2021-03-15

Metrics for Performance Quantification of Adaptive Mesh Refinement

Nicole Beisiegel
Technological University Dublin, nicole.beisiegel@tudublin.ie

Cristóbal E. Castro
Departamento de Ingeniería Mecánica, Universidad de Tarapacá, Arica, Chile

Jörn Behrens
Department of Mathematics, Universität Hamburg, Hamburg, Germany

Recommended Citation

Beisiegel, N., Castro, C.E. & Behrens, J. Metrics for Performance Quantification of Adaptive Mesh Refinement. J Sci Comput 87, 36 (2021). https://doi.org/10.1007/s10915-021-01423-0

Metrics for Performance Quantification of Adaptive Mesh Refinement

N. Beisiegel*[1], C. E. Castro[3], and J. Behrens[2,4]

[1]School of Mathematical Sciences, Technological University Dublin, Dublin, Ireland
[2]Department of Mathematics, Universität Hamburg, Hamburg, Germany
[3]Departamento de Ingeniería Mecánica, Universidad de Tarapacá, Arica, Chile
[4]Center for Earth System Research and Sustainability, Universität Hamburg, Hamburg, Germany

Non-uniform, dynamically adaptive meshes are a useful tool for reducing computational complexities for geophysical simulations that exhibit strongly localised features such as is the case for tsunami, hurricane or typhoon prediction. Using the example of a shallow water solver, this study explores a set of metrics as a tool to distinguish the performance of numerical methods using adaptively refined versus uniform meshes independent of computational architecture or implementation. These metrics allow us to quantify how a numerical simulation benefits from the use of adaptive mesh refinement. The type of meshes we are focusing on are adaptive triangular meshes that are non-uniform and structured. Refinement is controlled by physics-based indicators that capture relevant physical processes and determine the areas of mesh refinement and coarsening. The proposed performance metrics take into account a number of characteristics of numerical simulations such as numerical errors, spatial resolution, as well as computing time. Using a number of test cases we demonstrate that correlating different quantities offers insight into computational overhead, the distribution of numerical error across various mesh resolutions as well as the evolution of numerical error and run-time per degree of freedom.

Keywords: adaptive mesh, refinement criteria, discontinuous Galerkin, computational efficiency, metrics

1 Introduction

A large part of today's research depends on computer simulations that are used to predict or simulate experiments that would be too difficult, costly, dangerous or, straight out, impossible to carry out in the real world. The complexity of these experiments requires solving mathematical equations to a very high level of detail. Increasing the level of detail in uniform mesh simulations, however, requires exponentiation in the number of unknowns and thus computational resources, which ultimately limits the number and type of experiments that are practically feasible.

For over two decades element-based models have been studied e.g. to solve equations describing geophysical phenomena. These models include continuous and discontinuous finite elements, see for example Brenner and Scott (2000); Hesthaven and Warburton (2008) as well as finite volumes, see for example Castro et al. (2012); LeVeque (2002) . They are based on polygonal cells in which quantities of interest are approximated. For flooding and drying, for example, these quantities could be the water height D and the horizontal momentum $D\mathbf{u} = (Du_1, Du_2)^\top$. Detailed simulations require large numbers of (smaller) cells, which add to the computational expense. As described in Behrens (2006) this does not mean that smaller scale processes are necessarily resolved. The quality of the resolution of underlying physics heavily depends on the numerical model that is used.

One approach to reduce computational cost is that of non-uniform resolution. Static, locally refined meshes have been successfully used in applications involving advection and storm surge (see Giraldo (2000); Luettich and Westerink (1995)) and have been continuously improved. In contrast, adaptive meshes are dynamic and can capture evolving features as presented in Behrens (2006). Especially suited for the simulation of spatially localised and moving phenomena, these non-uniform meshes can take the form of nested meshes as in Mandli and Dawson (2014) for the simulation of hurricane storm surge, or of

*nicole.beisiegel@tudublin.ie

unstructured meshes, see for example Piggott et al. (2008). The general goal of adaptive mesh refinement (AMR) is to reduce the number of degrees of freedom, while locally retaining high resolution in relevant areas. Methods for AMR, however, cause additional computational overhead for grid management and manipulation.

In this paper we aim to fill a gap in performance description and assessment of AMR methods. While parallel or computational performance in terms of run-time, efficiency or scalability is a well-introduced and understood concept, the performance behaviour for adaptively dynamically changing problem sizes is more difficult to asses. Furthermore, we contribute to the understanding of the performance gain from AMR compared to uniform meshes, considering the overhead of dynamic mesh manipulations. Using the set of proposed metrics, we observe an improvement of the asymptotic behaviour of AMR methods compared to uniform refinement, which we strive to assess quantitatively. This complements the understanding of AMR methods in terms of asymptotic error minimisation, mesh optimality, or numerical approximation properties of the underlying meshes. Mesh optimization is nicely covered in Alauzet and Loseille (2016); Castro-Díaz et al. (1997), whereas mesh quality metrics can be found in Gu et al. (2001), in which a focus is on the derivation of a posteriori estimates for simulation accuracy.

The performance of an AMR method depends on refinement indicators η_{τ_i} on elements τ_i that determine the areas to be refined. These will require insight into the physical problem, are closely related to model sensitivities, and are needed for automated mesh manipulation as we will describe in more detail in subsection 3.1. Automatic mesh refinement has been shown to improve simulation accuracy Dietachmayer and Droegemeier (1992) for idealised meteorological applications even when disjoint areas of high resolution are used. Furthermore it was shown in Iselin et al. (2002) that for advection problems, they can reduce diffusive and phase errors. Three different types of refinement indicators have been discussed in the literature: (a) heuristic/ physics-based error indicators, see for example Müller et al. (2013); (b) adjoint-based refinement indicators as in Beckers et al. (2019); Farrell et al. (2013), and (c) error-based indicators in the presence of analytical solutions. It is not in the scope of this work to evaluate the best refinement criterion. The performance of AMR methods is influenced by the quality of refinement, but does not depend on an optimal error estimator.

The remainder of this manuscript is organised as follows. In section 2 we define a set of metrics for performance measurement of AMR methods, section 3 then describes the numerical 2D shallow water model. We will furthermore consider AMR using structured adaptive triangular meshes as described in Behrens et al. (2005) and discuss the used refinement strategy in more detail. In section 4 we then apply the performance metrics to a number of test cases. Subsection 4.1 discusses a localised travelling vortex with an almost constant relative local area that is refined until the smallest mesh width. This allows to discuss relative error distribution on the mesh levels and how AMR can influence the relationship of resolution to degrees of freedom depending on mesh parameters. In subsection 4.2 we then study the mesh metrics for a test case that involves wetting and drying and a very sensitive refinement indicator. In this case the relative area of refinement changes during evolution, hence making it more challenging for the automated AMR. In section 4.3 we apply the metrics to a centred impulse at time $t = 0$ which evolves and produces waves throughout the entire domain. Those are captured by the adaptive mesh and we can use the metrics to quantify an increase in computation efficiency for the entire simulation. Section 4.4 applies the metrics to a simulation of the 1993 Okushiri Tsunami. Overall, we find that the use of the adaptive mesh improves computational efficiency. Finally Section 5 summarises the findings of the paper and discusses how the presented metrics will be of use for numerical modellers who seek to assess the computational efficiency of AMR-based models.

2 Metrics of Adaptivity

In order to set the scope for the performance assessment of adaptive mesh refinement (AMR) methods, we introduce a number of evaluation criteria – *metrics* – especially selected for AMR methods. Note, we use the term *metric* to mean a quantitative measure of certain aspects of AMR performance. This is in contrast to *mesh metrics* as used in e.g. Alauzet and Loseille (2016); Castro-Díaz et al. (1997) as a quality measure of numerical properties. While in principle all accuracy metrics applied to uniform grid methods are available for AMR methods, our focus is on performance evaluation, with each metric introducing a different viewpoint of performance measurement. This approach is independent of computational implementation and ensures a fair comparison as the same method implemented in two different ways can expose very different performance characteristics.

Performance of AMR methods can be assessed based on criteria such as accuracy, resolution, or computational requirements such as memory or run time. The latter allows to further asses the overhead imposed by the adaptive mesh capability. In the remainder of this section, we will introduce three groups of metrics and relate them to common evaluation hypotheses.

2.1 Metrics focusing on accuracy

Accuracy or numerical error, can be measured in appropriate norms. Commonly the ℓ^1, ℓ^2 and ℓ^∞ error which we denote with e_1, e_2, and e_∞ respectively is used, defined as follows:

$$\begin{aligned} e_1 &:= \|u - u_h\|_1 = \int_\Omega |u - u_h| \; d\mathbf{x}, \qquad e_2^2 := \|u - u_h\|_2^2 = \int_\Omega (u - u_h)^2 \; d\mathbf{x}, \\ e_\infty &:= \|u - u_h\|_\infty = \max_{\mathbf{x}\in\Omega} |u(\mathbf{x}) - u_h(\mathbf{x})|, \end{aligned} \tag{1}$$

where u and u_h are the exact/reference and computed solution respectively, and Ω represents the computational domain. Exact solutions presume that an analytical solution exists. In case of unknown analytical solutions, convergence behaviour can be used as a replacement, assuming that the algorithm is correct and converges to the exact solution. In that case a (costly) high resolution uniformly computed solution can serve as a reference. In any case, u needs to be projected appropriately to the discrete space to be able to compare it with u_h.

For problems that require a certain accuracy and the minimisation of the cost for achieving it, one might be interested in how many degrees of freedom (DOF) - which we denote with n_Ω, a given algorithm needs for a fixed error. Inversely, this corresponds to the error a methods generates for a fixed n_Ω. Since in most cases the number of DOF proportionately relates to the computational cost, these considerations are closely related to measuring the run time or memory requirements with respect to the error. In summary, we consider ratios of certain values:

- The ratio of error over DOF: $r_{\mathrm{d_{fixed}}} = \frac{e_\rho}{n_\Omega}$ for $\rho \in \{1, 2, \infty\}$,
- The ratio of time over error $r_{\mathrm{t-to-sol}} = \frac{t}{e_\rho}$, which corresponds to a time-to-solution with given accuracy and $\rho \in \{1, 2, \infty\}$.

The errors are as defined in Eq. 1. In the naming of the metrics $\mathrm{d_{fixed}}$ stands for number of DOF fixed which stresses that numerical errors can differ for equal (or fixed) numbers of DOF depending on the location of DOFs, while $\mathrm{t-to-sol}$ stands for time to solution.

2.2 Metrics focusing on resolution

In many practical applications, for which accuracy is hard to determine, resolution, i.e. the (local) mesh size or shortest wave length resolvable, plays an important role. Since in this manuscript triangular meshes are considered, we use the measures *shortest edge length* and *radius of inscribed circle*. Note, these are equivalent in case of quasi-uniform meshes (i.e. meshes in which the smallest angles are bounded away from zero). In both cases we will use the expression $h = \min(h_\tau)$, where h_τ denotes either the length of edge τ or the radius of the inscribed circle in cell τ. The minimum is taken globally over all edges or cells, respectively. Note that in higher order methods, h_τ needs to consider the distance between DOF and not the shortest edge length of cells, since waves can be resolved within cells.

In Gu et al. (2001), an attempt was made to quantify how well the simulated flow was resolved. Their approach focused on grid quality measures - a different view point from what we will investigate as we are concerned with how many DOF are needed in order to resolve certain features. This leads to another question of, given a certain number of DOF (limited e.g. by the given computing infrastructure), how fine can the (local) resolution be and therefore the size of the physical model. An interesting question for method inter-comparison is what error can be achieved by which resolution; a question that can only be answered if a converging reference solution is known. Then we can define the following metrics, again as ratios:

- The ratio of resolution over error $r_{\mathrm{eff-res}} = \frac{h}{e_\rho}$ for $\rho \in \{1, 2, \infty\}$,
- The ratio of resolution over DOF $r_{\mathrm{res}} = \frac{h}{n_\Omega}$.

2.3 Metrics focusing on computational resources

Trying to assess computational demands is a highly challenging task, since inefficient implementations can hamper an objective assessment of such characteristics for different methods. Quantifying those computational demands, however, is vital in practical applications as solution accuracy is constrained by minimising computational effort. Naturally, the above mentioned metric $r_{\mathrm{t-to-sol}}$ is relevant here as is the time an implementation needs per DOF. By this, an efficient implementation can be characterised and computational overheads quantified which are often only assessable by rigorous profiling of a running program. It is also interesting to look at this aspect for changing problem sizes. Optimally, the ratio should keep constant for increasing problem sizes, but often cache effects or non-optimal algorithmic complexity inhibits this. Interesting questions arise also from memory requirements. The amount of memory necessary to achieve a certain error or resolution is an important measure for AMR methods. This may include memory used for handling DOFs or mesh as well as other implementational aspects. The following ratios may be helpful to assess computational efficiency of adaptive algorithms:

- The ratio of time over error $r_{\mathrm{t-to-sol}}$ (see above),
- The ratio of time over DOF $r_{\mathrm{t-per-DOF}} = \frac{t}{n_\Omega}$,
- The ratio of memory required for resolution $r_{\mathrm{mem-res}} = \frac{MB}{h}$, where MB is the memory in bytes.

2.4 Further remarks

An important prerequisite for a rigorous assessment of AMR methods is an accurate and robust error estimator or refinement criterion. An adaptation strategy that selects a wrong (or insufficient) area for better resolution can corrupt comparisons and render the error-based metrics presented in subsections 2.1 and 2.2 useless as in this case the error will always be high due to the refinement strategy failing to capture relevant features. It is therefore paramount to compare a uniform grid solution to an equally resolved adaptive solution first and to make sure the difference with respect to solution error is not dominating. A description of our refinement strategy can be found in Section 3.1.

In the test cases presented in Section 4 we have access to some useful error norms and our error indicators refine relevant areas. However, in many application fields the numerical error might not be the most relevant quality measure or might not be accessible at all. When considering multi-physics applications the source of the error might even be unclear. Hence, in the assessment of performance one may as well replace the numerical error by some other adequate norm for solution quality. As an example, in geoscientific applications, the resolvable wave length or frequency may be more relevant than the obscure mathematical error.

3 The Numerical Model and Adaptive Mesh Refinement

As an example numerical model for this study we chose a discontinuous Galerkin (DG) model that solves the depth-integrated shallow water equations in two dimensions which can be written in flux form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \qquad \text{in } \Omega \times T,$$

where the prognostic variables are $\mathbf{U} = (D, D\mathbf{u})^\top$: the water depth D and the 2D momentum $D\mathbf{u}$ defined on a spatial domain of interest $\Omega \subset \mathbb{R}^2$. Spatial coordinates are denoted as $\mathbf{x} = (x, y)^\top \in \Omega$. $\frac{\partial \mathbf{U}}{\partial t} =: \mathbf{U}_t$ denotes the temporal derivative and $\nabla \cdot = \frac{\partial}{\partial x} \cdot + \frac{\partial}{\partial y} \cdot$ is the divergence operator. $\mathbf{F}$ is a flux function and S a source term defined as

$$\mathbf{F}(\mathbf{U}) = \begin{bmatrix} D\mathbf{u} \\ D\mathbf{u} \otimes \mathbf{u} + \frac{g}{2} D^2 \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{S}(\mathbf{U}) = -\begin{bmatrix} 0 \\ gD\nabla b \end{bmatrix}$$

where $g = 9.81\mathrm{m\ s}^{-2}$ is the acceleration due to gravity, $\otimes$ a vector product in $\mathbb{R}^2$, and $\mathbf{I}_2$ is the 2×2 identity matrix. For the reader familiar with shallow water models, we note that we chose to denote the water height with D instead of the more commonly used h, which in this manuscript, we have reserved for the resolution as we think it improves readability. The source term models a temporally constant bathymetry $b = b(\mathbf{x})$. Throughout this paper vector valued quantities are indicated by a bold print while all other quantities are assumed to be scalar.

The discretisation that we are using is a discontinuous Galerkin (DG) discretisation obtained through three steps: (a) decomposing the computational domain $\Omega = \sum_i \tau_i$ into conforming triangles with i the index of the triangles; (b) approximating the prognostic variables $\mathbf{U} = \sum_k \mathbf{U}_k(t)\phi_k(\mathbf{x})$ by linear Lagrange polynomials with k the corresponding index for the basis functions; and (c) integrating the resulting equations in space against test functions. In the present model the test functions coincide with the basis functions ϕ_k used for the approximation, so that the resulting semi-discrete system reads

$$\int_{\tau_i} \mathbf{U}_t \phi_j \mathrm{d}\mathbf{x} + \int_{\tau_i} \nabla \cdot \mathbf{F}(\mathbf{U})\ \phi_j \mathrm{d}\mathbf{x} + \int_{\partial\tau_i} (\mathbf{F}^*(\mathbf{U}) - \mathbf{F}(\mathbf{U})) \cdot \mathbf{n}\ \phi_j \mathrm{d}S = \int_{\tau_i} \mathbf{S}(\mathbf{U})\phi_j \mathrm{d}\mathbf{x}, \tag{2}$$

for all triangles τ_i and $j = 1, 2, 3$ leading to a second order accurate spatial discretisation. Here, $\mathbf{F}^*$ is a numerical approximation of the flux at the cell interfaces. In our model, we used Rusanov's flux (see for example Toro (2009)). Note that we integrated the flux term by parts twice to obtain the *strong* form of the equations as in Hesthaven and Warburton (2008). In this form, we integrate the jump over the edges which has desirable properties with respect to wellbalancing as shown in Beisiegel (2014) and more recently in Vater et al. (2019). After numerical integration and re-organisation, the system (2) can be written as

$$\frac{d\mathbf{U}_k}{dt} = \mathbf{H}(\mathbf{U}_k), \tag{3}$$

where $\mathbf{H}$ denotes the discretised version of the right hand side which includes the fluxes as well as the source terms. As this model is used for flood simulations, slope limiting is required to prevent spurious velocities at wet/dry interfaces. The slope limiter we are using is velocity-based and described in detail in Vater et al. (2019). In the latter study we show that it can be successfully applied to tsunami benchmarks and to model flood scenarios. It is almost parameter-free and robust when applied to unstructured meshes. The system (3) can be solved using a strong stability preserving (SSP) multi-stage Runge Kutta method. We used Heun's method (RK22) for this study.

3.1 Computational Efficiency and Adaptive Meshes

The main focus of this study is on performance characteristics of the underlying adaptive mesh independent of the numerical discretisation. The model described above and in more detail in Beisiegel et al. (2019) uses the grid generator `amatos` (see Behrens et al. (2005)) to create dynamically adaptive and conforming triangular meshes. Finer triangles are obtained by bisecting coarse triangles along a marked (longest) edge as suggested in Rivara (1984). The dynamic mesh adaptation process involves problem-dependent refinement indicators η_{τ_i}, such as the absolute value of the gradient of fluid height at time t:

$$\eta_{\tau_i}(t) = \max_{\mathbf{x}\in\tau_i} \|\nabla D(\mathbf{x}, t)\|_\infty$$

for each element τ_i as in Behrens (2006), to control the element-wise refinement and coarsening. Using problem dependent and user-defined tolerances $0 \le \theta_{crs} < \theta_{ref} \le 1$ the mesh manipulation is then carried out as follows:

$$\begin{array}{lll} \text{if} & \eta_{\tau_i} \le \theta_{crs}\ \eta_{max} & \rightarrow \text{coarsen element } \tau_i \\ \text{if} & \eta_{\tau_i} \ge \theta_{ref}\ \eta_{max} & \rightarrow \text{refine element } \tau_i, \end{array} \tag{4}$$

with $\eta_{max} = \eta_{max}(t) := \max_{\tau_i\subset\Omega} \eta_{\tau_i}(t)$ the maximum value of the refinement indicator over all elements. The values $0 \le \theta_{crs} < \theta_{ref} \le 1$ mark the fraction of the maximum error below/above which an element is considered for coarsening/refinement. After this mesh manipulation step, the computation of the dynamics are then repeated on the new mesh. The size of the smallest and largest element of the mesh is determined through mesh levels $\lambda_{crs}, \lambda_{ref} \in \mathbb{N}$ with $\lambda_{crs} \le \lambda_{ref}$. Starting from an initial (coarse) triangular mesh, the mesh generator will uniformly refine λ_{crs} times to establish the coarsest mesh level and then, using the refinement indicator, refine areas of interest until the desired finest mesh level is reached using the tolerances as described in (4).

Mesh modification requires modification of nodal values. New mesh nodes are interpolated using the known Lagrange basis functions ϕ_k in each element to determine values of prognostic variables. When source terms are interpolated this might not give the desired accuracy. Hence, if data or parameterisations are available, source terms are updated by reloading data onto the new mesh nodes. To retain well-balancing, prognostic variables are then modified accordingly. After coarsening, nodal values along coarsened edges need to be updated to achieve mass conservation. For linear Lagrange functions this can be achieved by replacing them with interpolated mean values of the old (finer) edges. Higher order generalisations of this procedure are beyond the focus of this study.

The `amatos` mesh, created as described above, has the further advantage of using a cache-efficient space-filling curve-ordering of elements (see Behrens and Bader (2009)), which allows fast access of neighbouring elements: A feature that is particularly beneficial for localised numerical methods such as the presented discontinuous Galerkin since elements only communicate over edges. However, it comes at the cost of not allowing for unstructured meshes.

For convenience, the meshes are kept conforming, i.e. free of hanging nodes, throughout the simulation. We stress that this is not required by the method itself. Hanging nodes would require to combine two or more Riemann solutions over one (coarse) edge as is for example recently done in Kopera and Giraldo (2014) and Hermann et al. (2011).

4 Application of Metrics and Numerical Tests

The mesh metrics defined in Sec. 2 are a computational tool for quantifying simulation performance. Applying them to four dynamically different test cases, we show that and how the metrics provide inside into the efficiency and performance gain of AMR methods. In the following, we consider

- a quasi-stationary travelling vortex in Subsec. 4.1, with a constant high resolution area,
- long wave resonance in Subsec. 4.2, that comprises wetting and drying and changing size of high resolution areas,
- a centred impulse in Subsec. 4.3 that only initially exhibits a localised feature,
- a simulation of a more realistic testcase, the 1993 Okushiri tsunami, in Subsec. 4.4.

4.1 A Strongly Localised Feature: A Quasi-Stationary Vortex

In a domain $\Omega = [0,4] \times [0,2]$ with a flat bathymetry $b \equiv 0$, periodic boundaries at $\{\mathbf{x}|x = 0 \vee x = 4\}$ and reflecting boundaries at $\{\mathbf{x}|y = 0 \vee y = 2\}$, a vortex around $\mathbf{c} = (1,1)^\top$ is defined with tangential velocity $v_\vartheta(r)$

$$v_\vartheta(r) = \begin{cases} v_{\max} \dfrac{s \cdot r}{r_m^2 - r^2} \cdot \sqrt{2 \exp\left(\dfrac{1}{r^2 - r_m^2}\right)} & \text{for } 0 \le r < r_m \\ 0 & \text{otherwise} \end{cases}$$

where $r = \|\mathbf{x} - \mathbf{c}\|_2$ is the radial distance from $\mathbf{c}$, the parameters $v_{\max} = 0.5$ and $r_m = 0.45$ are maximum tangential velocity and the vortex radius respectively and the scaling factor s is defined as

$$s = \frac{|r_{vm}^2 - r_m^2|}{r_{vm}\sqrt{2\exp(1/(r_{vm}^2 - r_m^2))}}, \quad \text{where} \quad r_{vm} = \frac{1}{2}\sqrt{-2 + 2\sqrt{1 + 4r_m^4}}$$

is the radius of maximum winds. With a background height, $D_{\text{bg}} = 1$, and a background velocity, $\mathbf{u}_{bg} = (u_{\text{bg}}, v_{\text{bg}})^\top = (1,0)^\top$, the initial conditions are

$$D(\mathbf{x},0) = \begin{cases} D_{\text{bg}} - \dfrac{v_{\max}^2 s^2}{g} \exp\left(\dfrac{1}{r^2 - r_m^2}\right) & \text{for } 0 \le r < r_m \\ D_{\text{bg}} & \text{otherwise} \end{cases}$$

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_{\text{bg}} - v_\vartheta(r)(\sin\vartheta, \cos\vartheta)^\top, \quad \text{with } \vartheta = \arctan\left((y-2)/(x-1)\right),$$

where x and y are the spatial coordinates $\mathbf{x} = (x,y)^\top$. The final time of the simulation is $T_{end} = 4$s, i.e. until the vortex reaches its starting position again. The analytical solution for this test can be found in Vater (2013).

4.1.1 Discussion of Simulation Results

We ran a number of uniform and adaptive simulations with $6 \le \lambda_{ref} \le 14$ and a time step of $\Delta t = 0.1$ s for the finest resolution $h = 3.2 \cdot 10^{-3}$m (equivalent to $\lambda_{ref} = 14$) and larger time steps for coarser resolutions

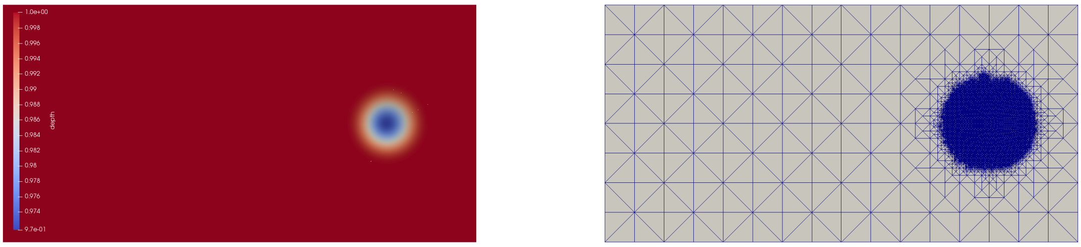

Figure 1: Quasi-stationary vortex: Simulated water depth at time $t = 1$s (left) and corresponding adaptive mesh (right).

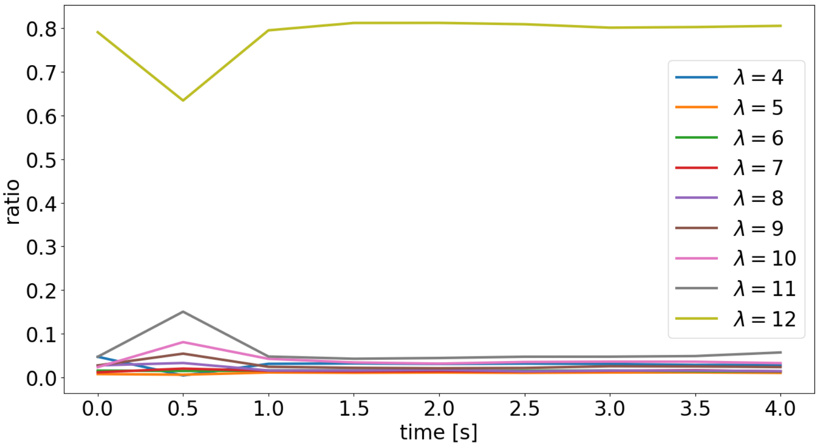

Figure 2: Quasi-stationary vortex: Percentage of elements on mesh levels over time for an adaptive simulation with $\lambda_{ref} = 12, \lambda_{crs} = 4$, and parameters $\theta_{ref} = 0.2$, and $\theta_{crs} = 0.1$.

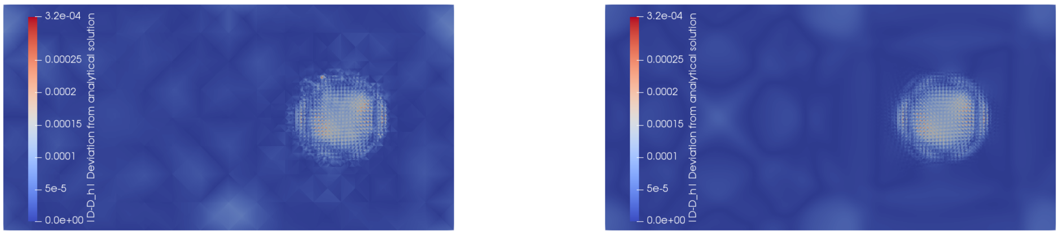

Figure 3: Quasi-stationary vortex: Difference between numerical and analytical solution of adaptive (left) and uniform (right) simulation at time $t = 1$s.

$\lambda_{ref} =$	6	8	9	10	11	12
$\lambda = 4$	9.6					
$\lambda = 5$	8.0	34.1				
$\lambda = 6$	15.1	6.7	43.3			
$\lambda = 7$	67.4	8.1	4.7	51.9		
$\lambda = 8$		51.2	6.6	3.2	58.6	
$\lambda = 9$			45.4	4.1	1.8	61.6
$\lambda = 10$				40.9	3.1	1.8
$\lambda = 11$					36.5	2.6
$\lambda = 12$						34.0

$\lambda_{ref} =$	6	8	9	10	11	12
$\lambda = 4$	6.4					
$\lambda = 5$	7.7	15.5				
$\lambda = 6$	8.8	6.3	18.6			
$\lambda = 7$	77.1	5.8	6.7	22.6		
$\lambda = 8$		72.3	6.8	9.1	25.4	
$\lambda = 9$			68.0	9.4	8.3	28.6
$\lambda = 10$				59.0	9.5	11.7
$\lambda = 11$					56.9	13.3
$\lambda = 12$						46.5

Table 1: Quasi-stationary vortex: Percentage of number of elements (left) and numerical error (right) on different mesh levels λ for adaptive simulations using strategy A with $\lambda_{crs} - \lambda_{ref} = 3$ and $\lambda_{crs} = 4$, λ_{ref} as stated and parameters $\theta_{crs} = 0.1, \theta_{ref} = 0.2$.

$\lambda_{ref} =$	6	8	9	10	11	12
$\lambda = 4$	3.5	15.6	11.0	8.1	5.1	2.8
$\lambda = 5$	1.6	5.6	3.6	2.7	1.6	0.9
$\lambda = 6$	94.9	7.9	5.6	3.6	2.3	1.3
$\lambda = 7$		9.5	6.1	3.8	2.3	1.4
$\lambda = 8$		61.3	8.5	5.1	2.8	1.7
$\lambda = 9$			65.2	7.0	4.1	2.4
$\lambda = 10$				69.6	6.6	3.9
$\lambda = 11$					72.5	5.6
$\lambda = 12$						79.9

$\lambda_{ref} =$	6	8	9	10	11	12
$\lambda = 4$	0.1	11.9	11.1	11.9	10.8	9.2
$\lambda = 5$	1.6	5.9	5.4	6.9	5.3	5.9
$\lambda = 6$	98.6	6.3	6.0	7.5	5.7	6.9
$\lambda = 7$		6.1	5.9	7.1	5.5	6.3
$\lambda = 8$		69.8	6.4	7.7	5.7	6.6
$\lambda = 9$			65.3	8.4	6.9	7.2
$\lambda = 10$				50.5	8.1	8.5
$\lambda = 11$					51.9	9.5
$\lambda = 12$						39.7

Table 2: Quasi-stationary vortex: Percentage of number of elements (left) and numerical error (right) on different mesh levels λ for adaptive simulations using strategy B and $\theta_{crs} = 0.1, \theta_{ref} = 0.2$.

such that the CFL stability condition was fulfilled. A decrease of λ_{ref} by 2 allows an increase of Δt by a factor of 2. The vortex is captured by the adaptive mesh refinement using $\eta_{\tau_i} = \max_{\mathbf{x} \in \tau_i} \|\nabla D(\mathbf{x}, t)\|_\infty$ with mesh parameters $\theta_{crs} = 0.1, \theta_{ref} = 0.2$. We studied two systematic ways to select λ_{crs} for a given and increasing λ_{ref}:

(A) A constant mesh level difference $d_\lambda := \lambda_{ref} - \lambda_{crs} = 3$. Higher resolved simulations were obtained by increasing λ_{crs} and λ_{ref} simultaneously.

(B) A fixed coarse mesh level $\lambda_{crs} = 4$. Higher resolved simulations were obtained by increasing λ_{ref};

The numerical water depth at $t = 1$s with a corresponding adaptive mesh is depicted in Figure 1. We observe that after an initial calibration period, the mesh follows the vortex and refines only the area of interest. The calibration period is needed because the initial conditions are not exactly balanced on a discrete level, so that gravity waves are emitted which vanish as soon as the model reaches a balanced state. Over time, the percentage of elements on a given mesh level do not show a lot of fluctuation as shown in Figure 2. Since the vortex is not changing in size over time, this is an indication that it is well captured by the adaptive mesh and internal iteration to flag elements for refinement and coarsening are kept minimal which adds to the computational efficiency of the mesh manipulation.

4.1.2 Performance Metrics Evaluation

The proposed mesh metrics have been plotted in Figure 4. The solid line corresponds results obtained on a uniform mesh while the two non-solid lines correspond to adaptively refined meshes: The dashed line represents strategy (A) and the dashed-dotted line represents strategy (B). The adaptive simulations both used the same refinement indicator η_{τ_i} as stated above.

We observe that both sets of adaptive simulations, using strategy (A) and (B), overall, give better results than the set of uniform simulations. In particular, the top left panel shows $r_{\mathrm{d}_{\mathrm{fixed}}}$ for all three sets of simulations and indicates an accelerated convergence with respect to the number of degrees of freedom (DOF) n_Ω. Using n_Ω instead of the traditional mesh width h improves the comparability of adaptive and uniform simulations. Note furthermore that both definitions of convergence lead to equivalent results for

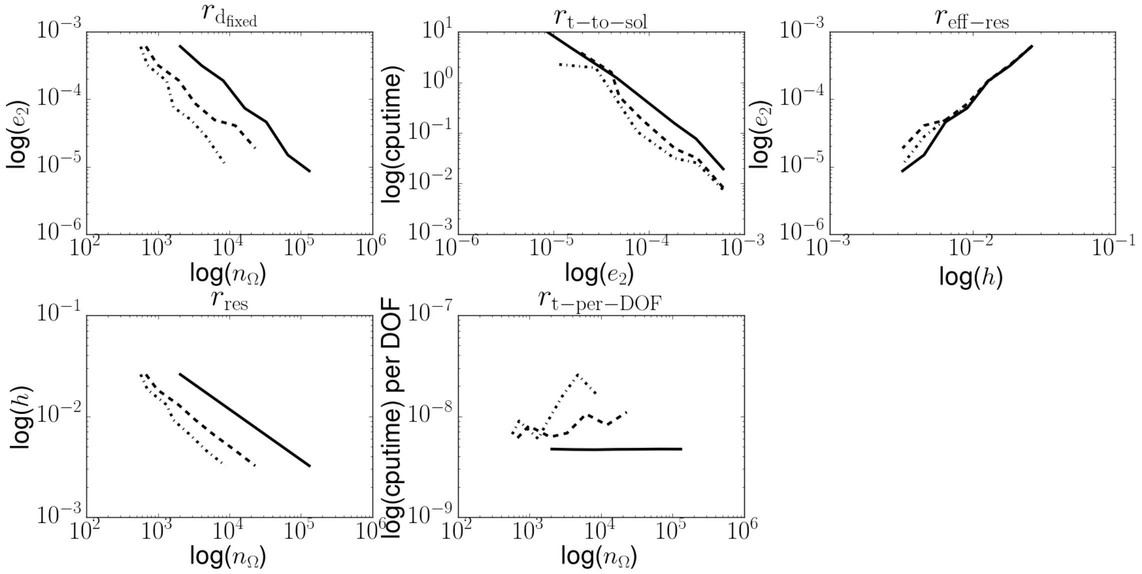

Figure 4: Quasi-stationary vortex: Plots of metrics $r_{\mathrm{d_{fixed}}}, r_{\mathrm{t-to-sol}}, r_{\mathrm{eff-res}}, r_{\mathrm{res}}$, and $r_{\mathrm{t-per-DOF}}$ from top left to bottom right for uniform simulations (solid line), dynamically adaptive simulations following strategy A (dashed line), and dynamically adaptive simulations using strategy B (dashed dotted line).

uniform simulations. Using linear least squares regression analysis, we find that the mean slope of the solid line for $r_{\mathrm{d_{fixed}}}$ is 1.04 while the mean slope of the dashed dotted is 1.41 - an increase of over 40%.

An increase in computational efficiency can be deduced from the top middle panel of Figure 4 that plots the metric $r_{\mathrm{t-to-sol}}$ and visualises that using AMR, we consistently achieve the same L_2-error with less CPU time. Per degree of freedom, however, AMR is computationally more expensive as the metric $r_{\mathrm{t-per-DOF}}$ (bottom middle panel in Figure 4) shows. There, we show that for a given number of DOFs n_Ω, the uniform simulation (solid line) requires less CPU time per DOF because of the computational overhead caused by mesh manipulation and management. This overhead is almost constant (the plotted lines are merely shifted by a constant) and approximately does not increase with increasing n_Ω. For a decreasing mesh width h, i.e. increasing λ_{ref}, AMR yields a higher effective resolution as is shown by $r_{\mathrm{eff-res}}$ in the top right panel of Fig. 4. Hence, given a fixed number of DOF, n_Ω, AMR resolves finer features if the underlying equations model them. As an example Behrens (2006) mentions that even fine resolution fails to resolve fine scale features if model equations don't model them. Finally metric r_{res} shows how the local resolution changes by increasing the number of DOF (bottom left panel of Figure 4). Since this depends on the systematic ways in which λ_{crs} and λ_{ref} are chosen, we elaborate on this in the following subsection.

4.1.3 A Note on Refinement Strategies

Considering mainly r_{res} allows us to compare the refinement strategies (A) and (B) more rigorously. As opposed to uniform mesh methods AMR methods comprise elements on a number of mesh levels between λ_{crs} and λ_{ref}. The total number of elements is an indicator for the number of DOF and the computational work load. The exact number of unknowns can be determined by Euler's formula for plane graphs that relates the number of vertices, edges and cells by $\#\text{vertices} - \#\text{edges} + \#\text{cells} = 2$, considering the location of unknowns. For the method in Sec. 3, n_Ω is directly related to the number of cells. The refinement strategy implemented by the grid generator `amatos` bisects elements, thus a refinement step generates two elements out of the elements marked for refinement. This allows us to study the effect of the different refinement strategies.

Uniform refinement will increase the mesh level by one (i.e. $\lambda_{ref} \to \lambda_{ref} + 1$, note that for uniform meshes $\lambda_{crs} = \lambda_{ref}$) and will double the number of elements (i.e. $n_\Omega \to 2 \cdot n_\Omega$). Refinement strategy (A) will increase both, λ_{crs} and λ_{ref} by one, but will leave d_λ unchanged. This means all elements of levels λ_{crs} through λ_{ref} will be refined, which is all elements altogether. Therefore, $n_\Omega \to 2 \cdot n_\Omega$ as in the uniform case. This is confirmed by the metric r_{res} as it shows the dashed line (adaptive strategy A) being parallel to the solid line (uniform simulation) - the number of DOF n_Ω evolve in the same way. In contrast to this, refinement strategy (B) will only increase λ_{ref}, in other words $\lambda_{crs} \to \lambda_{crs}$, $\lambda_{ref} \to \lambda_{ref} + 1$, and $d_\lambda \to d_\lambda + 1$. Note that only those elements found by the refinement indicator will be refined. Most likely, these will be only those that are already on the (previous) finest level. Therefore, the additional number of elements depends on the fraction of the domain found relevant by the error indicator. Let us assume that we start with a uniform mesh in which about 10% of the elements are flagged for refinement. Then only those 10% elements are refined which means the number of unknowns is doubled only for those 10% (i.e. $n_\Omega \to n_\Omega + 0.1 \cdot n_\Omega = 1.1 \cdot n_\Omega$). Using this formula recursively for increasing numbers of refinement levels shows how dramatic the reduction of complexity can be.

To conclude our investigation of the refinement strategies, we consider the spatial distribution of numerical error as capturing it within the fine resolution area minimises it. Overall, it is expected that AMR leads to a more uniform distribution of the numerical error compared to uniform meshes if the refinement indicator sufficiently captures model sensitivities. We show examples of the distribution of the numerical error in Figure 3 and observe that the majority of the error is captured in the high resolution area of both simulations. We note that the uniform simulation used 131072 elements as opposed to the adaptive simulation which used on average 9476 elements, or less than 10% the number of elements compared to the uniform one.

Our results using the two adaptive strategies (A) and (B) confirm the hypothesis that keeping λ_{crs} as small as possible increases efficiency. Tables 2 and 1 show the distribution (per cent) of elements on a certain mesh level (left) as well as the distribution of numerical L_2 errors per mesh level (right). The columns of the tables correspond to one simulation with mesh levels as stated, e.g. the first column in table 2 corresponds to an adaptive simulation with mesh parameters $\lambda_{crs} = 4$ and $\lambda_{ref} = 6$. Using $\theta_{crs} = 0.1, \theta_{ref} = 0.2$ for all simulations, we observe that the majority of elements resulting from refinement strategy (B), are on the finest mesh level in comparison to configuration (A), where the majority of elements are both on the finest and coarsest mesh level. We furthermore see that for configuration (B), the largest part of the error is also to be found on the finest mesh level while the error on level 4 is decreasing with increasing λ_{ref} and all intermediate mesh levels $\lambda_{crs} \leq \lambda \leq \lambda_{ref}$ only contain small errors. For configuration (A) we obtain the majority of errors on the finest mesh level as well. The error on the coarsest level, however, is increasing with increasing the overall resolution.

4.2 A Dynamically Changing Area of Maximum Refinement: Long Wave Resonance in a Paraboloid Basin

This nonlinear problem can be found in Lynett et al. (2002) in a scaled version and its original analytical solution was first determined in Thacker (1981). In a domain $\Omega = [0, 8000]^2$, a parabolic basin of shape $b(\mathbf{x}) = h_0(1 - \frac{r^2}{a^2})$ with $h_0 = 1$ the centre of the basin, $a = 2500$ the distance from the centre to the shoreline and $r = \|\mathbf{x}\|^2$ the radius, the initial water depth is described as

$$D(\mathbf{x}, 0) = h_0 \left(\frac{(1 - A^2)^{1/2}}{1 - A} - 1 - \frac{r^2}{a^2} \left(\frac{1 - A^2}{(1 - A^2)^2} - 1 \right) \right)$$
$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{0}$$

where the constants are determined as

$$A = \frac{a^4 - r_0^4}{a^4 + r_0^4} \qquad \text{with } r_0 = 2000.$$

4.2.1 Discussion of Simulation Results

We ran sets of simulations with uniform and adaptive mesh refinement with mesh parameters ranging from $8 \leq \lambda_{crs}, \lambda_{ref} \leq 17$, $\theta_{ref} = 0.05$, and $\theta_{crs} = 0.001$ with a time step of $\Delta t = 16$ s for $h = 207.11$ m which corresponds to $\lambda_{ref} = 8$ and smaller Δt for larger mesh levels in a way that the CFL condition is still fulfilled.

For the adaptive simulations, we chose the refinement strategy labelled strategy (A) in the previous subsection 4.1 with $d_\lambda = 3$ to ensure that the wet area of the domain is completely captured within the

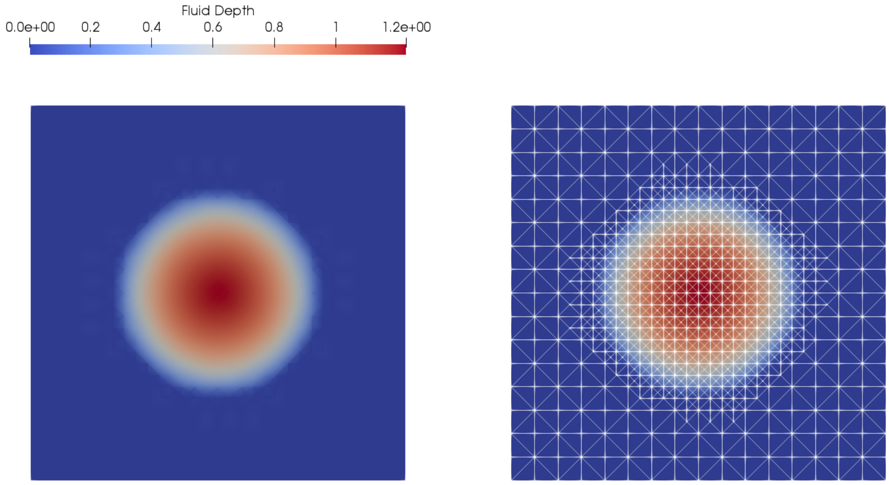

Figure 5: Long Wave Resonance in a Paraboloid Basin: Plot of fluid depth at time t=2000[s] (left) and corresponding adaptive mesh (right) with mesh parameters $\lambda_{ref} = 12, \lambda_{crs} = 9, \theta_{ref} = 0.05$, and $\theta_{crs} = 0.001$.

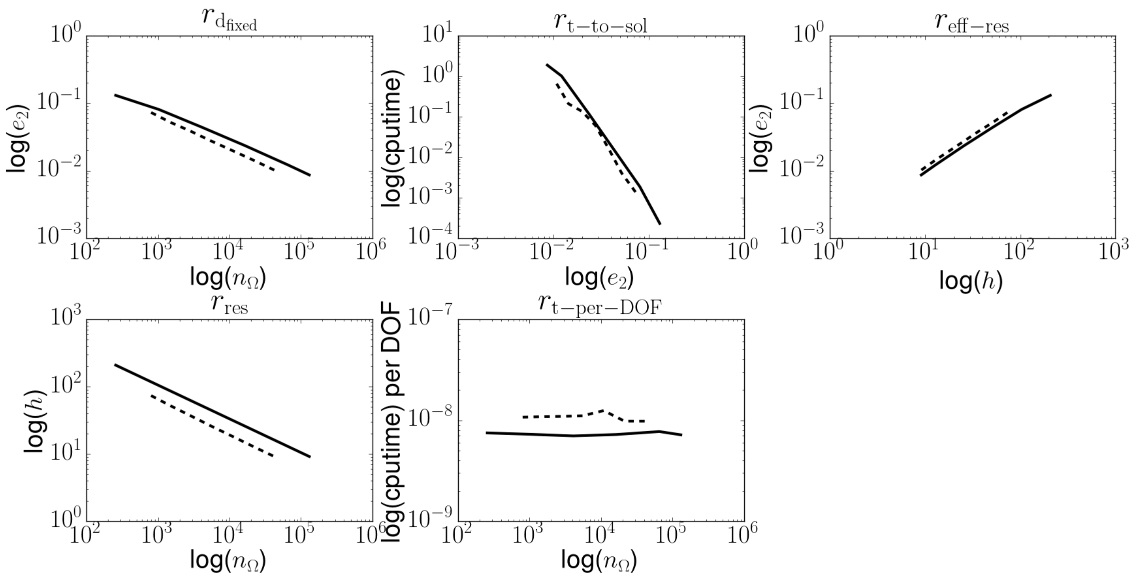

Figure 6: Long Wave Resonance in a Paraboloid Basin: lots of metrics $r_{\mathrm{d_{fixed}}}, r_{\mathrm{t-to-sol}}, r_{\mathrm{eff-res}}, r_{\mathrm{res}}$, and $r_{\mathrm{t-per-DOF}}$ from top left to bottom right for uniform simulations (solid line), and dynamically adaptive simulations (dashed line).

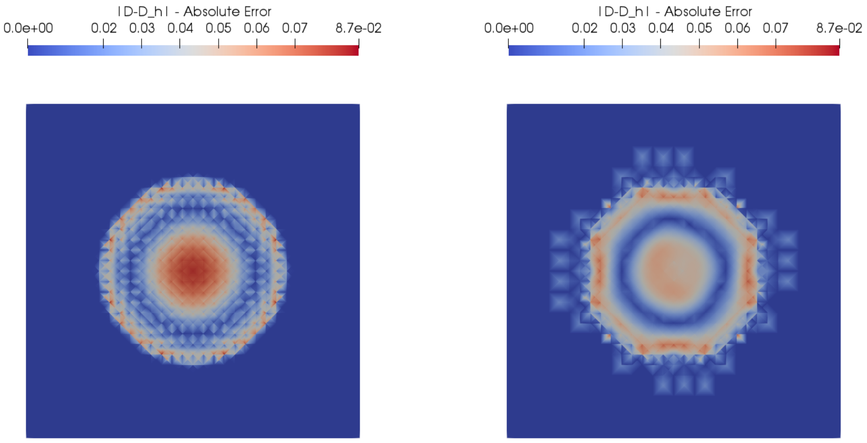

Figure 7: Long Wave Resonance in a Paraboloid Basin: Plot of absolute deviation of numerical fluid depth D from exact solution at time $t = 2000[s]$ on a uniform mesh (left) with $\lambda_{ref} = 12$ and adaptive mesh (right) with mesh parameters $\lambda_{ref} = 12, \lambda_{crs} = 9, \theta_{ref} = 0.05$, and $\theta_{crs} = 0.001$.

high resolution area of the adaptive mesh. A more detailed description of this can be found in subsection 4.2.3.

We chose $\eta_{\tau_i} = \overline{D}\Big|_{\tau_i}$, the mean value of the fluid depths over each element $\tau_i \subset \Omega$ as a refinement indicator. Present high-order information, when not controlled properly through η_{τ_i}, will propagate into coarser mesh regions and possibly cause large numerical error. This is partially due to the non-optimality of the marking strategy that we employ. As shown in Dörfler (1996) for Poisson's equation, using the L_2-norm instead of the maximum norm in the definition of η_{τ_i}, we obtain the least amount of refined triangles at the cost of a possibly larger number of iterations for the mesh manipulation. We remark that the computational model described in section 3 is capable of simulating wetting and drying using slope limiters to prevent spurious velocities close to the wet/dry interface. The fluid depth D will not be set to zero in these areas unless below a small cut off tolerance $\epsilon = 10^{-K}$ with $K \in \mathbb{N}$ which may lead to small water films remaining in dry areas - an effect that makes the wetting and drying very sensitive to adaptive mesh refinement of the wet area of the domain Ω. High-resolution information propagating into coarse mesh cells can be particularly problematic in wet/dry areas where refinement and interpolation of partially dry cells can affect model stability. For this reason, we ensure surrounding elements of partially dry elements are refined too, hence restricting mesh manipulation to wet cells only.

Fig. 5 shows an example of the simulated fluid depth D and a corresponding adaptive mesh at time $t = 2000$s. We observe that the fluid is well captured by the adaptive mesh and the dry area is kept coarse throughout the simulation, hence minimising computations in areas which do not influence model dynamics. Fig. 7 furthermore shows that the adaptive mesh captures the majority of the numerical error in the high resolution part of the mesh (right display) and that compared to the uniform simulation, the point wise error is smaller.

4.2.2 Performance Metrics Evaluation

Applying the metrics defined in section 2 to all simulations of this test case, we obtain Fig. 6. Depicted are results obtained on a uniform mesh (solid line) and on an adaptive mesh (dashed line). We observe that the adaptive simulation consistently achieves a lower numerical error for a given number of DOF n_Ω as demonstrated by the metric $r_{\mathrm{d}_{\mathrm{fixed}}}$ in the top left panel of Fig. 6. Using the modified definition of convergence based on n_Ω we see a convergence acceleration by about 10% in this case which is significantly lower than the result reported in subsection 4.1, but can be attributed to the lacking smoothness of the solution at the wet/dry interface. In line with the findings in subsection 4.1, the mesh refinement using strategy (A) makes it possible to achieve a finer spatial resolution for a given number of DOF as indicated by r_{res} in the bottom left panel of Fig. 6. This plot furthermore shows that the corresponding dashed line is a linear translate of the uniform simulation as expected from the use of strategy (A) (see also 4.1.3).

θ_{crs}
1
$\frac{1}{2}$
0
more coarsening
less coarsening
θ_{ref}
1
$\frac{1}{2}$
0
less refinement
more refinement

Figure 8: Sketch of influence of grid parameters θ_{crs} and θ_{ref} on the mesh manipulation process.

Furthermore, $r_{\text{eff}-\text{res}}$ in the top right panel of Fig. 6, shows that the effective resolution is consistently higher for a given numerical error.

The rapid speed of change of the wet/dry interface and with that the high resolution part of the mesh are challenging for this physics-based, automated type of AMR. Changing the relative size of the high resolution area requires a large number of mesh manipulations. This is in contrast to the test case in subsection 4.1 that comprised a propagating feature of constant relative size over time. Although there is a significant amount of time spent on mesh manipulation and management, the metric $r_{\text{t}-\text{to}-\text{sol}}$ (top middle display of Fig. 6) shows that the time to solution is smaller for the adaptive simulations while the time per degree of freedom (see $r_{\text{t}-\text{per}-\text{DOF}}$ in bottom middle display of Fig. 6) is slightly larger for the adaptive simulation. Confirming the findings from the previous subsection.

4.2.3 A Note on Grid Parameters

The AMR used in this study heavily relies on a good choice of the refinement indicator η_{τ_i} (see also subsection 3.1) as it drives the automated mesh manipulation process. This stresses the importance for η_{τ_i} to be a good proxy for numerical error. For example, in the presence of wetting and drying (see previous subsection 4.2), the (even only partially) wet area of the domain has to be finely resolved in order to prevent artificial waves close to the wet/dry interface as those might not be completely filtered out by the slope limiter in the sense of $D \equiv 0$, and, hence, might be artificially amplified if energy from high-order modes is propagated into lower-order modes without additional filtering and with that causing numerical error.

Moreover, the automated AMR described in subsection 3.1 and with that the simulation quality depends on four key parameters $\lambda_{crs}, \lambda_{ref}, \theta_{crs}, \theta_{ref}$. Our observations confirm that the mesh refinement is sensitive towards the choice of the λs which determine the range of spatial resolution of the simulation, or in other words the length of the waves that are resolved in the simulation, and of the θs as shown in Figure 8, which determine the fraction of the maximum error below/above which an element is considered for coarsening/refinement. In general, a higher resolution is achieved by increasing the number of DOF n_Ω which can be initiated by mesh refinement (h-refinement) as in this study or through increasing the order of the used polynomials (p-refinement). We note that it was found in Jameson (2000) that the number of points needed for a fixed numerical error is not constant with respect to the order of the numerical scheme - in our case the order of the polynomials ϕ_k - but rather declines with increasing order. In theory, d_λ can be arbitrarily large. However, sufficient physical space surrounding the feature of interest is required such that elements on all mesh levels from λ_{crs} to λ_{ref} may envelop it. This is because mesh levels continuously increase from coarse to fine and in the absence of the required space, the feature of interest either will not be refined with the desired high resolution or the coarse area will be refined more than desired, hence the maximum benefit from AMR might not be obtained. For that reason, we solely chose strategy (A) in subsection 4.2.

The choice of the θs can be particularly crucial where the refinement indicator is sensitive towards small perturbations as the one described in section 4.2. In this case an increase of the resolution (an increase of θ_{ref}) can lead to an increase of numerical error as small errors at the wet/dry are amplified when they are higher resolved by the finer mesh. This supports that the adaptive mesh refinement can only be as good as the underlying refinement indicator.

4.3 An Only Initially Localised Feature: A Centred Impulse

In this subsection we investigate the performance of the AMR method for a problem involving an initially localised impulse that develops into distinct fronts travelling throughout the domain. To illustrate this further, let us consider the following: Let $\Omega = [2, 2]^2$ be a square domain with periodic boundary conditions. The initial water height is defined by $D(\mathbf{x}, 0) = 1.5 \cdot \mathrm{e}^{\frac{-\|\mathbf{x}-\mathbf{x}_c\|_2^2}{\alpha}}$ with a shape parameter $\alpha = 0.001$,

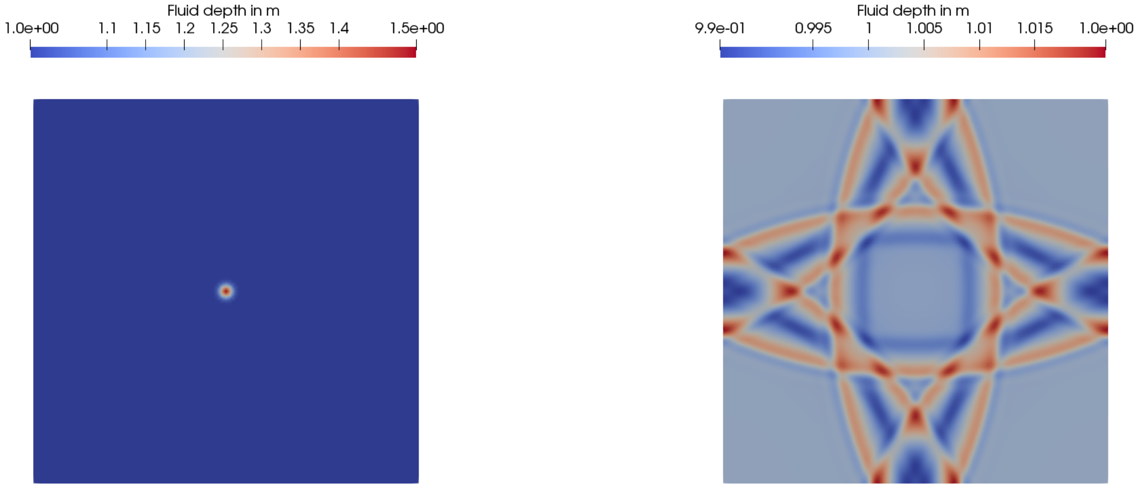

Figure 9: Centred Impulse: Plots of fluid height D at times $t = 0$s (left) and $t = 0.75$s (right) on a uniform mesh. The colour scales have been adjusted in both plots to increase readability.

and $\mathbf{x}_c = (1, 1)^\top$, the centre of the domain. The bathymetry is flat $b(\mathbf{x}) = 0$ and velocities are assumed to be zero, i.e. $\mathbf{u} = \mathbf{0}$.

4.3.1 Discussion of Simulation Results

Fig. 9 shows two snapshots of the numerical solution of the fluid height D on a uniformly refined grid with $\lambda_{ref} = 13$ demonstrating the dynamics of the problem. We have run simulations with a number of varying spatial resolutions with corresponding mesh levels $6 \leq \lambda_{ref} \leq 13$. In the absence of an analytical solution we compute the numerical error e_ρ defined in Eq. 1 using the numerical solution on a uniformly fine mesh with $\lambda_{ref} = 13$ as a reference solution. Fig. 10 shows an example of such a comparison with an adaptive mesh simulation with $\lambda_{crs} = 6$ and $\lambda_{ref} = 12$. The adaptive mesh as plotted in the third row of Fig. 10 captures the emerging waves. The overall numerical error is increasing over time as can be seen in the bottom row in Fig. 10 where we show the absolute point-wise difference between the adaptive simulation and the numerical reference solution.

4.3.2 Performance Metrics Evaluation

Using the L_2-norm at the final time step to compute the numerical error as described above, we can compute the metrics defined in Section 2. The result can be found in Figure 11 and allows for many of the same conclusions as the previous sections. Using $r_{\mathrm{d_{fixed}}}$, we can see that consequently in the ℓ_2 norm, the same error can be achieved with a lower number of degrees of freedom using the adaptive mesh capabilities. Furthermore, r_{res} allows for the conclusion that a smaller mesh width is achieved with less DOF (n_Ω). Both metrics, $r_{\mathrm{t-to-sol}}$ and $r_{\mathrm{eff-res}}$ however show an interesting behaviour. Below an error threshold of about 10^{-4}, the AMR simulation is computationally more expensive. This is directly caused by the high average number of DOFs and the costs of constant mesh manipulation. Overall, this can be seen as due to the problem dynamics and the relatively large area of the domain that is covered by wave features at this later time. The effective resolution metric correspondingly shows an analogous behaviour. The reason for this can be seen from $r_{\mathrm{t-per-DOF}}$: Since the refined area (area of the mesh, where the mesh level is λ_{ref}) increases over time, additional computational overhead is created. This overhead increases with increased number of DOF because additional elements are flagged for refinement and manipulated. We remark that overall the metrics do not lead to the conclusion that AMR is not useful for this type of application as some benefits are still achieved. The cost per DOF slightly increases over time with increasing area of maximum refinement as can be seen from Figure 12. It shows that towards the end of the simulation the cost per DOF increases by a factor of 2 at certain time points. We conclude that especially for highly time sensitive applications such as tsunami propagation adaptive simulations do lead to an increased efficiency especially during the important first moments of the simulation. This is furthermore supported by Figure 13 which shows (left display) that the number of elements is increasing

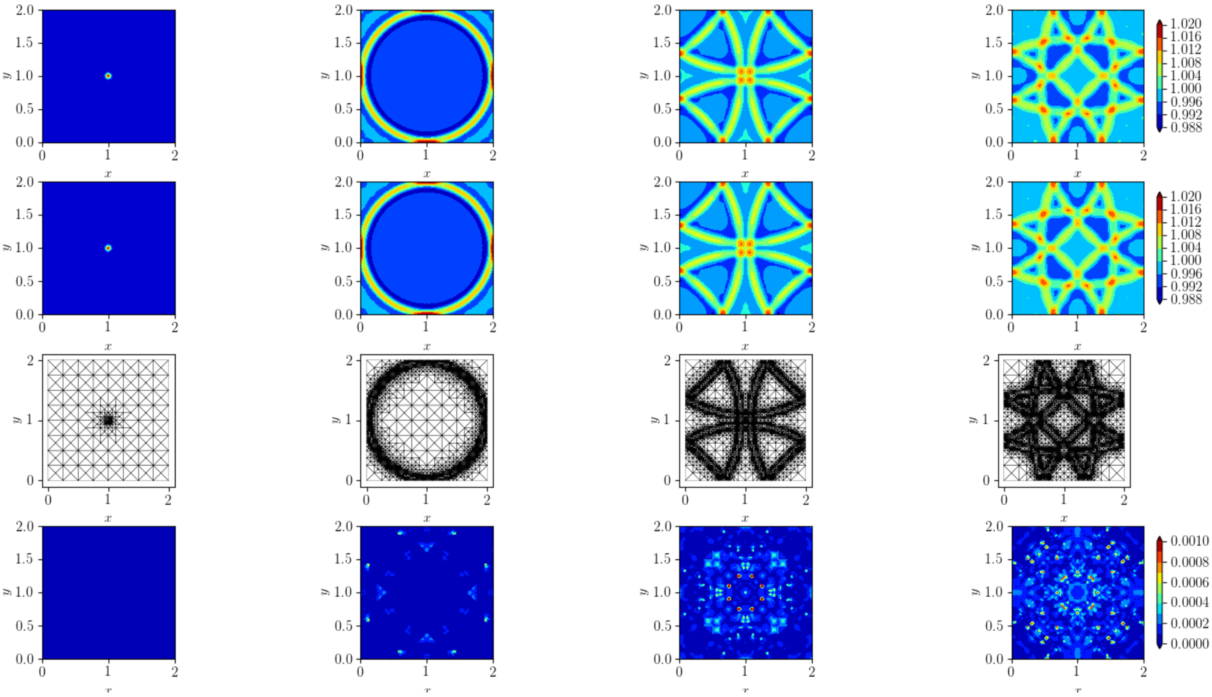

Figure 10: Centred Impulse: Snapshots of numerical solution for D at times $t = 0, 0.5, 1, 2$s (from left to right). Depicted are the simulation result on the uniform reference mesh (top), an adaptive simulation (second row) using $\lambda_{ref} = 12$ and $\lambda_{crs} = 6$ with the corresponding mesh below and the point wise numerical error (bottom).

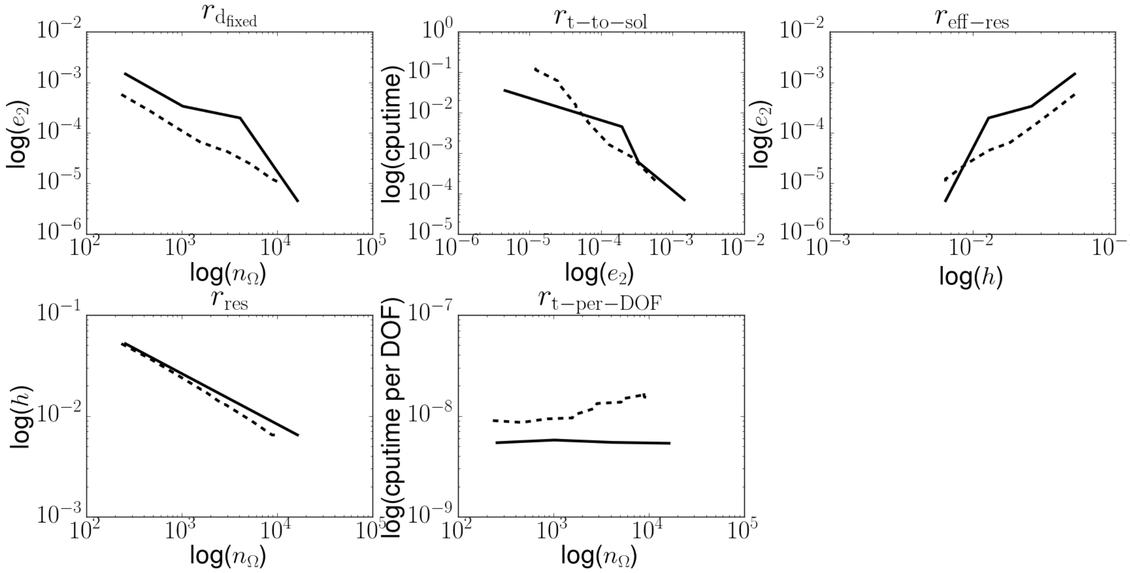

Figure 11: Centred Impulse: Plots of metrics $r_{\mathrm{d_{fixed}}}, r_{\mathrm{t-to-sol}}, r_{\mathrm{eff-res}}, r_{\mathrm{res}}$, and $r_{\mathrm{t-per-DOF}}$ from top left to bottom right for uniform simulations (solid line), adaptive simulations (dashed line).

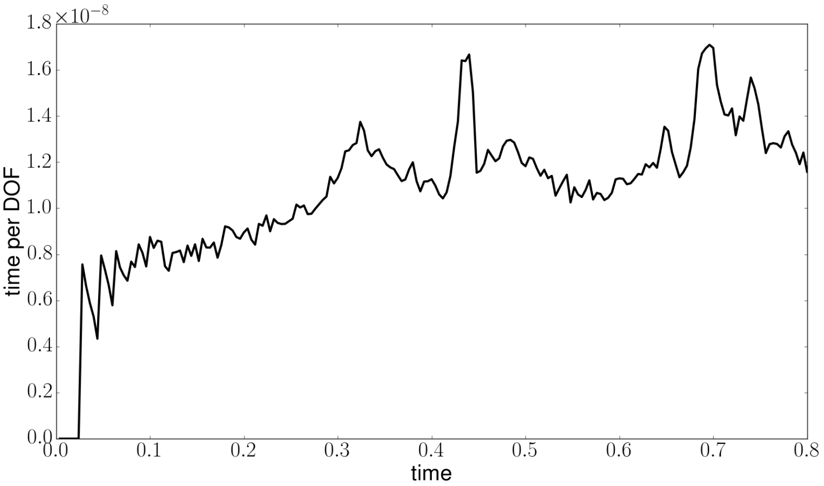

Figure 12: Centred Impulse: Time spent per DOF over time until $t = 0.8$s for an adaptive simulation with $\lambda_{ref} = 10$ and $\lambda_{crs} = 7$.

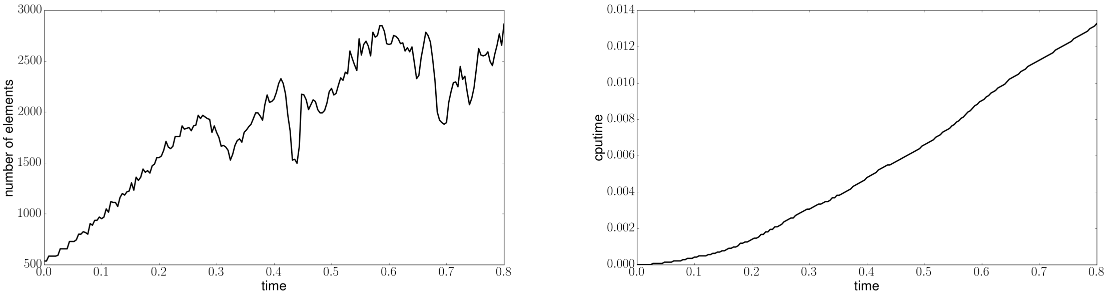

Figure 13: Centred Impulse: Number of elements over time (left) and cpu time over time (right) until $t = 0.8$s for an adaptive simulation with $\lambda_{ref} = 10$ and $\lambda_{crs} = 7$.

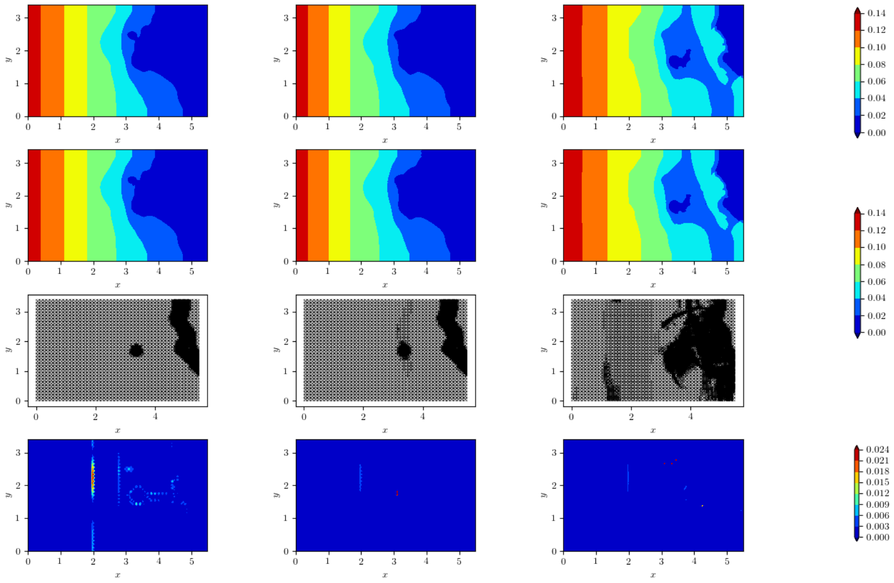

Figure 14: Okushiri Tsunami: Comparison of uniform (top), and adaptive simulation (second row) with corresponding adaptive mesh (third row) and point-wise numerical error (bottom).

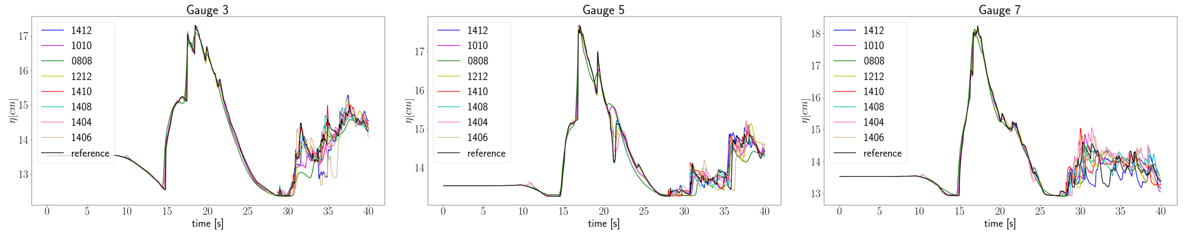

Figure 15: Okushiri Tsunami: Gauge Data Comparison

over time in order to capture the occurring waves. Moreover, as can be seen from the right display, we observe that the CPU time is not linearly increasing over time. This is an indication that AMR is still useful even if only temporarily the feature of interest is spatially localised.

4.4 A Realistic Test Case: The Okushiri Tsunami

Finally, we will apply the metrics to a realistic test case. This test case has already been presented on a uniform mesh in Vater et al. (2019) and is reproduced here using the aforementioned adaptive mesh capabilities. We have run the simulation with adaptive mesh refinement and refined according to total height divergence:

$$\eta_\Omega(t) = \nabla \cdot (D(\mathbf{x},t) + b(\mathbf{x})) \qquad \text{for } \mathbf{x} \in \Omega.$$

This indicator ensures that areas are refined where total height gradients occur, i.e. coastal features in shallow water where D is small as well as travelling waves on the open ocean where the effect of b is dwarfed by variations of D. A highly resolved coastline is important for this test case to ensure that the numerical solution is not affected by artificial waves stemming from interpolation errors impacting well-balancing or mass conservation. The numerical solution on a uniform and adaptive mesh at times $t = 0, 8.6, 17$s can be seen in the first and second row of Fig. 14. The associated adaptive mesh is shown in the third row and shows that the mesh is refined to capture the waves. Finally, the bottom row shows the point-wise difference between the numerical solutions and shows good agreement between the two. The numerical model used in this study has been verified using discrete data at several wave gauges. Simulation results using different meshes at three of these gauges are shown in Fig. 15. We can see that

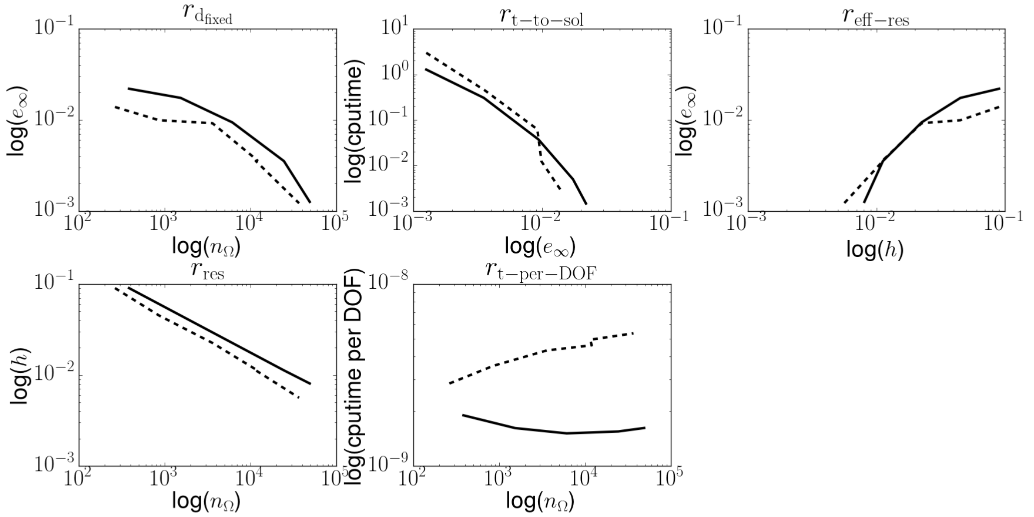

Figure 16: Okushiri Tsunami: Plots of metrics $r_{d_{fixed}}, r_{t-to-sol}, r_{eff-res}, r_{res}$, and $r_{t-per-DOF}$ from top left to bottom right for uniform simulations (solid line), and adaptive simulations (dashed line).

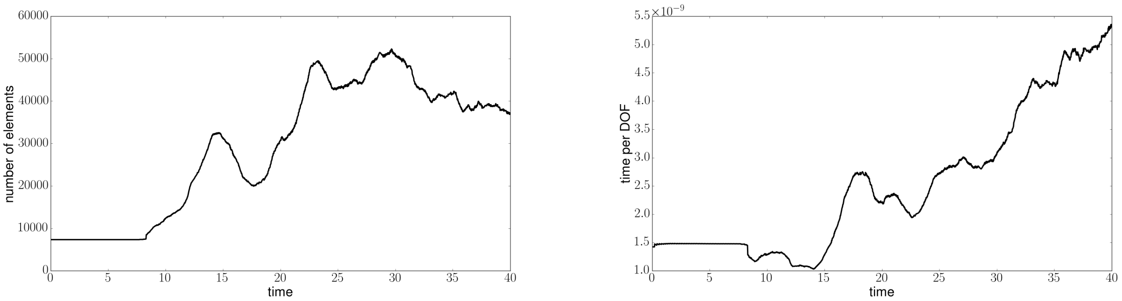

Figure 17: Okushiri Tsunami: Number of elements over time (left) and CPU time spent per DOF over time (right) until $t = 40$s for an adaptive simulation with $\lambda_{ref} = 14$ and $\lambda_{crs} = 6$.

all model resolutions reproduce the peaked wave well and only differ in the reproduction of a second wave after $t = 26$s.

4.4.1 Performance Metrics Evaluation

Applying the metrics defined in Sec. 2 to this test case, we obtain Fig. 16. For the error, we computed the mean maximum error over all three gauges

$$e_\infty = \frac{1}{3}\sum_k \max_{t\leq 26} |D_h(t) - D_{\mathrm{ref}}(t)|$$

where $k = 1, 2, 3$ are the three wave gauges as shown in Fig. 15 and D_{ref} is a high-resolution numerical solution with $\lambda_{ref} = 14$. The metric $r_{d_{fixed}}$ shows us that the AMR simulations achieve a smaller error at these wave gauges given a fixed number of DOF which is in line with the previous test cases. Furthermore, we can see from r_{res} in the bottom left of Fig. 16 that for a given number of DOF, we achieve a finer resolution with the AMR method. This is important when we are interested in effects that we know require a certain (fine) resolution. One drawback for this realistic test case can be seen through the metrics $r_{t-to-sol}$ and $r_{t-per-DOF}$: The computational overhead for a realistic test case is significant and outweighs the reduction in DOF if an error below a certain threshold (in our case approximately 10^{-2} is to be achieved. An explanation can be found in the number of mesh levels that are present in high-resolution simulations in combination with the very fast moving dynamics of the problem. Similar to Sec. 4.3, this does not indicate that the AMR method does not yield any advantage. A look at Fig. 17 shows that the number of elements increases by a factor of five during the simulation time which leads to an increase in CPU time spent per DOF as mesh manipulation becomes more involved. Overall, this points towards the AMR method being useful especially for the first $10 - 15$ seconds where its advantages are not outweighed by the model dynamics demanding high resolution almost everywhere.

5 Discussion and Concluding Remarks

In this study we have introduced a set of metrics that measures the computational efficiency of adaptive mesh refinement (AMR). Our focus then was on dynamically adaptive triangular meshes that were refined by bisection along longest edges using physics-based refinement indicators $\eta_{\mathcal{T}_i}$ as described in subsection 3.1. Insight into the dynamics of the numerical test problems allowed us to find indicators that were good proxies for numerical errors. Using the Discontinuous Galerkin model described in Beisiegel et al. (2019) as an example, we studied a set of four test cases:

(1) A spatially localised quasi-stationary travelling vortex;

(2) a challenging long wave resonance in a paraboloid basin with wetting and drying;

(3) a centred impulse that is only initially spatially localised; and

(4) a realistic simulation of the Okushiri tsunami.

The test cases cover a range of different aspects, from the size of the area of high resolution, degrees of localisation of features of interest as well as realism of the simulation. Using the metrics in Section 2, we then compared a number of uniform simulations with AMR simulations using the same numerical model. The metrics correlate numerical error to numbers of DOF n_Ω and CPU time as well as local (minimal) resolution h and offered insights into AMR performance.

Overall, using the metric $r_{d_{fixed}} = e_2/n_\Omega$, we consistently find that simulations using AMR achieve the same numerical ℓ_2 error with significantly fewer DOFs. Re-defining convergence by using n_Ω instead of mesh width h, the metric $r_{d_{fixed}}$ can furthermore be used to compare convergence properties of uniform and adaptive simulations. For all test cases we find that the dynamically adaptive mesh leads to an accelerated convergence, in case of the stationary vortex in subsection 4.1 even by up to 40%.

This can have an impact on the total run time of the solution as well. For fairly localised and well controlled features of interest as in test cases (1) and (2), the metric $r_{t-to-sol} = t/e_2$ shows that the same simulation accuracy can be achieved with less computational effort when using AMR.

A different behaviour can be seen for test cases (3) and (4) which after being initially localised exhibit more complicated and dynamically changing wave structures that need to be resolved. For lower resolutions, the comparison with uniform simulations matches test cases (1) and (2). However, the higher the resolution (and the lower the error), the more computational overhead is created through mesh manipulation, ultimately increasing the overall cost of the AMR simulation above the uniform simulation. Two conclusions can be drawn from this. For one, depending on the numerical error that one is interested in achieving, a uniform simulation might be cheaper. On the other hand, using information on the number of DOF over time as in Fig. 17, one could also use an AMR simulation for as long as it is computationally advantageous and then, switch to a uniform simulation for times after this. In any case, the metrics presented allow for an accurate quantitative assessment of whether to use AMR or uniform computations.

One of the major arguments against AMR is that mesh manipulation and management add additional computational overhead to the overall cost of the simulation. Using the metric $r_{t-per-DOF} = t/n_\Omega$ we studied the computational cost per DOF and find that although adaptive simulations are more expensive per DOF, the additional cost is widely independent of n_Ω. In fact, we find that the overhead appears to be constant per DOF for bounded areas of high resolution. These advantages vanish when the dynamics of the problem exhibit not localised phenomena with growing areas of high resolution such as was the case in subsections 4.3 and 4.4.

Since AMR is known as a tool that allows for the consistent simulation of multi-scale phenomena where small scale features interact with larger scales, we studied local resolution h using the two mesh metrics $r_{res} = h/n_\Omega$ and $r_{eff-res} = e_2/h$. These show that using AMR achieves the same ℓ_2 error with a smaller spatial resolution, i.e. at a given error, you will resolve finer scale phenomena. Furthermore, as r_{res} shows, AMR achieves a higher spatial resolution given a fixed number of DOF. In a comparison of two different mesh refinement strategies for the quasi-stationary vortex in subsection 4.1, we furthermore found that AMR is most efficient if the coarse mesh level λ_{crs} is kept as small (i.e. as coarse) as possible. This is because for a set of adaptive simulations with constant d_λ the metric r_{res} evolves in a same way as for the uniform simulations. Finally, we demonstrated in subsections 4.3 and 4.4 that for even only initially localised phenomena, we can achieve a computationally more efficient simulation using AMR at least for an initial period of time. In practice we expect this to be especially important for highly time critical applications such as tsunami propagation.

The presented metrics may in future help to investigate computational and numerical properties of AMR methods more rigorously. It was our intention to demonstrate their usefulness along one example code. It would now be interesting to see other AMR code developers adopt these metrics and see how different numerical schemes and different implementations expose their characteristic properties.

Acknowledgements

We acknowledge funding by the Irish Research Council (IRC) under the research project "NIMBUS^3: Next-Generation Integrated Model for Better and Unified Storm Surge Simulations" (GOIPD/2018/248). Furthermore, we wish to acknowledge the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support. C.E.C. acknowledges support by Proyecto Mayor UTA 8718-16, Universidad de Tarapacá. J.B. also acknowledges support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2037 'CLICCS – Climate, Climatic Change, and Society' – Project Number: 390683824, contribution to the Center for Earth System Research and Sustainability (CEN) of Universität Hamburg.

References

Alauzet, F., and A. Loseille, 2016: A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Computer-Aided Design*, **72**, 13–39.

Beckers, S., J. Behrens, and W. Wollner, 2019: Duality based error estimation in the presence of discontinuities. *Applied Numerical Mathematics*, **144**, 83 – 99, doi: 10.1016/j.apnum.2019.05.016.

Behrens, J., 2006: *Adaptive Atmospheric Modeling*. Springer, Heidelberg, Berlin, doi: 10.1007/3-540-33383-5.

Behrens, J., and M. Bader, 2009: Efficiency considerations in triangular adaptive mesh refinement. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **367 (1907)**, 4577 –4589, doi: 10.1098/rsta.2009.0175.

Behrens, J., N. Rakowsky, W. Hiller, D. Handorf, M. Läuter, J. Päpke, and K. Dethloff, 2005: amatos: Parallel adaptive mesh generator for atmospheric and oceanic simulation. *Ocean Modelling*, **10 (1-2)**, 171–183, doi: 10.1016/j.ocemod.2004.06.003.

Beisiegel, N., 2014: High-order Adaptive Discontinuous Galerkin Inundation Modeling. Ph.D. thesis, Universität Hamburg, URL http://ediss.sub.uni-hamburg.de/volltexte/2014/7036.

Beisiegel, N., S. Vater, J. Behrens, and F. Dias, 2019: An Adaptive Discontinuous Galerkin Method on Unstructured Meshes for the Simulation of Hurricane Storm Surge. *Ocean Dynamics*, URL 10.31223/osf.io/5kedt, under review.

Brenner, S. C., and L. R. Scott, 2000: *The Mathematical Theory of Finite Element Methods*. Third edition ed., No. 14, Texts in Applied Mathematics, Springer.

Castro, C., E. Toro, and M. Käser, 2012: ADER scheme on unstructured meshes for shallow water: simulation of tsunami waves. *Geophysical Journal International*, **189 (3)**, 1505–1520, doi: 10.1111/j.1365-246X.2012.05471.x.

Castro-Díaz, M., F. Hecht, B. Mohammadi, and O. Pironneau, 1997: Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, **25 (4)**, 475–491.

Dietachmayer, G. S., and K. K. Droegemeier, 1992: Application of Continuous Dynamic Grid Adaption Techniques to Meteorological Modeling. Part I: Basic Formulation and Accuracy. *Monthly Weather Review*, **120 (8)**, 1675–1706, doi: 10.1175/1520-0493(1992)120⟨1675:AOCDGA⟩2.0.CO;2.

Dörfler, W., 1996: A Convergent Adaptive Algorithm for Poisson's Equation. *SIAM Journal on Numerical Analysis*, **33 (3)**, 1106–1124.

Farrell, P., D. Ham, S. Funke, and M. Rognes, 2013: Automated Derivation of the Adjoint of High-Level Transient Finite Element Programs. *SIAM Journal on Scientific Computing*, **35 (4)**, C369–C393, doi: 10.1137/120873558.

Giraldo, F. X., 2000: The LagrangeGalerkin method for the two-dimensional shallow water equations on adaptive grids. *International Journal for Numerical Methods in Fluids*, **33 (6)**, 789–832, doi: 10.1002/1097-0363(20000730)33:6⟨789::AID-FLD29⟩3.0.CO;2-1.

Gu, X., H. Schock, T. Shihand, E. Hernandez, and D. Chu, 2001: Grid-quality measures for structured and unstructured meshes. *39th Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, Aerospace Sciences Meetings, doi: doi:10.2514/6.2001-652, URL https://doi.org/10.2514/6.2001-652.

Hermann, V., M. Kaser, and C. E. Castro, 2011: Non-conforming hybrid meshes for efficient 2D wave propagation using the Discontinuous Galerkin method. *Geophys. J. Int.*, **184 (2)**, 746–758.

Hesthaven, J. S., and T. Warburton, 2008: *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer.

Iselin, J. P., J. M. Prusa, and W. J. Gutowski, 2002: Dynamic Grid Adaptation Using the MP-DATA Scheme. *Monthly Weather Review*, **130 (4)**, 1026–1039, doi: 10.1175/1520-0493(2002)130⟨1026:DGAUTM⟩2.0.CO;2.

Jameson, L., 2000: High Order Schemes for Resolving Waves: Number of Points per Wavelength. *Journal of Scientific Computing*, **15 (4)**, 417–439, doi: 10.1023/A:1011180613990.

Kopera, M. A., and F. X. Giraldo, 2014: Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *Journal of Computational Physics*, **275**, 92–117.

LeVeque, R. J., 2002: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics, Cambridge University Press, doi: 10.1017/CBO9780511791253.

Luettich, R., and J. Westerink, 1995: Implementation and Testing of Elemental Flooding and Drying in the ADCIRC Hydrodynamic Model. Tech. rep., DEPARTMENT OF THE ARMY, Coastal Engineering Research Center, Waterways Experiment Station, US Army Corps of Engineers. Final Contractors Report.

Lynett, P. J., T.-R. Wu, and P. L. Liu, 2002: Modeling wave runup with depth-integrated equations. *Coastal Engineering*, **46**, 89 –107.

Mandli, K. T., and C. N. Dawson, 2014: Adaptive mesh refinement for storm surge. *Ocean Modelling*, **75**, 36–50.

Müller, A., J. Behrens, F. X. Giraldo, and V. Wirth, 2013: Comparison between adaptive and uniform discontinuous Galerkin simulations in dry 2d bubble experiments. *Journal of Computational Physics*, **235**, 371–393, doi: https://doi.org/10.1016/j.jcp.2012.10.038.

Piggott, M. D., C. C. Pain, G. J. Gorman, D. P. Marshall, and P. D. Killworth, 2008: Unstructured adaptive meshes for ocean modeling. *Ocean Modeling in an Eddying Regime*, H. Hasumi, and M. W. Hecht, Eds., AGU, 383–408, doi: 10.1029/177GM22.

Rivara, M. C., 1984: Algorithms for Refining Triangular Grids Suitable for Adaptive and Multigrid Techniques. *International Journal for Numerical Methods in Engineering*, **20**, 745–756.

Thacker, W. C., 1981: Some exact solutions of the nonlinear shallow-water wave equations. *J. Fluid Mech.*, **107**, 499–508.

Toro, E. F., 2009: *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer.

Vater, S., 2013: A Multigrid-based Multiscale Numerical Scheme for Shallow Water Flows at Low Froude Number. Ph.D. thesis, Freie Universität Berlin, URL http://www.diss.fu-berlin.de/diss/receive/FUDISS$_$thesis$_$000000093897.

Vater, S., N. Beisiegel, and J. Behrens, 2019: A limiter-based well-balanced discontinuous Galerkin method for shallow-water flows with wetting and drying: Triangular grids. *Int J Num Meth Fluids*, **91 (8)**, 395–418, doi: 10.1002/fld.4762.