{aligned}
\tag{16}
$$

Five different domains are defined in order to analyse the sensitivity of the model to the number of cells, CellNo={200, 100, 50, 25} (grid resolution of Δx={25, 50, 100, 200} m). Five corresponding smallest allowable time steps are defined to each domain Δt_{min}={0.001, 0.05, 0.5, 1.0} s such that the variable time step remains smooth and the solution free of instabilities. The Manning's coefficient=0.01 $m^{-1/3}/s$.

Figure 1 compares the evolution of the Water-surface level for eight time steps predicted by the Model and the analytical solution. The P-DWave solution produces a water level profile consistent with the analytical solution across all number of cells discretized CellNo={200, 100, 50, 25} (grid resolution of Δx={25, 50, 100, 200} m). Naturally as the number of cells decreases the ability to represent the curved stepped front is slightly impaired. Nonetheless the front location does not show signs of overshooting or delay. As discussed in **section 2.2** depending on the case study it may be required testing different Δt_{min} in order to find a solution with the wished level of accuracy. **Figure 2** shows a smooth evolution of the time step with the iteration number (it). For the finer cell resolutions CellNo={200, 100} (Δx={25, 50}) Eq. (13) controls the maximum allowable time step. For CellNo={50, 25}(Δx={100, 200}) a maximum time step of 1 sec is imposed to provide a detailed output of the solution. The latter has no effect on the accuracy of the model since 1 sec is smaller than the maximum allowable time step. Quantitatively, **Table 1** presents the Root Mean Square Error (RMSE) errors statistics of the Model solution compared with the analytical solution. The errors remain small across all solutions; the error exhibits similar behaviour and magnitudes to those obtained by Hunter et al. (2005).

3.2. Inundation “wetting and drying” of a planar beach S≠0

The inundation “wetting and drying” test of a planar beach with S≠0 allows testing the model ability to simulate advancing and receding of an inundation front. The test consists of a 5 km long channel with a slope of 0.001, whereby the left boundary condition is defined by a sinusoidal wave of amplitude 4m and 3000 sec period, as seen in Eq. (17).

$$h_1^t = 4\sin\left(t\frac{\pi}{3000}\right)\ (m) \tag{17}$$

$$u_1 = 1\ (m/s)$$

In order to analyse the sensitivity of the Model to Manning's coefficient, four different values are simulated n={0.01,0.02,0.04,0.08} $m^{-1/3}s$. The domain is discretised with 100 cells with Δx=50 m, and the smallest allowable time step Δt_{min} is equal to 0.001 s. **Figure 3** shows the evolution of the Water-surface level for eight time steps predicted by the Model for the four tested Manning values. Although there is no analytical solution, the shape and front propagation is intuitively correct as it shows a marked step front which is delayed with the increase of the Manning's coefficient and its behaviour is similar to the solution found in Hunter et al. (2005). This test case is nonetheless more demanding than the previous one because a nearly flat surface appears at x=0 m and becomes more pronounced before and after the receding phase (i.e. between t=1100 s and t=1900 s). The variable time step in **Figure 4** shows a controlled jerky oscillation indicating that the time step has reached the smallest allowable time step. As the Manning's coefficient is increased the required time step becomes larger than $\Delta \mathrm{t}_{\mathrm{min}}$ (as defined in Eq. (13)) and the oscillatory behaviour disappears.

In the absence of an analytical solution, and in order to decide an acceptable value for the smallest allowable time step, a sensitivity analysis of the Model to Δt_{min} is sought. Here four different Δt_{min}={0.001, 0.01, 0.1,1} s are compared with Δt_{min}=0.0005 s. **Table 2** shows the corresponding RMSE (m) error statistics for four instants in time. In this case the error is calculated assuming that the solution with Δt_{min}=0.0005 s is our true solution. It is clear that as Manning's coefficient reduces the Δt_{min} required becomes smaller; this is in line with Eq. (13). It is also noteworthy that the higher errors are found between t=1100 s and t=1900 s during the rising limb of the inundation front, clearly signalling that the smallest allowable time step has been reached and it should be decreased. For Δt_{min} smaller than 0.01 s the RMSE become negligible. Finally after the receding phase (or falling limb), the model recovers and reduces its RMSE. This rather surprising result can be partially explained by the wetting and dry treatment used herein; using Eq. (15) mass conservation is always ensured such that the actual volume of water within the model remains always correct. Depending on a favourable variation of the boundary conditions (such as in this test) it is possible that the model recovers to a state closer to the correct solution.

3.3. Flooding a disconnected water body test

This test is retrieved from the EA benchmarking test. It allows assessing the accuracy of the model to handle disconnected water bodies, and the wetting and drying of floodplains. The domain is defined by a rectangular channel of 100x700 m^2 with a Manning's coefficient=0.03 $m^{-1/3}s$ and discretised into CellNo=10x70 cells (Δx=10 m). Δt_{min} is set to 0.05 s and the maximum time step is set to 10 s. The left boundary condition is an inflow hydrograph specified by water levels in Eq.(18):

$$h_{ij}^{t}+z_{ij}^{t}=\begin{cases} 9.7 & for \quad t=0 \\ 9.7+\frac{0.65}{3600}t & for \quad t<3600 \\ 10.35 & for \quad t=[3600,36960] \\ 10.35-\frac{0.65}{6240}(t-36960) & for \quad t=[36960,43200] \\ 9.7 & for \quad t>43200 \end{cases} \tag{18}$$

The profile of the digital elevation model and the water-surface levels predicted by P-DWave at two specific points along the channel are presented in **Figure 5.** In addition the results from the various models in the EA are superimposed in order to enable easy accuracy comparison.

In terms of accuracy, P-DWave predicted the beginning of the flow in Point 1 starting at approximately one hour and reaching the maximum water level after approximately four hours. The water level rise and receding (starting at hour 12) are also predicted in good agreement with all other models (see e.g. ISIS 2D dynamic model ("Wallingford Software Ltd," 2006) or the UIM diffusive wave model (Chen *et al.*, 2005)). It is reasonable to conclude that the inertial terms could indeed be neglected as no obvious improvement in the results is seen by the dynamic models. In terms of computating time, P-DWave run is completed after 173 s (models' times in the EA report vary between 1 s and 349 s), with a total number of 1410303 computational time steps, the observed average time step is 0.051 s which indicated that the model is for the most of the computational time steps equal to Δt_{min} = 0.05 s.

3.4. Filling of floodplain depressions

This test aims to assess the model's ability to predict the inundation extent on a complex topography and to handle disconnected water bodies. The domain is defined by a squared area of 2000x2000 m^2 with a Manning's coefficient=0.03 $m^{-1/3}/s$ and discretised into CellNo=100x100 cells (Δx=20 m). The maximum time step is set to 10 s and Δt_{min} is set to 1.0 s The boundary condition is an inflow hydrograph (Q_{ij}^t) at the top left corner defined by Eq. (19):

$$Q_{ij}^t = \begin{cases} 0 & for \quad t = [0, 300] \\ 0 + \frac{20}{300}(t - 300) & for \quad t < 600 \\ 20 & for \quad t = [600, 5160] \\ 20 - \frac{20}{30}(t - 5160) & for \quad t = [5160, 5460] \\ 0 & for \quad t > 5460 \end{cases} \tag{19}$$

The final distribution of the flood inundation extent is consistent with that predicted by the full dynamic models used in the B-EA (**Figure 6**), as well as the filling up sequence and time as can be seen in the final water level points presented in **Figure 7**. The travels times in Points 4 and 2 are again consistent with all models, however some delay on the flood front can be observed in the points located further away from the inflow point (e.g. Point 10) as well as a slight overshoot of the flood peak in Point 4. It should be noted that while the overshoot is more noticeable in diffusive models's results (e.g. UIM; Chen *et al.*, 2005) similar delays can also been seen in the dynamic models' results (e.g. JFLOW+; Bradbrook, 2006). Point 9 is never inundated as expected. Overall, the results support that the diffusive equations are indeed sufficient to simulate this test case. P-DWave run takes 110 s to complete (EA models' times vary between 1 s and 1130 s), with 172224 computational time steps and an average time step of 1.001 s.

3.5. Rainfall and point source surface flow in urban areas

This test aims to assess the model's ability to simulate shallow inundation from a point source and from rainfall. The domain is defined by an area of 0.4x0.96 m^2 with a Manning's coefficient=0.03 $m^{-1/3}/s$ for roads and pavements, and 0.05 $m^{-1/3}/s$ elsewhere. The domain is discretised into CellNo=483x201 cells (Δx=2 m) (**Figure 8**). The maximum time step is set to 10 s and Δt_{min} is set to 0.03 s The point source boundary condition is an inflow hydrograph (please refer to Pender and Néelz (2010)) and a uniform rainfall of 400 mm/h with 4 min duration and starting at minute 1. Total simulation time is 5 h.

In terms of the final flood inundation extent the results are consistent with that predicted by the full dynamic models (**Figure 8**), although there are some differences in the maximum water levels reached which remain nonetheless within 0.1m (**Figure 9**). The differences are more obvious during the second flood peak caused by the inflow hydrograph in Points 1 and 2. As in the previous example similar behaviour is found in other diffusive models (e.g. UIM; Chen *et al.*, (2005) and RFSM; Jamieson et al., (2012)). Overall, the flood peak times and water levels are within the limits predicted by the models in EA. P-DWave run takes 10378 s to complete (EA models' times vary between 1 s and 18470 s), with 596795 computational time steps and an average time step of 0.0301 s.

3.6. Axisymmetric test

The Axisymmetric test allows testing the accuracy of the model to propagate an inundation front in a two-dimensional space (2D). The domain is defined by a squared area of 160.93x160.93 km^2 with a

Manning’s coefficient=1.0 $m^{-1/3}/s$ and discretised in CellNo=50x50 cells (Δx =3218.7 m). The initial condition is a smooth cosine function as defined by Eq. (20):

$$h_{ij}^{1} = \left[0.4575 + 0.1525\cos\left(\frac{\pi r_{ij}}{r_{max}}\right)\right](m) \quad for \quad r_{ij} \leq r_{max}$$
$$h_{ij}^{1} = 0.305\ (m) \qquad\qquad otherwise \tag{20}$$

In Eq. (20) r_{ij} is the distance of each grid point from the domain centre with a $r_{max} = 32.188\ m$. Due to its symmetry around the axis it is possible to derive an axisymmetric continuity equation for shallow water flows. The modified 1D diffusion equation can then be solved using a very fine grid with fix Δt =26s and compared with the 2D model as in Wasantha Lal (1998). The solution from this fine model is termed herein the Axisymmetric solution.

The water-surface level and velocity fields are shown in **Figure 10** for four time steps t={2, 3, 9, 12} (t is here represented in days (d)) predicted by the model for four different smallest allowable time steps Δt_{min}={100, 500, 2000, 8000} s and compared with the Axisymmetric solution. The model solution is in good agreement with the Axisymmetric solution with a slight smoothing of the solution for higher Δt_{min}; this can also be inferred by the RMSE error statistics in **Table 3** which exhibit smaller errors than the numerical solutions obtained by Wasantha Lal (1998). The velocity fields show that the solution remains symmetric in respect to both axis with time. Also noteworthy are the velocity low values of this particular test due to the high Manning’s coefficient used[2].

It is clear that Δt_{min} is the dominant restriction in Eq. (13). **Figure 11** shows that as Δt_{min} reduces, the variable time step is subsequently reduced. Despite the fact that the variable time step is often

[2] In the Author’s opinion, the Manning’s value is unrealistically high and can only be justified by the large Δx =3218.7 m which would then encompass the added roughness by houses, roads and other overland flow obstructions.

equal to Δt_{min} and not the one obtained through the stability analysis in Eq. (12), **Figures 10 and 11** show that it is possible that the model simulation still converges to the correct solution. There are nonetheless, some visible oscillations in the water level for the larger Δt_{min}=8000 s. This test will also be used in **Section 3.7** for testing the efficiency of the parallelization coding of the model.

3.7. Parallel performance test: Speed-up and efficiency

The final test has the objective of verifying the parallel performance. The test in **section 3.6** is here recovered because it uses a 2D mesh and it is easily scalable. **In section 3.6** a mesh with CellNo=50x50 cells was used, herein we will test four different CellNo={300x300, 500x500, 700x700, 900x900} ($\Delta x =${536.4, 321.9, 229.9, 178.8} m) with Δt_{min}=500 s. In order to compare and verify the parallel performance we raise the number of cells and therefore increase the computational effort by a quadratic exponent (see Eq. (14)). The two common metrics used in this paper are speed-up and efficiency (**Table 4**) and follow the notation by Yu (2010). Speed up is defined as the ratio between the single processor execution time and that of the multi-processor:

$$Sp(nc,P) = \frac{T_{Single}(nc)}{T_{multiple}(nc,P)} \tag{21}$$

In Eq. (21) $T_{Single}(nc)$ is the run time of the sequential algorithm, and $T_{multiple}(nc,P)$ is the run time of the parallel algorithm using P core-processors. Efficiency is defined by Eq. (22):

$$E(nc,P) = \frac{Sp(nc,P)}{\mathrm{P}} \tag{22}$$

The total run time is set to 1 hour since we are only focusing on the computational efficiency of the model. The tests were conducted on a workstation with AMD Opteron™ Processor 6276 with 12 cores 2.3GHz CPU and 192GB of RAM at RUHR University of Bochum.

It is interesting to notice that although Matlab computational times are larger than Fortran, Matlab speed-up performs better than Fortran. Two possible explanations could be the highly efficient Fortran code which sees smaller gains through parallelizing than Matlab or that a more complex Fortran MPI approach is required to increase the gains in speed-up closer to Matlab performance. In any case it is clear that the model developed is indeed scalable. It is also noteworthy that depending on the CellNo and the measure adopted to describe efficiency, the optimal use of number of processors might be different. Purely looking at the computational time it seems obvious that the maximal possible number of processers should always be selected; however once one focus on the speed-up, it becomes obvious that there is an improvement limit, simply because the communication costs between processors becomes too high (Yu, 2010). In that case Efficiency can be a simple way to decide on the number of processors to use. For example, if one selects a minimum efficiency of 0.75 and the Matlab code two processors would be the optimal choice for CellNo=300x300, four processors would only be worth it from CellNo=700x700 and for 12 processors a much larger CellNo would be necessary.

Lastly, it is worth mentioning that the speedup and efficiency obtained herein with the Matlab code exhibit a behaviour similar (and magnitudes) to the ones obtained by Yu (2010) and Neal, Fewtrell, & Trigg (2009). Future work will see the implementation of the MPI approach using Fortran.

4. Conclusion

In this paper we presented a parallelized two-dimensional diffusive wave model (P-DWave) with adaptive time step. The parallelization was achieved in the Matlab environment with the use of the parfor loop, and using computational vectorization whenever possible, while in Fortran it was achieved using OpenMP API. The model was validated in seven tests against known analytical solutions, and diffusive and dynamic models results from an EA benchmark report. The model converged regardless of the spatial resolution as long as the selected minimum step was not too limiting (this limit is found to be case study dependent), and showed sensitivity to the changes of Manning's roughness in a sloped planar beach. Symmetry was kept in the test case of a horizontal plane, and the model was proven robust even in the presence of strong irregular geometries. The process devised to represent Wet-Dry fronts was effective in keeping a sharp front, while the variable time step kept the solution stable and oscillations-free in all tests

The parallelization strategy was indeed effective, by improving the speed-up times from 1.7 to 5.1 and from 1.2 to 1.7 respectively for Matlab and Fortran, depending on the domain size and the number of processors used. The speed-up increases as the domain size becomes larger or the number of processors progressively increases. Efficiency follows a similar trend in relation to the domain size increase, but it can half its value as the number of processors change from 2 to 12 processors. Similarly to other Authors' results this is attributed to the communication costs between processors. Future work may see the parallelization of this same code in a different programming language or using another parallelization strategy to analyse the potential benefits, and inclusion of the dynamic terms in the code developed for a thorough discussion on the differences in computational run times.

5. Acknowledgments

The Authors would like to acknowledge the support of the DFG - Deutsche Forschungsgemeinschaft - trough Project GZ: LE 3220/1-1. Three anonymous reviewers are also acknowledged for their valuable comments and suggestions on an early version of this paper. The authors would also like to thank the UK Environment Agency for the EA benchmarks datasets.

6. References

Bates, P.D., Horritt, M.S., Fewtrell, T.J., 2010. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. J. Hydrol. 387, 33–45.

Bates, P.D., Roo, A.P.J. De, 2000. A simple raster-based model for flood inundation simulation. J. Hydrol. 236, 54–77.

Bradbrook, K., 2006. JFLOW: a multiscale two-dimensional dynamic flood model. Water Environ. J. 20, 79–86.

Bradbrook, K.F., Lane, S.N., Waller, S.G., Bates, P.D., Consulting, J.B.A., Barn, S., Hall, B., Yorkshire, N., 2013. Two dimensional diffusion wave modelling of flood inundation using a simplified channel representation. Int. J. River Basin Manag. Taylor 2 (3), 37–41.

Burger, G., Rauch, W., 2012. Parallel Computing in Urban Drainage Modeling : A Parallel Version of EPA SWMM. UDM 2012, Belgrade, Serbia 1–9.

Cea, L., Garrido, M., Puertas, J., 2010. Experimental validation of two-dimensional depth-averaged models for forecasting rainfall–runoff from precipitation data in urban areas. J. Hydrol. 382, 88–102.

Ceyhan, E., Ou, S., Estrade, B., Kosar, T., 2007. Towards a faster and improved ADCIRC (ADvanced Multi-Dimensional CIRCulation) model. J. Coast. Res. Special Is.

Chen, A.S., Evans, B., Djordjević, S., Savić, D.A., 2012. Multi-layered coarse grid modelling in 2D urban flood simulations. J. Hydrol. 470–471, 1–11.

Chen, A.S., Hsu, M.H., Chen, T.S., Chang, T.J., 2005. An integrated inundation model for highly developed urban areas. Water Sci. Technol. 51, 221–9.

Da Paz, A.R., Collischonn, W., Bravo, J.M., Bates, P.D., Baugh, C., 2013. The influence of vertical water balance on modelling Pantanal (Brazil) spatio-temporal inundation dynamics. Hydrol. Process.

Di Baldassarre, G., Schumann, G., Bates, P.D., 2009. A technique for the calibration of hydraulic models using uncertain satellite observations of flood extent. J. Hydrol. 367, 276–282.

Dottori, F., Baldassarre, D., Todini, E., 2013. Detailed data is welcome, but with a pinch of salt: accuracy, precision, and uncertainty in flood inundation modeling. Water Resour. Res. 49, 6079–6085.

Dottori, F., Todini, E., 2011. Developments of a flood inundation model based on the cellular automata approach: Testing different methods to improve model performance. Phys. Chem. Earth, Parts A/B/C 36, 266–280.

Hirsch, C., 2007. Numerical computation of internal and external flows. The fundamentals of computational fluid dynamics, second. ed. Elsevier, Butterworth-Heinemann.

Horritt, M.S., 2006. A methodology for the validation of uncertain flood inundation models. J. Hydrol. 326, 153–165.

Hubbard, M.E., Dodd, N., 2002. A 2D numerical model of wave run-up and overtopping. Coast. Eng. 47, 1–26.

Hunter, N.M., Horritt, M.S., Bates, P.D., Wilson, M.D., Werner, M.G.F., 2005. An adaptive time step solution for raster-based storage cell modelling of floodplain inundation. Adv. Water Resour. 28, 975–991.

Hunter, N.M., Villanueva, I., Pender, G., Lin, B., Mason, D.C., Falconer, R. a., Neelz, S., Crossley, a. J., Bates, P.D., Liang, D., Wright, N.G., Waller, S., 2008. Benchmarking 2D hydraulic models for urban flooding. Proc. ICE - Water Manag. 161, 13–30.

Jamieson, S.R., Wright, G., Lhomme, J., Gouldby, B.P., 2012. Validation of a computationally efficient 2D inundation model on multiple scales. FloodRisk 2012, Novemb. 2012, Rotterdam. 20–22.

Judi, D., Burian, S., McPherson, T., 2011. Two-dimensional fast-response flood modeling: desktop parallel computing and domain tracking. J. Comput. Civ. Eng. 25, 184–191.

Liang, Q., Borthwick, A.G.L., 2009. Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. Comput. Fluids 38, 221–234.

Maksimovic, C., Prodanović, D., 2001. Modelling of urban flooding—breakthrough or recycling of outdated concepts. Urban Drain. Model. 1–9.

Mark, O., Weesakul, S., Apirumanekul, C., Aroonnet, S.B., Djordjević, S., 2004. Potential and limitations of 1D modelling of urban flooding. J. Hydrol. 299, 284–299.

Neal, J., Fewtrell, T., Trigg, M., 2009. Parallelisation of storage cell flood models using OpenMP. Environ. Model. Softw. 24, 872–877.

Neal, J., Villanueva, I., Wright, N., Willis, T., Fewtrell, T., Bates, P., 2012. How much physical complexity is needed to model flood inundation? Hydrol. Process. 26, 2264–2282.

Nikolos, I.K., Delis, A.I., 2009. An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography. Comput. Methods Appl. Mech. Eng. 198, 3723–3750.

Pender, G., Néelz, S., 2010. Benchmarking of 2D hydraulic modelling packages. SC080035/R2 Environ. Agency Bristol 169 pp.

Prestininzi, P., 2008. Suitability of the diffusive model for dam break simulation: Application to a CADAM experiment. J. Hydrol. 361, 172–185.

Rossman, L.A., 2005. Storm water management model - user's manual Version 5.0. EPA - United States, EPA - United States, Cincinnati http://www.epa.gov/ednnrmrl/models/swmm/epaswmm5–m.

Wallingford Software Ltd, 2006. . ISIS Flow / Hydrol. http//www.wallingfordsoftware.com/products/isis/.

Wasantha Lal, A.M., 1998. Performance comparison of overland flow algorithms. J. Hydraul. Eng. 124, 342–349.

Yu, D., 2010. Parallelization of a two-dimensional flood inundation model based on domain decomposition. Environ. Model. Softw. 25, 935–945.

Zhang, X.D., Trepanier, J.-Y., Reggio, M., Camarero, R., 1994. Time-accurate local time stepping method based on flux updating. Fluid Mech. Heat Transf. 32, 1926–1928.

Figures

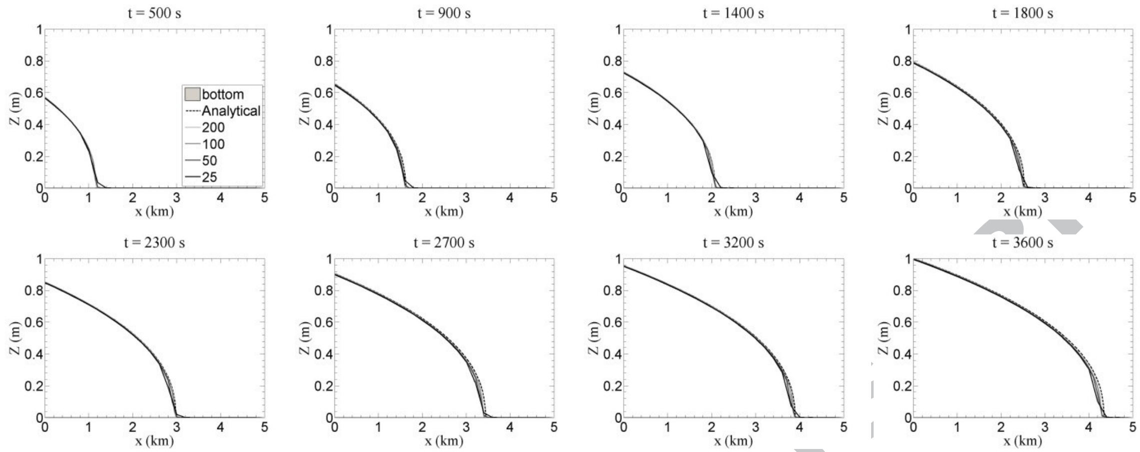


Figure 1. Water-surface level for the "horizontal plane wetting" test predicted by P-DWave and the analytical solution (dashed line). Sensitivity analysis to the number of cells CellNo={200, 100, 50, 25} (Δx={25, 50, 100, 200} m) with Δt_{min}={0.001, 0.05, 0.5, 1.0} s on the front propagation.

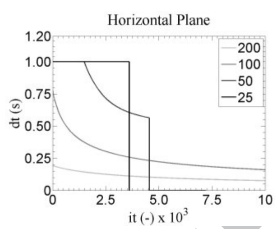


Figure 2. Evolution of the time step solution with the iteration number (it) as a function of the number of cells discretised CellNo={200, 100, 50, 25} (Δx={25, 50, 100, 200} m) with Δt_{min}={0.001, 0.05, 0.5, 1.0} s during the horizontal plane wetting simulation.

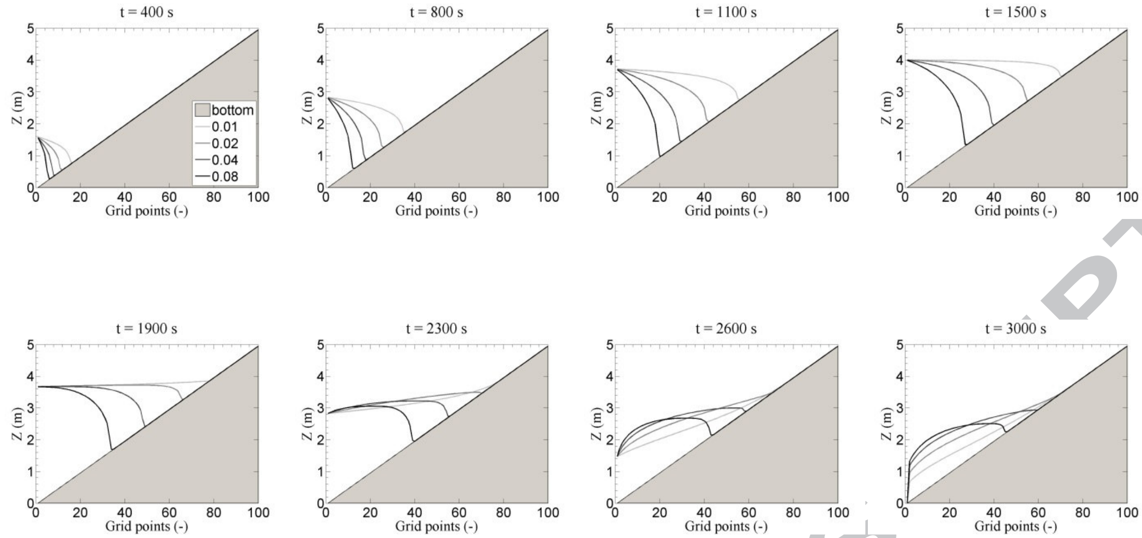


Figure 3. Sensitivity analysis of the predicted water-surface level propagation to Manning's coefficient for the "inundation of a planar beach" S≠0 test (Δx=50 m and Δt_{min}=0.001 s).

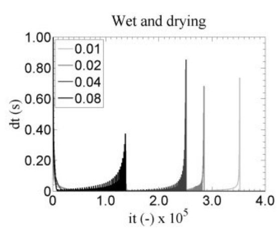


Figure 4. Evolution of the time step solution with the iteration number (it) as a function of Manning's coefficient n={0.01,0.02,0.04,0.08} $m^{-1/3}s$ and Δt_{min}=0.001 s during the Inundation of a planar beach S≠0 simulation.

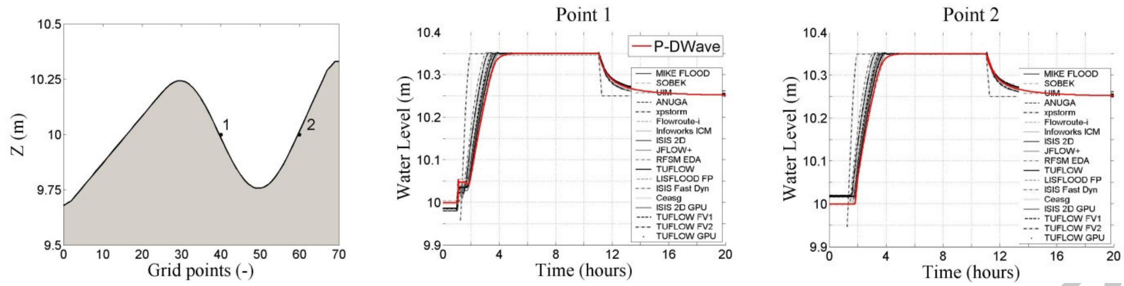


Figure 5. Profile of the digital elevation model (DEM) (left). Water-surface levels at points 1 and 2 for the "flooding a disconnected water body" test predicted by P-DWave and superimposed with the various models' results published in the EA benchmark (Pender and Néelz, 2010) test for comparison (middle and right).

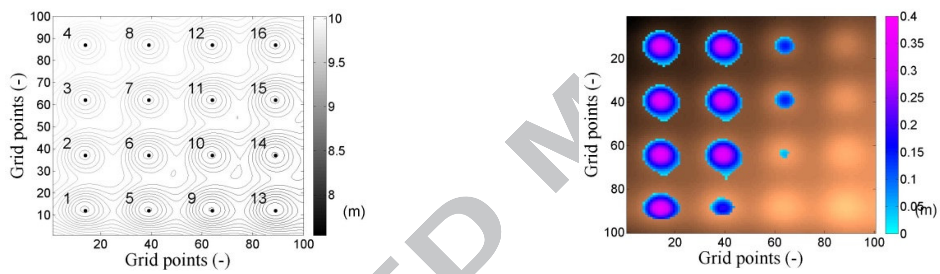


Figure 6. Map of the DEM showing the points locations where water levels are recorded (left). Final inundation predicted by P-DWave (right).

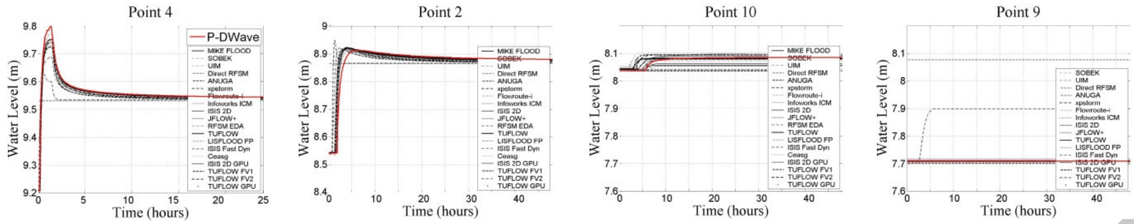


Figure 7. Water-surface level for the “filling of floodplain depressions” test predicted by P-DWave superimposed with the results from the models published in the EA benchmark (Pender and Néelz, 2010) test for comparison.

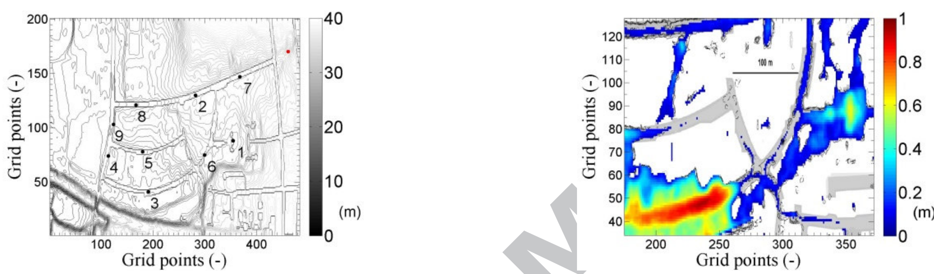


Figure 8. Map of the DEM showing the points locations where water levels are recorded (black dots) and the inflow hydrograph location (red dot) (left). Final Inundation predicted by P-DWave overlapped with the models’s flood extents in the EA report (Pender and Néelz, 2010) (right).

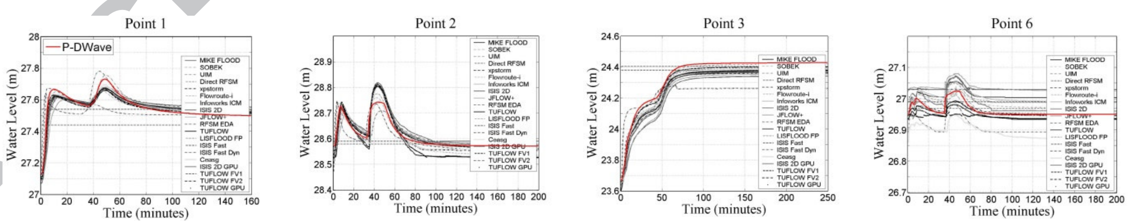


Figure 9. Water-surface level for the “rainfall and point source surface flow in urban areas” test predicted by P-DWave superimposed with the results from the models published in the B-EA test (Pender and Néelz, 2010).

488

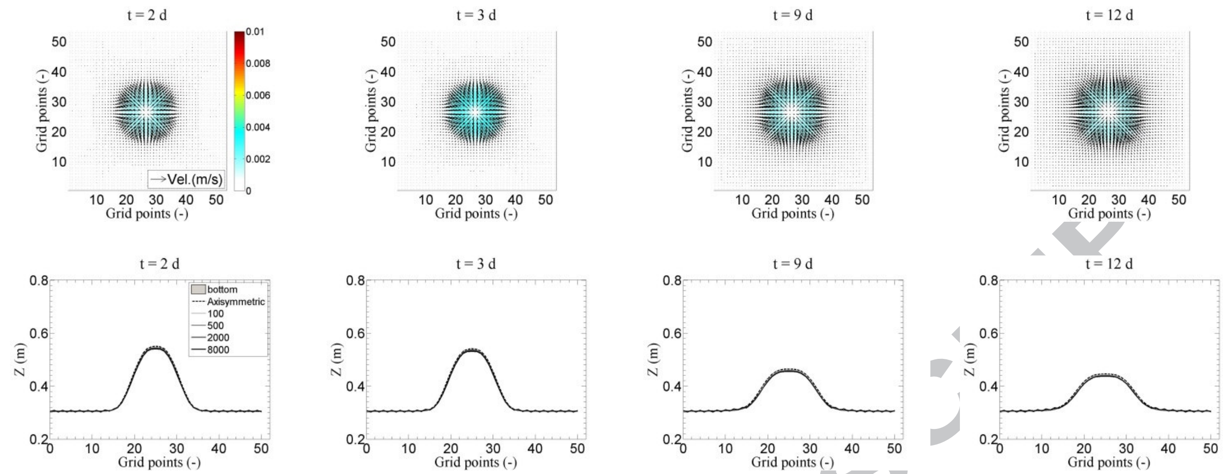


Figure 10. Water-surface level and velocity fields (Δt_{min}=100 s) for the "Axisymmetric" test predicted by the P-DWave and the Axisymmetric model solution (dashed line) on the front propagation. Sensitivity analysis of the water-surface centre profile for Δt_{min}={100, 500, 2000, 8000} s and CellNo=50x50 (Δx =3218.7 m).

489

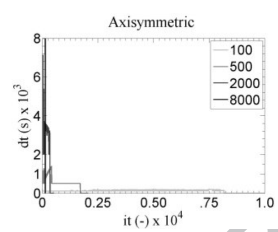


Figure 11. Evolution of the time step solution with the iteration number (it) as a function of the smallest allowable time step Δt_{min}={100, 500, 2000, 8000} s with CellNo=50x50 (Δx =3218.7 m) during the Axisymmetric test.

490

491

Tables

Table 1. RMSE (m) statistics for the "horizontal plane wetting" predicted by using CellNo={200, 100, 50, 25} (grid resolution of Δx={25, 50, 100, 200} m) with Δt_{min}={0.001, 0.05, 0.5, 1.0} s compared with the analytical solution for 8 instants in time.

RMSE (m)	t (s)							
CellNo	500	900	1400	1800	2300	2700	3200	3600
200	0.005	0.007	0.009	0.010	0.012	0.013	0.015	0.016
100	0.010	0.013	0.016	0.018	0.022	0.025	0.027	0.030
50	0.019	0.025	0.031	0.034	0.040	0.045	0.050	0.054
20	0.028	0.038	0.048	0.058	0.065	0.077	0.084	0.093

Table 2. RMSE (m) statistics for the "inundation of a planar beach" S≠0 predicted by using Δt_{min}={0.001, 0.01, 0.1,1} s and n={0.01,0.02,0.04,0.08} $m^{-1/3}s$ compared with Δt_{min}=0.0005 s, for four instants in time.

RMSE (m)	t = 400 s				t = 1100 s			
	Δt_{min} (s)				Δt_{min} (s)			
n ($s/m^{1/3}$)	0.001	0.01	0.1	1	0.001	0.01	0.1	1
0.01	0.000	0.000	0.000	0.245	0.000	0.000	0.515	1.357
0.02	0.000	0.000	0.000	0.016	0.000	0.000	0.004	1.009
0.04	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.552
0.08	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.017

RMSE (m)	t = 1900 s				t = 3000 s			
	Δt_{min} (s)				Δt_{min} (s)			
n ($s/m^{1/3}$)	0.001	0.01	0.1	1	0.001	0.01	0.1	1
0.01	0.000	0.001	0.537	1.347	0.000	0.000	0.052	0.167
0.02	0.000	0.000	0.173	1.252	0.000	0.000	0.029	0.323
0.04	0.000	0.000	0.022	0.994	0.000	0.000	0.001	0.356
0.08	0.000	0.000	0.000	0.328	0.000	0.000	0.000	0.121

Note: RMSE >0.20m , >0.50m , >1.00m

Table 3. RMSE (m) error statistics for the Axisymmetric test predicted by using Δt_{min}={100, 500, 2000, 8000} s and CellNo=50x50 (Δx =3218.7 m) compared with the analytical solution, for four instants in time.

RMSE (m)	t (d)			
Δt_{min}	2	3	9	12
100	0.004	0.004	0.004	0.004
500	0.004	0.004	0.005	0.005
2000	0.004	0.004	0.005	0.005
8000	0.006	0.006	0.006	0.006

Table 4. Efficiency statistics of the Matlab and Fortran codes for the Axisymmetric test for CellNo={300x300, 500x500, 700x700, 900x900} ($\Delta x =${536.4, 321.9, 229.9, 178.8} m) with Δt_{min}=500 s and one hour simulation time: Computational Time in seconds, Speed-up and Efficiency.

Code	Parallel Performance	Computational Time (s)				Speed-up				Efficiency			
		Number of processors				Number of processors				Number of processors			
	CellNo (Δx (m))	1	2	4	12	1	2	4	12	1	2	4	12
Matlab	300x300 (536.4)	125.9	75.2	47.8	33.1	1	1.7	2.6	3.8	1	0.84	0.66	0.32
	500x500 (321.9)	348.8	196.0	120.9	75.5	1	1.8	2.9	4.6	1	0.89	0.72	0.38
	700x700 (229.9)	716.0	379.6	227.6	139.0	1	1.9	3.1	5.2	1	0.94	0.79	0.43
	900x900 (178.8)	1171.0	625.7	373.5	230.1	1	1.9	3.1	5.1	1	0.94	0.78	0.42
Fortran	300x300 (536.4)	10.7	9.4	8.8	7.3	1	1.1	1.2	1.5	1	0.57	0.30	0.12
	500x500 (321.9)	34.5	28.2	25.6	20.5	1	1.2	1.3	1.7	1	0.61	0.34	0.14
	700x700 (229.9)	66.0	56.3	52.5	39.8	1	1.2	1.3	1.7	1	0.59	0.31	0.14
	900x900 (178.8)	108.6	89.8	82.3	72.7	1	1.2	1.3	1.5	1	0.60	0.33	0.12

A 2D Parallel Diffusive Wave Model for floodplain inundation with variable time step (P-DWave)

Leandro J. [1,3], Chen, A. S. [2], and Schumann, A. [1]

[1] Institute of Hydrology, Water Management and Environmental Techniques, Ruhr-University Bochum, Germany. E-mail: Jorge.leandro@ruhr-uni-bochum.de, Andreas.Schumann@ruhr-uni-bochum.de

[2] Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK. E-mail: a.s.chen@exeter.ac.uk

[3] IMAR, Department of Civil Engineering , University of Coimbra ,Portugal, R. Luís Reis Santos, Polo 2, 3030-788 Coimbra, Portugal.

Figure Caption

Figure 1. Water-surface level for the “horizontal plane wetting” test predicted by P-DWave and the analytical solution (dashed line). Sensitivity analysis to the number of cells CellNo={200, 100, 50, 25} (Δx={25, 50, 100, 200} m) with Δt_{min}={0.001, 0.05, 0.5, 1.0} s on the front propagation.

Figure 2. Evolution of the time step solution with the iteration number (it) as a function of the number of cells discretised CellNo={200, 100, 50, 25} (Δx={25, 50, 100, 200} m) with Δt_{min}={0.001, 0.05, 0.5, 1.0} s during the horizontal plane wetting simulation.

Figure 3. Sensitivity analysis of the predicted water-surface level propagation to Manning’s coefficient for the “inundation of a planar beach” S≠0 test ($\Delta\mathrm{x}$=50 m and $\Delta\mathrm{t}_{\mathrm{min}}$=0.001 s).

Figure 4. Evolution of the time step solution with the iteration number (it) as a function of Manning’s coefficient n={0.01,0.02,0.04,0.08} $m^{-1/3}s$ and $\Delta\mathrm{t}_{\mathrm{min}}$=0.001 s during the Inundation of a planar beach S≠0 simulation.

Figure 5. Profile of the digital elevation model (DEM) (left). Water-surface levels at points 1 and 2 for the “flooding a disconnected water body” test predicted by P-Dwave and superimposed with the various models’ results published in the EA benchmark (Pender and Néelz, 2010) test for comparison (middle and right).

Figure 6. Map of the DEM showing the points locations where water levels are recorded (left). Final inundation predicted by P-Dwave (right).

Figure 7. Water-surface level for the “filling of floodplain depressions” test predicted by P-Dwave superimposed with the results from the models published in the EA benchmark (Pender and Néelz, 2010) test for comparison.

Figure 8. Map of the DEM showing the points locations where water levels are recorded (black dots) and the inflow hydrograph location (red dot) (left). Final Inundation predicted by P-Dwave overlapped with the models’s flood extents in B-EA (Pender and Néelz, 2010) (right).

Figure 9. Water-surface level for the "rainfall and point source surface flow in urban areas" test predicted by P-Dwave superimposed with the results from the models published in the B-EA test (Pender and Néelz, 2010).

Figure 10. Water-surface level and velocity fields (Δt_{min}=100 s) for the "Axisymmetric" test predicted by the P-Dwave and the Axisymmetric model solution (dashed line) on the front propagation. Sensitivity analysis of the water-surface centre profile for Δt_{min}={100, 500, 2000, 8000} s and CellNo=50x50 (Δx =3218.7 m).

Figure 11. Evolution of the time step solution with the iteration number (it) as a function of the smallest allowable time step Δt_{min}={100, 500, 2000, 8000} s with CellNo=50x50 (Δx =3218.7 m) during the Axisymmetric test.

A 2D Parallel Diffusive Wave Model for floodplain inundation with variable time step (P-DWave)

Leandro J. [1,3], Chen, A. S. [2], and Schumann, A. [1]

[1] Institute of Hydrology, Water Management and Environmental Techniques, Ruhr-University Bochum, Germany. E-mail: Jorge.leandro@ruhr-uni-bochum.de, Andreas.Schumann@ruhr-uni-bochum.de

[2] Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK. E-mail: a.s.chen@exeter.ac.uk

[3] IMAR, Department of Civil Engineering , University of Coimbra ,Portugal, R. Luís Reis Santos, Polo 2, 3030-788 Coimbra, Portugal.

Graphical Abstract

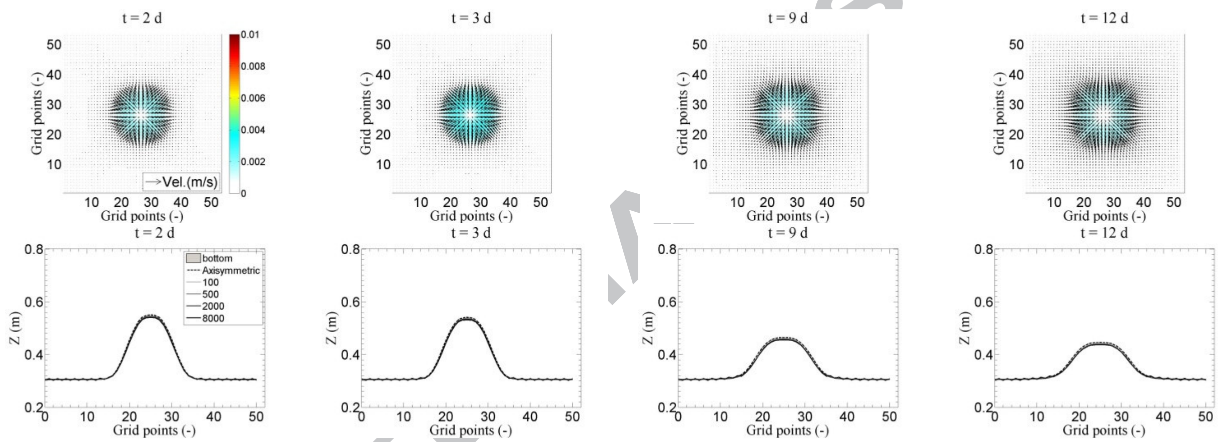


Figure 10. Water-surface level and velocity fields (Δt_{min}=100 s) for the "Axisymmetric" test predicted by the P-DWave and the Axisymmetric model solution (dashed line) on the front propagation. Sensitivity analysis of the water-surface centre profile for Δt_{min}={100, 500, 2000, 8000} s and CellNo=50x50 (Δx =3218.7 m).

Table 4. Efficiency statistics of the Matlab and Fortran codes for the Axisymmetric test for CellNo={300x300, 500x500, 700x700, 900x900} (Δx ={536.4, 321.9, 229.9, 178.8} m) with Δt_{min}=500 s and one hour simulation time: Computational Time in seconds, Speed-up and Efficiency.

Code	Parallel Performance	Computational Time (s) Number of processors				Speed-up Number of processors				Efficiency Number of processors			
	CellNo (Δx (m))	1	2	4	12	1	2	4	12	1	2	4	12
Matlab	300x300 (536.4)	125.9	75.2	47.8	33.1	1	1.7	2.6	3.8	1	0.84	0.66	0.32
	500x500 (321.9)	348.8	196.0	120.9	75.5	1	1.8	2.9	4.6	1	0.89	0.72	0.38
	700x700 (229.9)	716.0	379.6	227.6	139.0	1	1.9	3.1	5.2	1	0.94	0.79	0.43
	900x900 (178.8)	1171.0	625.7	373.5	230.1	1	1.9	3.1	5.1	1	0.94	0.78	0.42
Fortran	300x300 (536.4)	10.7	9.4	8.8	7.3	1	1.1	1.2	1.5	1	0.57	0.30	0.12
	500x500 (321.9)	34.5	28.2	25.6	20.5	1	1.2	1.3	1.7	1	0.61	0.34	0.14
	700x700 (229.9)	66.0	56.3	52.5	39.8	1	1.2	1.3	1.7	1	0.59	0.31	0.14
	900x900 (178.8)	108.6	89.8	82.3	72.7	1	1.2	1.3	1.5	1	0.60	0.33	0.12

A 2D Parallel Diffusive Wave Model for floodplain inundation with variable time step (P-DWave)

Leandro J. [1,3], Chen, A. S. [2], and Schumann, A. [1]

[1] Institute of Hydrology, Water Management and Environmental Techniques, Ruhr-University Bochum, Germany. E-mail: Jorge.leandro@ruhr-uni-bochum.de, Andreas.Schumann@ruhr-uni-bochum.de

[2] Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK. E-mail: a.s.chen@exeter.ac.uk

[3] IMAR, Department of Civil Engineering , University of Coimbra ,Portugal, R. Luís Reis Santos, Polo 2, 3030-788 Coimbra, Portugal.

Highlights

- We develop a parallel 2D diffusive wave model in Matlab and Fortan
- We achieved speed-up times ranging from 1.2 to 5.2 using 2 to 12 processors
- The variable time step method and the process for Wet-dry fronts kept the solution stable
- Absolute mass conservation is obtained in all seven tests used to validate the Model