

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. In situations of wave propagation around sharp re-entrant obstacles (such as headlands), spurious eddies are generated. 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 is not a reasonable approach. 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 1. Introduction

² [Derek: Say something general about lake modelling here, motivate the
³ desire for high-order methods, explain what has been done with Jocs/OM

*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)

⁴ paper, and how we seek to bridge the gap between simple and complex
⁵ geometries here with DG.]

⁶ Since a lake's coastal boundary generally specifies a physical domain
⁷ with complex/irregular boundaries, the pseudo spectral methods presented
⁸ in Steinmoeller et al. (2012, 2013) are not sufficient for modelling real-world
⁹ lakes. To represent more general geometries, we turn to the Discontinuous
¹⁰ Galerkin finite element method (DG-FEM) as a high-order alternative to
¹¹ the low-order finite volume and finite element methods that are commonly
¹² used for irregular geometries. The results presented in this manuscript typ-
¹³ ically use local polynomial orders between $N = 4$ and $N = 8$. The methods
¹⁴ are thus high-order in contrast to traditional finite element methods that
¹⁵ typically use piece-wise linear or quadratic basis functions. See for instance
¹⁶ Walkley (1999), who solved a Boussinesq-type system with a low-order fi-
¹⁷ nite element method (FEM).Marek: may be one or two more low order
¹⁸ references?

¹⁹ It is worth stressing that the high-order DG-FEM is not the same as
²⁰ the spectral element method (SEM) (see Karniadakis and Sherwin (2005))
²¹ that represents the high-order extension of the traditional FEM. Both FEM
²² and SEM are continuous Galerkin formulations which require C^0 continuity
²³ at element interfaces. Although DG-FEM and SEM both use a high-order
²⁴ orthogonal polynomial basis, the DG-FEM only imposes continuity in a
²⁵ weak sense through the specification of a numerical flux function at ele-
²⁶ ment edges in order to allow for stable advection schemes (see for instance
²⁷ Cockburn and Shu (1989); Cockburn et al. (1990); Hesthaven and Warbur-
²⁸ ton (2008)). The requirement of C^0 continuity in the SEM means that the
²⁹ method is not ideal for advection problems since an upwind-type scheme
³⁰ cannot be formulated to account for the preferred direction of propagation
³¹ of information as explained by Hesthaven and Warburton (2008). This
³² shortcoming can lead to situations where Gibbs oscillations are trapped at
³³ element interfaces, as has been illustrated for the spectral element ocean
³⁴ model by Levin et al. (2006). However, it should be noted that modern
³⁵ treatments of FEM/SEM seek to overcome this shortcoming for advective
³⁶ problems by considering stabilization techniques such as the SUPG (stream-
³⁷ line upwind/Petrov–Galerkin) method Hughes (1987) as well as the class
³⁸ of entropy-based viscosity methods Nazarov and Popov (2014 (submitted)).
³⁹ An alternative to the purely discontinuous approach has been recently pro-
⁴⁰ posed in the form of the hybridizable discontinuous Galerkin method that
⁴¹ imposes strong continuity only in the edge-normal component of fluxes (see

42 Rhebergen and Cockburn (2012)).

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

50 In the following sections, we follow the techniques and developments for
51 nodal DG-FEM as presented by Hesthaven and Warburton (2008), and we
52 build upon their techniques where necessary. We briefly explain the basic
53 nodal DG-FEM formulation as the spatial discretization method for both
54 hyperbolic and elliptic systems and the corresponding reduction to local
55 operators in the context of a one-layer dispersive shallow water model. Fol-
56 lowing this, a comparison with the pseudospectral methods of Steinmoeller
57 et al. (2012, 2013) is carried out as a means of validating the numerical
58 scheme presented here and illustrating the resolution characteristics of the
59 DG-FEM at varying polynomial orders. The necessity of curvilinear el-
60 ements for general situations is illustrated due to singular/spurious flow
61 features that can emerge due to a piece-wise linear representation of the
62 boundary. It is then explained how the nodal DG-FEM method should be
63 augmented with high-order cubature and quadrature integration rules to
64 deal with the non-constant mapping Jacobians introduced by curvilinear
65 elements. Highlight simulations are provided alongside the developments,
66 and we conclude with a simulation of a real-world lake: Pinehurst Lake,
67 Alberta, Canada.

68 2. Methods

69 2.1. Governing Equations

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

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

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

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

70 where $\mathbf{u}(x, y, t) = (u(x, y, t), v(x, y, t))$ is the velocity field, $h(x, y, t) =$
 71 $H(x, y) + \eta(x, y, t)$ is the total depth with H representing the undisturbed
 72 depth, and η is the free surface displacement. The constants g' and f are the
 73 reduced acceleration due to gravity and the Coriolis frequency, respectively.
 74 The main difference between the set of equations (1)–(3) and the traditional
 75 hyperbolic shallow water model are the dispersive terms $\frac{H^2}{6}\nabla(\nabla \cdot (\mathbf{u}h)_t)$
 76 found in the momentum equations (2) & (3). The above system was first
 77 proposed by Brandt et al. (1997) in their study of internal waves in the
 78 Strait of Messina.

An efficient scheme for evolving the dispersive terms can be obtained by adding an auxiliary elliptic equation to the Boussinesq system. The method introduces the scalar auxiliary variable

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

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

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

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

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

79 A full derivation of (5) and (6) can be found in Steinmoeller et al. (2012).
 80 We have neglected bottom and surface stresses in equations (1)–(3) since
 81 their inclusion into the numerical scheme is conceptually easy and con-
 82 tributes little to the discussion. We have also chosen to focus on the case of
 83 a single fluid layer of constant density, since the inclusion of multiple layers
 84 adds considerable complexity to numerical formulations that rely on ap-
 85 proximate solutions to the corresponding nonlinear Riemann problem. See
 86 Mandli (2011) for a discussion in the context of the finite volume method.

87 2.2. Time-Stepping Techniques

88 The time-stepping technique applied to the DG-FEM discretized version
 89 of the one-layer model closely follows the ‘scalar approach’ used for the
 90 pseudospectral discretization in Steinmoeller et al. (2012, 2013) and for the
 91 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 the two.

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 as discussed in Appendix A, we recover the system of ordinary differential equations

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

where $\mathbf{Q} = (h, hu, hv)^T$ is the vector of unknowns and \mathcal{R} is the DG spatial discretization operator for the advection, Coriolis, and bathymetry source terms. It is assumed that we have left-multiplied by the inverse of the local mass matrix, i.e., $(\mathcal{M}^k)^{-1}$, in arriving at \mathcal{R} so that there is no matrix operator acting on the time-derivative on the left-hand side of (7). 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, yielding

$$\mathbf{V}^{(1)} = \mathbf{Q}^n + \Delta t \mathcal{R}(\mathbf{Q}^n), \quad (8)$$

$$\mathbf{V}^{(2)} = \frac{1}{4} (3\mathbf{Q}^n + \mathbf{V}^{(1)} + \Delta t \mathcal{R}(\mathbf{V}^{(1)})), \quad (9)$$

$$\mathbf{Q}^\dagger = \frac{1}{3} (\mathbf{Q}^n + 2\mathbf{V}^{(2)} + 2\Delta t \mathcal{R}(\mathbf{V}^{(2)})). \quad (10)$$

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. However, shocks and discontinuous features are not of concern for the equations under consideration here since the presence of third derivatives in the governing equations

¹⁰⁹ presumes a greater degree of smoothness than that assumed by a purely
¹¹⁰ 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 $\mathcal{R}(\mathbf{Q}^\dagger)$ for only the $\mathbf{h}\mathbf{u}$ and $\mathbf{h}\mathbf{v}$ equations. The auxiliary variable \mathbf{z}^\dagger is then computed by inverting the matrix representation of the SIP-DG formulation of the Helmholtz operator (E.8). The momentum equations are finally updated via

$$(\mathbf{h}\mathbf{u})^{n+1} = (\mathbf{h}\mathbf{u})^\dagger + \gamma \Delta t \nabla z^{n+1}, \quad (11)$$

¹¹¹ where the DG-FEM discretization of the source terms involving nonlinear
¹¹² products with gradients of known quantities is discussed in Appendix D,
¹¹³ and $\gamma = H^2/6$. Hence, the vector of unknowns at time t_{n+1} is updated via
¹¹⁴ $\mathbf{Q}^{n+1} = (h^\dagger, h\mathbf{u}^{n+1}, h\mathbf{v}^{n+1})^\top$.

¹¹⁵ For the simulations considered in this manuscript, we have found that
¹¹⁶ the SIP-DG sparse matrix can be factored using the sparse LU-decomposition,
¹¹⁷ as was explained in Steinmoeller et al. (2012), so that the factors may be
¹¹⁸ re-used at each time-step for fast inversions, and we have side-stepped the
¹¹⁹ issue of using an iterative solver such as GMRES that is required for the
¹²⁰ pseudospectral methods of Steinmoeller et al. (2012, 2013). At high resolu-
¹²¹ tions, a linear iterative method will certainly be necessary due to memory
¹²² restrictions prohibiting the storage of the LU factors.

¹²³ 2.3. DG-FEM Spatial Discretization

¹²⁴ The DG-FEM method is primarily suited to solving strictly hyperbolic
¹²⁵ equation sets. To accommodate the method as much as possible, the equa-
¹²⁶ tions are re-cast in the form of a conservation law plus source terms:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{B} + \mathbf{C} + \mathbf{N}, \quad (12)$$

with

$$\mathbf{Q} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}. \quad (13)$$

The terms

$$\mathbf{B} = gh \begin{pmatrix} 0 \\ \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial y} \end{pmatrix}, \quad \mathbf{C} = f \begin{pmatrix} 0 \\ vh \\ -uh \end{pmatrix}, \quad \mathbf{N} = \frac{H^2}{6} \begin{pmatrix} 0 \\ \frac{\partial z}{\partial x} \\ \frac{\partial z}{\partial y} \end{pmatrix}, \quad (14)$$

127 are the bed slopes, Coriolis terms, and the dispersive terms, respectively.
 128 As explained in Section ??, The variable $z = \nabla \cdot (\mathbf{u}h)_t$ is governed by
 129 the Helmholtz problem (5). In the case of the traditional non-rotating
 130 hydrostatic shallow water model with a flat bottom, the right-hand side of
 131 (A.1) vanishes.

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

$$h_h^k(\mathbf{x}, t) = \sum_{i=1}^{N_p} h_h^k(\mathbf{x}_i^k, t) \ell_i^k(\mathbf{x}), \quad (15)$$

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

The nodal DG-FEM weak integral form statement is obtained by substituting the approximate local solutions into each of equations (A.1), 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 and focus only on (A.1) for the moment, this gives

$$\int_{\mathbf{D}^k} \frac{\partial \mathbf{Q}_h^k}{\partial t} \ell_j^k - \mathbf{F}_h^k \frac{\partial \ell_j^k}{\partial x} - \mathbf{G}_h^k \frac{\partial \ell_j^k}{\partial y} - \mathbf{C}_h^k \ell_j^k d\mathbf{x} = - \int_{\partial \mathbf{D}^k} \ell_j^k (\mathbf{F}^*, \mathbf{G}^*) \cdot \hat{\mathbf{n}} d\mathbf{x} \quad (16)$$

where $\hat{\mathbf{n}}$ is the unit outward-pointing normal. Due to the fact that 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 $(\mathbf{F}^+, \mathbf{G}^+)$. Since we have not explicitly imposed continuity at element interfaces, 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)). The choice of numerical

flux considered in this manuscript is the local Lax–Friedrichs (L-F) flux

$$(\hat{n}_x \mathbf{F}_h + \hat{n}_y \mathbf{G}_h)^* = \hat{n}_x \{\!\{ \mathbf{F}_h \}\!\} + \hat{n}_y \{\!\{ \mathbf{G}_h \}\!\} + \frac{\lambda}{2} [\![\mathbf{Q}_h]\!] , \quad (17)$$

where

$$\{\!\{ \mathbf{u} \}\!\} = \frac{\mathbf{u}^- + \mathbf{u}^+}{2}, \quad [\![\mathbf{u}]\!] = \hat{\mathbf{n}}^- \cdot \mathbf{u}^- + \hat{\mathbf{n}}^+ \cdot \mathbf{u}^+, \quad (18)$$

are the average and jump in \mathbf{u} across the interface, respectively. The numerical flux choice (A.7) represents the local Lax-Friedrichs flux where λ is an approximation to the maximum linearized wave speed

$$\lambda = \max_{\mathbf{s} \in [\mathbf{Q}_h^-, \mathbf{Q}_h^+]} \left(\|\mathbf{u}(\mathbf{s})\| + \sqrt{gh(\mathbf{s})} \right). \quad (19)$$

138 The integral formulation is subsequently reduced to mass and stiffness
139 matrix operators and implemented in MATLAB following the methods dis-
140 cussed in the Appendix and in Hesthaven and Warburton (2008).

141 2.3.1. DG-FEM Discretization for Elliptic Operators

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

To re-write (A.4) as a first-order system, we introduce the auxiliary variable

$$\mathbf{q} = (q_x, q_y) = \sqrt{\gamma} \nabla z, \quad (20)$$

yielding the system

$$\nabla \cdot (\sqrt{\gamma} \mathbf{q}) - z = -\nabla \cdot \mathbf{a}, \quad (21)$$

$$q_x = \sqrt{\gamma} \frac{\partial z}{\partial x}, \quad (22)$$

$$q_y = \sqrt{\gamma} \frac{\partial z}{\partial y}. \quad (23)$$

Once volume-type integrals have been replaced with local matrix operators, tThe strong DG formulation of (E.3)–(E.4) together with the weak formulation of (E.2) is given by

$$\mathcal{M}^k \mathbf{q}_x^k = \sqrt{\gamma}^k \mathcal{S}_x \mathbf{z}^k - \sqrt{\gamma}^k \int_{\partial \mathbf{D}^k} \ell_j^k (z^k - z^*) n_x d\mathbf{x}, \quad (24)$$

$$\mathcal{M}^k \mathbf{q}_y^k = \sqrt{\gamma}^k \mathcal{S}_y \mathbf{z}^k - \sqrt{\gamma}^k \int_{\partial \mathbf{D}^k} \ell_j^k (z^k - z^*) n_y d\mathbf{x}, \quad (25)$$

$$-(\mathcal{S}_x^k)^T (\sqrt{\gamma} \mathbf{q}_x)^k - (\mathcal{S}_y^k)^T (\sqrt{\gamma} \mathbf{q}_y)^k + \int_{\partial \mathbf{D}^k} \ell_j^k (\sqrt{\gamma} \mathbf{q})^* \cdot \hat{\mathbf{n}} d\mathbf{x} - \mathcal{M}^k \mathbf{z}^k = \\ (\mathcal{S}_x^k)^T \mathbf{a}_x^k + (\mathcal{S}_y^k)^T \mathbf{a}_y^k - \int_{\partial \mathbf{D}^k} \ell_j^k \mathbf{a}^* \cdot \hat{\mathbf{n}} d\mathbf{x}, \quad (26)$$

147 Here, we choose the central flux for the right-hand side, i.e. $\mathbf{a}^* = \{\{\mathbf{a}\}\}$
 148 together with the interior penalty (IP) flux for the elliptic operator, i.e.
 149 $z^* = \{\{z\}\}$, $(\sqrt{\gamma} \mathbf{q})^* = \{\{\sqrt{\gamma} \nabla z\}\} - \tau[z]$, $\tau > 0$. The point of the penalty
 150 term is to penalize large jumps at the element interfaces. These equations
 151 can be combined to find a single equation in terms of the unknown variable
 152 z . This technique is known as the method of Bassi & Rebay ?, where the
 153 auxiliary variable is introduced in order to facilitate a DG-FEM formulation,
 154 but in the final stage the auxiliary variables are eliminated locally.

155 2.4. Filter Stabilization of Aliasing-driven Instabilities

156 The governing equations do not possess any viscosity terms and thus
 157 lack any physical energy dissipation mechanism. As a result, the quadratic
 158 nonlinearity terms can cause energy to accumulate at the small scales in
 159 an unphysical manner. Additionally, aliasing errors that occur due to the
 160 “pointwise product” treatment of the nonlinear terms can drive weak nu-
 161 matical instabilities that can destroy the numerical solutions as explained
 162 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 wavenumber filter of the form

$$\sigma(n) = \begin{cases} 1, & 0 \leq n < N_{crit} \\ \exp\left(-\alpha \left(\frac{n-N_{crit}}{N-N_{crit}}\right)^s\right), & N_{crit} \leq n \leq N \end{cases} \quad (27)$$

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

169 3. Curvilinear Elements

170 In addition to solution singularities, it is also known that the conver-
 171 gence rates of a high-order method may be limited to sub-optimal rates

172 as a result of an inaccurate representation of the boundary. This fact was
 173 demonstrated in Hesthaven and Warburton (2008) who found poor conver-
 174 gence rates for the solution to Maxwell's equations on a circular domain with
 175 a piece-wise linear representation of the boundary. Dupont (2001) has also
 176 suggested that rounding singular corners is necessary to suppress poor poly-
 177 nomial behaviour resulting from the high-order DG-FEM in his inter-model
 178 comparison of the oceanic shallow water equations. It is thus apparent that
 179 a high-order method begs for a smooth and accurate representation of the
 180 boundary, and hence, deformed or curvilinear elements along the boundary
 181 will be necessary to achieve accurate solutions on general lake geometries
 182 with the high-order discontinuous Galerkin method.

183 *3.1. Constructing coordinates systems for curvilinear elements*

184 We have adopted the approach in Hesthaven and Warburton (2008) that
 185 avoids some of the difficulties and cumbersome work associated with explic-
 186 itly constructing two-dimensional mapping functions for high-order curvilinear
 187 elements, e.g., explicitly calculating high-order “shape-functions”. The
 188 technique discussed here generalizes well to elements with an arbitrary num-
 189 ber of nodes and thus allows for the robust construction of high-order curvi-
 190 linear elements. The method represents an extension of the technique used
 191 in Hesthaven and Warburton (2008) for circular boundaries, since we con-
 192 sider arbitrary domain boundaries represented by cubic splines.

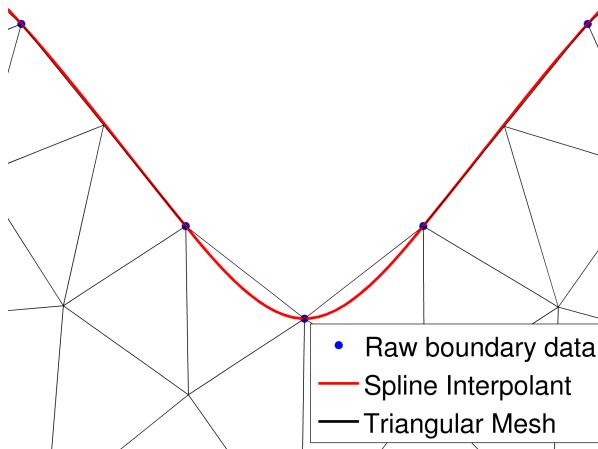


Figure 1: Illustration of straight-sided element mesh along with a smooth representation of the boundary, the spline interpolant, that will be used to produce deformed elements.

193 Assume we have generated a straight-sided finite element mesh that ap-
 10

proximates the boundary in a piece-wise linear manner, and also assume we have retained a smooth representation of the boundary in a parameterized curve \mathcal{C} : $\mathbf{x}_b(t) = (x_b(t), y_b(t))$ (see Fig. 1), that we assume to be parameterized by arc-length $0 \leq t \leq S$. 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-Lobatto (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.,

$$f(x) = \left(\frac{x - x_0}{x_1 - x_0} \right) f_1 . \quad (28)$$

²¹⁹ In a sense, we have found the appropriate blending function to be $\ell_1(x)$
²²⁰ 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 $b(r = -1, s) = 1$ 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 $s = -r$ for $-1 \leq r \leq 1$, leading us to define the blending function as

$$b(r, s) = \left(\frac{s + r}{s - 1} \right), \quad (29)$$

²²¹ that satisfies $b(r = -1, s) = 1$ and $b(r, s = -r) = 0$, as required. The one
²²² issue that remains is the apparent singularity at the point $(-1, 1)$. This
²²³ point corresponds to a location where $\mathbf{w} = 0$ since it is a vertex of the finite
²²⁴ element mesh that does not need to be deformed. Thus, we can simply
²²⁵ apply the blending at nodal points not corresponding to the singular point
in step 4 above.

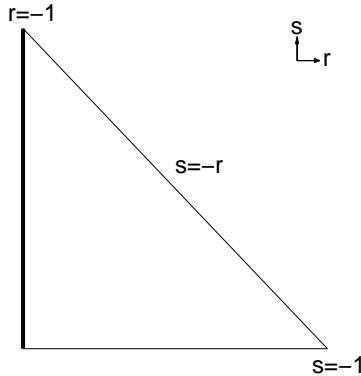


Figure 2: Diagram of the reference triangle and illustration of (r, s) coordinates.

²²⁶
²²⁷ While the “blending” procedure discussed is a straight-forward extension
²²⁸ of linear Lagrange interpolation to two-dimensions, one subtle difference be-
²²⁹ tween Lagrange interpolation is that the two-dimensional blending function
²³⁰ is chosen to be zero or one along entire *line segments*, and not at points
²³¹ in space. It is for this reason that the technique has been referred to as

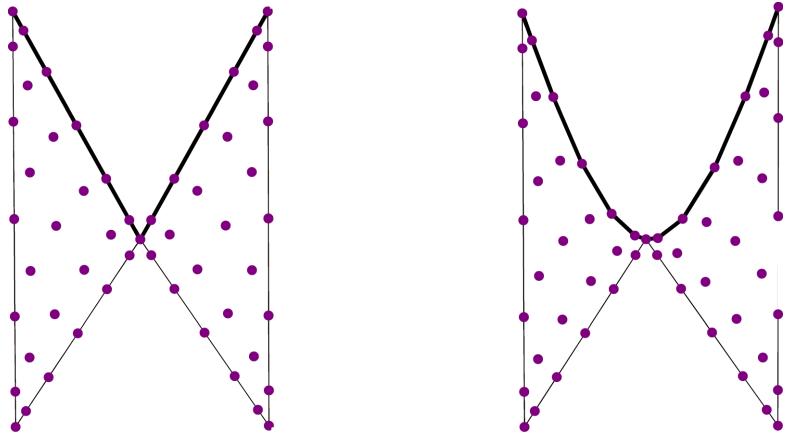


Figure 3: **Left:** A pair of elements before being deformed. **Right:** The same elements after being deformed to match the cubic-spline representation of the boundary with interior nodes re-distributed via Gordon–Hall blending.

“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.

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 hence 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 $3(N+1)$.

254 A general inner-product of two functions f and g is thus evaluated as

$$\int_{\mathbf{D}^k} fg d\mathbf{x} \approx \sum_{i=1}^{N_c} f(\mathbf{r}_i^c) g(\mathbf{r}_i^c) J_i^k w_i^c , \quad (30)$$

255 where J_i^k is the Jacobian of the mapping from the standard element D^k ,
256 and w_i^c are the cubature weights associated with cubature nodes $\{\mathbf{r}_i^c\}_{i=1}^{N_c}$.
257 The cubature nodes and weights are provided by the symmetric rules of
258 Wandzura and Xiao (2003) and implemented in MATLAB in Hesthaven
259 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

$$\mathcal{M}_{lm}^k = \int_{\mathbf{D}^k} \ell_l^k(\mathbf{x}) \ell_m^k(\mathbf{x}) d\mathbf{x} \quad (31)$$

$$\approx \sum_{i=1}^{N_c} \ell_l(\mathbf{r}_i^c) \ell_m(\mathbf{r}_i^c) w_i^c J_i^{k,c} , \quad (32)$$

Hence,

$$\mathcal{M}^k = (\mathcal{V}^c)^T \mathcal{W}^k \mathcal{V}^c . \quad (33)$$

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,

$$\mathcal{S}_{x,nm}^k = \int_{\mathbf{D}^k} \ell_n^k(\mathbf{x}) \frac{\partial \ell_m^k}{\partial x}(\mathbf{x}) d\mathbf{x} \quad (34)$$

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

$$\mathcal{S}_x^k = (\mathcal{V}^c)^T \mathcal{W}^k (\text{diag}(r_x^k(\mathbf{r}_i^c)) \mathcal{V}^c \mathcal{D}_r + \text{diag}(s_x^k(\mathbf{r}_i^c)) \mathcal{V}^c \mathcal{D}_s) . \quad (35)$$

An identical argument gives

$$\mathcal{S}_y^k = (\mathcal{V}^c)^T \mathcal{W}^k (\text{diag}(r_y^k(\mathbf{r}_i^c)) \mathcal{V}^c \mathcal{D}_r + \text{diag}(s_y^k(\mathbf{r}_i^c)) \mathcal{V}^c \mathcal{D}_s) . \quad (36)$$

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

269 4. Results and Discussion

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

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

278 Unlike the 1D comparison carried out in Steinmoeller et al. (2012), it
 279 should be noted that the comparison between the Fourier method and the
 280 DG methods at various orders here is not a “fair” one, since the number of
 281 degrees of freedom has not been held fixed in all cases. Here, the point is
 282 to illustrate that spectral-like resolution characteristics can become possible
 283 with increasing polynomial orders on a fixed finite element mesh triangula-
 284 tion. The mesh used here consists of $K = 1330$ triangular elements, and
 285 it was chosen to directly correspond with the annular-shaped lake of Stein-
 286 moeller et al. (2013). The mesh, shown in Fig. 5 (b) was generated with the
 287 `mesh2d` MATLAB algorithm that uses an adaptive Delaunay-based triangu-
 288 lation algorithm implemented using quadtrees. Modal filtering was applied
 289 using an exponential cut-off filter (27). Here, the cut-off polynomial order
 290 was set to $N_c = 3$ and the filter order was set to $s = 4$, and the same
 291 filtering parameters were used in both the $N = 4$ and $N = 8$ cases.

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

$$E = \iint_{\Omega} \frac{1}{2} h(u^2 + v^2) + \frac{1}{2} g' \eta^2 dA \quad (37)$$

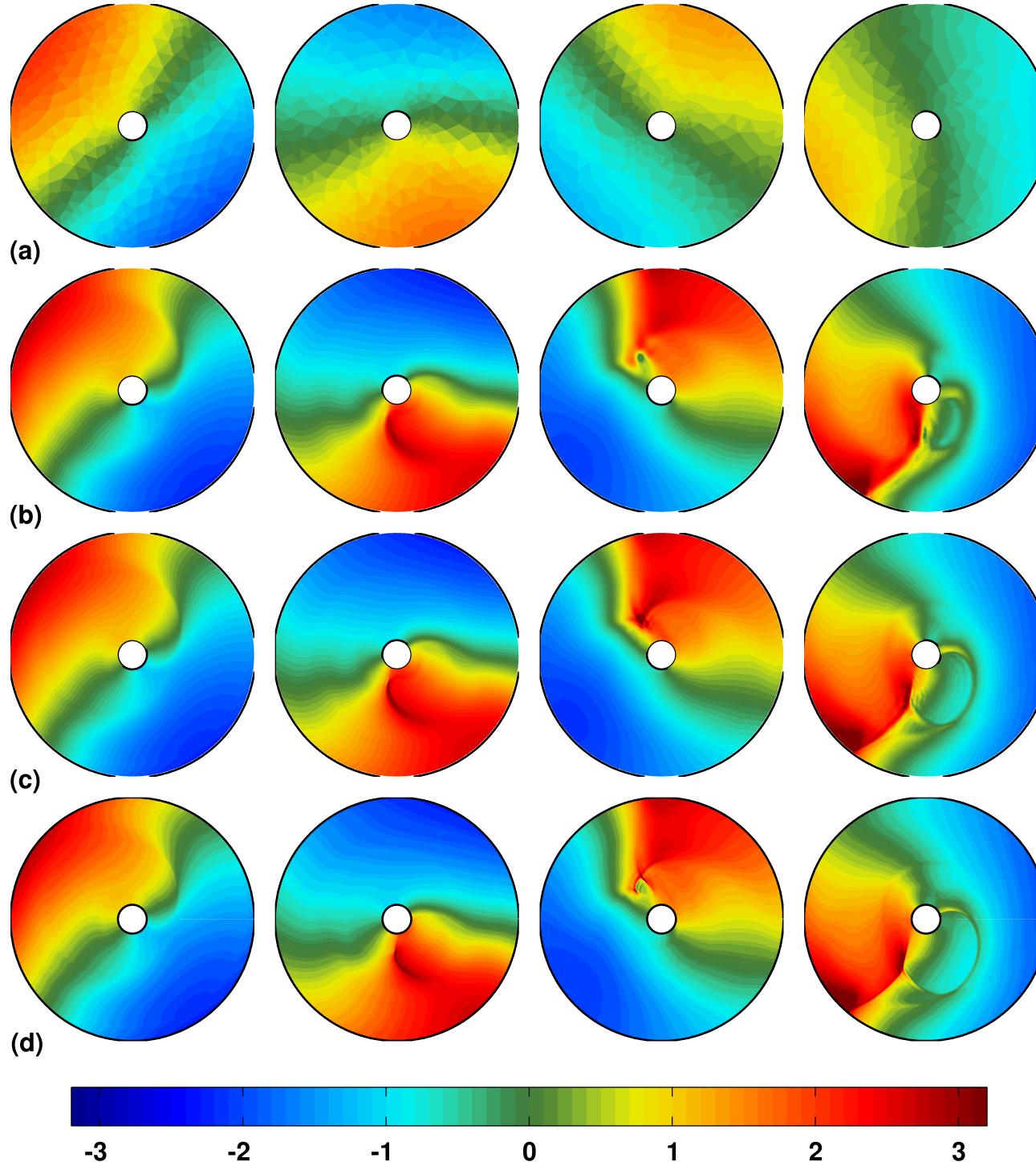


Figure 4: Comparison between the DG-FEM method at orders (a) $N = 1$, (b) $N = 4$, and (c) $N = 8$ to the Fourier-Chebyshev pseudospectral method (row (d)) for the simulation presented in Steinmoeller et al. (2013) with $N_r \times N_\theta = 256 \times 1024$ points. The number of elements in the DG 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.

292 (scaled by its initial value) against time. We see that the initial energy is
 293 rapidly dissipated in the low-order $N = 1$ case. The $N = 4$ and $N = 8$ cases
 294 exhibit nearly identical energy profiles, however a more detailed view would
 295 reveal that the $N = 8$ line is slightly above the $N = 4$ line. As expected,
 296 the Fourier–Chebyshev method outperforms all of the DG methods with the
 297 least energy lost. Once again, we illustrate here the utility of pseudospectral
 298 methods as a benchmark numerical method. These results also validate the
 299 DG-FEM method since the details of the nonlinear wave fronts in Figure 4
 300 are reasonably represented for orders $N \geq 4$, and the amount of numerical
 301 dissipation approaches that of the pseudospectral method for increasing N
 (see Figure 5 (a)).
 302

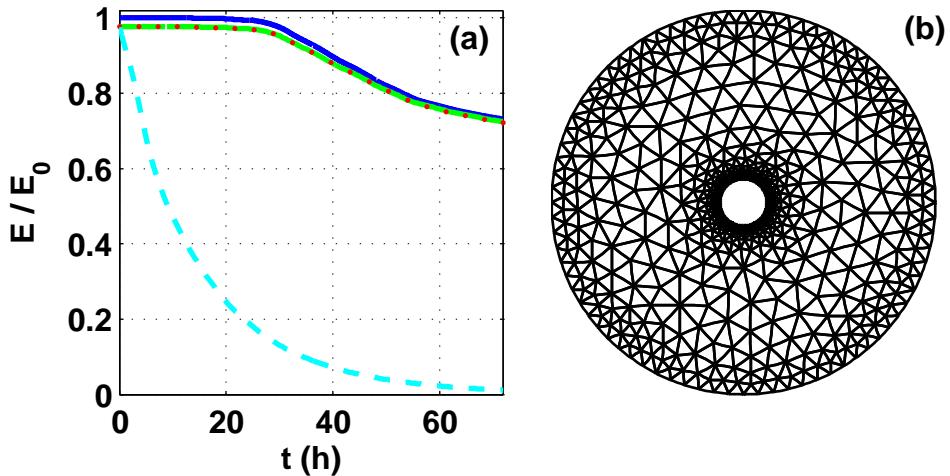


Figure 5: Panel (a): Scaled domain-integrated total energy (E/E_0) time-series for the simulations in Fig. 4. The lines correspond to the Fourier–Chebyshev method with 256×1024 points (blue, solid), DG-FEM with $N = 8$ (green, solid), DG-FEM with $N = 4$ (red, dots), DG-FEM with $N = 1$ (cyan, dashed). Panel (b): Finite element mesh with $K = 1330$ elements used in the DG-FEM simulations.
 302

303 6. Spurious eddies in inviscid DG-FEM solutions

304 While exploring other geometries with the DG-FEM code, it was found
 305 that under certain conditions spurious eddies, corresponding to an unphys-
 306 ical production of vorticity, appear to form in the domain near obstacles
 17

307 resembling re-entrant corners (i.e., boundary corners that protrude inwards
 308 into the domain). This effect is illustrated in Figure 6 where our annular
 309 basin has been perturbed to include a peninsula. The DG-FEM solver with
 310 polynomial order $N = 4$ was initialized with the initial conditions used in
 311 Section 5. A numerical instability occurred shortly after $t = 27$ h, prevent-
 312 ing further time-stepping, though the reason for the instability was evident
 earlier due to the sharp gradients visible near the corner in Figure 6.

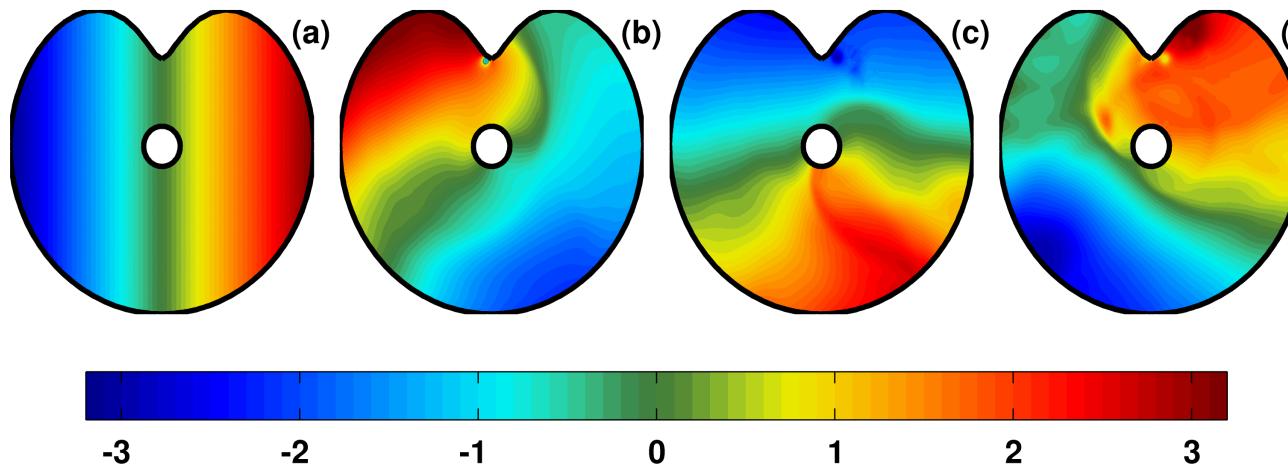


Figure 6: Snapshots of the η -field in the order $N = 4$ DG-FEM simulation of a rotating seiche on a perturbed circular domain with a re-entrant peninsula at (a) $t = 0$ h, (b) $t = 6.8$ h, (c) $t = 14.0$ h, (d) $t = 20.9$ h. Note the apparent separation eddies visible near the peninsula in panels (b)–(d).

313
 314 The eddies bear a striking resemblance to boundary-layer separation
 315 eddies that would occur due to flow past an obstacle in viscous flow (see
 316 Kundu and Cohen (2008)). However, since our model equations do not
 317 contain any viscous terms, the formation of a viscous boundary-layer is
 318 not possible and hence boundary-layer separation should not be possible.
 319 These spurious eddies are thus artifacts, and appear to coincide with the
 320 presence of a sharp re-entrant corner. Even in the cases where the actual
 321 boundary is smooth, re-entrant corners at the element-scale may result as a
 322 consequence of the piece-wise linear representation of the boundary assumed
 323 in mesh generation. Although these artifacts are spurious in the sense that
 324 inviscid flow around an obstacle should not separate, from a theoretical
 325 stand-point they should be expected. Below, we explain why this is the
 326 case and propose methods for remedying the situation.

In order to gain some insight from theory, consider the simpler situation of inviscid, incompressible and irrotational (i.e., potential) flow. Under these constraints, the velocity field has both a velocity potential and a streamfunction, i.e.,

$$\mathbf{u} = \nabla\phi = \nabla^\perp\psi , \quad (38)$$

where $\nabla^\perp = \mathbf{k} \times \nabla = \left(-\frac{\partial}{\partial y}, \frac{\partial}{\partial x}\right)$, where both ϕ and ψ satisfy Laplace's equation. The no-normal flow boundary conditions imply that

$$\frac{\partial\phi}{\partial n} = 0 , \quad \text{on } \partial\Omega , \quad (39)$$

$$\psi = \text{constant} , \quad \text{on } \partial\Omega . \quad (40)$$

It is useful to define a complex velocity potential $w(z) = \phi + i\psi$, where $z = x + iy$, so that

$$\frac{dw}{dz} = \frac{\partial(\phi + i\psi)}{\partial x} = u - iv . \quad (41)$$

It is known that the complex potential for flow around a wall angle $\alpha = \pi/n$ is given by Kundu and Cohen (2008):

$$w = Az^n , \quad \left(n \geq \frac{1}{2}\right) . \quad (42)$$

In polar coordinates, $z = re^{i\theta}$ this yields

$$\phi = Ar^n \cos(n\theta) , \quad \psi = Ar^n \sin(n\theta) . \quad (43)$$

Noticing that $\psi = 0$ for $\theta = 0, \pi/n$ and using the fact that any streamline along which $\psi = \text{constant}$ can represent a boundary, we recognize that this particular complex potential corresponds to flow around a sharp corner with angle $\theta = \pi/n$.

At this point, we can recover the velocity using equation (41)

$$u = \Re(nAz^{n-1}) = \Re\left(\frac{A\pi}{\alpha}z^{\frac{\pi-\alpha}{\alpha}}\right) , \quad (44)$$

$$v = \Im(nAz^{n-1}) = \Im\left(\frac{A\pi}{\alpha}z^{\frac{\pi-\alpha}{\alpha}}\right) . \quad (45)$$

Paying close attention to the expression on the right-hand side, it is clear that the exponent to which z is raised is negative whenever the wall-angle exceeds π , and thus the velocity is not defined when $z = 0$ (the corner).

334 Even in the case where $0 < \alpha < \pi$, the point $z = 0$ is a singular point
 335 since the velocity's derivatives do not exist. We thus reach the main point
 336 of this discussion, that is also highlighted by Kundu and Cohen (2008): the
 337 velocity at a wall-corner is infinite if the wall-angle is greater than 180°
 338 and is zero for wall-angles less than 180° . Thus, near re-entrant corners
 339 the numerical solution should be *expected* to be poorly behaved since the
 340 exact potential flow solution is also. Although the velocity derivatives do
 341 not exist at corners less than 180° , this does not appear to be an issue for
 342 the numerical solution. [Derek: we can probably axe most of the equations
 343 in the discussion above, save the $w = \dots$ business, as long as it's clear the
 344 things break for big angles.]

The “spurious eddies” encountered in simulations begin as very steep free-surface depressions that diffuse away from the boundary, to understand this phenomenon, consider Bernoulli’s equation

$$p + \frac{1}{2}\rho_0\|\mathbf{u}\|^2 = \text{const.}, \quad (46)$$

where p is the pressure. Taking the gradient and re-arranging gives

$$-\nabla p = \frac{1}{2}\rho_0\nabla(\|\mathbf{u}\|^2). \quad (47)$$

345 Since the flow is infinite at the corner corresponding to $z = 0$, the upstream
 346 fluid must accelerate as it approaches the corner and decelerate as it moves
 347 away from it. This fact, together with equation (47), implies that $-\nabla p$
 348 points in the direction with the flow upstream of the corner (i.e., a favourable
 349 pressure gradient) and *against* the flow downstream of the corner (i.e., an
 350 adverse pressure gradient). Hence, a region of very low pressure must exist
 351 in the vicinity of the corner. It is thus clear then why the singularity
 352 would manifest itself as a large (localized) free-surface depression since in
 353 the shallow water framework $p = \rho_0 g \eta$. In general, we found that the
 354 free-surface depression is advected downstream of the corner immediately
 355 after appearing, since the numerical method cannot compensate for the
 356 singularity.

357 In real flow around a corner, the region of adverse pressure gradient
 358 would cause the flow to separate from the corner resulting in the formation
 359 of eddies due to vorticity produced in the viscous boundary layer (again,
 360 see Kundu and Cohen (2008)). In the DG-FEM simulations discussed, the
 361 observed eddies are a result of the local modal filtering that attempts to
 362 stabilize the pressure singularity by diffusing it away from the boundary,

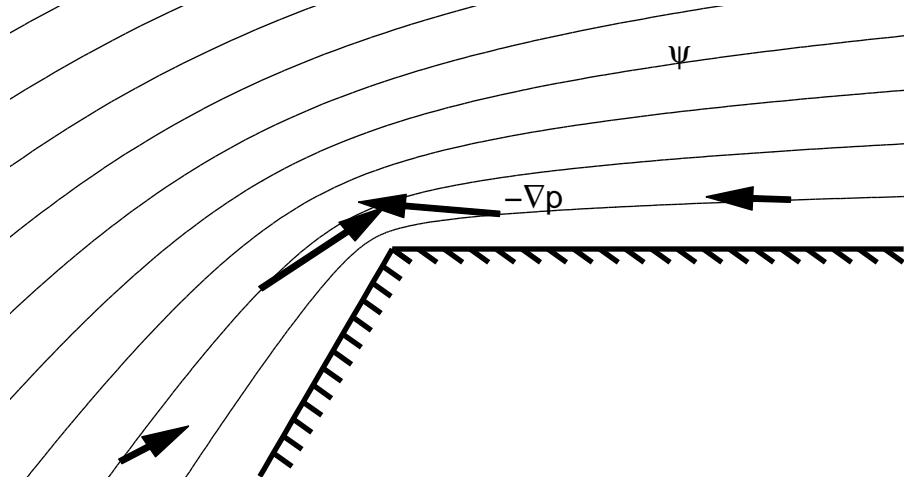


Figure 7: Cartoon diagram of potential flow around a wall corner of angle $\alpha > \pi$. Contours depict streamlines (lines where $\psi = \text{const.}$) and arrows illustrate the relative strength and direction of $-\nabla p$.

363 taking over the role of viscosity in realistic flows. This effect of the filter
 364 was discovered by turning off the filter and observing singular growth at the
 365 corner that led to numerical blow-up with no eddy introduced. It was also
 366 found that spurious eddy generation is more prominent in simulations where
 367 nonlinear effects are non-negligible. The fact that a standard filter coupled
 368 with the presence of re-entrant corners will typically lead to spurious eddies
 369 is a dangerous feature of the numerical model, since a modeller may be led
 370 to believe that these eddies are physical, when in fact they are the result of
 371 the filter's action on a part of the solution that is singular. For instance, in
 372 Zhang et al. (2012), spurious eddies due to a limiter are presented as physical
 373 for the situation of supersonic compressible flow past an equilateral triangle.
 374 Despite the effort of filtering, it has been found that this singular behaviour
 375 can still lead to numerical blow-up. Thus, some effort must be taken to
 376 remedy this problem, as discussed below.

377 7. Internal Rotating Seiche Simulation using curvilinear elements

378 We now consider the same simulation shown in Section 6 where a circular
 379 basin has been perturbed to include a peninsula. The difference here is that

380 we employ the developments on curvilinear elements described in the above
 381 sections along with polynomial order $N = 8$. All boundary elements have
 382 been deformed such that their boundary edges conform to a cubic spline
 383 interpolant of the boundary.

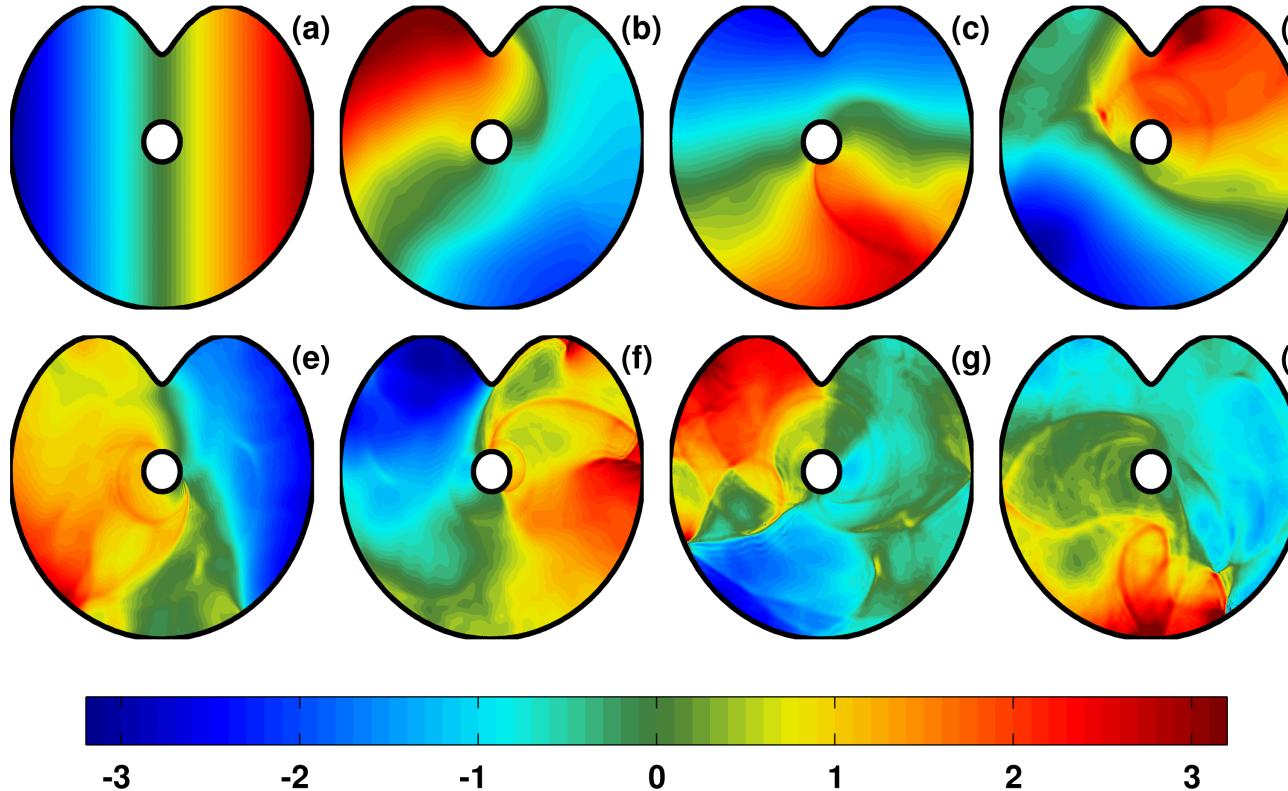


Figure 8: Panels (a)–(d): Like Figure 6 but with curvilinear elements along the boundary. The other panels correspond to the later times (e) $t = 28.1$ h, (f) $t = 34.9$ h, (g) $t = 42.1$ h, (h) $t = 49.0$ h.

384 The results of the simulation are shown in Figure 8. In addition to
 385 finding that the simulation is apparently long-term stable, unlike in the
 386 straight-sided case, we also note that the spurious eddies associated with
 387 the sharp re-entrant corner have been suppressed since the peninsula is now
 388 represented in a geometrically smooth way.

389 It is important to note that although the spurious eddies have been
 390 suppressed, the region of the flow at the tip of the peninsula still represents
 391 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 6, 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.

8. 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.agls.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 9 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 9(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 9(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 the DG-FEM solver unstable.

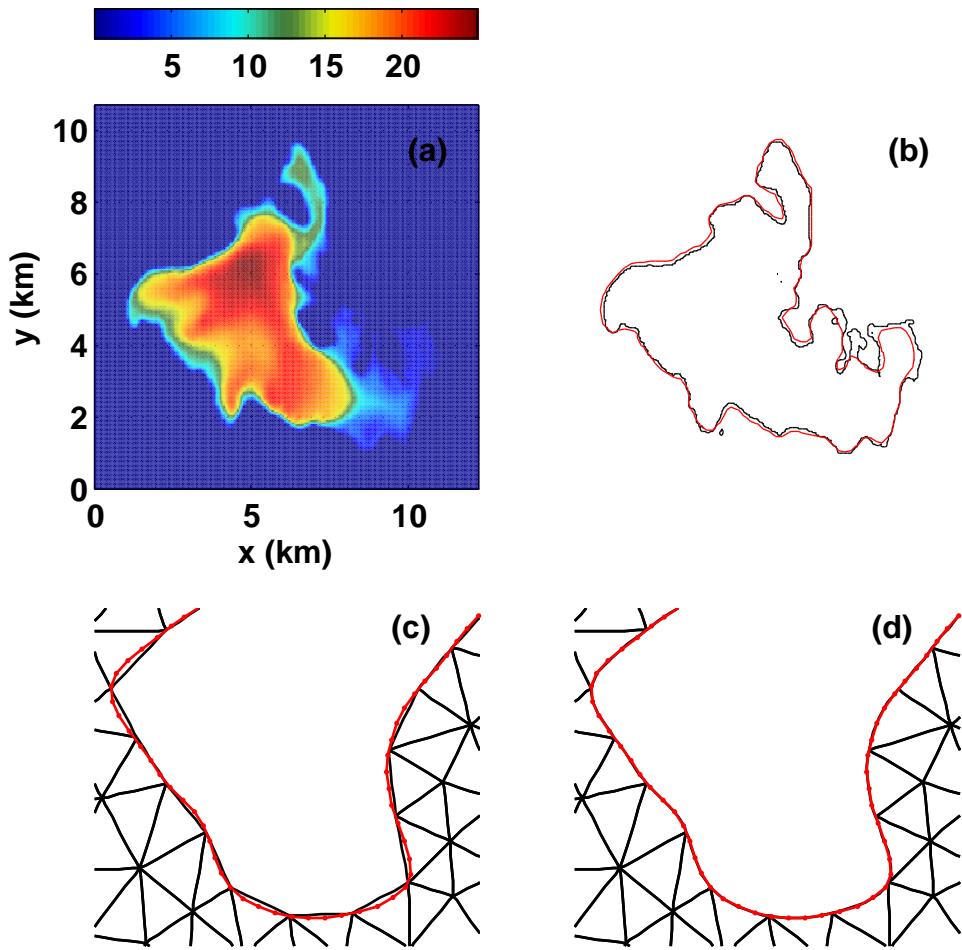


Figure 9: Panel (a): Depth (in m) of Pinehurst Lake, AB from raw 50 m bathymetry data, and panel (b): corresponding $H = 0$ contour (black) with smoothed coastline super-imposed (red). The lower panels show a zoomed-in section of the (c) straight-sided and (d) curved ($N = 6$) finite element mesh with $K = 1807$ elements near $(x, y) = (7 \text{ km}, 5 \text{ km})$ with cubic spline interpolant super-imposed (red).

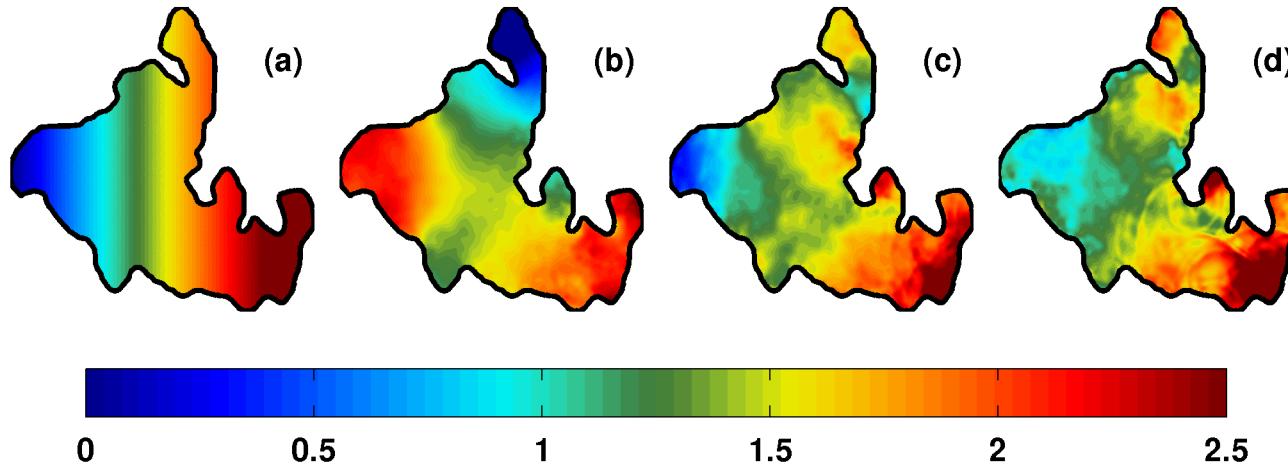


Figure 10: Evolution of an interfacial tilt in Pinehurst Lake, AB using the $N = 6$ DG-FEM with curvilinear boundary elements at times (a) $t = 0$ h, (b) $t = 19.4$ h, (c) $t = 39.3$ h, (d) $t = 62.7$ h.

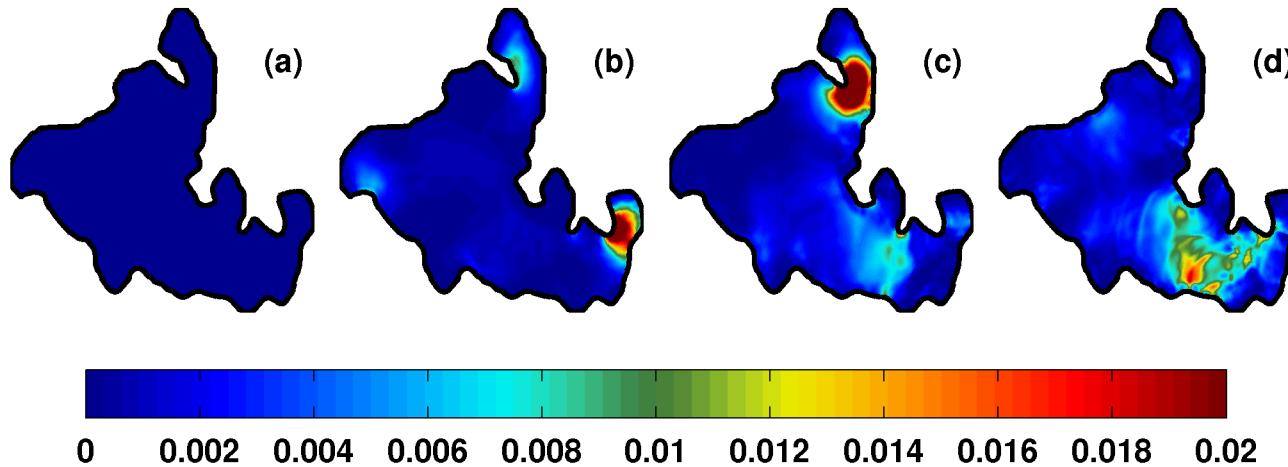


Figure 11: Like Fig. 10, except the kinetic energy density, $\frac{1}{2}h(u^2 + v^2)$ is plotted.

429 Here, the reduced gravity is $g' = (\Delta\rho/\rho_0)g = 0.024525 \text{ ms}^{-2}$, where
 430 $(\Delta\rho/\rho_0) = 0.0025$. The Coriolis parameter was taken to be $f = 1.1863 \times$
 431 10^{-4} s^{-1} , corresponding to the 54.65° N latitude of Pinehurst Lake. Results
 432 of an $N = 6$ DG-FEM simulation from an initial east-west interfacial tilt
 433 taken to increase linearly from $\eta = 0$ to $\eta = 2.5 \text{ m}$ are shown in Figure 10
 434 that illustrates the evolving density interface at fixed-time snapshots with

435 the initial condition plotted in panel **(a)**. Since the relative amplitude of
436 the initial condition compared to the depth is, on average, not as large as
437 in previous simulations in this manuscript, nonlinear effects are expected
438 to be weaker. In spite of this fact, panels **(c)** and **(d)** show that nonlinear
439 waves emerge in the shallows in the southeastern part of the basin after
440 sufficient time has passed. As a result, small scale waves have proliferated
441 throughout the entire basin by $t = 62.7$ h.

442 Figure 10 should be compared closely to Figure 11 where the kinetic
443 energy density is plotted at the same times. At the earlier times (panels **(b)**
444 and **(c)**), the most energetic features correspond to attached flow around
445 peninsulas or other coastal obstacles. It is apparent that geometric focusing
446 intensifies such features when they occur in narrow, confined parts of the
447 basin. Panel **(d)** illustrates the kinetic energy fingerprint of small scale
448 internal wave activity localized in the shallow eastern end of the lake at
449 later times.

450 9. Conclusions

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

457 The basics of the DG-FEM integral formulation for the dispersive shal-
458 low water system and its reduction to local matrix operators was discussed
459 in Section Appendix A. The time-stepping method used with the DG-FEM,
460 that closely mirrors the scalar approach from Steinmoeller et al. (2012,
461 2013); Eskilsson and Sherwin (2005), was presented in Section ??.

462 Comparisons between the pseudospectral method of Steinmoeller et al.
463 (2013) and the DG-FEM methods developed here were carried out in Sec-
464 tion 5. From these comparisons, we conjecture that the DG-FEM can reach
465 comparable resolution and energy-conserving characteristics to the Fourier-
466 Chebyshev methods for sufficiently high polynomial order N .

467 It was demonstrated that the DG-FEM is poorly behaved in the neigh-
468 borhood of sharp re-entrant corners in Section 6, since sharp gradients and
469 spurious eddies appear. An explanation of this phenomenon in terms of
470 potential flow theory was offered. The remedy of rounding the corners

471 using curvilinear elements along the boundary was proposed and the imple-
 472 mentation was explained in Section 3. The conclusion we draw from that
 473 discussion is that general coastlines need a more computationally expensive
 474 treatment than simple circular geometries since the integrals in the DG-
 475 FEM formulations must be evaluated with cubature and quadrature rules
 476 of higher order than the approximating basis polynomials.

477 Finally, applications using the curvilinear element methodology were
 478 carried out. In Section 7, it was illustrated that the spurious eddies reported
 479 in Section 3 did not manifest when the re-entrant corner was represented
 480 in a smooth manner. The same methodology was then applied to the real-
 481 world situation of Pinehurst Lake, Alberta. The resulting high-resolution
 482 numerical solution was able to pinpoint a hot-spot of small-scale wave activ-
 483 ity in the shallow eastern end of the basin. From this, we conclude that the
 484 DG-FEM solution of a weakly non-hydrostatic layered model may be a use-
 485 ful tool in helping to identify regions in lakes where internal wave-induced
 486 mixing is most dominant. The ability to identify such hot-spots has strong
 487 ecological consequences, as noted by Pannard et al. (2011).

488 **Appendix A. Details of the Discontinuous Galerkin Spatial Dis-
 489 cretization Method**

Since the DG-FEM method is primarily suited to solving hyperbolic conservation laws, it is useful to consider the augmented system (1)-(3) in the conservation form

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{B} + \mathbf{C} + \mathbf{N}, \quad (\text{A.1})$$

with

$$\mathbf{Q} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}. \quad (\text{A.2})$$

The terms

$$\mathbf{B} = gh \begin{pmatrix} 0 \\ \frac{\partial H}{\partial x} \\ \frac{\partial H}{\partial y} \end{pmatrix}, \quad \mathbf{C} = f \begin{pmatrix} 0 \\ vh \\ -uh \end{pmatrix}, \quad \mathbf{N} = \frac{H^2}{6} \begin{pmatrix} 0 \\ \frac{\partial z}{\partial x} \\ \frac{\partial z}{\partial y} \end{pmatrix}, \quad (\text{A.3})$$

are the bed slopes, Coriolis terms, and the dispersive terms, respectively. As explained in Section ??, The variable $z = \nabla \cdot (\mathbf{u}h)_t$ is governed by the

Helmholtz problem (5) restated here as

$$\nabla \cdot (\gamma \nabla z) - z = -\nabla \cdot \mathbf{a}, \quad (\text{A.4})$$

490 with $\gamma = H^2/6$ and \mathbf{a} is as defined previously by equation (6). In the case
491 of the traditional non-rotating hydrostatic shallow water model with a flat
492 bottom, the right-hand side of (A.1) vanishes.

493 To apply the DG-FEM method in two-dimensions, we first assume that
494 the domain Ω can be triangulated using K elements (or sub-domains) and
495 that the triangulation is assumed to be geometrically conforming. That
496 is, the boundary $\partial\Omega$ is represented by a piece-wise linear approximation
497 with each line segment belonging to a side of a triangle. The boundary can
498 be approximated in a piece-wise curvilinear sense by using triangles with
499 curved edges as explored later in Section 3, but this approach is typically
500 much more expensive and does not lend itself easily to the nodal approach of
501 Hesthaven and Warburton (2008) owing mainly to the fact that the mapping
502 Jacobian to the reference triangle is non-constant. Additionally, we assume
503 that the nodes along a triangle edge that are shared between two elements
504 are duplicated, so as to ensure that a purely local scheme can be recovered.
505 This is a fundamental difference between DG-FEM and FEM, which uses
506 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

$$h_h^k(\mathbf{x}, t) = \sum_{i=1}^{N_p} h_h^k(\mathbf{x}_i^k, t) \ell_i^k(\mathbf{x}), \quad (\text{A.5})$$

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

The nodal DG-FEM weak integral form statement is obtained by substituting the approximate local solutions into each of equations (A.1), 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 and focus only on (A.1) for the moment, this gives

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

where $\hat{\mathbf{n}}$ is the unit outward-pointing normal. Due to the fact that 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 $(\mathbf{F}^+, \mathbf{G}^+)$. Since we have not explicitly imposed continuity at element interfaces, 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)). The choice of numerical flux considered in this manuscript is the local Lax–Friedrichs (L-F) flux

$$(\hat{n}_x \mathbf{F}_h + \hat{n}_y \mathbf{G}_h)^* = \hat{n}_x \{\!\{ \mathbf{F}_h \}\!\} + \hat{n}_y \{\!\{ \mathbf{G}_h \}\!\} + \frac{\lambda}{2} [\![\mathbf{Q}_h]\!] , \quad (\text{A.7})$$

where

$$\{\!\{ \mathbf{u} \}\!\} = \frac{\mathbf{u}^- + \mathbf{u}^+}{2}, \quad [\![\mathbf{u}]\!] = \hat{\mathbf{n}}^- \cdot \mathbf{u}^- + \hat{\mathbf{n}}^+ \cdot \mathbf{u}^+ , \quad (\text{A.8})$$

are the average and jump in \mathbf{u} across the interface, respectively. The numerical flux choice (A.7) represents the local Lax-Friedrichs flux where λ is an approximation to the maximum linearized wave speed

$$\lambda = \max_{\mathbf{s} \in [\mathbf{Q}_h^-, \mathbf{Q}_h^+]} \left(\|\mathbf{u}(\mathbf{s})\| + \sqrt{gh(\mathbf{s})} \right) . \quad (\text{A.9})$$

In order to reduce the statement (??) to a form useful for numerical computations, it is important to rewrite it in terms of matrices wherever possible. As an illustration, consider the first component of (??) with nodal expansions of the form (A.5) explicitly substituted in. We can then write this first component as

$$\mathcal{M}^k \frac{d\mathbf{h}^k}{dt} = -\mathcal{S}_x^k (\mathbf{h}\mathbf{u})^k - \mathcal{S}_y^k (\mathbf{h}\mathbf{v})^k + \int_{\partial\mathbf{D}^k} \ell_j^k ((hu)_h^k - (hu)^*, (hv)_h^k - (hv)^*) \cdot \hat{\mathbf{n}} d\mathbf{x} \quad (\text{A.10})$$

where

$$\mathbf{h}^k = [h_h^k(\mathbf{x}_1) \cdots h_h^k(\mathbf{x}_{N_p})]^\top , \quad (\text{A.11})$$

$$(\mathbf{h}\mathbf{u})^k = [(hu)_h^k(\mathbf{x}_1) \cdots (hu)_h^k(\mathbf{x}_{N_p})]^\top , \quad (\text{A.12})$$

$$(\mathbf{h}\mathbf{v})^k = [(hv)_h^k(\mathbf{x}_1) \cdots (hv)_h^k(\mathbf{x}_{N_p})]^\top , \quad (\text{A.13})$$

and we have left the surface integral contribution alone for now. Here the local mass matrix is given by

$$\mathcal{M}_{ij}^k = \int_{\mathbf{D}^k} \ell_i^k(\mathbf{x}) \ell_j^k(\mathbf{x}) d\mathbf{x} = J^k \int_{\mathbf{I}} \ell_i(\mathbf{r}) \ell_j(\mathbf{r}) d\mathbf{r} = J^k \mathcal{M}, \quad (\text{A.14})$$

where $J^k = x_r^k y_s^k - x_s^k y_r^k$ 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}_x^k is given as follows:

$$\mathcal{S}_{x,ij}^k = \int_{\mathbf{D}^k} \ell_i^k(\mathbf{x}) \frac{\partial \ell_j^k}{\partial x} d\mathbf{x} = J^k \int_{\mathbf{I}} \ell_i(\mathbf{r}) \left(\frac{\partial \ell}{\partial r} r_x^k + \frac{\partial \ell}{\partial s} s_x^k \right) d\mathbf{r}, \quad (\text{A.15})$$

$$= J^k \int_{\mathbf{I}} \ell_i(\mathbf{r}) \left(\frac{\partial \ell}{\partial r} \frac{y_s^k}{J^k} - \frac{\partial \ell}{\partial s} \frac{y_r^k}{J^k} \right) d\mathbf{r}, \quad (\text{A.16})$$

$$= y_s^k \mathcal{S}_r - y_r^k \mathcal{S}_s \quad (\text{A.17})$$

where we have used the fact that the Jacobian matrices have the inverse property, i.e.,

$$\frac{\partial \mathbf{x}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \begin{bmatrix} x_r & x_s \\ y_r & y_s \end{bmatrix} \begin{bmatrix} r_x & r_y \\ s_x & s_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (\text{A.18})$$

hence,

$$r_x = \frac{y_s}{J}, \quad r_y = -\frac{x_s}{J}, \quad s_x = -\frac{y_r}{J}, \quad s_y = \frac{x_r}{J}, \quad (\text{A.19})$$

in going from (A.15) to (A.16). Similarly, for \mathcal{S}_y^k , we have

$$\mathcal{S}_{y,ij}^k = \int_{\mathbf{D}^k} \ell_i^k(\mathbf{x}) \frac{\partial \ell_j^k}{\partial y} d\mathbf{x} = -x_s^k \mathcal{S}_r + x_r^k \mathcal{S}_s. \quad (\text{A.20})$$

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

$$\mathcal{S}_{r,ij} = \int_{\mathbf{I}} \ell_i(\mathbf{r}) \frac{\partial \ell_j}{\partial r} d\mathbf{r}, \quad \mathcal{S}_{s,ij} = \int_{\mathbf{I}} \ell_i(\mathbf{r}) \frac{\partial \ell_j}{\partial s} d\mathbf{r}. \quad (\text{A.21})$$

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.

525 *Appendix A.1. Evaluating the Inner Products: Modal approach vs. Nodal
526 approach*

We follow Hesthaven and Warburton's discussion by introducing a modal expansion for each solution field as an alternative to the nodal representations (e.g. (A.5)). For example, for an arbitrary field $u(\mathbf{r})$ defined on \mathbf{I} , we have

$$u(\mathbf{r}) \approx u_h(\mathbf{r}) = \sum_{n=1}^{N_p} \hat{u}_n \psi_n(\mathbf{r}) = \sum_{i=1}^{N_p} u(\mathbf{r}_i) \ell_i(\mathbf{r}), \quad (\text{A.22})$$

where $\{\psi(\mathbf{r})\}_{i=1}^{N_p}$ 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.,

$$\int_{\mathbf{I}} u(\mathbf{r}) \psi_m(\mathbf{r}) d\mathbf{r} = \sum_{n=1}^{N_p} \hat{u}_n \int_{\mathbf{I}} \psi_n(\mathbf{r}) \psi_m(\mathbf{r}) d\mathbf{r}, \quad (\text{A.23})$$

or, in matrix-vector notation,

$$\mathbf{v} = \mathcal{H}\hat{\mathbf{u}}, \quad (\text{A.24})$$

where

$$\hat{\mathbf{u}} = [\hat{u}_1, \dots, \hat{u}_{N_p}], \quad \mathcal{H}_{ij} = \int_{\mathbf{I}} \psi_i \psi_j d\mathbf{r}, \quad \mathbf{v}_i = \int_{\mathbf{I}} u \psi_i d\mathbf{r}. \quad (\text{A.25})$$

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

$$\psi_m(\mathbf{r}) = \sqrt{2} P_i(a) P_j^{(2i+1,0)}(b)(1-b)^i, \quad (\text{A.26})$$

where

$$a = 2 \frac{1+r}{1-s} - 1, \quad b = s, \quad (\text{A.27})$$

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

$$N_p = \sum_{i=1}^{N+1} i = \binom{N+2}{2}, \quad (\text{A.28})$$

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 (A.23). If the numerical method uses a purely modal approach, one may consider using a cubature (2D quadrature) formula at the nodes, i.e.

$$\hat{u}_n \approx \sum_{i=1}^{N_p} u(\mathbf{r}_i) \psi_n(\mathbf{r}_i) w_i, \quad (\text{A.29})$$

where the r_i 's must be taken to be cubature points, and the w_i 's are the associated cubature weights. Although the modal DG method is not considered in detail in this thesis, we later employ cubature and quadrature integration rules as a means of removing aliasing errors when curvilinear elements are employed in Section 3.

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.,

$$u_h(\mathbf{r}_i) = \sum_{n=1}^{N_p} \hat{u}_n \psi_n(\mathbf{r}_i). \quad (\text{A.30})$$

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

$$\mathcal{V}\hat{\mathbf{u}} = \mathbf{u}, \quad (\text{A.31})$$

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 (A.31) with the uniqueness statement (A.22), one can obtain the following useful formula for the Lagrange polynomials in terms of the basis polynomials

$$\ell_i(\mathbf{r}) = \sum_{n=1}^{N_p} (\mathcal{V}^\top)_{in}^{-1} \psi_n(\mathbf{r}). \quad (\text{A.32})$$

⁵³⁴ **Appendix B. Local Operators for the Nodal Approach**

We can now apply all of the developments of the above section to the evaluation of the inner products posed by equations (A.14) and (A.21). Substituting the formula (A.32) into the expression for the standard local mass matrix, we recover

$$\mathcal{M} = (\mathcal{V}\mathcal{V}^T)^{-1}. \quad (\text{B.1})$$

Before considering the stiffness matrices, it is useful to define the differentiation matrices

$$\mathcal{D}_{r,ij} = \left. \frac{\partial \ell_j}{\partial r} \right|_{\mathbf{r}_i}, \quad \mathcal{D}_{s,ij} = \left. \frac{\partial \ell_j}{\partial s} \right|_{\mathbf{r}_i} \quad (\text{B.2})$$

whose entries may be furnished directly by appropriate differentiation of the formula (A.32). It can be shown that the local stiffness matrices can be recovered by

$$\mathcal{M}\mathcal{D}_r = \mathcal{S}_r, \quad \mathcal{M}\mathcal{D}_s = \mathcal{S}_s. \quad (\text{B.3})$$

In other words,

$$\mathcal{D}_r = \mathcal{M}^{-1}\mathcal{S}_r, \quad \mathcal{D}_s = \mathcal{M}^{-1}\mathcal{S}_s. \quad (\text{B.4})$$

⁵³⁵ This is useful because it implies that an explicit semi-discrete scheme can be
⁵³⁶ obtained by multiplying (A.10) by $(\mathcal{M}^k)^{-1} = \frac{1}{j^k} \mathcal{M}^{-1}$. As a consequence of
⁵³⁷ the fact that the local mass matrix only varies by a constant factor on each
⁵³⁸ element, it follows that this operation is computationally cheap since \mathcal{M} is
⁵³⁹ an $N_p \times N_p$ matrix. For example, with order $N = 8$ basis functions, the local
⁵⁴⁰ mass matrix is a 45×45 full matrix. This is another key difference between
⁵⁴¹ DG-FEM and the classical FEM, where explicit semi-discrete schemes often
⁵⁴² cannot be recovered since the time-derivative operator is multiplied by the
⁵⁴³ global mass matrix, that may be large and expensive to invert explicitly.

⁵⁴⁴ *Appendix B.1. Surface Integral Contributions*

To close our numerical scheme, it remains to discuss the surface integral term in equation (A.10)

$$\int_{\partial\mathbf{D}^k} \ell_j^k(\mathbf{x}) \mathbf{g}_h \cdot \hat{\mathbf{n}} d\mathbf{x} \quad (\text{B.5})$$

where $\mathbf{g}_h = ((hu)_h^k - (hu)^*, (hv)_h^k - (hv)^*)$ 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

$$\int_{\partial\mathbf{D}^k} \ell_j^k(\mathbf{x}) \mathbf{g}_h \cdot \hat{\mathbf{n}} d\mathbf{x} = \sum_{e=1}^3 \hat{\mathbf{n}}_e \cdot \int_{\text{edge}_e} \ell_j^k(\mathbf{x}) \mathbf{g}_h d\mathbf{x}. \quad (\text{B.6})$$

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

$$\sum_{e=1}^3 \sum_{i=1}^{N+1} \hat{\mathbf{n}} \cdot \mathbf{g}_i \mathcal{M}_{ij}^{k,e}, \quad (\text{B.7})$$

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

$$\mathcal{M}_{ij}^{k,e} = \int_{\text{edge}_e} \ell_j^k(\mathbf{x}) \ell_i^k(\mathbf{x}) d\mathbf{x} = J^{k,e,1} \mathcal{M}^1, \quad (\text{B.8})$$

545 where $J^{k,e,1}$ is the Jacobian of the mapping from the edge to the standard
 546 interval $[-1, 1]$. Using the 1D developments in Hesthaven and Warburton
 547 (2008), the standard 1D mass matrix is related to the Vandermonde matrix
 548 for 1D polynomial interpolation by $\mathcal{M}^1 = (\mathcal{V}^1(\mathcal{V}^1)^T)^{-1}$.

549 Appendix C. 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

$$h^+ = h^-, \quad (\text{C.1})$$

$$hu^+ = hu^- - 2(n_x hu + n_y hv)n_x, \quad (\text{C.2})$$

$$hv^+ = hv^- - 2(n_x hu + n_y hv)n_y, \quad (\text{C.3})$$

550 The second and third conditions are equivalent to imposing no normal $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$
 551 along the wall. The first condition is equivalent to imposing $\nabla h \cdot \hat{\mathbf{n}} = 0$
 552 which states that the interface should be parallel to the bathymetry at the
 553 wall.

554 Appendix D. Bathymetry and Non-hydrostatic terms

555 So far, we have not discussed the treatment of the bathymetry and
 556 nonhydrostatic terms contained in the vectors \mathbf{B} and \mathbf{N} , respectively. We
 557 have avoided these terms so far because they cannot be addressed by the
 558 standard nodal DG-FEM treatment.

As an example of the issues that arise, let us 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

$$\int_{\mathbf{D}^k} gh^k \frac{\partial H^k}{\partial x} \ell_j(x) d\mathbf{x} - \int_{\partial \mathbf{D}^k} gh (H^k - H^*) d\mathbf{x}, \quad (\text{D.1})$$

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, on the other hand 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 in for H , yielding (after index relabelling)

$$\begin{aligned} \int_{\mathbf{D}^k} gh^k \frac{\partial H^k}{\partial x} \ell_j(\mathbf{x}) d\mathbf{x} &= \sum_{j=1}^{N_p} g H^k(x_j) \int_{\mathbf{D}^k} h^k \ell_i(\mathbf{x}) \frac{\partial \ell_j}{\partial x} d\mathbf{x}, \\ &= \sum_{j=1}^{N_p} g H^k(x_j) \mathcal{S}_{ij}^{k,h}, \end{aligned} \quad (\text{D.2})$$

$$= g \mathcal{S}^{k,h} \mathbf{H}^k, \quad (\text{D.3})$$

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 now different on every element and must be updated after each time-step. Although this approach is computationally expensive in general, it should not be completely disregarded since it becomes necessary in situations where curvilinear elements are used since the mapping Jacobian is no longer constant. See Sections 3–3.2.

To pursue a less expensive approach, let us introduce the auxiliary variable

$$\kappa(\mathbf{x}) = \frac{\partial H}{\partial x}. \quad (\text{D.4})$$

Following previous discussion, we can approximate κ with the DG-FEM method by

$$\mathcal{M}^k \boldsymbol{\kappa} = \mathcal{S}_x \mathbf{H}^k - \int_{\partial \mathbf{D}^k} (H^k - H^*) n_x d\mathbf{x}, \quad (\text{D.5})$$

or,

$$\boldsymbol{\kappa} = \mathcal{D}_x \mathbf{H}^k - (\mathcal{M}^k)^{-1} \int_{\partial \mathbf{D}^k} (H^k - H^*) n_x d\mathbf{x}. \quad (\text{D.6})$$

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

$$\int_{\mathbf{D}^k} gh^k(\mathbf{x}) \kappa^k(\mathbf{x}) \ell_j(\mathbf{x}) d\mathbf{x}. \quad (\text{D.7})$$

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

$$h^k(\mathbf{x}) \kappa^k(\mathbf{x}) \approx \sum_{i=1}^{N_p} h^k(\mathbf{x}_i) \kappa^k(\mathbf{x}_i) \ell_i^k(\mathbf{x}), \quad (\text{D.8})$$

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

$$\begin{aligned} \int_{\mathbf{D}^k} gh^k \frac{\partial H^k}{\partial x} \ell_j(\mathbf{x}) d\mathbf{x} &\approx \sum_{j=1}^{N_p} g \kappa^k(\mathbf{x}_j) h^k(\mathbf{x}_j) \int_{\mathbf{D}^k} \ell_i(\mathbf{x}) \ell_j(\mathbf{x}) d\mathbf{x}, \\ &= g \mathcal{M}^k (\boldsymbol{\kappa} \mathbf{h})^k, \end{aligned} \quad (\text{D.9})$$

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. As discussed above for the pseudospectral methods in section 2.4, modal filtering can be used to prevent these aliasing errors from driving weak instabilities. The form of the filter used is precisely the same as in section 2.4 with the exception that it is applied to the 2D basis on each element.

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 (A.4) within the DG-FEM framework below.

586 **Appendix E. DG-FEM for elliptic problems**

587 At first, it may not be clear how the DG-FEM can be applied to second-
 588 order elliptic equations since such equations are not hyperbolic and thus
 589 do not have well-posed Riemann problems that may be considered to im-
 590 pose weak continuity across element interfaces. However, it is explained in
 591 Hesthaven and Warburton (2008) that elliptic equations may be recast as
 592 a first-order system of equations and appropriate numerical fluxes can be
 593 obtained from penalty methods.

To re-write (A.4) as a first-order system, we introduce the auxiliary variable

$$\mathbf{q} = (q_x, q_y) = \sqrt{\gamma} \nabla z , \quad (\text{E.1})$$

yielding the system

$$\nabla \cdot (\sqrt{\gamma} \mathbf{q}) - z = -\nabla \cdot \mathbf{a} , \quad (\text{E.2})$$

$$q_x = \sqrt{\gamma} \frac{\partial z}{\partial x} , \quad (\text{E.3})$$

$$q_y = \sqrt{\gamma} \frac{\partial z}{\partial y} . \quad (\text{E.4})$$

594 Inspecting the system (E.2)-(E.4) it may be unclear how, given an input
 595 right-hand side $-\nabla \cdot \mathbf{a}$, one can recover z . This is generally achieved by
 596 considering the inverse situation, i.e., if z is known, then \mathbf{q} can be com-
 597 puted by solving equations (E.3)-(E.4), and $-\nabla \cdot \mathbf{a}$ can be recovered using
 598 (E.2). This set of operations can be considered a non-singular linear trans-
 599 formation, and hence there must exist an inverse transformation.

The strong DG formulation of (E.3)–(E.4) together with the weak for-
 mulation of (E.2) is given by

$$\mathcal{M}^k \mathbf{q}_x^k = \sqrt{\gamma}^k \mathcal{S}_x \mathbf{z}^k - \sqrt{\gamma}^k \int_{\partial \mathbf{D}^k} \ell_j^k (z^k - z^*) n_x d\mathbf{x} , \quad (\text{E.5})$$

$$\mathcal{M}^k \mathbf{q}_y^k = \sqrt{\gamma}^k \mathcal{S}_y \mathbf{z}^k - \sqrt{\gamma}^k \int_{\partial \mathbf{D}^k} \ell_j^k (z^k - z^*) n_y d\mathbf{x} , \quad (\text{E.6})$$

$$\begin{aligned} -(\mathcal{S}_x^k)^T (\sqrt{\gamma} \mathbf{q}_x)^k - (\mathcal{S}_y^k)^T (\sqrt{\gamma} \mathbf{q}_y)^k + \int_{\partial \mathbf{D}^k} \ell_j^k (\sqrt{\gamma} \mathbf{q})^* \cdot \hat{\mathbf{n}} d\mathbf{x} - \mathcal{M}^k \mathbf{z}^k = \\ (\mathcal{S}_x^k)^T \mathbf{a}_x^k + (\mathcal{S}_y^k)^T \mathbf{a}_y^k - \int_{\partial \mathbf{D}^k} \ell_j^k \mathbf{a}^* \cdot \hat{\mathbf{n}} d\mathbf{x} \end{aligned} \quad (\text{E.7})$$

600 Here, we choose the central flux for the right-hand side, i.e. $\mathbf{a}^* = \{\{\mathbf{a}\}\}$
 601 together with the interior penalty (IP) flux for the elliptic operator, i.e.
 602 $z^* = \{\{z\}\}$, $(\sqrt{\gamma}\mathbf{q})^* = \{\{\sqrt{\gamma}\nabla z\}\} - \tau[\![z]\!]$, $\tau > 0$. The point of the penalty
 603 term is to penalize large jumps at the element interfaces. If $\tau = 0$, a
 604 numerical calculation of the eigenfunctions of the Laplacian would reveal
 605 a spurious $\lambda = 0$ mode with all elements completely de-coupled, and the
 606 system would be singular (see Hesthaven and Warburton (2008)). The use
 607 of the penalty term pushes the spurious eigenmode out of the operator's
 608 null space to guarantee invertibility. In general, a sufficiently large penalty
 609 parameter will suppress any other spurious modes to the high- λ part of
 610 the eigenspectrum as well. This property represents an advantage over
 611 continuous Galerkin discretizations of elliptic operators that often possess
 612 spurious *convergent* modes whose corresponding eigenvalues can lie within
 613 the physical range of the eigenspectrum. Modes of this type were discussed
 614 in an ocean modelling context by Cotter et al. (2009).

Other flux possibilities for DG-discretized Laplacian and Helmholtz operators include the penalized central flux $z^* = \{\{z\}\}$, $(\sqrt{\gamma}\mathbf{q})^* = \{\{\sqrt{\gamma}\mathbf{q}\}\} - \tau[\![z]\!]$ and the local discontinuous Galerkin (LDG) flux $z^* = \{\{z\}\} + \hat{\mathbf{n}}[\![z]\!]$, $(\sqrt{\gamma}\mathbf{q})^* = \{\{\sqrt{\gamma}\mathbf{q}\}\} - [\![\sqrt{\gamma}\mathbf{q}]\!] \cdot \hat{\mathbf{n}} - \tau[\![z]\!]$. The central flux should not be used in general since it only has optimal convergence characteristics for even polynomial orders N , and its matrix-form is not as sparse as the other choices (i.e., large stencil). The LDG flux, on the other hand, has optimal convergence rates at all orders and offers the most sparse representation, but it is known to be the most poorly conditioned operator. The IP flux offers a balance between these two, giving optimal convergence at all orders, a middle-ground in terms of sparsity, and similar condition numbers to the central-flux operator (see Hesthaven and Warburton (2008)). Furthermore, with some algebraic manipulations, the auxiliary variable \mathbf{q} can be eliminated locally, and this allows the operator to be efficiently set-up directly as a symmetric sparse matrix. Such an elimination gives the following statement in terms of local operators

$$\begin{aligned}
 & - \left((\mathcal{D}_x^k)^T \mathcal{M}^k \Gamma^k (\mathcal{M}^k)^{-1} \Gamma^k \mathcal{M}^k \mathcal{D}_x^k + (\mathcal{D}_y^k)^T \mathcal{M}^k \Gamma^k (\mathcal{M}^k)^{-1} \Gamma^k \mathcal{M}^k \mathcal{D}_y^k + \mathcal{M}^k \right) \mathbf{z}^k \\
 & + \sum_{e=1}^3 (\mathcal{D}_n^{k,e})^T \mathcal{M}^{k,e} \Gamma^{k,e} (\mathcal{M}^{k,e})^{-1} \Gamma^{k,e} \mathcal{M}^{k,e} \left(\frac{\mathbf{z}^- - \mathbf{z}^+}{2} \right) \\
 & + \sum_{e=1}^3 \mathcal{M}^{k,e} \left[\Gamma^{k,e} \left(\mathcal{D}_n^{k,e} \left(\frac{\mathbf{z}^- - \mathbf{z}^+}{2} \right) + \tau(\mathbf{z}^- - \mathbf{z}^+) \right) \right] = \text{RHS} ,
 \end{aligned} \tag{E.8}$$

615 where Γ^k is the diagonal matrix with the entries of $\sqrt{\gamma^k}$ written along its
616 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
617 along edge e of element k .

618 The discontinuous Galerkin IP discretization method has become known
619 as the ‘symmetric interior penalty discontinuous Galerkin’ (SIP-DG) method
620 in the literature, and has been applied to the pressure Poisson equation and
621 viscous operator of the incompressible Navier–Stokes equations (see Ferrer
622 and Willden (2011); Shahbazi et al. (2007)).

623 Acknowledgement

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

626 References

- 627 Bassi, F., Rebay, S., 1997. A high-order accurate discontinuous finite element method for
628 the numerical solution of the compressible Navier-Stokes equations. *J. Comp. Phys.* 131, 267–279.
629
630 Brandt, P., Rubino, A., Alpers, W., Backhaus, J., 1997. Internal waves in the Strait of
631 Messina studied by a numerical model and synthetic aperture radar umages from *ERS*
632 *1/2* Satellites. *J. Phys. Oceangr.* 27, 648–663.
633 Cockburn, B., Hou, S., Shu, C., 1990. The Runge–Kutta local projection discontinuous
634 Galerkin finite-element method for conservation-laws. 4. The multidimensional case.
635 *Math. Comp.* 54 (190), 545–581.
636 Cockburn, B., Shu, C., 1989. TVB Runge–Kutta local projection discontinuous Galerkin
637 finite-element method for conservation-laws. 2. General framework. *Math. Comp.*
638 52 (186), 411–435.
639 Cotter, C. J., Ham, D. A., Pain, C. C., Reich, S., 2009. LBB stability of a mixed Galerkin
640 finite element pair for fluid flow simulations. *J. Comp. Phys.* 228 (2), 336–348.
641 de la Fuente, A., Shimizu, K., Imberger, J., Niño, Y., 2008. The evolution of internal
642 waves in a rotating, stratified, circular basin and the influence of weakly nonlinear and
643 nonhydrostatic accelerations. *Limnol. Oceanogr.* 53 (6), 2738–2748.
644 Dupont, F., 2001. Comparison of numerical methods for modelling ocean circulation in
645 basins with irregular coasts. Ph.D. thesis, McGill University.
646 Eskilsson, C., Sherwin, S., 2005. Spectral/*hp* discontinuous Galerkin methods for mod-
647 ellling 2D Boussinesq equations. *J. Sci. Comp.* 22, 269–288.
648 Ferrer, E., Willden, R., 2011. A high order Discontinuous Galerkin Finite Element solver
649 for the incompressible Navier–Stokes equations. *Comput. Fluids* 46, 224–230.
650 Geuzaine, C., Remacle, J.-F., 2009. A three-dimensional finite element mesh generator
651 with built-in pre- and post-processing facilities. *Int. J. Numer. Methods Eng.* 79, 1309–
652 1331.
653 Gordon, W., Hall, C., 1973. Construction of curvilinear coordinate systems and applica-
654 tion to mesh generation. *Int. J. Numer. Meth. Eng.* 7, 461–477.

- 655 Hesthaven, J., Warburton, T., 2008. Nodal Discontinuous Galerkin Methods. Springer.
- 656 Hughes, T., 1987. Recent progress in the development and understanding of SUPG meth-
657 ods with special reference to the compressible euler and navier-stokes equations. Int.
658 J. Numer. Method. Fluids 7, 1261–1275.
- 659 Karniadakis, G., Sherwin, S., 2005. Spectral/hp Element Methods for Computational
660 Fluid Dynamics, 2nd Edition. Oxford University Press, USA.
- 661 Kundu, P., Cohen, I., 2008. Fluid Mechanics, 4th Edition. Elsevier Academic Press.
- 662 Leveque, R., 2007. Finite Difference Methods for Ordinary and Partial Differential Equa-
663 tions: Steady-State and Time-Dependent Problems. Society for Industrial and Applied
664 Mathematics.
- 665 Levin, J. C., Iskandarani, M., Haidvogel, D. B., 2006. To continue or discontinue: Com-
666 parisons of continuous and discontinuous Galerkin formulations in a spectral element
667 ocean model. Ocean Modelling 15 (1-2), 56–70.
- 668 Mandli, K., 2011. Finite volume methods for the multilayer shallow water equations
669 with applications to storm surges. Ph.D. thesis, University of Washington, Seattle,
670 Washington, USA.
- 671 Nazarov, M., Popov, B., 2014 (submitted). A posteriori error estimation for the com-
672 pressible euler equations using entropy viscosity. Comp. & Fluids.
- 673 Pannard, A., Beisner, B., Bird, D., Braun, J., Planas, D., Bormans, M., 2011. Recurrent
674 internal waves in a small lake: Potential ecological consequences for metalimnetic
675 phytoplankton populations. L&O:F&E 1, 91–109.
- 676 Rhebergen, S., Cockburn, B., 2012. A space–time hybridizable discontinuous Galerkin
677 method for incompressible flows on deforming domains. J. Comp. Phys. 231, 4185–
678 4204.
- 679 Shahbazi, K., Fischer, P., Ethier, C., 2007. A high-order discontinuous Galerkin method
680 for the unsteady incompressible Navier–Stokes equations. J. Comp. Phys 222, 391–407.
- 681 Steinmoeller, D., Stastna, M., Lamb, K., 2012. Fourier pseudospectral methods for 2D
682 Boussinesq-type equations. Ocean Modelling 52–53, 76–89.
- 683 Steinmoeller, D., Stastna, M., Lamb, K., 2013. Pseudospectral methods for Boussinesq-
684 type equations in an annular domain with applications to mid-sized lakes. J. Comp.
685 Sci. 4, 3–11.
- 686 Toro, E., 1999. Riemann Solvers and Numerical Methods for Fluid Dynamics, 2nd Edi-
687 tion. Springer.
- 688 Walkley, M., 1999. A numerical method for extended boussinesq shallow-water wave
689 equations. Ph.D. thesis, University of Leeds, UK.
- 690 Wandzura, S., Xiao, H., 2003. Symmetric quadrature rules on a triangle. Comput. Math.
691 Applica. 45, 1829–1840.
- 692 Zhang, X., Xia, Y., Shu, C.-W., 2012. Maximum-principle-satisfying and positivity-
693 preserving high order Discontinuous Galerkin schemes for conservation laws on tri-
694 angular meshes. J. Sci. Comput. 50, 29–62.