



p-adaptive discontinuous Galerkin method for the shallow water equations with a parameter-free error indicator

Sara Faghih-Naini^{1,2} · Vadym Aizinger¹

Received: 28 March 2022 / Accepted: 23 September 2022 / Published online: 11 October 2022
© The Author(s) 2022

Abstract

We propose a p-adaptive quadrature-free discontinuous Galerkin method for the shallow water equations based on a computationally efficient adaptivity indicator which works without any problem-dependent parameters. The error and smoothness of the solution are detected using the information collected for slope limiting and, for piecewise constant discretizations, by carrying out a reconstruction procedure. The accuracy and robustness of the new scheme are evaluated using several benchmarks and compared to other adaptivity indicators. Our results indicate that the proposed indicator finds a good balance between solution quality and computational overhead.

Keywords p-adaptivity · Error and smoothness indicator · Discontinuous Galerkin method · Quadrature-free · Shallow water equations · Ocean modeling

Mathematics Subject Classification 65 · 76 · 86

1 Introduction

Discontinuous Galerkin (DG) methods combine some of the most attractive features of the classical (continuous) finite element and finite volume methods; they are widely used nowadays in many areas of computational science, particularly for fluid dynamics simulation. Thanks to local approximation stencils, DG schemes have a very favorable computation/communication ratio and are therefore well suited for high performance computing applications that utilize distributed memory parallelization or heterogeneous hardware architectures.

✉ Vadym Aizinger
vadym.aizinger@uni-bayreuth.de

¹ Chair of Scientific Computing, University of Bayreuth, 95440 Bayreuth, Germany

² Chair for System Simulation, Friedrich-Alexander University Erlangen-Nuremberg, 91058 Erlangen, Germany

One of the most interesting aspects of the DG methodology is the support for the computational mesh (h-) or the local approximation space (p-) adaptivity. However, adaptive discretizations need indicators to guide the refinement/derefinement. A wide variety of different error and smoothness indicators exists in the literature (see, for example, an overview in Naddei et al. 2019). Generally, these approaches can be subdivided into several classes according to the information used to evaluate the local discretization error (or the local solution regularity). A simple and computationally efficient scheme (Eskilsson 2011; Tumolo et al. 2013) measures the so-called local spectral decay of the DG solution, i.e., compares the absolute or relative magnitudes of the degrees of freedom corresponding to different polynomial orders. Another indicator, which has been already successfully used for the shallow water equations (SWE), estimates local solution gradients (Kubatko et al. 2009; Michoski et al. 2011). In Michoski et al. (2011), the first attempt was made to combine a slope limiter with a p-adaptive DG method; however, contrary to our approach, no slope limiting information was utilized by the adaptivity procedure and vice versa.

Our indicator uses techniques introduced in Krivodonova et al. (2004) for so-called non-conformity estimators which measure the absolute or relative size of discontinuity between the element-local solutions – mostly by integrating solution jumps over the inter-element boundaries. However, a new indicator became necessary due to additional requirements resulting from the specifics of our application:

- Highly dynamic character of typical simulation scenarios for the SWE (e.g., tidal waves or tsunamis) requires a computationally efficient procedure,
- Support for piecewise constant polynomial spaces – the indicator must be able to detect local errors using the lowest order approximation space—cannot be provided using, e.g., the spectral decay methods.

The adaptivity indicator proposed in this work aims to fulfill the above requirements, does not need any problem-specific (e.g. sensitivity) parameters, and is integrated into the vertex-based slope limiter (Aizinger 2011; Kuzmin 2010; Aizinger et al. 2017; Hajduk et al. 2018, 2019) further reducing the total computational cost of the scheme. In addition, our DG discretization based on modal hierarchical bases is highly suited for p-adaptive schemes – even more so since it relies on a quadrature-free formulation introduced in Faghih-Naini et al. (2020). Analytic evaluation of all element and edge integrals completely avoids the main overhead of varying-order approximation spaces: The necessity to use the most accurate (and thus the most expensive) quadrature rule or to maintain several quadrature rules of different orders.

We begin, in the next section, by introducing the mathematical model and its discretization by a quadrature-free DG method. Section 3 details our adaptivity indicator and two other indicators used for comparison. Numerical results evaluating the performance of our scheme are presented in Sect. 4 followed by a short Conclusions and outlook section.

2 Quadrature-free DG formulation for the shallow water equations

2.1 Governing equations

The presentation in this section closely follows (Faghih-Naini et al. 2020) and is included here for completeness. We start with the classical 2D shallow water equations defined on some 2D domain Ω and given by

$$\partial_t \xi + \nabla \cdot \mathbf{q} = 0, \tag{1}$$

$$\partial_t \mathbf{q} + \nabla \cdot \left(\mathbf{q} \mathbf{q}^T / H \right) + \tau_{\text{bf}} \mathbf{q} + \begin{pmatrix} 0 & -f_c \\ f_c & 0 \end{pmatrix} \mathbf{q} + gH \nabla \xi = \mathbf{F}, \tag{2}$$

where ξ represents the elevation of the free water surface with respect to some datum (e.g., the mean sea level). Using h_b to denote the bathymetric depth, $H = h_b + \xi$ is then the total water depth. $\mathbf{q} := (U, V)^T$ denotes the depth integrated horizontal velocity field, f_c the Coriolis coefficient, g the gravitational acceleration, τ_{bf} the bottom friction coefficient, and \mathbf{F} the body force accounting for the variable atmospheric pressure and tidal potential.

Introducing the notation $\mathbf{c} := (\xi, U, V)^T$, system (1), (2) can be re-formulated in the following compact form:

$$\partial_t \mathbf{c} + \nabla \cdot \tilde{\mathbf{A}}(\mathbf{c}) = \mathbf{r}(\mathbf{c}), \tag{3}$$

where

$$\tilde{\mathbf{A}} = \begin{pmatrix} \frac{U}{H} + \frac{g\xi(H+h_b)}{2} & \frac{UV}{H} \\ \frac{UV}{H} & \frac{V}{H} + \frac{g\xi(H+h_b)}{2} \end{pmatrix}, \quad \mathbf{r} = \begin{pmatrix} 0 \\ -\tau_{\text{bf}}U + f_cV + g\xi \partial_x h_b + F_x \\ -\tau_{\text{bf}}V - f_cU + g\xi \partial_y h_b + F_y \end{pmatrix}.$$

Taking into account the relation $\mathbf{q} = \mathbf{u}H$, where $\mathbf{u} = (u, v)^T$ is the depth averaged velocity, system (3) can be re-formulated as follows:

$$\partial_t \mathbf{c} + \nabla \cdot \mathbf{A}(\mathbf{c}, \mathbf{u}) = \mathbf{r}(\mathbf{c}), \tag{4}$$

$$\mathbf{u}H = \mathbf{q}, \tag{5}$$

where

$$\mathbf{A}(\mathbf{c}, \mathbf{u}) = \begin{pmatrix} Uu + \frac{g\xi(H+h_b)}{2} & Uv \\ Vu & Vv + \frac{g\xi(H+h_b)}{2} \end{pmatrix}. \tag{6}$$

With the help of the auxiliary vector \mathbf{u} calculated *diagnostically* from (5), this formulation, first presented in Faghih-Naini et al. (2020), avoids fraction-type nonlinearities in the momentum equation (2) and is better suited for quadrature-free integration.

This work uses the following types of boundary conditions for the SWE:

Land boundary: At a land boundary, we assume no normal flow

$$\mathbf{q} \cdot \mathbf{n} = 0.$$

Open-sea boundary: Denoting by $\hat{\xi}$ the prescribed free surface elevation we set at open sea boundaries

$$\xi(t, \mathbf{x}) = \hat{\xi}(t, \mathbf{x}).$$

River boundary: For supercritical flow examples, we set the following river (inflow) boundary conditions:

$$q_n(t, \mathbf{x}) = \hat{q}_n(t, \mathbf{x}), \quad q_\tau(t, \mathbf{x}) = \hat{q}_\tau(t, \mathbf{x}), \quad \xi(t, \mathbf{x}) = \hat{\xi}(t, \mathbf{x})$$

with the normal and tangential integrated velocities $\hat{q}_n(t, \mathbf{x})$ and $\hat{q}_\tau(t, \mathbf{x})$.

Radiation boundary: At the outflow boundary of supercritical flow examples, we specify radiation boundary conditions where no unknowns are prescribed.

Lastly, we provide initial conditions for the elevation and integrated velocity

$$\xi(0, \mathbf{x}) = \xi_0(\mathbf{x}), \quad \mathbf{q}(0, \mathbf{x}) = \mathbf{q}_0(\mathbf{x}). \tag{7}$$

2.2 Spatial discretization by a quadrature-free DG method

For $\{\mathcal{T}_\Delta\}_{\Delta>0}$, a family of triangulations of $\Omega \subset \mathbb{R}^2$, let Ω_e be an element of \mathcal{T}_Δ . The variational formulation of system (4), (5) is obtained by multiplication with sufficiently smooth test functions ϕ and ψ followed by the integration by parts on each element $\Omega_e \in \mathcal{T}_\Delta$ (we use the standard notation $(\cdot, \cdot)_{\Omega_e}$ and $\langle \cdot, \cdot \rangle_{\partial\Omega_e}$ for the L^2 -scalar products on elements and edges, respectively)

$$(\partial_t \mathbf{c}, \phi)_{\Omega_e} - (\mathbf{A}, \nabla \phi)_{\Omega_e} + \langle \mathbf{A} \cdot \mathbf{n}_e, \phi \rangle_{\partial\Omega_e} = (\mathbf{r}, \phi)_{\Omega_e}, \tag{8}$$

$$(\mathbf{u}H, \psi)_{\Omega_e} = (\mathbf{q}, \psi)_{\Omega_e}, \tag{9}$$

where \mathbf{n}_e denotes an exterior unit normal to $\partial\Omega_e$.

Denoting by $\mathbb{P}^p(\Omega_e)$ the polynomial space of order (i.e. the highest polynomial degree) p on Ω_e , the initial conditions for the semi-discrete problem are generated using an L^2 -projection of (7) into the corresponding discrete space. The (local) semi-discrete formulation is then obtained from (8), (9) by replacing \mathbf{c} and \mathbf{u} by their finitely-dimensional counterparts $\mathbf{c}_\Delta, \mathbf{u}_\Delta$ and utilizing test functions $\phi_\Delta \in \mathbb{P}^p(\Omega_e)^3, \psi_\Delta \in \mathbb{P}^p(\Omega_e)^2$:

$$(\partial_t \mathbf{c}_\Delta, \phi_\Delta)_{\Omega_e} - (\mathbf{A}, \nabla \phi_\Delta)_{\Omega_e} + \langle \hat{\mathbf{A}}, \phi_\Delta \rangle_{\partial\Omega_e} = (\mathbf{r}, \phi_\Delta)_{\Omega_e}, \tag{10}$$

$$(\mathbf{u}_\Delta H_\Delta, \psi_\Delta)_{\Omega_e} = (\mathbf{q}_\Delta, \psi_\Delta)_{\Omega_e}. \tag{11}$$

The edge flux $\mathbf{A}(\mathbf{c}_\Delta, \mathbf{u}_\Delta) \cdot \mathbf{n}_e$ is approximated on $\partial\Omega_e$ by a numerical flux $\hat{\mathbf{A}}(\mathbf{c}_\Delta, \mathbf{u}_\Delta, \tilde{\mathbf{c}}_\Delta, \tilde{\mathbf{u}}_\Delta, \mathbf{n}_e)$ that depends on discontinuous values of the solution on element Ω_e (without tilde) and on its edge neighbors (with tilde). On exterior domain

boundaries, the specified boundary values of the elevation and velocity are utilized in the flux computation. In this work, we rely – unless specified otherwise – on the Lax–Friedrichs flux (Hajduk et al. 2018) defined as

$$\widehat{\mathbf{A}}^{LF}(\mathbf{c}_\Delta, \mathbf{u}_\Delta, \tilde{\mathbf{c}}_\Delta, \tilde{\mathbf{u}}_\Delta, \mathbf{n}_e) = \frac{1}{2}(\mathbf{A}(\mathbf{c}_\Delta, \mathbf{u}_\Delta) + \mathbf{A}(\tilde{\mathbf{c}}_\Delta, \tilde{\mathbf{u}}_\Delta)) \cdot \mathbf{n}_e + \frac{\lambda}{2}(\mathbf{c}_\Delta - \tilde{\mathbf{c}}_\Delta),$$

where λ denotes the largest (in absolute value) eigenvalue of $\frac{\partial}{\partial \mathbf{c}} \tilde{\mathbf{A}}(\mathbf{c}) \cdot \mathbf{n}_e$. Denoting by \mathbf{x}_f the midpoint of edge $f \subset \partial\Omega_e$, our quadrature-free scheme uses the following approximation of $\lambda|_f$:

$$\lambda|_f := \max \{ |\mathbf{u}_\Delta(\mathbf{x}_f) \cdot \mathbf{n}_e|, |\tilde{\mathbf{u}}_\Delta(\mathbf{x}_f) \cdot \mathbf{n}_e| \} + \max \left\{ \sqrt{g H_\Delta(\mathbf{x}_f)}, \sqrt{g \tilde{H}_\Delta(\mathbf{x}_f)} \right\}. \tag{12}$$

Since the Lax–Friedrichs flux sometimes leads to rather diffusive results, we use in some cases the FORCE flux (Toro et al. 2009) instead with λ as described above. The FORCE flux can be constructed as the arithmetic mean between the Lax–Friedrichs flux and the two-step version of the Lax–Wendroff flux

$$\widehat{\mathbf{A}}^F = \frac{1}{2} \left(\widehat{\mathbf{A}}^{LF} + \widehat{\mathbf{A}}^{LW} \right) \tag{13}$$

where the latter is defined as $\widehat{\mathbf{A}}^{LW} = \mathbf{A}(\mathbf{Q}^{LW})$ with

$$\mathbf{Q}^{LW}(\mathbf{c}_\Delta, \mathbf{u}_\Delta, \tilde{\mathbf{c}}_\Delta, \tilde{\mathbf{u}}_\Delta, \mathbf{n}_e) = \frac{\lambda}{2}(\mathbf{A}(\mathbf{c}_\Delta, \mathbf{u}_\Delta) - \mathbf{A}(\tilde{\mathbf{c}}_\Delta, \tilde{\mathbf{u}}_\Delta)) \cdot \mathbf{n}_e + \frac{1}{2}(\mathbf{c}_\Delta + \tilde{\mathbf{c}}_\Delta).$$

Since the FORCE flux is more computationally expensive, it is only used if the Lax–Friedrichs flux produces too much numerical diffusion. Note that with either Lax–Friedrichs or FORCE flux and the parameter λ approximated as in (12), our semi-discrete formulation (10), (11) only contains nonlinearities in product form, thus all edge integrals are well suited for an analytical integration using a quadrature-free approach.

For simplicity, we formulate our methodology in the remainder of this section and in Sect. 3 for a generic scalar function $w(\mathbf{x})$ defined on Ω and its discretized counterpart $w_\Delta(\mathbf{x})$. Given $\varphi_{ei}(\mathbf{x})$, $i = 1, \dots, P(p)$, a basis of $\mathbb{P}^p(\Omega_e)$, w_Δ can be represented as

$$w_\Delta(t, \mathbf{x})|_{\Omega_e} = \sum_{i=1}^{P(p)} w_{ei}(t) \varphi_{ei}(\mathbf{x}) =: w_e^p,$$

that is, the superscript indicates the approximation order; no superscript means the default (full) approximation order. We also use a short-hand notation $p0, p1, \dots$ for piecewise constant, linear, \dots approximation spaces. The number of basis functions $P(p)$ depends on the respective polynomial order and the space dimension; it has the following values in \mathbb{R}^2 : $P(0) = 1$, $P(1) = 3$, and $P(2) = 6$. The basis functions used

in this work are chosen to be orthonormal with respect to the L^2 -scalar product on Ω_e (see, for example, Frank et al. 2015).

2.3 Temporal discretization and slope limiting

To avoid spurious oscillations in linear and superlinear DG solutions, we apply a vertex-based slope limiter following (Aizinger 2011; Kuzmin 2010) to piecewise linear and higher order approximations where appropriate. For orthonormal basis functions used in our work, the local representation of the limiting operator $\Pi : \mathbb{P}^p(\Omega_e) \rightarrow \mathbb{P}^p(\Omega_e)$ can be given by

$$\Pi(w_\Delta)|_{\Omega_e} = w_{e1}\varphi_{e1} + \alpha_e \sum_{i=2}^{P(p)} w_{ei} \varphi_{ei} \tag{14}$$

where the limiting factor $\alpha_e \in [0, 1]$ is computed as follows:

$$\alpha_e = \min_{i \in \{1,2,3\}} \begin{cases} \min\left(1, \frac{w_i^{\max} - w_e^0}{w_e(\mathbf{a}_{ei}) - w_e^0}\right), & \text{if } w_e(\mathbf{a}_{ei}) - w_e^0 > \varepsilon, \\ \min\left(1, \frac{w_i^{\min} - w_e^0}{w_e(\mathbf{a}_{ei}) - w_e^0}\right), & \text{if } w_e(\mathbf{a}_{ei}) - w_e^0 < -\varepsilon, \\ 1, & \text{otherwise,} \end{cases} \tag{15}$$

choosing $\varepsilon = 10^{-5}$ in the following. w_i^{\max} and w_i^{\min} denote the solution bounds given by the maximum and minimum of piecewise constant solutions w_e^0 on all elements sharing vertex \mathbf{a}_{ei} . Note that for an orthonormal basis, one has

$$w_e^0 = \frac{1}{|\Omega_e|} \int_{\Omega_e} w_\Delta(t, \mathbf{x})|_{\Omega_e} d\mathbf{x}.$$

If (15) results in $\alpha_e < 1$, all degrees of freedom corresponding to superlinear (quadratic and higher) basis functions in (14) are set to zero. In such cases, this limiting does not guarantee the exact preservation of bounds w_i^{\max} and w_i^{\min} , but the solution is mostly oscillation-free in most practical scenarios.

The temporal discretization of system (10), (11) is performed using a SSP (strong stability preserving) Runge–Kutta method (Gottlieb and Shu 1998). Following the presentation in Reuter et al. (2016), let $0 = t_1 < t_2 < \dots < t_{\text{end}}$ be a not necessarily equidistant decomposition of the time interval and $\Delta_n t := t_{n+1} - t_n$ the size of the n th time step. The update scheme of the s -stage SSP Rung–Kutta method (including limiting) is then given by

$$\begin{aligned} \mathbf{c}_\Delta^{(0)} &:= \mathbf{c}_\Delta(t_n, \cdot), \\ \mathbf{c}_\Delta^{(i)} &:= \Pi \left(\omega_i \mathbf{c}_\Delta^{(0)} + (1 - \omega_i) \left\{ \mathbf{c}_\Delta^{(i-1)} + \Delta_n t \mathbf{L}(\mathbf{c}_\Delta^{(i-1)}, t_n + \delta_i \Delta_n t) \right\} \right), \quad i = 1, \dots, s, \\ \mathbf{c}_\Delta(t_{n+1}, \cdot) &:= \mathbf{c}_\Delta^{(s)}, \end{aligned}$$

where L denotes the spatial discretization operator specified by (10), (11). In the test cases presented in Sect. 4, we use a two-stage SSP Runge–Kutta method with the coefficients $\omega_1 = 0, \omega_2 = 1/2, \delta_1 = 0, \delta_2 = 1$ (cf. Eq. (2.4) (Gottlieb and Shu 1998)).

3 Adaptivity indicators

3.1 Design of a new parameter-free adaptivity indicator

The goal of the new indicator, which will be referred to as JRL (Jump-Reconstruction-Limiting), is to identify resolved and under-resolved solution parts while distinguishing between smooth and non-smooth regions (e.g., shocks). Non-constant smooth regions should be approximated by piecewise linear or quadratic polynomials, constant regions should automatically revert to constant approximations. In regions with shocks or discontinuities, over- and undershoots should be avoided either by limiting, where possible, or by reducing the approximation order where necessary. Among the existing techniques in this area particularly relevant for our approach are the following ones:

- for continuous finite element methods, (Kuzmin and Schieweck 2013) utilized a gradient reconstruction (using a local L^2 -projection) as a parameter-free smoothness indicator for the unsteady linear advection equation;
- (Aizinger et al. 2017) introduced a flux-based gradient reconstruction which was then employed in the framework of anisotropic slope limiters for the DG method.

The indicator’s scheme is shown in Fig. 1; it combines the above two ideas with a vertex-based slope limiter to re-use pre-computed data in an effort to increase computational efficiency and to smoothly blend p-adaptivity with slope limiting. The following subsections detail indicator’s working principles.

3.1.1 Error detection

First, we need to introduce some notation. For the *base approximation order* $b := \max\{p - 1, 0\}$, the *jump* of element Ω_e is defined as

$$\llbracket w \rrbracket_e := \int_{\partial\Omega_e \cap \partial\Omega_{\tilde{e}}} |w_e^p - w_{\tilde{e}}^b| ds = \int_{\partial\Omega_e \cap \partial\Omega_{\tilde{e}}} \left| \sum_{i=1}^{P(p)} w_{ei} \varphi_{ei} - \sum_{i=1}^{P(b)} w_{\tilde{e}i} \varphi_{\tilde{e}i} \right| ds, \tag{16}$$

where \tilde{e} denotes the index of the edge-neighbor of element Ω_e . The *base jump* of element Ω_e is the jump when using the base order approximation on Ω_e

$$\llbracket w^b \rrbracket_e := \int_{\partial\Omega_e \cap \partial\Omega_{\tilde{e}}} |w_e^b - w_{\tilde{e}}^b| ds$$

It is mainly used in combination with $\llbracket w \rrbracket_e$ to estimate the solution regularity. In the numerical results section (Sect. 4), the adaptivity indicator as shown in Fig. 1 is applied

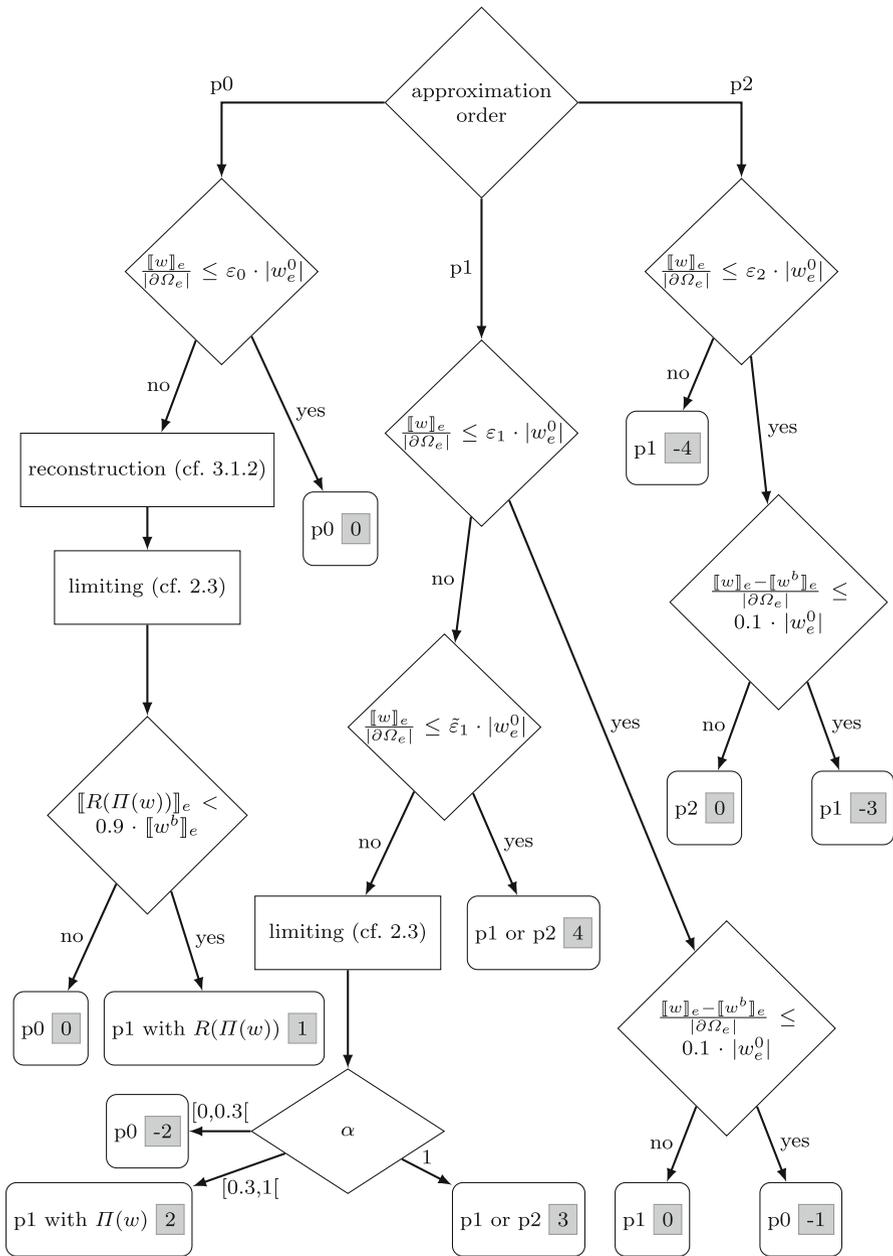


Fig. 1 Flow chart of adaptivity indicator JRL for orders 0, 1, and 2. Values in gray boxes stand for different adaptation cases, zero means no adaption, negative values indicate a decrease, and positive ones point to an increase of the approximation order

to the free surface elevation, that is, $w = \xi$. The decisions are made depending on the following thresholds: $\varepsilon_0 = 0.01$, $\varepsilon_1 = 0.005$, $\varepsilon_2 = 0.001$ and $\tilde{\varepsilon}_1 = 0.01$, which are problem independent and remain constant for all test cases and resolutions.

Furthermore, to avoid frequent in- and decrements in the local approximation order, we add an additional constraint: once the order is increased, it cannot be decreased for a fixed number of time steps chosen equal to 10 in all our test cases. Additionally, the approximation order of an element can only be increased by one at a time. This explains why the jump definition in (16) is somewhat different from the standard jump definition (cf. (20) in Sect. 3.2.1) where a full order approximation is considered. Our approach is independent of the decisions on approximation order in- and decrements involving neighboring elements—since we only permit changes by one polynomial order.

3.1.2 Gradient reconstruction

For a p0 solution $w_\Delta^0(t, \mathbf{x})$ with $\frac{\|w_\Delta^0\|_e}{|\partial\Omega_e|} \leq 0.01 \cdot |w_e^0|$, we construct a linear solution as follows using a rotationally invariant gradient reconstruction

$$R(w_\Delta)|_{\Omega_e} = w_e^0 + \frac{\partial w_e^1}{\partial x} \psi_{e2}(\mathbf{x}) + \frac{\partial w_e^1}{\partial y} \psi_{e3}(\mathbf{x}). \tag{17}$$

This reconstruction uses the linear Taylor basis functions ψ_{ei} as in Eq. (6) of (Kuzmin 2010) defined on the element Ω_e as

$$\psi_{e1}(\mathbf{x}) = 1, \quad \psi_{e2}(\mathbf{x}) = x - x_e^c, \quad \psi_{e3}(\mathbf{x}) = y - y_e^c,$$

where $\mathbf{x}_e^c = (x_e^c, y_e^c)^T = \frac{1}{3}(\mathbf{a}_{e1} + \mathbf{a}_{e2} + \mathbf{a}_{e3})$ denotes the centroid of Ω_e . The first coefficient is just the mean of the solution, the other two are obtained by taking the directional derivatives on all edges. Consider $\Omega_{\tilde{e}}$ with centroid $\tilde{\mathbf{x}}^c = (\tilde{x}^c, \tilde{y}^c)$ sharing an edge with Ω_e . The directional derivative of w_e on $\partial\Omega_e \cap \partial\Omega_{\tilde{e}}$ in direction $\mathbf{d}_{\tilde{e}} = \mathbf{x}_{\tilde{e}}^c - \mathbf{x}_e^c$ can then be approximated by

$$\frac{\partial w_e}{\partial \mathbf{d}_{\tilde{e}}} \approx \frac{w_{\tilde{e}}^0 - w_e^0}{|\mathbf{d}_{\tilde{e}}|}. \tag{18}$$

After computing the directional derivatives, we solve a least squares problem to obtain the gradient. Let N be the matrix containing the directions $\mathbf{d}_{\tilde{e}}$ in its rows, and let \mathbf{v}_e be the vector of directional derivatives defined in (18) for all neighboring elements $\Omega_{\tilde{e}}$. The missing coefficients from (17) are then obtained as the solution of the least squares problem:

$$\left(\frac{\partial w_e^1}{\partial x}, \frac{\partial w_e^1}{\partial y} \right)^T = (N^T N)^{-1} N^T \mathbf{v}_e. \tag{19}$$

Finally, the solution is transformed back into the orthonormal basis. For elements at the boundary of Ω , one can define the directional derivatives using a ghost layer. The above approach trivially generalizes to arbitrary polygons.

3.2 Two further indicators for comparison

To evaluate the performance of our new indicator, we perform in Sect. 4 comparisons to two other types of adaptivity indicators. The first one, described in Sect. 3.2.1, is a well established scheme which uses inter-element jumps to assess the local solution regularity. The other one, described in Sect. 3.2.2, is our own version of the gradient indicator enhanced in order to be applicable for p0 discretizations. It needs to be noted here that these two indicators need either one (cf. Sect. 3.2.1) or three (cf. Sect. 3.2.2) user-defined thresholds as input parameters. These parameters have to be calibrated manually and may differ for different scenarios, for limited and unlimited simulations as well as for different refinement levels of the same scenario.

3.2.1 Jump indicator

The indicator introduced in Remacle et al. (2003) computes the sum of the jumps across the edges of an element Ω_e , in our notation as

$$\llbracket w_\Delta(t, \mathbf{x}) \rrbracket_e := \int_{\partial\Omega_e \cap \partial\Omega_e^c} |w_e^p - w_e^p| ds. \quad (20)$$

The local approximation order is then increased if the total jump over the element is greater than the user-provided threshold, and a decrease is performed if it is smaller (also here the minimum of ten time steps between an increase and a decrease is enforced). This indicator can be considered as a simplified version of the JRL indicator since it also uses jumps for error detection. We apply it to the free surface elevation, that is set $w = \xi$, and refer to it in the following as JE (Jump-Estimation).

3.2.2 Gradient indicator

There exist gradient indicators in the literature such as the one introduced in Burbeau and Sagaut (2005) for the compressible Navier-Stokes equations and employed for the SWE in Kubatko et al. (2009). It computes gradients using the element-local solutions and therefore cannot be applied to p0-discretizations. To address this deficiency we designed a new scheme by taking directional derivatives in the directions $\mathbf{d}_f = \mathbf{x}_f - \mathbf{x}_e^c$, where \mathbf{x}_f denotes the midpoint of edge $f \subset \partial\Omega_e$. The directional derivative is then approximated using the element centroid value and the edge midpoint values taken from the current element and from its edge neighbor as follows

$$\frac{\partial w_e}{\partial \mathbf{d}_f} \approx \frac{w_e(\mathbf{x}_f) - w_e(\mathbf{x}_e^c)}{|\mathbf{d}_f|} \quad \text{and} \quad \frac{\partial \tilde{w}_e}{\partial \mathbf{d}_f} \approx \frac{w_e(\mathbf{x}_f) - w_e(\mathbf{x}_e^c)}{|\mathbf{d}_f|}.$$

To decide, whether an increment in the approximation order is necessary, comparison is done by evaluating

$$\left| \frac{\partial w_e}{\partial \mathbf{d}_f} - \frac{\partial \tilde{w}_e}{\partial \mathbf{d}_f} \right| < \varepsilon_w (\Delta_e)^p. \quad (21)$$

where $\Delta_e = \sqrt{2|\Omega_e|}$. In our implementation, this indicator is applied to three primary unknowns, that is, $w \in \{\xi, U, V\}$, in the following way. If the inequality does not hold for all directions and all three unknowns, the local approximation order of the respective element is increased. If all of them hold, the same comparison (21) is performed for the base order solution w_e^b (see Sect. 3.1.1) on the current element and using $(\Delta_e)^b$ in the upper bound in (21). If the criterion still holds for w_e^b , then the solution is approximated well enough by the lower order approximation, thus the order is decreased. The threshold ε_w has to be chosen individually for every unknown and scenario but can be chosen the same for different mesh resolutions. We refer to this indicator as the GRE (Gradient-Reconstruction-Extended).

4 Numerical results

To evaluate the accuracy and robustness of the proposed adaptivity indicator, we use three different benchmarks representing a wide range of spatial (constant, smoothly varying, shocks) and temporal (stationary, time-varying) flow regimes. In addition to our indicator, simulations utilizing two other indicators described in Sect. 3.2 were carried out and used for comparison purposes.

The simulations were performed with the ExaStencils code generation framework (Lengauer et al. 2020), which is based on the domain-specific language ExaSlang (Schmitt et al. 2014b) and outputs an optimized C++ or CUDA code. Our work extends the framework using a Python frontend – called GHODDESS and available at <https://i10git.cs.fau.de/ocean/ghoddess-release> – responsible for mapping the DG scheme to ExaSlang (Faghih-Naini et al. 2020). Recently, much effort was put into improving the performance of our p-adaptive scheme via algorithmic optimizations and various hybridization strategies controlling the distribution of the compute kernels between CPUs and GPUs. Runs on different architectures showed a speedup compared to a pure CPU or GPU implementation (see Faghih-Naini et al. 2023).

Note that no bottom friction τ_{bf} , Coriolis force f_c , or forcing F is used in any of the following test examples (cf. (2)).

4.1 Radial dam break

The radial dam break example is based on (LeVeque 2002; Hajduk 2021). We set $\Omega = [0, 5] \times [0, 5]$, $g = 1$, and a constant bathymetry $h_b = 0$. To make the test problem better suited for a p-adaptive approach, we chose the initial condition as

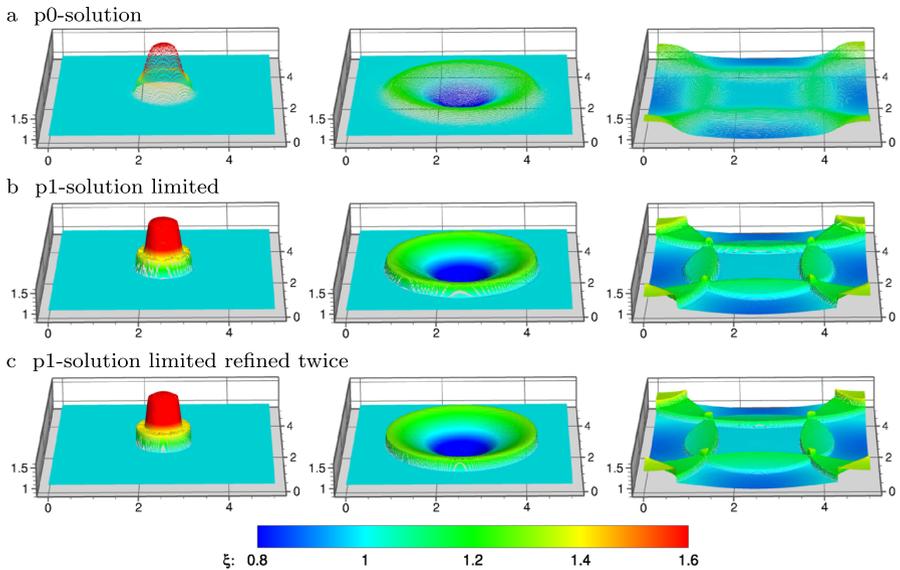


Fig. 2 Radial dam break solution for $t = 0.1$ s (left), $t = 1$ s (middle), and $t = 3$ s (right): constant (p0) (top), limited linear (p1) (middle), and limited linear (p1) on a twice uniformly refined mesh using the FORCE Riemann solver (bottom)

$$\xi(x, y, t) = \begin{cases} 2 + 0.5 e^{-15((x-2.5)^2 + (y-2.5)^2)}, & \text{if } (x - 2.5)^2 + (y - 2.5)^2 < 0.25, \\ 1, & \text{otherwise,} \end{cases}$$

$$U(x, y, t) = 0, \quad V(x, y, t) = 0.$$

The simulation results shown in Fig. 2 (top and middle) were obtained on a uniform mesh with 32 768 triangles, whereas those displayed in Fig. 2 (bottom) used the same mesh refined twice in a uniform fashion. The latter solution serves as a reference for our comparisons and, in order to reduce numerical diffusion, employs the FORCE (see (13)) instead of the Lax–Friedrichs flux. Since all external boundaries use land boundary conditions, the wave is reflected as illustrated in Fig. 2 (right) which correspond to $t = 3$ s. For better comparability, we limit the free surface elevation for the non-adaptive cases and show limited and unlimited results for the adaptive ones.

A comparison between the p-adaptive solutions with different indicators is shown in Fig. 3(a) for $t = 0.1$ s, in Fig. 3(b) for $t = 1$ s, and in Fig. 3(c) for $t = 3$ s, respectively. For indicator JE, a threshold value of 0.0003 led to the best results. For indicator GRE, in the unlimited case, the threshold for the free surface elevation of 0.8 and for the velocities of 1.7, and, in the limited case, the threshold for the free surface elevation of 0.2 and for the velocities of 0.5 produced the best solution. Our new indicator avoids over- and undershoots while not suffering from excessive numerical diffusion such as shown by the JE indicator with and without limiting. The GRE indicator produces a reasonable solution if limiting is active; otherwise, it is very diffusive and leads to over- and undershoots.

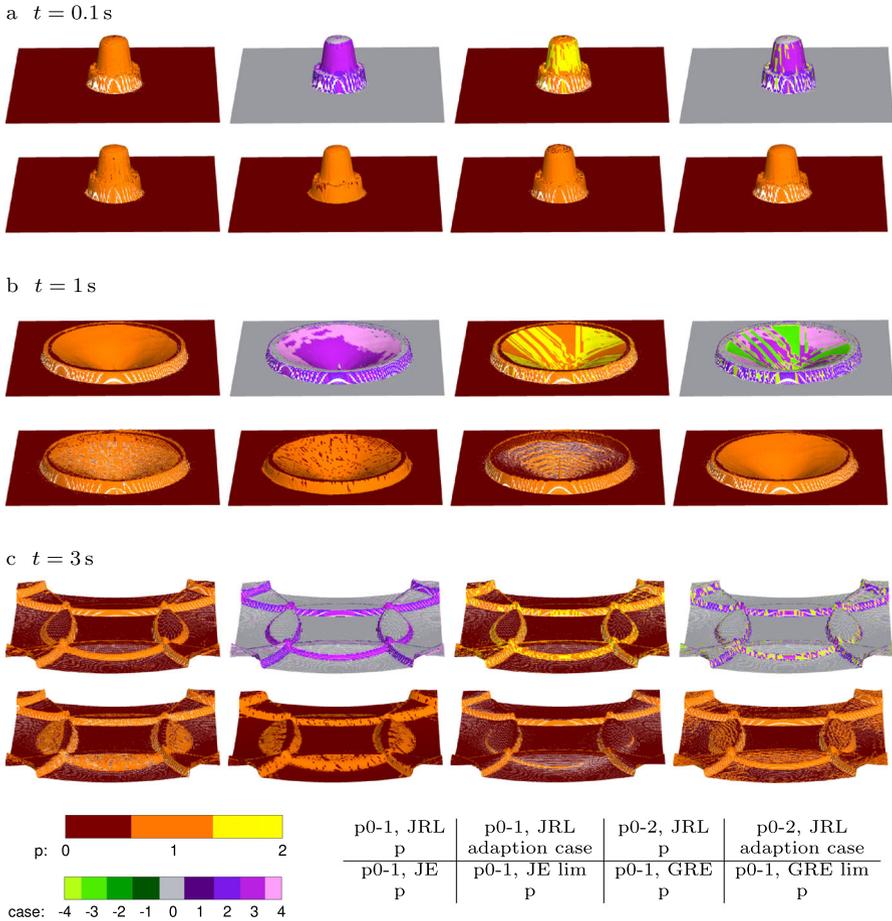


Fig. 3 Radial dam break: Free surface elevation at different time levels. Adaption range: Constant-linear (p0-1) or constant-quadratic adaptive (p0-2). Indicator: JRL, JE, JE limited, GRE, GRE limited. Local approximation order $p \in \{0, 1, 2\}$. Adaption case: see Fig. 1

These findings are further corroborated by the difference plots for the free surface elevation $\xi - \xi_{ref}$ shown in Fig. 4 and the L^1 -errors with respect to the reference solution listed in Table 1 alongside the fraction of elements with a specific order and the number of degrees of freedom. For $t = 0.1$ s in the adaptive cases, more than 93% of the elements are approximated by order 0, and only between 35% and 40% of the degrees of freedom of the linear approximation are needed to reach comparable L^1 -errors. For $t = 1$ s, in order to reach a comparable L^1 -error, one needs up to approx. 61% of degrees of freedom of the uniformly linear approximation for p0-1 and up to approx. 82% of degrees of freedom for p0-2. Here, a significant amount of p1 elements is necessary to represent the curvature correctly. For $t = 3$ s, the distribution of the approximation order is somewhat more dependent on the resolution, i.e., for higher resolutions, the fraction of lower order elements is higher and vice versa. Yet even in

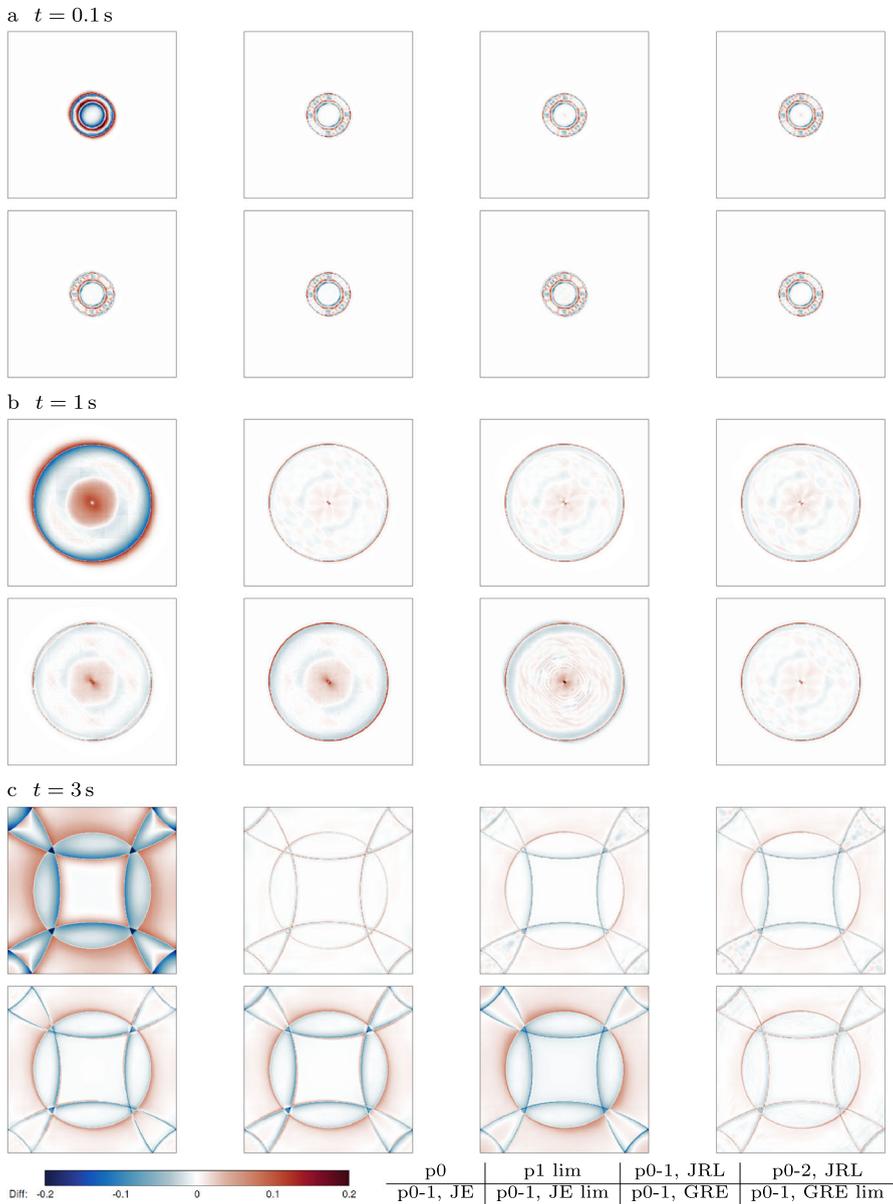


Fig. 4 Radial dam break: $\xi - \xi_{ref}$ difference plots at different time levels

the worst case only approx. 81% of the degrees of freedom compared to the uniform higher order approximation are needed. The highest resolution solution suffers from excessive numerical diffusion for orders 0 and 1 which can be remedied for order 1 by using the FORCE flux.

Table 1 Radial dam break: L^1 -error for the free surface elevation, fraction of elements of each discretization order, number of degrees of freedom (dof), and time step used throughout the simulation. Color coding indicates the quality/efficiency (green = good, red = poor)

Time in s	# Elements	Approximation order	L^1 -error	Order fraction			# dof	Time step size
				0	1	2		
0.1	8 192	0	1.79E-01	1.000	0.000	0.000	24 576	0.01
		1 lim	8.26E-02	0.000	1.000	0.000	73 728	0.005
		2 lim	9.64E-02	0.000	0.000	1.000	147 456	0.00125
		0-1 JRL	8.19E-02	0.938	0.062	0.000	27 624	0.005
		0-2 JRL	8.08E-02	0.938	0.042	0.020	29 124	0.005
	32 768	0	1.24E-01	1.000	0.000	0.000	98 304	0.005
		1 lim	4.02E-02	0.000	1.000	0.000	294 912	0.0025
		2 lim	4.64E-02	0.000	0.000	1.000	589 824	0.00625
		0-1 JRL	3.95E-02	0.945	0.055	0.000	109 104	0.0025
		0-1 JE	4.55E-02	0.939	0.061	0.000	110 292	0.0025
		0-1 JE_lim	4.19E-02	0.940	0.060	0.000	110 064	0.0025
		0-1 GRE	4.32E-02	0.948	0.052	0.000	108 492	0.0025
		0-1 GRE_lim	4.04E-02	0.936	0.064	0.000	110 820	0.0025
		0-2 JRL	3.90E-02	0.945	0.036	0.018	114 468	0.0025
		131 072	0	1.15E-01	1.000	0.000	0.000	393 216
	1 lim_force		1.41E-02	0.000	1.000	0.000	1 179 648	0.00125
	2 lim		1.94E-02	0.000	0.000	1.000	2 359 296	0.0003125
	0-1 JRL		1.60E-02	0.964	0.036	0.000	421 848	0.001
	0-2 JRL		1.58E-02	0.962	0.027	0.012	437 016	0.001
	1	8 192	0	7.01E-01	1.000	0.000	0.000	24 576
1 lim			1.49E-01	0.000	1.000	0.000	73 728	
2 lim			1.69E-01	0.000	0.000	1.000	147 456	
0-1 JRL			1.34E-01	0.585	0.415	0.000	44 988	
0-2 JRL			1.31E-01	0.580	0.213	0.207	60 510	
32 768		0	4.39E-01	1.000	0.000	0.000	98 304	
		1 lim	7.19E-02	0.000	1.000	0.000	294 912	
		2 lim	7.68E-02	0.000	0.000	1.000	589 824	
		0-1 JRL	7.97E-02	0.643	0.357	0.000	168 456	
		0-1 JE	1.03E-01	0.652	0.348	0.000	166 776	
		0-1 JE_lim	1.36E-01	0.676	0.324	0.000	161 976	
		0-1 GRE	1.31E-01	0.871	0.129	0.000	123 732	
		0-1 GRE_lim	8.05E-02	0.617	0.383	0.000	173 568	
		0-2 JRL	7.55E-02	0.641	0.191	0.168	218 472	
		131 072	0	1.45E+00	1.000	0.000	0.000	393 216
1 lim_force			2.47E-02	0.000	1.000	0.000	1 179 648	
2 lim			3.39E-02	0.000	0.000	1.000	2 359 296	
0-1 JRL			6.43E-02	0.779	0.221	0.000	567 168	
0-2 JRL			6.39E-02	0.782	0.133	0.085	664 134	
3		8 192	0	1.29E+00	1.000	0.000	0.000	24 576
	1 lim		2.16E-01	0.000	1.000	0.000	73 728	
	2 lim		3.18E-01	0.000	0.000	1.000	147 456	
	0-1 JRL		2.37E-01	0.292	0.708	0.000	59 400	
	0-2 JRL		2.26E-01	0.286	0.458	0.255	78 492	
	32 768	0	8.49E-01	1.000	0.000	0.000	98 304	
		1 lim	1.04E-01	0.000	1.000	0.000	294 912	
		2 lim	1.67E-01	0.000	0.000	1.000	589 824	
		0-1 JRL	1.43E-01	0.770	0.230	0.000	143 532	
		0-1 JE	2.86E-01	0.682	0.318	0.000	160 812	
		0-1 JE_lim	3.07E-01	0.678	0.322	0.000	161 565	
		0-1 GRE	3.63E-01	0.916	0.084	0.000	114 732	
		0-1 GRE_lim	1.51E-01	0.590	0.410	0.000	178 860	
		0-2 JRL	1.41E-01	0.759	0.177	0.065	164 760	
		131 072	0	2.03E+00	1.000	0.000	0.000	393 216
	1 lim_force		3.68E-02	0.000	1.000	0.000	1 179 648	
	2 lim		1.46E-01	0.000	0.000	1.000	2 359 296	
	0-1 JRL		1.27E-01	0.907	0.093	0.000	466 668	
	0-2 JRL		1.30E-01	0.902	0.077	0.021	498 808	

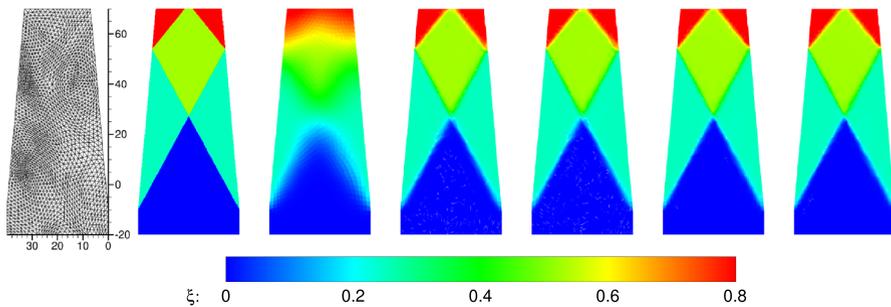


Fig. 5 Supercritical flow in a constricted channel: Free surface elevation. Left to right: Block-structured mesh with 3 584 elements, exact solution projected onto a mesh with approx. 230 000 elements, constant (p_0), limited linear (p_1), limited quadratic (p_2), constant-linear adaptive (p_0-1), constant-quadratic adaptive (p_0-2) solution with indicator JRL

4.2 Supercritical flow in a constricted channel

In our next benchmark, we consider a supercritical flow in a constricted channel (constriction angle of 5 degrees) with constant bathymetry based on the configuration proposed in Zienkiewicz and Ortiz (1995). Flow is induced through the inflow (bottom) boundary and there is no flow through the left and right boundaries, whereas the radiation boundary conditions are specified at the outflow (upper) boundary. Denoting by u_i and H_i the axial velocity and water depth at the inlet, respectively, the inlet Froude number Fr defined by $Fr = u_i / \sqrt{g H_i}$ is chosen equal to 2.5 corresponding to a supercritical regime.

For better comparability and to avoid instabilities in the p_2 approximation, we limit the free surface elevation and velocity components using the same limiting factor α_e (15) for the non-adaptive cases and plot limited and unlimited results for the adaptive ones. Fig. 5, shows our (topologically) block-structured grid (Zint et al. 2019) with 3 584 elements along with the steady-state solution for different approximations. Whereas a piecewise constant DG approximation is very diffusive, the limited linear and quadratic solutions look much better. Thanks to the adaption scheme, the constant-linear and the constant-quadratic solutions using indicator JRL do not suffer from over- or undershoots while representing the jumps very accurately without introducing too much diffusion.

The robustness and accuracy of our indicator are illustrated in Fig. 6, which details the local approximation order and the used adaption case (cf. Fig. 1). As expected, the constant plateaus are approximated by order 0 while higher orders are only activated in the vicinity of the discontinuities. When comparing the JE and GRE indicators, the unlimited simulations produce large under- and overshoots in jump regions whereas the limited JE and GRE indicators demonstrate acceptable performance, even though the results become somewhat diffusive. For JE, a threshold of 0.005 led to the best results. For GRE, the threshold for the free surface elevation of 0.02 and for the velocity of 0.05 appears to be optimal.

To quantify the performance of different adaptivity indicators, Fig. 7 displays the difference plots for the free surface elevation $\xi - \xi_{exact}$ using the exact solution (this

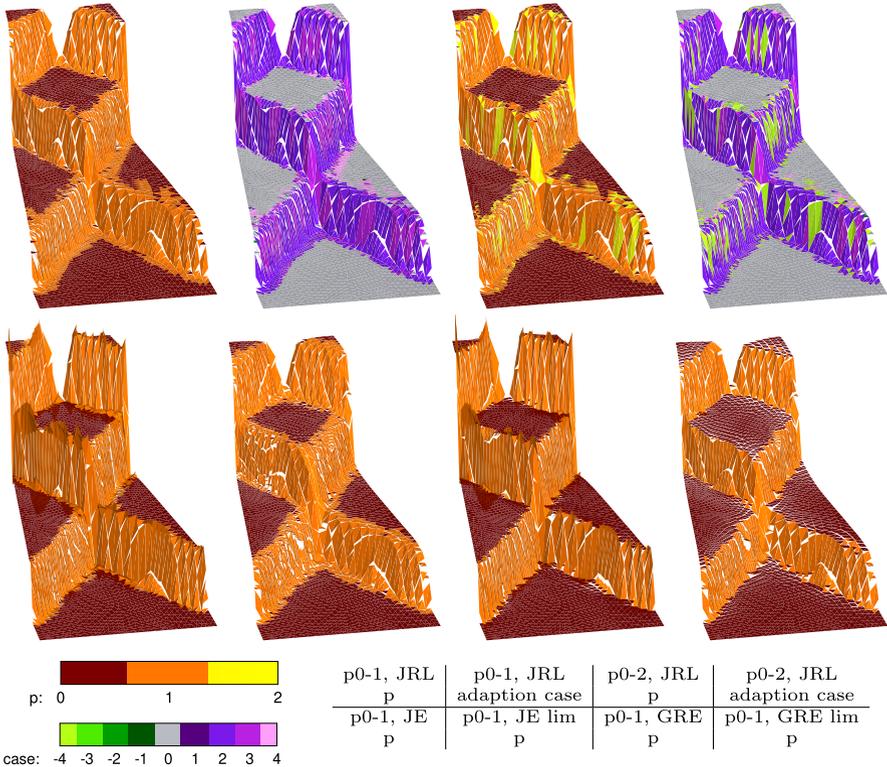


Fig. 6 Supercritical flow in a constricted channel. Free surface elevation. Adaption range: Constant-linear (p0-1) or constant-quadratic adaptive (p0-2). Indicator: JRL, JE, JE limited, GRE, GRE limited. Local approximation order $p \in \{0, 1, 2\}$. Adaption case: see Fig. 1. The z-axis is scaled by the factor 50

problem can be solved analytically, see, for example, (Ippen 1951) projected on a high resolution mesh with approx. 230 000 elements as the reference. The constant solution is clearly diffusive, the limited linear and quadratic solutions look reasonable. The darker colors in the unlimited constant-linear solutions indicate over- and undershoots which are absent in the fully limited adaptive solutions and in our adaption scheme with built-in liming. In Table 2, the L^1 -errors with respect to the exact solution are listed next to the fraction of elements with a given approximation order and the total number of degrees of freedom. It can be seen that between 64 % and 82 % of the elements use a constant approximation, and the p-adaptive solutions only need between 45 % and 58 % of the degrees of freedom of the uniformly linear approximation for p0-1 and approx. 63 % for p0-2.

4.3 Circular hydraulic jump

The last test example used to study our adaptivity indicator considers an instationary circular hydraulic jump based on the setup presented in Ketcheson and Quezada de

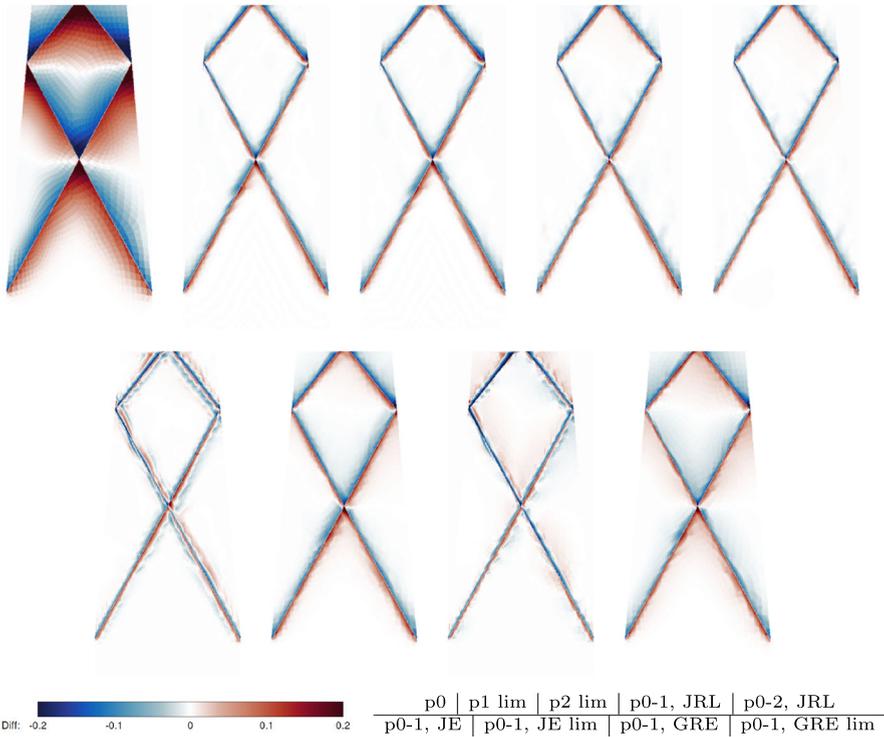


Fig. 7 Supercritical flow in a constricted channel: $\xi - \xi_{exact}$ difference plots using the exact solution projected to a mesh with approx. 230 000 elements as the reference

Table 2 Supercritical flow in a constricted channel: L^1 -error for the free surface elevation, fraction of elements of each discretization order, number of degrees of freedom, and time step used throughout the simulation. Color coding indicates the quality/efficiency (green = good, red = poor)

Approximation order	L^1 -error	Order fraction			# Degrees of freedom	Time step size
		0	1	2		
0	1.52e+02	1.000	0.000	0.000	10 752	0.2
1 lim	3.40E+01	0.000	1.000	0.000	32 256	0.1
2 lim	3.37E+01	0.000	0.000	1.000	64 512	0.05
0-1 JRL	3.08E+01	0.645	0.355	0.000	18 390	0.1
0-1 JE	2.98E+01	0.652	0.348	0.000	18 228	0.1
0-1 JE_lim	4.48E+01	0.630	0.370	0.000	18 702	0.1
0-1 GRE	2.96E+01	0.809	0.191	0.000	14 850	0.1
0-1 GRE_lim	4.97E+01	0.817	0.183	0.000	14 682	0.1
0-2 JRL	3.04E+01	0.654	0.283	0.063	20 211	0.1

Luna (2022). The domain geometry is a quarter annulus with $r_0 \in (0.1, 1)$ (cf. Fig. 8 (left)) and a constant bathymetry $h_b = 0.1$. The solution to this problem contains constant plateaus, smooth non-constant regions, and discontinuities. There is a constant inflow at $r_0 = 0.1$, that is, $\hat{\xi}(r_0 = 0.1, t) = 0.2$, $\hat{q}_n(r_0 = 0.1, t) = -0.75$ and

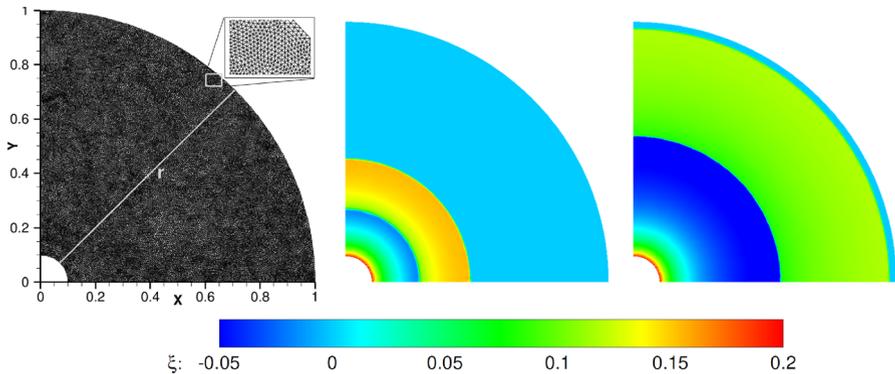


Fig. 8 Hydraulic jump: Computational mesh with the line ‘r’ indicating the position of the line plot in Fig. 11 (left), piecewise constant (p0) reference solution on the twice refined mesh for $t = 0.5$ s (middle), and for $t = 1.3$ s (right)

$\hat{q}_\tau(r_0 = 0.1, t) = 0$. Differing from the original setup, we prescribe a land boundary condition at $r_0 = 1$ causing a reflection. The initial conditions are $\xi(r_0, 0) = 0$ and $q(r_0, 0) = \mathbf{0}$. The gravity acceleration g is set to 1.

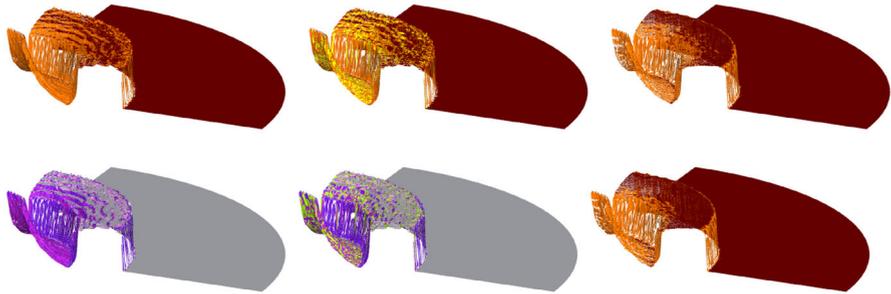
The employed block-structured grid (Zint et al. 2019) consists of 48 000 triangles, and r in Fig. 8 (left) indicates the line along which the free surface elevation is plotted in Fig. 11. In the middle and right panels of Fig. 8, the p0 high-resolution solution at $t = 0.5$ s and $t = 1.3$ s on a mesh with 768 000 elements is displayed, which serves as a reference in the following evaluation.

To avoid instabilities at the inflow, we limit the free surface elevation and velocity components with the same limiting factor α_e (15) for all cases except for the JRL indicator. The JRL indicator automatically reduces the local approximation order on critical elements at the inflow boundary to zero making limiting unnecessary there. The JE and the GRE indicators fail to detect the need for the lowest approximation order on critical elements therefore only the limited solution is stable.

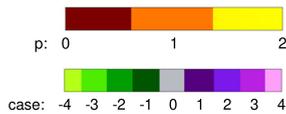
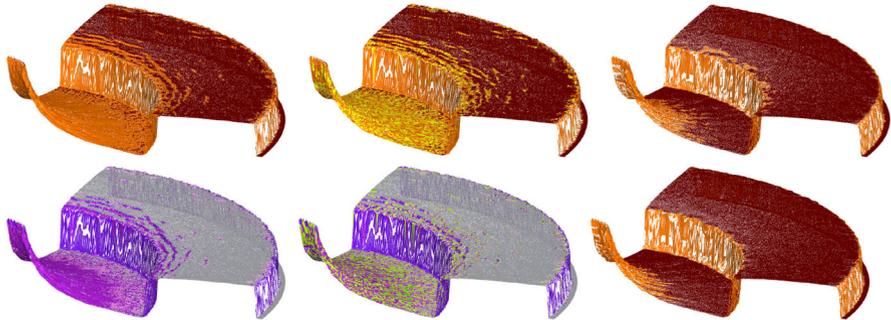
The p-adaptive solution with different indicators is shown for $t = 0.5$ s (Fig. 9a) and $t = 1.3$ s (Fig. 9b), respectively. For indicator JE, a threshold of 0.00001 led to the best results. For indicator GRE, the threshold for the free surface elevation of 0.6 and for the velocities of 1.0 resulted in the best simulation outcome.

The difference plots for $\xi - \xi_{ref}$ in Fig. 10 confirm that our indicator is better at capturing jumps than the two other indicators as pointed out by lighter colors in the corresponding regions. Furthermore, a linear approximation for the drop-off after the inflow is correctly identified. The L^1 -errors with respect to the reference solution are shown in Table 3 along with the fraction of elements with a given approximation order. All adaptive cases approximately match the L^1 -errors of the uniformly linear solution for $t = 0.5$ s while using fewer than 44 % of the degrees of freedom for p0-1 and approx. 50 % for p0-2. The L^1 -errors of the adaptive solutions (except for the JE indicator) are even lower for $t = 1.3$ s when compared to the uniformly linear approximation and utilize up to approx. 60 % of degrees of freedom for the p0-1 approximation and approx. 78 % for the p0-2 approximation.

a $t = 0.5\text{ s}$



b $t = 1.3\text{ s}$



p0-1, JRL p	p0-2, JRL p	JE lim p
p0-1, JRL adaption case	p0-2, JRL adaption case	p0-1, GRE lim p

Fig. 9 Hydraulic jump: Free surface elevation at different time levels. Adaption range: Constant-linear (p0-1) or constant-quadratic adaptive (p0-2). Indicator: JRL, JE limited, GRE limited. Local approximation order $p \in \{0, 1, 2\}$. Adaption case: see Fig. 1

Figure 11 displays the free surface elevation along the line 'r' highlighted in Fig. 8 (left). Noting that $h_b=0.1$, it shows a very good agreement with Fig. 7 from (Ketcheson and Quezada de Luna 2022). The zoom-ins also demonstrate a better accuracy of the approximations using the JRL indicator in the jump regions compared to those relying on the JE or GRE indicator with limiting.

5 Conclusions and outlook

We presented a parameter-free adaptivity indicator and validated its ability to identify resolved and underresolved areas of the computational domain while distinguishing between smooth and non-smooth solution regions and adapting the local approximation order accordingly. On a range of numerical examples, our new indicator demonstrated comparable or better solution quality than that of a limited uniformly

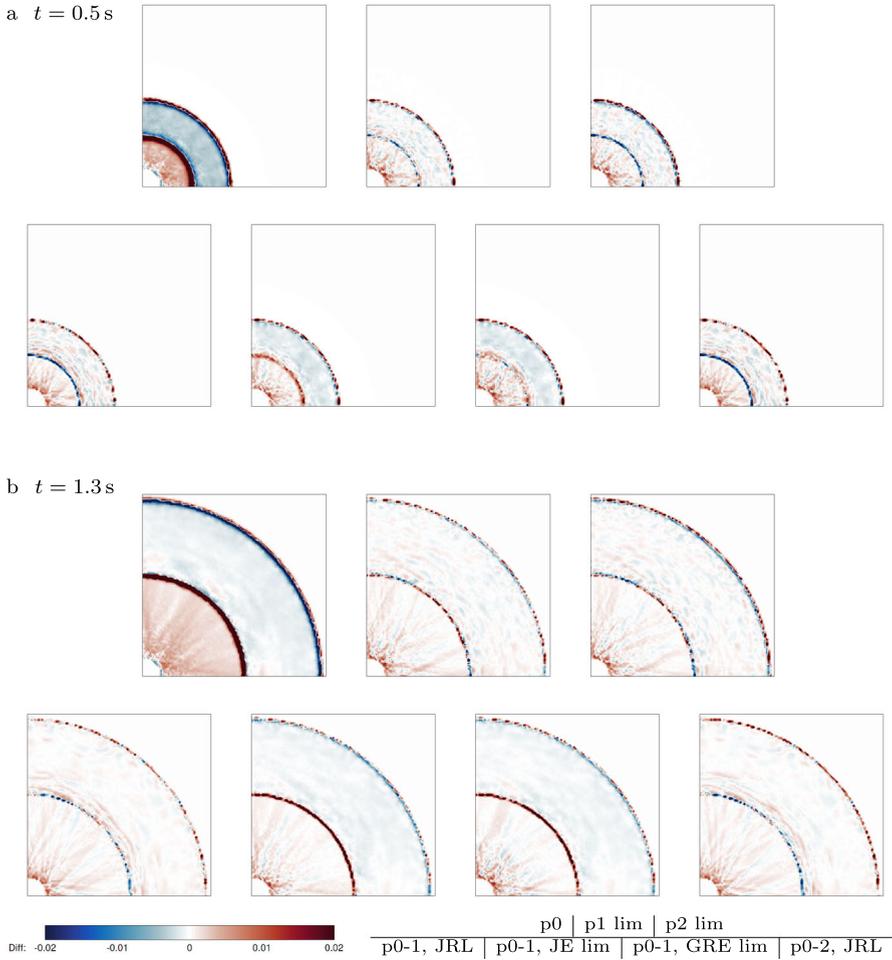


Fig. 10 Hydraulic jump: $\xi - \xi_{ref}$ difference plots at different time levels

linear approximation. These results were achieved without any calibration parameters while using significantly fewer degrees of freedom. A comparison to two further indicators suggests that our new scheme achieves a good balance between solution quality and number of degrees of freedom used.

This work is an integral part of our ongoing effort to produce a p-adaptive DG solver for the SWE capable of efficient utilization of massively parallel and heterogeneous computing architectures (Faghih-Naini et al. 2023). It bundles a number of other numerical and computational techniques such as a novel block-structured grid generator for ocean domains (Zint et al. 2019, 2022; Faghih-Naini et al. 2022) and the code generation framework (Lengauer et al. 2014; Schmitt et al. 2014a; Kuckuk and Köstler 2016). Our future research goals include adding the FPGA support—already available for the unstructured-mesh version of the model (Kenter et al. 2021; Faj et al.

Table 3 Hydraulic jump: L^1 -error for the free surface elevation, fraction of elements of each discretization order, number of degrees of freedom, and time step used throughout the simulation. Color coding indicates the quality/efficiency (green = good, red = poor)

Time in s	Approximation order	L^1 -error	Order fraction			# Degrees of freedom	Time step size
			0	1	2		
0.5	0	1.56E-03	1.000	0.000	0.000	144 000	0.001
	1 lim	4.50E-04	0.000	1.000	0.000	432 000	0.0005
	2 lim	5.63E-04	0.000	0.000	1.000	864 000	0.0002
	0-1 JRL	4.73E-04	0.846	0.154	0.000	188 262	0.0001
	0-1 JE_lim	4.73E-04	0.873	0.127	0.000	180 696	0.00005
	0-1 GRE_lim	4.87E-04	0.897	0.103	0.000	173 694	0.0001
	0-2 JRL	4.99E-04	0.839	0.100	0.061	216 636	0.00005
1.3	0	3.07E-03	1.000	0.000	0.000	144 000	
	1 lim	8.67E-04	0.000	1.000	0.000	432 000	
	2 lim	1.02E-03	0.000	0.000	1.000	864 000	
	0-1 JRL	6.72E-04	0.593	0.407	0.000	261 132	
	0-1 JE_lim	1.06E-03	0.822	0.178	0.000	195 180	
	0-1 GRE_lim	1.01E-03	0.870	0.130	0.000	181 398	
	0-2 JRL	7.82E-04	0.580	0.248	0.172	338 988	

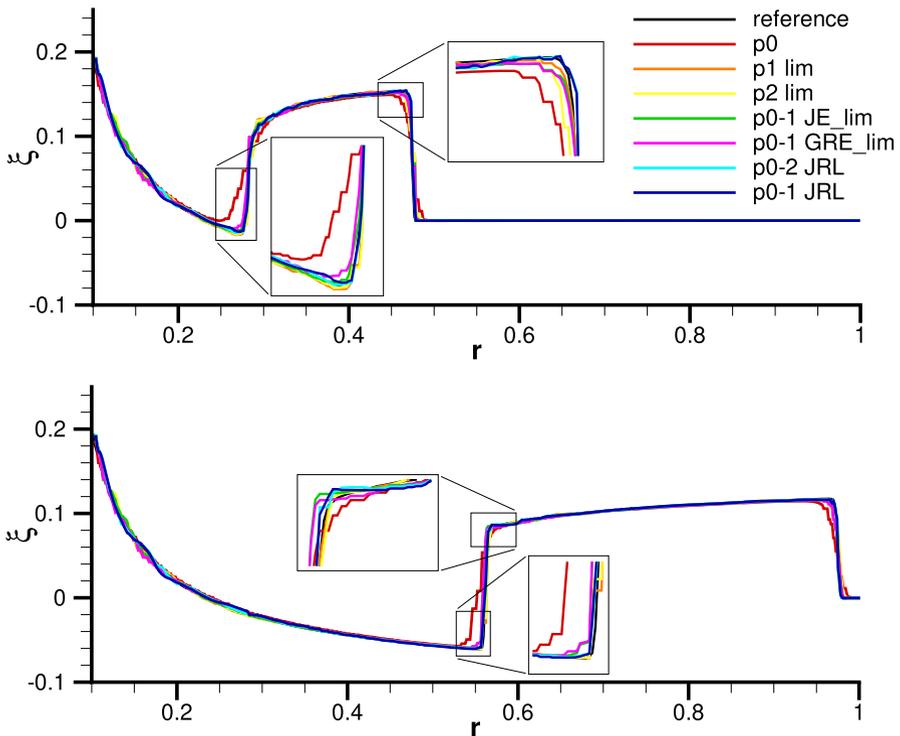


Fig. 11 Hydraulic jump: Free surface elevation along the line $(0.1,0.1)$ to $(1/\sqrt{2},1/\sqrt{2})$ (cf. the gray line in Fig. 8 marked as 'r') for $t = 0.5$ s (top) and $t = 1.3$ s (bottom)

2023)—and implementing semi-implicit time stepping schemes based on the hierarchical scale separation (HSS) approach (Aizinger et al. 2015; Schütz and Aizinger 2017) related to the p-multigrid method.

Acknowledgements The authors gratefully acknowledge fruitful discussions with Dr. Sebastian Kuckuk and Dr. Hennes Hajduk and the support by NHR@FAU. This work was supported in part by the German Research Foundation (DFG) through grant AI 117/6-1 ‘Performance optimized software strategies for unstructured-mesh applications in ocean modeling’.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aizinger, V.: A geometry independent slope limiter for the discontinuous Galerkin method. In: Krause, E., Shokin, Y., Resch, M., Kröner, D., Shokina, N. (eds.) *Computational Science and High Performance Computing IV, Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pp. 207–217. Springer, Berlin (2011)
- Aizinger, V., Kosik, A., Kuzmin, D., Reuter, B.: Anisotropic slope limiting for discontinuous Galerkin methods. *Int. J. Numer. Methods Fluids* **84**(9), 543–565 (2017). <https://doi.org/10.1002/flid.4360>
- Aizinger, V., Kuzmin, D., Korouš, L.: Scale separation in fast hierarchical solvers for discontinuous Galerkin methods. *Appl. Math. Comput.* **266**, 838–849 (2015). <https://doi.org/10.1016/j.amc.2015.05.047>
- Burbeau, A., Sagaut, P.: A dynamic p-adaptive discontinuous Galerkin method for viscous flow with shocks. *Comput. Fluids* **34**, 401–417 (2005). <https://doi.org/10.1016/j.compfluid.2003.04.002>
- Eskilsson, C.: An hp-adaptive discontinuous Galerkin method for shallow water flows. *Int. J. Numer. Methods Fluids* **67**, 1605–1623 (2011). <https://doi.org/10.1002/flid.2434>
- Faghih-Naini, S., Kuckuk, S., Aizinger, V., Zint, D., Grosso, R., Köstler, H.: Quadrature-free discontinuous Galerkin method with code generation features for shallow water equations on automatically generated block-structured meshes. *Adv. Water Resour.* **138**, 103552 (2020). <https://doi.org/10.1016/j.advwatres.2020.103552>
- Faghih-Naini, S., Kuckuk, S., Zint, D., Kemmler, S., Aizinger, V.: Discontinuous Galerkin method for the shallow water equations on complex domains using masked block-structured grids (2022)
- Faghih-Naini, S., Aizinger, V., Kuckuk, S., Angersbach, R., Köstler, H.: p-adaptive discontinuous Galerkin method for the shallow water equations on heterogeneous computing architectures (2023)
- Faj, J., Plessl, C., Kenter Tobias Faghih-Naini, S., Aizinger, V.: Scalable Multi-FPGA Design of a Discontinuous Galerkin Shallow-Water Model on Unstructured Meshes. In: submitted to Proceedings of The 2023 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA ’23). Association for Computing Machinery, New York, NY, USA (2023)
- Frank, F., Reuter, B., Aizinger, V., Knabner, P.: FESTUNG: a MATLAB/GNU Octave toolbox for the discontinuous Galerkin method, Part I: diffusion operator. *Comput. Math. Appl.* **70**(1), 11–46 (2015). <https://doi.org/10.1016/j.camwa.2015.04.013>

- Gottlieb, S., Shu, C.W.: Total variation diminishing runge-kutta schemes. *Math. Comp.* **67**(221), 73–85 (1998). <https://doi.org/10.1090/S0025-5718-98-00913-2>
- Hajduk, H.: Monolithic convex limiting in discontinuous Galerkin discretizations of hyperbolic conservation laws. *Comput. Math. Appl.* **87**, 120–138 (2021). <https://doi.org/10.1016/j.camwa.2021.02.012>
- Hajduk, H., Hodges, B.R., Aizinger, V., Reuter, B.: Locally Filtered Transport for computational efficiency in multi-component advection-reaction models. *Environ. Modell. Softw.* **102**, 185–198 (2018). <https://doi.org/10.1016/j.envsoft.2018.01.003>
- Hajduk, H., Kuzmin, D., Aizinger, V.: New directional vector limiters for discontinuous Galerkin methods. *J. Comput. Phys.* **384**, 308–325 (2019). <https://doi.org/10.1016/j.jcp.2019.01.032>
- Ippen, A.: High-velocity flow in open channels: a symposium: mechanics of supercritical flow. *Trans. Am. Soc. Civ. Eng.* **116**(1), 268–295 (1951)
- Kenter, T., Shambhu, A., Faghih-Naini, S., Aizinger, V.: Algorithm-hardware co-design of a discontinuous Galerkin shallow-water model for a dataflow architecture on FPGA. In: Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '21. Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3468267.3470617>
- Ketcheson, D.I., Quezada de Luna, M.: Numerical simulation and entropy dissipative cure of the carbuncle instability for the shallow water circular hydraulic jump. *Int. J. Numer. Methods Fluids* (2022). <https://doi.org/10.1002/fld.5070>
- Krivodonova, L., Xin, J., Remacle, J.F., Chevaugeon, N., Flaherty, J.: Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Appl. Numer. Math.* **48**(3), 323–338 (2004). <https://doi.org/10.1016/j.apnum.2003.11.002>. Workshop on Innovative Time Integrators for PDEs
- Kubatko, E.J., Bunya, S., Dawson, C., Westerink, J.J.: Dynamic p-adaptive Runge-Kutta discontinuous Galerkin methods for the shallow water equations. *Comput. Methods Appl. Mech. Eng.* **198**(21), 1766–1774 (2009). *Advances in Simulation-Based Engineering Sciences—Honoring J. Tinsley Oden.* <https://doi.org/10.1016/j.cma.2009.01.007>
- Kuckuk, S., Köstler, H.: Automatic generation of massively parallel codes from ExaSlang. *Computation* **4**(3), 27:1–27:20 (2016). <https://doi.org/10.3390/computation4030027>. Special Issue on High Performance Computing (HPC) Software Design
- Kuzmin, D.: A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *J. Comput. Appl. Math.* **233**(12), 3077–3085 (2010). <https://doi.org/10.1016/j.cam.2009.05.028>. *Finite Element Methods in Engineering and Science (FEMTEC 2009)*
- Kuzmin, D., Schieweck, F.: A parameter-free smoothness indicator for high-resolution finite element schemes. *Open Math.* **11**(8), 1478–1488 (2013). <https://doi.org/10.2478/s11533-013-0254-4>
- Lengauer, C., Apel, S., Bolten, M., Chiba, S., Rüde, U., Teich, J., Größlinger, A., Hannig, F., Köstler, H., Claus, L., Grebhahn, A., Groth, S., Kronawitter, S., Kuckuk, S., Rittich, H., Schmitt, C., Schmitt, J.: Exastencils: advanced multigrid solver generation. In: Bungartz, H.J., Reiz, S., Uekermann, B., Neumann, P., Nagel, W.E. (eds.) *Software for Exascale Computing—SPPEXA 2016–2019*, pp. 405–452. Springer International Publishing, Cham (2020)
- Lengauer, C., Apel, S., Bolten, M., Größlinger, A., Hannig, F., Köstler, H., Rüde, U., Teich, J., Grebhahn, A., Kronawitter, S., Kuckuk, S., Rittich, H., Schmitt, C.: ExaStencils: Advanced stencil-code engineering. In: *Euro-Par 2014: Parallel Processing Workshops, Lecture Notes in Computer Science*, vol. 8806, pp. 553–564. Springer (2014). https://doi.org/10.1007/978-3-319-14313-2_47
- LeVeque, R.J.: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press (2002). <https://doi.org/10.1017/CBO9780511791253>
- Michoski, C., Mirabito, C., Dawson, C., Wirasaet, D., Kubatko, E., Westerink, J.: Adaptive hierarchical transformations for dynamically p-enriched slope-limiting over discontinuous Galerkin systems of generalized equations. *J. Comput. Phys.* **230**(22), 8028–8056 (2011). <https://doi.org/10.1016/j.jcp.2011.07.009>
- Naddei, F., de la Llave Plata, M., Couaillier, V., Coquel, F.: A comparison of refinement indicators for p-adaptive simulations of steady and unsteady flows using discontinuous Galerkin methods. *J. Comput. Phys.* **376**, 508–533 (2019). <https://doi.org/10.1016/j.jcp.2018.09.045>
- Remacle, J.F., Flaherty, J.E., Shephard, M.S.: An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Rev.* **45**(1), 53–72 (2003). <https://doi.org/10.1137/S00361445023830>

- Reuter, B., Aizinger, V., Wieland, M., Frank, F., Knabner, P.: FESTUNG: a MATLAB/GNU Octave toolbox for the discontinuous Galerkin method, Part II: advection operator and slope limiting. *Comput. Math. Appl.* **72**(7), 1896–1925 (2016). <https://doi.org/10.1016/j.camwa.2016.08.006>
- Schmitt, C., Kuckuk, S., Hannig, F., Köstler, H., Teich, J.: ExaSlang: A domain-specific language for highly scalable multigrid solvers. In: *Proceedings of the Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing (WOLFHPC)*, pp. 42–51. IEEE Computer Society (2014)
- Schmitt, C., Kuckuk, S., Hannig, F., Köstler, H., Teich, J.: Exaslang: A domain-specific language for highly scalable multigrid solvers. In: *2014 Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing*, pp. 42–51 (2014). <https://doi.org/10.1109/WOLFHPC.2014.11>
- Schütz, J., Aizinger, V.: A hierarchical scale separation approach for the hybridized discontinuous Galerkin method. *J. Comput. Appl. Math.* **317**, 500–509 (2017). <https://doi.org/10.1016/j.cam.2016.12.018>
- Toro, E.F., Hidalgo, A., Dumbser, M.: FORCE schemes on unstructured meshes i: conservative hyperbolic systems. *J. Comput. Phys.* **228**(9), 3368–3389 (2009). <https://doi.org/10.1016/j.jcp.2009.01.025>
- Tumolo, G., Bonaventura, L., Restelli, M.: A semi-implicit, semi-lagrangian, p-adaptive discontinuous Galerkin method for the shallow water equations. *J. Comput. Phys.* **232**(1), 46–67 (2013). <https://doi.org/10.1016/j.jcp.2012.06.006>
- Zienkiewicz, O.C., Ortiz, P.: A split-characteristic based finite element model for the shallow water equations. *Int. J. Numer. Methods Fluids* **20**(8–9), 1061–1080 (1995). <https://doi.org/10.1002/flid.1650200823>
- Zint, D., Grosso, R., Aizinger, V., Faghih-Naini, S., Kuckuk, S., Köstler, H.: Automatic Generation of Load-Balancing-Aware Block-Structured Grids for Complex Ocean Domains. In: Robinson, T., Moxey, D., Tomov, V.Z. (eds.) *Proceedings of the 2022 SIAM International Meshing Roundtable*. Zenodo (2022). <https://doi.org/10.5281/zenodo.6562440>
- Zint, D., Grosso, R., Aizinger, V., Köstler, H.: Generation of block structured grids on complex domains for high performance simulation. *Comput. Math. Math. Phys.* **59**(12), 2108–2123 (2019). <https://doi.org/10.1134/S0965542519120182>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.