Response to Reviewer #1

We would like to thank the reviewer for their precious time and for the thorough review and the many helpful comments and suggestions made to improve the present work. Please find below the reviewer's comments and author's replies to these comments.

Assessment and Further comments

Q1. "[…] I don't understand the criteria followed for leaving out some important models, especially those models used by different Tsunami Warning Centers along the World […]". "[…] Of course, it's not possible to include all the developed tsunami models, but as the development of TRITON-G was focused on its operational capabilities, it would be expectable to focus this analysis on models used in this context, moreover if some these models have been developed in the GPU […]"

A1. We based the previous existing work on an historical development and on the models we considered well known and that we were more familiar with, as well as some models mentioned in the National Tsunami Hazard Mitigation Program (NTHMP, 2012). As mentioned by the reviewer is not possible to include all models and we appreciate the important references provided about these other models left out which will be included in the introduction, specially the GPU models suggested since they are relevant to our research (p.2, at lines 8,13, 16 and 27).

Section 2

Q2. So, regarding the numerical methods used in this paper: in the propagation stage it's used the method of characteristics (MOC) in combination with a cubic-polynomial interpolation to find the interpolated values on the "[…] Is the total mass conserved? For instance, what about the mass conservation in the experiment described in p8 line 10?"

A2. On this regards, the discretization used for the run-up method is non-conservative. However, the difference from the initial mass and final mass is negligible and well below the 5% criteria that the NOAA Technical Memorandum OAR PMEL-135 suggests (Synolakis et al., 2007). The long wave typically associated with tsunamis implies a small error in the result and the analytical solution show excellent agreement in the experiment mentioned which implies that no large mass loss is present. A comment about mass conservation was added in the manuscript (p.9 line 11).

Q3. Regarding the water height positivity (h), is it guaranteed in the propagation stage even considering the cubic polynomial interpolation process?

A3. The propagation stage of the program is not used in the run-up calculation or on land areas. Coasts not flagged as "*Inundation*" have a wall boundary treatment. In the case of the run-up calculation, the surface gradient method is used. It is constructed as a monotone scheme with flux correction and a slope limiter to preserve height positivity (added in p.9 line 21). Dam-break problem tests with thin water layers have been satisfactory with the wave remaining positive even in shallow conditions. In addition, the new run-up benchmark problems (BP) added to the paper contribute as another demonstration of positivity preservation.

Q4. On other hand, what is the convergence order of the numerical models used on each area?

A4. The convergence of the method of characteristic used for propagation and the method for run-up are different. The propagation is 3rd order while the run-up is a 1.5 order. A measurement of this convergence for the parabolic bowl problem is included to illustrate this point (Fig. A). Figure included in the manuscript Fig. 3 (p.10).
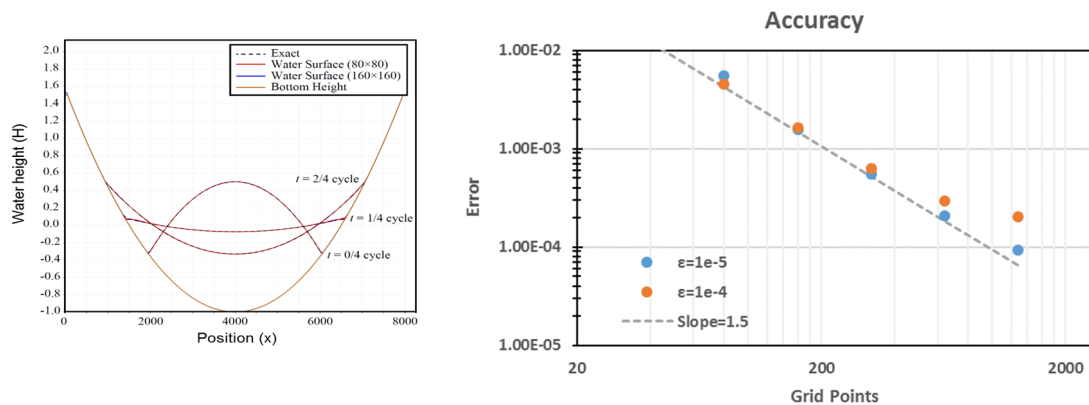

Fig A. Parabolic bowl problem cross section with $\epsilon = 10^{-4}$ on left panel. Water depth error for parabolic bowl problem on right panel

Section 3

Q5. As the coordinate system used combines spherical coordinates and cartesian coordinates, how is treated the boundary between the considered domains on each coordinate system?

A5. The Cartesian coordinate system is used exclusively for the blocks set as "*Inundation*". These blocks are only found at level 7 (highest level) and only inside focal areas. While the focal areas can be user-defined in any size and located on any coast in the domain, by design the focal areas shall consist of just a few kilometers in length. In this way, the area represented by the inundation areas is extremely small compared to the total domain size. This makes it possible not to use a special boundary treatment between systems since the difference of the incoming wave is almost negligible.

Additionally, the domain used centered in the Indian Ocean does not extend to high latitudes that might introduce large discrepancies in the grid. In Fig. B, the current focal areas used by TRITON-G are shown with the approximate length in kilometers. As noted, they cover just a few kilometers, the largest case being Seychelles, which is just around 27 km by 17 km. We added a comment explaining this boundary in the revised manuscript (p12, line 3) as well as modified Fig. 5 (p.14).
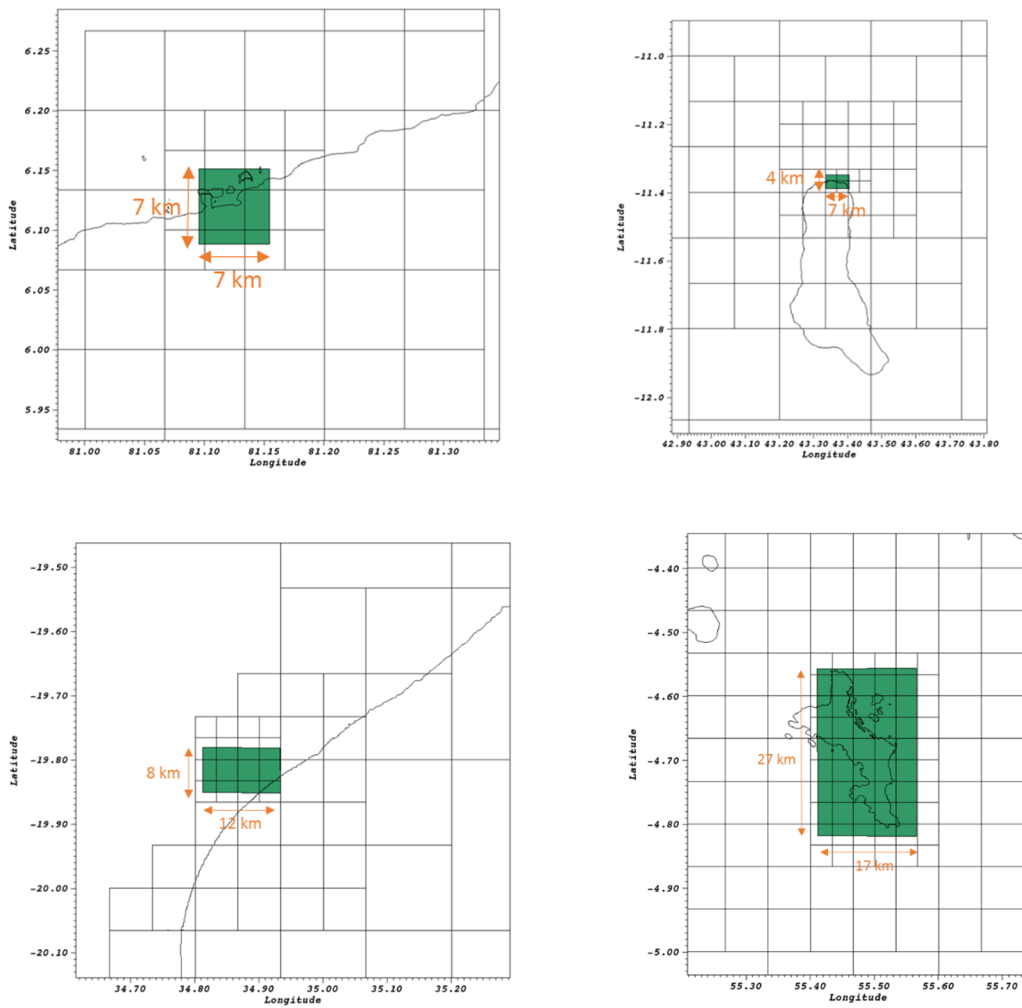
Fig. B. Focal Areas highlighted in green with the approximate length labeled in kilometers. Top Left: Sri Lanka. Top Right: Comoros. Bottom left: Mozambique. Bottom Right: Seychelles

Q6. Regarding the run-up calculation, in section 3.2 authors propose a technic based on considering (hu=hv=0) when h is less than a certain small fixed quantity. They confirm that the proposed implementation has been proven to be robust and stable under different benchmarks. Please, include a reference where this numerical technic is used.

A6. Some references (p. 9, line 20) about the thin layer technique and its implementation are:

- Vincent, Stéphane and Caltagirone, J.P and Bonneton, Philippe. (2001). "Numerical modeling of bore propagation and run-up on sloping beaches using a MacCormack TVD scheme." Journal of Hydraulic Research – 39: pp 41-49.
- Xin, L and Abdolmajid, M and Infants Sedano, J. "A Well-Balanced 2-D Model for Dam-Break Flow with Wetting and Drying" Journal of Fluid flow, Heat and Mass Transfer. Vol 1 (2014): pp: 30-37
- Wang, L.X. (1987) ''Composite-model of 1-d and 2d flows'', Advanced course and workshop on mathematical modeling of alluvial river, IRTCES-Beijing, China.

Q7. Finally, in this section it's included a parabolic bowl problem as validation problem. That's a good synthetic test but I consider that it's not enough to define a numerical model as validated when such numerical model that is going to be used with real topobathymetries in real cases, moreover when, as it's remarked by authors, the model is going to use in the RIMES context. In order to validate this numerical scheme to be used in real cases I suggest, for instance, the use of the inundation benchmark experiments proposed by the National Tsunami Hazard Mitigation Program (NTHMP) where problems with analytical solutions, laboratory experiments and real problems with field measurements are proposed. I would suggest studying the behavior of TRITONG at least in the "mandatory" benchmark problems: BP1, BP4, BP6, BP7 and BP9. Regarding the field measurement experiments, authors show in section 6.1.2 some inundation maps in different locations (Fig. 18 and 19) where the comparisons are made basically against other models. I think that it cannot be considered a numerical model as validated with field measurements by making comparisons basically with other models results. There are available many tsunami field measurements to validate the inundation process. In this sense you can consider cases where there are more available data than in the case studied in section 6.1.2. For instance: BP7, BP9 or many inundation scenarios related to the Tohoku, 2011 event where detailed data are available.

A7. We appreciate the suggestion to validate TRITON-G against the inundation benchmark problems. We have completed the results for benchmarks BP4, BP6, BP7 and BP9. We skipped benchmark problem BP1 considering that it is a one-dimensional analytical problem and we already showed results with good agreement for the parabolic bowl problem, which is a two-dimensional analytical problem. Additionally BP4 is the experimental version of BP1 and results for BP4 are now included. This serves to demonstrate the behavior obtained with TRITON-G under these conditions.

Regarding the map comparison with another model (Fig. 18 and Fig 19) we consider that is an acceptable way to show agreement since the results from these models are peer-reviewed and published (Supparsri et al., 2011). However, in order to produce a more complete validation of our model we agree with the importance of using tsunami field measurements and known benchmark

problems. For this reason, we also use BP7 and BP9 as comparison to validate the inundation process.

A new section was added to the manuscript to include results for BP9 (p.28, Section 6) while the rest of the benchmarks are located in the appendix (p. 44).

1. Benchmark problem #4: Solitary wave on a simple beach – Laboratory

Figure C depicts the domain for this test. In this problem, the wave height H is located at a distance L from the beach toe. This test was replicated in a wave tank 31.73-cm-long, 60.96-cm-deep and 39.97-cm-wide at the California Institute of Technology. Several experiments with different water heights were performed. Benchmark Problem 4 (BP4) uses the datasets for $H/d = 0.0185$ non-breaking wave and $H/d = 0.30$ breaking wave for code validation. Results use dimensionless units with the help of parameters like length d, velocity scale U $= \sqrt{gd}$ and time scale T $= \sqrt{d/g}$.
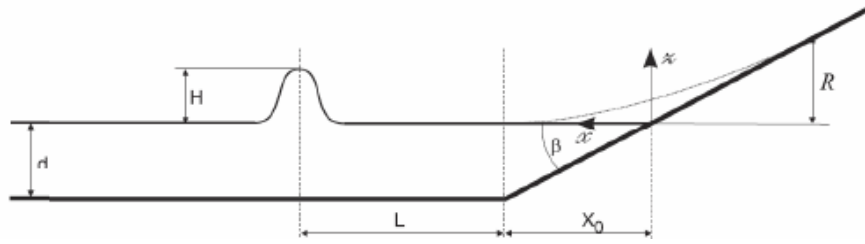
Fig. C Domain sketch for BP4 with slope 1:19.85 (figure taken from benchmark description)

1.1 Problem setup

- *Parameters*: d = 1, g = 9.8, case A with $H/d = 0.0185$ and case C with $H/d = 0.30$.
- *Friction*: Manning coefficient set to 0.01
- *Computational domain*: the domain along x direction spanned from x $= -20$ to x $= 80$
- *Boundary conditions*: the right side of the computational domain uses a non-reflective boundary condition.
- *Grid resolution*: the numerical results are solved with a resolution of $\Delta x = 0.1$
- CFL: 0.9
- *Initial condition*: the initial wave is computed based on the following equations for height (η) and velocity (u) respectively

$$\eta(x.0) = H sech^2[\gamma(x - x_s)/d] \,, \tag{1}$$

$$u(x, 0) = -\eta(x, 0)\sqrt{\frac{g}{d}}. \tag{2}$$

1.2 Tasks to be performed

To accomplish this problem, the following tasks should be performed:

1. Compare numerically calculated surface profiles at t/T=30:10:70 for the non-breaking case $H/d = 0.0185$ with the lab data (Case A).

2. Compare numerically calculated surface profiles at t/T=15:5:30 for the breaking case $H/d = 0.30$ with the lab data (Case C).

3. Compute maximum run-ups for at least one non-breaking and one breaking wave case.

1.3 Numerical results

We present the numerical results obtained using TRITON-G. Figure D shows the comparison between water surface level measured in the experiment and the modeled numerical results obtained by our model for times 30, 40, 50, 60 and 70 for case A ($H/d = 0.0185$). Our results show good agreement between the numerical simulation and the non-breaking experiment.

Table A shows the errors computed for the normalized root mean square deviation (*NRMSD*) and for the maximum wave amplitude error (*MAX*). The error values obtained by the NTHMP workshop models are also included for comparison (taken from Table 1-8, page 41 in (NTHMP, 2012)). These values are divided into two columns, one with results for the non-dispersive models (*ND*) and the other with results for the non-dispersive and dispersive models together (labeled *ALL*). Errors obtained from our simulation tend to be similar or smaller than those errors obtained by other ND models, with just slight exception for time 70. Additionally, except for time 70 our errors are smaller than those obtained combining non-dispersive and dispersive mean error value.

Water level comparison for case C ($H/d = 0.30$) at times 15, 20, 25 and 30 is shown in Fig. E. Table B gathers the values for NRMSD and MAX errors for our numerical results and for the NTHMP workshop models. In this case, only the results of models that reported their errors are included (taken from Table 1-8, page 41 in (NTHMP, 2012)).

For case C conditions, the shallow water equations are no longer appropriate for modeling and hydrostatic models tend to produce larger differences than non-hydrostatic ones. Our numerical results in general show good agreement with the experiment. The difference with the steepening of the crest that is noticeable in the results is expected from a hydrostatic model. In spite of that, this steeping in our model is not very large and it can trace the wave front well. Once the wave breaking occurs, our model can simulate reasonably well the run-up. This is also partly reflected in the small NRMSD error estimation obtained by our model after the wave breaking.

Maximum run-up for case A and case C were calculated. For the non-breaking case A, the obtained run-up value is 0.091 and for the breaking case C, the run-up estimated is 0.588. These values are plotted in Fig. F with a yellow and red dot respectively, it can be seen that both values lie well within the experimental results.

	NRMSD			MAX		
	TRITON-G	NTHMP		TRITON-G	NTHMP	
		ND	ALL		ND	ALL
H = 0.0185						
T 30	8.8	11	11	4.0	6	4
T 40	6.6	9	8	4.8	3	3
T 50	3.5	6	5	7.4	13	7
T 60	3	4	5	1.4	1	3
T 70	11	33	16	13.5	15	9

Table A. Model surface profile errors with respect to laboratory experiments for case A $H/d = 0.0185$ at times 30, 40, 50, 60, and 70. Results from the NTHMP workshop errors are divided into non-dispersive (ND) models and all models (ALL)

	NRMSD		MAX	
	TRITON-G	NTHMP	TRITON-G	NTHMP
H = 0.3		ALL		ALL
T 15	11.3	7	5.4	6
T 20	5.9	9	23.3	11
T 25	6.5	6	11.1	10
T 30	2.9	4	1.4	6

Table B. Modeled surface profile errors with respect to laboratory experiments for case C $H/d = 0.30$ at times 15, 20, 25 and 30. Results from the NTHMP workshop model errors available are shown (ALL)
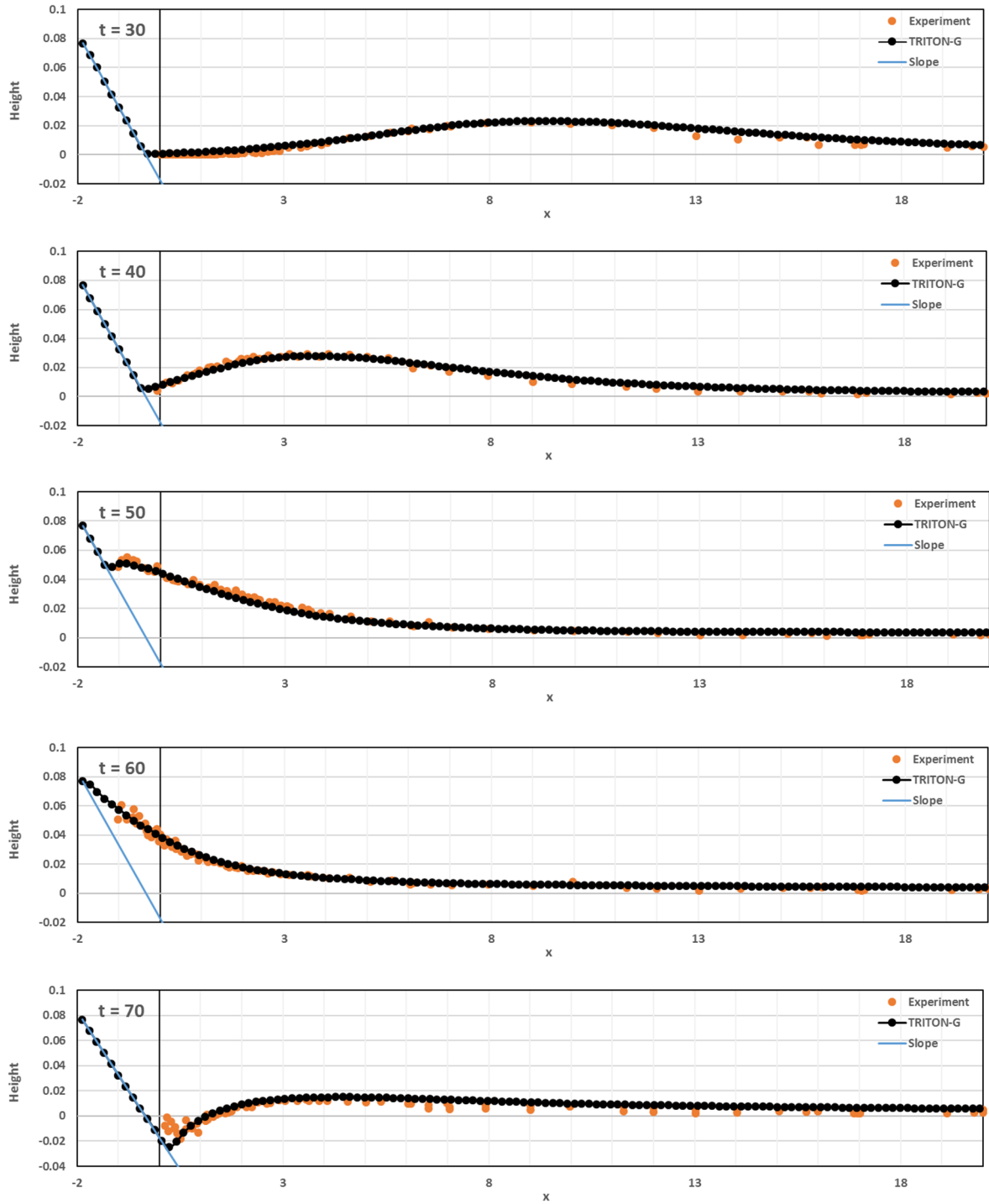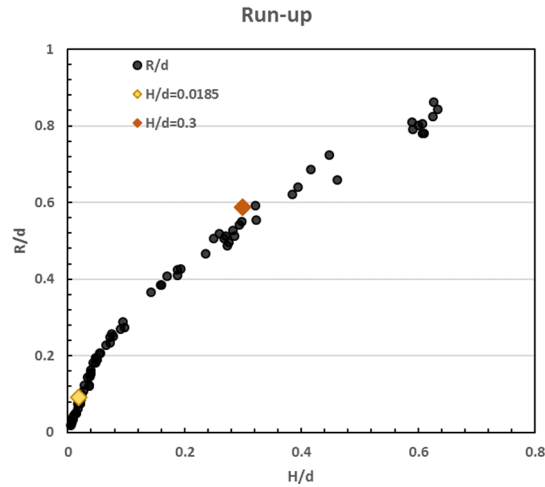
Fig. D. Comparison of numerically calculated free surface profile at different dimensionless times for the non-breaking case $H/d = 0.0185$
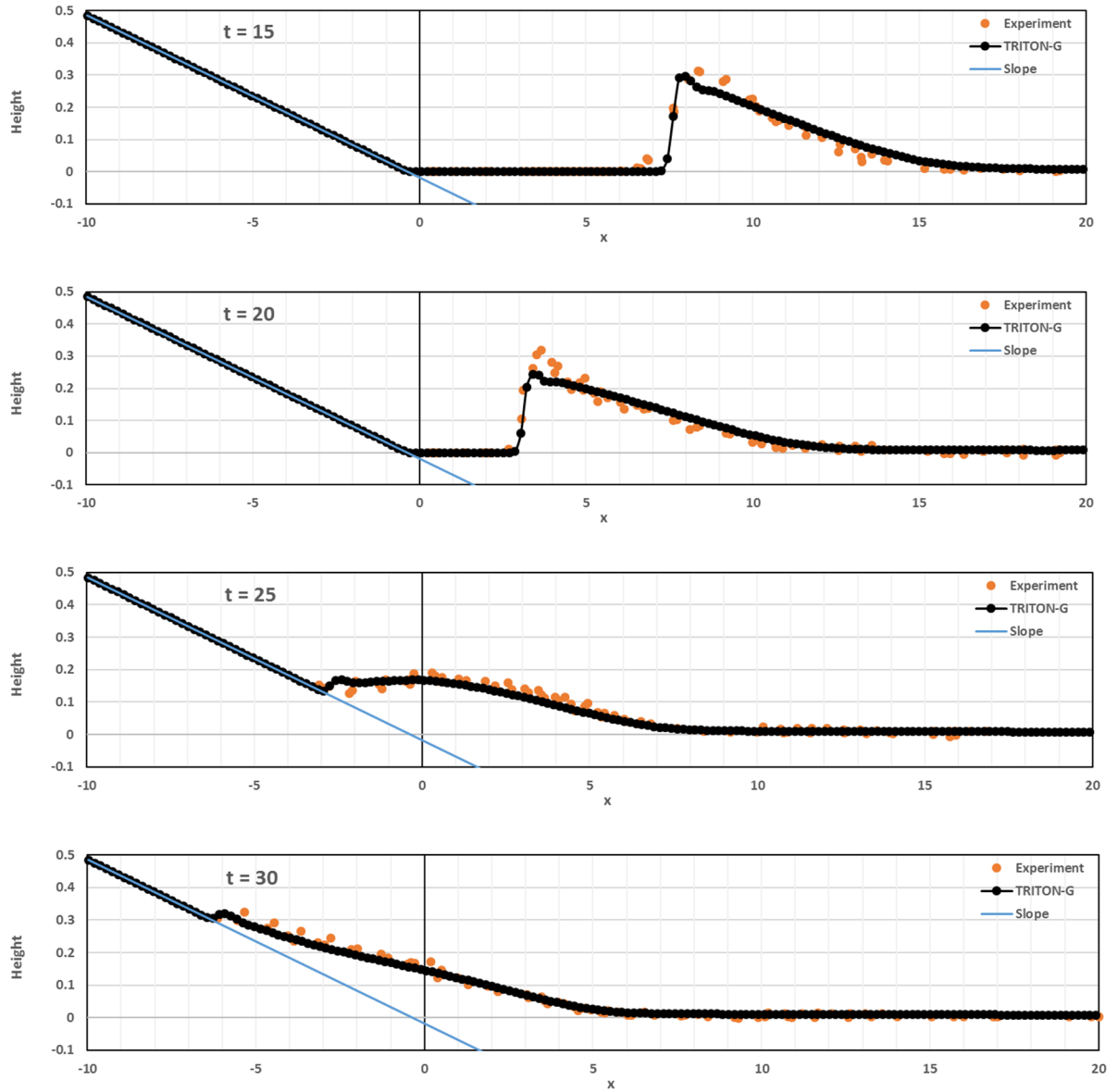
Fig. E. Comparison of numerically calculated free surface profile at different dimensionless times for the breaking case $H/d = 0.30$

Fig. F. Scatter plot of non-dimensional maximum run-up from a total of more than 40 experiments conducted by Y. Joseph Zhan (Synolakis, 1987)**. Orange point indicates TRITON-G result for the breaking case $H/d = 0.30$ and yellow point indicates the result for the non-breaking run-up case $H/d = 0.0185$**

2. Benchmark problem #6: Solitary wave on a conical island – Laboratory

The goal of this benchmark is to compare computed model results with laboratory measurements obtained during a physical modeling experiment conducted at the Coastal and Hydraulic Laboratory Engineer Research and Development Center of the U.S. Army Corps of Engineers. The laboratory physical model was constructed as an idealized representation of Babi Island, in the Flores Sea, Indonesia, to compare with Babi Island run-up measured shortly after the 12 December 1992 Flores Island tsunami (Yeh et al., 1994). Figure G shows schematics of the experiment.

2.1 Tasks to be performed

To accomplish this benchmark, it is suggested that, for

    Case A: water depth d= 32.0 cm, target H=0.05, measured H=0.045

   Case B: water depth d= 32.0 cm, target H=0.20, measured H=0.096

    Case C: water depth d= 32.0 cm, target H=0.05, measured H=0.181

model simulations be conducted to address the following:

1. Demonstrate that two wave fronts split in front of the island and collide behind it
2. Compare computed water levels with laboratory data at gauge 6, 9, 16 and 22
3. Compare computed island run-up with laboratory gauge data

2.2 Problem setup

- *Computational domain (in meters)*: [-5,23] × [0, 28]
- *Boundary condition*: open boundaries
- *Initial condition*: same solitary wave as proposed in BP4 with the correction for two dimensions
- *Grid resolution*: the numerical results presented are solved with a resolution of $\Delta x = 0.05$
- *CFL*: 0.9
- *Friction*: Manning coefficient set to 0.02
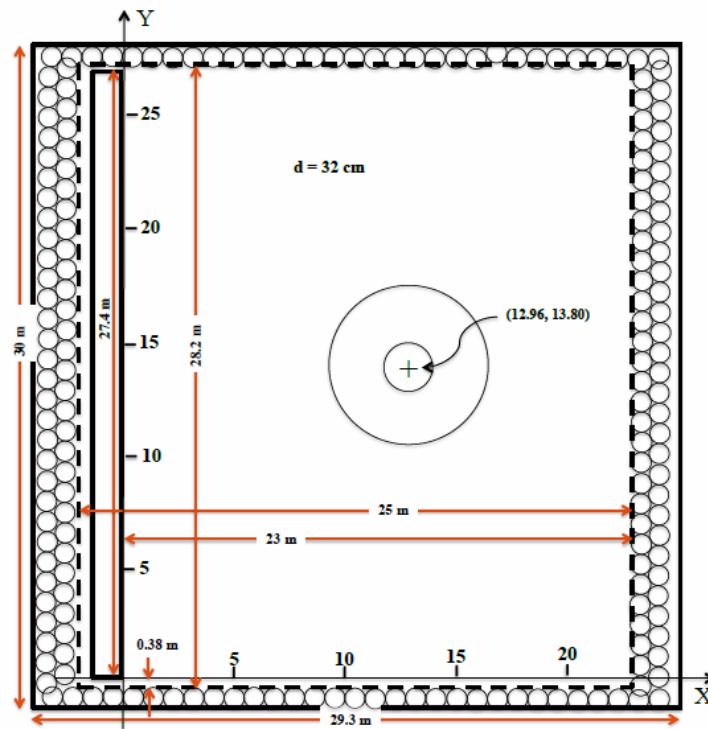
Fig. G Basin geometry and coordinate system. Solid lines represent approximate basin and wave maker surfaces. Circles along walls and dashed lines represent wave-absorbing material
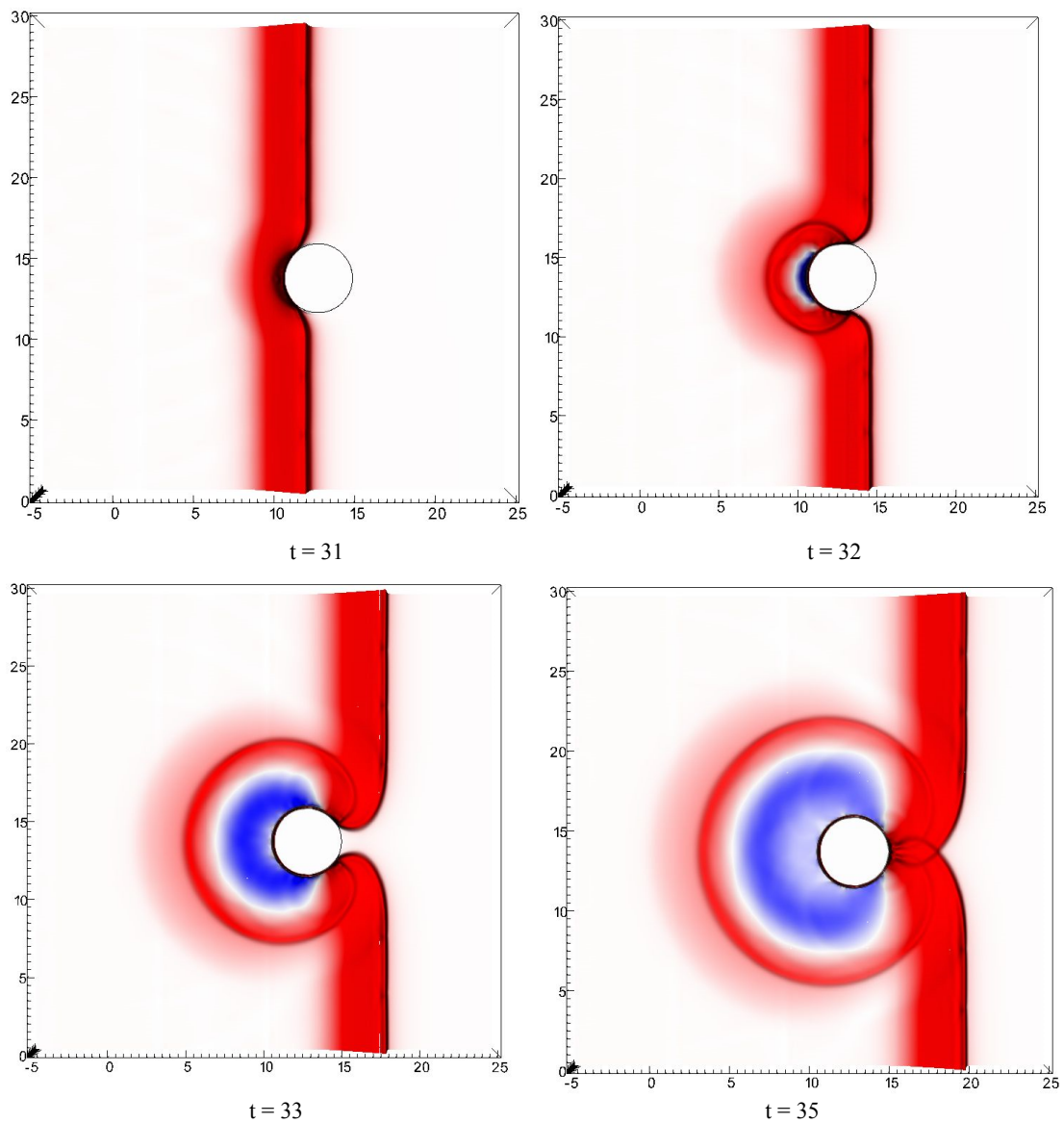
Fig. H Snapshots at several times showing the wave front splitting in front of the island and colliding behind it for BP6 case B.

2.3 Numerical results

We present the numerical results obtained using TRITON-G for the three cases (A, B and C) except for the splitting-colliding item. For this item, Figure H shows the wave front splitting in front of the island and then colliding again behind it for case B (H=0.096), analogue behavior was obtained for the other two cases.

		NRMSD			MAX		
		TRITON-G	NTHMP		TRITON-G	NTHMP	
Case A			ND	ALL		ND	ALL
Gauge	6	10	6	7	4	9	8
Gauge	9	9	7	8	4	14	10
Gauge	16	7	10	9	5	10	12
Gauge	22	9	8	8	4	25	18
Case B							
Gauge	6	10	8	8	7	6	6
Gauge	9	9	8	8	2	7	9
Gauge	16	9	7	7	14	7	7
Gauge	22	8	9	9	6	40	27
Case C							
Gauge	6	13	10	8	3	6	5
Gauge	9	12	11	11	2	9	13
Gauge	16	10	9	8	4	3	3
Gauge	22	9	8	8	10	18	15

Table C. Water level time series TRITON-G model errors with respect to laboratory experiment data for case A, B and C. Mean values obtained for the performing NTHMP models is separated in non-dispersive models (ND) and dispersive and non-dispersive models together (ALL)

Water level comparison uses values for gauges 6, 9, 16 and 22 for each of the 3 cases. Gauge 6 is located at $(9.36, 13.80, 31.7)$, Gauge 9 is located at $(10.36, 13.80, 8.2)$, Gauge 16 is located at $(12.96, 11.22, 7.9)$ and Gauge 22 is located at $(15.56, 13.80, 8.3)$.

Numerical results for Case A, B and C are shown in Figures I, J and K respectively. In the three cases results were stable and in good agreement with the experimental values. The incident wave height and arrival time was captured well for all gauges. Similarly as with BP4, the steepening of the wave with increasing H is expected in a non-hydrodynamic model. After the wave hit the island, some differences between experimental and model wave are noticeable as the initial wave height increased. These oscillations in the experimental data represent the effects of dispersion, which our non-dispersive numerical method is not designed to capture. Despite this, the modeled waves show good agreement with the shape of the experimental waves and the errors estimated tend to be small.
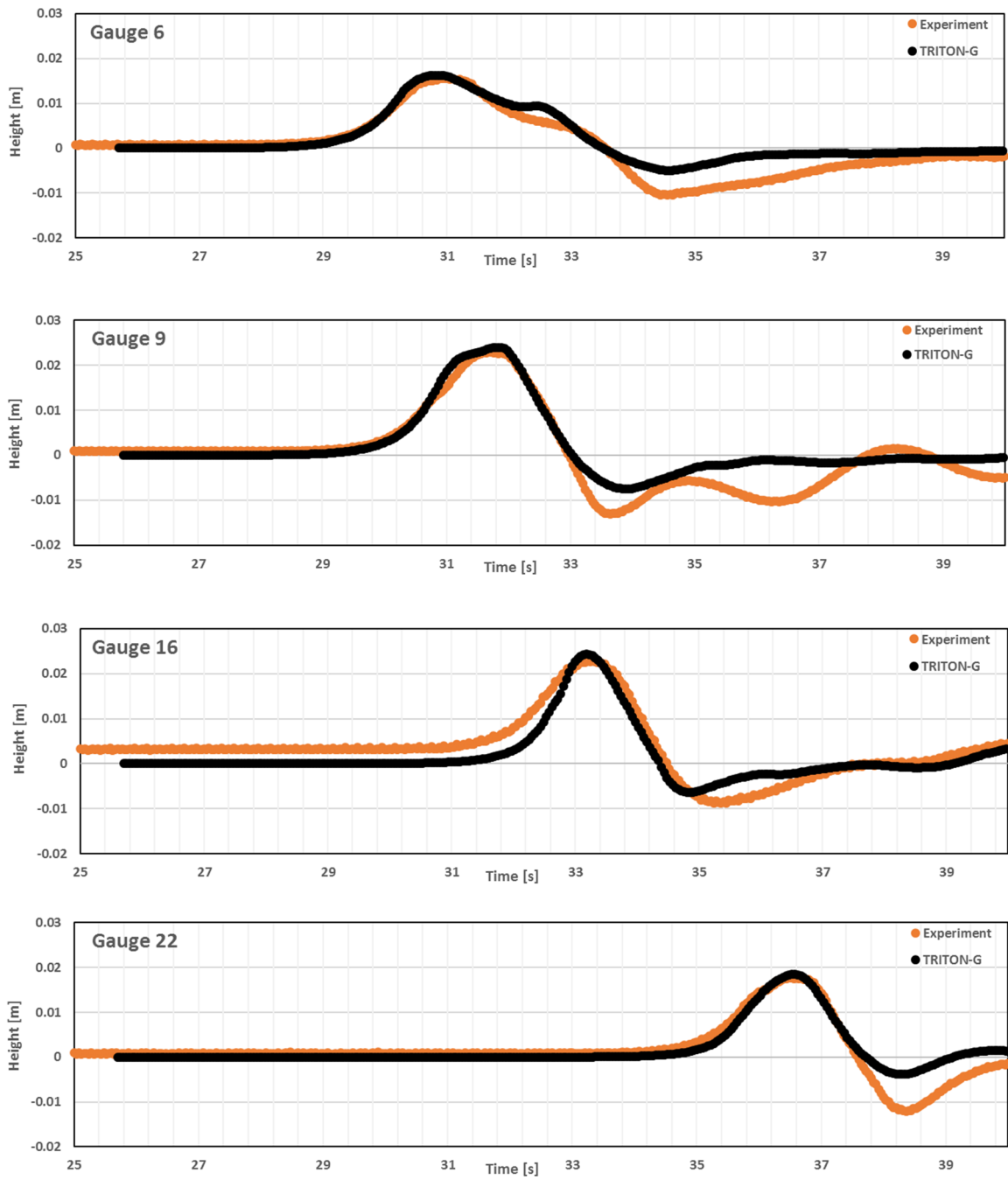
Fig. I Comparison between computed and measured water levels at gauges 6, 9, 16 and 22 for case A (H=0.045)
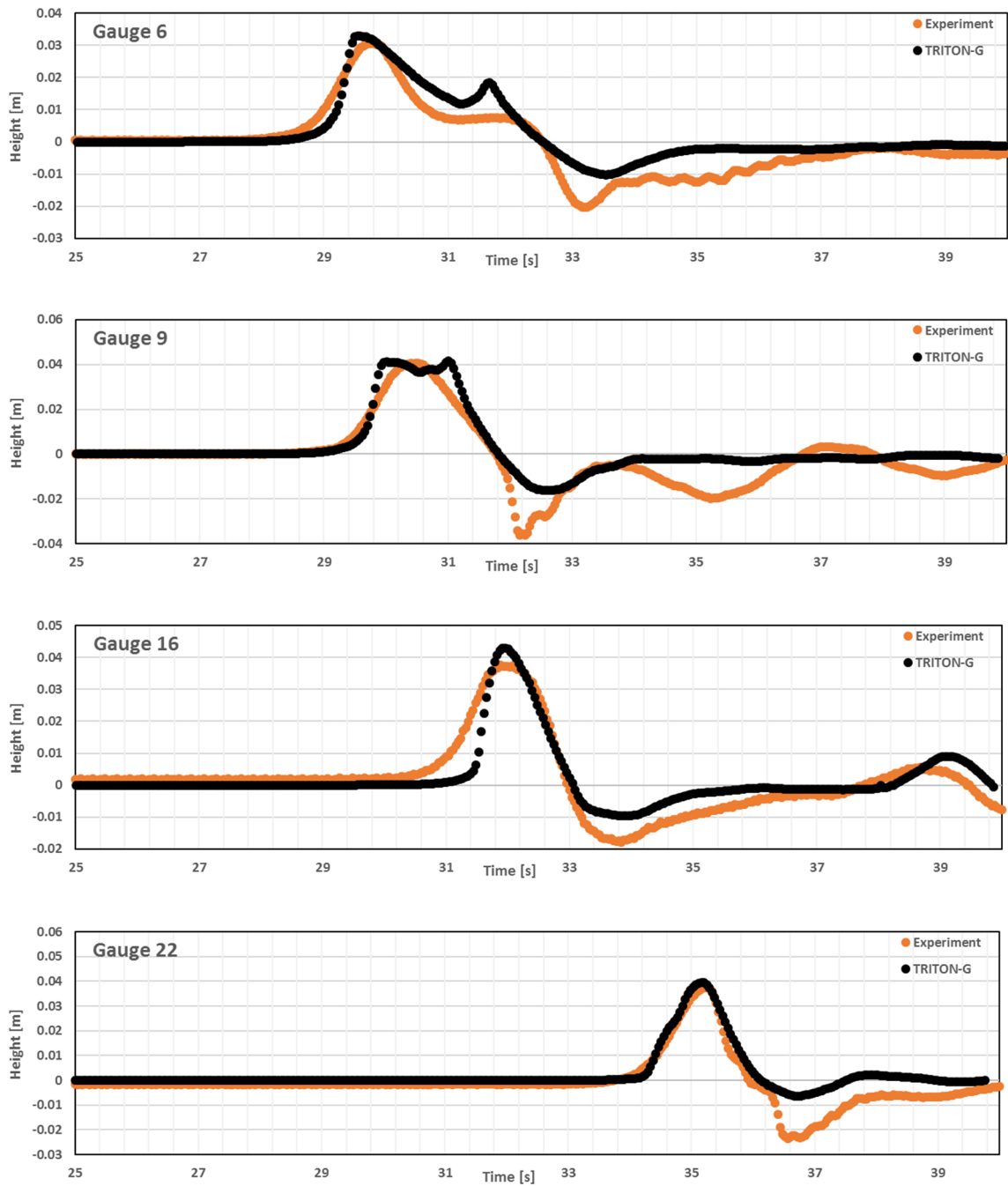
Fig. J Comparison between computed and measured water levels at gauges 6, 9, 16 and 22 for case B (H=0.096)
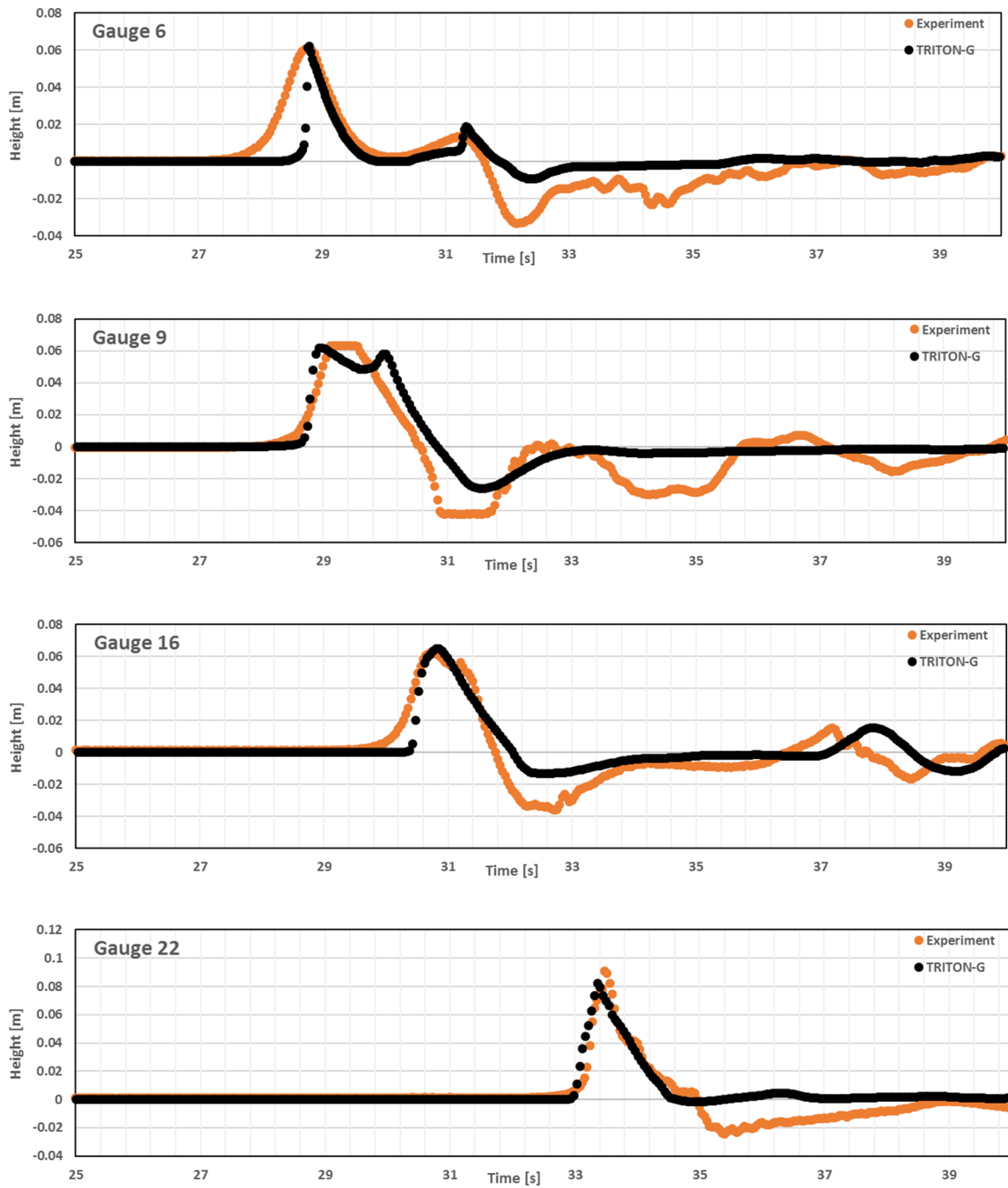
Fig. K Comparison between computed and measured water levels at gauges 6, 9, 16 and 22 for case C (H=0.181)

Fig. L Comparison between computed and measured run-up around the island for the three cases in BP6

	NRMSD			MAX		
	TRITON-G	NTHMP		TRITON-G	NTHMP	
Run-up		ND	ALL		ND	ALL
Case A	9	18	18	0.6	12	7
Case B	19	21	18	9	2	5
Case C	20	12	11	14	5	5

Table D. Run-up TRITON-G model errors with respect to laboratory experiment data for case A, B and C. Mean error values obtained by the performing NTHMP models are separated in non-dispersive models (ND) and all models (ALL)

Table C gathers the normalize root mean square deviation (*NRMSD*) error and the maximum wave height (*MAX*) error. For comparison, mean errors obtained by the participating models in the NTHMP workshop are also included. These are separated in two columns, one for non-dispersive (ND) models and the other for non-disperse and disperse models together (ALL).

NRMSD errors for our model tend to be not very large and in similar range than those of the other non-dispersive models. In the case of the maximum height error (MAX), in almost all cases our model produced smaller error values than the non-dispersive model counterparts. Additionally, in most cases our MAX errors are smaller than those errors of the combined non-dispersive and dispersive mean values.

Figure L shows the comparisons between computed and experimental run-up around the island for the three cases. Case A represent the best agreement with the experimental values. Differences increased with steeper wave cases B and C as several reflections and refraction possibly occur in the basin.

Table D gathers the errors obtained by our model and by the participating models in the NTHMP workshop for run-up cases A, B and C. Figure L showed the good agreement for Case A and this is also reflected in the NRMSD and MAX error results. Both values are considerably smaller than those errors obtained by the NTHMP non-dispersive (ND) models and by the non-dispersive and dispersive together (ALL). For cases B and C, the errors tend to be larger but still similar to those obtained by other non-dispersive models. In all cases, the error stayed below the 20% recommended criteria.

3 Benchmark problem #7: The tsunami run-up onto a complex 3-D beach. Laboratory.

A laboratory experiment using a large-scale tank at the central Research Institute for Electric Power Industry in Abiko, Japan was focused on modelling the run-up of a long wave on a complex beach near the village of Monai (Liu et al., 2008). The beach in the tank was a 1:400-scale model of the bathymetry and topography around a very narrow gully, where extreme run-up was measured.

3.1 Problem setup

The following parameters were used for the computation:

- *Grid resolution*: 393×244 was used with the same resolution 0.014 m as the bathymetry.
- *CFL*: 0.9
- *Initial condition*: water at rest.
- *Friction*: Manning coefficient set to 0.01
- *Boundary conditions*: Solid wall boundary used at the top and bottom. At the left boundary, the given initial wave (shown in Fig. M) was used to specify the condition up to time t=22.5 s, after that it became a wall boundary condition.
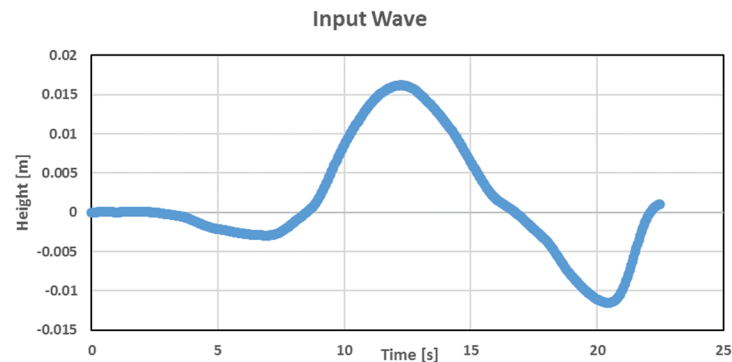
Fig. M Prescribed input wave for the left boundary condition defined from t=0 to t=22.5 s

3.2 Tasks to be performed

To accomplish this benchmark it is suggested to:

1. Model propagation of the incident and reflective wave accordingly to the benchmark-specified boundary condition.
2. Compare the numerical and laboratory-measured water level dynamics at gauges 5, 7 and 9.
3. Show snapshots of the numerically computed water level at the time synchronous with those of the video frames.
4. Compute maximum run-up in the narrow valley.

3.3 Numerical results

This section presents the numerical results for BP7 obtained with TRITON-G to achieve the required tasks.
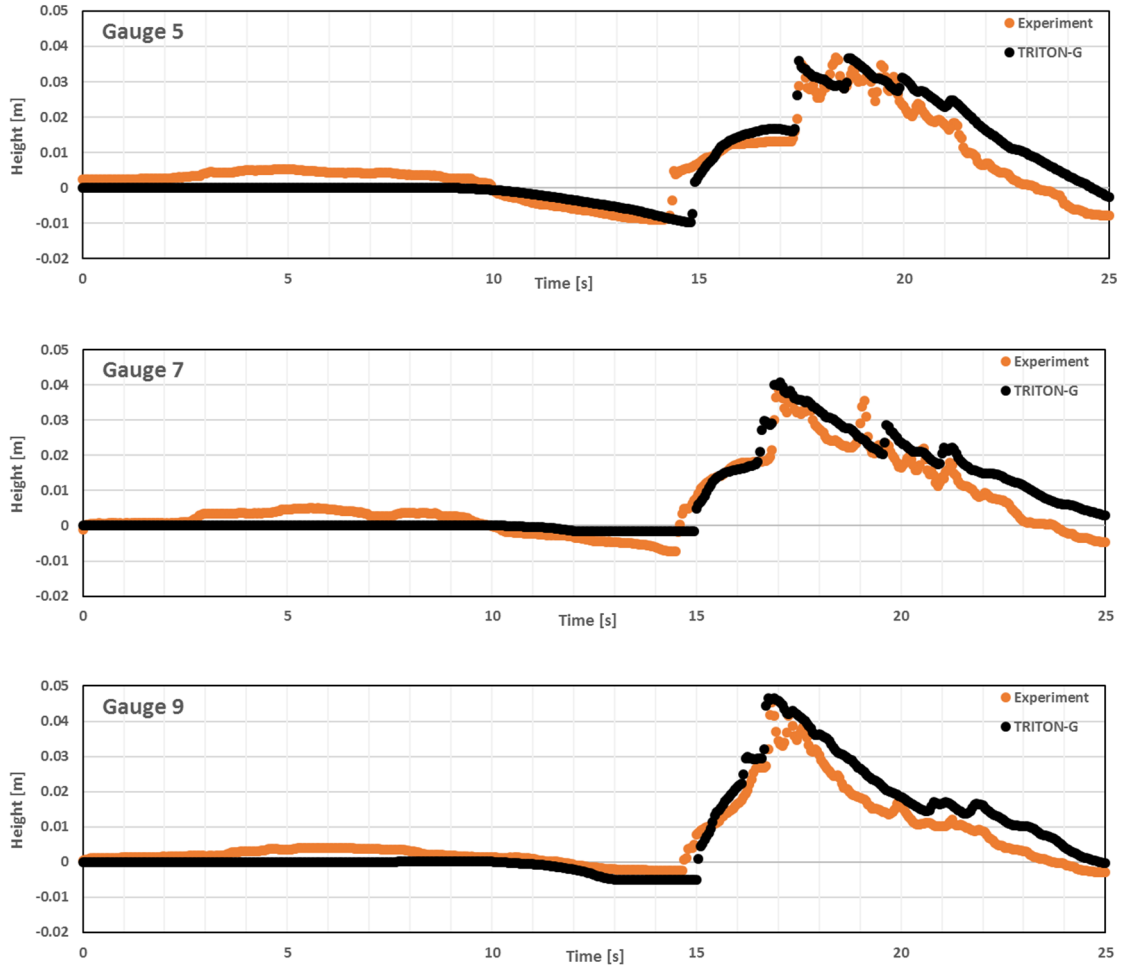
Fig. N Water level comparison for BP7 between experiment and our model for gauges 5, 7 and 9

The comparison with the three requested gauges 5, 7 and 9 is shown in Fig. N from $t = 0$ to $t = 25$ s. For the three cases, good agreement is found between modeled and experimental wave.

Values for the normalized Root Mean Square deviation error (NRMSD) and maximum wave amplitude error (MAX) were estimated for the gauge comparison. For gauge 5, the NRMSD error is 10% and MAX is 0.89%. For gauge 7, NRMSD is 10% and MAX is 4.81%. For gauge 9, the NRMSD error is 6.57% and MAX is 2.66%.
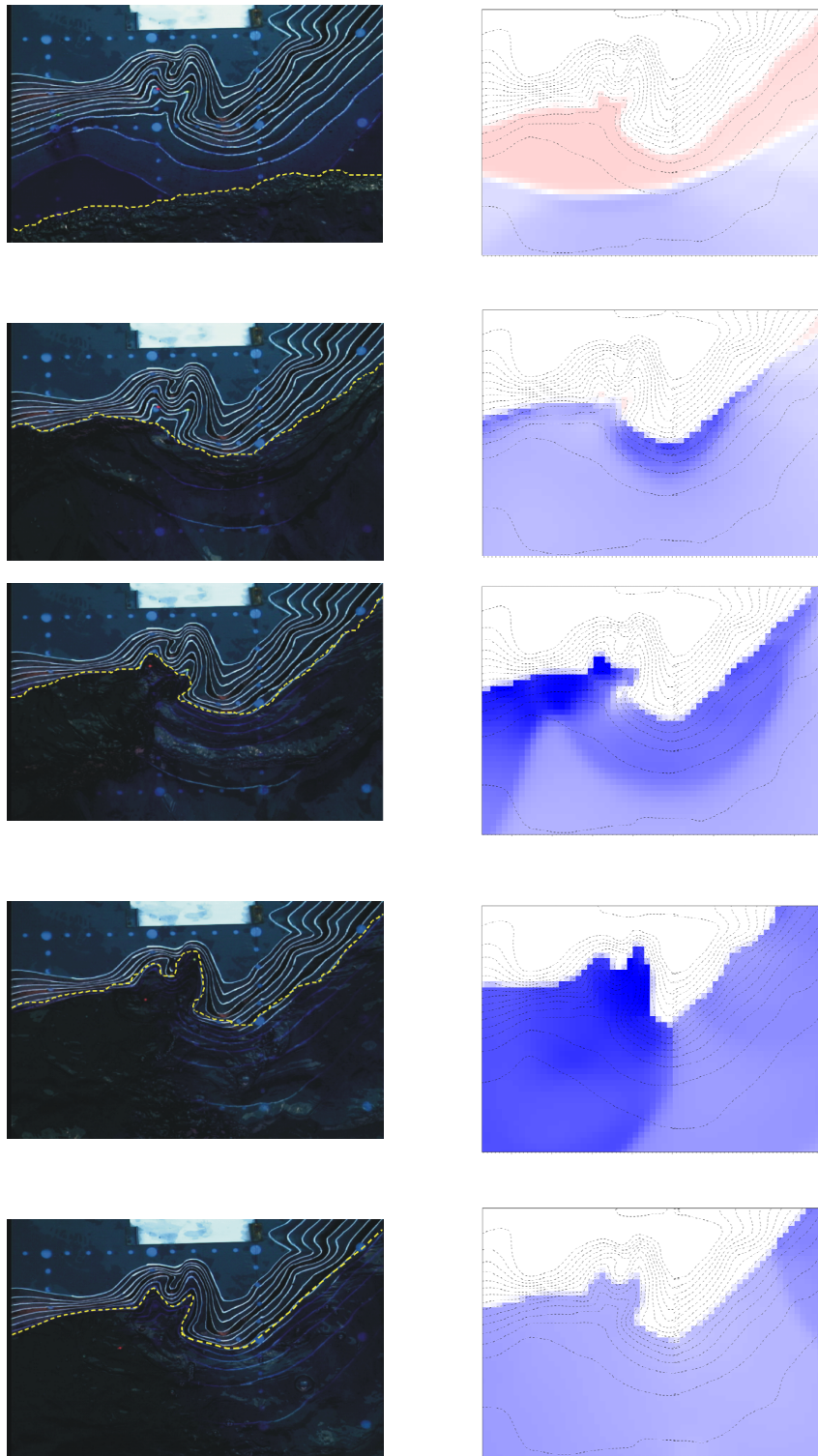
Fig. O Comparison between extracted movie frames (left) and TRITON-G simulation (right) for times 15, 15.5, 16, 16.5 and 17 seconds

Comparison with the extracted movie frames is shown in Fig. O. In the left column are the five frames provided from the laboratory recording. These are frames 10, 25, 40, 55 and 70, extracted from the video with a 0.5 s interval. We found good agreement in time and space for times 15 s to 17 s in 0.5 s increments, shown in the right column. The side-by-side comparison shows that the modeled wave follows the experimental wave front well. Additionally, the model captures the rapid run-up/run-down in the narrow gully.

Finally, the data provided by the benchmark workshop include a series of experiment tests for maximum run-up. Its maximum run-up is recorded at $\times = 5.1575$ and $y = 1.88$ m with an average value of approximately 0.09 m. In comparison, our numerical result recorded a maximum run-up at around $t = 16.5$ with a height of 0.0936 m at $\times = 5.15$ and $y = 1.88$ m.

4 Benchmark Problem #9: Okushiri Island Tsunami - Field

This benchmark problem (BP9) is based on the data collected from the Mw 7.8 Hokkaido-Nansei-Oki tsunami around Okushiri Island in Japan in 1993. The goal is to compare computed model results with the field measurements.

4.1 Problem setup

The following parameters were used for the computation:
- *Bathymetry*: taken from databases provided by (NTHMP, 2012), interpolated where necessary.
- *CFL*: 0.9
- *Simulated time*: 60 minutes
- *Initial condition*: source generated from the database provided by DCRC (Disaster Control Research Center) Japan solution DCRC17a, described in (Takahashi, 1996).
- *Boundary conditions*: open boundaries at the four domain edges.
- *Friction*: Manning coefficient set to 0.02
- *Computational domain*: a mesh refinement is used on the entire domain (shown in Fig. P). Seven levels are used in total. The resolution of base level 1 is 450 m and the resolution of level 7 is approximately 7 m. Dry blocks that did not take part in the computation were removed in the mesh generation process.
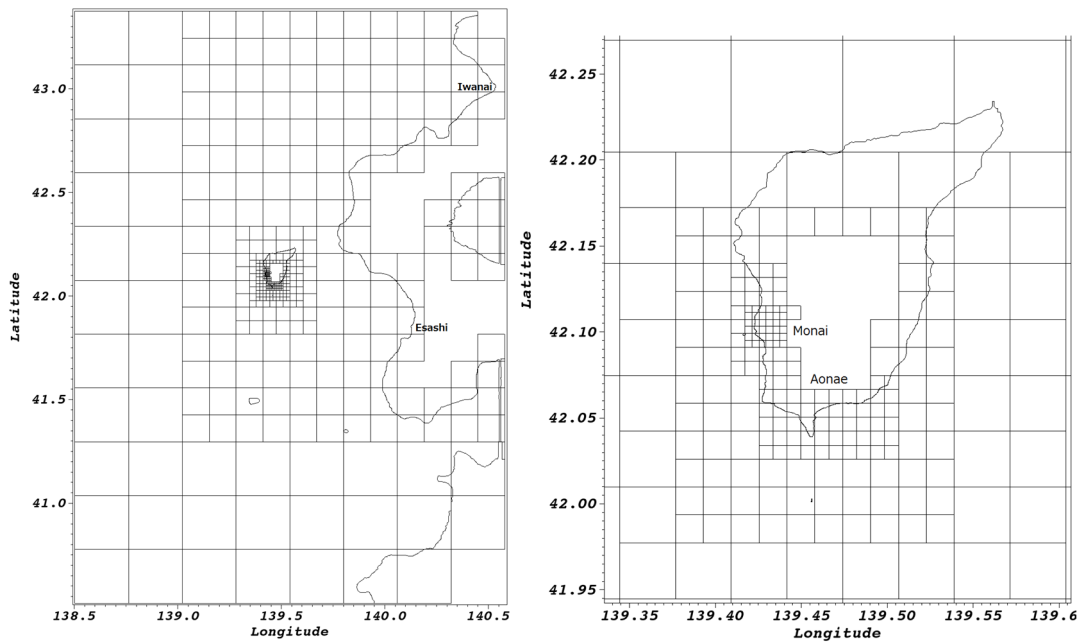
Fig. P Left: entire domain refined mesh containing seven levels. Right: zoom on Okushiri Island. Higher resolution used around Monai Valley at level 7 (7 m approx.) and Aonae region at level 6 (14 m approx.)

4.2 Tasks to be performed

This benchmark requires the following tasks to be performed:

1. Compute run-up around Aonae

2. Compute arrival of the first wave to Aonae

3. Show two waves at Aonae approximately 10 minutes apart; the first wave came from the wet,

the second wave came from the east

4. Compute water level at Iwanai and Esashi tide gauges

5. Maximum modeled run-up distribution around Okushiri island

6. Modeled run-up height at Hamatsumae

7. Modeled run-up height at a valley north of Monai.

4.3 Numerical results

In this section, we present the numerical results obtained with TRITON-G for benchmark problem #9.

4.3.1 Run-up around Aonae

The maximum inundation around Aonae peninsula modeled during the simulation is shown in Fig Q. Contours every 4 meters are drawn to show the outline of the topography. Maximum

inundation height computed was nearly 15 meters but the scale used is set to the upper limit of 10 m to highlight the areas where major inundation occurred.

The west side of the peninsula received the impact of the first wave, which produced the largest inundation height. Maximum values of nearly 15 m were obtained in the simulation. Despite a relatively lower inundation height in the east side of the peninsula, deep penetration was found due to the flatter topography in this area. The inundation on the east side was mainly produced by the second wave coming from the east. The south side of the peninsula experienced the impact of both first and second waves and run-up of over 12 m was estimated.
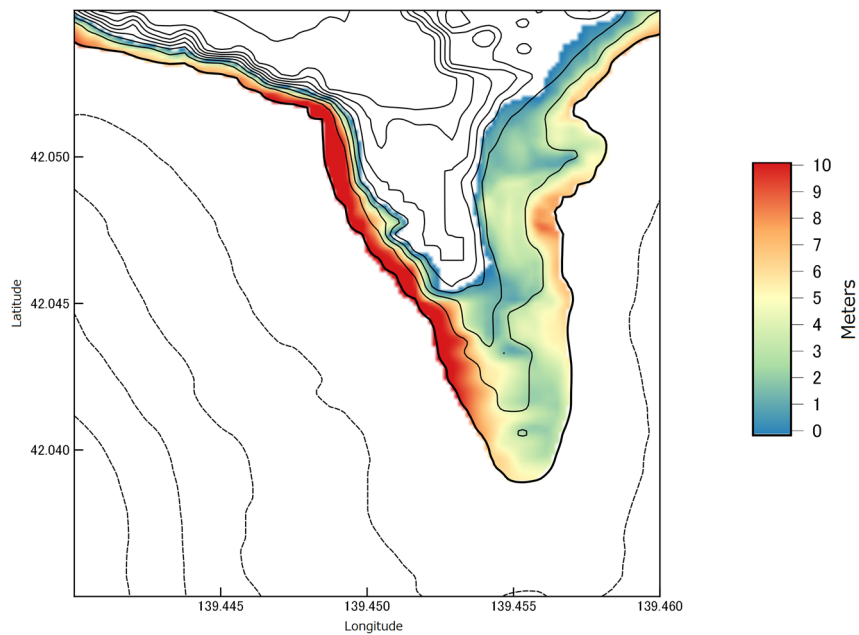
Fig. Q Inundation map of Aonae region with 4-m contours of bathymetry and topography

## 4.3.2 Arrival of first wave to Aonae

The arrival of the first wave at Aonae peninsula is shown in Fig. R. This wave is coming from the west. Snapshots are approximately 5 seconds apart at times 4.9 min and 5.0 min to illustrate the wave arrival. From these snapshots, we estimate that the wave made impact at around 5 minutes after the tsunami generation.
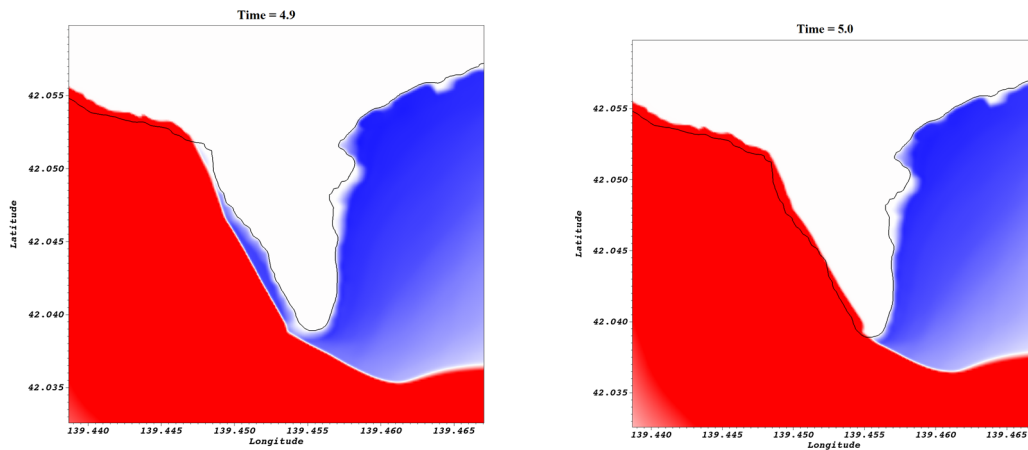
Fig. R Arrival wave at Aonae peninsula coming from the west, snapshots of the wave at times 4.9 min and 5.0 min after tsunami generation

4.3.3 Two waves arriving at Aonae
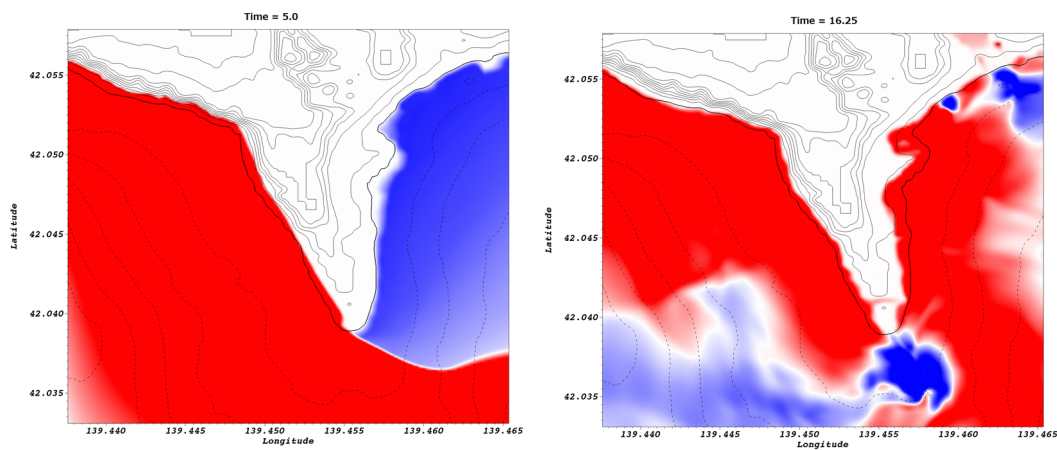
Fig. S Two waves arriving at Aonae peninsula. Left: first wave coming from the west arrived at around t=5 min. Right: second wave coming from the east arrived at around t=16 min

The two waves arriving at Aonae peninsula are shown in Fig. S. The first one came from the west (Fig. S left) and made impact at around 5.0 min after the tsunami generation. The second major wave to hit the peninsula came from the east and made impact at around 16 min (Fig. S right). Slightly over 10 minutes separated the first and second wave.

4.3.4 Tide gauge comparison at Iwanai and Esashi

Comparison between computed and observed water levels at Iwanai and Esashi tide gauges is presented in Fig. T. The arrival time of the computed wave shows good agreement for Esashi station. The computed wave positive and negative phases also follows rather well the observed values. In the

case of Iwanai station the arrival time is slightly sooner than the observed however the observed wave phase is followed generally well in the computed results. The discrepancies between observed and computed values can be attributed to several reasons. Inaccuracies in the source used for the initial condition can influence greatly the result. Additionally, lack of realistic bathymetry including man-made structures around the area can affect the results as well.
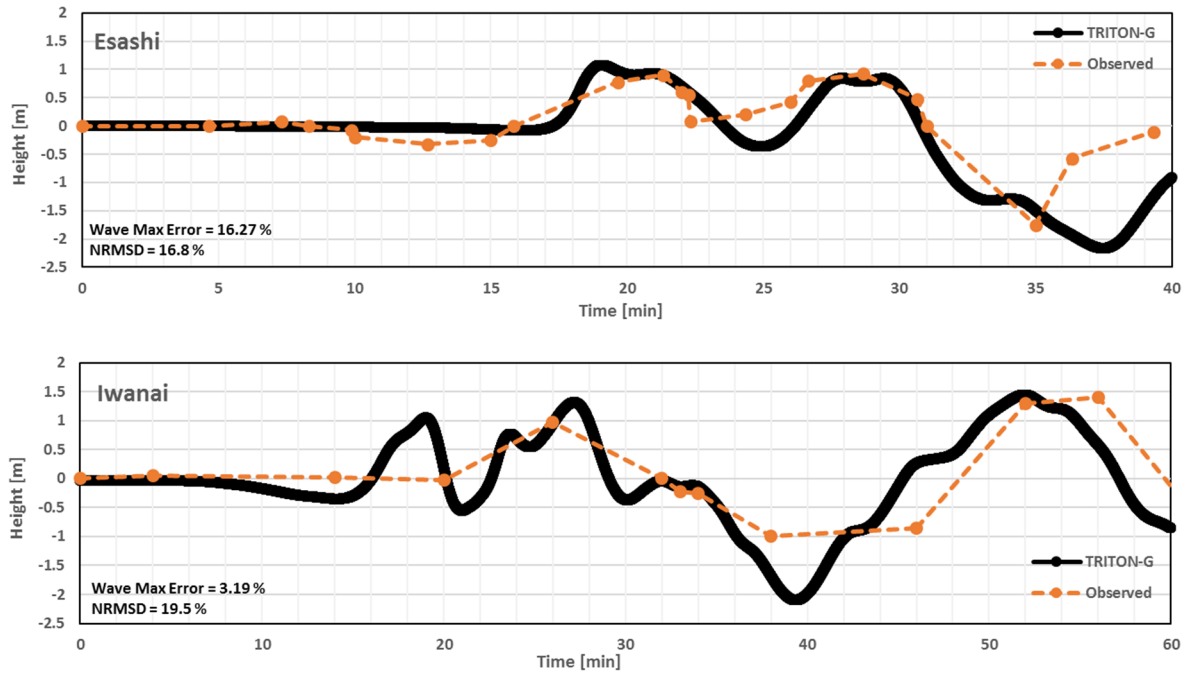

Fig. T Water level comparison between observations and TRITON-G results for Esashi (upper panel) and Iwanai (lower panel) tide gauges.

Inserted in each panel of Fig. T are the estimated errors for the gauge comparison. The maximum wave amplitude error for Esashi station is 16.27% and for Iwanai 3.19%. These are considerably lower than the mean values obtained by the models reported in the workshop (NTHMP, 2012) of 43% and 36% respectively. Although no values are reported in (NTHMP, 2012), the NRMSD error is also estimated for our model and included in the panels, both values are under 20%.

4.3.5 Maximum run-up around Okushiri

The computed maximum run-up distribution around Okushiri Island is shown in Fig. U. Observations were taken from (Kato and Tsuji, 1994). Good agreement is found between observed and computed values around the coast. Most values are within the observed range or within a small difference from the field measurement. The simulation seems to capture well the variations that occurred along the coast.
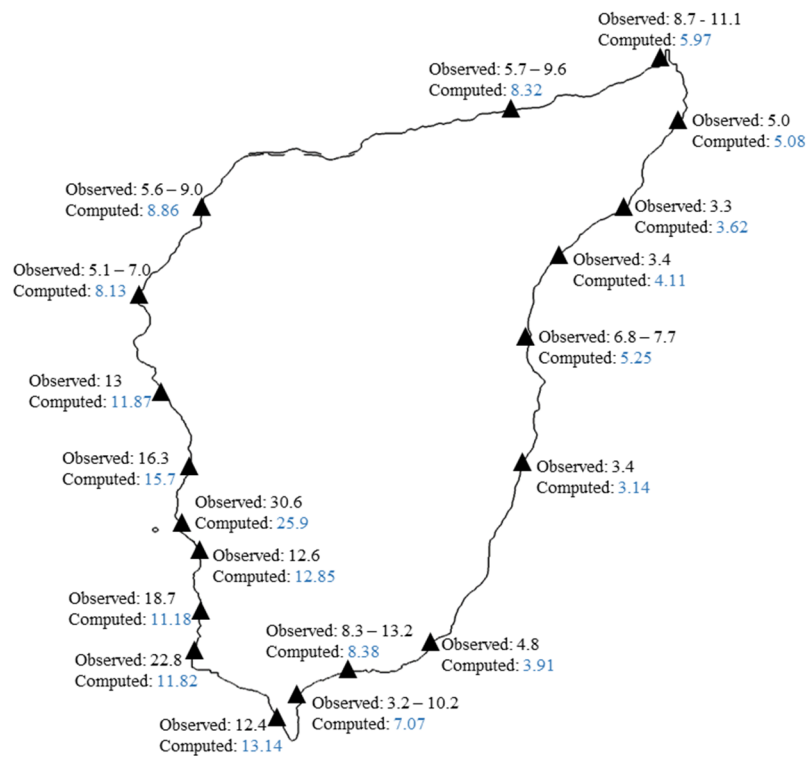
Fig. U Computed and observed run-up values in meters along the Coast of Okushiri Island.

The model could simulate well the maximum run-up observed around Monai valley within a reasonable 15% error. The major differences are found in the southwest side of the island where run-up values were underestimated with larger difference. The discrepancies could be explained by the use of different grid around the island coast. Additionally, the lack of an accurate high-resolution bathymetry database everywhere can also influence the computed values as well as an inaccurate initial condition.
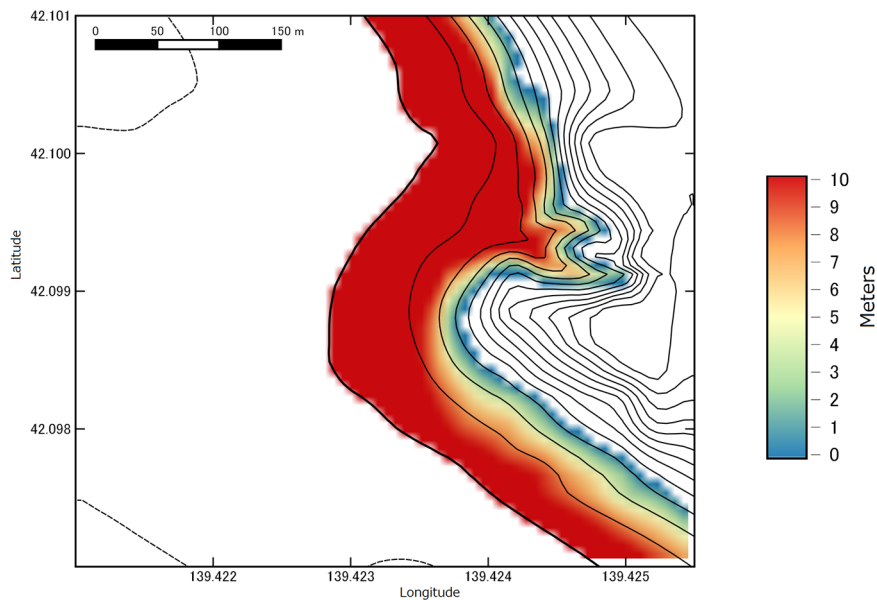
4.3.6 Run-up height at Hamatsumae

The maximum inundation map for Hamatsumae region is shown in Fig. V. Topography and bathymetry contours are outlined every 4 meters. A grid resolution of approximately 14 m was used for this region. Near the center of the region and to the east, run-ups of nearly 16 meters were computed. Additionally, inundation values ranging from 8 to 10 meters were obtained which match well with field observations.
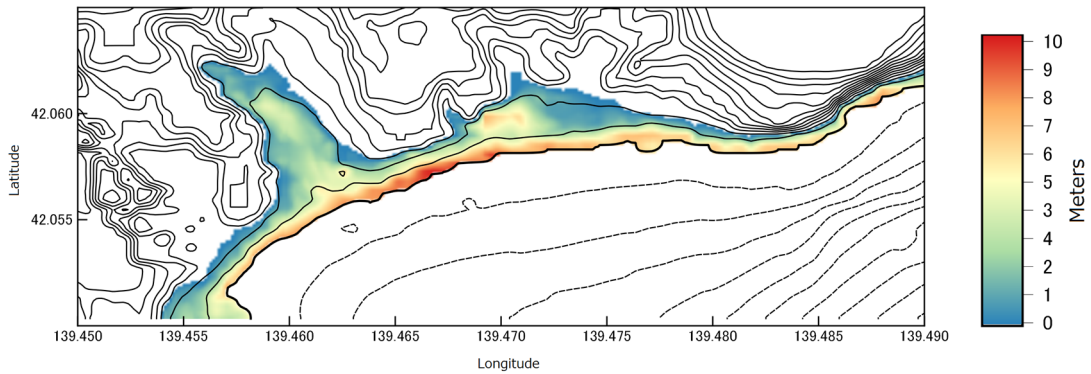
Fig. V Inundation map of Hamatsumae region with 4-m contours of bathymetry and topography.

4.3.7 Run-up height at a valley north of Monai

Fig. W Inundation map for the valley north of Monai with 4-m contours of bathymetry and topography.

The maximum inundation map for the valley north of Monai is shown in Fig. W. Topography and bathymetry contours are outlined every 4 meters. A grid resolution of approximately 7 m was used for this region. Inundation of around 26 m was computed, relatively close to the 30.6 m observed in the field data.

Q8. Regarding the generation process described in section 3.3, authors use the coseismic deformation proposed by Smylie and Manshiha, 1971. I'm surprised at this point because in [Okada, Y. (1985), Surface deformation due to shear and tensile faults in a half-space, Bulletin of the Seismological Society of America, 75(4), 1135-1154] this work is extended, and it's provided also an analytical

way to get the "Okada model" coseismic deformation that can easily be extended to CUDA in the GPU context. I would like to know the reason for not considering Okada as model for the generation process.

A8. Perhaps presenting our model as a *three-step simulation* program might have been misleading. The goal is our model focus on propagation and inundation. Porting to CUDA does not constitute any constraint however the fault generation was beyond our decision. The fault generation and theory used is based on RIMES internal decision. Their fault parameters are submitted to our model directly in order to start the simulation. We modified the manuscript to make this more clear (p.11, line 8). In future work we will focus on studying and including different fault theories such as the Okada model or dynamic generation, ported to GPU.

Section 4

Q9. In section 4, authors use a tree-based mesh refinement similar to the AMR technic used by R.J. LeVeque in GeoClaw. In this case the refinement is not made automatically as in GeoClaw, but it's customized to be refined in coastal areas and focal areas. It's an interesting alternative to the use of nested meshes when more detailed information is required in certain coastal areas.

In p14, line 9-10 it is discussed the number of blocks necessary under the considered resolution and according the 65 x 65 node-centered cells if the number of blocks is 230,000 you would have 971,750,000 node-centered cells. How many information is stored on each node-centered cell (in double precision) to represent over 100GB of memory space?

A9. For that test case, each cell stores approximately 112 bytes in double precision. This includes 14 different values to store information about latitude, longitude, the governing variables (h^t, hu^t, hv^t), the next time-step variables (h^{t+1}, hu^{t+1}, hv^{t+1}), the constant H, the bathymetry (z), manning coefficient and three constant values used for optimization.

Later, the introduction of focal areas reduced the domain mesh size and increased the available memory. Using this freed memory, three more constants were stored per cell. Additionally, the inundation output blocks were stored in GPU memory during the simulation for optimization.

Section 5

Q10. I have some comments, but my main concern is that I think that this section is out of the scope of this journal and my recommendation would be to publish it in a journal more related to this field. Anyway, I will make some comments regarding this section. To my knowledge, the overall GPU

A10. We appreciate your comment and suggestion however, while we understand that the GPU computing is not the scope of this journal we think that including the GPU implementation is very important since is a key element of our research. Tsunami forecasting requires a fast result and there lies the relevance of implementing our GPU calculation. For this reason, we give a brief description about GPU computing and a general explanation of our CUDA implementation and optimizations.

A11. To answer the first part of the question, in general it is stable to use a CFL condition closer to 1. The Semi-Lagrangian scheme used for the propagation stage allows a large time-step. Additionally, the added inundation benchmark problems (see Question 7) use CFL = 0.9, producing good results while being stable.

CFL values closer to 1 tend to produce stable results if the bathymetry varies smoothly. Based on our experience, some instabilities may arise in cases where the bathymetry presents sudden large gradients or very irregular shapes. The Indian Ocean bathymetry used in our research contain several of these cases. A common solution is to smooth the bathymetry before usage. Clearly, this introduces changes in the measurements and simulation. However, in order to keep the results as realistic as possible, we decided not to smooth the bathymetry and instead trade off a higher CFL value.

About the second part of the question, the CFL condition is consistent when using $\Delta t = 1.6$ and numerical stability is ensured in this situation. In order to illustrate this, Table E contains four columns with the values used for the simulation. The second column shows the maximum Δt allowed in each level using CFL = 0.8. In order to speed up the computation, a sub-cycling method was introduced. A global $\Delta t = 1.6$ s is set to calculate the number of cycles that blocks in each level must take. The value of 1.6 is chosen to avoid a sub-cycling overhead since around 80% of the blocks are distributed in levels 1 to 4. The third column in Table E shows the resulting number of sub-cycles per level (*ns*). Finally, the fourth column shows the new CFL values obtained for each level when using sub-cycling (*S.C. CFL*). As it can be seen, in all levels, the CFL condition is less than 1 and stability is ensured.

We included Table E in the manuscript (p. 25, line 10) with an explanation to make clearer that stability is ensured when using sub-cycling.

Level	Max Δt (CFL=0.8)	ns (Δt =1.6)	S.C. CFL
L1	10.71	1	0.12
L2	5.13	1	0.25
L3	2.37	1	0.54
L4	1.65	1	0.78
L5	0.95	2	0.68
L6	0.55	4	0.59
L7	0.26	8	0.39

Table E. CFL values used after introducing sub-cycling (S.C. CFL) for each of the seven levels. The second column shows the maximum Δt per level using CFL = 0.8 and the third column shows the number of sub-cycles (ns) required in each level when using Δt =1.6

Q12. In section 5.3.2 it is showed the runtime performance by comparing results obtained with two different GPU architectures: Tesla K80 and Tesla P100. It must be remarked that the configuration of the computation nodes that you are using is different for each architecture. While the Tesla P100 nodes have 4 GPU's per node, the Tesla K80 nodes have 2 GPU's per node. Unless your tests are using only two P100 per node, the network communications will reduce the performance between the K80 in front of P100. What kind of network is being used in these clusters?

A12. In the case of *Tsubame 3.0,* there are four Tesla P100 GPUs per node and the network is Intel Omni-Path HFI 100Gbps. In the case of the K80 machine used, there are four cards in one node (eight GPU in total), connected through PCI-Express 3.0. Confusion might arise from the term "Tesla K80 (12GB×2)". The values in parenthesis refer to the memory distribution. One K80 card includes two GPU chips, each with 12 GB. We modified the manuscript to describe this machine specification more clearly (p.26, lines 12 and 13).

We agree that the performance might be different depending on the GPU/node distribution. The aim on comparing between these machines is to demonstrate the portability of our code from an older architecture to a much newer one. Additionally, it serves to show a more general performance results by using more than one more machine and to highlight that the code can be implemented on different hardware without requiring changes or creating any problem (p.26, line 15).

Another point is, are output data being stored into hard disks in these tests or are they related only to pure computation time?

A13. All runtime measurements include the output time storing into hard disks. We included a mention to this fact in the revised manuscript (p.27, line 7).

Section 6

Q14. Finally, my main comments about section 6 have been included in my recommendations for section 3.

A14. Once again, we thank the reviewer for their important recommendation. We followed the advice given about section 3 and the full answer including the results of the new benchmark problems can be consulted in Question #7 (Q7).

Q15. Anyway, I have some specific comments about this section. It would be nice if the color scale used in Fig. 18 and 19 are the same for each subplot. With these graphics we can compare the inundation extensions, but it's difficult to compare the inundation height when different scales are used on each graphic. On other hand, and as I pointed before, this test is not enough to say that the model is validated, so I don't agree with the sentence of p33, lines 8-9.

A15. The subplots in Fig. 18 and 19 have been redone to match better the scales for comparison. The new subplots are shown in Fig. X and Fig. Y respectively. The new figures are included in the revised manuscript (p.42, Figure 29).

We have modified the sentenced in p33 about the inundation validation using just one test. It is now noted (p.43, line 19) that the additional standard inundation benchmark problems computed produced good results and served as complementary demonstration of TRITON-G's ability to estimate tsunami inundation. A mention to the new benchmark results are also included in the conclusion as support for the validation (p.44, line 7).
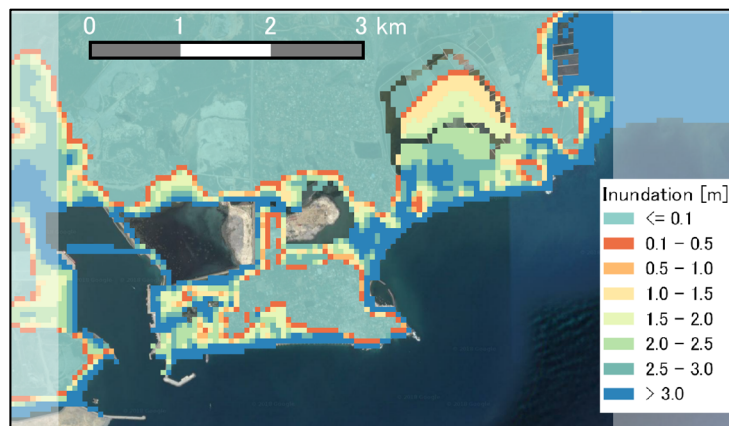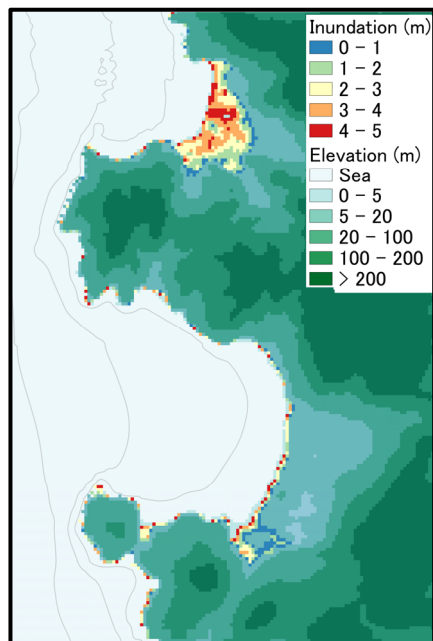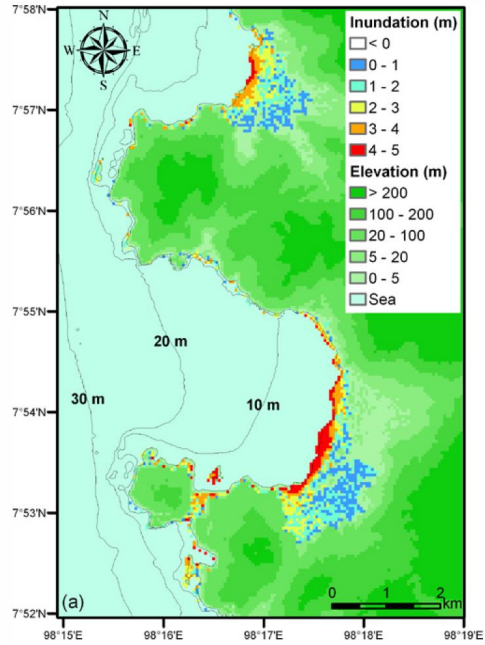
Fig X. Inundation comparison for Hambantota, Sri Lanka. Top: RIMES model. Bottom: TRITON-G model.

Fig Y. Kamala (North) and Patong (South) inundation maps comparison. Top: inundation result by (Supparsri et al., 2011)**. Bottom: TRITON-G inundation result.**

References

Kato, K. and Tsuji, Y.: Estimation fo fault parameters of the 1993, Hokkaido-Nansei-Oki earthquake and tsunami characteristics., Bull. Earthq. Rest. Inst., 69, 39–66, 1994.

Liu, P. L., Yeh, H. and Synolakis, C.: Advanced numerical models for simulating Tsunami waves and runup. Advances in coastal and ocean engineering., New Jersey: World Scientific, 10, 2008.

NTHMP: NTHMP, 2012. National Tsunami Hazard Mitigation Program (NTHMP). 2012. Proceedings and Results of the 2011 NTHMP Model Benchmarking Workshop, Department of Commerce/NOAA/NTHMP; NOAA Special Report., 2012.

Supparsri, A., Koshimura, S. and Imamura, F.: Developing tsunami fragility curves based on the satellite remote sensing and the numerical modeling of the 2004 Indian Ocean tsunami in Thailand, J Natural Hazards and Earth Sc., 11, 173–189, 2011.

Synolakis, C. E.: The runup of solitary Waves, J. Fluid Mechanics, 185, 523–545, 1987.

Synolakis, C. E., Bernard, E. N., Titov, V. V., Kânolu, U. and González, F. I.: Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models., NOAA., 2007.

Takahashi, T.: Benchmark Problem 4. The 1993 Okushiri tsunami. Data collected, Conditions and Phenomena, in Long waves runup models, edited by H. Yeh, P. Piu, and C. Synolakis, pp. 384–403, Word Scientific Publishing Co., 1996.

Yeh, H., Liu, P., Briggs, M. and Synolakis, C.: Propagation and amplification of tsunamis at coastal boundaries, Nature, 372, 353–355, 1994.

Corollary

Author's comment: Additional to all the reviewers' suggestions, we decided to remove the paragraph about the circular shoal benchmark from the original manuscript in page 26, from line 11 to 17. With the introduction of several new benchmark problems (Reviewer #1 Question #7) and the modifications to the original manuscript, it felt unnecessary to keep this reference since the new results covered far more than what this benchmark offered.

Response to Reviewer #2

We want to thank the reviewer for using their precious time to check our paper and for giving useful comments to improve our work. Please find below our replies to your comments.

General Comments

Q1. The paper presents an operational model for the fast simulation of the generation, propagation and inundation of tsunamis in wide areas by exploiting modern multiGPU hardware. The model is tested and compared under a real tsunami scenario, obtaining a nice performance results from the operational point of view. The implementation of this operational model on a cluster of multiGPU computers involves the suitable integration of numerical schemes (MOC with dimensional splitting to solve spherical SWE for the Tsunami propagation, Surface Gradient Method to solve the cartesian SWE for inundation areas, ...) and computing techniques (quadtree-based mesh refinement to save resources, Hilbert Space-filling curves to preserve locality in the parallel partitioning, CUDA for GPU programming and MPI for remote communication, overlapping the computation in GPU and the generation of output files and rendering in CPU, etc.) to obtain an efficient complete CPU-multi-GPU operational model for Tsunami forecasting. This model would make it possible a very fast simulation which can help in the early identification of the tsunami consequences. In my opinion, the techniques which are presented and the scientific data which are included are coherent and relevant and can be useful to scientists working in this area because all the approaches and techniques are devised in conjunction to perform very quickly realistic simulations. Although the paper is well organized and written, the reading of several pieces of the sections which explain the multiGPU implementation is not easy to understand and several implementation decisions which are presented are not clear. Moreover, in Section 5, I think that the use of technical and English language should be checked (several corrections are included in the Section of Technical Corrections).

A1. We appreciate your kind description of our work. We considered your suggestion to check the use of technical English in section 5 (p.17) and made the section about multi-GPU implementation more clear where possible. We appreciate your technical corrections in this matter as well.

It should be noted that even though one of the key elements of our work is GPU computing, the scope of the journal is not this area. For this reason, we tried to find a balance giving an appropriate description of our implementation without getting into extensive details. We tried to focus on the model and on the simulation results.

A2. In order to have more validation data we have now included two sections in the manuscript to show results of several standard benchmark problems. Not having accurate enough initial input data is always an issue for all tsunami simulation models. Currently, the best approach to validate models consists on comparing results with existing analytical solutions and experimental data. For this reason we follow the NOAA Technical Memorandum OAR PMEL-135 (Synolakis et al., 2007) where several standard benchmark problems (BP) are given. We also thank Reviewer #1 for this suggestion.

Section 3 presents the results for the analytical 2D Parabolic Bowl benchmark. A new section was added to the manuscript to include results for benchmark problem 9 (p.28, Section 6) while the rest of the benchmarks are located in the appendix (p.44). The benchmark problems added are:

- BP4, Solitary wave on a simple beach
- BP6, Solitary wave on a conical island
- BP7, The tsunami run-up onto a complex 3D beach
- BP9, Okushiri Island Tsunami

Specific Comments

A3. We appreciate the suggestion; however, we consider that this kind of comparison would be unfair to do. For instance, each model utilizes different numerical schemes; the machines used might have different specifications; the domain used in each case might be different and in our case, we used an AMR-like technique for mesh refinement when grid nesting is more common.

However, we have included as references in section 1 (p.2, at lines 8,13, 16 and 27) other works that use GPU in their simulations. Two examples are:

- Vazhenin, A., Lavrentiev, M., Romanenko, A. and Marchuk, A.: Acceleration of tsunami wave propagation modeling based on re-engineering of computational components, International Journal of Computer Science and Network Security, 13, 32–70, 2013.
- Macías, J., Castro, M. J., Ortega, S., Escalante, C. and González-Vida, J. M.: Performance benchmarking of Tsunami-HySEA model for NTHMP's inundation mapping activitie, Pure and Applied Geophysics, 174, 3147–3183, 2017.

Additionally, we have included new benchmark results in the manuscript (Section 6 and Appendix) that are considered standard in the tsunami field. Using the data presented in the National Tsunami Hazard Mitigation workshop (NTHMP, 2012), error comparison with results of other models was included in our discussion and figures when available.

Q4. In Section 5.1., the description of the configuration of the main CUDA kernels (second paragraph of the section 5.1.) is not easy to understand. A graphical description of the configuration and a description of the calculations assigned to each CUDA thread (relating this section with section 3 and 4) would be very useful to understand it.

A4. A graphical description of the CUDA kernel, shown in Fig. A, has been added to the revised manuscript (p.18, Fig. 7). A more clear explanation was included as well (p.18, lines 9-16).

The grid is composed of 16 CUDA blocks in *y*-direction, each with 4 threads for 64 threads in total. In *x*-direction, the grid has one CUDA block with 64 threads.

One CUDA block processes a portion equal in size of the mesh block. One CUDA thread computes one mesh block cell. The specific calculation varies depending on the block type (*Wet*, *Dry*); however, the configuration remains the same. In both cases, threads compute the governing equations described in section 3.1. The main difference occurs in the case of a *Dry* block; in this case, cells that represent land or coastline compute a reflective wall boundary.
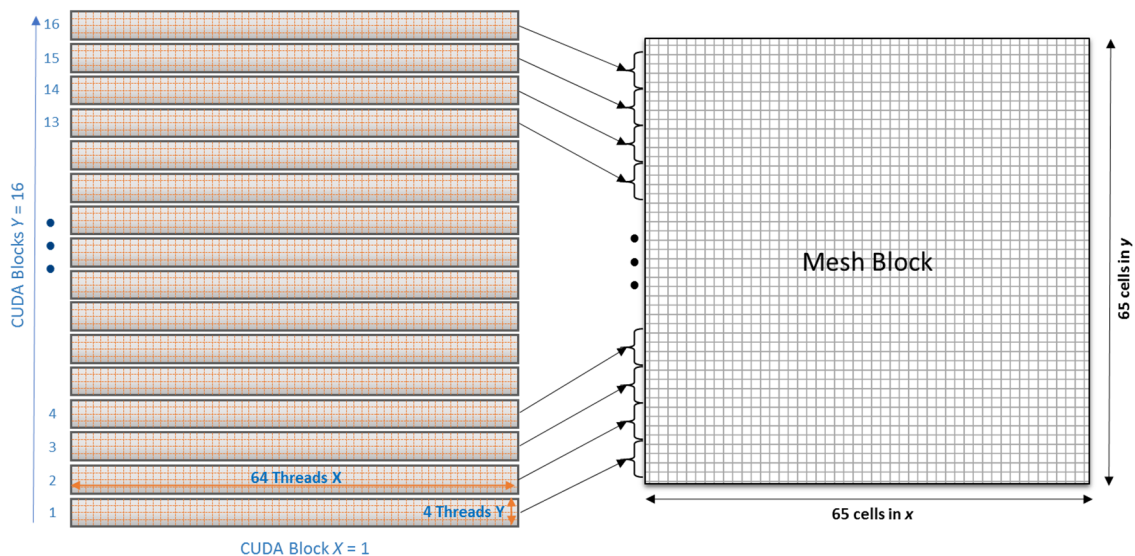

Fig. A Mesh block computation using CUDA kernels. Each CUDA block is made of 64×4 threads and computes a portion of the mesh block. One CUDA thread computes one mesh block cell

Q5. The specialized kernel types presented in section 5.1.2. could affect the load balancing between GPUs because the computational execution cost of each kernel type on a mesh block would be possibly different. I do not know if this fact is taken into account for the considerations included in section 5.2.

A5. Thank you for pointing this detail. This fact is taken into account during the load balancing, by assigning a different weight to the space-filling curve (SFC) based on the block type. This was not mentioned explicitly in the manuscript. A mention to this has been added in section 5.1.3 where SFC weights are discussed with sub-cycling (p.25, line 24).

Q6. I think it would be interesting to report graphically execution times for each particular GPU in order to evaluate the effectiveness of the domain partitioning and even to rethink the approach by designing a dynamic load balancer.

A6. Without being a detailed dynamic load balancer, our model includes this feature. During the mesh generation, blocks are assigned a different weight based on its type and the sub-cycling number. This weight is used in the space-filling curve to find a good domain partition. Not being a static process, this means that if a new domain mesh is required, the program will balance the new load.
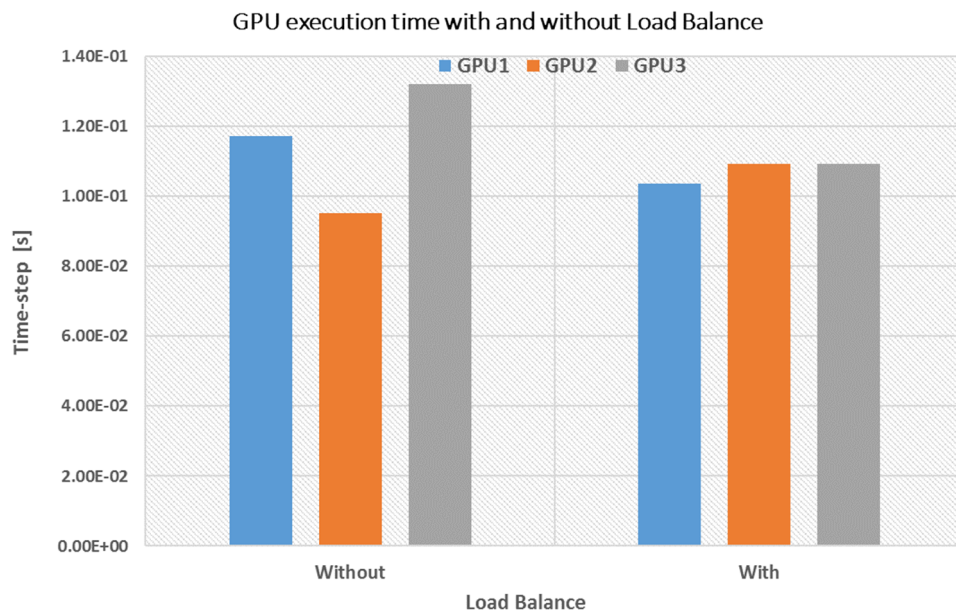
Fig. B GPU execution time with and without load balance

Evidently, the load balance is problem dependent. However, we include a chart show in Fig. B with the balancing results for our Indian Ocean domain using 4 focal areas (p.26, Fig 15). The effect of including the load balance can be seen on the right side of the chart. All GPUs spend almost the

same time to execute a time-step. This avoids large overheads created by one GPU idling waiting for another to complete the tasks (p.26, line 1).

Q7. In Section 5.3.2., the configuration for the network which interconnects the TeslaP100- based nodes and the Tesla K80 nodes should be included to analyze Fig. 15.

A7. In the case of *Tsubame 3.0,* there are four Tesla P100 GPUs per node and the network is Intel Omni-Path HFI 100Gbps. In the case of the K80 machine used, there are four cards in one node (eight GPU in total), connected through PCI-Express 3.0. These network configurations have been added to the manuscript (p.26, line 12 and 13).

Q8. In Section 5.3.2. and in the Conclusions, authors underline evidences about the wall clock time and the speedup which are obtained with 3 GPUs. However, the particular performance results for 3 GPUs are not reported and they are not included in Figure 15.

A8. Thank you for noticing this detail. The runtime for 3 GPUs with K80 cards is 39.96 min and 12.1 min with P100 cards. These values have been now reported in the manuscript in section 5.3.2 (p.28, Fig 17 and p.27, line 15). Additionally, they have been included in Figure 15; the modified figure can be seen in Fig. C.
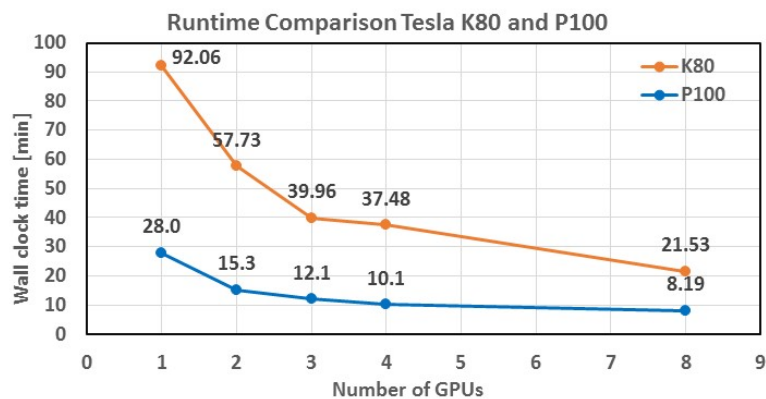
Fig. C Wall clock comparison of 10-hour simulation on Tesla K80 and Tesla P100

Q9. In Section 5.3.2., absolute performance measures on one GPU for the main kernels (it can be obtained by using the Nvidia CUDA profiler) could be useful to evaluate the efficiency of the CUDA implementation.

A9. We measured the FLOP/s performance of the main kernels for one GPU. The results obtained are shown in Table A, where *Inund* stands for Inundation kernel, *Wall* stands for the wall kernel, *Wet* for the *Wet* kernel and *X* and *Y* for the direction of the computation equivalent to longitude and latitude respectively. These results have been added to the revised manuscript (p.27, line 1 and p.27 Table 2).

Kernel	GFLOP/S
WallX	549.57
WallY	549.56
WetX	706.98
WetY	712.51
Inund	87.12

Table A. Kernel performance for one GPU in Giga FLOP per second

Q10. In section 6.1.2., the figure 18 Top which presents the RIMES results for Hambantota Inundation is not introduced textually in page 31, line 18. On the other hand, it is difficult to compare visually the RIMES and TRITON G inundation maps if the colours are used in a completely different manner for each map.

A10.
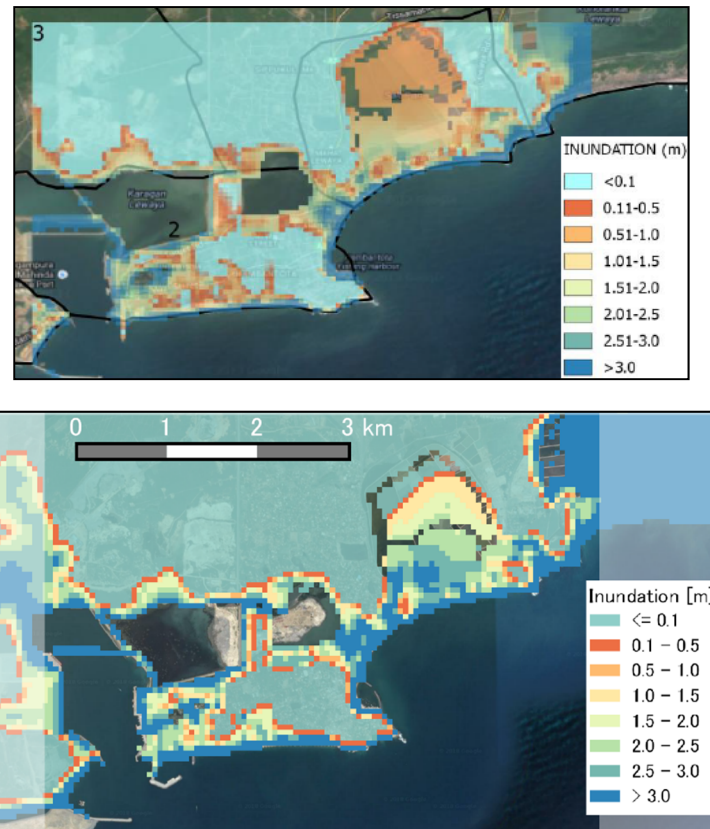
Fig. D Inundation comparison for Hambantota, Sri Lanka. Top: RIMES model. Bottom: TRITON-G model.

We reordered and modified the paragraph in page 31 to introduce textually RIMES' figure (p.41, line 7).

Additionally, we modified TRITON-G figure's color scale to make it match better with RIMES' scale (p.41, Fig 28). The result is shown in Fig D in these notes.

A11. Changed sentence accordingly (p.13, line 17).

A12. Sentences were rewritten to make them more clear (p. 15, line 17).

"Additionally, all dry blocks at Level 7 (highest resolution) that are inside a FA are considered inundation areas. This implies that run-up is computed on the coastlines instead of using a reflective boundary."

A13. Sentence rewritten (p.17, line 28).

 "CUDA provides kernels as the way to define functions that are executed in parallel on GPU."

A14. Changed sentence accordingly (p. 17, line 28).

A15. Thank you for the suggestion. Changed sentence accordingly (p.19, line 8).

A16. Changed sentence accordingly (p. 19, line 12).

Q17. - Page 18, Line 18: "... This way the kernels can be launched ...".

A17. Changed sentence accordingly (p.19, line 18).

Q18. - Page 19, Line 5: "... is illustrated in Fig. 7 ...".

A18. Changed sentence accordingly (p.20, line 5).

Q19. - Page 23, Line 5: "... was introduced in order ..."

A19. Changed sentence accordingly (p.24, line 5).

References

NTHMP: NTHMP, 2012. National Tsunami Hazard Mitigation Program (NTHMP). 2012. Proceedings and Results of the 2011 NTHMP Model Benchmarking Workshop, Department of Commerce/NOAA/NTHMP; NOAA Special Report., 2012.

Synolakis, C. E., Bernard, E. N., Titov, V. V., Kânolu, U. and González, F. I.: Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models., NOAA., 2007.

Corollary

Author's comment: Additional to all the reviewers' suggestions, we decided to remove the paragraph about the circular shoal benchmark form the original manuscript in page 26, from line 11 to 17. With the introduction of several new benchmark problems (Reviewer #1 Question #7) and the modifications to the original manuscript, it felt unnecessary to keep this reference since the new results covered far more than what this benchmark offered.

Tree-based mesh-refinement GPU accelerated tsunami simulator for real time operation

Marlon Arce Acuña[1], Takayuki Aoki[2]

[1]Department of Nuclear Engineering, Tokyo Institute of Technology, 2-12-1-i7-3, Ookayama, Meguro, Tokyo, Japan
5 [2]Global Scientific Information and Computing Center, Tokyo Institute of Technology, 2-12-1-i7-3, Ookayama, Meguro, Tokyo, Japan

Correspondence to: Marlon Arce Acuña (marlon.arce@sim.gsic.titech.ac.jp)

Abstract. This paper presents a fast and accurate tsunami real time operational model to compute across-ocean wide-simulations completely on GPU. The spherical shallow water equations are solved using the method of characteristics
10 and upwind cubic-interpolation, to provide high accuracy and stability. A customized, user interactive, tree based mesh refinement method is implemented based on distance from the coast and focal areas to generate a memory efficient domain with resolutions of up to 50m. Three GPU kernels, specialized and optimized (wet, wall and inundation) are developed to compute the domain block mesh. Multi-GPU is used to further speed up the computation and a weighted Hilbert space filling curve is used to produce balanced work load. Hindcasting of the 2004 Indonesia tsunami is presented to validate and
15 compare the agreement of the arrival times and main peaks at several gauges. Inundation maps are also produced for Kamala and Hambantota to validate the accuracy of our model. Test runs on three Tesla P100 cards on Tsubame 3.0 could fully simulate 10 hours in just under 10 minutes wall clock time.

1 Introduction

The turn of the 21st century showed us as never before the reality of the terrible and devastating damage and death that
20 tsunamis can cause. In 2004, a massive earthquake of magnitude M9.0 on the Richter scale off Sumatra Island triggered a tsunami with deadly consequences. According to the World Health Organization the death toll for these events exceeds 200,000 (WHO, 2014) in several countries spread along the Indian Ocean. Not much later in 2011 a tsunami triggered by a M9.0 earthquake on the east coast of Japan produced in the Tohoku region yet another disaster. Over 15,000 people died from these events with massive destruction in port and city infrastructure, housing, tele-communications and the subsequent
25 nuclear crisis due to the tsunami-induced damage of several reactors in the Fukushima nuclear power plant (Motoki and Toshihiro, 2012).

These events highlight the importance of developing accurate and fast tsunami forecasting models. For several decades, efforts have been made to develop such models. These can be classified in two main groups: depth-average, hydrostatic and non-hydrostatic long wave equations. Hydrostatic models for the shallow water equations (SWE) started by solving their
30 linear form based on finite difference methods (FDM) taking after the work of (Hansen, 1956) and (Fischer, 1959) in the 1950s.

1

The model TUNAMI (Tohoku University's Numerical Analysis Model for Investigation) (Imamura et al., 1995) came from these initial steps but solved the shallow water equations in a non-linear form instead, formulated in a flux-conservative way for mass conservation and also introduced a discharge computation (Imamura, 1996) for the elevation near the shoreline. In a very similar manner the ALASKA-tectonic and Landslide models (GI'-T) (GI'-L) were introduced, also

5   solving the non-linear shallow water and using leapfrog FDM (Nicolsky et al., 2011) as TUNAMI. Later came MOST (Method of Splitting Tsunami) (Titov and Synolakis, 1995), an extensively used model for tsunami simulation, that tried to incorporate the effect of dispersion during simulation (Burwell et al., 2007), also it was original by introducing a function to add points in the shoreline to keep better tracking. Recently MOST has been ported for GPU computing (Vazhenin et al., 2013). A more recent model is GeoClaw which implements a unique approach to deal with the issue of transferring fluid

10   kinematic throughout nested grids by refining specified cells during simulation getting better resolution in those areas (Berger and LeVeque, 1998). More recent models incorporate a real-time application such as RIFT (Real-Time Inundation Forecasting of Tsunamis) (Wang et al., 2012). Like several of the previous models a leap-frog scheme is also used for these real-time models and a linear SWE is solved in certain areas for lighter computation. COMCOT from Cornell University is another example using this approach (Liu, 1998). EasyWave is another known model (Babeyko, 2017), which employs

15   linear approximations for speed up and leap-frog scheme as its numerical scheme. The latest version of EasyWave introduced GPU to accelerate parts of the existing CPU code. More recently, GPU-based models have been developed like NAMI DANCE (Zaytsev et al., 2006) in its latest version. Also a better known GPU model is TsunamiHySEA (Macías et al., 2017) which has been extensively tested and currently used by the Centro di Allerta di Tsunami (CAT) in Italy.

       In order to include the effect of pressure, since the 1990s some models took the direction of solving non-hydrostatic

20   models using the depth-integrated Boussinesq equations (BE) instead of the SWE for tsunami propagation. Initial efforts considered a weak nonlinear model (Peregrine, 1967) however, models for the nonlinear equations were also developed not long after, for instance (Nwogu, 1993), (Lynett et al., 2002). Solving the Boussinesq equation is in general more computationally demanding than solving the SWE and in order to reduce the computational time some techniques have been implemented, such as using parallel clusters or introducing nested-grids. An example of this is FUNWAVE-TVD (Shi et al.,

25   2012), which is an extended version of FUNWAVE, a run-up and propagation model based on fully nonlinear and dispersive Boussinesq equations (Wei et al., 1995). FUNWAVE introduced a nested grid method and its later version has been fully parallelized using MPI-FORTRAN. A well-known non-hydrostatic model which implements also two-way grid nesting is NEOWAVE (Yamazaki et al., 2011). Another of these models is BOSZ (Roeber and Cheung, 2012)), which combines the dispersive effect from the BE with the shock capturing ability of the nonlinear SWE. BOSZ is mainly used for near-shore

30   simulation since is based on Cartesian coordinates and not suited for large areas, also it does not implement nested grids. Recently, efforts to solve the modelling equations in three dimensions have been made as well. Although these models tend to capture difficult coastlines very well and can include multiple fluids or even materials, the computation cost is still so great that it makes it possible to apply them effectively only in small areas and not viable for transoceanic propagations.

Some examples are SELFE (Semi-Implicit Eulerian-Lagrangian Finite Elements) (Zhang and Baptista, 2008), (Abadie et al., 2010), (Horrillo et al., 2013) and (Abadie et al., 2012).

In this work we present a new approach for a tsunami operational model that retains a high degree of the complexities of the physics involved, and delivers a fast and accurate simulation. This speed also enables real-time operation: a user can
5 start forecasting simultaneously as a tsunami event occurs. Results are generated faster-than-real-time. The main goal is to accomplish a wide-area, ocean-size, computation in short time and using resources efficiently. Our model, referred to hereinafter as *TRITON-G (Tsunami Refinement and Inundation Real-Time Operational Numerical Model for GPU),* implements a full-GPU computing approach for the whole tsunami model, composed of generation-propagation-inundation. Specialized kernels are developed for each part of the tsunami computation and multi-GPU is used for further acceleration.
10 Load balance is obtained using a weighted Hilbert space filling curve. TRITON-G solves the non-linear spherical shallow water equations across the entire domain to preserve the complexity of the propagation and effects near the coastline. The method of characteristics with directional splitting and a 3rd Order Interpolation Semi-Lagrangian numerical scheme is used to solve the governing equations. This allows high accuracy and minimizes effects of numerical dispersion and diffusion, also give the ability of choosing a larger time step compared to that of using a Runge-Kutta scheme and at the same time
15 permits a light stencil suitable for fast computation. We implement a tree-based block refinement to generate a computational mesh that is flexible, light and can track complex coastlines. Customized refinements by distance and focal area were developed, which permits an efficient use of memory and computational resources. In a collaboration project with RIMES (Regional Integrated Multi-hazard Early Warning System) (Regional Integrated Multi-Hazard Early Warning System, 2017) we utilize their existing databases for bathymetry and fault sources where available, and successfully
20 deployed TRITON-G as their tsunami forecast operational model.

This article is organized as follows, a review of the governing equations is given in Section 2. The numerical method and boundaries are explained in Section 3. In Section 4 a description of tree-based refinement and its customization is given. Topography and bathymetry used are also described. GPU and parallel computing is covered in Section 5. In Section 6 we present comparison results with a known benchmark inundation problem. In section 7 we present several numerical results
25 including TRITON-G validation with existing tsunami propagation data and run-up measurements. Section 8 presents the conclusions of this study. Results from several standard inundation benchmark problems are included in the appendix.

2 Governing Equations

The spherical non-linear shallow water equations (SSWE) are used to compute the tsunami propagation. In specific and small areas where inundation needs to be computed, the Cartesian coordinate version of the SWE are solved instead, see
30 (Toro, 2010). The SSWE (Williamson et al., 1992), (Swarztrauber et al., 1997) can be written as

$$\frac{\partial h}{\partial t} + \frac{1}{a\cos\theta}\frac{\partial}{\partial\lambda}(hu) + \frac{1}{a}\frac{\partial}{\partial\theta}(hv) - \frac{hv}{a}\tan\theta = 0,$$

$$\frac{\partial hu}{\partial t} + \frac{1}{a\cos\theta}\frac{\partial}{\partial\lambda}\left(hu^2 + \frac{g}{2}h^2\right) + \frac{1}{a}\frac{\partial huv}{\partial\theta} - \frac{hv}{a}\tan\theta - \left(f + \frac{u}{a}\tan\theta\right)hv + \frac{gh}{a\cos\theta}\frac{\partial z}{\partial\lambda} + \tau_\lambda = 0, \tag{3}$$

$$\frac{\partial hv}{\partial t} + \frac{1}{a\cos\theta}\frac{\partial hvu}{\partial\lambda} + \frac{1}{a}\frac{\partial}{\partial\theta}\left(hv^2 + \frac{g}{2}h^2\right) - \frac{hv^2}{a}\tan\theta + \left(f + \frac{u}{a}\tan\theta\right)hu + \frac{gh}{a}\frac{\partial z}{\partial\theta} + \tau_\theta = 0$$

where λ stands for the longitude coordinate, θ for the latitude coordinate, h is the water depth, hu and hv are the momentum in longitude and latitude respectively with corresponding velocities u and v, g is gravity, a is the radius of the Earth, z is the bathymetry (submarine topography), f is the Coriolis force defined as $f = 2\Omega\sin\theta$ with Ω being the rotation rate of the Earth and τ is the bottom friction term. The bottom friction is determined using the Manning formula

$$\tau_\lambda = \frac{gn^2}{h^{7/3}}hu\sqrt{(hu)^2 + (hv)^2},$$

$$\tag{4}$$

$$\tau_\theta = \frac{gn^2}{h^{7/3}}hv\sqrt{(hu)^2 + (hv)^2}$$

where n is the Manning's roughness coefficient, the default value used for n is 0.025 across all domain except for specific areas where more detailed values in the coastline are given in a database. The parameters used in this work are $a = 6.37122 \times 10^6\ [m]$, $\Omega = 7.292 \times 10^{-5}\ [s^{-1}]$ and $g = 9.81\ [ms^{-2}]$.

3 Numerical methods and boundary conditions

3.1 Methods of characteristics for SSWE

The SSWE are solved using the method of characteristics (MOC). A method developed in the 1960s, explained in detail by Rusanov (Rusanov, 1963). MOC is applied to reduce hyperbolic partial differential equations, such as the SSWE, to a family of ordinary differential equations. A traditional approach when using MOC is to introduce a dimensional splitting (Nakamura et al., 2001) in the 2-dimensional equations to create a smaller stencil and lighter computation. A numerical scheme is regarded as well-balanced, or satisfying the C-property (Bermúdez and Vázquez, 1994) if it preserves steady states

at rest, for instance, the undisturbed surface of lake. When the fluid is at rest i.e. $u(x, t) = 0$ then the constant water height H defined as $H(x, t) = h(x, t) + z(x)$ represent a steady state that should hold in time and not produce spurious oscillations (LeVeque, 1998). In order to make the model well-balanced, the SSWE equations are solved for H during the simulation to guarantee this steady state. The original variable h is simply obtained back from the expression $h = H - z$.

5   In order to apply the method of characteristics, first the SSWE Eq. (3) are re-written in vector form as

$$\frac{\partial U}{\partial t} + A\frac{\partial U}{\partial \lambda} + B\frac{\partial U}{\partial \theta} + S = 0 \tag{5}$$

with

10

$$U = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}$$

$$A = \frac{1}{a\cos\theta}\begin{bmatrix} 0 & 1 & 0 \\ \Gamma^2 - u^2 & 2u & 0 \\ -uv & v & u \end{bmatrix}$$

15

$$B = \frac{1}{a}\begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ \Gamma^2 - v^2 & 0 & 2v \end{bmatrix}$$

$$S = \begin{bmatrix} \dfrac{-hv\tan\theta}{a} \\ -\left(f + \dfrac{u}{a}\tan\theta\right)hv - \dfrac{huv}{a}\tan\theta + \dfrac{gh}{a\cos\theta}\dfrac{\partial z}{\partial \lambda} \\ \left(f + \dfrac{u}{a}\tan\theta\right)hu - \dfrac{hv^2}{a}\tan\theta + \dfrac{gh}{a}\dfrac{\partial z}{\partial \theta} \end{bmatrix}$$

20

where $\Gamma \equiv \sqrt{gh}$. Using the directional splitting technique on Eq. (3) three equations are produced. An equation for each coordinate, longitude λ, and latitude θ, and the third for the source term S. The latter equation simply represents an ordinary partial differential equation for the source term while Eq. (6) and Eq. (12) for the coordinates are in advection form. These

25   last two equations are written in diagonal form in order to find the Riemann invariants and characteristics curves, a detailed

5

description of this procedure can be found in (Ogata and Takashi, 2004) or (Stoker, 1992). The equation for the longitude coordinate λ given by

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial \lambda} = 0 \tag{6}$$

has eigenvalues Λ given by

$$\Lambda_{\pm}{}^{\lambda} = \frac{1}{a\cos\theta}(u + \Gamma), \quad \Lambda_{3}{}^{\lambda} = \frac{1}{a\cos\theta}u, \tag{7}$$

which inserted in the diagonal form of Eq. (6) leads to

$$\frac{D^{\pm}}{Dt}\left(\Gamma \pm \frac{u}{2}\right) = 0 \tag{8}$$

where D/Dt represents the material derivative. Equation (8) means that the solution at a given grid point i is determined from two characteristic curves along C^{+} and C^{-} (Fig. 1). The result at a time $n+1$ can be found by adding and subtracting the expressions in Eq. (8) respectively to obtain

$$\Gamma_{i}{}^{n+1} = \frac{1}{2}\left\{\Gamma^{+} + \Gamma^{-} + \frac{1}{2}(u^{+} - u^{-})\right\}, \tag{9}$$

$$u_{i}{}^{n+1} = \frac{1}{2}\{u^{+} + u^{-} + 2(\Gamma^{+} + \Gamma^{-})\} \tag{10}$$

where $\Gamma^{\pm}$ and $u^{\pm}$ are the values at a time n, at positions which might not necessarily lie on a grid point. An interpolation in applied in order to determine these values and with them solve Eq. (9) and Eq. (10).

Following a similar procedure as (Yabe and Aoki, 1991), (Yabe et al., 2001), (Utsumi et al., 1997) we utilize a cubic-polynomial approximation on the grid profile to find the interpolated values. The polynomial is defined as

$$F(\lambda) = a\lambda^{3} + b\lambda^{2} + c\lambda + d \tag{11}$$

6

with

$$u\Delta t > 0 \begin{cases} a = \dfrac{f_{i+1} - 3f_i + 3f_{i-1} - f_{i-2}}{6\Delta\lambda^3} \\[2mm] b = \dfrac{f_{i+1} - 2f_i + f_{i-1}}{2\Delta\lambda^2} \\[2mm] c = \dfrac{2f_{i+1} + 3f_i - 6f_{i-1} + f_{i-2}}{6\Delta\lambda} \\[2mm] d = f_i \end{cases}$$

$$u\Delta t \leq 0 \begin{cases} a = \dfrac{f_{i+2} - 3f_{i+1} + 3f_i - f_{i-1}}{6\Delta\lambda^3} \\[2mm] b = \dfrac{f_{i+1} - 2f_i + f_{i-1}}{2\Delta\lambda^2} \\[2mm] c = \dfrac{-f_{i+2} + 6f_{i+1} - 3f_i - 2f_{i-1}}{6\Delta\lambda} \\[2mm] d = f_i \end{cases}$$

A similar analysis can be made for the latitude equation θ obtained from the splitting method, given by

$$\frac{\partial \mathbf{U}}{\partial t} + B \frac{\partial \mathbf{U}}{\partial \theta} = 0 \tag{12}$$

with analogous results for the eigenvalues and curves

$$\Lambda^\theta{}_\pm = \frac{1}{a}(v + \Gamma), \quad \Lambda^\theta{}_3 = \frac{1}{a}v, \tag{13}$$

$$\frac{D^\pm}{Dt}\left(\Gamma \pm \frac{v}{2}\right) = 0 \tag{14}$$

From which similar expressions as Eq. (9) and Eq. (10) can be found in order to estimate the values for h and hv.

The equations for the coordinates are solved using the fractional step method. Following this method, the source term given by

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{S} = 0 \tag{15}$$

5   is added to the solution obtained for Eq. (6) and Eq. (12). For the source term, central finite differences are used to solve the bathymetry term while the remaining values (cosine, tangent terms) can be solved analytically at each grid point since the variables are known straightforwardly.
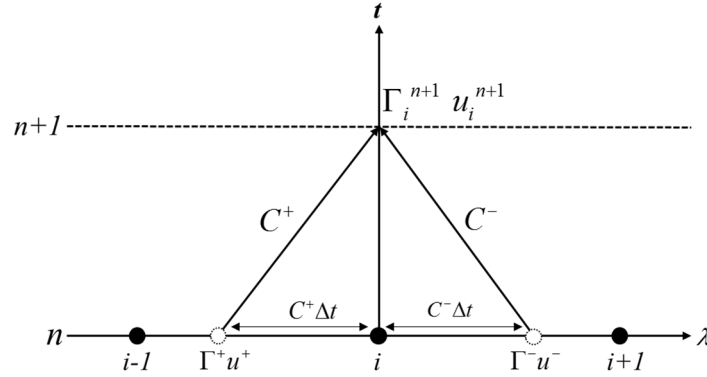
Fig. 1 Space-time diagram showing the characteristic curves $C^{\pm}$ where black dots represent the grid points, dotted
10   **points represent the values $\Gamma^{\pm}$ and $u^{\pm}$ to be interpolated to find Γ^{n+1} and u^{n+1}.**

In order to validate the implementation of the numerical methods for the SSWE, we used the benchmark described in (Kirby et al., 2013), where an initial Gaussian wave is propagated on an idealized sphere with water depth h=3000m. Results after 5000s show good agreement with the results reported which confirms the accurate propagation of the wave on the
15   sphere and the effects of the curvature and Coriolis force.

3.2 Run-up calculation

The Cartesian SWE are solved in specific areas of just a few kilometers where inundation has to be calculated. For this case we use a finite volume implementation (Bradford, 2002), (LeVeque and George, 2014) briefly described here. The Surface
20   Gradient method (SGM) (Zhou et al., 2001) is utilized to solve the SWE. This method uses the data at cell center to determine the fluxes. In general, depth gradient methods cannot accurately determine the water depth value at cell interface, since effects of the bed slope or small variations in the free surface cannot be determined accurately. These inaccuracies are

8

spread during the computation resulting in an incorrect simulation of the inundation. In order to overcome this, the SGM uses a constant water level H. Figure 2 depicts the stencil for the water depth reconstruction, by using the constant H as the total water depth at the cell interface ($i+0.5$) instead, the water depth be can determined accurately. In order to reconstruct the water depth the following expression is used

$$h_{L,R\ i+0.5} = \max(H_{L,R\ i+0.5} - \overline{z}_{i+0.5}, 0) \tag{16}$$

where $\overline{z}$ is given by

$$\overline{z}_{i+0.5} = (z_i + z_{i+1})/2 \tag{17}$$

A MUSCL scheme (Yamamoto and Daiguji, 1993) is used to find the flux value while Local-Lax-Friedrichs (LeVeque, 2002) is used to solve the bed slope source term. For the time integration a 3rd Order TVD Runge-Kutta scheme was used. s method is non-conservative however in tests the difference on mass conservation has shown to be almost negligible. Lastly, the bottom friction is computed using Manning's formula.
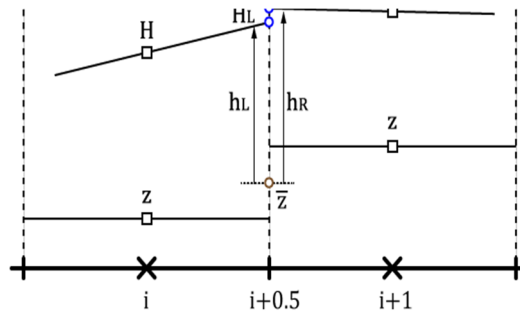
Fig. 2 Reconstructed water depth $h_{L,R}$ for inundation (LeVeque and George, 2014).

This run-up implementation assumes a thin film of water on land defined as ε. This parameter, set much smaller compared to the wave height, allows the computation of the wave inundation over land while keeping it stable. If the water height is less than ε (i.e. $h<\varepsilon$) then the height value is fixed as ε and the momentum is set as rest (i.e. $hu=hv=0$) on that grid point. This implementation has proven to be robust and stable under different benchmarks and simulations (cent et al., 2001). s numerical method implementation together a slope limiter produces a monotone scheme that preserves water positivity.

The one-dimensional dam break benchmark (Stoker, 1992) was used to compare the results with its analytical solution and good agreement was found. The shock wave was successfully captured for different initial water heights.
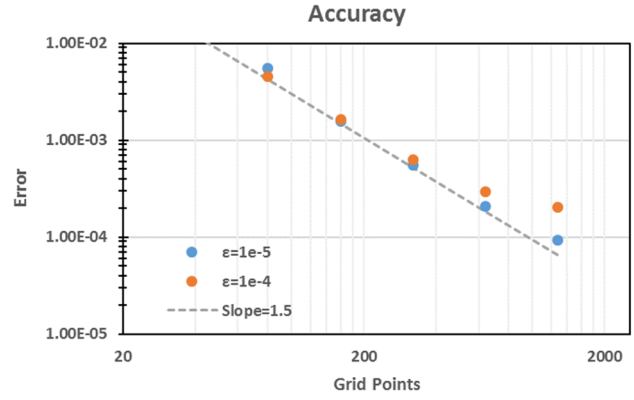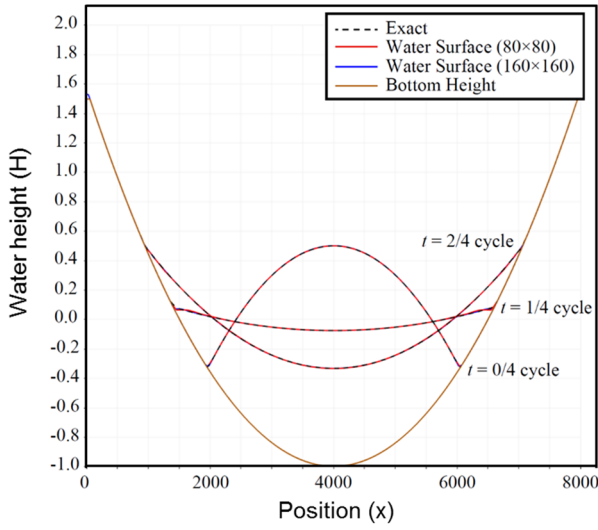
Fig. 3 Parabolic bowl problem cross section with $\epsilon = 10^{-4}$ on left panel. [water] depth error for parabolic bowl problem on right panel.

5   The Parabolic Bowl problem proposed by (Thacker, 1981) was also used to compare the accuracy of the inundation. The bottom bathymetry is given by

$$z(r) = -D_0(1 - \frac{r^2}{L^2}) \tag{18}$$

while the water height at a time t can be found from the analytical solution

10

$$H(r,t) = D_0\left(\frac{(1 - A^2)^{\frac{1}{2}}}{1 - A\cos\omega t} - 1 - \frac{r^2}{L^2}\frac{1 - A^2}{(1 - A\cos\omega t)}\right),$$

$$r = (x - L_x/2)^2 + \left(y - L_y/2\right)^2, \tag{19}$$

$$\omega = \sqrt{8gD_0/L^2},$$

$$A = \frac{(D_0 + \eta)^2 - D_0^2}{(D_0 + \eta)^2 + D_0^2}.$$

10

We use these parameters $L_x=L_y=8000$, $L=2500$, $D_0 = 1$ and $\eta=0.5$. Two grid sizes were used for testing, 80×80 and 160×160. Figure 3 shows the oscillating water in the bowl at different times. As it can be seen, the inundation method is able to capture well the analytical solution of the water height as it evolves in time on the different grid sizes. Measurements on this tests showed a third-order reduction of the error as the value ε was decreased.

3.3 Tsunami source model

TON-G focuses on propagation and inundation while relies on external parameters for the generation stage. In order to start a simulation, the initial condition is provided directly by RIMES using their prefered fault theory and model. In the *generation* process, a good initial source model is essential in order to obtain an accurate simulation. However due to the complex nature of the source dynamics during an earthquake and the difficulty to track it in real time (as it happens), currently it is beyond our grasp to obtain these parameters precisely and instantly. For these reasons we opted for a coseismic deformation. This deformation is calculated from the theory of displacement fields proposed by (Smylie and Mansinha, 1971). Their objective is to provide a closed analytical expression that "facilitates the interpretation of near-fault measurements". The expressions provided, valid at depth and surface, consist solely on algebraic and trigonometric functions that can be readily evaluated numerically based on a few source parameters like dip, strike, slip and length. These values are obtained from RIMES' databases online or loaded from a file. The original source generation code, provided by RIMES, was written for CPU and ported by us to GPU for this study.

3.4 Boundary conditions

Two kind of boundary conditions are used, *open and closed*. Open boundary sets conditions to allow waves from within the model to leave the domain through an edge without affecting the interior solution. *Closed* boundary which keep the fluid inbound in the domain, physically it means that no water flows across the edges. A wall boundary condition creates a physical total reflection when a wave hits a dry point.

In Eq. (3) the term $\cos\theta$ in their denominators produces a singularity at the poles of the spherical coordinate system. When working on a complete sphere, special techniques and treatment are required to compute values over the poles without divergence. In this study, the domain chosen represents a portion of the Earth centered in the Indian Ocean and does not extend near the poles in any circumstance which permits us to avoid this pole singularity.

The boundaries for the computational domain are set as open boundary condition at the South and East edges, and closed boundary condition at the North and West edges. All coastlines have wall boundary condition except for the special cases where particular regions set as *inundation* are defined. In those cases a complete run-up is computed using the methods

described in previous sections. Since the inundation method is relatively computationally intensive, using two kinds of boundaries in the coasts permits to focus computational resources just in areas of interest.

boundary between spherical and cartesian coordinates that occur in specific areas where inundation is computed has no special treatment since the area covering the inundation consists, by design, of just a few kilometers (Fig. 5 Top). This makes the difference between meshes almost negligible and does not affect noticeably the result.

4 Tree-based mesh refinement and bathymetry

An efficient use of resources, memory and computation, requires a mesh that covers areas of interest with high resolution only where desired, and leaves the rest of the domain coarser. The concept of this approach is similar to that of the adaptive mesh refinement, initially introduced by (Berger and Oliger, 1984), (Colella, 1989) in the 1980s as a method to solve PDEs on an automatically changing hierarchal grid, solving for a set accuracy on certain areas of the interest instead of unnecessarily overly refining on the entire domain.

To generate the mesh for the domain, we use a customized tree-based mesh refinement, without the need of re-meshing during simulation since the geometrical features remain unchanged. We briefly explain the process of tree-based refinement (Yerry and Shephard, 1991). Figure 4 illustrates this procedure using a moon-shaped green point as the area of interest. At each level, the domain and its tree structure, called *quadtree*, is presented. Initially just a quadrant and its *quadtree* root exist. Each quadrant represents a block of domain points. At level 2, one refinement has occurred and the original quadrant (father) is replaced by four new ones (children). By containing the same number of points as their parent quadrant, these *children* allow for greater resolution. Each child is represented as a *leaf* of the tree's root. Level 3 shows the refinement of two of the Level 2 quadrants and are represented as two new leaves deeper on the *quadtree*. Focusing around the point of interest, Levels 4 and 5 show the subsequent refinement of two quadrants of their respective previous levels. As it can be seen, each refined quadrant is replaced by four new ones and these extend deeper on the tree. This process can continue recursively until reaching a desired goal, usually based on resolution or minimum error. Using this block refinement allows for greater resolution only around the points of interest while the *quadtree* data structure associate with it keeps track of the blocks connectivity.

12
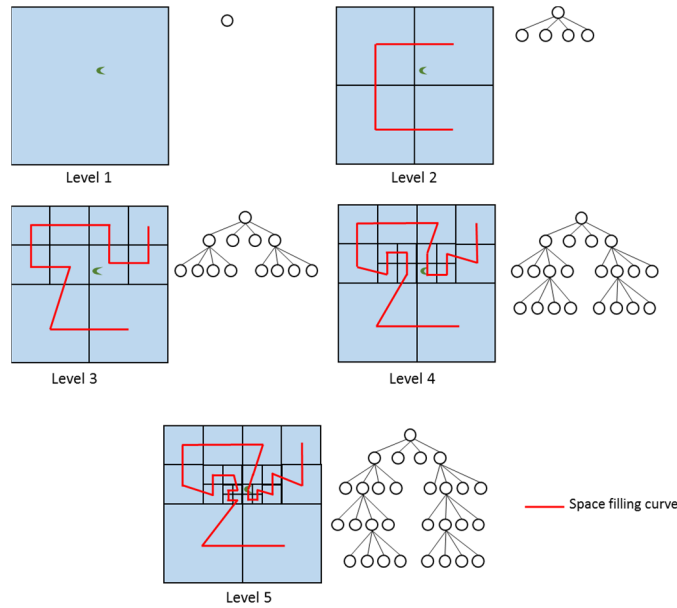
Fig. 4 Tree-Based block refinement with quadtree structure and Hilbert space filling curve for 5 levels

The difference on spatial resolution between two adjacent levels is called the refinement ratio. For nested grids, this ratio is any positive integer. However using large integers tend to introduce inaccuracies in the computation. The existence of an

5 abrupt change from one level to the next requires special boundary treatment, especially when complex bathymetry or topography is involved. For tree-based refinement this ratio is fixed as $\Delta x_l / \Delta x_{l+1} = 2$, where l represents the block level and Δx the grid resolution. This constant and small ratio creates a smooth wave transition between levels.

4.1 Customized mesh generation

10

The domain used for this work represents a large portion of the Indian Ocean (Fig. 5), which consists initially of a uniform mesh of 56×30 blocks, each made up of 65×65 node-centered cells. Using the tree-based refinement, specialized customizations are developed to adapt it to our specific needs. In general, mesh refinement methods utilize an error estimation as the rule to determine if a block should be refined, however in this implementation the refinement depends on a

15 *target* grid resolution combined with two factors, the block's distance from the coastline and the presence of a focal area. The refinement rule's first factor depends on the distance of the block to the shoreline, the objective is to recursively refine ks close to the coast until reaching a target high-resolution threshold, while blocks far in the ocean remain with a coarser resolution. This process involves two steps, determining the block's distance from the coast and checking if its distance is within refinement.

13

To accurately estimate the geo-distance between two points can be a complex task since the surface of the Earth is not a perfect sphere. However, for our refining purposes, a rough estimate is enough to determine the distances between the shoreline and the blocks. This is achieved by creating a signed distance function based on the Level-set method. A detailed explanation of this procedure can be found in (Fedkiw and Osher, 2003). The distance function's *zero* level is represented by the cells along the shoreline ($z=0$). Positive distances represent cells on land while negative distances represent cells on the ocean. Using this distance values, each block is tested for refinement. Blocks with one cell or more within a certain distance from the coast, called *refinement stripe*, are flagged for refinement until they reach the fine-target resolution. The width of the refinement stripe is problem dependent and is input by the user based on their needs.
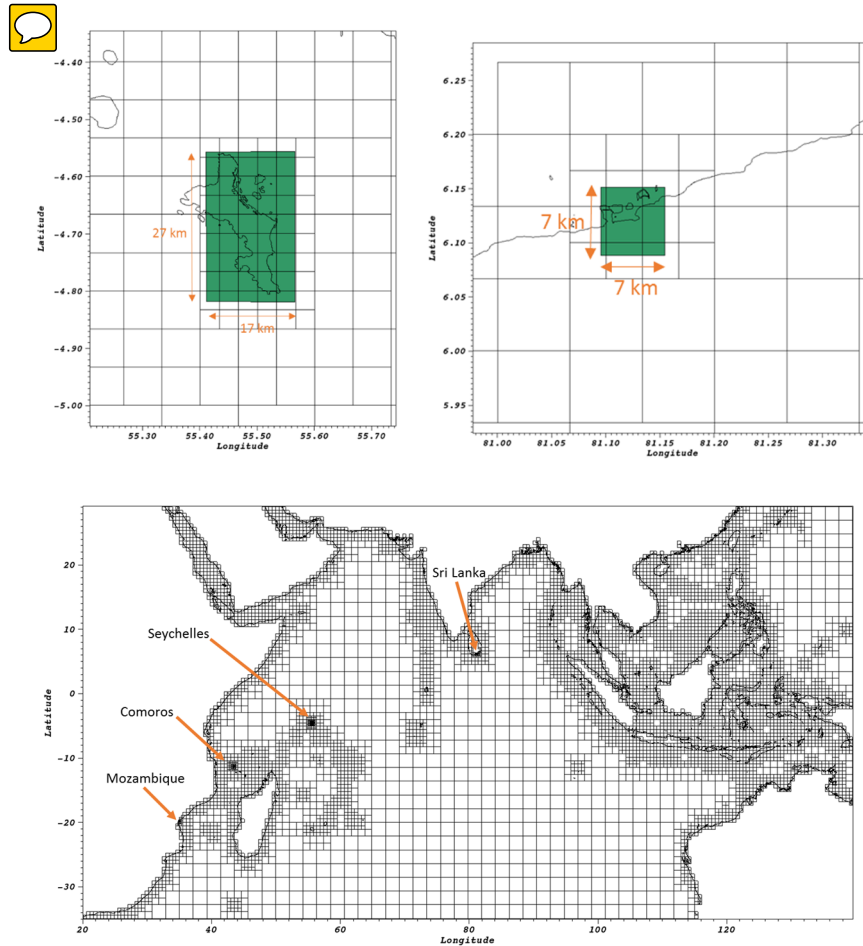
Fig. 5 Bottom: Mesh Refinement for Indian Ocean Domain with 4 Focal Areas; Mozambique, Comoros, Seychelles and Sri Lanka. Top: Zoom on Sri Lanka and Seychelles regions, FA highlighted in green.

14

For this study the initial resolution at ground Level 1 is 2 arc-min (an arc-minute being 1/60 of a degree, at Earth's equator equivalent to 1852 m) and the target finest resolution is 0.03125 arc-min (approximately 50 m), generating a total of 7 levels. This block refinement process can accurately trace complex coastlines and focus high resolution only in the shores. A downside is the considerably large number of total blocks generated, over 230,000 in initial tests, which represents over
5   100 GB of memory storage.

In order to reduce the memory footprint, we use the fact that only certain regions need high resolution, which inspired us to use a second refinement factor named *focal areas* (FA). This second factor is an additional constraint which consists in locating on the domain a convex polygonal area which serves as a refinement delimiter. It is possible to locate more than one at a time and since this is an additional constraint to the first refinement step, only blocks flagged for refinement at the first
10   step need to be tested again. On this second test, a block is tested if it is inside or outside a focal area. If a block is completely outside the focal area, then it is un-flagged for refinement. Only blocks partially or totally inside the focal area are refined. The process of determining if a block lies inside or outside a focal area is based on collision detection theory using the Separating Axis Theorem (SAT). This is a well-known theorem applied to physical simulations (Szauer, 2017) and consists of a relatively *light* algorithm for 2D, which allows to test large number of blocks rapidly. A description of the SAT
15   can be consulted in (Moller et al., 1999) or (Gottschalk et al., 1996). Since the focal area is an additional constraint, it can be toggled active after any chosen level. A specific number of levels can be refined without this constraint while the following are affected and delimited. Additionally, all dry blocks at Level 7 (highest resolution) that are inside a FA are considered inundation areas. This implies that run-up is computed on the coastlines instead of using a reflective boundary.

The last step in the mesh generation consists in the removal of *land dry-blocks*. Considering that tsunami inundations,
20   with few exceptions, generally extend tens to hundreds of meters inland, it becomes clear that blocks located deep inland are unnecessary for the computation. For this reason all blocks whose cells' distances are larger than a *land-distance* threshold are considered *land dry-block* and deleted from the domain.

The complete result of the customized refinement in the Indian Ocean domain is shown in Fig. 5. Four focal areas are used located in Mozambique, Comoros, Seychelles and Sri Lanka. The focal area constraint start after Level 3. This value is
25   chosen to coincide with GEBCO's 30 arc-second bathymetry, using the highest available accuracy for the coasts without needing to interpolate. The final result shows the refinement at higher levels limited to within the focal areas. All dry blocks exceeding the land-distance threshold of 10km were removed from the mesh. This reduced the number of blocks generated drastically to 7849, while the memory needed to store them became less than 15GB. This customized refinement procedure proved to be fast and efficient, taking just around a minute to produce the results. The meshes generated by TRITON-G can
30   be either computed real-time or loaded from a repository at the beginning of the simulation.

15

4.2 Halo exchange

Blocks must exchange results with their neighbors after each time step for the next iteration. For this purpose they share a boundary layer in their adjoining sides. This layer or *halo* extends over the neighbor's grid and updating represents one of three kinds of operation: copying, coarsening or interpolating.

If two neighbor blocks have the same level, then the halo is readily updated by exchanging values directly without any further computation, this represents a copying swap. If the neighbors are at different levels (*l* and *l+1*) then additional computation is required before the halo exchange. If the block's neighbor is one level up then values for the halo are averaged down from the block with higher accuracy before swapping, this has the effect of passing down better accuracy to blocks with lower resolution like in a cascade effect. The last case, interpolating, occurs when the block's neighbor is one level down. For this, the values for the halo are interpolated from the neighbor block using a third-order polynomial interpolation, similarly as in Eq. (11). The portion of the boundary stencil used for interpolation is shown in Fig. 6.

(a) West, South cases

(b) East, North cases
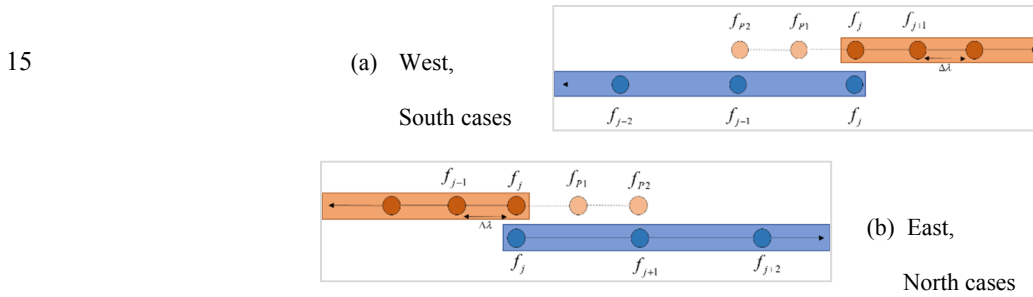
Fig. 6 Halo interpolation stencil for the four edges (a) west, south and (b) east, north.

The new values for the halo for the north (*N*) and east (*E*) edges can be found from

$$f^{N,E}_{P1} = \frac{1}{4}(f_j + 4f_{j+1} - f_{j+2}),$$

$$f^{N,E}_{P2} = \frac{1}{4}(-f_j + 6f_{j+1} - f_{j+2})$$

(20)

since they are analogous orientations. For the south (*S*) and west (*W*) edges similar expressions are used

$$f^{S,W}_{P1} = \frac{1}{4}(-f_{j-2} + 4f_{j+1} + f_j),$$

$$f^{S,W}_{P2} = \frac{1}{4}(-f_j + 6f_{j+1} - f_{j+2})$$

(21)

16

In order to avoid spurious waves that might be generated from interpolating the water height value h, constant water level H is used instead, and the original variable is recovered by using the relation $h = H - z$.

4.3 Topography and bathymetry

The data used in this study for bathymetry and topography comes from different sources. Initially, The General Bathymetric Chart of the Oceans (Oceans (GEBCO), 2017) database is used on the entire domain. GEBCO is freely available in 30 arc-second spatial resolution. When coarser resolution is needed, values are averaged from this database. On the contrary, if finer resolution is needed, a third order interpolation is implemented to generate the new values. Where available, databases with more precise measurements are used to replace the original GEBCO database. For the focal areas in Mozambique, Comoros, Seychelles and Sri Lanka, RIMES' proprietary databases generated from field measurements were provided to us to estimate the inundation more accurately.

5 GPU computing

The introduction of C-language extension CUDA (NVIDIA, 2017a) by NVIDIA® represented a disruption in the traditional way simulations were done. The availability to program GPU cards general purpose permitted researchers to perform calculations no longer exclusively on CPU. Due to the intrinsic parallelism of graphics, GPUs evolved to deliver in a card hundreds and thousands of processors more than CPUs. The main reason behind the exceptional performance of GPUs lies in the specialized design for compute-intensive, highly parallel computation, with transistors dedicated exclusively to processing as opposed to flow control and data caching. The latest NVIDIA Tesla cards P100, with Pascal architecture have a peak performance of 9.3 Teraflops on single precision and 4.7 Teraflops on double precision (NVIDIA, 2017b). We take advantage of this technology to develop a full-GPU implementation to deliver fast forecasting results.

5.1 SSWE GPU kernels

CUDA provides kernels as the way to define functions that are executed in parallel on GPU. A kernel launch is organized in a grid of blocks of CUDA threads. The clear analogy between CUDA blocks and mesh blocks provided a guide to organize the grid for GPU execution. The SSWE are computed exclusively on GPU by processing the mesh blocks created during the domain refinement step and are stored in a structure of arrays on GPU global memory. Each mesh block have a

size of (65+4) × (65+4), where the 4 corresponds to the total size of the halo. CUDA threads can be organized in any three-dimensional block configuration as needed by the problem. Since GPUs process threads in warps of 32, using multiples of this number is desirable to avoid performance penalties.
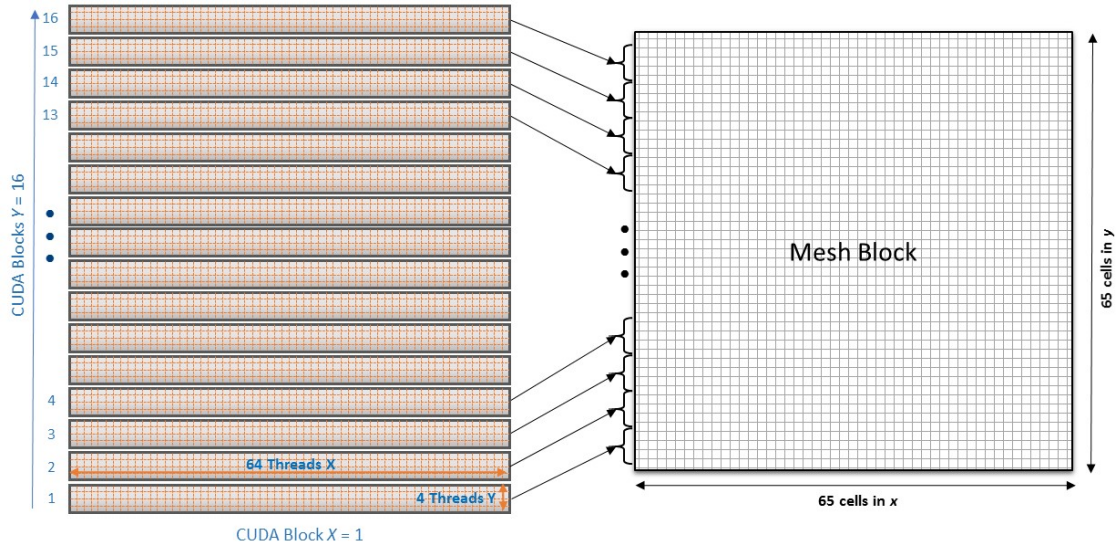
5

Fig. 7 Mesh block computation using CUDA kernels. Each CUDA block is made of 64×4 threads and computes a portion of the mesh block. One CUDA thread computes one mesh block cell

The kernel grid configuration for the SSWE is described briefly and shown in Fig. 7. CUDA threads are organized in two
10 dimensional blocks of size 64×4. The 64 threads in the x dimension cover the length of a mesh block requiring only one CUDA block. For the grid's y dimension, 16 CUDA blocks are set with 4 threads each, for a total of 16×4=64 threads, covering the height of the mesh block. With this configuration, one CUDA block computes a portion equal in size of the mesh block and the 16 CUDA blocks cover the entire mesh block. Additionally, one CUDA thread computes one mesh block cell. The specific calculation of each thread varies depending on the block type (*Wet*, *Dry*); however, the configuration
15 remains the same. In both cases, threads compute the governing equations described in section 3.1. The main difference occurs in the case of a *Dry* block; in this case, cells that represent land or coastline compute a reflective wall boundary.

To process all the mesh blocks, this two-dimensional CUDA block configuration is extended along the z-direction as many times as mesh blocks exist. The computation of the 65th cells is done separately with a specialized kernel based on the
20 SSWE kernel.

18

In the case of Cartesian SWE kernel, the grid chosen for this kernel is different than that of the kernel for SSWE. In this case, a mesh block is sub-divided and covered by CUDA blocks of 16×16 threads. The excess of threads at the edges is not computed using a conditional limiting the grid size.

The source fault code was ported to GPU from the original C version. Due to the exclusively arithmetic operations and lack of a stencil memory access involved, a 20× speed up was achieved, reducing the computation of the initial condition to just a few seconds.

Several kernel optimizations were applied in order to accelerate the model's time-to-solution. This includes using the latest CUDA version to take advantage of the latest compiler updates. To avoid branch divergence as much as possible parts of the numerical method were re-written to eliminate conditionals. Precomputing terms that do not change in time like trigonometric terms depending on the longitude θ, storing them on arrays and reusing them during the simulation. Using built-it functions to compute complicated exponentials like those in the Manning formula. Although the optimizations provided speed up, no sacrifice was incurred on precision. All GPU computations are performed on double precision.

5.1.1 Halo update on GPU

Update of the halo region of each mesh block after each time step with the latest values from neighbor blocks represents three different kinds of exchanges: copying, coarsening or interpolating. These operations are performed entirely on GPU. Kernels designed for each kind of exchange were created. In order to efficiently process the block edges, three lists are generated containing the list of halos that require each operation. This way the kernels be launched concurrently and each focus on a different task minimizing the need for conditional divergences.

5.1.2 Specialized kernel types

By analyzing the domain's bathymetry it is easy to notice that some mesh blocks contain only wet points while others are a combination of dry and wet points. This idea is used to replicate the SSWE kernel in two variations.

The first SSWE kernel, named *Wet*, is used to compute the wave free propagation on wet-only blocks. The second SSWE kernel, named *Dry*, is used to compute the wave propagation with coastline boundaries in wet-dry mixed blocks. The main difference in the code between them being the additional treatment for the wall boundaries at coastlines in the case of the *Dry* kernel. A third kind of kernel called *Inundation*, specializes in computing the run-up on dry blocks inside focal areas.

19
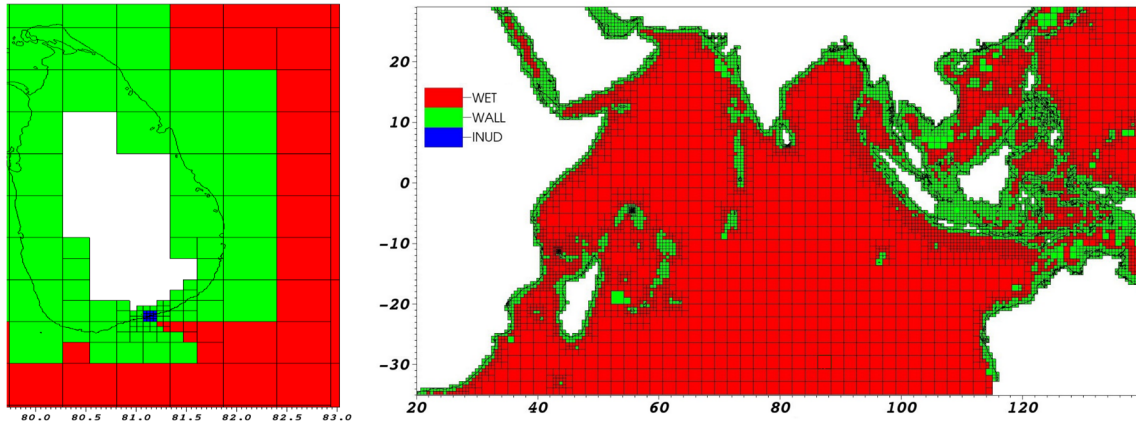
Fig. 8 Mesh blocks colored by kernel type, Wet, Wall and Inundation. Left: zoom over Sri Lanka FA to highlight the inundation kernels shaded in blue. Right: Kernel type distribution on the entire Indian Domain.

5   The result of the kernel assignment is illustrated in Fig. 8 where blocks flagged as *Wet* are shaded in red, *Dry* blocks are shaded in green and *Inundation* blocks in blue. As expected *Dry* blocks tend to extend where coastlines lie while *Wet* blocks are spread out in the open ocean. When inside a focal area, dry-type blocks at level 7 are re-flagged as Inundation type. An example of this can be seen in the left image of Fig. 8 for the Sri Lanka FA with inundation blocks in blue. Whereas a single kernel would be too complicated and inefficient to compute the entire domain, splitting down the computation in specialized

10   kernels for each type of block not only provides a simpler way to process the blocks through lists but also gives the ability to fine tune them independently for higher performance.

5.2 Space filling curve and multi-GPU

15   In order to implement multi-GPU for further acceleration, first an appropriate domain partition must be chosen to guarantee an even work load among cards. Since a uniform mesh is not being used, this partition is non-trivial. Although block connectivity is kept using a *quadtree* structure, this does not provide information about the blocks ordering. For this purpose we use the space filling curve (SFC) (Sagan, 1994) as a way to trace the blocks ordering on the domain.

      SFC is a curve that fills up multi-dimensional spaces and map them into one dimension. It has many properties desirable

20   for domain partition, it is self-similar and it visits all blocks exactly once. We use the Hilbert curve in this work since it tends to preserve locality, keeping neighbors together and does not produce large jumps in the linearization like other curves tend to, such as the Morton curve. Figure 4 shows the Hilbert curve generation as a red line overlying the quadrants. It starts as a bracket on the first four quadrants, and with each spatial refinement, the bracket gets replicated subject to rotations and

reflections to guarantee the characteristic of the curve. The result of generating a Hilbert SFC for the Indian Ocean domain is shown in Fig. 9. By using this curve as a reference it is possible to establish the block ordering to partition the domain on even portions. The result of splitting the domain for 8 GPUS is shown in Fig. 10, where each portion is represented by a different color. In this case, 7 GPUs have a total of 981 blocks each, and the 8th a total of 982, making it a well balanced partition. Different tests using 1, 2, 3 and 4 GPUs also achieved balanced partitions.
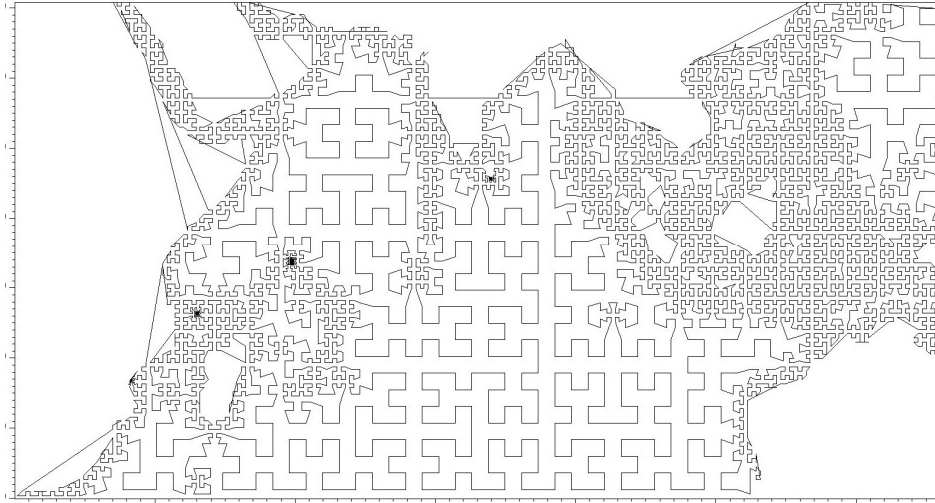
Fig. 9 Hilbert Space Filling Curve for Indian Ocean Domain with four FAs

10

Introducing multi-GPU also introduces the need of a buffer communication between cards. In the current CUDA GPU memory model, global memory cannot be accessed between different cards. This exchange is achieved by preparing buffers on GPU memory, downloading to CPU memory, using MPI to exchange the messages and uploading the received buffer to GPU memory.

15 In order to handle the communication structures and to produce buffers that do not represent a large communication overhead, we construct buffers following the *user datagram protocol* (UDP) (Reed, 1980) design, a concept traditionally used in network and cellular data communication. In this way, it is possible to eliminate the need for communication look-up tables while at the same time it makes the buffer exchange smooth and simplified. As depicted in Fig. 11, the first step consists on collecting all the halos to be transferred in a single buffer on GPU memory. This buffer is designed like in UDP,

20 with a header in front of every chunk of data. This header contains three bits of simple information: the destination block, the destination edge and the total size of its data. By including a simple 3-data header before the sent values, it is possible to organize the buffer in any way that packing/unpacking occurs smoothly and seamlessly.

21
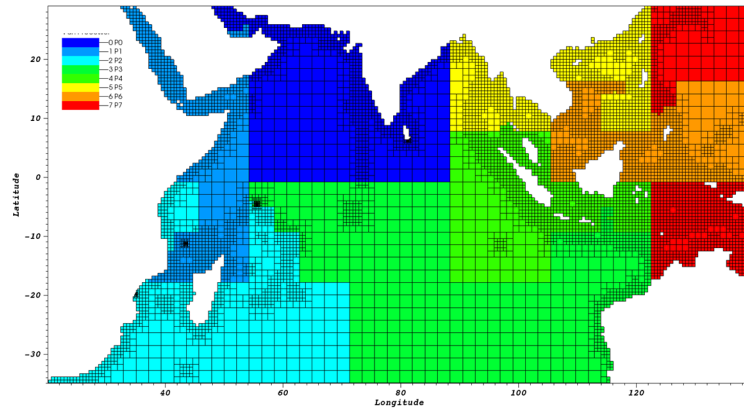
Fig. 10 Indian Ocean Domain partition for load balance for 8GPUs. Each color represents a different GPU.

By using this method no extra memory is needed to store communication tables or exchange them between processors. A single-buffer transfer between processes drastically reduces the communication time as opposed to transferring each halo individually.
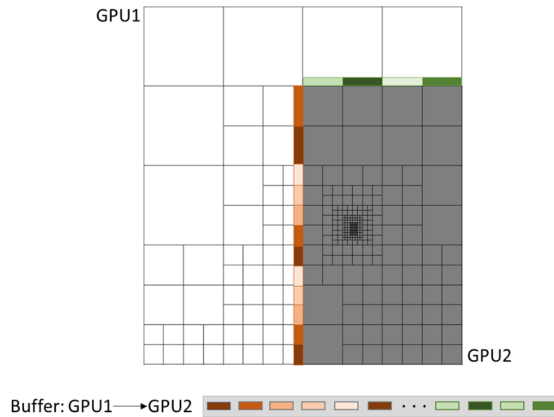
Fig. 11 Multi-GPU communication. GPU buffer data collected and packed for a single communication.

5.3 Variables and rendering output

The full work-flow of TRITON-G is depicted in Fig. 12, where the GPU flow is composed of two parts, the main simulation, which includes computing the fault source, wave propagation and inundation and the output compute and storage.

22
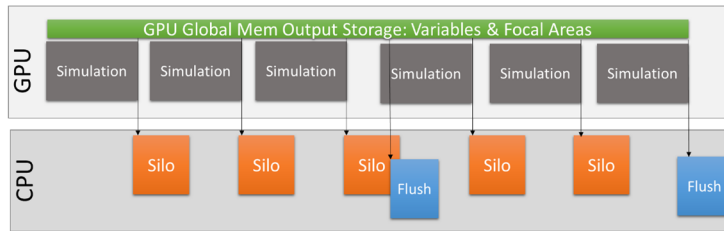
Fig. 12 TRITON-G computational flow.

5   For post-processing analysis purposes, output for the wave maximum height, maximum inundation, arrival time, flux and gauges is created. These are computed during simulation and stored on GPU memory, then flushed to CPU when required by the user. A full-domain rendering at a regular frequency is also produced during simulation, while for the FAs, wave values at a much higher frequency are stored. These values are used for rendering at post-processing to avoid unnecessary output overhead.

10   TRITON-G generates SILO format files (Lawrence Livermore National Laboratory, 2017) filled with values from all blocks to generate the rendering images. Even though the image generation for the entire domain is not very frequent, the process of generating a SILO file for such a large mesh represented a considerable overhead of around 15 to 20% of the total runtime. In order to minimize this unwanted effect we took advantage of the Piping mechanism. *Pipe* is a system call that creates a communication between two processes that run independently. In this way, a parent program can launch a child

15   program and both run completely different tasks at the same time without interrupting each other. Using this concept, first a utility to create the SILO files for the entire domain was created as a stand-alone application. During execution, TRITON-G calls this sub-program when a SILO file has to be written, running both simultaneously. Data between them is shared through the CPU shared memory. Figure 12 shows the advantage of implementing Pipe asynchronous output. Unlike traditional asynchronous output that relays on a large computational time to hide output, this Pipe method provides the

20   ability to hide the output processing behind several computational time-steps. The result is an almost total elimination of the output overhead. Measurements showed that the output process after optimization represented just 1 to 2% of the total time, practically removing the overhead.

       The size of the output produced during simulation depends on user input parameters. For a 10-hour simulation with an output frequency of 4 minutes for the entire domain and 5 seconds for four FAs the required memory storage is around

25   65GB.
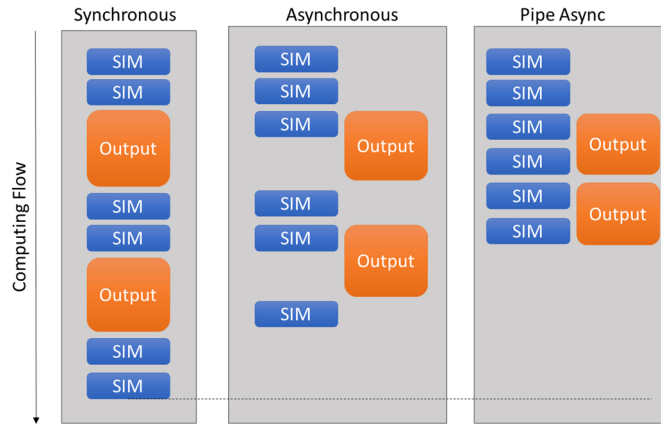
Fig. 13 Output overlap and optimization using Pipes

5.3.1 Sub-cycling implementation

A sub-cycling technique 💬 introduced in order to increase the computational time step and speed the computation further up. Sub-cycling consists in setting a larger than the minimum time-step as a global time-step Δt, and making blocks with smaller local time-step cycle in sub-steps (*ns*) to match the global Δt. The time-step Δt is calculated in each level using the Courant-Friedrichs-Lewy condition (CFL) (Courant et al., 1967). Initially the CFL number is set to 0.8 for this work.
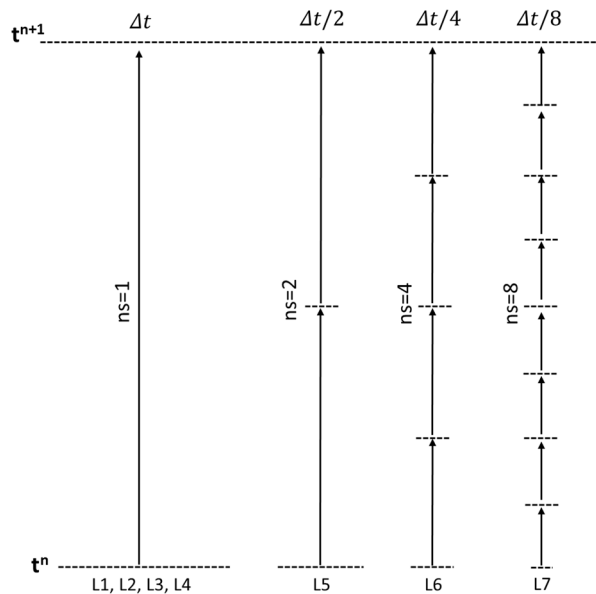
Fig. 14 Illustration of the sub-cycling process for each level.

A graphical illustration of the sub-cycling implementation is shown in Fig. 14. Blocks with the same number of sub-cycles (Levels L1, L2, L3 and L4) are grouped in a single list. A block at level 5 (L5) has a time step of $\Delta t /2$, which implies that it requires two cycles to match the global Δt.

While in theory the larger time step brings up speed, a potential downside is that too many blocks sub-cycling can create a large work-load overhead resulting in a slow-down of the whole computation. To avoid this, a global Δt of 1.6s is chosen to sub-cycle only blocks with levels over level 4. The reason being, that around 80% of the total number of mesh blocks are level 3 and sub-cycling them would represent too large an overhead and would defeat the purpose of applying this technique. Table 1 gathers the CFL numbers per level after implementing the sub-cycling. The second column shows the maximum Δt allowed in each level using the initial CFL = 0.8. The third column shows the resulting number of sub cycles per level (ns) and the fourth column shows the new CFL values obtained for each level. In all cases the new CFL values remain below 1 to guarantee stability.

In general after a large Δt step, corresponding boundary conditions are interpolated in time to update the sub steps. However this procedure introduces an additional computational overhead. To pursue the fastest modelling possible, TRITON-G rescinds the boundary generation and instead uses the available boundary values at time *n*. Based on the benchmark and hindcast comparison, this decision proved to be acceptable based on the good agreement and accuracy of the results.

Level	Max Δt (CFL=0.8)	ns (Δt =1.6)	S.C. CFL
L1	10.71	1	0.12
L2	5.13	1	0.25
L3	2.37	1	0.54
L4	1.65	1	0.78
L5	0.95	2	0.68
L6	0.55	4	0.59
L7	0.26	8	0.39

Table 1 CFL values used after introducing sub-cycling (S.C. CFL) for each of the seven levels. The second column shows the maximum Δt per level using CFL = 0.8 and the third column shows the number of sub-cycles (ns) required in each level when using Δt =1.6

Introducing this sub-cycling technique varies the GPU load initially created since a single block might be computed more than once. In order to guarantee load balance, two weights are applied to the space filling curve. The first weight takes into account the different type of block and the second the number of sub-cycles. Each block gets attributed a *weight* during the SFC generation equal to the number of sub-cycles it requires. This approach for the domain partition allows to create a

fair work re-balance on the GPUs The effect of implementing the weighted load balance can be seen in Fig. 15 where GPU execution times per time-step are presented, with and without load balance. Implementation of the sub-cycling technique showed a speed up of around 15% in the total wall clock runtime.
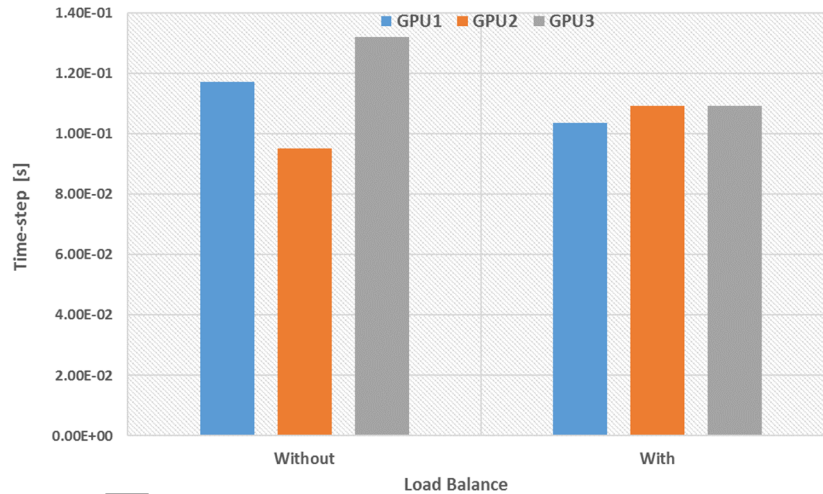
Fig. 15 GPU execution time with and without load balance

5.3.2 Runtime performance

Several tests to estimate the performance of TRITON-G were done. Results ran on the Supercomputer *Tsubame* 3.0 (Tsubame, 2017) are presented, with Intel Xeon E5-2680 2.4GHz × 2, RAM 256GB, NVIDIA Tesla P100 (16GB) × 4/node, CUDA 8.0, gcc 4.8.5, Openmpi 2.1.1 and Omni-Path HFI 100 Gpbs network.

As comparison, results on a second machine are also presented, using four Tesla K80 (12GB×2) cards in a node (eight GPUs in total), GPUs are connected through PCI-Express 3.0, Intel Xeon CPU E5-2640 @2.6 GHz, RAM 128GB, CUDA 8.0, gcc 4.7.7 and Openmpi 1.8.6. These performance tests serve to show very good portability of our program on different hardware, older and much newer, without requiring changes or producing problems.

The breakdown of the main parts of the simulation using 3 GPUs is shown in Fig. 16, where *Inund* stands for Inundation kernel, *Wall* stands for the wall kernel, *Wet* for the *Wet* kernel and *X* and *Y* for the direction of the computation equivalent to longitude and latitude respectively. The process of updating the halos, presented in the graph as *Bnd* represent only 9% of the total running time. It can be seen that the Wet and Wall kernel have similar performance despite the fact that the wall includes additional treatment for the coast boundaries. Since this treatment consists of many conditionals and they were replaced during optimization, it is understandable that the performance is similar. The slide *Others* include several values,

most importantly communications which represents around 1.5-2.0% of the total running time. ...ormance of the main kernels on one GPU in floating point operations per second (FLOP/s) is gathered in Table 2.

Kernel	GFLOP/s
WallX	549.57
WallY	549.56
WetX	706.98
WetY	712.51
Inund	87.12

Table 2. Kernel performance for one GPU in Giga FLOP per second

5

Results for runtimes using Tesla P100 cards and Tesla K80 cards are presented in Fig. 17 for 1, 2, 3, 4 and 8 GPUs. For this test, 10 hours were simulated on the mesh initially generated for the Indian Ocean Domain (Fig. 5). All runtimes measurements include output time.

Fig. 16 Computing breakdown shown in percentage

10

A comparison between both GPU cards shows a speed up of almost 4 times from the older K80 cards to the latest P100 on Tsubame 3.0. In our collaboration project with RIMES an objective to complete this test under 15 minutes was set, which could be fulfilled by using 3 to 8 GPUs in this configuration. Run time for 3 GPU with K80 cards was 39.96 min and 12.1 min with P100 cards.

15

27
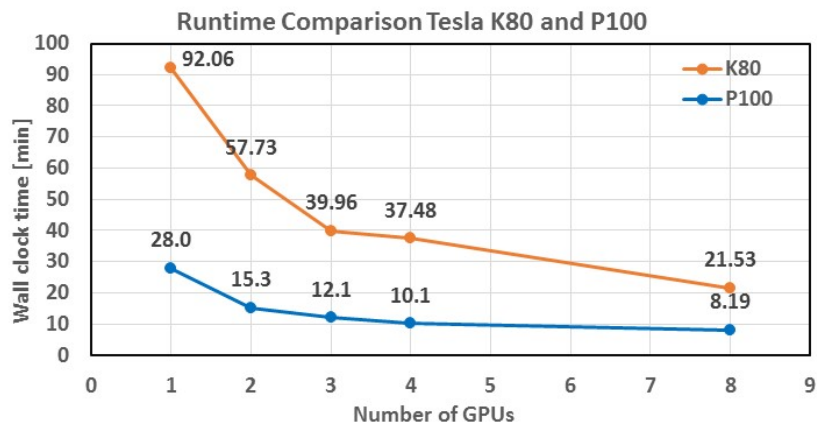
Fig. 17 Wall clock comparison of 10-hour simulation on Tesla K80 and Tesla P100

A saturation is noticeable in Fig. 17 as the number of GPUs are increased. A possible reason for this phenomenon is related to the increase of buffer preparation, packing/unpacking, and the communication exchange. Using the same domain size for all cases is another possible reason. Having fewer blocks on each GPU generates lower occupancy which might degrade performance. However, having met this study's time-to-solution objective of less than 15 minutes, no further optimization was deemed necessary.

By measuring the time required for the first wave to arrive in the focal areas, it was found that for Sri Lanka. Using 4 GPUs just 2 minutes wall clock time is required to generate the results of the inundation. The real tsunami wave took approximately 2 hours to propagate from the initial source to Sri Lanka, obtaining simulation results faster than real time gives authorities sufficient time to make decisions regarding evacuations.

6 Tsunami inundation benchmark comparison

In order to compare the numerical results of TRITON-G with existing benchmarks and test its ability to estimate inundation, we present the results obtained using the main benchmark tests proposed in the National Tsunami Hazard Mitigation workshop (NTHMP, 2012). Results from other models participating in the workshop can be consulted in that reference. In this section, the comparison of the benchmark "1993 Hokkaido-Nansei-Oki (Okushiri). Field" is shown. Further comparison results with benchmark problems 4, 6 and 7 (abbreviated as BP4, BP6, BP7) can be found in the appendix section.

A detailed description of the benchmarks can be found in (NTHMP, 2012) and the data needed for them can be found in the repository https://gitub.com/rjleveque/nthmp-benchmark-problems . For completeness we give a brief explanation of the benchmark and the tasks it involves.

6.1 Benchmark Problem #9: Okushiri Island Tsunami - Field

This benchmark problem (BP9) is based on the data collected from the Mw 7.8 Hokkaido-Nansei-Oki tsunami around Okushiri Island in Japan in 1993. The goal is to compare computed model results with the field measurements.
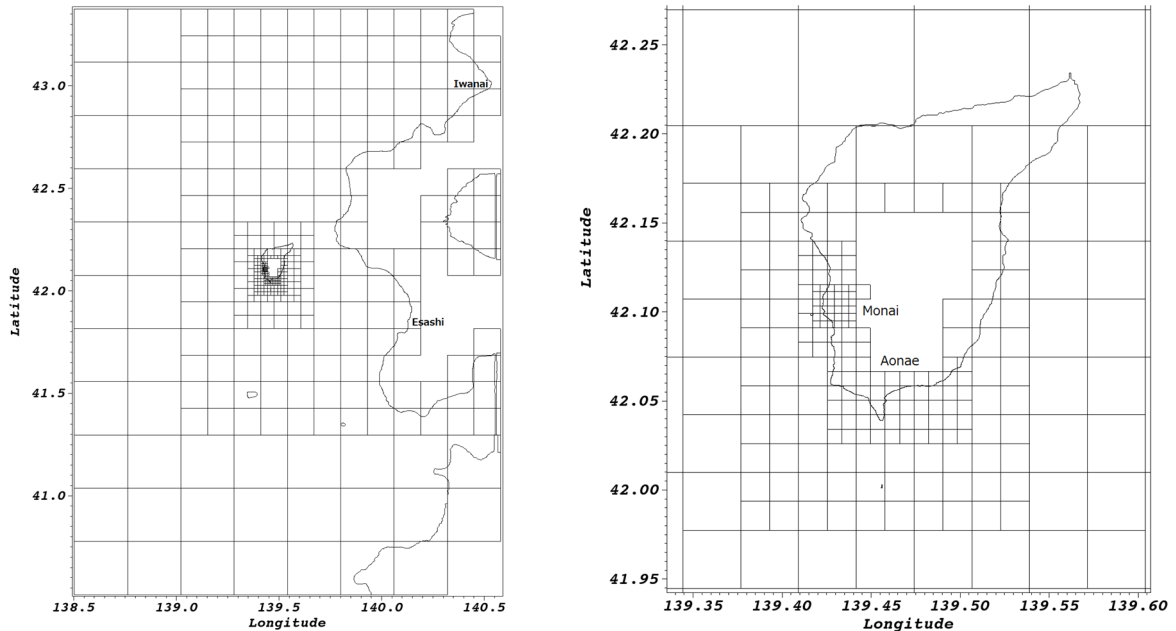

Fig. 18 Left: entire domain refined mesh containing 7 levels. Right: zoom on Okushiri island. Higher resolution used around Monai Valley at level 7 (7 m approx.) and Aonae region at level 6 (14 m approx.)

6.1.1 Problem setup

The following parameters were used for the computation:

- *Bathymetry*: taken from databases provided by (NTHMP, 2012), interpolated where necessary.
- *CFL*: 0.9
- *Simulated time*: 60 minutes
- *Initial condition*: source generated from the database provided by DCRC (Disaster Control Research Center) Japan solution DCRC17a, described in (Takahashi, 1996).
- *Boundary conditions*: open boundaries at the four domain edges.
- *Friction*: Manning coefficient set to 0.02

- *Computational domain*: a mesh refinement is used on the entire domain (shown in Fig. 18). Seven levels are used in total. The resolution of base level 1 is 450 m and the resolution of level 7 is approximately 7 m. Dry blocks that did not take part in the computation were removed in the mesh generation process.

6.1.2 Tasks to be performed

This benchmark requires the following tasks to be performed:

8. Compute run-up around Aonae
9. Compute arrival of the first wave to Aonae
10. Show two waves at Aonae approximately 10 minutes apart; the first wave came from the wet, the second wave came from the east
11. Compute water level at Iwanai and Esashi tide gauges
12. Maximum modeled run-up distribution around Okushiri island
13. Modeled run-up height at Hamatsumae
14. Modeled run-up height at a valley north of Monai.

6.1.3 Numerical results

In this section we present the numerical results obtained with TRITON-G for benchmark problem #9.

6.1.3.1 Run-up around Aonae

The maximum inundation around Aonae peninsula modeled during the simulation is shown in Fig. 19. Contours every 4 meters are drawn to show the outline of the topography. Maximum inundation height computed was nearly 15 meters but the scale used is set to the upper limit of 10 m to highlight the areas where major inundation occurred.
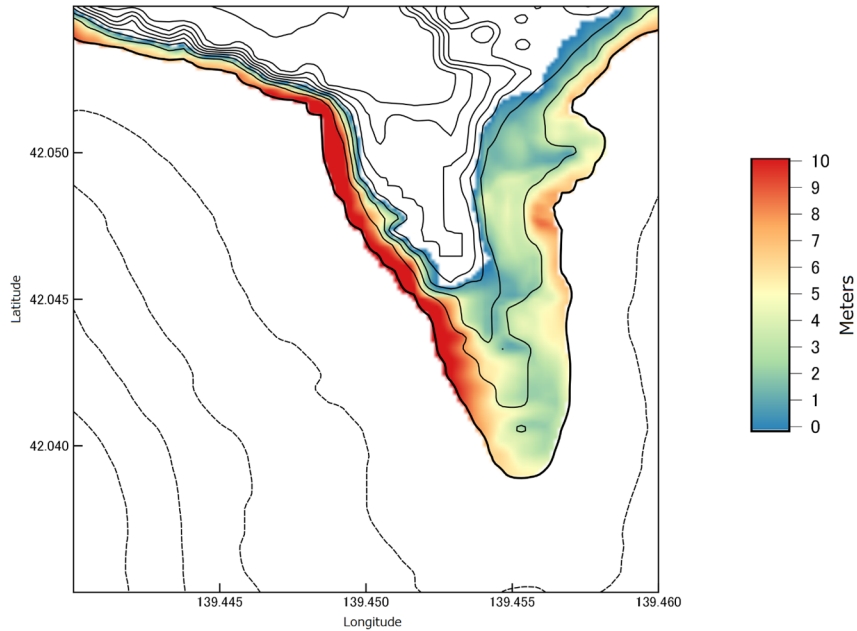
Fig. 19 Inundation map of Aonae region with 4-m contours of bathymetry and topography

The west side of the peninsula received the impact of the first wave, which produced the largest inundation height. Maximum values of nearly 15 m were obtained in the simulation. Despite a relatively lower inundation height in the east side of the peninsula, deep penetration was found due to the flatter topography in this area. The inundation on the east side was mainly produced by the second wave coming from the east. The south side of the peninsula experienced the impact of both first and second waves and run-up of over 12 m was estimated.

6.1.3.2 Arrival of first wave to Aonae

The arrival of the first wave at Aonae peninsula is shown in Fig. 20. This wave is coming from the west. Snapshots are approximately 5 seconds apart at times 4.9 min and 5.0 min to illustrate the wave arrival. From these snapshots, we estimate that the wave made impact at around 5 minutes after the tsunami generation.
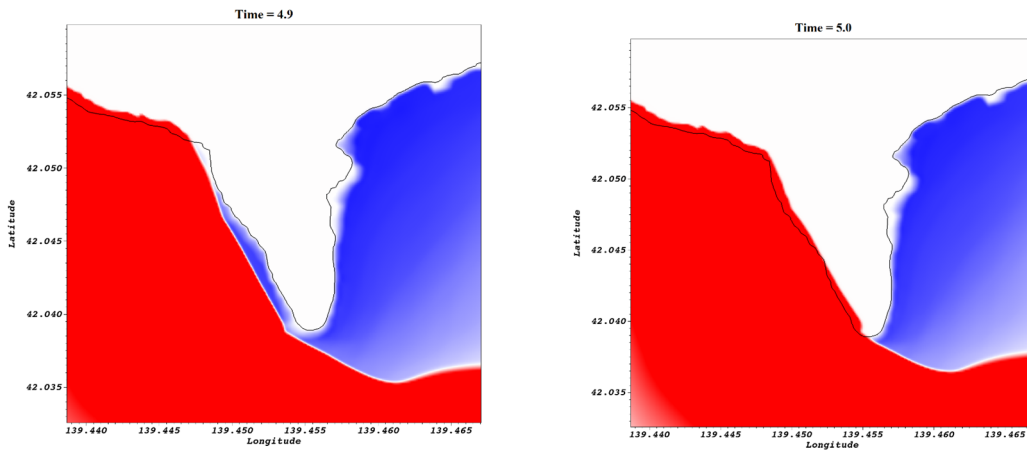
Fig. 20 Arrival wave at Aonae peninsula coming from the west, snapshots of the wave at times 4.9 min and 5.0 min after tsunami generation

5 **6.1.3.3 Two waves arriving at Aonae**
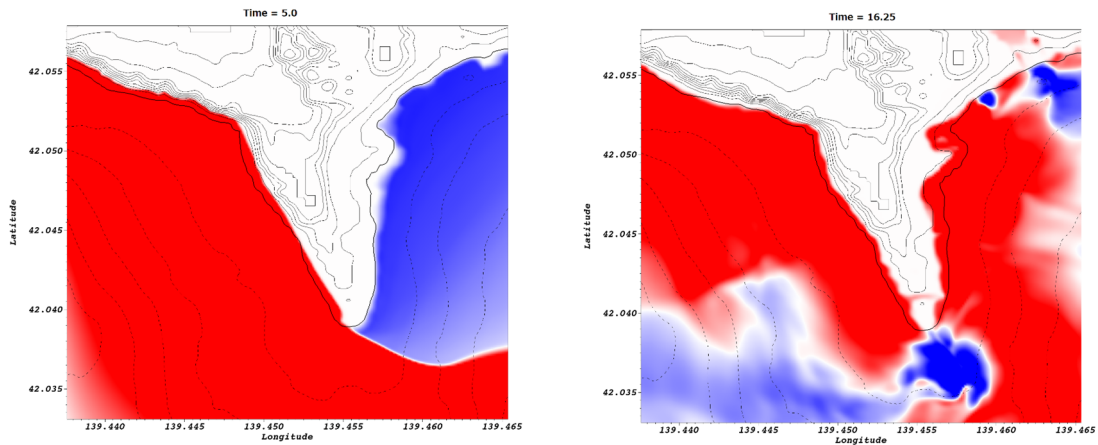
Fig. 21 Two waves arriving at Aonae peninsula. Left: first wave coming from the west arrived at around t=5 min. Right: second wave coming from the east arrived at around t=16 min

10   The two waves arriving at Aonae peninsula are shown in Fig. 21. The first one came from the west (Fig. 21 left) and made impact at around 5.0 min after the tsunami generation. The second major wave to hit the peninsula came from the east and made impact at around 16 min (Fig. 21 right). Slightly over 10 minutes separated the first and second wave.

32

6.1.3.4 Tide gauge comparison at Iwanai and Esashi

Comparison between computed and observed water levels at Iwanai and Esashi tide gauges is presented in Fig. 22. The arrival time of the computed wave shows good agreement for Esashi station. The computed wave positive and negative phases also follows rather well the observed values. In the case of Iwanai station the arrival time is slightly sooner than the observed however the observed wave phase is followed generally well in the computed results. The discrepancies between observed and computed values can be attributed to several reasons. Inaccuracies in the source used for the initial condition can influence greatly the result. Additionally, lack of realistic bathymetry including man-made structures around the area can affect the results as well.
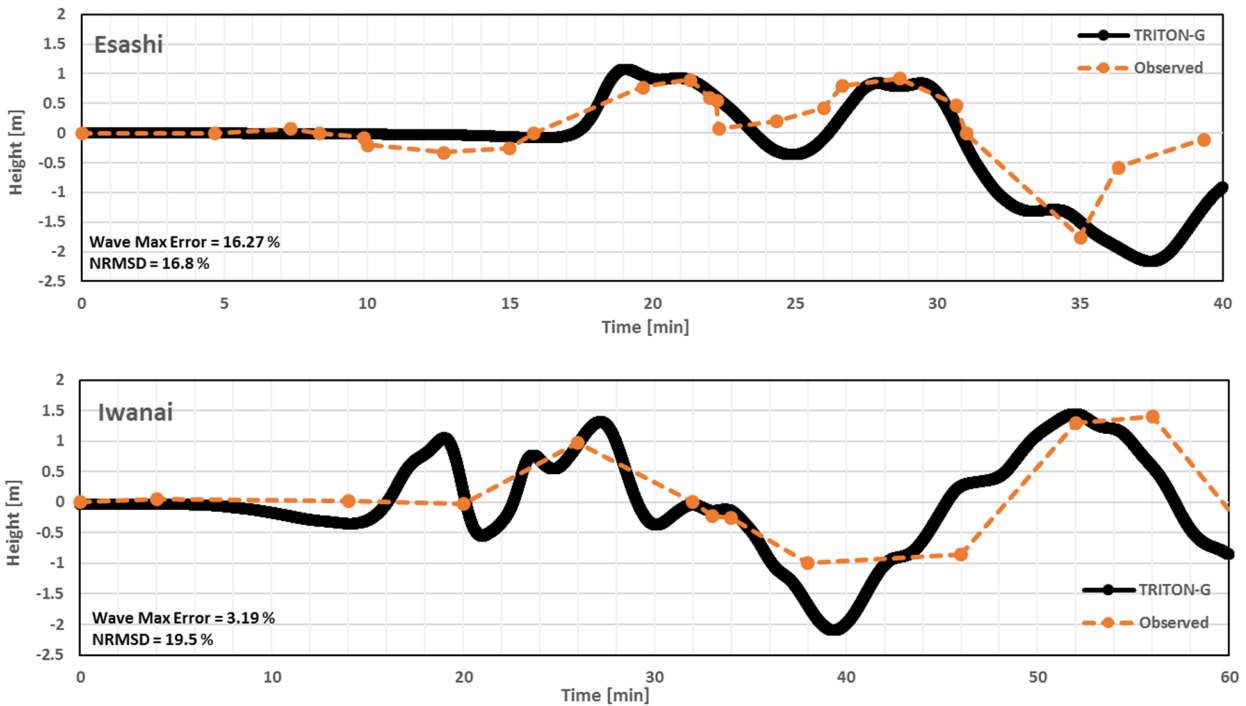
Fig. 22 Water level comparison between observations and TRITON-G results for Esashi (upper panel) and Iwanai (lower panel) tide gauges.

Inserted in each panel of Fig. 22 are the estimated errors for the gauge comparison. The maximum wave amplitude error for Esashi station is 16.27% and for Iwanai 3.19%. These are considerably lower than the mean values obtained by the models reported in the workshop (NTHMP, 2012) of 43% and 36% respectively. Although no values are reported in (NTHMP, 2012), the NRMSD error is also estimated for our model and included in the panels, both values are under 20%.

33

6.1.3.5 Maximum run-up around Okushiri

The computed maximum run-up distribution around Okushiri Island is shown in Fig. 23. Observations were taken from (Kato and Tsuji, 1994). Good agreement is found between observed and computed values around the coast. Most values are within the observed range or within a small diference from the field measurement. The simulation seems to capture well the
5   variations that occurred along the coast.

Fig. 23 Computed and observed run-up values in meters along the coast of Okushiri island.

10   The model could simulate well the maximum run-up observed around Monai valley within a reasonable 15% error. The major differences are found in the southwest side of the island where run-up values were underestimated with larger difference. The discrepancies could be explained by the use of different grid around the island coast. Additionally, the lack of an accurate high-resolution bathymetry database everywhere can also influence the computed values as well as an inaccurate initial condition.

15

6.1.3.6 Run-up height at Hamatsumae

The maximum inundation map for Hamatsumae region is shown in Fig. 24. Topography and bathymetry contours are outlined every 4 meters. A grid resolution of approximately 14 m was used for this region. Near the center of the region and to the east, run-ups of nearly 16 meters were computed. Additionally, inundation values ranging from 8 to 10 meters were obtained which match well with field observations.
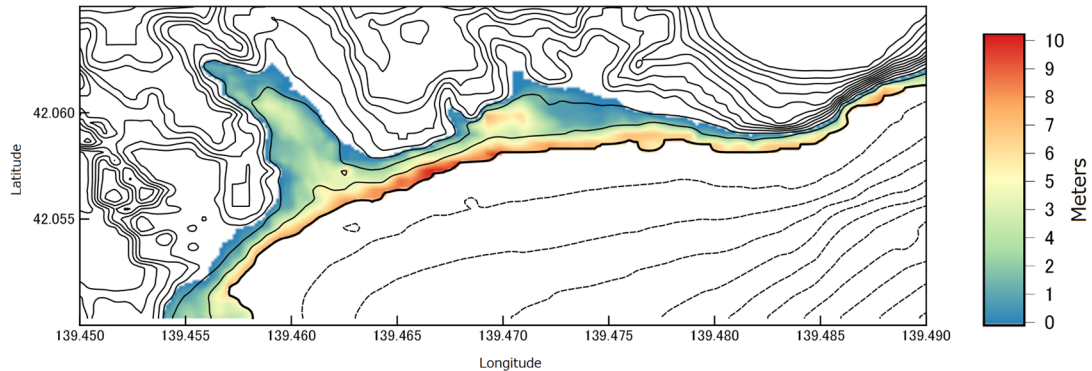
Fig. 24 Inundation map of Hamatsumae region with 4-m contours of bathymetry and topography.

6.1.3.7 Run-up height at a valley north of Monai

The maximum inundation map for the valley north of Monai is shown in Fig. 25. Topography and bathymetry contours are outlined every 4 meters. A grid resolution of approximately 7 m was used for this region. Inundation of around 26 m was computed, relatively close to the 30.6 m observed in the field.
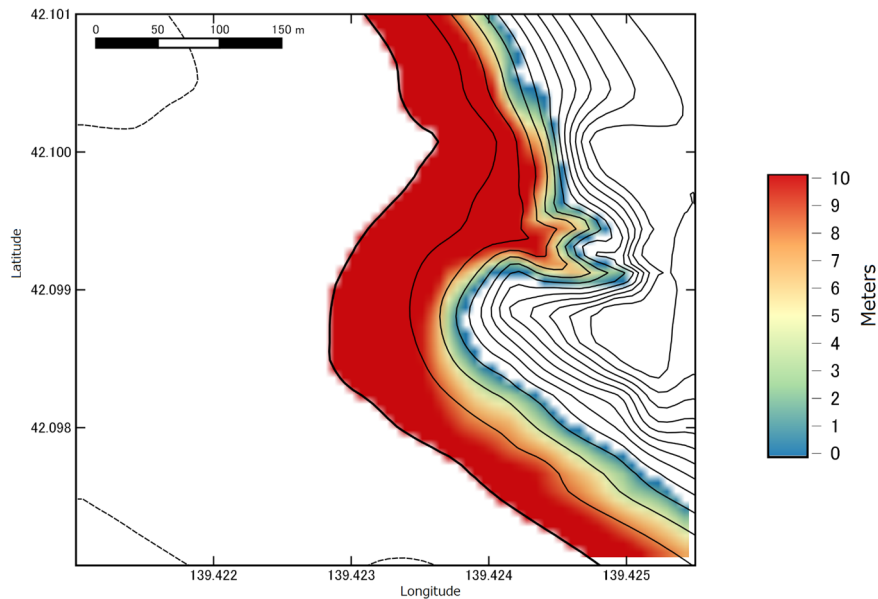
Fig. 25 Inundation map for the valley north of Monai with 4-m contours of bathymetry and topography.

7 Case study

In order to compare and validate the results of TRITON-G under a real tsunami scenario we use the hindcast of the 2004 Indonesia tsunami. Results for propagation, gauges and inundation comparison are presented.

7.1 Indonesia 2004 tsunami hindcast

This event occurred at 7:58 am on December 26th, 2004, with a magnitude of 9.0 Mw generated by the subduction of the Indian Plate by the Burma plate. Nearly 1600 km of fault was affected around 160km off the coast of Sumatra (Titov et al., 2005). This massive earthquake generated a large tsunami that spread over the Indian Ocean in the following hours.

The tsunami wave propagation computed by TRITON-G is depicted in Fig. 26. Each subsequent snapshot represents three hours after the earthquake's main event. A synoptic qualitative comparison with existing field surveys and simulations confirmed a correct propagation of the initial wave train, however to check the validity of the results, two kind of comparison are presented for tide gauge records and for inundation map simulations.

36
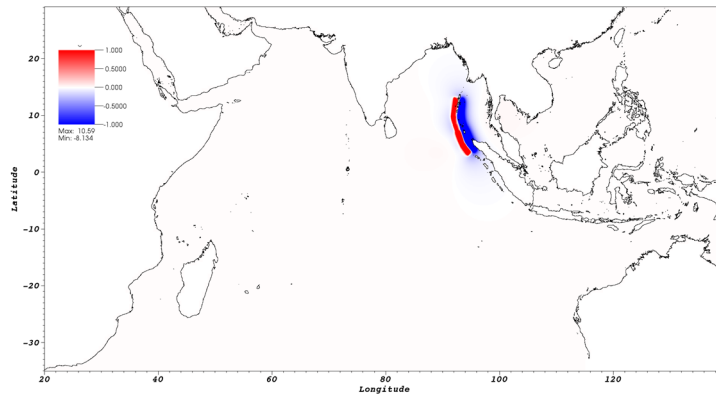
Fig. 26 (a) Time = 0 hours. Initial Source in Sumatra.

5
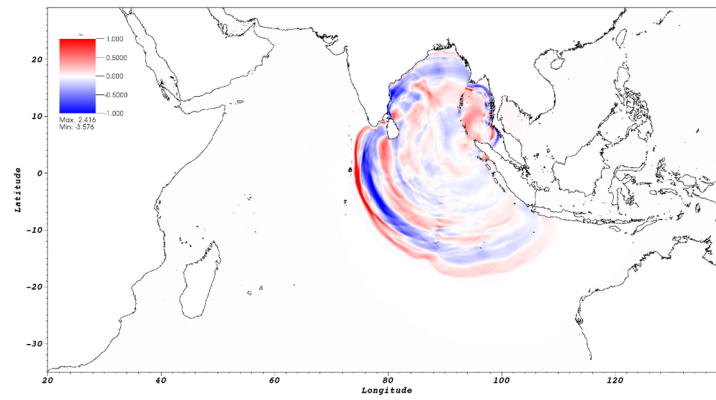
Fig. 26 (b) Time = 3 hours.
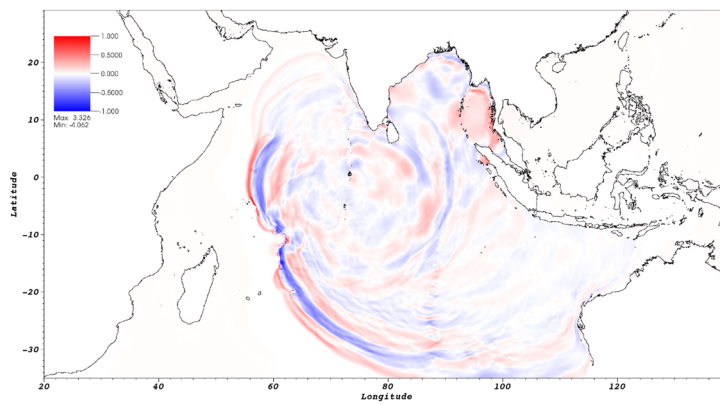
Fig. 26 (c) Time = 6 hours.

Fig. 26 Snapshots every three hours (a) – (c) of the Indonesian 2004 tsunami propagation simulated by TRITON G

7.1.1 Tide gauges comparison

To check the correctness of the wave propagation, buoys located in different parts of the Indian Ocean were used to compare TRITON-G results. These buoys measure the ocean sea level at regular intervals and serve as a critical factor to determine tsunami wave arrival times and heights. Gauges recorded at the moment of this event were obtained from NOAA's tsunami events database and inundation maps were obtained through RIMES. Results from RIMES previous operational model are also included for comparison. Their previous model was based on a customization of TUNAMI (Srivihoka et al., 2014) to include four nested grids with fixed resolution of 2 arc-min, 15 arc-second, 5 arc-second and 5/3 arc-second.

Results for five stations are shown. *Diego Garcia* Fig. 27(a) in an atoll in the Chagos Archipelago, located at 7°30'N 72° 38' E. *Male* Fig. 27(b) near the Maldives Islands, located at 4°18'N 73° 52' E. *Gan* Fig. 27 (c) near the Maldives Islands, located at 0°68'N 73° 17' E. *Colombo* Fig. 27 (d) in Sri Lanka, located at 64°93'N 79°83' E. *Point La Rue* Fig. 27 (e) near Seychelles, located at 4°68'S *55°53'* E.
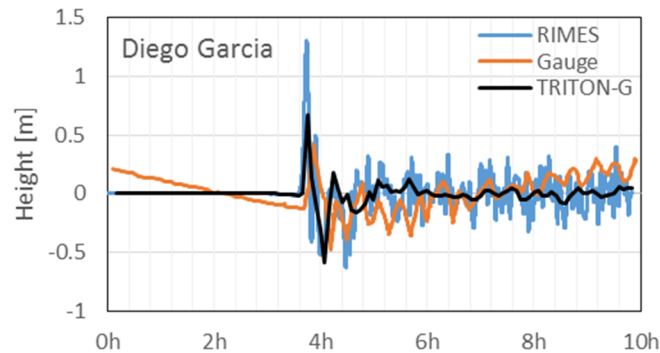
Fig. 27 (a) Comparison of arrival wave at Diego Garcia, tide gauge and model results
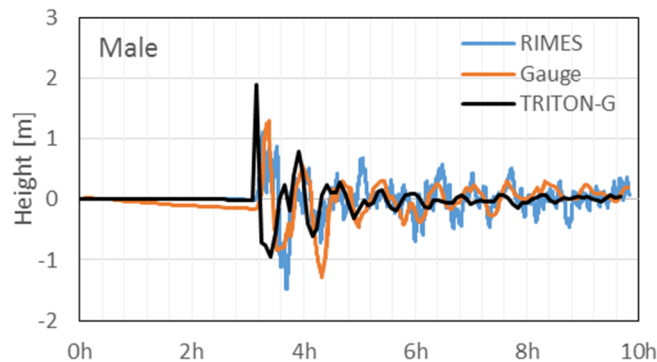
Fig. 27 (b) Comparison of arrival wave at Male, tide gauge and model results
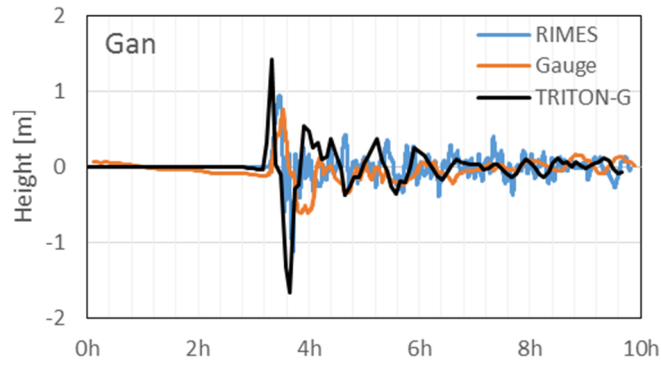
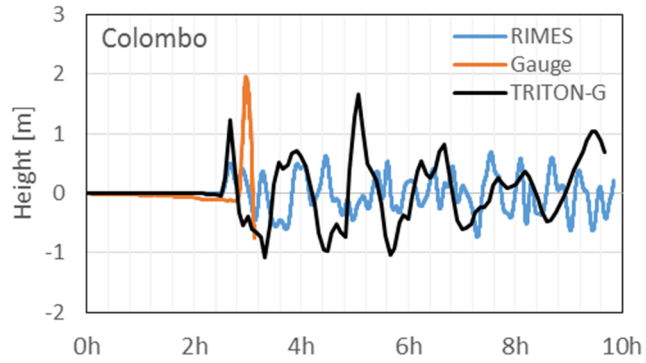**Fig. 27 (c) Comparison of arrival wave at Gan, tide gauge and model results**.

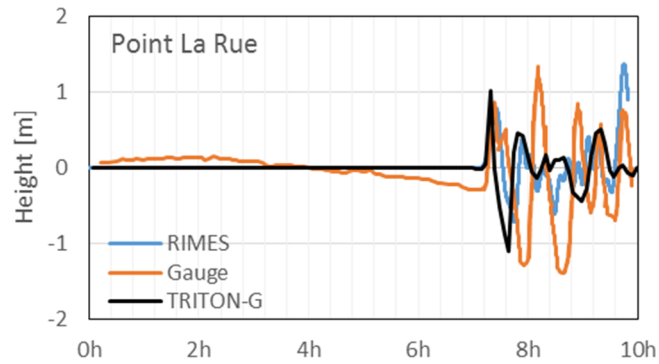**Fig. 27 (d) Comparison of arrival wave at Colombo, tide gauge and model results.**

Fig. 27 (e) Comparison of arrival wave at Point La Rue, tide gauge and model results.

The comparison between the tide gauges TRITON-G and RIMES's model based on TUNAMI are shown in Fig. 27. As it can be seen, the arrival times are in good agreement with the measured ones. The main event peaks are also reproduced in all cases with the crests' signs are in accordance with the measured values. The effect of tide is not considered in the current model which explains the height differences at initial times in the results. In the case of Male, three of the first peaks were also estimated in the simulation. The case of Diego Garcia serves also as a test for long propagation, since it is located around 2700km and there is no topography between source and station. This makes it a good way to validate that the wave is properly propagated at the right speed and no effects of diffusion on the wave height are present. Diego Garcia and Colombo (which recorded only around 3 hours before being damaged) are two examples of obtaining a more accurate and closer results than the previous model used at RIMES, where a closer height to the measured peak was obtained. Point La Rue represents also a good test for long propagation for TRITON-G numerical model since the location is over 4500km from the source and the wave has travelled over complex bathymetry and reflected on multiple coastlines. However the arrival time is still in good agreement as well as the wave arrival peak height. No effect of wave main peak diffusion is noticeable.

The time arrival differences of a few minutes between measurement and TRITON-G simulation can be partly explained by the location of simulated gauge. Even though the main events could be reproduced, a tendency to overshoot is noticed, nonetheless this did not affect the ability of the model to transport the wave along far distances and in no case an arrival wave sign was reported incorrectly. We discuss briefly three main reasons for the difference in arrival height and wave oscillation after the main event. The first is related to bathymetry and topography. Although databases for bathymetry and topography with good accuracy are available, these are still far from representing in detail the real shape of the ocean's bottom and topography. This difference makes it challenging to reproduce the wave reflections on coasts and effects of traveling through the ocean bottom completely realistic. Based on this, it is expected that some differences are found in the wave reflections and oscillations. A study about the influence on bathymetry resolution can be found in (Plant et al., 2009). The second reason relates to the dependence of every tsunami model on a good and accurate initial condition to obtain good simulations. The use of inaccurate initial fault source can affect the resulting simulation especially in locations near the source. This is particularly challenging since it is not possible to measure precisely the ocean surface at the moment of a tsunami event. The third reason is related to dispersion. Waves traveling through the Ocean bottom experience physical dispersion due to the effect of the bathymetry. In general, this dispersion is compensated by numerical dispersion introduced by the truncation error. However TRITON-G utilizes a cubic interpolation upwind scheme that has the advantage of minimizing dispersion and diffusion. An almost homogeneous traveling train wave with minimum dispersion effect is produced instead, reducing the possibility of seeing the higher oscillatory behavior of the arrival tsunami wave seen in the gauges. These kinds of discrepancies had been observed and reported on several other operational models as well (Dao and Tkalich, 2007), (Grilli et al., 2007) or (Arcas and Titov, 2006).

7.1.2 Inundation maps comparison

A further validation for TRITON-G model is to compute inundation in certain areas and compare it with field surveys or existing maps. Since inundation maps that are exactly measured do not exist, we present comparisons with RIMES' existing simulated inundation maps (RIMES, 2014) and post-tsunami field surveys. Two cases are presented, the first in Hambantota (Sri Lanka) and the second in Phuket (Thailand).

The first inundation validation presented is the result for Hambantota in Sri Lanka. The inundation map for Hambantota generated by TRITON-G is shown in Fig. 28 bottom panel. For comparison, we include in Fig. 28 top panel the previous result obtained by RIMES in their report "Tsunami Hazard and Risk Assessment and Evacuation Planning - Hambantota, Sri Lanka" (RIMES, 2014).
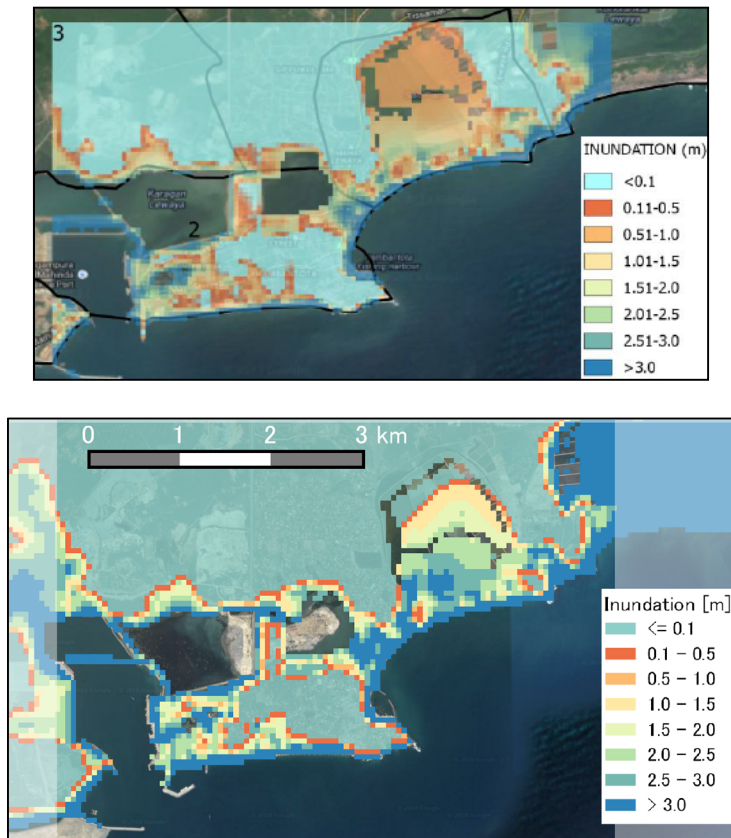
Fig. 28 Inundation comparison for Hambantota, Sri Lanka. Top: RIMES model. Bottom: TRITON G model.

Eye-witness accounts report the arrival time of the first tsunami wave around 9 am the morning of the 26th, some two hours after the initial earthquake in Sumatra. This coincides with TRITON-G's predicted arrival time of two hours for this region. According to measurements done post-tsunami, it was determined that the arrival waves had heights of over 8 meters and produced run-ups inland in certain areas of up to 2 km.
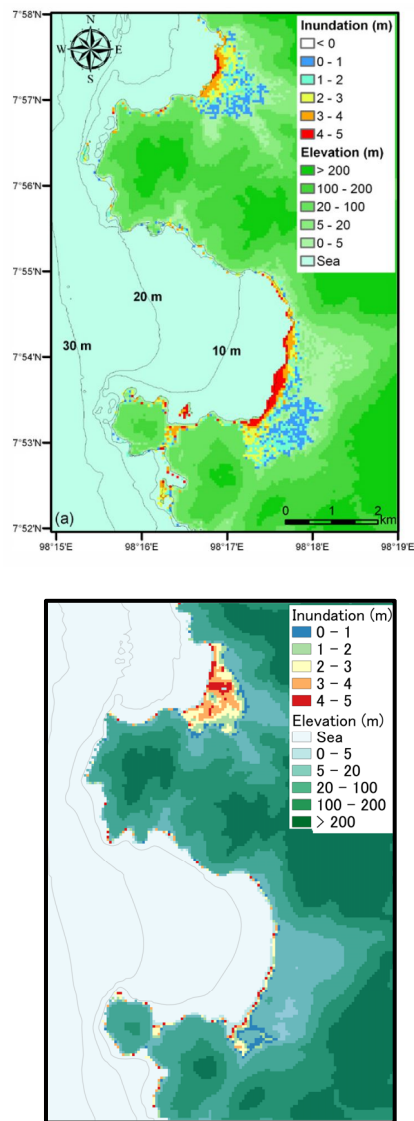
5

Fig. 29 Kamala (North) and Patong (South) inundation maps comparison. Top: inundation result by (Supparsri et al., 2011). **Bottom: TRITON G inundation result.**

10

TRITON-G inundation result also shows areas up the coastal bay where run-up produced hundreds of meters deep run-ups in land, coinciding with the recounts. By comparing it with the result provided by RIMES we found that both simulations show agreement with each other on the areas that experienced and did not experience inundation. The decisive factor that made some areas more prone to inundation than others was the topography. The arrival tsunami wave hit the coast with heights of around 8-10 meters. Coastal areas that faced the ocean with higher topographic heights were spared from being inundated. On the contrary, coast shores that were practically flat were overtaken by the incoming wave as shown in the results.

Results for the second inundation validation in Phuket are compared with those of (Supparsri et al., 2011). The wave arrival time for this region is of around 181 minutes, which agrees with the values obtained by TRITON-G model of 180 min. Inundation results are shown in Fig. 29, the image on top presents the inundation simulation obtained in the report while the image on the bottom depicts the results of TRITON-G model.

The results around the Kamala region coincide very well between models. Both report maximum inundation heights of around 5-6 meters and the run-up distances follow the same pattern. In the south, at Patong region however there is a difference in the run-up distances. This is explained by the difference in the bathymetry used by TRITON-G. While in the (Supparsri et al., 2011) study a 52m resolution was used on the entire inundation area, in our model 50m resolution bathymetry was available only in Kamala. For Patong, values were interpolated from a lower 150m resolution database which produced a smoother topography and less accurate run-up results. This highlights the importance and the effect of having accurate and realistic bathymetry for the simulation.

This test, together with the good results obtained in the inundation benchmark comparisons (Section 6 and Appendix), served to validate the ability of TRITON-G to estimate tsunami inundation.

8 Conclusions

The tragic events of recent tsunamis showed the importance of developing fast and accurate forecasting models. We implemented several techniques to reduce the time-to-solution to meet our runtime goals in the successful development of this fast and accurate tsunami operational real-time model. In a short time, wide-area simulations (ocean size) can be obtained much faster than real time, meeting our goal for results in less than 15 minutes. The combination of highly accurate numerical methods with light stencils provided an excellent solution to the governing equations, and gave stability on complex bathymetry. A customized, tree-based refinement that captured complex coastline shapes was successfully implemented using two factors; distance and focal areas. Using the distance from the coast to refine allowed to leave coarser blocks in the open ocean while blocks near the shoreline were refined to a higher 50m resolution. Focal areas were also successfully introduced in the refinement to delimit the regions where the high-resolution blocks were generated, and to use memory and computational resources efficiently. A full-GPU double precision implementation was proven successful in

delivering a large speed up. All parts of this simulation, including output storage are processed entirely on GPU with specialized kernels. For multi-GPU, the use of a weighted Hilbert space filling curve successfully generate balanced domain partitions and work load.

Using Tsubame 3.0's GPU Tesla P100 cards for a full scale simulation of 10 hours resulted on a wall clock time of just under 10 minutes with 3 GPU cards, including considerably-sized output (65GB) and using double precision. The hindcast of the Indonesian 2004 tsunami served to compare and validate TRITON-G simulation results, finding very good agreement with gauge propagation and inundations. Additionally, good agreement with standard inundation benchmark problem BP4, BP6, BP7 and BP9 was obtained. The flexibility and robustness of TRITON-G allows it to be an excellent operational model that can be easily adjusted for different tsunami scenarios, and its speed permits it to be a real-time forecasting tool. For these reasons, and under the collaboration with RIMES, TRITON-G has been successfully deployed as their operational model since August 2017.

Appendix

Numerical results for benchmarks 4, 6 and 7 are presented in this section. Detail description of the problems can be found in (NTHMP, 2012), we give a brief explanation in each section for completeness.

A1 Benchmark problem #4: Solitary wave on a simple beach – Laboratory

The domain for this test is shown in Fig. C. In this problem, the wave height H is located at a distance L from the beach toe. This test was replicated in a wave tank 31.73-cm-long, 60.96-cm-deep and 39.97-cm-wide at the California Institute of Technology. Several experiments with different water heights were performed. Benchmark Problem 4 (BP4) uses the datasets for $H/d = 0.0185$ non-breaking wave and $H/d = 0.30$ breaking wave for code validation. Results use dimensionless units with the help of parameters like length d, velocity scale $U = \sqrt{gd}$ and time scale $T = \sqrt{d/g}$.
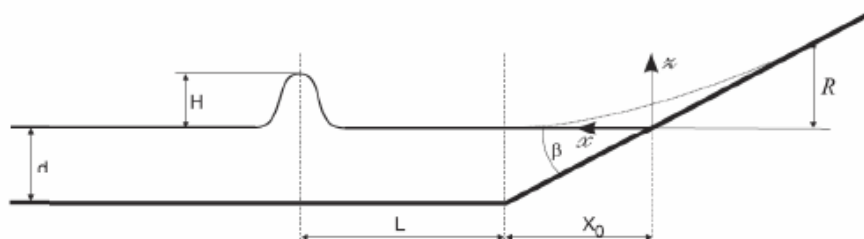

Fig. A1 Domain sketch for BP4, slope 1:19.85 (figure taken from benchmark description)

A1.1 Problem setup

- *Parameters*: d = 1, g = 9.8, case A with $H/d = 0.0185$ and case B with $H/d = 0.30$.

- *Friction*: Manning coefficient set to 0.01

- *Computational domain*: the domain along x direction spanned from x = −20 to x = 80.

- *Boundary conditions*: a non-reflective boundary condition is used at the right side of the computational domain.

- *Grid resolution*: the numerical results presented are solved with a resolution of Δx = 0.1

- CFL: 0.9

- *Initial condition*: the initial wave is computed based on the following equations for height (η) and velocity (u) respectively

$$\eta(x. 0) = H sech^2[\gamma(x - x_s)/d], \tag{22}$$

$$u(x, 0) = -\eta(x, 0)\sqrt{\frac{g}{d}}. \tag{23}$$

A1.2 Tasks to be performed

To accomplish this problem, the following tasks should be performed:

1. Compare numerically calculated surface profiles at t/T=30:10:70 for the non-breaking case $H/d = 0.0185$ with the lab data (Case A).

2. Compare numerically calculated surface profiles at t/T=15:5:30 for the breaking case $H/d = 0.30$ with the lab data (Case C).

3. Compute maximum runups for at least one non-breaking and one breaking wave case.

A1.3 Numerical results

We present the numerical results obtained using TRITON-G. Figure A2 shows the comparison between water surface level measured in the experiment and the modeled numerical results obtained by our model for times 30, 40, 50, 60 and 70 for case A ($H/d = 0.0185$). Our results show good agreement between the numerical simulation and the non-breaking experiment.

Table A1 shows the errors computed for the normalized root mean square deviation (*NRMSD*) and for the maximum wave amplitude error (*MAX*). The error values obtained by the NTHMP workshop models are also included for comparison. These values are divided into two columns, one with results for the non-dispersive models (*ND*) and the other with results for the non-dispersive and dispersive models together (labeled *ALL*).

45
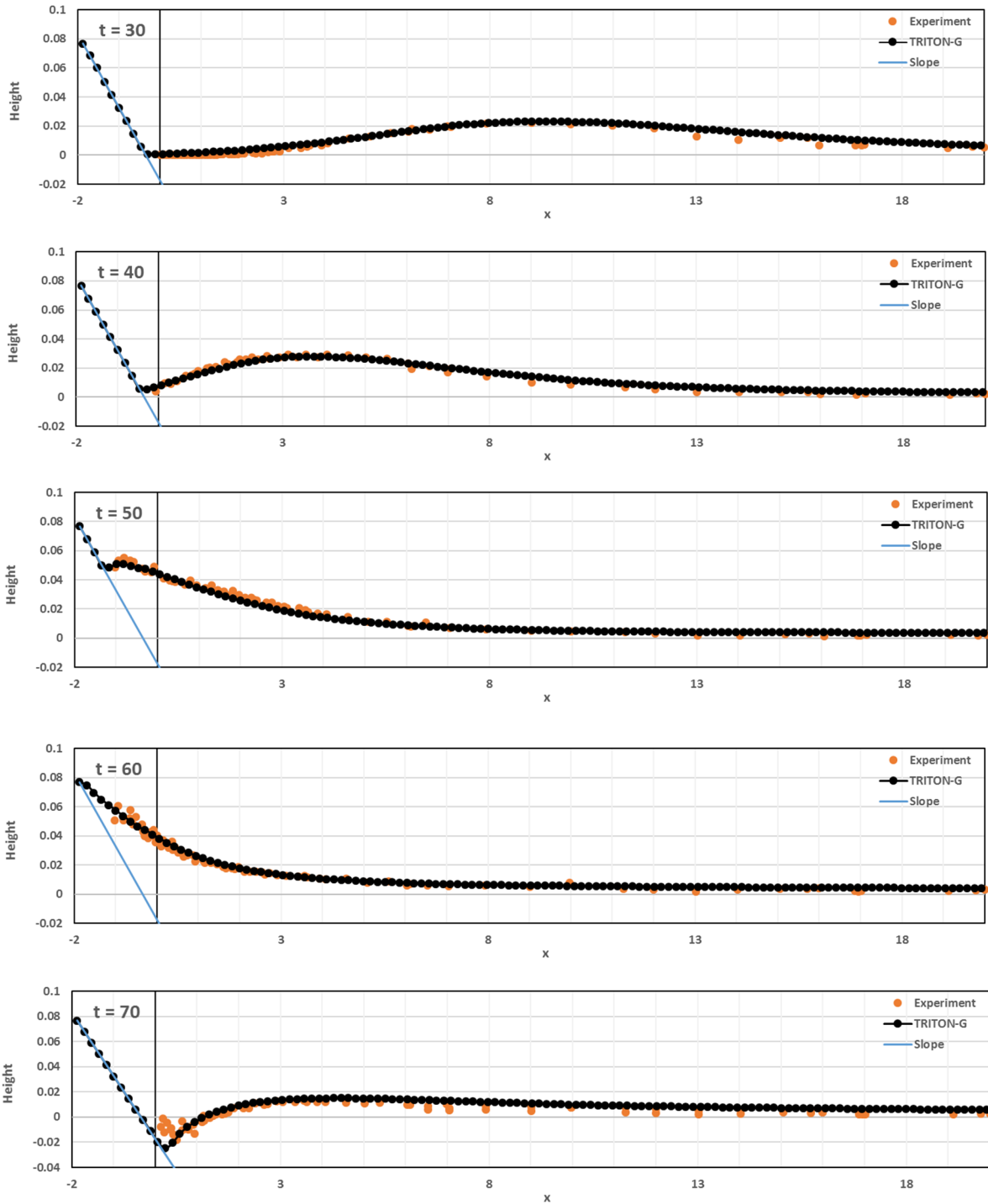
5

Fig. A2. Comparison of numerically calculated free surface profile at different dimensionless times for the non-breaking case $H/d = 0.0185$.

Errors obtained from our simulation tend to be similar or smaller than those errors obtained by other ND models, with just slight exception for time 70. Additionally, except for time 70 our errors are smaller than those obtained combining non-dispersive and dispersive mean error value.

Water level comparison for case C ($H/d = 0.30$) at times 15, 20, 25 and 30 is shown in Figure A2. Table A2 gathers the values for NRMSD and MAX errors for our numerical results and for the NTHMP workshop models. In this case, only the results of models that reported their errors are included (taken from Table 1-8, page 41 in (NTHMP, 2012)).

For case C conditions, the shallow water equations are no longer appropriate for modeling and hydrostatic models tend to produce larger differences than non-hydrostatic ones. Our numerical results in general show good agreement with the experiment.

The difference with the steepening of the crest that is noticeable in the results is expected from a hydrostatic model. In spite of that, this steeping in our model is not very large and it can trace the wave front well. Once the wave breaking occurs, our model can simulate reasonably well the run-up. This is also partly reflected in the small NRMSD error estimation obtained by our model after the wave breaking.

Maximum run-up for case A and case C were calculated. For the non-breaking case A, the obtained run-up value is 0.091 and for the breaking case C the run-up estimated is 0.588. These values are plotted in Fig. A4 with a yellow and red dot respectively, it can be seen that both values lie well within the experimental results.

		NRMSD			MAX		
		TRITON-G	NTHMP		TRITON-G	NTHMP	
H = 0.0185			ND	ALL		ND	ALL
T	30	8.8	11	11	4.0	6	4
T	40	6.6	9	8	4.8	3	3
T	50	3.5	6	5	7.4	13	7
T	60	3	4	5	1.4	1	3
T	70	11	33	16	13.5	15	9

Table A1. Model surface profile errors with respect to laboratory experimentso for case A $H/d = 0.0185$ at times 30, 40, 50, 60, and 70. Results from the NTHMP workshop errors are separated in non-dispersive (ND) models and all models (ALL).

Fig. A3. Comparison of numerically calculated free surface profile at different dimensionless times for the breaking case $H/d = 0.30$.

5

10

48

		NRMSD		MAX	
		TRITON-G	NTHMP	TRITON-G	NTHMP
H = 0.3			ALL		ALL
T	15	11.3	7	5.4	6
T	20	5.9	9	23.3	11
T	25	6.5	6	11.1	10
T	30	2.9	4	1.4	6

Table A2. Modeled surface profile errors with respect to laboratory experimentso for case A $H/d = 0.30$ at times 15, 20, 25 and 30. Results from the NTHMP workshop model errors available are shown (ALL).
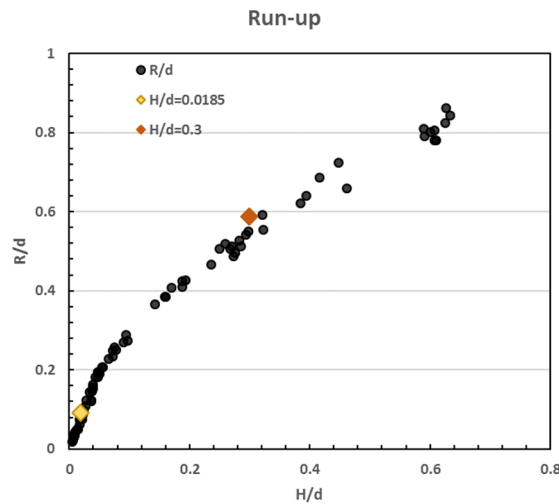
Fig. A4. Scatter plot of non-dimensional maximum run-up from a total of more than 40 experiments conducted by Y. Joseph Zhan (Synolakis, 1987)**. Orange point indicates TRITON-G result for the breaking case $H/d = 0.30$ and yellow point indicates the result for the non-breaking run-up case $H/d = 0.0185$.**

A2 Benchmark problem #6: Solitary wave on a conical island – Laboratory

The goal of this benchmark is to compare computed model results with laboratory measurements obtained during a physical modeling experiment conducted at the Coastal and Hydraulic Laboratory Engineer Research and Development Center of the U.S. Army Corps of Engineers. The laboratory physical model was constructed as an idealized representation of Babi Island, in the Flores Sea, Indonesia, to compare with Babi Island run-up measured shortly after the 12 December 1992 Flores Island tsunami (Yeh et al., 1994). Figure A5 show schematics of the experiment.

A2.1 Tasks to be performed

To accomplish this benchmark, it is suggested that, for

Case A: water depth d= 32.0 cm, target H=0.05, measured H=0.045

Case B: water depth d= 32.0 cm, target H=0.20, measured H=0.096

Case C: water depth d= 32.0 cm, target H=0.05, measured H=0.181

model simulations be conducted to address the following:

4. Demonstrate that two wave fronts split in front of the island and collide behind it
5. Compare computed water levels with laboratory data at gauge 6, 9, 16 and 22
6. Compare computed island run-up with laboratory gauge data

7.
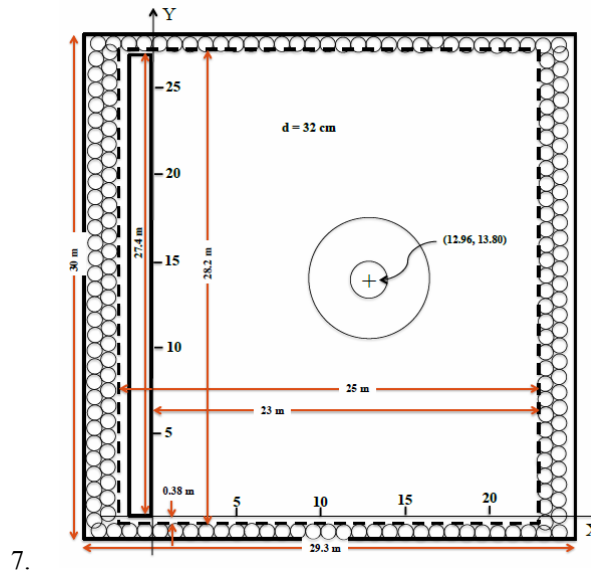
8. Fig. A5 Basin geometry and coordinate system. Solid lines represent approximate basin and wavemaker surfaces. Circles along walls and dashed lines represent wave absorbing material.

A2.2 Problem setup

- *Computational domain*: [-5,23] × [0, 28]
- *Boundary condition*: open boundaries
- *Initial condition*: same solitary wave as proposed in BP4 with the correction for two dimensions.
- *Grid resolution*: the numerical results presented are solved with a resolution of Δx = 0.05

- *CFL*: 0.9
- *Friction*: Manning coefficient set to 0.02

t = 31

t=32

5
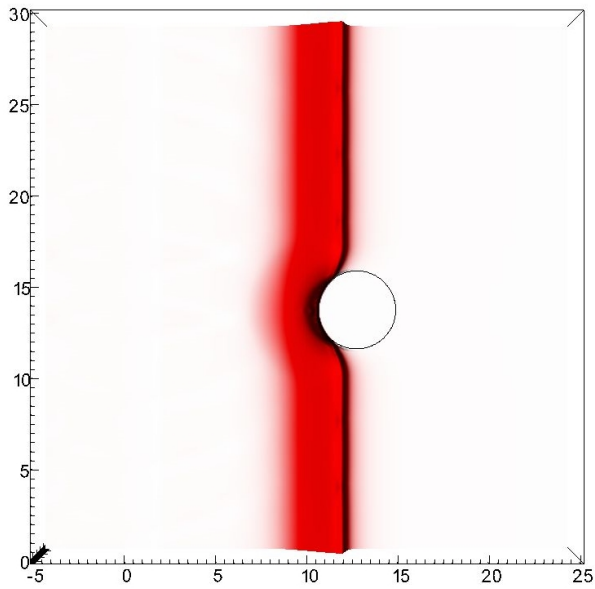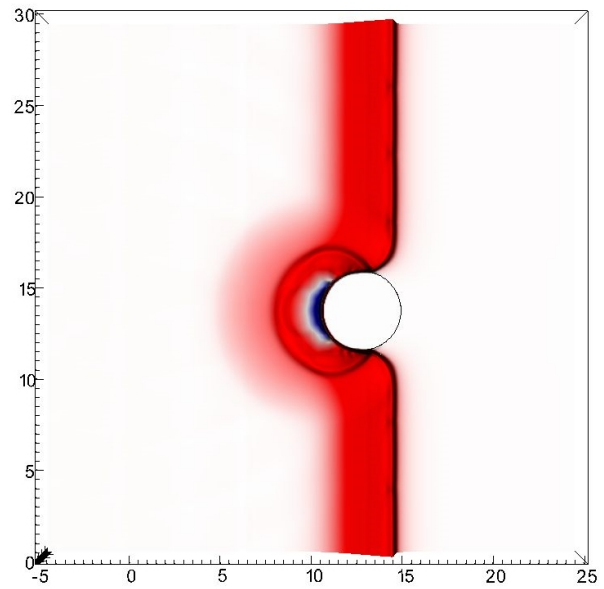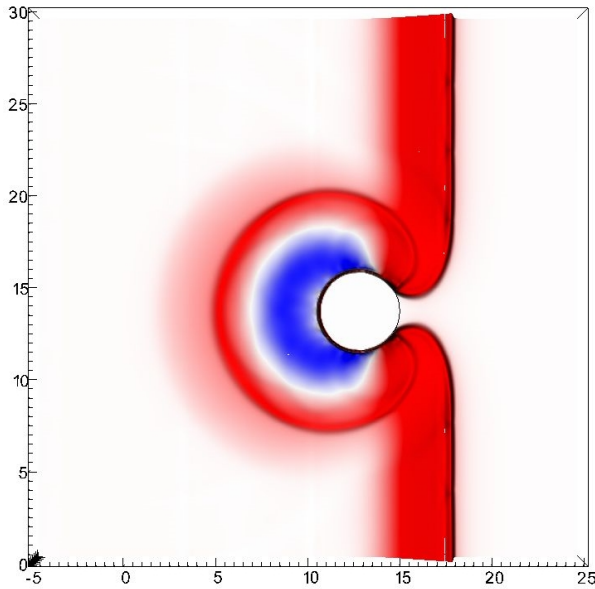
t= 33
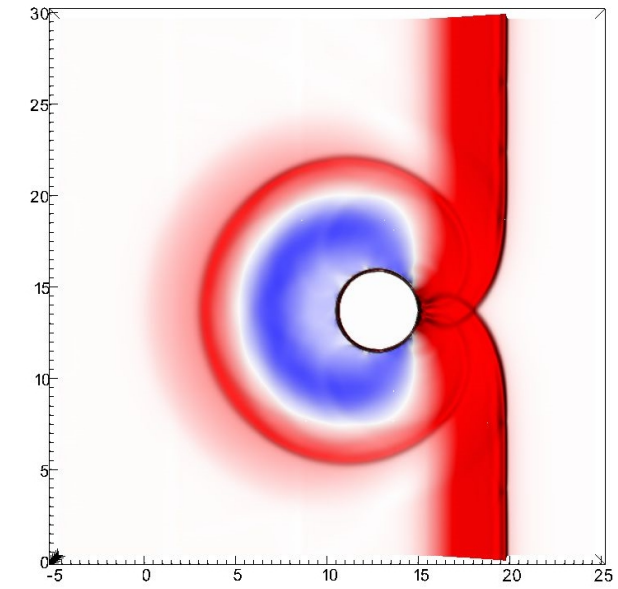
t=35

Fig. A6 Snapshots at several times showing the wavefront splitting in front of the island and colliding behind it for case B.

A2.3 Numerical results

We present the numerical results obtained using TRITON-G for the three cases (A, B and C) except for the splitting-colliding item. For this item, Figure A6 shows the wave front splitting in front of the island and then colliding again behind it for case B (H=0.096), analogue behavior was obtained for the other two cases.

5

	NRMSD			MAX		
	TRITON-G	NTHMP		TRITON-G	NTHMP	
Case A		ND	ALL		ND	ALL
Gauge 6	10	6	7	4	9	8
Gauge 9	9	7	8	4	14	10
Gauge 16	7	10	9	5	10	12
Gauge 22	9	8	8	4	25	18
Case B						
Gauge 6	10	8	8	7	6	6
Gauge 9	9	8	8	2	7	9
Gauge 16	9	7	7	14	7	7
Gauge 22	8	9	9	6	40	27
Case C						
Gauge 6	13	10	8	3	6	5
Gauge 9	12	11	11	2	9	13
Gauge 16	10	9	8	4	3	3
Gauge 22	9	8	8	10	18	15

Table A3. Water level time series TRITON-G model errors with respect to laboratory experiment data for case A, B and C. Mean values obtained for the performing NTHMP models is separated in non-dispersive models (ND) and
10 **non-dispersive and dispersive models together (ALL)**

Water level comparison uses values for gauges 6, 9, 16 and 22 for each of the 3 cases. Gauge 6 is located at $(9.36, 13.80, 31.7)$, Gauge 9 is located at $(10.36, 13.80, 8.2)$, Gauge 16 is located at $(12.96, 11.22, 7.9)$ and Gauge 22 is
15 located at $(15.56, 13.80, 8.3)$.
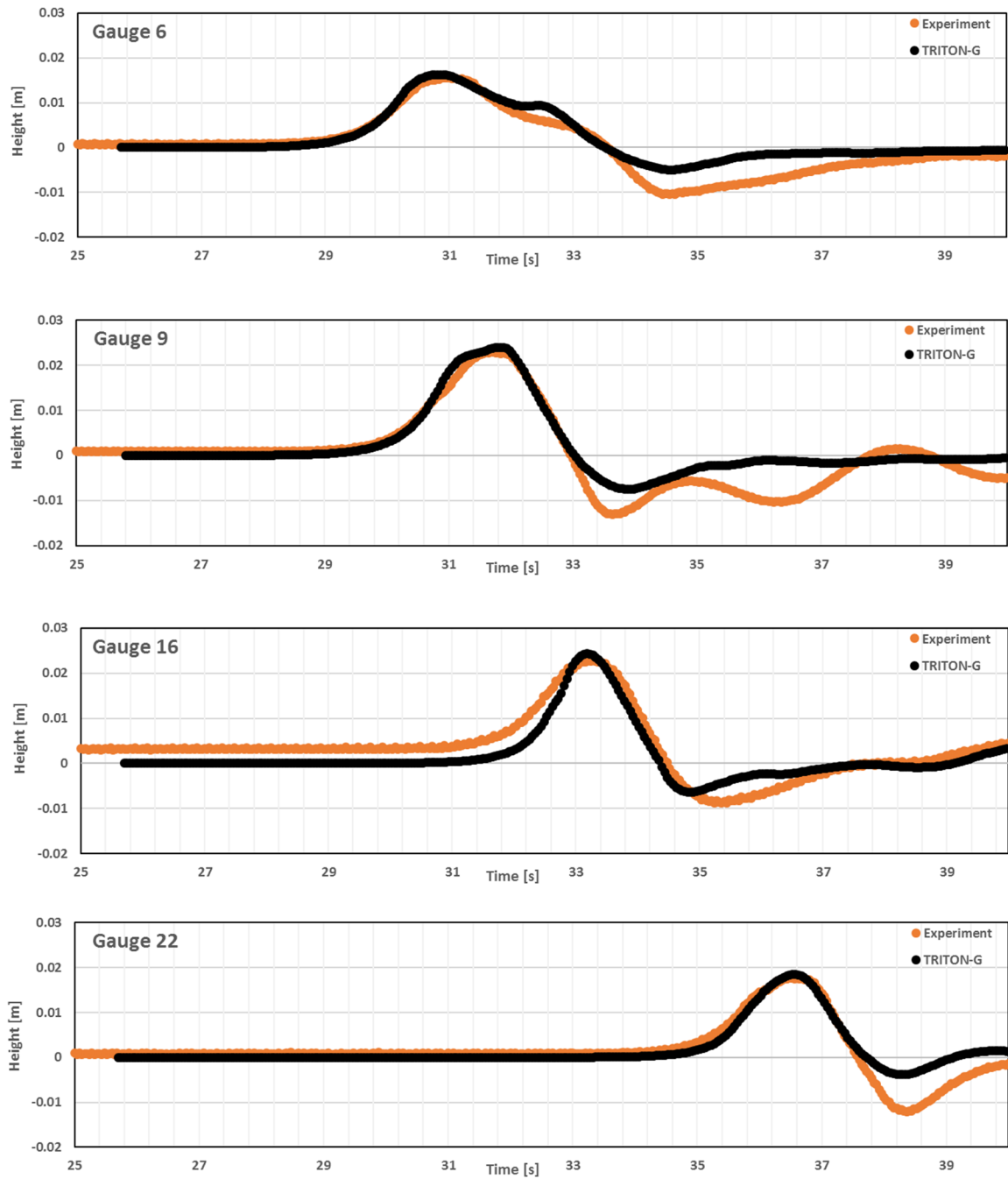
Fig. A7 Comparison between computed and measured water levels at gauges 6, 9, 16 and 22 for case A (H=0.045)
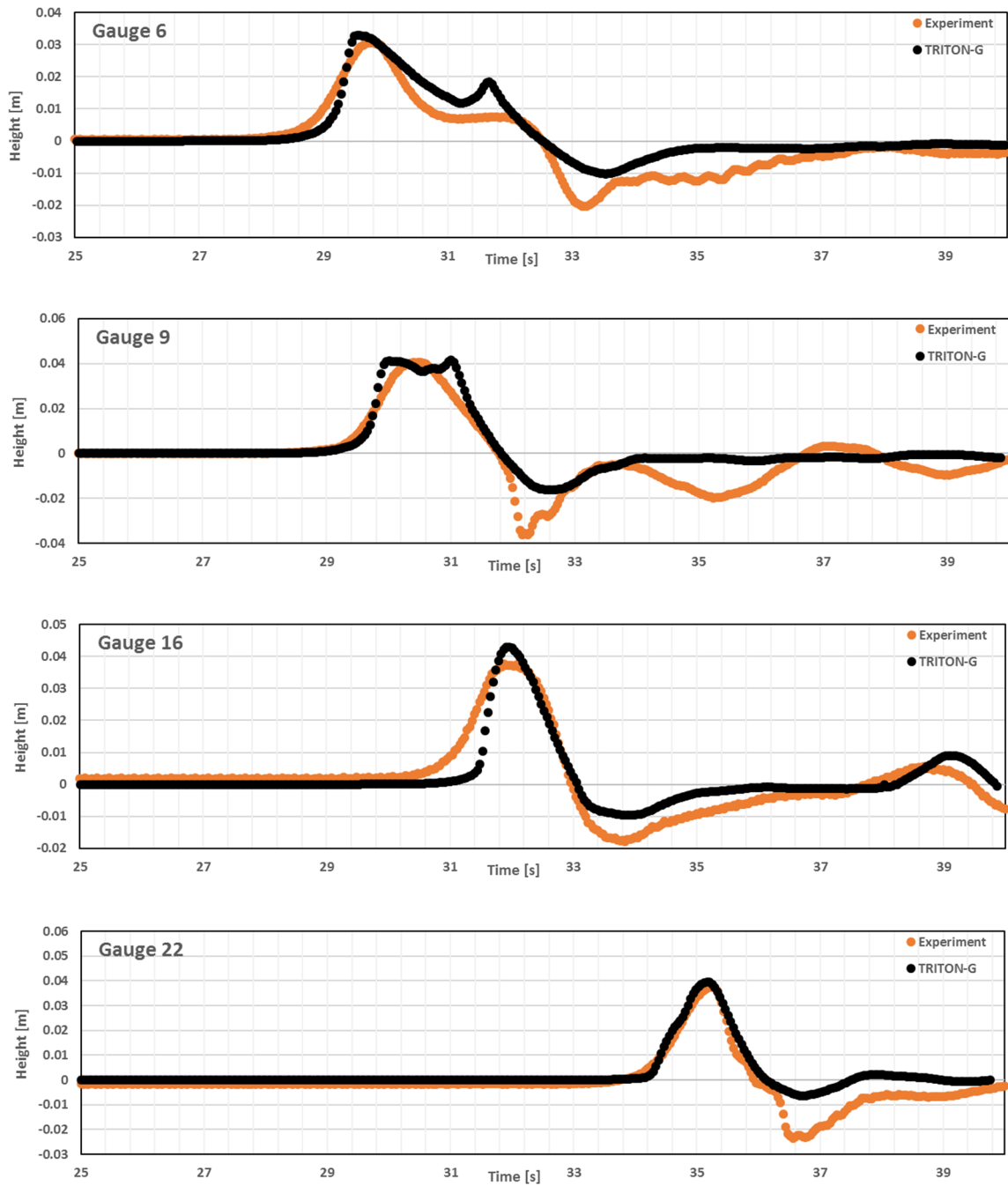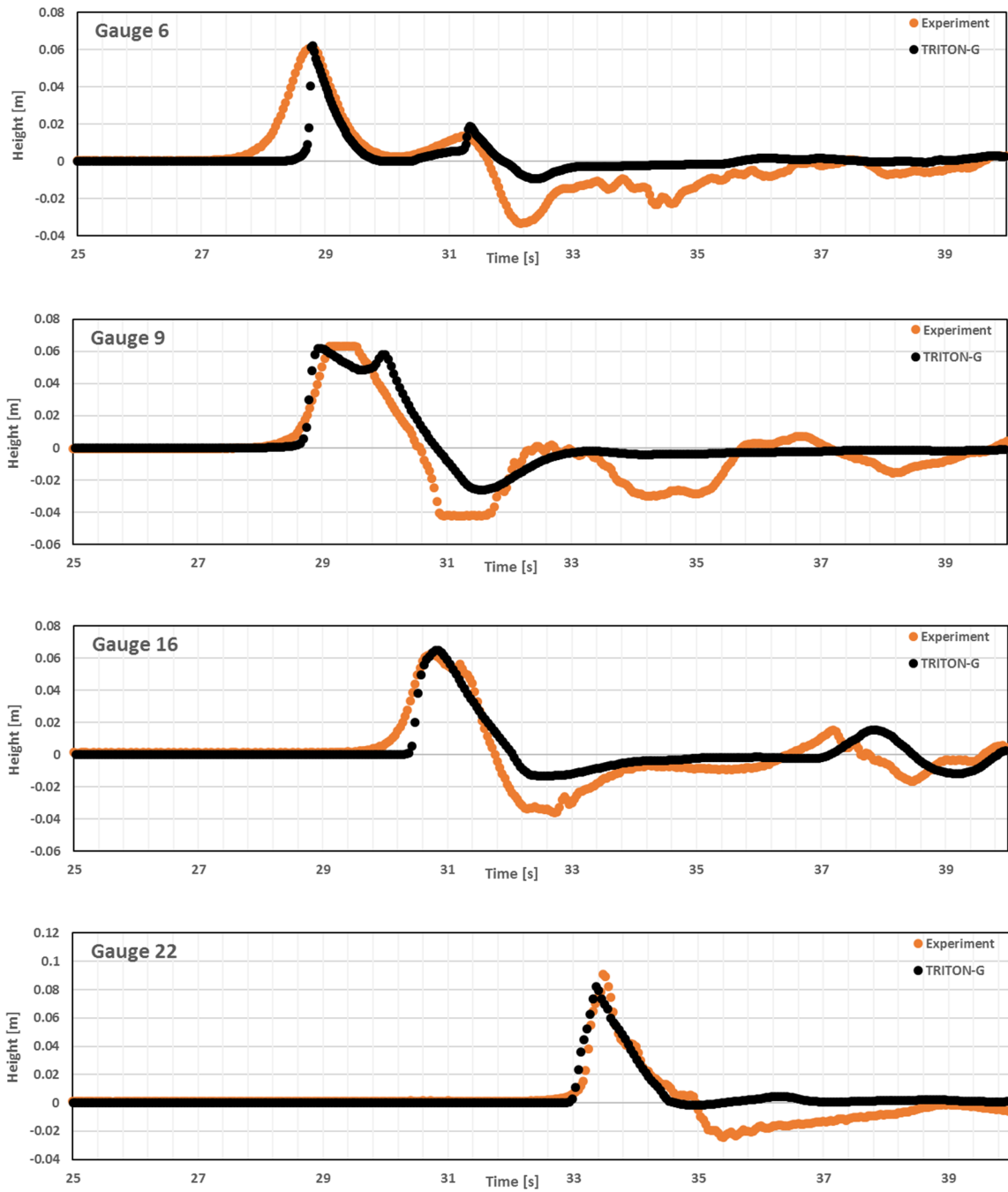
5   **Fig. A8 Comparison between computed and measured water levels at gauges 6, 9, 16 and 22 for case B (H=0.096)**

5 **Fig. A9 Comparison between computed and measured water levels at gauges 6, 9, 16 and 22 for case C (H=0.181)**
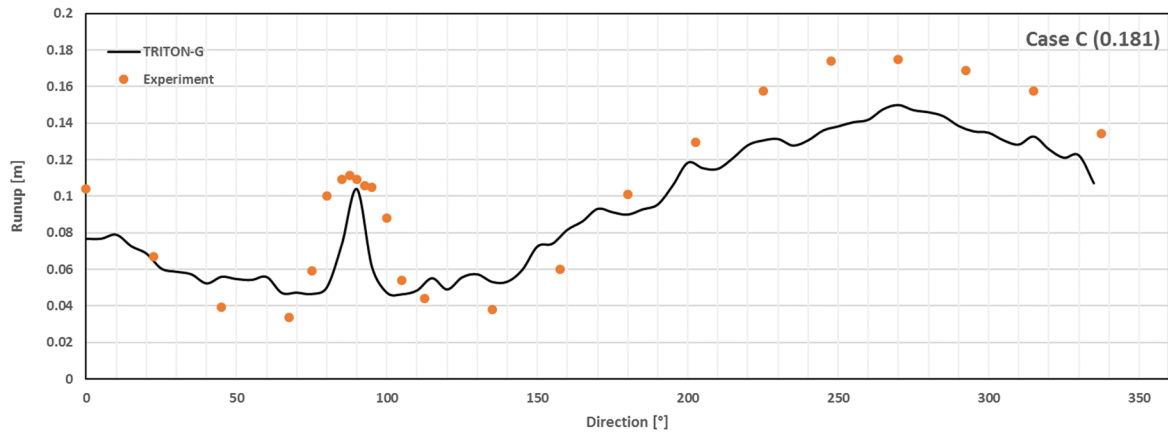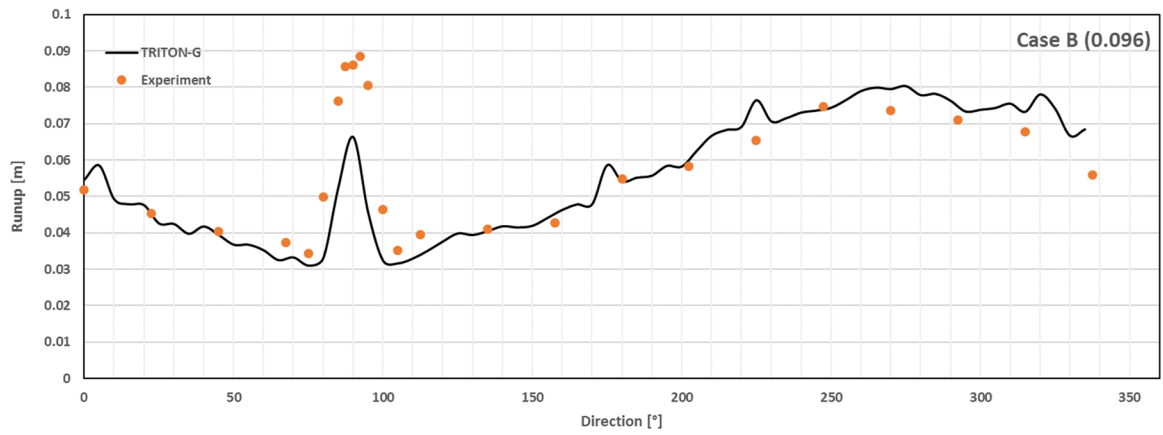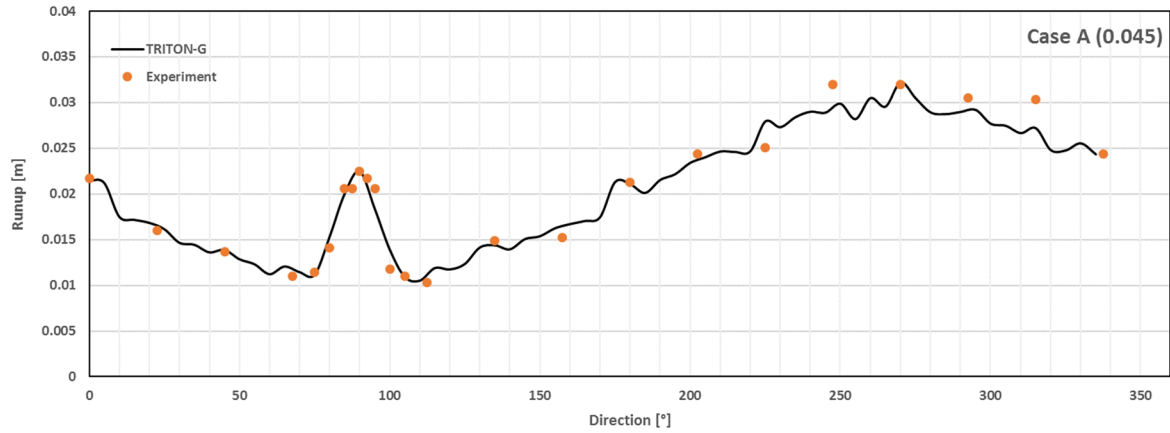
Fig. A10 Comparison between computed and measured run-up around the island for the three cases.

Numerical results for Case A, B and C are shown in Fig. A7, Fig. A8 and Fig. A9 respectively. In the three cases results were stable and in good agreement with the experimental values. The incident wave height and arrival time was captured well for all gauges. Similarly as with BP4, the steepening of the wave with increasing H is expected in a non-hydrodynamic model.

5   After the wave hit the island, some differences between experimental and model wave are noticeable as the initial wave height increased. These oscillations in the experimental data represent the effects of dispersion, which our non-dispersive numerical method is not designed to capture. Despite this, the modeled waves show good agreement with the shape of the experimental waves and the errors estimated tend to be small.

10

Runup		TRITON-G NRMSD	NTHMP ND NRMSD	NTHMP ALL NRMSD	TRITON-G MAX	NTHMP ND MAX	NTHMP ALL MAX
Case	A	9	18	18	0.6	12	7
Case	B	19	21	18	9	2	5
Case	C	20	12	11	14	5	5

Table A4. Run-up TRITON-G model errors with respect to laboratory experiment data for case A, B and C. Mean values obtained for the performing NTHMP models is separated in non-dispersive models (ND) and all models (ALL) and presented for better comparison.

15

    Table A3 gathers the normalize root mean square deviation (*NRMSD*) error and the maximum wave height (*MAX*) error. For comparison, mean errors obtained by the participating models in the NTHMP workshop are also included. These are separated in two columns, one for non-dispersive (ND) models and the other for non-disperse and disperse models together (ALL).

20   NRMSD errors for our model tend to be not very large and in similar range than those of the other non-dispersive models. In the case of the maximum height error (MAX), in almost all cases our model produced smaller error values than the non-dispersive model counterparts. Additionally, in most cases our MAX errors are smaller than those errors of the combined non-dispersive and dispersive mean values.

    Figure A10 shows the comparisons between computed and experimental run-up around the island for the three cases. 25 Case A represent the best agreement with the experimental values. Differences increased with steeper wave cases B and C as several reflections and refraction possibly occur in the basin.

    Table A4 gathers the errors obtained by our model and by the participating models in the NTHMP workshop for run-up cases A, B and C. Figure A10 showed the good agreement for Case A and this is also reflected in the NRMSD and MAX

error results. Both values are considerably smaller than those errors obtained by the NTHMP non-dispersive (ND) models and by the non-dispersive and dispersive together (ALL). For cases B and C, the errors tend to be larger but still similar to those obtained by other non-dispersive models. In all cases, the error stayed below the 20% recommended criteria.

A3 Benchmark problem #7: The tsunami run-up onto a complex 3-D beach. Laboratory.

A laboratory experiment using a large-scale tank at the central Research Institute for Electric Power Industry in Abiko, Japan was focused on modelling the runup of a long wave on a complex beach near the village of Monai (Liu et al., 2008). The beach in the tank was a 1:400-scale model of the bathymetry and topograpgy around a very narrow gully, where extreme runup was measured.

A3.1 Problem setup

The following parameters were used for the computation:

- *Grid resolution*: 393×244 was used with the same resolution 0.014 m as the bathymetry.
- *CFL*: 0.9
- *Initial condition*: water at rest.
- *Friction*: Manning coefficient set to 0.01
- *Boundary conditions*: Solid wall boundary were used at the top and bottom. At the left boundary, the given initial wave (shown in Fig. A11) was used to specify the condition up to time t=22.5 s, after that it became a wall boundary condition.
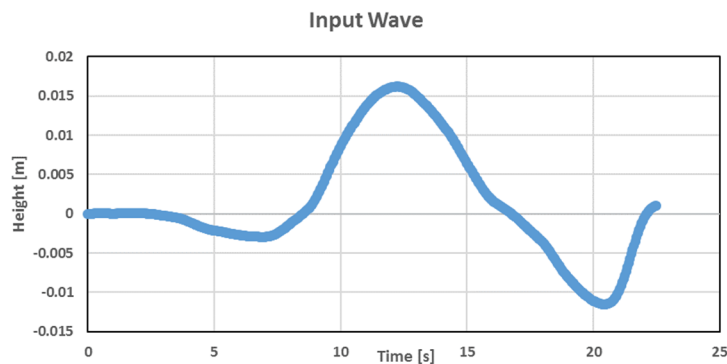
Fig. A11 Prescribed input wave for the left boundary condition, defined from t=0 to t=22.5 s

58

A3.2 Tasks to be performed

To accomplish this benchmark it is suggested to:

5. Model propagation of the incident and reflective wave accordingly to the benchmark-specified boundary condition.
6. Compare the numerical and laboratory-measured water level dynamics at gauges 5, 7 and 9.
7. Show snapshots of the numerically computed water level at the time synchronous with those of the video frames.
8. Compute maximum runup in the narrow valley.

A3.3 Numerical results

This section presents the numerical results for BP7 obtained with TRITON-G to achieve the required tasks.

The comparison with the three requested gauges 5, 7 and 9 is shown in Fig. A12 from $t = 0$ to $t = 25$ s. Good agreement is found between modeled and experimental wave for the three cases.

Values for the normalized Root Mean Square deviation error (NRMSD) and maximum wave amplitude error (MAX) were estimated for the gauge results. For gauge 5, the NRMSD error is 10% and MAX is 0.89%. For gauge 7, NRMSD is 10% and MAX is 4.81%. For gauge 9, the NRMSD error is 6.57% and MAX is 2.66%.

Comparison with the extracted movie frames is shown in Fig. A13. In the left column are the five frames provided from the laboratory recording. These are frames 10, 25, 40, 55 and 70, extracted from the video with a 0.5 s interval. We found good agreement in time and space for times 15 s to 17 s in 0.5 s increments, shown in the right column. The side-by-side comparison shows that the modeled wave follows the experimental wave front well. Additionally, the model captures the rapid run-up/run-down in the narrow gully.

Finally, the data provided by the benchmark workshop include a series of experiment tests for maximum run-up. Its maximum run-up is recorded at $\times = 5.1575$ and $y = 1.88$ m with an average value of approximately 0.09 m. In comparison, our numerical result recorded a maximum run-up at around $t = 16.5$ with a height of 0.0936 m at $\times = 5.15$ and $y = 1.88$ m.
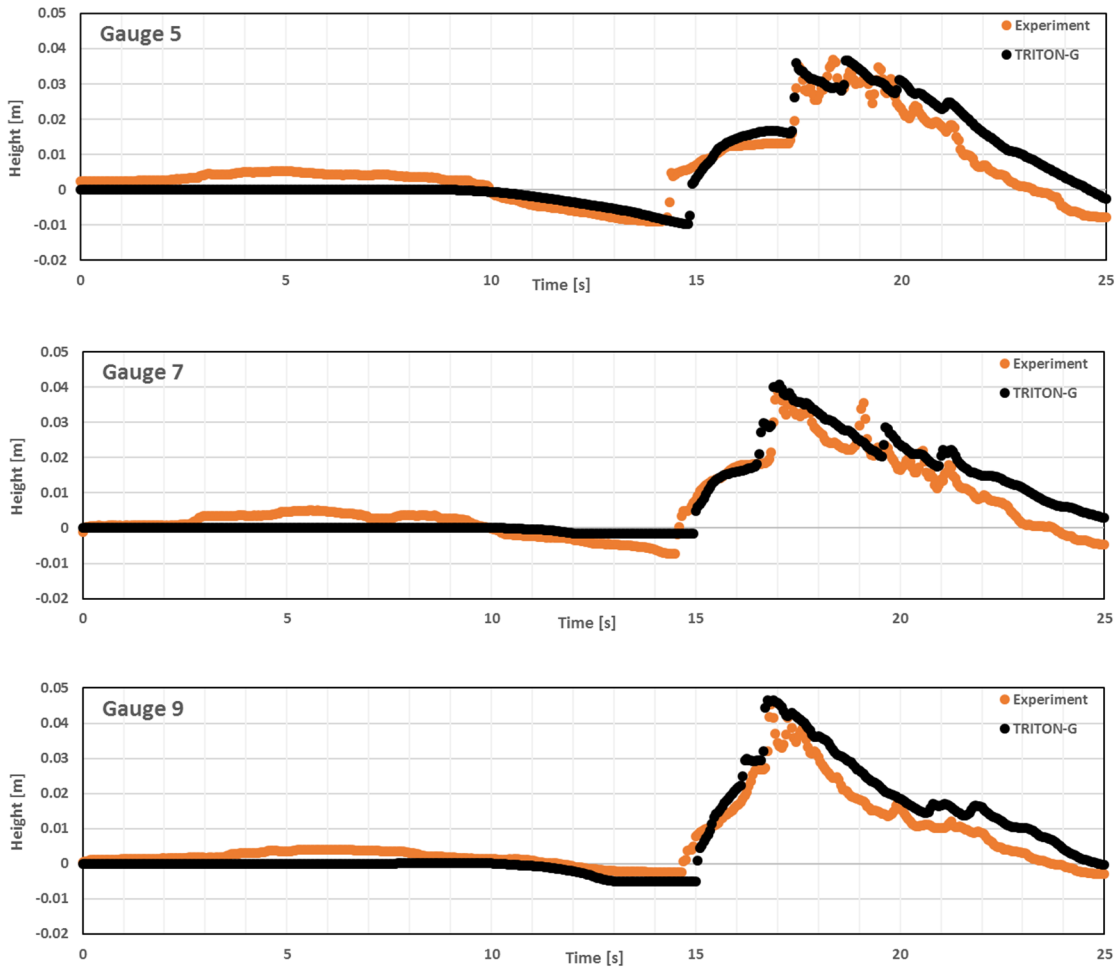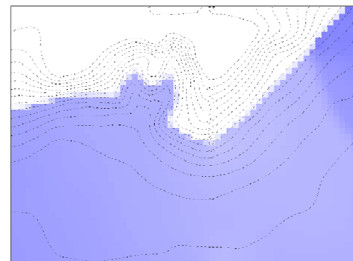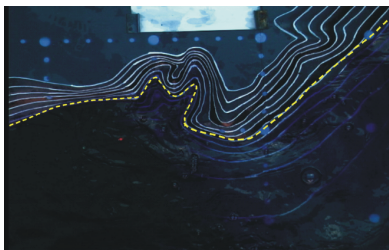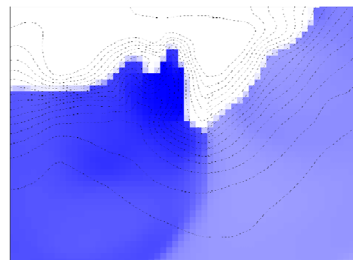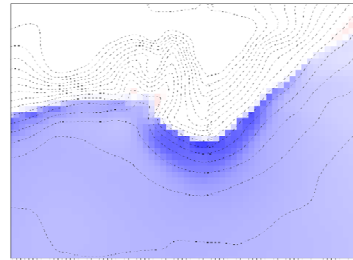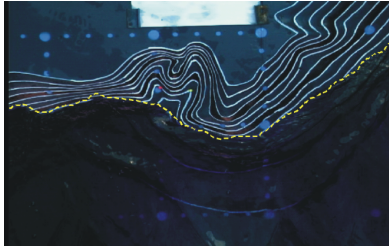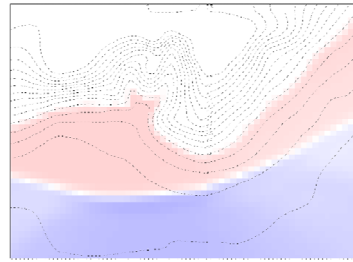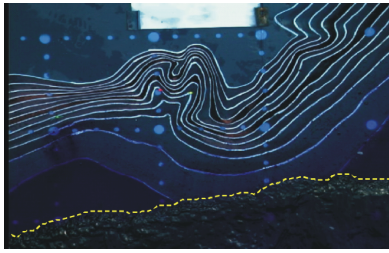
5

Fig. A12 Water level comparison for BP7 between experiment and TRITON-G for gauges 5, 7 and 9

10
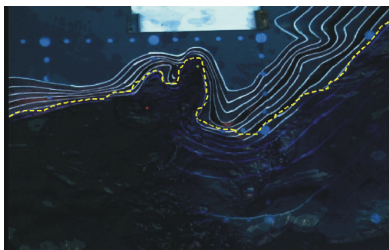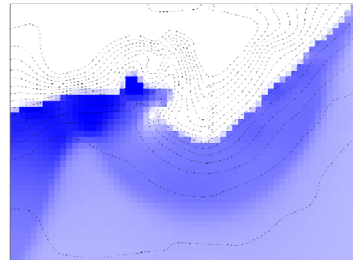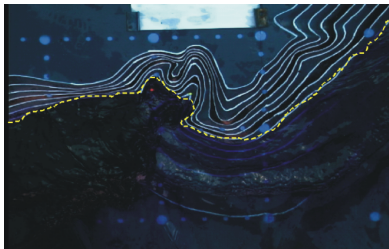
15

Fig. A13 Comparison between extracted movie frames (left) and TRITON-G simulation (right) for times 15, 15.5, 16, 16.5 and 17 seconds

5

10

61

Data availability

Underlying research data can be found in Open Science Framework repository:

Acuna, M. A., Takayuki, A.: TRITON-G, [online] Available from: osf.io/fydz8, 2017.

Competing interests

5   The authors declare that they have no conflict of interest.

15   **References**

Abadie, S. D., Morichon, S. D., Grilli, S. and Glockner, S.: Numerical simulation of waves generated by landslides using a
multiple-fluid Navier–Stokes model, Coastal Engineering, 24(9), 779–794, 2010.

Abadie, S. D., Harris, J. C., Grilli, S. T. and Fabre, R.: Numerical modeling of tsunami waves generated by the flank collapse
of the Cumbre Vieja Volcano (La Palma, CanaryIslands) : Tsunami source and near field effects, J Geophys. Res., 117, 50–
20   30, 2012.

Arcas, D. and Titov, V.: Sumatra tsunami: lessons from modeling, Surv. Geophys, 27, 679–705, 2006.

Babeyko, A.: Fast Tsunami Simulation Tool for Early Warning, [online] Available from:
https://docs.gempa.de/toast/current/apps/easywave.html, 2017.

Berger, M. and LeVeque, R.: Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, SIAM
25   J. Numer. Anal., 35, 2298–2316, 1998.

Berger, M. and Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations, J Comp. Physics, 53, 484–
512, 1984.

Bermúdez, A. and Vázquez, M.: Upwind methods for hyperbolic conservation laws, Comput Fluids, 8, 1049–1071, 1994.

Bradford, B. S; Sanders: Finite-Volume Model for Shallow-Water Flooding of Arbitrary Topography, in Conference: 17th International Conference on Coastal Engineering, vol. 129., 2002.

Burwell, D., E, T. and Chawla, A.: Diffusion and dispersion characterization of a numerical tsunami model, Ocean Modelling, 19(1-2), 10–30, 2007.

5   Colella, P. M.J; Berger: Local Adaptive Mesh Refinement for Shock Hydrodynamics, J. Comp. Physics, 82, 64–84, 1989.

Courant, R., Friedrichs, F. and Lewy, H.: On the partial difference equations of mathematical physics, IBM J, 11(2), 215–234, 1967.

Dao, M. and Tkalich, P.: Tsunami Propagation modelling - a sensitivity study, Natural Hazards and Earth System Sciences, 7, 741–754, 2007.

10 Fedkiw, S. and Osher, R.: Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag., 2003.

Fischer, G.: Ein numerisches verfahren zur errechnung von windstau und gezeiten in randmeeren, Tellus, 11, 60–76, 1959.

Gottschalk, S., Ming, L. and Manocha, D.: OBBTree: A Hierarchical Structure for Rapid Interference Detection., in ACM Siggraph '96., 1996.

Grilli, S., Ioualalen, M., Asavanant, J., Shi, J., Kirby, T. and Watts, P.: Source Constrainsts and Model Simulation of the
15 December 26, 2004 Indian Ocean Tsunamia, Port, Ocean and Coastal Engineering, 133(6), 414–428, 2007.

Hansen, W.: Theorie zur errechnung des wasserstands und derstromungen in randemeeren, Tellus, 8, 287–300, 1956.

Horrillo, J., Wood, G., Kim, B. and Parambath, A.: A simplified 3-D Navier–Stokes numerical model for landslide tsunami: Application to the Gulf of Mexico, J Geophysics Res Oceans, 118, 6934–6950, 2013.

Imamura, F.: Review of tsunami simulation with a finite difference method., Word Scientific Publishing Co., 1996.

20 Imamura, F., Goto, C., Ogawa, Y. and Shuto, N.: Numerical Method of Tsunami Simulation with the Leap-Frog Scheme, IUGG/IOC Time Project Manuals., 1995.

Kato, K. and Tsuji, Y.: Estimation fo fault parameters of the 1993, Hokkaido-Nansei-Oki earthquake and tsunami characteristics., Bull. Earthq. Rest. Inst., 69, 39–66, 1994.

Kirby, J. T., Fengyan, S., Babak, T., Harrisb, J. C. and Stephan, T.: Dispersive tsunami waves in the ocean: Model equations
25 and sensitivity to dispersion and Coriolis effects, Ocean Modelling, 62, 39–55, 2013.

Lawrence Livermore National Laboratory, L.: Silo User's Guide, [online] Available from: https://wci.llnl.gov/codes/silo/media/pdf/LLNL-SM-453191.pdf, 2017.

LeVeque, R. and George, D.: Advanced Numerical Models for Simulating Tsunami Waves and Runup, pp. 43–74, World Scientific World Scientific Publishing., 2014.

30 LeVeque, R. J.: Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm, J. Comput. Phys., 146(1), 346–365, 1998.

LeVeque, R. J.: Finite volume methods for hyperbolic problems, Cambridge University Atlas de Radiologie Clinique de la Presse Medicale., 2002.

Liu, P. L., Yeh, H. and Synolakis, C.: Advanced numerical models for simulating Tsunami waves and runup. Advances in coastal and ocean engineering., New Jersey: World Scientific, 10, 2008.

5   Liu, P. W. G. C.: COMCOT, Cornell Multi-grid Coupled Tsunami Model, [online] Available from: http://223.4.213.26/archive/tsunami/cornell/comcot.htm, 1998.

Lynett, P., Wu, T. and Lui, P.: Modeling wave runup with depth-integrated equations, Coastal Engineering, 46(2), 89–107, 2002.

Macías, J., Castro, M. J., Ortega, S., Escalante, C. and González-Vida, J. M.: Performance benchmarking of Tsunami-
10 HySEA model for NTHMP's inundation mapping activitie, Pure and Applied Geophysics, 174, 3147–3183, 2017.

Moller, T., Hoffman, N. and Haine, E.: Real-Time Rendering, AK Peters Ltd., 1999.

Motoki, K. and Toshihiro, N.: Damage statistics (Summary of the 2011 off the Pacific Coast of Tohoku Earthquake damage), Soils and Foundations, 52(5), 780–792, 2012.

Nakamura, T., Tanaka, R., Yabe, T. and Takizawa, K.: Exactly conservative semi-Lagrangian scheme for multi-dimensional
15 hyperbolic equations with directional splitting technique, J. Comput. Phys, 174, 171–207, 2001.

Nicolsky, D., Sileimani, E. and Hansen, R.: Validation and verification of a numerical model for tsunami propagation and runup, Pure and Applied Geophysics, 168(6), 1199–1222, 2011.

NTHMP: NTHMP, 2012. National Tsunami Hazard Mitigation Program (NTHMP). 2012. Proceedings and Results of the 2011 NTHMP Model Benchmarking Workshop, Department of Commerce/NOAA/NTHMP; NOAA Special Report., 2012.

20 NVIDIA: CUDA Zone, [online] Available from: https://developer.nvidia.com/cuda-zone, 2017a.

NVIDIA: Tesla P100 Datasheet, [online] Available from: https://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-PCIe-datasheet.pdf, 2017b.

Nwogu, O.: An alternative form of the Boussinesq equations for nearshore wave propagation, Coastal, and Ocean Engineering, 119, 618–638, 1993.

25 Oceans (GEBCO), T. G. B. C. of the: GEBCO, [online] Available from: http://www.gebco.net/, 2017.

Ogata, Y. and Takashi, Y.: Multi-Dimensional Semi-Lagrangian Characteristic Approach to the Shallow Water Equations by the CIP Method, International Journal of Computational Engineering Science, 05(03), 2004.

Peregrine, D.: Long waves on a beach, Journal of Fluid Mechanics, 27(4), 815–827, 1967.

Plant, N., Kacey, E., Kaihatu, J., Veeramony, J., Hsu, L. and Todd, H.: The effect of bathymetric filtering on nearshore
30 process model results, Coastal Engineering, 56(4), 484–493, 2009.

Reed, D.: User Datagram Protocol (UDP), RFC 768., 1980.

Regional Integrated Multi-Hazard Early Warning System, R.: RIMES, [online] Available from: http://www.rimes.int/, 2017.

RIMES: Tsunami Hazard and Risk Assessment and Evacuation Planning - Hambantota, Sri Lanka, Regional Integrated Multi-Hazard Early Warning System., 2014.

Roeber, V. and Cheung, K. F.: Boussinesq-type model for energetic breaking waves in fringing reef enviroments, Coastal Engineering, 70, 1–20, 2012.

5 Rusanov, V.: Characteristics of the general equations of gas dynamics, Zhurnal Vychislistelnoi Mathematiki Mathematicheskoi Fiziki, 3, 508–527, 1963.

Sagan, H.: Space-Filling Curves, Universitext., 1994.

Shi, F., Kirby, J. T., Geiman, J. D. and Grilli, S.: A high-order adaptive time-stepping TVD solver for Boussinesq modeling of breaking waves and coastal inundation, Ocean Modeling, 43, 36–51, 2012.

10 Smylie, L. and Mansinha, D. E.: The Displacement Fields of Inclined Faults, B. Seismological Soc. Ame., 61(5), 1433–1440, 1971.

Srivihoka, P., Honda, K., Ruangrassamee, A., Muangsinc, V., Naparatb, P., Foytong, P., Promdumrong, N., Aphimaeteethomrong, P., Intaveec, A., Layug, J. E. and Kosinc, T.: Development of an online tool for tsunami inundation simulation and tsunami loss estimation, Continental Shelf Research, 79, 3–15, 2014.

15 Stoker, J. J.: Water Waves: The Mathematical Theory with Applications, Wiley-Interscience., 1992.

Supparsri, A., Koshimura, S. and Imamura, F.: Developing tsunami fragility curves based on the satellite remote sensing and the numerical modeling of the 2004 Indian Ocean tsunami in Thailand, J Natural Hazards and Earth Sc., 11, 173–189, 2011.

Swarztrauber, P. N., Williamson, D. L. and Drake, J. B.: The cartesian method for solving partial differential equations in spherical, Dynamics of Atmospheres and Oceans, 27, 679–706, 1997.

20 Synolakis, C. E.: The runup of solitary Waves, J. Fluid Mechanics, 185, 523–545, 1987.

Szauer, G.: Game Physics Cookbook, Amazon Digital Services., 2017.

Takahashi, T.: Benchmark Problem 4. The 1993 Okushiri tsunami. Data collected, Conditions and Phenomena, in Long waves runup models, edited by H. Yeh, P. Piu, and C. Synolakis, pp. 384–403, Word Scientific Publishing Co., 1996.

Thacker, W. C.: Some exact solutions to the nonlinear shallow-water wave equations, J. Fluid Mechanics, 107, 499–508, 25 1981.

Titov, V. and Synolakis, C.: Evolution and runup of the breaking and nonbreaking waves using VTSC2, J. Waterway, Port, Coastal and Ocean Eng., 126(6), 308–316, 1995.

Titov, V., Rabinovich, A., Mojfeld, H., Thomson, R. and Gonzales, F.: The Global Reach of the 26 December 2004 Sumatra Tsunami, Science, 309(5743), 2045–2048, 2005.

30 Toro, F.: Shock-capturing methods for free-surface shallow flows, John Wisley&Sons AK Peters AK Peters Ltd., 2010.

Tsubame, T. I. of T.: Manual Tsubame 3.0, [online] Available from: http://www.t3.gsic.titech.ac.jp/, 2017.

Utsumi, T., Kunugi, T. and Aoki, T.: Stability and accuracy of the cubic interpolated propagation scheme, Computer Physics Communications, 101, 9–20, 1997.

Vazhenin, A., Lavrentiev, M., Romanenko, A. and Marchuk, A.: Acceleration of tsunami wave propagation modeling based on re-engineering of computational components, International Journal of Computer Science and Network Security, 13, 32–70, 2013.

Vincent, S., Caltagirone, J. P. and Bonneton, P.: Numerical modeling of bore propagation and run-up on sloping beaches using a MacCormack TVD scheme, Journal of Hydraulic Research, 39, 41–49, 2001.

Wang, D., Becker, N. C., Walsh, D., Fryer, G. J., Weinstein, S. A., McCreery, C. S., Sardina, V., Hsu, V., Hirshorn, B. F., Hayes, G. P., Duputel, Z., Rivera, L., Kanamori, H., Koyangai, K. and Shiro, B.: Real-time forecasting of the April 11, 2012, Sumatra Tsunami, Geophys. Res. Lett., 39(19), L19601, 2012.

Wei, G., Kirby, J., Grilli, S. T. and Subramanya, R.: Fully nonlinear Boussinesq model for free surface waves. Part 1: Highly nonlinear unsteady waves, J Fluid Mech, 294, 71–92, 1995.

WHO, W. H. O.: Indonesia situation reports, [online] Available from: http://www.who.int/hac/crises/idn/sitreps/en/, 2014.

Williamson, D. L., Drake, J. B., Hack, J. J., Jakob, R. and Swarztraube, P. N.: A standard test set for numerical approximations to the shallow water equations in spherical geometry, J. Comput. Phys, 102, 211–224, 1992.

Yabe, T. and Aoki, T.: A universal solver for hyperbolic equations by Cubic-Polynomial Interpolation I. One-dimensional solver, Comp. Physic Comm, 66, 219–232, 1991.

Yabe, T., Tanaka, R., Nakamura, T. and Xiao, F.: An Exactly Conservative Semi-Lagrangian Scheme (CIP–CSL) in One Dimension, Monthly Weather Rev., 129, 332–344, 2001.

Yamamoto, S. and Daiguji, H.: Higher-order-accurate upwind schemes for solving the compressible Euler and Navier-Stokes equations, Computers and Fluids, 22(2), 259–270, 1993.

Yamazaki, Y., Cheung, K. F. and Kowalik, Z.: Depth-integrated, non-hydrostatic model with grid nesting for tsunami generation, propagation, and run-up, International Journal for Numerical Methods in Fluids, 67(12), 2081–2107, 2011.

Yeh, H., Liu, P., Briggs, M. and Synolakis, C.: Propagation and amplification of tsunamis at coastal boundaries, Nature, 372, 353–355, 1994.

Yerry, M. and Shephard, M.: Automatic three-dimensional mesh generation by the modified-octree technique, J. Numer. Meth. Eng., 32(4), 709–749, 1991.

Zaytsev, A., Yalciner, A., Chernov, A., Pelinovsky, E. and Kurkin, A.: NAMI DANCE, [online] Available from: http://namidance.ce.metu.edu.tr, 2006.

Zhang, Y. and Baptista, A. M.: An efficient and robust tsunami model on unstructured grids, Pure and Applied Geophysics, 165, 2229–2248, 2008.

Zhou, J. G., Causon, M. D., Mingham, C. and Ingram, G.: The surface gradient method for the treatment of source terms in the shallow-water equations, J Comp. Physics, 168, 1–52, 2001.