Discontinuous Galerkin Methods for Dispersive Shallow Water Models in Closed Basins: Spurious eddies and their removal using curved boundary methods

D.T. Steinmoeller[a,*], M. Stastna[a], K.G. Lamb[a]

[a] *200 University Ave. W., Waterloo, ON, N2L 3G1*

Abstract

Discontinuous Galerkin methods offer a promising methodology for treating nearly hyperbolic systems such as dispersion-modified shallow water equations in complicated basins. Use of straight-edged triangular elements can lead to the generation of spurious eddies when wave fronts propagate around sharp, re-entrant obstacles such as headlands. While these eddies may be removed by adding strong artificial dissipation (e.g., eddy viscosity), for nearly inviscid simulations that focus on wave phenomena this approach is not reasonable. We demonstrate that the moderate order Discontinuous Galerkin methodology may be extended to curved triangular elements provided that the integral formulations are computed with high-order quadrature and cubature rules. Simulations with the new technique do not exhibit spurious eddy generation in idealized complex domains or real-world basins as exemplified by Pinehurst Lake, Alberta, Canada.

Keywords: Water waves, Wave dispersion, Mathematical models, Fluid dynamics, Boussinesq equations, Shallow water equations

1. Introduction

Understanding the physical processes in lakes is of fundamental importance in a vast array of applications, ranging from water quality management to bio-geochemical cycling. Numerical modelling is perhaps the best

*Corresponding author at: Department of Applied Mathematics, University of Waterloo, Waterloo, ON, Canada. N2L 3G1. Tel.: +1 519 888 4567 ext. 32588; fax: +1 519 746 4319.

Email address: dsteinmo@uwaterloo.ca (D.T. Steinmoeller)

tool available for improving and extending the current understanding of lake dynamics. The solution of a weakly non-hydrostatic single-layer model in periodic and annular domains with the high-order Fourier and Chebyshev pseudospectral methods has been recently explored by Steinmoeller et al. (2012, 2013). The methods developed in these works allow for the numerical modeling of circular or channel–like basins. While circular basins and channel–like basins may seem to be esoteric cases, they form a well studied class of problems in physical limnology dating back over a century (Thomson, 1872; Stocker and Imberger, 2003). High order numerical methods for such basins allow the robustness of classical solutions to be explored without the uncertainty associated with the inherent dissipation in many low order methods. This, in turn, allows for a rational set of hypotheses to be formulated for subsequent testing against field data. Regardless of these advances, the need for methods that can capture more general geometries has been clear for some time and specific reasons for this need were identified in Steinmoeller et al. (2013).

Since a lake's coastal boundary generally specifies a physical domain with complex/irregular boundaries, the pseudospectral methods presented in Steinmoeller et al. (2012, 2013) are not sufficient for modelling real-world lakes. To represent more general geometries, we turn to the Discontinuous Galerkin finite element method (DG-FEM) as a high-order alternative to the low-order finite volume and finite element methods that are commonly used for irregular geometries. The results presented in this manuscript typically use local polynomial orders between $N = 4$ and 8. The methods are thus high-order in contrast to traditional finite element methods that typically use piece-wise linear or quadratic basis functions. See for instance Walkley (1999), who solved a Boussinesq-type system with a low-order finite element method (FEM). Low-order numerical work with Boussinesq-type systems applied in process studies has been carried out by Tomasson and Melville (1992), Brandt et al. (1997) and de la Fuente et al. (2008), for example.

It is worth stressing that the high-order DG-FEM is not the same as the spectral element method (SEM) (see Karniadakis and Sherwin (2005)) that represents the high-order extension of the traditional FEM. Both FEM and SEM are continuous Galkerin formulations which require C^0 continuity at element interfaces. Although DG-FEM and SEM both use a high-order orthogonal polynomial basis, the DG-FEM only imposes continuity in a weak sense through the specification of a numerical flux function at element edges in order to allow for stable advection schemes (Cockburn

and Shu, 1989; Cockburn et al., 1990; Hesthaven and Warburton, 2008). The requirement of C^0 continuity in the SEM means that the method is not ideal for advection problems since an upwind-type scheme cannot be formulated to account for the preferred direction of propagation of information (Hesthaven and Warburton, 2008). This shortcoming can lead to situations where Gibbs oscillations are trapped at element interfaces, as has been illustrated for the spectral element ocean model by Levin et al. (2006). However, it should be noted that modern treatments of FEM/SEM seek to overcome this shortcoming for advective problems by considering stabilization techniques such as the SUPG (streamline upwind/Petrov–Galerkin) method (Hughes, 1987) as well as the class of entropy-based viscosity methods (Nazarov and Popov, 2014 (submitted)). An alternative to the purely discontinuous approach has been recently proposed in the form of the hybridizable discontinuous Galerkin method that imposes strong continuity only in the edge-normal flux component (Rhebergen and Cockburn, 2012).

The specification of an upwind-biased numerical flux is usually furnished through the well-established theory of approximate Riemann solvers that are commonly used in the formulation of finite volume methods in order to propagate information between finite volume cells (see Toro (1999) for an overview). It is for this reason that DG-FEM with piece-wise constant basis functions (order $N = 0$) is identical to the low-order finite volume method, as explained by Hesthaven and Warburton (2008).

In the following sections, we follow the techniques and developments for nodal DG-FEM presented by Hesthaven and Warburton (2008), building upon their techniques as necessary. We briefly explain the basic nodal DG-FEM formulation as the spatial discretization method for both hyperbolic and elliptic systems and the corresponding reduction to local operators in the context of a one-layer dispersive shallow water model. Following this, a comparison with the pseudospectral methods of Steinmoeller et al. (2012, 2013) is carried out as a means of validating the numerical scheme presented here and illustrating the resolution characteristics of the DG-FEM at varying polynomial orders. The necessity of curvilinear elements for general situations is illustrated by the formation of singular/spurious flow features that emerge because of the piece-wise linear representation of the boundary. It is then explained how the nodal DG-FEM method should be augmented with high-order cubature and quadrature integration rules to deal with the non-constant mapping Jacobians introduced by curvilinear elements.

In section 2 we describe the basic numerical methods. Because standard

techniques are used many of the details are left to appendices. The modifications based on the use of curvilinear elements are described in section 3. Results are presented in section 4 which includes simulations on internal waves in a real-world lake: Pinehurst Lake, Alberta, Canada.

2. Methods

2.1. Governing Equations

The governing equations for a single-layer reduced gravity model with non-hydrostatic corrections to the hydrostatic pressure (de la Fuente et al., 2008; Steinmoeller et al., 2012, 2013) are

$$\frac{\partial h}{\partial t} + \nabla \cdot (h\mathbf{u}) = 0\,, \tag{1}$$

$$\frac{\partial (uh)}{\partial t} + \nabla \cdot ((uh)\mathbf{u}) = -g'h\frac{\partial \eta}{\partial x} + fvh + \frac{H^2}{6}\frac{\partial}{\partial x}\left(\nabla \cdot \frac{\partial (\mathbf{u}h)}{\partial t}\right)\,, \tag{2}$$

$$\frac{\partial (vh)}{\partial t} + \nabla \cdot ((vh)\mathbf{u}) = -g'h\frac{\partial \eta}{\partial y} - fuh + \frac{H^2}{6}\frac{\partial}{\partial y}\left(\nabla \cdot \frac{\partial (\mathbf{u}h)}{\partial t}\right)\,, \tag{3}$$

where $\mathbf{u}(x,y,t) = (u(x,y,t), v(x,y,t))$ is the velocity field, $h(x,y,t) = H(x,y) + \eta(x,y,t)$ is the total depth with H representing the undisturbed depth, and η is the free surface displacement. The constants g' and f are the reduced gravitational acceleration and the Coriolis frequency, respectively. These equations differ from the traditional hyperbolic shallow water model through the addition of the dispersive terms $\frac{H^2}{6}\nabla(\nabla \cdot (\mathbf{u}h)_t)$. The above system was developed for surface waves by Peregrine (1967) and used by Brandt et al. (1997) in their study of internal waves in the Strait of Messina.

An efficient scheme for evolving the dispersive terms can be obtained by introducing the scalar auxiliary variable

$$z = \nabla \cdot (\mathbf{u}h)_t\,, \tag{4}$$

in order to reduce the momentum equations (2)–(3) to a hyperbolic problem of the shallow water type plus the elliptic problem

$$\nabla \cdot \left(\frac{H^2}{6}\nabla z\right) - z = -\nabla \cdot \mathbf{a}\,, \tag{5}$$

that is referred to as a *wave continuity* equation by Eskilsson and Sherwin (2005). Here

$$\mathbf{a} = \begin{pmatrix} -\nabla \cdot ((uh)\mathbf{u}) - gh\eta_x + fvh \\ -\nabla \cdot ((vh)\mathbf{u}) - gh\eta_y - fuh \end{pmatrix}\,. \tag{6}$$

We have neglected bottom and surface stresses in equations (1)–(3) since their inclusion into the numerical scheme is conceptually easy and contributes little to the discussion. We have also chosen to focus on the case of a single fluid layer of constant density, since the inclusion of multiple layers adds considerable complexity to numerical formulations that rely on approximate solutions to the corresponding nonlinear Riemann problem. See Mandli (2011) for a discussion on the two-layer Riemann problem in the context of the finite volume method.

2.2. Time-Stepping Technique

The time-stepping technique applied to the DG-FEM discretized version of the one-layer model closely follows the 'scalar approach' used for the pseudospectral discretization in Steinmoeller et al. (2012, 2013) and for the DG-FEM method in Eskilsson and Sherwin (2005) where splitting is applied such that advective and source terms are time-stepped first, followed by the dispersive terms. As in the works mentioned above, the time-stepping approach relies heavily on the 'method of lines' (see Leveque (2007)) where temporal and spatial discretizations are treated completely separately and a layer of abstraction may exist between these two discretizations.

Neglecting the dispersive terms for the time-being since they are not a part of the first splitting step, the method of lines can be applied by noticing that once the DG-FEM integral form has been written purely in terms of matrix operators (see (A.6), appendix A), we recover the system of ordinary differential equations

$$\frac{d\mathbf{Q}}{dt} = \mathcal{R}(\mathbf{Q}) \ , \tag{7}$$

where $\mathbf{Q} = (h, hu, hv)^{\mathsf{T}}$ is the vector of unknowns and $\mathcal{R}$ is the DG spatial discretization operator for the advection, Coriolis, and bathymetry source terms. We have followed Eskilsson and Sherwin (2005) and time-discretized (7) beginning at time-level $t_n = n\Delta t$ using the three-stage third-order strong stability preserving Runge-Kutta (SSP-RK) method (Hesthaven and Warburton, 2008). Modal filtering is applied to the spatial discretization operator $\mathcal{R}$ after each stage to help tame aliasing and nonlinearity-driven instabilities as explained in Section 2.4. The choice of SSP-RK time-stepper here is not a unique one, and we have mainly used it here since it offers third-order accuracy and allows for a simple adaptive time-stepping scheme. That is, Δt can be adjusted after each time-step without changing the coefficients of the scheme. The SSP-RK methods have gained favour in the DG-FEM literature (see Hesthaven and Warburton (2008); Cockburn

and Shu (1989)) since they guarantee no oscillations are introduced as a result of time-stepping for problems involving discontinuities and shocks. Such features are not of concern for the equations under consideration here due to the dispersive terms in the momentum equations which results in a greater degree of smoothness than in a purely hyperbolic system.

As mentioned above, the next step in the 'scalar approach' is to solve the wave continuity equation. Its continuous form is given by (5) with z and $\mathbf{a}$ replaced by $z^\dagger$ and $\mathbf{a}^\dagger$, respectively. The spatially-discretized vector $\mathbf{a}^\dagger$ can be computed quite simply by evaluating the contributions to $\mathcal{R}(\mathbf{Q}^\dagger)$ from the momentum equations. The auxiliary variable $\mathbf{z}^\dagger$ is then computed by inverting the matrix representation of the 'symmetric interior penalty discontinuous Galerkin' (SIP-DG) formulation of the Helmholtz operator (G.4). The momentum equations are finally updated via

$$(h\mathbf{u})^{n+1} = (h\mathbf{u})^\dagger + \gamma \Delta t \nabla z^{n+1} \,, \tag{8}$$

where the DG-FEM discretization of the source terms involving nonlinear products with gradients of known quantities is discussed in Appendix F, and $\gamma = H^2/6$. Hence, the vector of unknowns at time t_{n+1} is updated via $\mathbf{Q}^{n+1} = (h^\dagger, hu^{n+1}, hv^{n+1})^\mathsf{T}$.

For the simulations considered in this manuscript the SIP-DG sparse matrix can be factored using the sparse LU-decomposition (Steinmoeller et al., 2012), so that the factors may be re-used at each time-step for fast inversions, and we have side-stepped the issue of using an iterative solver such as GMRES that is required for the pseudospectral methods of Steinmoeller et al. (2012, 2013). At high resolutions, a linear iterative method will certainly be necessary due to memory restrictions prohibiting the storage of the semi-dense LU factors.

2.3. DG-FEM Spatial Discretization

The DG-FEM method is primarily suited to solving strictly hyperbolic equation sets. To accommodate the method as much as possible, the equations are re-cast in the form of a conservation law plus source terms:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{B} + \mathbf{C} + \mathbf{N} \,, \tag{9}$$

with

$$\mathbf{Q} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \; \mathbf{F} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \; \mathbf{G} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}. \tag{10}$$

The terms

$$\mathbf{B} = gh \begin{pmatrix} 0 \\ \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial y} \end{pmatrix}, \ \mathbf{C} = f \begin{pmatrix} 0 \\ vh \\ -uh \end{pmatrix}, \ \mathbf{N} = \frac{H^2}{6} \begin{pmatrix} 0 \\ \frac{\partial z}{\partial x} \\ \frac{\partial z}{\partial y} \end{pmatrix}, \tag{11}$$

are the bed slopes, Coriolis terms, and the dispersive terms, respectively. As explained in Section 2.1, the variable $z = \nabla \cdot (\mathbf{u}h)_t$ is governed by the Helmholtz problem (5). In the case of the traditional non-rotating hydrostatic shallow water model with a flat bottom, the right-hand side of (9) vanishes. Details are provided in Appendix A

The Helmholtz problem (5) cannot be treated using the standard DG-FEM methodology since it does not correspond to a hyperbolic problem. However, a DG-FEM type discretization is possible by re-writing the second-order equations as a first-order system and resorting to penalty methods to appropriately enforce continuity between elements. To do this we introduce the auxiliary variable

$$\mathbf{q} = (q_x, q_y) = \sqrt{\gamma} \nabla z \ , \tag{12}$$

yielding the system

$$\nabla \cdot (\sqrt{\gamma}\mathbf{q}) - z = -\nabla \cdot \mathbf{a} \ , \tag{13}$$

$$q_x = \sqrt{\gamma} \frac{\partial z}{\partial x} \ , \tag{14}$$

$$q_y = \sqrt{\gamma} \frac{\partial z}{\partial y} \ . \tag{15}$$

Details of the methodology employed to solve this system are given in the Appendices.

2.4. Filter Stabilization of Aliasing-driven Instabilities

The governing equations do not possess any viscosity terms and thus lack any physical energy dissipation mechanism. As a result, the quadratic nonlinearity terms can cause energy to accumulate at the small scales in an unphysical manner. Additionally, aliasing errors that occur due to the "pointwise product" treatment of the nonlinear terms can drive weak numerical instabilities that can destroy the numerical solutions as explained in Hesthaven and Warburton (2008).

In light of these issues, filtering is implemented as a procedure to both dissipate energy as it accumulates at the small scales and to prevent aliasing

errors from driving weak instabilities. A low-pass filter of the form

$$\sigma(n) = \begin{cases} 1, & 0 \leq n < N_{crit} \\ \exp\left(-\alpha \left(\frac{n-N_{crit}}{N-N_{crit}}\right)^s\right), & N_{crit} \leq n \leq N \end{cases} \tag{16}$$

is applied in the space of the modal coefficients to the solution fields after each time-step. Typical parameters used in the simulations presented in Section 4 are $N_{crit} = \lceil 0.65N \rceil$, $s = 4$, $\alpha = 18.4$, where N is the order of the highest-order modal basis polynomial. The parameters α, s, and N_c are tunable and, in general, their values must be determined through experimentation.

3. Curvilinear Elements

In addition to the solution singularities mentioned in Section 1, it is also known that the convergence rates of a high-order method may be limited to sub-optimal rates as a result of an inaccurate representation of the boundary. This was demonstrated in Hesthaven and Warburton (2008) who found poor convergence rates for the solution of Maxwell's equations on a circular domain with a piece-wise linear representation of the boundary. Dupont (2001) suggested that rounding singular corners is necessary to suppress poor polynomial behaviour resulting from the high-order DG-FEM in his inter-model comparison of the oceanic shallow water equations. It is thus apparent that a high-order method begs for a smooth and accurate representation of the boundary, and hence, deformed or curvilinear elements along the boundary will be necessary to achieve accurate solutions in general lake geometries with the high-order discontinuous Galerkin method.

3.1. Constructing coordinates systems for curvilinear elements

We have adopted the approach in Hesthaven and Warburton (2008) that avoids some of the difficulties and cumbersome work associated with explicitly constructing two-dimensional mapping functions for high-order curvilinear elements, e.g., explicitly calculating high-order "shape-functions". The technique discussed here generalizes well to elements with an arbitrary number of nodes and thus allows for the robust construction of high-order curvilinear elements. The method represents an extension of the technique used in Hesthaven and Warburton (2008) for circular boundaries, since we consider arbitrary domain boundaries represented by cubic splines.

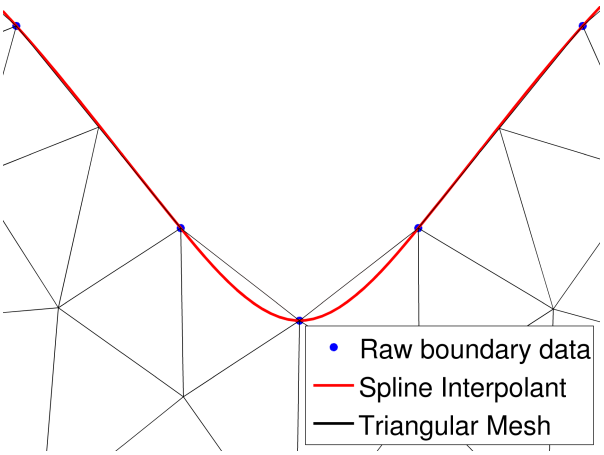

Figure 1: Illustration of straight-sided element mesh along with a smooth representation of the boundary, the spline interpolant, that will be used to produce deformed elements.

Assume we have generated a straight-sided finite element mesh that approximates the boundary in a piece-wise linear manner, and that we have a smooth representation of the boundary in a parameterized curve $\mathcal{C}$: $\mathbf{x}_b(t) = (x_b(t), y_b(t))$ parameterized by arc-length $0 \leq t \leq S$ (see Fig. 1). In practice, we have found taking $\mathcal{C}$ to be a parametric cubic-spline interpolant of the boundary to be a simple and effective choice. The algorithm for a particular element that is to be curved is as follows:

1. Adjust the straight-sided finite element mesh by moving the vertices (i.e., end-points only) of the straight-sided element's boundary edge e so that they lie exactly at points on $\mathcal{C}$, say $\mathbf{x}_b(t_1)$ and $\mathbf{x}_b(t_2)$.
2. Distribute the 1D Legendre-Gauss-Lobotto (LGL) nodes along the curved edge by arc-length using the parameterization $\mathbf{x}_b(t)$ for $t_1 \leq t \leq t_2$ to obtain new local coordinates along the curved edge, denoted $\mathbf{x}_{curved}(r,s)|_e$, where (r,s) are the coordinates of the reference triangle (see Fig. 2).
3. Calculate the deformation (displacement field) in moving from the edge nodes from the straight edge to the curve $\mathcal{C}$, i.e., $\mathbf{w}(r,s) = \mathbf{x}_{curved}(r,s)|_e - \mathbf{x}_{straight}(r,s)|_e$, also called the *warp* factor.
4. "Blend" the edge deformation to the interior nodes using Gordon–Hall blending (see below) to obtain new local coordinates for the whole element: $\mathbf{x}_{curved}(r,s) = \mathbf{x}_{straight}(r,s) + b(r,s)\mathbf{w}(r,s)$, where $b(r,s)$ is a *blending* function.

5. Compute local metric factors, i.e., x_r,y_r,x_s,y_s, and Jacobian $J = x_r y_s - x_s y_r$, numerically using the differentiation matrices on the reference element $\mathcal{D}_r$, $\mathcal{D}_s$.

The one point that requires further attention is how to choose a blending function $b(r, s)$ to appropriately "blend" the edge deformation on the element boundary to the interior of the element. To motivate our discussion, consider the simplistic one-dimensional case where two function values f_0 and f_1 are known at points x_0 and x_1 and we wish find a function $f(x)$ to interpolate to points inside the interval $[x_0, x_1]$. If, for additional simplicity, we assume $f_0 = 0$, we realize that the only way to interpolate to interior points with the information that we have is by the linear Lagrange interpolant $\ell_1(x) = (x - x_0)/(x_1 - x_0)$, i.e.,

$$f(x) = \left(\frac{x - x_0}{x_1 - x_0} \right) f_1 \ . \tag{17}$$

In a sense, we have found the appropriate blending function to be $\ell_1(x)$ since this function satisfies the desired properties: $\ell_1(x_1) = 1$, $\ell_1(x_0) = 0$.

Now consider the two-dimensional case where, for example, our edge deformation $\mathbf{w}(r, s)$ is known along the triangle edge corresponding to the line $r = -1$ for $-1 \leq s \leq 1$ on the reference element (Fig. 2). Clearly, we require the blending function to satisfy $b(-1, s) = 1$ since this is the only region where information is known. It also seems sensible that the effect of the edge-deformation would decay to zero at the opposite triangle edge that lies on the line $s = -r$, leading us to define the blending function as

$$b(r, s) = \left(\frac{s + r}{s - 1} \right) . \tag{18}$$

The one issue that remains is the apparent singularity at the point $(-1, 1)$. This point corresponds to a location where $\mathbf{w} = 0$ since it is a vertex of the finite element mesh that does not need to be deformed. Thus, we can simply apply the blending at nodal points not corresponding to the singular point in step 4 above.

While the "blending" procedure discussed is a straight-forward extension of linear Lagrange interpolation to two-dimensions, one subtle difference between Lagrange interpolation is that the two-dimensional blending function is chosen to be zero or one along entire *line segments*, and not at points in space. It is for this reason that the technique has been referred to as "transfinite interpolation" by Gordon and Hall (1973) since, in general, the data is being sampled over a continuum and not just at a finite set of points.

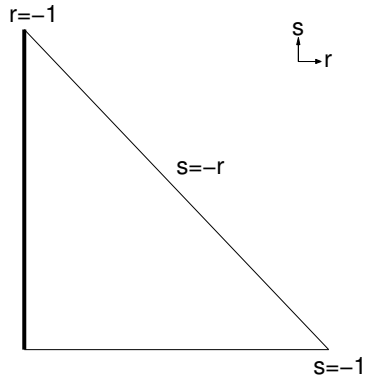

Figure 2: Diagram of the reference triangle and illustration of (r, s) coordinates.

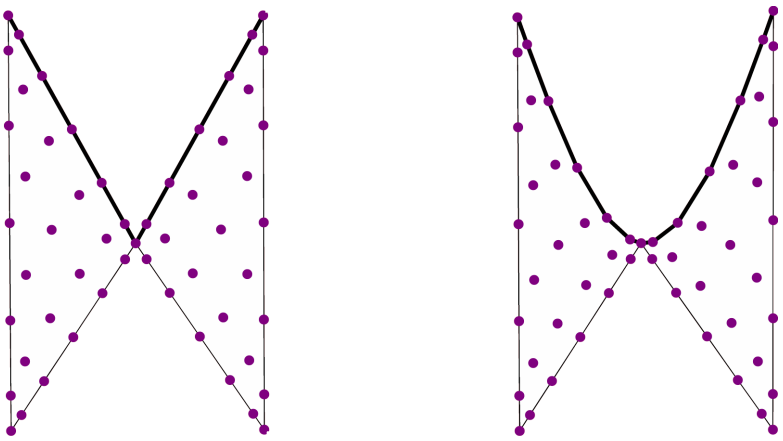

Figure 3: **Left**: A pair of elements before being deformed. **Right**: The same elements after being deformed to match the cubic-spline representation of the boundary with interior nodes re-distributed via Gordon–Hall blending.

3.2. Cubature and Quadrature Integration

The computationally inexpensive approach of evaluating the strong/weak form integrals using a nodal evaluation described in Hesthaven and Warburton (2008) relies heavily on the assumption that the Jacobian of the mapping from a particular element to the standard element is a constant, and may be brought outside of the integrals in the nodal DG-FEM formulation. This is not the case for curvilinear elements, and we must thus pay a computational price. Firstly, a separate mass matrix must be stored for each curvilinear element, thereby driving up computational storage costs.

Secondly, the Jacobians of the mappings used here are rational functions of the standard element's coordinates, and their product with the solution fields will in general lead to aliasing errors.

Nonlinearities involving rational functions cannot be de-aliased completely (as, for example, a quadratic nonlinearity could) since their polynomial representation would consist of a Taylor series with infinitely many terms. Nevertheless, a great deal of aliasing error can be removed by evaluating the integrals with cubature rules that are of higher order than the approximating polynomials. Here, 'cubature' refers to the higher-dimensional analogy to 1D quadrature rules. For polynomials of order N, we follow Hesthaven and Warburton (2008) and employ cubature rules of order $3(N+1)$. A general inner-product of two functions f and g is thus evaluated as

$$\int_{\mathbf{D}^k} fg \, d\mathbf{x} \approx \sum_{i=1}^{N_c} f(\mathbf{r}_i^c) g(\mathbf{r}_i^c) J_i^k w_i^c \,, \tag{19}$$

where J_i^k is the Jacobian of the mapping from the standard element D^k, and w_i^c are the cubature weights associated with cubature nodes $\{\mathbf{r_i^c}\}_{i=1}^{N_c}$. The cubature nodes and weights are provided by the symmetric rules of Wandzura and Xiao (2003) and implemented in MATLAB in Hesthaven and Warburton (2008).

The use of cubature integration makes the evaluation of the local mass and stiffness matrices more computationally expensive, since additional interpolation operations must be carried out to interpolate integrands to the cubature nodes. In particular, we define the $N_c \times N_p$ interpolation matrix $V_{ij}^c = \ell_j(\mathbf{r}_i^c)$ to interpolate functions defined at the polynomial interpolation nodes to the cubature nodes. The $N_p \times N_p$ mass matrix can then be found as follows

$$\mathcal{M}_{lm}^k = \int_{\mathbf{D}^k} \ell_l^k(\mathbf{x}) \ell_m^k(\mathbf{x}) \, d\mathbf{x} \tag{20}$$

$$\approx \sum_{i=1}^{N_c} \ell_l(\mathbf{r}_i^c) \ell_m(\mathbf{r}_i^c) w_i^c J_i^{k,c}. \tag{21}$$

Hence,

$$\mathcal{M}^k = (\mathcal{V}^c)^\mathsf{T} \mathcal{W}^k \mathcal{V}^c. \tag{22}$$

where $\mathcal{W}^k$ is the $N_c \times N_c$ diagonal matrix with entries $\mathcal{W}_{ii}^k = w_i^c J_i^{k,c}$. For

the local stiffness matrix,

$$\mathcal{S}^k_{x,nm} = \int_{\mathbf{D}^k} \ell^k_n(\mathbf{x}) \frac{\partial \ell^k_m}{\partial x}(\mathbf{x})\, d\mathbf{x} \tag{23}$$

we must invoke the chain rule to express the operators in terms of the $N_p \times N_p$ differentiation matrices on the reference triangle, $\mathcal{D}_r$ and $\mathcal{D}_s$, yielding

$$\mathcal{S}^k_x = (\mathcal{V}^c)^\mathsf{T} \mathcal{W}^k \left(\mathrm{diag}(r^k_x(\mathbf{r}^c_i))\, \mathcal{V}^c \mathcal{D}_r + \mathrm{diag}(s^k_x(\mathbf{r}^c_i))\, \mathcal{V}^c \mathcal{D}_s \right). \tag{24}$$

An identical argument gives

$$\mathcal{S}^k_y = (\mathcal{V}^c)^\mathsf{T} \mathcal{W}^k \left(\mathrm{diag}(r^k_y(\mathbf{r}^c_i))\, \mathcal{V}^c \mathcal{D}_r + \mathrm{diag}(s^k_y(\mathbf{r}^c_i))\, \mathcal{V}^c \mathcal{D}_s \right). \tag{25}$$

In addition to volume (two-dimensional) integrals, surface integral (element-coupling) terms must also be computed using Gaussian quadrature, with analogous two-dimensional interpolation operators used to evaluate the integrand at the appropriate quadrature points along an edge. We again follow Hesthaven and Warburton (2008) and use order $N_G = 2(N+1)$ Gaussian quadrature along the edges. Again, it should be stressed that this approach is more expensive since it requires an $(N_G + 1) \times N_p$ matrix (interpolation) operation along each edge, whereas the purely nodal approach simply requires the evaluation of $N + 1$ nodal values on each edge.

4. Results and Discussion

4.1. DG-FEM vs. Fourier–Chebyshev method in 2D

In this section, we validate our DG-FEM solver for the one-layer weakly non-hydrostatic model equations (1)–(3) against the Fourier–Chebyshev method presented in Steinmoeller et al. (2013) for the particular test-case in Figure 8 of that manuscript. The domain is an annulus with inner and outer radii of 1 and 8.435 km. An initial tilt of amplitude $a = 0.25H$ is released from rest, where the depth is $H = 12.8$ m. The Coriolis frequency is $f = 7.8828 \times 10^{-5}$ s^{-1} and the reduced gravity is $g' = 0.024525$ m s^{-2}.

Unlike the 1D comparison carried out in Steinmoeller et al. (2012), it should be noted that the comparison between the Fourier method and the DG methods at various orders here is not a "fair" one, since the number of degrees of freedom has not been held fixed in all cases. Here, the point is to illustrate that spectral-like resolution characteristics can become

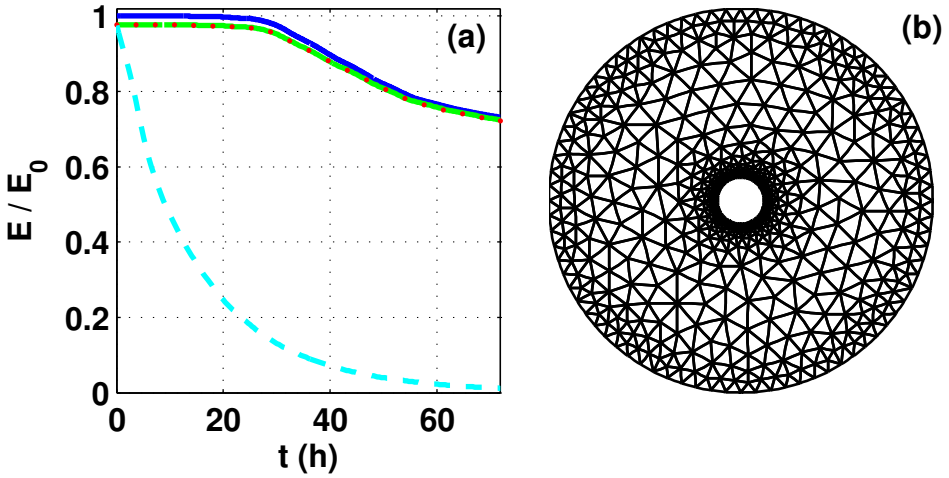

Figure 4: Panel **(a)**: Scaled domain-integrated total energy (E/E_0) time-series for the simulations in Fig. 5. The lines correspond to the Fourier–Chebyshev method with 256×1024 points (blue, solid), DG-FEM with $N = 8$ (green, solid), DG-FEM with $N = 4$ (red, dots), DG-FEM with $N = 1$ (cyan, dashed). Panel **(b)**: Finite element mesh with $K = 1330$ elements used in the DG-FEM simulations.

possible with increasing polynomial orders on a fixed finite element mesh triangulation. The mesh used here consists of $K = 1330$ triangular elements, and it was chosen to directly correspond with the annular-shaped lake of Steinmoeller et al. (2013). Curvilinear boundaries are not used for this case. The mesh, shown in Fig. 4**(b)** was generated with the `mesh2d` MATLAB algorithm that uses an adaptive Delaunay-based triangulation algorithm implemented using quadtrees. Modal filtering was applied using an exponential cut-off filter (16). Here, the cut-off polynomial order was set to $N_c = 3$ and the filter order was set to $s = 4$, and the same filtering parameters were used in both the $N = 4$ and $N = 8$ cases.

The energy characteristics of the various methods from Figure 5 are compared in Figure 4 **(a)** by plotting the total energy

$$E = \iint_\Omega \frac{1}{2} h(u^2 + v^2) + \frac{1}{2} g' \eta^2 \, dA \tag{26}$$

(scaled by its initial value) against time. We see that the initial energy is

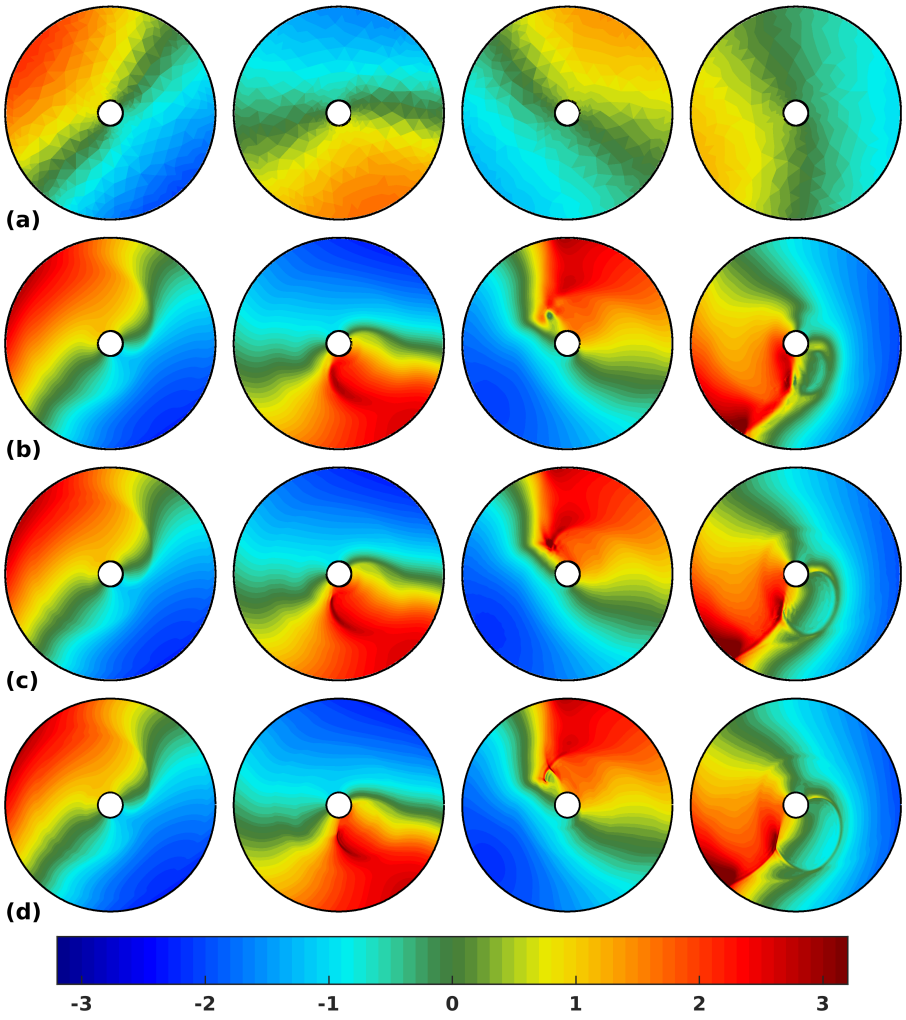

Figure 5: Comparison between the DG-FEM method at orders **(a)** $N = 1$, **(b)** $N = 4$, and **(c)** $N = 8$ to the Fourier–Chebyshev pseudospectral method (row **(d)**) for the simulation presented in Steinmoeller et al. (2013) with $N_r \times N_\theta = 256 \times 1024$ points. The number of elements in the DG simulations was $K = 1330$ in all cases. In all rows, snapshots of the η field are given at times (from left-to-right): $t = 7$ h, $t = 14$ h, $t = 20$ h, $t = 27$ h.

rapidly dissipated in the low-order $N = 1$ case. The $N = 4$ and $N = 8$ cases exhibit nearly identical energy profiles, however a more detailed view would reveal that the $N = 8$ line is slightly above the $N = 4$ line. As expected, the Fourier–Chebyshev method outperforms all of the DG methods with the least energy lost. Once again, we illustrate here the utility of pseudospectral methods as a benchmark numerical method. These results also validate the DG-FEM method since the details of the nonlinear wave fronts in Figure 5 are reasonably represented for orders $N \geq 4$, and the amount of numerical dissipation approaches that of the pseudospectral method for increasing N (see Figure 4**(a)**.

4.2. Spurious eddies in inviscid DG-FEM solutions

While exploring other geometries with the DG-FEM code, it was found that under certain conditions spurious eddies, corresponding to an unphysical production of vorticity, form in the domain near boundaries that protrude into the domain. This effect is illustrated in Figure 6 where our annular basin has been perturbed to include a peninsula. The DG-FEM solver with polynomial order $N = 4$ was initialized with the initial conditions used in Section 4.1. A numerical instability occurred shortly after $t = 27$ h, preventing further time-stepping, though the reason for the instability was evident earlier due to the sharp gradients visible near the tip of the peninsula in Figure 6.

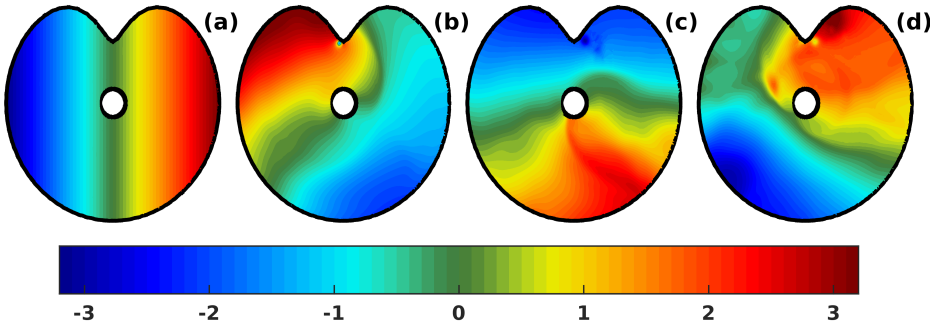

Figure 6: Snapshots of the η-field in the order $N = 4$ DG-FEM simulation of a rotating seiche on a perturbed circular domain with a re-entrant peninsula at **(a)** $t = 0$ h, **(b)** $t = 6.8$ h, **(c)** $t = 14.0$ h, **(d)** $t = 20.9$ h. Note the apparent separation eddies visible near the peninsula in panels **(b)**–**(d)**.

The eddies bear a striking resemblance to boundary-layer separation eddies that would occur due to flow past an obstacle in viscous flow (see Kundu and Cohen (2008)). However, since our model equations do not contain any viscous terms, the formation of a viscous boundary-layer is not possible and hence boundary-layer separation should not be possible. These spurious eddies are thus artifacts, and appear to coincide with the presence of a sharp re-entrant corner. Even in cases where the actual boundary is smooth, re-entrant corners at the element-scale may result as a consequence of the piece-wise linear representation of the boundary assumed in mesh generation. Although these artifacts are spurious in the sense that inviscid flow around an obstacle should not separate, from a theoretical stand-point they should be expected. Below, we explain why this is the case and propose methods for remedying the situation.

As is well known from potential flow theory (Kundu and Cohen, 2008), the velocity of a potential flow at a corner is infinite if the wall-angle is greater than 180° and is zero for wall-angles less than 180°. In the former case the pressure goes to $-\infty$ at the corner. Thus, near re-entrant corners the numerical solution should be *expected* to be poorly behaved since the exact potential flow solution is also. Although the velocity derivatives do not exist at corners less than 180°, this does not appear to be an issue for the numerical solution.

The "spurious eddies" encountered in simulations begin as very steep free-surface depressions that form due to the low pressures that form at the re-entrant corner. In real flow around a corner, the region of adverse pressure gradient would cause the flow to separate from the corner resulting in the formation of eddies due to vorticity produced in the viscous boundary layer. In the DG-FEM simulations discussed, the observed eddies are a result of the local modal filtering that attempts to stabilize the pressure singularity by diffusing it away from the boundary, taking over the role of viscosity in realistic flows. This effect of the filter was discovered by turning off the filter and observing singular growth at the corner that led to numerical blow-up with no eddy introduced. It was also found that spurious eddy generation is more prominent in simulations where nonlinear effects are non-negligible. The fact that a standard filter coupled with the presence of re-entrant corners will typically lead to spurious eddies is a dangerous feature of the numerical model, since a modeller may be led to believe that these eddies are physical, when in fact they are the result of the filter's action on a part of the solution that is singular. For instance, in Zhang et al. (2012), spurious eddies due to a limiter are presented as physical for the situation of supersonic compressible flow past an equilateral triangle. Despite the effort of filtering, it has been found that this singular behaviour can still lead to numerical blow-up. Use of curvilinear boundary elements eliminates this problem as illustrated below.

4.3. Internal Rotating Seiche Simulation using curvilinear elements

We now consider the same simulation shown in Section 4.2 where a circular basin has been perturbed to include a peninsula. The difference here is that we employ the developments on curvilinear elements described in the above sections along with polynomial order $N = 8$. All boundary elements have been deformed such that their boundary edges conform to a cubic spline interpolant of the boundary.

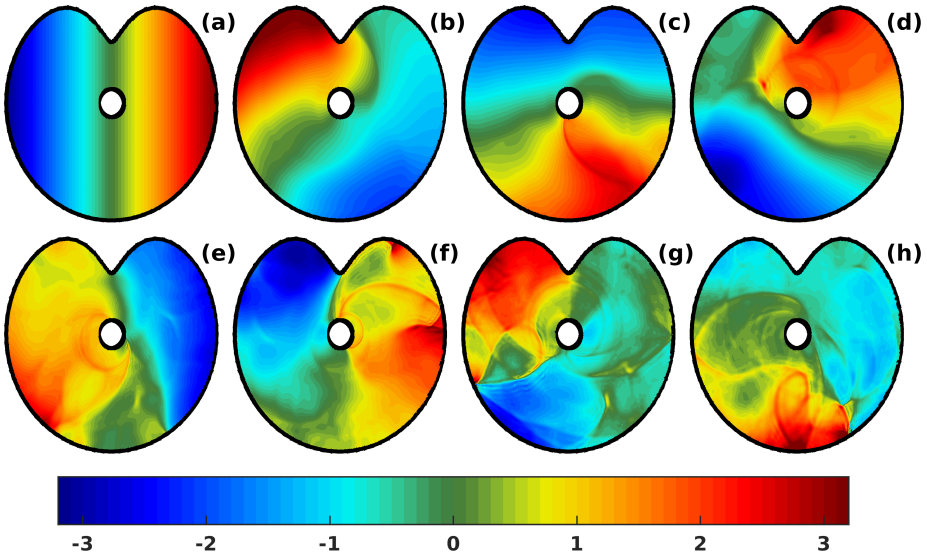

Figure 7: Panels **(a)**–**(d)**: Like Figure 6 but with curvilinear elements along the boundary. The other panels correspond to the later times **(e)** $t = 28.1$ h, **(f)** $t = 34.9$ h, **(g)** $t = 42.1$ h, **(h)** $t = 49.0$ h.

The results of the simulation are shown in Figure 7. In addition to finding that the simulation is apparently long-term stable, unlike in the straight-sided case, we also note that the spurious eddies associated with the sharp re-entrant corner have been suppressed since the peninsula is now represented in a geometrically smooth way.

It is important to note that although the spurious eddies have been suppressed, the region of the flow at the tip of the peninsula still represents a geometric feature where a strong adverse pressure gradient must appear in order to decelerate flow around the obstacle. Indeed, an adverse η-gradient appears between $t = 2.5$ h and $t = 5.4$ h (not shown).

As discussed in Section 4.2, in real-world flows it is certainly reasonable to expect the flow around the peninsula to separate and generate eddies due to viscous boundary-layer effects, but since we have not included a physical model for such processes we are left in the somewhat precarious situation in which we demand the flow to remain 'attached' to the peninsula in all cases.

4.4. Internal Rotating Seiche Simulation in a real-world lake

In this section, we provide proof-of-concept that the high-order DG-FEM methodology of this manuscript can be applied to real-world lake geometries involving irregular coastlines. Bathymetry data at a resolution of 50 m for the mid-sized Pinehurst Lake, Alberta has been obtained from the Alberta Geological Survey website `http://www.ags.gov.ab.ca/`. The raw data consists of a cartesian grid with 216×245 data points containing both land and water measurements. A plot of the 50 m bathymetry data is shown in Figure 8 where land values have been set to zero.

A parametric representation of the coastline was obtained using the data returned by MATLAB's `contour` function used to obtain the zero-depth contour and is shown in Figure 8**(b)**. It was found that finite element meshes generated from the raw data contained $O(10,000)$ elements and possessed poor mesh quality (i.e., large aspect ratio triangles and large element size gradients) since the raw 0-depth contour is far from smooth. A smoothed piece of coastline is shown in Figure 8**(c)** with corresponding $N = 6$ curved finite element mesh in panel **(d)**. The smoothed coastline was found by convolving the two-dimensional bathymetry data with the 2D cardinal B-spline 16 times and sub-sampling the result to a 200 m resolution data set. A piece-wise cubic spline interpolant of the coastline was then constructed so that boundary elements could be deformed using the techniques explained in Section 3. The straight-sided finite element mesh, that is later deformed by our DG-FEM solver, was constructed using the open-source `gmsh` software of Geuzaine and Remacle (2009) that was found to give better quality meshes than `mesh2d` in this case. Finally, the depth-profile $H(x, y)$ was linearly interpolated from the Cartesian data to our unstructured DG-FEM mesh for use during simulations. The depth-profile was capped at a minimum depth of 6 m to avoid dry states that would drive instabilities in the DG-FEM solver.

Here, the reduced gravity is $g' = (\Delta\rho/\rho_0)g = 0.024525$ ms^{-2}, where $(\Delta\rho/\rho_0) = 0.0025$. The Coriolis parameter was taken to be $f = 1.1863 \times 10^{-4}$ s^{-1}, corresponding to the 54.65° N latitude of Pinehurst Lake. Results of an $N = 6$ DG-FEM simulation from an initial east-west interfacial tilt taken to increase linearly from $\eta = 0$ to $\eta = 2.5$ m are shown in Figure 9 that illustrates the evolving density interface at fixed-time snapshots with the initial condition plotted in panel **(a)**. Since the relative amplitude of the initial condition compared to the depth is, on average, not as large as in previous simulations in this manuscript, nonlinear effects are expected

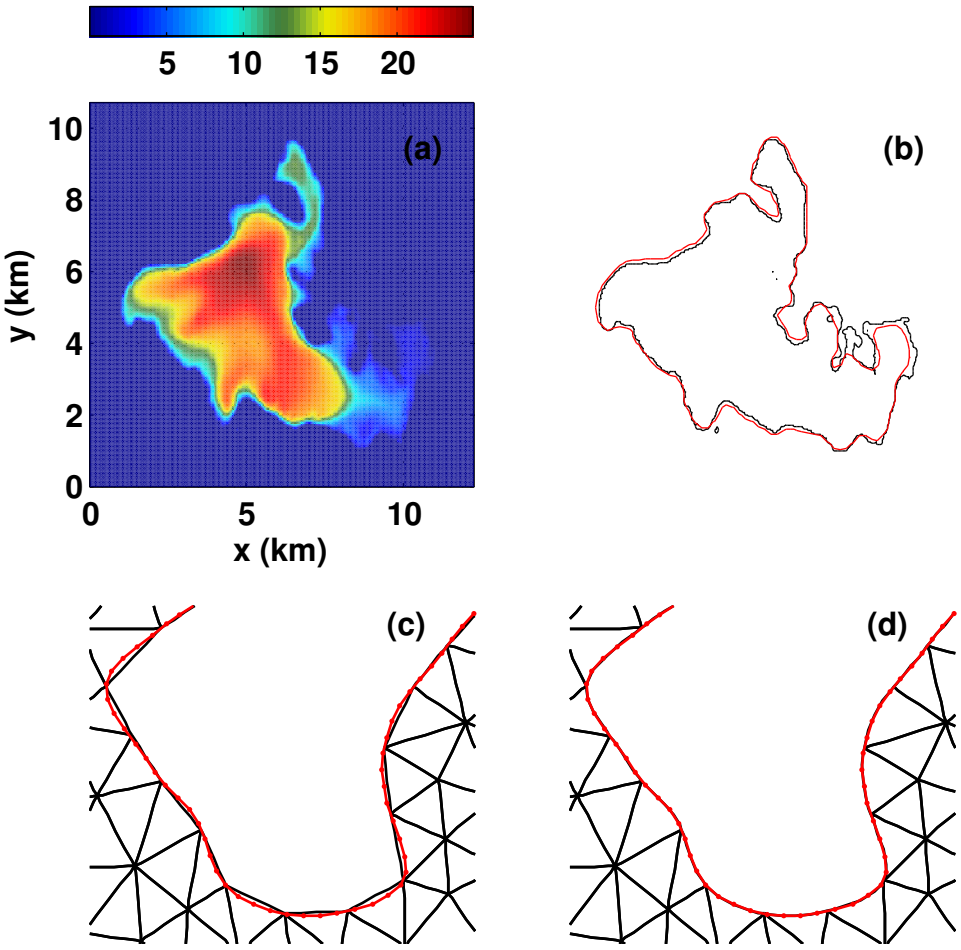

Figure 8: Panel **(a)**: Depth (in m) of Pinehurst Lake, AB from raw 50 m bathymetry data, and panel **(b)**: corresponding $H = 0$ contour (black) with smoothed coastline super-imposed (red). The lower panels show a zoomed-in section of the **(c)** straight-sided and **(d)** curved ($N = 6$) finite element mesh with $K = 1807$ elements near $(x, y) = (7 \text{ km}, 5 \text{ km})$ with cubic spline interpolant super-imposed (red).

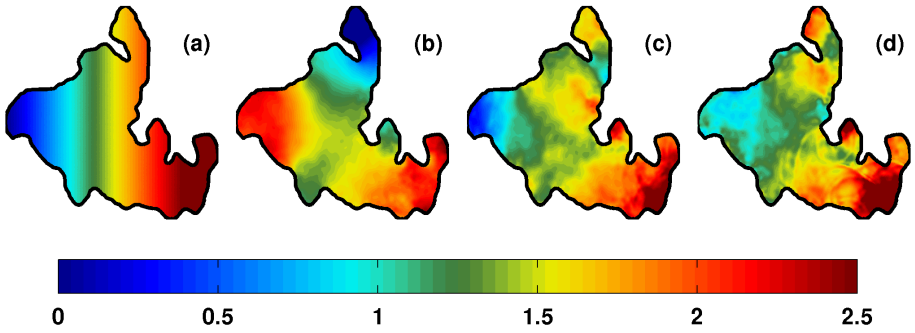

Figure 9: Evolution of an interfacial tilt in Pinehurst Lake, AB using the $N = 6$ DG-FEM with curvilinear boundary elements at times **(a)** $t = 0$ h, **(b)** $t = 19.4$ h, **(c)** $t = 39.3$ h, **(d)** $t = 62.7$ h.

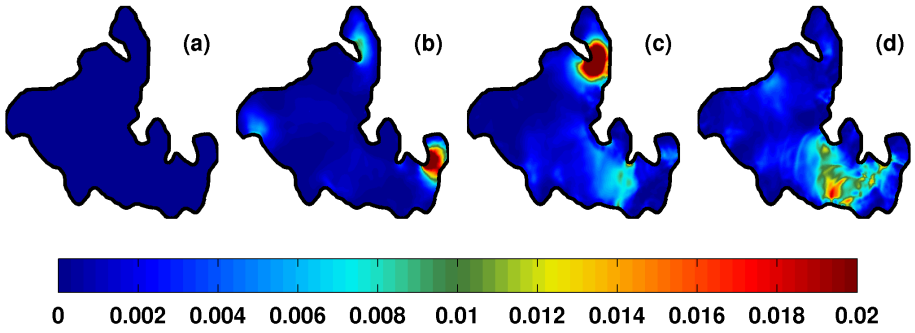

Figure 10: Like Fig. 9, except the kinetic energy density, $\frac{1}{2}h(u^2 + v^2)$ is plotted.

to be weaker. In spite of this fact, panels **(c)** and **(d)** show that nonlinear waves emerge in the shallows in the southeastern part of the basin after sufficient time has passed. As a result, small scale waves have proliferated throughout the entire basin by $t = 62.7$ h.

Figure 9 should be compared closely to Figure 10 where the kinetic energy density is plotted at the same times. At the earlier times (panels **(b)** and **(c)**), the most energetic features correspond to attached flow around peninsulas or other coastal obstacles. It is apparent that geometric focusing intensifies such features when they occur in narrow, confined parts of the basin. Panel **(d)** illustrates the kinetic energy fingerprint of small scale internal wave activity localized in the shallow eastern end of the lake at later times.

5. Conclusions

In this manuscript, we have discussed solving dispersive shallow water models of the Boussinesq-type using the discontinuous Galerkin finite element method (DG-FEM) for general geometries. The DG-FEM represents a high-order alternative to finite volume or finite element methods that allows for both high-order polynomial approximations and upwind biased advection schemes (see Hesthaven and Warburton (2008).)

Comparisons between the pseudospectral method of Steinmoeller et al. (2013) and the DG-FEM methods developed here were carried out in Section 4.1. From these comparisons, we conjecture that the DG-FEM can reach comparable resolution and energy-conserving characteristics to the Fourier-Chebyshev methods for sufficiently high polynomial order N.

It was demonstrated that the DG-FEM is poorly behaved in the neighbourhood of sharp re-entrant corners in Section 4.2, since sharp gradients and spurious eddies appear. An explanation of this phenomenon in terms of potential flow theory was offered. The remedy of rounding the corners using curvilinear elements along the boundary was proposed and the implementation was explained in Section 3. The conclusion we draw from that discussion is that general coastlines need a more computationally expensive treatment than simple circular geometries since the integrals in the DG-FEM formulations must be evaluated with cubature and quadrature rules of higher order than the approximating basis polynomials.

Finally, applications using the curvilinear element methodology were carried out. In Section 4.3, it was illustrated that the spurious eddies reported in Section 3 did not manifest when the re-entrant corner was represented in a smooth manner. The same methodology was then applied to the real-world situation of Pinehurst Lake, Alberta. The resulting high-resolution numerical solution was able to pinpoint a hot-spot of small-scale wave activity in the shallow eastern end of the basin. From this, we conclude that the DG-FEM solution of a weakly non-hydrostatic layered model may be a useful tool in helping to identify regions in lakes where internal wave-induced mixing is most dominant. The ability to identify such hot-spots has strong ecological consequences, as noted by Pannard et al. (2011). Future work should consider parametrizations of dissipative processes in very shallow regions, and incorporate wetting and drying schemes.

Appendix A. Details of the Discontinuous Galerkin Spatial Discretization Method

To apply the DG-FEM method to solve the two-dimensional shallow water equations (9)–(11) we assume that the domain Ω can triangulated using K elements (or sub-domains) and that the triangulation is geometrically conforming. That is, the boundary $\partial\Omega$ is initially represented by a piece-wise linear approximation with each line segment belonging to a side of a triangle. Subsequently, edges that lie along the boundary are deformed yielding a curvilinear boundary. The latter approach is typically much more expensive and does not lend itself easily to the nodal approach of Hesthaven and Warburton (2008) owing mainly to the fact that the mapping Jacobian to the reference triangle is non-constant. We assume that the nodes along a triangle edge that are shared between two elements are duplicated, so as to ensure that a purely local scheme can be recovered. This is a fundamental difference between DG-FEM and FEM, which uses shared nodes along a shared edge.

In each element $\mathbf{D}^k$, we form the approximate local solution $(h_h^k, (hu)_h^k, (hv)_h^k, z_h^k)$ with nodal representations

$$h_h^k(\mathbf{x},t) = \sum_{i=1}^{N_p} h_h^k(\mathbf{x}_i^k,t)\ell_i^k(\mathbf{x}) \ , \tag{A.1}$$

and similarly for the other fields. Here, $\ell_i^k(x)$ represents the i^{th} order two-dimensional Lagrange interpolating polynomial, $\mathbf{x} = (x,y)$, and N_p is the number of points within an element. We assume N_p is the same for all elements in the domain, although this is not required. The $\mathbf{x}_i^k$'s refer to the local grid points on element $\mathbf{D}^k$ with a distribution that we leave unspecified for the time being.

The nodal DG-FEM weak integral form statement is obtained by substituting the approximate local solutions into (9), multiplying by a member of the space of local test functions $V_h^k = \{\ell_j^k\}_{j=1}^{N_p}$, and integrating the flux terms by parts. If we neglect the $\mathbf{B}$ and $\mathbf{N}$ terms this gives

$$\int_{\mathbf{D}^k} \frac{\partial \mathbf{Q}_h^k}{\partial t}\ell_j^k - \mathbf{F}_h^k\frac{\partial \ell_j^k}{\partial x} - \mathbf{G}_h^k\frac{\partial \ell_j^k}{\partial y} - \mathbf{C}_h^k\ell_j^k d\mathbf{x} = -\int_{\partial\mathbf{D}^k} \ell_j^k\left(\mathbf{F}^*,\mathbf{G}^*\right)\cdot\hat{\mathbf{n}}d\mathbf{x} \tag{A.2}$$

where $\hat{\mathbf{n}}$ is the unit outward normal. As we do not require the solution to be continuous between elements, the value of $(\mathbf{F},\mathbf{G})$ in the surface integral

term on the right-hand side is not unique. Therefore, we have introduced $(\mathbf{F}^*, \mathbf{G}^*)$ as the numerical flux vector that represents some linear combination of information interior to the element $(\mathbf{F}^-, \mathbf{G}^-)$ and exterior information $(\mathbf{F}^+, \mathbf{G}^+)$. The numerical flux is our means for imposing continuity in a weak sense. Without it, the elements would completely decouple and a meaningful global solution would not be recovered. The numerical flux is typically chosen in a way that "mimics the flow of information in the underlying PDE" to ensure a stable and accurate scheme (see Hesthaven and Warburton (2008)). We use the local Lax–Friedrichs (L-F) flux

$$(\hat{n}_x \mathbf{F}_h + \hat{n}_y \mathbf{G}_h)^* = \hat{n}_x \{\!\{\mathbf{F}_h\}\!\} + \hat{n}_y \{\!\{\mathbf{G}_h\}\!\} + \frac{\lambda}{2} [\![\mathbf{Q}_h]\!] \,, \tag{A.3}$$

where

$$\{\!\{\mathbf{u}\}\!\} = \frac{\mathbf{u}^- + \mathbf{u}^+}{2}, \quad \text{and} \quad [\![\mathbf{u}]\!] = \hat{\mathbf{n}}^- \cdot \mathbf{u}^- + \hat{\mathbf{n}}^+ \cdot \mathbf{u}^+ \,, \tag{A.4}$$

are the average and jump in $\mathbf{u}$ across the interface, respectively, and λ is an approximation to the maximum linearized wave speed

$$\lambda = \max_{\mathbf{s} \in \left[\mathbf{Q}_h^-, \mathbf{Q}_h^+\right]} \left(\|\mathbf{u}(\mathbf{s})\| + \sqrt{g h(\mathbf{s})} \right) \,. \tag{A.5}$$

An alternative to (A.2) is the strong DG form

$$\int_{\mathbf{D}^k} \left(\frac{\partial \mathbf{Q}_h^k}{\partial t} + \frac{\partial \mathbf{F}_h^k}{\partial x} + \frac{\partial \mathbf{G}_h^k}{\partial y} - \mathbf{C}_h^k \right) \ell_j^k d\mathbf{x} = - \int_{\partial \mathbf{D}^k} \ell_j^k (\mathbf{F}_h^k - \mathbf{F}^*, \mathbf{G}_h^k - \mathbf{G}^*) \cdot \hat{\mathbf{n}} d\mathbf{x}. \tag{A.6}$$

The weak and strong form are analytically equivalent, but for computational and conceptual reasons we mainly use the strong form in our numerical methods, with the weak form being used for the Helmholtz operator acting on q in (13).

In order to reduce (A.6) to a form useful for numerical computations, it is important to rewrite it in terms of matrices wherever possible. For example, we can write the first component as

$$\mathcal{M}^k \frac{d\mathbf{h}^k}{dt} = -\mathcal{S}_x^k (\mathbf{hu})^k - \mathcal{S}_y^k (\mathbf{hv})^k + \int_{\partial \mathbf{D}^k} \ell_j^k \left((hu)_h^k - (hu)^*, (hv)_h^k - (hv)^* \right) \cdot \hat{\mathbf{n}} d\mathbf{x} \tag{A.7}$$

where

$$\mathbf{h}^k = \left[h_h^k(\mathbf{x}_1) \cdots h_h^k(\mathbf{x}_{N_p}) \right]^\mathsf{T} \,, \tag{A.8}$$

$$(\mathbf{hu})^k = \left[(hu)_h^k(\mathbf{x}_1) \cdots (hu)_h^k(\mathbf{x}_{N_p}) \right]^\mathsf{T} \,, \tag{A.9}$$

$$(\mathbf{hv})^k = \left[(hv)_h^k(\mathbf{x}_1) \cdots (hv)_h^k(\mathbf{x}_{N_p}) \right]^\mathsf{T} \,, \tag{A.10}$$

leaving the surface integral contribution alone for now. The is given by

$$\mathcal{M}^k_{ij} = \int_{\mathbf{D}^k} \ell^k_i(\mathbf{x})\ell^k_j(\mathbf{x})d\mathbf{x} = J^k \int_{\mathbf{I}} \ell_i(\mathbf{r})\ell_j(\mathbf{r})d\mathbf{r} = J^k\mathcal{M} \ , \tag{A.11}$$

where $J^k = x^k_r y^k_s - x^k_s y^k_r$ is the (constant) Jacobian of the linear mapping from the element $\mathbf{D}^k$ to the reference element $\mathbf{I} = \{\mathbf{r} = (r,s)|(r,s) \geq -1; r+s \leq 0\}$, and we have also introduced the mass matrix on the reference triangle, $\mathcal{M}$.

The local stiffness matrix $\mathcal{S}^k_x$ is

$$\mathcal{S}^k_{x,ij} = \int_{\mathbf{D}^k} \ell^k_i(\mathbf{x})\frac{\partial \ell^k_j}{\partial x}d\mathbf{x} \ = \ J^k \int_{\mathbf{I}} \ell_i(\mathbf{r}) \left(\frac{\partial \ell_j}{\partial r} r^k_x + \frac{\partial \ell_j}{\partial s} s^k_x \right) d\mathbf{r} \ , \tag{A.12}$$

$$= \ y^k_s \mathcal{S}_{r,ij} - y^k_r \mathcal{S}_{s,ij} \tag{A.13}$$

where we have used the fact that the Jacobian matrices have the inverse property

$$\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \begin{bmatrix} x_r & x_s \\ y_r & y_s \end{bmatrix} \begin{bmatrix} r_x & r_y \\ s_x & s_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \ , \tag{A.14}$$

hence,

$$r_x = \frac{y_s}{J}, \ r_y = -\frac{x_s}{J}, \ s_x = -\frac{y_r}{J}, \ s_y = \frac{x_r}{J}. \tag{A.15}$$

Similarly, for $\mathcal{S}^k_y$, we have

$$\mathcal{S}^k_{y,ij} = \int_{\mathbf{D}^k} \ell^k_i(\mathbf{x})\frac{\partial \ell^k_j}{\partial x}(\mathbf{x})d\mathbf{x} = -x^k_s \mathcal{S}_{r,ij} + x^k_r \mathcal{S}_{s,ij} \ . \tag{A.16}$$

The stiffness matrices defined on the standard triangle $\mathbf{I}$ are given by

$$\mathcal{S}_{r,ij} = \int_{\mathbf{I}} \ell_i(\mathbf{r})\frac{\partial \ell_j}{\partial r}(\mathbf{r}) \, d\mathbf{r} \ , \quad \mathcal{S}_{s,ij} = \int_{\mathbf{I}} \ell_i(\mathbf{r})\frac{\partial \ell_j}{\partial s} d\mathbf{r}(\mathbf{r}) \ . \tag{A.17}$$

We have hence written all local mass and stiffness matrices in terms of inner products over the standard triangle $\mathbf{I}$. For the moment, however, it is unclear how to evaluate these inner products since the explicit form of the two-dimensional Lagrange polynomials on a triangle are not known. The developments by Hesthaven and Warburton (2008) ensure that the evaluation of these inner products can be performed implicitly by considering an appropriate modal expansion that can be evaluated in a general way for arbitrary orders of approximation.

Appendix B. Evaluating the Inner Products: Modal vs. Nodal approaches

We follow Hesthaven and Warburton's discussion by introducing a modal expansion for each solution field as an alternative to the nodal representations (A.1). For example, for an arbitrary field $u(\mathbf{r})$ defined on $\mathbf{I}$, we have

$$u(\mathbf{r}) \approx u_h(\mathbf{r}) = \sum_{n=1}^{N_p} \hat{u}_n \psi_n(\mathbf{r}) = \sum_{i=1}^{N_p} u(\mathbf{r}_i)\ell_i(\mathbf{r}) \ , \tag{B.1}$$

where $\{\psi_i(\mathbf{r})\}_{i=1}^{N_p}$ is a two-dimensional basis. The relationship between the modes $\hat{u}_n$ and the nodes $u(\mathbf{r}_i)$ can be established by an L^2-projection onto a particular member of the basis ψ_m , i.e.,

$$\int_{\mathbf{I}} u(\mathbf{r})\psi_m(\mathbf{r})d\mathbf{r} = \sum_{n=1}^{N_p} \hat{u}_n \int_{\mathbf{I}} \psi_n(\mathbf{r})\psi_m(\mathbf{r})d\mathbf{r} \ , \tag{B.2}$$

or, in matrix-vector notation,

$$\mathbf{v} = \mathcal{H}\hat{\mathbf{u}} \ , \tag{B.3}$$

where

$$\hat{\mathbf{u}} = [\hat{u}_1, \cdots, \hat{u}_{N_p}], \ \ \mathcal{H}_{ij} = \int_{\mathbf{I}} \psi_i\psi_j d\mathbf{r}, \ \ \mathbf{v}_i = \int_{\mathbf{I}} u\psi_i d\mathbf{r} \ . \tag{B.4}$$

In order to ensure that $\mathcal{H}$ is well-conditioned (i.e., the basis functions are well-behaved) for an arbitrary-sized basis, we choose the basis $\{\psi(\mathbf{r})\}_{i=1}^{N_p}$ to be orthonormal in which case $\mathcal{H}$ is the identity matrix. An appropriate basis can be found be applying the Gramm-Schmidt process to the monomial basis $r^i s^j$ where $0 \leq i + j \leq N$. The result (Hesthaven and Warburton, 2008) is

$$\psi_m(\mathbf{r}) = \sqrt{2}P_i(a)P_j^{(2i+1,0)}(b)(1-b)^i \ , \tag{B.5}$$

where

$$a = 2\frac{1+r}{1-s} - 1, \ \ b = s \ , \tag{B.6}$$

and $P_n^{(\alpha,\beta)}$ is the n^{th}-order Jacobi polynomial and $P_n = P_n^{(0,0)}$ is the n^{th}-order Legendre polynomial. In one space dimension, the relationship between the order of the highest-degree basis polynomial and the number of points on the

element is given by $N_p = N + 1$. On the triangle, however, the relationship is given by the $(N+1)^{st}$ triangular number

$$N_p = \sum_{i=1}^{N+1} i = \begin{pmatrix} N+2 \\ 2 \end{pmatrix} , \tag{B.7}$$

that can be derived by counting the number of basis polynomials of degree at most N.

The only remaining question is how to evaluate the inner products on the left hand-side of the projection (B.2). If the numerical method uses a purely modal approach, one may consider using a cubature (2D quadrature) formula at the nodes, i.e.

$$\hat{u}_n \approx \sum_{i=1}^{N_p} u(\mathbf{r}_i)\psi_n(\mathbf{r}_i)w_i , \tag{B.8}$$

where the r_i's must be taken to be cubature points, and the w_i's are the associated cubature weights. Cubature and quadrature integration rules are used remove aliasing errors when curvilinear elements are employed.

In the nodal approach of Hesthaven and Warburton (2008), we assume the modal expansion interpolates u_h at the nodes $\mathbf{r}_i$, i.e.,

$$u_h(\mathbf{r}_i) = \sum_{n=1}^{N_p} \hat{u}_n \psi_n(\mathbf{r}_i) . \tag{B.9}$$

It follows that the relationship between the nodes and the modes can be established via the generalized Vandermonde matrix $\mathcal{V}$, that is

$$\mathcal{V}\hat{\mathbf{u}} = \mathbf{u} , \tag{B.10}$$

where $\mathcal{V}_{ij} = \psi_j(\mathbf{r}_i)$, $\hat{\mathbf{u}}_i = \hat{u}_i$, and $\mathbf{u}_i = u_h(\mathbf{r}_i)$. Combining (B.10) with the uniqueness statement (B.1), one can obtain the following useful formula for the Lagrange polynomials in terms of the basis polynomials

$$\ell_i(\mathbf{r}) = \sum_{n=1}^{N_p} \left(\mathcal{V}^{\mathsf{T}}\right)^{-1}_{in} \psi_n(\mathbf{r}) . \tag{B.11}$$

Appendix C. Local Operators for the Nodal Approach

Substituting (B.11) into the expression for the standard local mass matrix, we recover

$$\mathcal{M} = \left(\mathcal{V}\mathcal{V}^{\mathsf{T}}\right)^{-1} . \tag{C.1}$$

Defining the differentiation matrices

$$\mathcal{D}_{r,ij} = \left.\frac{\partial \ell_j}{\partial r}\right|_{\mathbf{r}_i}, \;\; \mathcal{D}_{s,ij} = \left.\frac{\partial \ell_j}{\partial s}\right|_{\mathbf{r}_i}, \tag{C.2}$$

whose entries may be furnished directly by appropriate differentiation of (B.11), the local stiffness matrices can be recovered by

$$\mathcal{M}\mathcal{D}_r = \mathcal{S}_r, \;\; \mathcal{M}\mathcal{D}_s = \mathcal{S}_s . \tag{C.3}$$

This is useful because it implies that an explicit semi-discrete scheme can be obtained by multiplying (A.7) by $(\mathcal{M}^k)^{-1} = \frac{1}{J^k}\mathcal{M}^{-1}$. As a consequence of the fact that the local mass matrix only varies by a constant factor on each element, it follows that this operation is computationally cheap since $\mathcal{M}$ is an $N_p \times N_p$ matrix. For example, with order $N = 8$ basis functions, the local mass matrix is a 45×45 full matrix. This is another key difference between DG-FEM and the classical FEM, where explicit semi-discrete schemes often cannot be recovered since the time-derivative operator is multiplied by the global mass matrix, that may be large and expensive to invert explicitly.

Appendix D. Surface Integral Contributions

To close our numerical scheme, it remains to discuss the surface integral term in equation (A.7)

$$\int_{\partial \mathbf{D}^k} \ell_j^k(\mathbf{x}) \mathbf{g}_h \cdot \hat{\mathbf{n}} \, d\mathbf{x} \tag{D.1}$$

where $\mathbf{g_h} = \left((hu)_h^k - (hu)^*, (hv)_h^k - (hv)^*\right)$ represents the jump in flux across an interface. Since the normal $\hat{\mathbf{n}}$ is constant along each edge, it is useful to break this expression up into three integrals

$$\int_{\partial \mathbf{D}^k} \ell_j^k(\mathbf{x}) \mathbf{g}_h \cdot \hat{\mathbf{n}} d\mathbf{x} = \sum_{e=1}^{3} \hat{\mathbf{n}}_e \cdot \int_{\text{edge}_e} \ell_j^k(\mathbf{x}) \mathbf{g}_h d\mathbf{x} . \tag{D.2}$$

If we substitute the nodal expression $\mathbf{g}_h = \sum_{i=1}^{N+1} \ell_i^k(\mathbf{x})\mathbf{g}_i$ the right hand side reduces to

$$\sum_{e=1}^{3}\sum_{i=1}^{N+1} \hat{\mathbf{n}}_e \cdot \mathbf{g}_i \mathcal{M}_{ij}^{k,e} \,, \tag{D.3}$$

where we have introduced the $(N+1)\times(N+1)$ edge mass matrix

$$\mathcal{M}_{ij}^{k,e} = \int_{edge_e} \ell_j^k(\mathbf{x})\ell_i^k(\mathbf{x})\, d\mathbf{x} = J^{k,e,1}\mathcal{M}_{ij}^1 \,. \tag{D.4}$$

Here $J^{k,e,1}$ is the Jacobian of the mapping from the edge to the standard interval $[-1,1]$. Using the 1D developments in Hesthaven and Warburton (2008), the standard 1D mass matrix is related to the Vandermonde matrix for 1D polynomial interpolation by $\mathcal{M}^1 = \left(\mathcal{V}^1(\mathcal{V}^1)^T\right)^{-1}$.

Appendix E. Boundary Conditions

The freedom in the numerical flux choice gives us a convenient way to impose boundary conditions through appropriately choosing imaginary "ghost" states, i.e. the '+' traces along boundary edges. For a purely reflective wall with no flow going through it, we impose

$$h^+ = h^- \,, \tag{E.1}$$
$$hu^+ = hu^- - 2(n_x hu + n_y hv)n_x \,, \tag{E.2}$$
$$hv^+ = hv^- - 2(n_x hu + n_y hv)n_y \,, \tag{E.3}$$

The first condition is equivalent to imposing $\nabla h \cdot \hat{\mathbf{n}} = 0$. The remaining conditions are equivalent to imposing no normal flow along the wall.

Appendix F. Bathymetry and Non-hydrostatic terms

So far, we have not discussed the treatment of the bathymetry and nonhydrostatic terms contained in the vectors $\mathbf{B}$ and $\mathbf{N}$, respectively, which cannot be addressed by the standard nodal DG-FEM treatment.

As an example of the issues that arise, consider the second entry of $\mathbf{B}_h$. If we remove the subscript-h notation for clarity, multiply by ℓ_j^k, and integrate over the element, the following integrals appear in the strong DG statement

$$\int_{\mathbf{D}^k} gh^k \frac{\partial H^k}{\partial x}\ell_j(x)d\mathbf{x} - \int_{\partial\mathbf{D}^k} gh\left(H^k - H^*\right) d\mathbf{x}. \tag{F.1}$$

The surface integral term does not pose a problem, and in the case where H is continuous across element interfaces, it vanishes. The first term does pose a problem because we cannot write it in terms of the local stiffness matrix $\mathcal{S}_x^k$. To see this, let us substitute a nodal expansion for H, yielding

$$\begin{aligned}\int_{\mathbf{D}^k} gh^k \frac{\partial H^k}{\partial x}\ell_j(\mathbf{x})d\mathbf{x} &= \sum_{i=1}^{N_p} gH^k(x_i)\int_{\mathbf{D}^k} h^k\ell_j(\mathbf{x})\frac{\partial \ell_i}{\partial x}d\mathbf{x} \\ &= \sum_{i=1}^{N_p} gH^k(x_i)\mathcal{S}_{ji}^{k,h} \\ &= g\big(\mathcal{S}^{k,h}\mathbf{H}^k\big)_j\ , \end{aligned} \tag{F.2}$$

where we have taken the integral on the right to be the modified local stiffness matrix, which depends on h. Since h is a function of both space and time, this approach is computationally expensive since the local stiffness matrix is different on every element and must be updated after each time-step. This approach is necessary in situations where curvilinear elements are used since the mapping Jacobian is no longer constant. See Section 3.2.

For a less expensive approach, we introduce the auxiliary variable

$$\kappa(\mathbf{x}) = \frac{\partial H}{\partial x}\ . \tag{F.3}$$

Following previous discussion, we can approximate κ by

$$\mathcal{M}^k\boldsymbol{\kappa} = \mathcal{S}_x\mathbf{H}^k - \int_{\partial\mathbf{D}^k}\left(H^k - H^*\right)n_x\,d\mathbf{x}\ , \tag{F.4}$$

or,

$$\boldsymbol{\kappa} = \mathcal{D}_x\mathbf{H}^k - \left(\mathcal{M}^k\right)^{-1}\int_{\partial\mathbf{D}^k}\left(H^k - H^*\right)n_x\,d\mathbf{x}\ . \tag{F.5}$$

If we now return to the bathymetry terms, we are charged with computing the integral

$$\int_{\mathbf{D}^k} gh^k(\mathbf{x})\kappa^k(\mathbf{x})\ell_j(\mathbf{x})d\mathbf{x}\ . \tag{F.6}$$

We could proceed as before and simply substitute in the nodal expansion for κ^k. We would then be left with a modified mass matrix $\mathcal{M}^{k,h}$ and we will not have gained much. On the other hand, if we approximate the nodal expansion product $h^k\kappa^k$ in the following manner

$$h^k(\mathbf{x})\kappa^k(\mathbf{x}) \approx \sum_{i=1}^{N_p} h^k(\mathbf{x}_i)\kappa^k(\mathbf{x}_i)\ell_i^k(\mathbf{x})\ , \tag{F.7}$$

i.e., we approximate the function product with a point-wise (or Schur) product, we then recover the scheme

$$\begin{aligned}\int_{\mathbf{D}^k} gh^k \frac{\partial H^k}{\partial x}\ell_j(\mathbf{x})d\mathbf{x} &\approx \sum_{i=1}^{N_p} g\kappa^k(\mathbf{x}_i)h^k(\mathbf{x}_i)\int_{\mathbf{D}^k}\ell_i(\mathbf{x})\ell_j(\mathbf{x})d\mathbf{x}\,,\\ &= g\big(\mathcal{M}^k\left(\boldsymbol{\kappa}\mathbf{h}\right)^k\big)_j\,, \end{aligned} \tag{F.8}$$

which is less computationally expensive than the former scheme since the local mass matrix only varies by a constant value between elements.

The price we pay when using this approximation is that we have essentially committed a couple of "variational crimes." Aliasing errors result from two distinct sources: 1) the fact that a product of two functions cannot be completely recovered by a point-wise product between the nodal values; and 2) the fact that the interpolant of a derivative is not the same thing as the derivative of an interpolant. We use modal filtering as described in section 2.4, with the exception that it is applied to the 2D basis on each element, to prevent these aliasing errors from driving weak instabilities.

The inexpensive nodal approach presented here is used in time-stepping both the bathymetry terms $gh\nabla H$ and the non-hydrostatic terms $\gamma\nabla z$. Both of these terms may be regarded as source terms in the DG-FEM formulation assuming that h and z are known. The gradient of z may be either evaluated using the central flux $z^* = \{\!\{z\}\!\}$ or the purely internal choice $z^* = z^-$. We explain how z is calculated by solving the elliptic problem (5) within the DG-FEM framework.

Appendix G. DG-FEM for elliptic problems

Inspecting the system (13)-(15) it may be unclear how, given an input right-hand side $-\nabla\cdot\mathbf{a}$, one can recover z. This is achieved by considering the inverse situation, i.e., if z is known, then $\mathbf{q}$ can be computed by solving equations (14)-(15), and $-\nabla\cdot\mathbf{a}$ can be recovered using (13). This set of operations can be considered a non-singular linear transformation, and hence there must exist an inverse transformation.

The strong DG formulation of (14)–(15) together with the weak formulation of (13) is given by

$$\mathcal{M}^k{\mathbf{q}_\mathbf{x}}^k = \sqrt{\boldsymbol{\gamma}}^k\mathcal{S}_x\mathbf{z}^k - \sqrt{\boldsymbol{\gamma}}^k\int_{\partial\mathbf{D}^k}\ell_j^k\left(z^k - z^*\right)n_x\,d\mathbf{x}\,, \tag{G.1}$$

$$\mathcal{M}^k{\mathbf{q}_\mathbf{y}}^k = \sqrt{\boldsymbol{\gamma}}^k\mathcal{S}_y\mathbf{z}^k - \sqrt{\boldsymbol{\gamma}}^k\int_{\partial\mathbf{D}^k}\ell_j^k\left(z^k - z^*\right)n_y\,d\mathbf{x}\,, \tag{G.2}$$

$$-(\mathcal{S}_x^k)^T \left(\sqrt{\gamma}\mathbf{q_x}\right)^k - (\mathcal{S}_y^k)^T \left(\sqrt{\gamma}\mathbf{q_y}\right)^k + \int_{\partial\mathbf{D}^k} \ell_j^k (\sqrt{\gamma}\mathbf{q})^* \cdot \hat{\mathbf{n}}\, d\mathbf{x} - \mathcal{M}^k \mathbf{z}^k$$
$$= (\mathcal{S}_x^k)^T \mathbf{a_x}^k + (\mathcal{S}_y^k)^\mathsf{T} \mathbf{a_y}^k - \int_{\partial\mathbf{D}^k} \ell_j^k \mathbf{a}^* \cdot \hat{\mathbf{n}}\, d\mathbf{x} \,. \qquad \text{(G.3)}$$

We use the central flux for the right-hand side, i.e. $\mathbf{a}^* = \{\!\{\mathbf{a}\}\!\}$ together with the interior penalty (IP) flux for the elliptic operator, i.e. $z^* = \{\!\{z\}\!\}$, $(\sqrt{\gamma}\mathbf{q})^* = \{\!\{\sqrt{\gamma}\nabla z\}\!\} - \tau[\![z]\!]$, $\tau > 0$. The penalty term penalizes large jumps at the element interfaces. If $\tau = 0$, a numerical calculation of the eigenfunctions of the Laplacian would reveal a spurious $\lambda = 0$ mode with all elements completely de-coupled, and the system would be singular (Hesthaven and Warburton, 2008). The use of the penalty term pushes the spurious eigenmode out of the operator's null space to guarantee invertibility. In general, a sufficiently large penalty parameter will suppress any other spurious modes to the high-λ part of the eigenspectrum as well. This property represents an advantage over continuous Galerkin discretizations of elliptic operators that often possess spurious *convergent* modes whose corresponding eigenvalues can lie within the physical range of the eigenspectrum. Modes of this type were discussed in an ocean modelling context by Cotter et al. (2009).

The IP flux offers a balance between the penalized central flux and local discontinuous Galerkin flux methods, giving optimal convergence at all orders, a middle-ground in terms of sparsity, and similar condition numbers to the central-flux operator (Hesthaven and Warburton, 2008). Furthermore, with some algebraic manipulations, the auxiliary variable $\mathbf{q}$ can be eliminated locally, allowing the operator to be efficiently set-up directly as a symmetric sparse matrix. In terms of local operators we get

$$\begin{aligned} - & \left(\left(\mathcal{D}_x^k\right)^T \mathcal{M}^k \Gamma^k \left(\mathcal{M}^k\right)^{-1} \Gamma^k \mathcal{M}^k \mathcal{D}_x^k + \left(\mathcal{D}_y^k\right)^T \mathcal{M}^k \Gamma^k \left(\mathcal{M}^k\right)^{-1} \Gamma^k \mathcal{M}^k \mathcal{D}_y^k + \mathcal{M}^k \right) \mathbf{z}^k \\ + & \sum_{e=1}^{3} \left(\mathcal{D}_n^{k,e}\right)^T \mathcal{M}^{k,e} \Gamma^{k,e} \left(\mathcal{M}^{k,e}\right)^{-1} \Gamma^{k,e} \mathcal{M}^{k,e} \left(\frac{\mathbf{z}^- - \mathbf{z}^+}{2} \right) \qquad \text{(G.4)} \\ + & \sum_{e=1}^{3} \mathcal{M}^{k,e} \left[\Gamma^{k,e} \left(\mathcal{D}_n^{k,e} \left(\frac{\mathbf{z}^- - \mathbf{z}^+}{2} \right) + \tau(\mathbf{z}^- - \mathbf{z}^+) \right) \right] = \text{RHS} \,, \end{aligned}$$

where Γ^k is the diagonal matrix with the entries of $\sqrt{\gamma}^k$ written along its diagonal and $\mathcal{D}_n^{k,e} = \mathcal{D}_x^k n_x^{k,e} + \mathcal{D}_y^k n_y^{k,e}$ is the discretized normal derivative along edge e of element k.

The discontinuous Galerkin IP discretization method has become known as the 'symmetric interior penalty discontinuous Galerkin' (SIP-DG) method

in the literature, and has been applied to the pressure Poisson equation and viscous operator of the incompressible Navier–Stokes equations (Ferrer and Willden, 2011; Shahbazi et al., 2007).

Acknowledgement

The research is supported by the Natural Sciences and Engineering Research Council of Canada through Discovery Grants to M.S. and K.G.L.

References

Brandt, P., Rubino, A., Alpers, W., Backhaus, J., 1997. Internal waves in the Strait of Messina studied by a numerical model and synthetic aperture radar umages from *ERS 1/2* Satellites. J. Phys. Oceangr. **27**, 648–663.

Cockburn, B., Hou, S., Shu, C., 1990. The Runge–Kutta local projection discontinuous Galerkin finite-element method for conservation-laws. 4. The multidimensional case. Math. Comp 54 (190), 545–581.

Cockburn, B., Shu, C., 1989. TVB Runge–Kutta local projection discontinuous Galerkin finite-element method for conservation-laws. 2. General framework. Math. Comp. 52 (186), 411–435.

Cotter, C. J., Ham, D. A., Pain, C. C., Reich, S., 2009. LBB stability of a mixed Galerkin finite element pair for fluid flow simulations. J. Comp. Phys. 228 (2), 336–348.

de la Fuente, A., Shimizu, K., Imberger, J., Niño, Y., 2008. The evolution of internal waves in a rotating, stratified, circular basin and the influence of weakly nonlinear and nonhydrostatic accelerations. Limnol. Oceanogr. 53 (6), 2738–2748.

Dupont, F., 2001. Comparison of numerical methods for modelling ocean circulation in basins with irregular coasts. Ph.D. thesis, McGill University.

Eskilsson, C., Sherwin, S., 2005. Spectral/*hp* discontinuous Galerkin methods for modelling 2D Boussinesq equations. J. Sci. Comp. **22**, 269–288.

Ferrer, E., Willden, R., 2011. A high order Discontinuous Galerkin Finite Element solver for the incompressible Navier–Stokes equations. Comput. Fluids 46, 224–230.

Geuzaine, C., Remacle, J.-F., 2009. A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Int. J. Numer. Methods Eng. 79, 1309–1331.

Gordon, W., Hall, C., 1973. Construction of curvilinear coordinate systems and application to mesh generation. Int. J. Numer. Meth. Eng. 7, 461–477.

Hesthaven, J., Warburton, T., 2008. Nodal Discontinuous Galerkin Methods. Springer.

Hughes, T., 1987. Recent progress in the development and understanding of SUPG methods with special reference to the compressible euler and navier-stokes equations. Int. J. Numer. Method. Fluids 7, 1261–1275.

Karniadakis, G., Sherwin, S., 2005. Spectral/hp Element Methods for Computational Fluid Dynamics, 2nd Edition. Oxford University Press, USA.

Kundu, P., Cohen, I., 2008. Fluid Mechanics, 4th Edition. Elsevier Academic Press.

625 Leveque, R., 2007. Finite Difference Methods for Ordinary and Partial Differential Equa-
626 tions: Steady-State and Time-Dependent Problems. Society for Industrial and Applied
627 Mathematics.
628 Levin, J. C., Iskandarani, M., Haidvogel, D. B., 2006. To continue or discontinue: Com-
629 parisons of continuous and discontinuous Galerkin formulations in a spectral element
630 ocean model. Ocean Modelling 15 (1-2), 56–70.
631 Mandli, K., 2011. Finite volume methods for the multilayer shallow water equations
632 with applications to storm surges. Ph.D. thesis, University of Washington, Seattle,
633 Washington, USA.
634 Nazarov, M., Popov, B., 2014 (submitted). A posteriori error estimation for the com-
635 pressible euler equations using entropy viscosity. Comp. & Fluids.
636 Pannard, A., Beisner, B., Bird, D., Braun, J., Planas, D., Bormans, M., 2011. Recurrent
637 internal waves in a small lake: Potential ecological consequences for metalimnetic
638 phytoplankton populations. L&O:F&E 1, 91–109.
639 Peregrine, D., 1967. Long waves on a beach. J. Fluid. Mech. **27**(4), 815–827.
640 Rhebergen, S., Cockburn, B., 2012. A space–time hybridizable discontinuous Galerkin
641 method for incompressible flows on deforming domains. J. Comp. Phys. 231, 4185–
642 4204.
643 Shahbazi, K., Fischer, P., Ethier, C., 2007. A high-order discontinuous Galerkin method
644 for the unsteady incompressible Navier–Stokes equations. J. Comp. Phys 222, 391–407.
645 Steinmoeller, D., Stastna, M., Lamb, K., 2012. Fourier pseudospectral methods for 2D
646 Boussinesq-type equations. Ocean Modelling 52–53, 76–89.
647 Steinmoeller, D., Stastna, M., Lamb, K., 2013. Pseudospectral methods for Boussinesq-
648 type equations in an annular domain with applications to mid-sized lakes. J. Comp.
649 Sci. 4, 3–11.
650 Stocker, R., Imberger, J., 2003. Energy partitioning and horizontal dispersion in a strat-
651 ified rotating lake. J. Phys. Oceanogr. 33, 512–529.
652 Thomson, W., 1872. On gravitational oscillations of rotating water. P. Roy. Soc. Edinb.
653 **82**, 92–100.
654 Tomasson, G., Melville, W., 1992. Geostrophic adjustment in a channel: nonlinear and
655 dispersive effects. J. Fluid Mech. 241, 23–48.
656 Toro, E., 1999. Riemann Solvers and Numerical Methods for Fluid Dynamics, 2nd Edi-
657 tion. Springer.
658 Walkley, M., 1999. A numerical method for extended boussinesq shallow-water wave
659 equations. Ph.D. thesis, University of Leeds, UK.
660 Wandzura, S., Xiao, H., 2003. Symmetric quadrature rules on a triangle. Comput. Math.
661 Applica. 45, 1829–1840.
662 Zhang, X., Xia, Y., Shu, C.-W., 2012. Maximum-principle-satisfying and positivity-
663 preserving high order Discontinuous Galerkin schemes for conservation laws on tri-
664 angular meshes. J. Sci. Comput. 50, 29–62.