Ocean Dynamics (2010) 60:1415–1445
DOI 10.1007/s10236-010-0351-x

High-order schemes for 2D unsteady biogeochemical ocean models

Mattheus P. Ueckermann · Pierre F. J. Lermusiaux

Received: 20 June 2010 / Accepted: 19 October 2010 / Published online: 16 November 2010

Abstract Accurate numerical modeling of biogeochemical ocean dynamics is essential for numerous applications, including coastal ecosystem science, environmental management and energy, and climate dynamics. Evaluating computational requirements for such often highly nonlinear and multiscale dynamics is critical. To do so, we complete comprehensive numerical analyses, comparing low- to high-order discretization schemes, both in time and space, employing standard and hybrid discontinuous Galerkin finite element methods, on both straight and new curved elements. Our analyses and syntheses focus on nutrient–phytoplankton–zooplankton dynamics under advection and diffusion within an ocean strait or sill, in an idealized 2D geometry. For the dynamics, we investigate three biological regimes, one with single stable points at all depths and two with stable limit cycles. We also examine interactions that are dominated by the biology, by the advection, or that are balanced. For these regimes and interactions, we study the sensitivity to multiple numerical parameters including quadrature-free and quadrature-based discretizations of the source terms, order of the spatial discretizations of advection and diffusion operators, order of the temporal discretization in explicit schemes, and resolution of the spatial mesh, with and without curved elements. A first finding is that both quadrature-based and quadrature-free discretizations give accurate results in well-resolved regions, but the quadrature-based scheme has smaller errors in under-resolved regions. We show that low-order temporal discretizations allow rapidly growing numerical errors in biological fields. We find that if a spatial discretization (mesh resolution and polynomial degree) does not resolve the solution, oscillations due to discontinuities in tracer fields can be locally significant for both low- and high-order discretizations. When the solution is sufficiently resolved, higher-order schemes on coarser grids perform better (higher accuracy, less dissipative) for the same cost than lower-order scheme on finer grids. This result applies to both passive and reactive tracers and is confirmed by quantitative analyses of truncation errors and smoothness of solution fields. To reduce oscillations in un-resolved regions, we develop a numerical filter that is active only when and where the solution is not smooth locally. Finally, we consider idealized simulations of biological patchiness. Results reveal that higher-order numerical schemes can maintain patches for long-term integrations while lower-order schemes are much too dissipative and cannot, even at very high resolutions. Implications for the use of simulations to better understand biological blooms, patchiness, and other nonlinear reactive dynamics in coastal regions with complex bathymetric features are considerable.

Responsible Editor: Vincent Legat

Electronic supplementary material The online version of this article (doi:10.1007/s10236-010-0351-x) contains supplementary material, which is available to authorized users.

M. P. Ueckermann · P. F. J. Lermusiaux (✉)
Massachusetts Institute of Technology,
77 Mass. Avenue, Cambridge, MA 02139, USA
e-mail: pierrel@mit.edu

M. P. Ueckermann
e-mail: mpuecker@mit.edu

Keywords Ocean straits · Shelfbreaks · Biogeochemical dynamics · High order · Curved boundary · Quadrature · Ocean modeling · Discontinuous Galerkin

1 Introduction

Accurate modeling of biogeochemical–physical ocean dynamics is required for multiple scientific and societal applications, covering a wide range of time and space scalves. With the increased understanding of biogeochemical interactions (Lalli and Parsons 1997; Robinson et al. 2002; Fennel and Neumann 2004), ecosystems models have substantially improved in the past decades (Fasham et al. 1990; Hofmann and Lascara 1998; Robinson and Lermusiaux 1999; Hofmann and Friedrichs 2002; Lynch et al. 2009). Coupled biogeochemical–physical models have been used from coastal regions (e.g., Anderson et al. 2005; Spitz et al. 2005; Ji et al. 2008; Stow et al. 2009) to basins and global ocean domains (e.g., Oschlies and Garcon 1998; Rothstein et al. 2006; Doney et al. 2009). However, in light of the strong nonlinearities observed in biological processes, an important subject that has been largely overlooked is the numerical requirements for such simulation studies. One of the major objectives of our work is to address such computational questions for reactive ocean tracers, directly including the latest advances in computational fluid dynamics (e.g., Chung 2002; Ferziger and Peric 2002; Lomax et al. 2003; Cebeci et al. 2005; Karniadakis and Sherwin 2005) and multiscale ocean modeling (Deleersnijder and Lermusiaux 2008).

Previous numerical ocean studies related to ours have primarily focused on passive or dynamic (density-related) tracer advections. The most significant progress include the results of Hecht et al. (1995), Hanert et al. (2004), and Budgell et al. (2007), but none of these advances has dealt with higher-order advection of reactive tracers on unstructured meshes with curved geometries. Iskandarani et al. (2005) applied and studied high-order schemes for passive tracer and density dynamics in two dimensions, including Hecht et al. (1995)'s test and the gravitational adjustment of density in a channel of constant depth (Haidvogel and Beckmann 1999), but they did not consider curved elements. Lévy et al. (2001) assessed five different low-order finite volume advection schemes for biological modeling and found a 30% difference in new production estimates, highlighting the need for careful numerical studies. In Bernard et al. (2009), high-order discontinuous Galerkin (DG) methods are used to solve tidal flows around shallow water islands with non-trivial geometries and using curved triangular meshes. Here, we are interested in biogeochemical tracers with possibly highly nonlinear reactive or source terms, and we compare a set of low- to high-order schemes, both in time and in space. We employ the DG finite element method (Cockburn 1998), using both straight and curved elements, and we study a varied set of numerical properties. As in previous computational studies, we restrict our numerical analyses to 2D flows, focusing on coupled dynamics in idealized straits.

Our ultimate dynamics motivation is to allow quantitative simulation studies of fundamental nonlinear biological–physical dynamics in coastal regions with complex bathymetric features such as straits, sills, ridges, and shelfbreaks. Such features strongly affect flows, and if they are shallow enough, one can expect biological responses in the euphotic zone. Multiple physical scales are possible, from rapid tidal effects to slow water-mass-driven overflows, and biological resonances at some of these scales are likely. Our focus is on the numerical requirements prerequisite to such studies. Our work is partly inspired by our experience in coastal regions with complex geometries (Haley and Lermusiaux 2010), especially with steep shelfbreaks such as the Massachusetts Bay and Stellwagen Bank (Besiktepe et al. 2002), Middle Atlantic Bight shelfbreak (Lermusiaux 1999), Monterey Bay shelfbreak (Haley et al. 2009), Taiwan region shelfbreak (Lermusiaux and Xu 2010), and Philippine Archipelago Straits (Haley and Lermusiaux 2010). The latter effort particularly motivated the present work, within the context of the Philippines Experiment (PhilEx) which is a 5-year joint research project focused on interdisciplinary modeling, data assimilation, and dynamical studies in the straits regions of the Philippine Archipelago to better understand, model, and predict sub-mesoscale and mesoscale physical and biogeochemical dynamics in complex regions. For realistic PhilEx simulations, we employ our MIT Multidisciplinary Simulation, Estimation, and Assimilation Systems (MSEAS-Group 2010). It includes a free surface hydrostatic ocean model over complex geometries with novel implicit schemes for telescoping nesting (Haley and Lermusiaux 2010). This physical model is coupled to biological models (Besiktepe et al. 2002), forced with multiscale barotropic tides (Logutov and Lermusiaux 2008), and initialized with new objective mapping schemes specific for multiconnected domains (Agarwal 2009; Agarwal and Lermusiaux 2010, in press). The multiresolution nested domains cover very shallow regions with strong tides, steep bathymetries, and the deep ocean. The MSEAS system was employed in real time, assimilating data sets from ships, gliders, and satellite remote sensing and issuing daily physical–biological forecasts with dynamical descriptions and adaptive sampling guidance (Lermusiaux et al. 2009). The complex, nonlinear, and multiscale biology in the region confirmed the need for the present computational studies.

Our work is part of an incubation for the next generation of ocean modeling systems, focusing on key numerical questions for biogeochemical dynamics. The biological model we employ is based on Flierl and McGillicuddy (2002), Burton (2009), and Ueckermann (2009). We restrict ourselves to a relatively simple model to focus on the numerics. However, the model is complex enough to reveal important characteristics and to complete a large number of parameter sensitivity studies which we can synthesize. We study three biological regimes, one with single stable points at all depths and two with stable limit cycles. We examine interactions that are dominated by the biology, by the advection, or that are balanced. We also consider idealized simulations of biological patchiness which is commonly observed in the coastal ocean. For these regimes and interactions, we study a wide range of temporal and spatial discretizations. In what follows, we give our dynamical problem statement, definitions, and notation in Section 2. Our new numerical schemes and discretization are formulated and studied in Section 3. The results of our varied numerical and scientific investigations are described in Section 4. Finally, our conclusions are stated in Section 5.

2 Dynamical problem statement, definitions, and notation

2.1 Dynamical problem statement

The biological dynamics are governed by the following advection–diffusion–reaction (ADR) equations:

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot (\mathbf{u}\Phi) - \kappa \nabla^2 \Phi = \mathbf{S}(\Phi, \mathbf{x}, t), \quad \text{in } \Omega \tag{1}$$

with boundary conditions

$$\begin{aligned} \Phi &= \mathbf{g}_{\mathrm{D}}, \quad \text{on } \Gamma_{\mathrm{D}} \\ (\mathbf{u}\Phi - \kappa\nabla\Phi)\cdot\hat{\mathbf{n}} &= \mathbf{g}_{\mathrm{N}}, \quad \text{on } \Gamma_{\mathrm{N}} \end{aligned} \tag{2}$$

where $\Phi(\mathbf{x}, t) = [\phi^1(\mathbf{x}, t), \ldots, \phi^{N_c}(\mathbf{x}, t)]$ is the vector of N_c biological components, $\mathbf{u}$ is the prescribed velocity field, κ is a positive diffusivity coefficient, $\mathbf{S}(\Phi, \mathbf{x}, t)$ is the biological reaction terms, and $\mathbf{g}_{\mathrm{D}}$, $\mathbf{g}_{\mathrm{N}}$ are the boundary conditions for the Dirichlet and Neumann boundaries, respectively. Equations 1 and 2 are solved on the domain $\Omega \in \mathbb{R}^d$, where d is the dimension of the problem, with boundary $\partial\Omega = \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}}$ such that $\Gamma_{\mathrm{D}} \cap \Gamma_{\mathrm{N}} = \emptyset$.

Since we are interested in strait dynamics, for the flowfield $\mathbf{u}$, we assume that earth rotational effects are negligible, which is true if the ratio of the strait width to the Rossby radius is small (Pedlosky 1987; Signell 1989; Cushman-Roisin 1987; Bourgault and Kelley 2004). Additionally, for uniform geometry across the strait with a rigid lid approximation, a small Froude number, and a homogeneous density, the velocity field can be approximated as a potential flow field. A similar setup was used by Signell (1989) for tidal flows. The potential velocity $\mathbf{u}$ is obtained by solving for the stream function

$$\nabla^2 \psi = 0, \quad \text{in } \Omega \tag{3}$$

$$\mathbf{u} = \nabla \times \psi \tag{4}$$

with boundary conditions

$$\psi = h_{\mathrm{D}}, \quad \text{on } \partial\Omega. \tag{5}$$

2.2 Finite element definitions and notation

We discretize Eqs. 1–5 using the DG finite element method. The first reported use of DG FEM was by Reed and Hill (1973) where DG was used to solve the steady-state neutron transport equation. However, DG drew little attention until a series of papers (Cockburn and Shu 1989, 1998b; Cockburn et al. 1989, 1990), where the Runge–Kutta DG methods were described. The extension of DG to higher-order derivatives by Bassi and Rebay (1997) made the method applicable to solving advection–diffusion equations, which can be extended to solving the Navier–Stokes equations. Since the late 1990s, DG has seen a number of realistic applications in aerospace, solid mechanics, and electromagnetism to name a few. For a review on the use of such schemes in next-generation physical ocean models, we refer to Pain et al. (2005), Slingo et al. (2009), and Ueckermann (2009).

In this section, we first describe the notation used for the domain and the discrete elements. Then we define the notation used for the solution on the element interior and on the element interfaces. Next, we described the discontinuous polynomial spaces and the necessary inner products for the DG discretization, followed by the discontinuous finite element space and inner product for the hybridized discontinuous Galerkin (HDG) method. A set of terms is also defined.

The basic domain notation is illustrated in Fig. 1. We let $\mathcal{T}_h = \cup K_i$ be a finite collection of nonoverlapping elements, K_i, that discretizes the domain Ω, where h denotes the characteristic size of an element. Also, let $\partial\mathcal{T}_h = \{\partial K : K \in \mathcal{T}_h\}$ be the set of interfaces of all

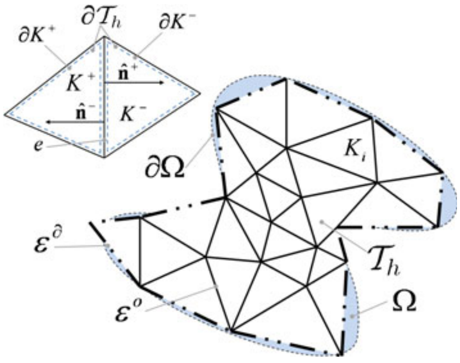

Fig. 1 Notation for domain and triangular elements

elements. For two elements K^+ and K^- belonging to $\mathcal{T}_h$, we define $e = \partial K^+ \cap \partial K^- \neq \emptyset$ as the unique interior interface between elements K^+ and K^-. For a single element K belonging to $\mathcal{T}_h$, $e = \partial K \cap \partial\Omega \neq \emptyset$ is a boundary interface. Let ε_h° and ε_h^∂ denote the set of unique interior and boundary interfaces, respectively, such that $\varepsilon_h = \varepsilon_h^\partial \cup \varepsilon_h^\circ$. We note that in the interior $\partial\mathcal{T}_h$ contains two interfaces, ∂K^+ and ∂K^-, at the same location (one for each element), whereas the set ε_h only contains a single interface, e, at the same location.

K^+ and K^- have outward pointing normals $\hat{\mathbf{n}}^+$ and $\hat{\mathbf{n}}^-$, respectively. We then let vector and scalar quantities $(\mathbf{q}^\pm, u^\pm)$ be the traces of $(\mathbf{q}, u)$ on the interface e from the interior of $K^\pm$. The mean value $\{\{.\}\}$ and jumps $[\![.]\!]$ on the interior interface $e \in \varepsilon_h^\circ$ for scalar and vector quantities are then defined as

$$\begin{aligned} \{\{\mathbf{q}\}\} &= (\mathbf{q}^+ + \mathbf{q}^-)/2 & \{\{u\}\} &= (u^+ + u^-)/2 \\ [\![\mathbf{q}\cdot\hat{\mathbf{n}}]\!] &= \mathbf{q}^+\cdot\hat{\mathbf{n}}^+ + \mathbf{q}^-\cdot\hat{\mathbf{n}}^- & [\![u\hat{\mathbf{n}}]\!] &= u^+\hat{\mathbf{n}}^+ + u^-\hat{\mathbf{n}}^-. \end{aligned}$$

On the set of boundary interfaces $e \in \varepsilon_h^\partial$ (with outward facing normal $\hat{\mathbf{n}}$ on $\partial\Omega$), we set

$$\begin{aligned} \{\{\mathbf{q}\}\} &= \mathbf{q} & \{\{u\}\} &= u \\ [\![\mathbf{q}\cdot\hat{\mathbf{n}}]\!] &= \mathbf{q}\cdot\hat{\mathbf{n}} & [\![u\hat{\mathbf{n}}]\!] &= u\hat{\mathbf{n}}. \end{aligned}$$

since here $\mathbf{q}$ and u are single-valued. Note that the jump in a vector is a scalar (involving only the normal component of the vector), whereas the jump in a scalar is a vector. Additionally, the jump will be zero for a continuous function.

The main difference between continuous Galerkin (CG) and DG lies in the approximation subspaces used. DG uses bases that are in normed space $L^2(\Omega)$ while CG uses bases that are in the Hilbert space $H^1(\Omega)$, that is, the function has to be continuous across elements. For a function $f(\mathbf{x})$ to be in $L^2(\Omega)$, it has to satisfy $\int_\Omega f(\mathbf{x})^2 d\Omega < \infty$, whereas a function in $H^1(\Omega)$ has to belong to a smaller space satisfying $\int_\Omega f(\mathbf{x})^2 + \nabla f(\mathbf{x}) \times \nabla f(\mathbf{x}) d\Omega < \infty$. Let $\mathcal{P}^p(D)$ denote the set of polynomials of maximum degree p existing on a domain D. For example, we will be using $p2$ to denote a second degree polynomial basis, which will result in a third-order accurate scheme. We introduce the discontinuous finite element spaces defined as

$$\begin{aligned} W_h^p &= \left\{ w \in L^2(\Omega) : w\,|_K \in \mathcal{P}^p(K), \forall K \in \mathcal{T}_h \right\} \\ \mathbf{V}_h^p &= \left\{ \mathbf{v} \in (L^2(\Omega))^d : \mathbf{v}\,|_K \in (\mathcal{P}^p(K))^d, \forall K \in \mathcal{T}_h \right\} \end{aligned}$$

where W_h^p is a scalar space, $\mathbf{V}_h^p$ is a vector space of dimension d, and $L^2(D)$ is the space of square integrable functions $f(\mathbf{x})$ such that $\int_D f(\mathbf{x})^2 dD < \infty$ on domain D.

Finally, we define the inner products over continuous domains $D \in \mathbb{R}^d$ and $\partial D \in \mathbb{R}^{d-1}$ as

$$(\mathbf{q}, \mathbf{v})_D = \int_D \mathbf{q}\cdot\mathbf{v}dD \qquad (u, w)_D = \int_D uwdD$$

$$\langle \mathbf{q}, \mathbf{v}\rangle_{\partial D} = \int_{\partial D} \mathbf{q}\cdot\mathbf{v}d\partial D \quad \langle u, w\rangle_{\partial D} = \int_{\partial D} uwd\partial D \tag{6}$$

for vector functions $\mathbf{q}$, $\mathbf{v}$ and scalar functions u, w. Over discontinuous domains, we also define

$$(q, v)_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\mathbf{q}, \mathbf{v})_K, \quad \langle u, w\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle u, w\rangle_{\partial K}, \tag{7}$$

for vector or scalar functions q, v defined on $\mathcal{T}_h$, and u, v defined on $\partial\mathcal{T}_h$.

To use the HDG framework for solving Eqs. 3–5, we will require the traced finite element space existing on the interfaces ε_h

$$M_h^p = \left\{ \mu \in L^2(\varepsilon_h) : \mu|_e \in \mathcal{P}^p(e), \forall e \in \varepsilon_h \right\}.$$

We also set $M_h^p(g_D) = \{\mu \in M_h^p : \mu = \mathsf{P} g_D \text{ on } \Gamma_D\}$, where P is the L^2 projection into the space $\{\mu|_{\partial\Omega} \forall \mu \in M_h^p\}$. Note that M_h^p is continuous on the interface, e, shared by K^+ and K^-, but discontinuous at the borders between different interfaces. We will also require the additional inner product on this discontinuous domain

$$\langle \mu, \eta\rangle_{\varepsilon_h} = \sum_{e\in\varepsilon_h} \langle \mu, \eta\rangle_e \tag{8}$$

for vector or scalar functions μ, η defined on ε_h.

2.3 Comparing numerical codes: defining efficiency, accuracy, and performance

To be clear, we use the term "efficiency" or "cost" to refer exclusively to the computational resources (elapsed time, memory) required for a simulation, and we do not use "efficiency" to imply any degree of correctness of the solution. We reserve the term "accuracy" to refer to the correctness of the solution. Finally, here we also use the term "performance" as the combined consideration between efficiency and accuracy (Chapra and Canale 2006).

Comparing different numerical schemes is not a straightforward task (see Kubatko et al. 2009). First, results are not universally applicable and are generally problem dependent. After focusing on a particular class of problems, the usual approach is to fix the computational efficiency of both schemes and then compare the accuracy, or vice versa. The scheme that performs better will then have a superior accuracy, since the efficiency will be the same for both. However, the efficiency of the scheme is dependent on its implementation, as well as the computer architecture on which the simulations are performed. A simple approach, then, is to fix the number of degrees of freedom (DOFs) of the different schemes, that is, having the same number of unconstrained parameters in both schemes. Because the DOFs are related to computational efficiency, this approach is useful for comparing similar numerical schemes with different implementations. However, it is not a good approach when comparing different numerical schemes where the computational cost per DOF is inherently and significantly different between the schemes, which is the case for comparisons between high-order and low-order schemes. Finally, conclusions drawn about the performance is also dependent on the particular definition of accuracy. The accuracy is normally defined in terms of a quantity useful to a particular researcher. Thus, researchers with different quantities of interest may draw different conclusions about the performance of a scheme. We address the efficiency issue by presenting results for multiple efficiencies, and we address the accuracy issue by using generic global error measures (see Section 3.5) and by using difference plots.

3 Numerical methodology

In this section, we first derive the basic FE formulations for Eqs. 1–5, using the notation from Section 2.2. Starting with Eqs. 1 and 2, we multiply each biological component by test function w and integrate over the domain. We seek approximations $\Phi_h = [\phi_h^1 \dots \phi_h^{N_c}]$ of Φ such that for all $K \in \mathcal{T}_h$,

$$\left(\frac{\partial \Phi_h}{\partial t}, w\right)_K + (\nabla \times (\mathbf{u}\Phi_h), w)_K - \left(\kappa \nabla^2 \Phi_h, w\right)_K$$
$$= (\mathbf{S}(\Phi_h, \mathbf{x}, t), w)_K \,, \forall w \in \mathcal{P}(K). \tag{9}$$

where each component $\phi_h^i \in W_h^p$. We set $\phi_h^i = \sum_{j=1}^{N_P} \phi_j^i(t)\theta_j(\mathbf{x})$ where $\phi_j^i(t)$ are N_P time varying coefficients with N_P corresponding spatial basis functions, $\theta_j(\mathbf{x}) \in \mathcal{P}(K)$. For convenience, we use Einstein summation notation $\Phi_h = \Phi_j\theta_j$, where the sum over the repeated index j is implied and we let $\Phi_j \equiv \Phi_j(t), \theta_j \equiv \theta_j(\mathbf{x})$.

To obtain our finite-element formulation of Eqs. 3–5, we multiply by test functions w and $\mathbf{v}$ and integrate over the domain. We seek approximations $\psi_h \in W_h^p$ of ψ and $\mathbf{u}_h \in \mathbf{V}_h^p$ of $\mathbf{u}$ such that for all $K \in \mathcal{T}_h$,

$$\left(\nabla^2 \psi_h, w\right)_K = 0, \qquad \forall w \in \mathcal{P}(K), \tag{10}$$

$$(\mathbf{u}_h, \mathbf{v})_K = (\nabla \times \psi_h, \mathbf{v})_K \,, \quad \forall \mathbf{v} \in (\mathcal{P}(K))^d \tag{11}$$

Next, we describe in detail the discretization of the source terms in Section 3.1, since one of the novel aspects of our work is the high-order schemes for ocean biogeochemical simulations. Our spatial discretizations are derived in Section 3.2 for the advection terms, Section 3.3 for the diffusive terms, and our temporal discretizations given in Section 3.4. The calculation of error norms is outlined in Section 3.5. Finally, brief details of our implementation are presented in Section 3.6, and our method for generating high-order curved meshes is described in Section 3.7.

3.1 Source term discretization

Quadrature-based integration approximates a definite integral by a weighted sum of function evaluations at discrete points (quadrature points). For example, Gaussian quadrature using N_g points can exactly evaluate polynomials of degree $2N_g - 1$. Quadrature-free methods avoid evaluating the weighted sum by using approximations and/or analytically evaluating definite integrals, and the solution to the definite integral is used directly in numerical implementations. For more details, see for example, Hesthaven and Warburton (2008). We further restrict the definition of "quadrature-free" to mean exact integration on the original polynomial basis. It is possible to use an expanded basis to evaluate the source terms, but additional cost is involved to interpolate the solution unto the higher degree basis. For the source term discretization, either approach could be used, and we will examine the impact of this choice.

To discretize the source terms with a quadrature-based approach, consider the source term for a single biological component i:

$$\begin{aligned}\left(S^i(\Phi_h, \mathbf{x}_i, t), w_k\right)_K &= \left(S^i(\Phi_j\theta_j, \mathbf{x}, t), w_k\right)_K \\ &\approx S^i\left(\Phi_j\theta_j(\mathbf{x}_i), \mathbf{x}_i, t\right) w_k(\mathbf{x}_i)\omega_i \\ &= \mathcal{W}_{ki}\left[S^i\left(\Phi_j\Theta_{ji}, \mathbf{x}_i, t\right)\omega_i J_i\right]\end{aligned} \tag{12}$$

where $\mathbf{x}_i$ are the locations of the N_g quadrature points with corresponding weights ω_i, $\mathcal{W}_{ki} = w_k(\mathbf{x}_i)$ are the values of the test functions evaluated at the quadrature points, and $\Theta_{ji} = \theta_j(\mathbf{x}_i)$ are the values of the basis functions evaluated at the quadrature points. To evaluate this integral numerically, we pre-compute the matrix $\mathcal{W} \in \mathbb{R}^{N_P \times N_g}$. Because a reference element is used, the corresponding Jacobians at the quadrature points J_i also need to be calculated, and these are multiplied together with the weights (ω). The integration over an element is performed as a matrix–vector multiplication with $\mathcal{O}(2N_g N_P)$ operations for a single biological component. In addition, we need to consider the interpolation of the values of Φ_h unto the quadrature points, $\Theta_{ji}\Phi_j$, resulting in an additional $\mathcal{O}(2N_g N_P)$ for each biological component. Finally, we have to evaluate the function describing the source terms at the quadrature points. The quadrature-based algorithm thus has a total of $\mathcal{O}(4N_c N_g N_P) + N_g C_s$ operations per element, where C_s is the cost of evaluating the source terms.

Alternatively, to discretize the source terms with a quadrature-free approach using the original basis, we use

$$\begin{aligned}\left(S^i(\Phi_h, \mathbf{x}, t), w_k\right)_K &\approx \left(S^i\left(\Phi_j, \mathbf{x}_j, t\right)\theta_j, w_k\right)_K \\ &= \left(\theta_j, w_k\right)_K S^i\left(\Phi_j, \mathbf{x}_j, t\right) \\ &= \mathcal{M}_{kj} S^i\left(\Phi_j, \mathbf{x}_j, t\right) \qquad (13)\end{aligned}$$

where $\mathbf{x}_j$ are the locations of the N_P nodal points and we can pre-compute the element local mass matrix $\mathcal{M}_{kj} = (\theta_j, w_k)_K$ such that $\mathcal{M} \in \mathbb{R}^{N_P \times N_P}$. Note that, for straight-sided elements, the mass matrix computed on the reference matrix can be used and multiplied by the element local scalar Jacobian. However, on elements with curved boundaries, a mass matrix using the element local spatially variable Jacobian is computed. In the quadrature-free case, we approximate the integral by essentially fitting the function $S^i(\Phi, \mathbf{x}, t)$ on the polynomial space spanned by θ. For a nodal basis, the coefficients of the basis are the values of the source terms evaluated at discrete points. Since these source term values at nodal points do not depend continuously on space, the second equality in Eq. 13 follows. This approach introduces an aliasing error since $S^i(\Phi, \mathbf{x}, t)$ may contain complicated functions which are not captured in span$\{\theta\}$. The operation count for this approach scales as $\mathcal{O}(2N_c N_p N_p) + N_p C_s$ if we need to multiply through by the mass matrix. If the number of quadrature points are equal to the number of basis functions, $N_g = N_p$, and the cost of evaluating the source terms is small, then the quadrature-free algorithm is two times more efficient in terms of total operations.

However, we can normally eliminate the mass matrix multiplication in front of the source term since there is also a mass matrix in front of the $\frac{\partial \phi_h^i}{\partial t}$ term, $\mathcal{M}_{kj}\frac{\partial \phi_j^i}{\partial t} = \mathcal{M}_{kj} S^i\left(\Phi_j, \mathbf{x}_j, t\right) \to \frac{\partial \phi_j^i}{\partial t} = S^i\left(\Phi_j, \mathbf{x}_j, t\right)$. Note that the mass matrix cannot be eliminated for all the terms in the partial differential equation (PDE), for example, a matrix will remain in front of the discretized advection operator. However, the operation count for evaluating the source terms of the quadrature-free algorithm reduces to $N_p C_s$. Therefore, from an efficiency perspective, it is desirable to use a quadrature-free algorithm.

The accuracy of the quadrature-based integration is limited by the quadrature rule used and the number of quadrature points N_g. Choosing a greater number of quadrature points, any desirable accuracy can be obtained, at the cost of reduced efficiency. Conversely, the accuracy of the quadrature-free integration is limited by the order of basis function used, and potentially large errors can be introduced due to inexact integration.

The biological source terms are the origin of nonlinear and nonhomogeneous dynamics for the whole PDE. They can lead to high-accuracy discretization requirements in the other terms of the PDE in both space and time. This is discussed in the next sections (Sections 3.2–3.4).

3.2 Spatial discretization of advection operators

Integrating the advection terms $(\nabla \cdot (\mathbf{u}\Phi_h), w)_K$ by parts and using the divergence theorem, we obtain the weak form of the advection operator

$$(\nabla \cdot (\mathbf{u}\Phi_h), w)_K = -(\mathbf{u}\Phi_h, \nabla w)_K + \langle \widehat{\mathbf{u}\Phi_h} \cdot \hat{\mathbf{n}}, \nabla w \rangle_{\partial K}, \qquad (14)$$

where the formulation is complete once we specify the value of the flux $\widehat{\mathbf{u}\Phi_h}$. Here we use the upwind flux

$$\widehat{\mathbf{u}\Phi_h} \cdot \hat{\mathbf{n}} = \mathbf{u} \cdot \hat{\mathbf{n}}\{\{\Phi_h\}\} - \frac{1}{2}|\mathbf{u}| \cdot [\![\Phi_h \hat{\mathbf{n}}]\!] \qquad (15)$$

The same quadrature versus quadrature-free discussion is relevant to the advection terms. However, here we choose to use a quadrature-based scheme to ensure the accuracy of the advection part of the discretization. Since we are focusing on evaluating the accuracy of the source terms, we do not want the additional consideration about the accuracy of the advection operators to complicate the discussion. Note, however, that considerable efficiency can be gained for the advection terms as well if a quadrature-free scheme is employed (Hesthaven and Warburton 2008). We then let

$$
\begin{aligned}
&-(\mathbf{u}\Phi_h, \nabla w_k)_K + \left\langle \widehat{\mathbf{u}\Phi_h} \cdot \hat{\mathbf{n}}, \nabla w_k \right\rangle_{\partial K} \\
&\quad \approx -\mathcal{D}_{ki}\left[\left(\mathbf{u}_i \Phi_j \Theta_{ji}\right) \omega_i J_i\right] \\
&\qquad + \sum_{e \in K} \mathcal{W}^e_{ki}\left[\left(\widehat{\mathbf{u}_i \Phi}_j \Theta^e_{ji}\right) \cdot \hat{\mathbf{n}}_i \omega^e_i J^e_i\right] \qquad (16)
\end{aligned}
$$

where $\mathcal{D}_{ki} = \nabla w_k(\mathbf{x}_i)$ and the superscript $(.)^e$ indicates that the quantity is defined on an element interface and $\mathbf{u}_i = \mathbf{u}_j \Theta_{ji}$ is the flow-field evaluated at the quadrature points.

3.3 Spatial discretization of diffusive operators

To discretize the diffusive operator $\left(\kappa \nabla^2 \Phi_h, w\right)_K$, we follow the standard practice with DG and consider the canonical problem

$$-\nabla \cdot (\kappa \nabla)\phi = s.$$

Introducing the auxiliary variable $\mathbf{q} = -\kappa \nabla \phi$, this equation is re-written as two coupled first-order equations

$$
\begin{aligned}
\nabla \cdot \mathbf{q} &= s \\
\mathbf{q} + \kappa \nabla \phi &= 0.
\end{aligned}
$$

Multiplying these equations by the appropriate test functions, integrating by parts, and applying the divergence theorem, we obtain the finite element formulation for all $K \in \mathcal{T}_h$

$$-(\mathbf{q}_h, \nabla w)_K + \left\langle \hat{\mathbf{q}}_h \cdot \hat{\mathbf{n}}, w \right\rangle_{\partial K} = (s_h, w)_K \quad \forall w \in \mathcal{P}(K) \qquad (17)$$

$$
\begin{aligned}
&\left(\kappa^{-1}\mathbf{q}_h, w\right)_K - (\phi, \nabla \cdot \mathbf{v})_K \\
&\quad + \left\langle \hat{\phi}_h, \mathbf{v} \cdot \hat{\mathbf{n}} \right\rangle_{\partial K} = 0 \qquad \forall \mathbf{v} \in (P(K))^d \qquad (18)
\end{aligned}
$$

where we have multiplied by κ^{-1} so that it does not appear as part of the flux (or interface) terms. This formulation is complete once we specify the form of the flux terms at the interfaces, $\hat{\mathbf{q}}$ and $\hat{\phi}_h$. The diffusive fluxes for DG schemes are normally reported in the form

$$\hat{\mathbf{q}}_{\mathbf{h}} = \{\{\mathbf{q}_h\}\} - C_{11}[\![\phi_h \hat{\mathbf{n}}]\!] + \mathbf{C}_{12}[\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!] \qquad (19)$$

$$\hat{\phi}_h = \{\{\phi_h\}\} - \mathbf{C}_{12} \cdot [\![\phi_h \hat{\mathbf{n}}]\!] - C_{22}[\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!] \qquad (20)$$

Biogeochemical diffusive fluxes Using explicit time integration to solve Eq. 1, we do not need to invert a matrix in Eq. 9, in which case we utilize the local discontinuous Galerkin fluxes (Cockburn and Shu 1998a)

$$C_{11} = \tau, \quad \mathbf{C}_{12} = \frac{\hat{\mathbf{n}}^{\pm}}{2}, \quad C_{22} = 0, \qquad (21)$$

although many other choices exist.

Potential flows For solving Eq. 3, a matrix inversion is required in Eq. 10, in which case we discretize the diffusive operators using the novel HDG method. For the full derivation of these equations, including more specific implementation details, refer to Cockburn et al. (2009) and Nguyen et al. (2009). The premise of HDG is to recognize that one can solve Eqs. 17–18 locally on an element as long as the flux quantities are known. Within an HDG framework, the local element unknowns are parameterized in terms of a new variable $\lambda_h \in M^p_h(0)$, where the notation $M^p_h(0)$ refers to the space M^p_h that is zero-valued on the boundaries of the domain. The fluxes are expressed as

$$\hat{\phi}_h = \begin{cases} \mathsf{P} g_D, \text{ on } \varepsilon^{\partial}_h \\ \lambda_h, \text{ on } \varepsilon^{\circ}_h \end{cases} \qquad (22)$$

$$\hat{\mathbf{q}}_h = \mathbf{q}_h + \tau(\phi_h - \hat{\phi}_h)\hat{\mathbf{n}}, \quad \text{on } \partial\mathcal{T}_h \qquad (23)$$

where τ is a tune-able stabilization parameter and P is, again, the L^2 projection into the space $\{\mu|_{\partial\Omega} \forall \mu \in M^p_h\}$. Now, once λ_h is known, Eqs. 17–18 can be solved efficiently on each element independently. What remains is an equation for λ_h, which can be found by enforcing continuity of the normal diffusive flux

$$\langle [\![\hat{\mathbf{q}} \cdot \hat{\mathbf{n}}]\!], \mu \rangle_{\varepsilon_h} = \langle g_N, \mu \rangle_{\Gamma_N}. \qquad (24)$$

Note that this is an equation with globally coupled unknowns. However, the number of unknowns is greatly reduced compared to the original system, since only unknowns on the interfaces ε_h are involved.

This solution method involves three steps:

1. The inversion of local operators on each element to form both the right-hand-side vector and the global matrix
2. The global solution to find λ_h
3. The local reconstruction of the solution on the element

The local operations are efficient because inversions are done on matrices which are of dimension $\mathbb{R}^{(1+d)N_p \times (1+d)N_p}$. This procedure dramatically increases the efficiency of solving elliptic problems with DG where implicit time integration is required. Additionally, when the stabilization parameter for this choice of fluxes is chosen optimally ($\tau \sim \mathcal{O}(1)$), the optimal convergence rate of $\mathcal{O}(p+1)$ is obtained for both the gradient ($\mathbf{q}$) and the solution (ϕ; Nguyen et al. 2009).

This allows a post-processing procedure that can produce a solution u_h^* which converges at $\mathcal{O}(p+2)$.

Reporting the fluxes for HDG in the standard form we have from Nguyen et al. (2009), the following

$$C_{11} = \frac{\tau^+\tau^-}{\tau^+ + \tau^-}, \quad \mathbf{C}_{12} = \frac{1}{2}\left(\frac{[\![\tau\hat{\mathbf{n}}]\!]}{\tau^+ + \tau^-}\right),$$
$$C_{22} = \frac{1}{\tau^+ + \tau^-}. \tag{25}$$

3.4 Temporal discretization

Motivated by strong nonlinearities in biogeochemical dynamics, an objective of this study is to evaluate effects of temporal discretization errors on the accuracy of numerical simulations. We investigate fourth-order, second-order, and first-order schemes in time. Specifically, for the majority of this work, we will use the following four-stage low-storage fourth-order accurate Runge–Kutta scheme for explicit time integration

$$\begin{aligned}
\Phi^a &= \Phi(t) + \frac{\Delta t}{4}\left.\frac{\partial \Phi}{\partial t}\right|_{\Phi(t)} \\
\Phi^b &= \Phi(t) + \frac{\Delta t}{3}\left.\frac{\partial \Phi}{\partial t}\right|_{\Phi^a} \\
\Phi^c &= \Phi(t) + \frac{\Delta t}{2}\left.\frac{\partial \Phi}{\partial t}\right|_{\Phi^b} \\
\Phi(t+\Delta t) &= \Phi(t) + \Delta t\left.\frac{\partial \Phi}{\partial t}\right|_{\Phi^c}.
\end{aligned}$$

It is implemented in a four-stage for loop, where the solution at the initial time $\Phi(t)$ is saved in a temporary array and the array containing the solution is updated using the three intermediate Φ^{a-c} variables.

We evaluate the temporal discretization error by considering a second accurate explicit Runge–Kutta scheme

$$\begin{aligned}
\Phi^a &= \Phi(t) + \frac{\Delta t}{2}\left.\frac{\partial \Phi}{\partial t}\right|_{\Phi(t)} \\
\Phi(t+\Delta t) &= \Phi(t) + \Delta t\left.\frac{\partial \Phi}{\partial t}\right|_{\Phi^a},
\end{aligned}$$

and the first-order-accurate explicit Euler scheme

$$\Phi(t+\Delta t) = \Phi(t) + \Delta t\left.\frac{\partial \Phi}{\partial t}\right|_{\Phi^t}.$$

In each case, $\frac{\partial \Phi}{\partial t}$ is evaluated using the right-hand-side spatial PDE.

3.5 Error norm calculation

Unless indicated otherwise, the global domain L^2 norm $\|e\|_2 = \left(\int_\Omega e^2 d\Omega\right)^{\frac{1}{2}}$ is calculated using the quadrature-based approach as described in Section 3.1. That is, the numerical solution is interpolated unto the quadrature points, the error $e = \phi_h - \phi$ is evaluated, and then multiplied by the quadrature weights and summed for an approximate integration. In some cases, we evaluate the global error using an interpolation approach (similar to quadrature-free), and this is mentioned when we do. In these cases, the error is evaluated at the nodal points, then the error is interpolated to the quadrature points, multiplied by quadrature weights, and summed. Where ambiguous, we indicate the quadrature-based error evaluation using $\|e\|_2^{\text{qp}}$ (quadrature points) and the interpolated error evaluation using $\|e\|_2^{\text{nd}}$ (nodal points).

The infinity norm $\|e\|_\infty = \max|e|$ is calculated by evaluating the error at nodal points and taking the maximum absolute value.

3.6 Implementation

The discretized equations were implemented for 2D triangular elements. Even though our codes are efficient, our implementation is not fully optimized, in particular, the higher-order simulations would most benefit from further optimization (e.g., see Lambrechts et al. 2010). This is fine for our purposes since if we find that our high-order implementations are more accurate for the same cost/efficiency than lower-order schemes, then further optimization would only accentuate this result.

The correctness of our implementation is verified by performing convergence studies using analytical test cases on curved and straight geometries. The implementation of each discretized operator is verified both separately and collectively. Results of some of the convergence studies are shown in Sections 4.1 and 4.2.

For the polynomial spaces restricted on each element $\mathcal{P}^p(K)$, a nodal basis with $N_p = \frac{(p+1)(p+2)}{2}$ nodal points is used in two dimensions and $N_p = (p+1)$ nodal points in one dimension. The total number of DOFs can then be calculated by multiplying the number of nodal points by the number of elements. The node locations are chosen according the method described in Hesthaven and Warburton (2008). More specifically, we have the basis θ_j such that

$$\theta_j(\mathbf{x}_i) = \delta_{ij}$$

where $\mathbf{x}_i$ are the nodal locations. This basis is constructed for the reference element [0, 0], [1, 0], [0, 1], and an isoparametric coordinate mapping is used for arbitrary triangles. The isoparametric coordinate mapping is described through the element nodal locations in the problem reference frame.

To integrate polynomials of degree $p \geq 10$, cubature rules (multidimensional quadrature rules) are constructed on triangles by using tensor products of 1D Gauss quadrature rules. For $p < 10$, we use tabulated rules from Solin et al. (2003) and Strang and Fix (1973).

In all cases, we utilize the Galerkin approach, that is, we choose the test functions to be the same as the basis functions, $w_j = \theta_j$.

3.7 Higher-order mesh generation

Since higher-order DG schemes have more degrees of freedom per element, a coarse mesh with large elements is required to keep a similar performance across discretizations. To obtain an accurate solution with a coarse, high-order discretization, it is necessary to use curved boundary interfaces, as will be demonstrated in Section 4.3. Here we describe our new method for creating such a coarse, high-order curved mesh.

When curving the boundary of an element, care needs to be taken because it is possible to create an element where two of the interfaces cross. The left triangular element shown in Fig. 2 has the true circular geometry crossing one of the straight interfaces. To avoid this situation, we need to ensure that

$$h < 2\rho(x)\sin(\theta), \tag{26}$$

where h is the length of the element side bordering the boundary, $\rho(x) = \frac{|1+f'(x)|^{3/2}}{|f''(x)|}$ is the radius of curvature of the boundary described by $f(x)$, and θ is the minimum angle of the two angles on the edge bordering the boundary. The element shown on the right side of Fig. 2 illustrates this limiting case for an equilateral triangle, but our condition (26) is trivially extended to arbitrary triangles as shown by the dashed lines.

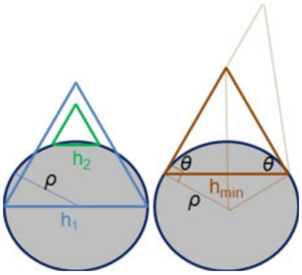

Fig. 2 Minimum triangle angle criterion (26) demonstrated on a circle with equilateral triangles. $h_1 = 2\rho$ does not satisfy the criterion, $h_2 = 2/3\rho$ satisfies the criterion, and $h_3 = 2\rho\sin(\pi/3)$ demonstrates the limiting case. This result can be extended to arbitrary triangles as shown by the *dashed lines*

Using our criteria (Eq. 26), we define the minimum edge spacing on the boundary as $h_{\min} = 2\rho\sin(30°) = \rho$. Then, we let the minimum edge length grow linearly by a certain percentage (fit to 12% here) away from the boundary up to a specified minimum edge length. Using these criteria, we create coarse base meshes, then uniformly refine these meshes to obtain finer discretizations. To create the meshes, we primarily used the free mesher *Distmesh* (Persson and Strang 2004), but we also used *Gmsh* (Geuzaine and Remacle 2009). *Distmesh* uses an implicit geometry representation, that is, we define the geometry by a distance function that gives the distance between a queried point and the nearest boundary. Using *Distmesh*, we create meshes with straight sides.

To curve the boundary interfaces, we use the same distance function provided to *Distmesh* and numerically calculate the gradient of the distance function to the boundary. The normalized gradient vector provides the direction of translation, but to determine the magnitude of the translation, a weight needs to be applied to the calculated distance. That is, $p_1^{new} = p_1^{old} + \mathcal{W}d\frac{\nabla d}{|\nabla d|}$, where d is the distance from the point p_1^{old} to the boundary and $\mathcal{W}$ is the weight. Now, points on the straight boundary interface are translated to the true, curved boundary with a weight 1, and points on interior interfaces are not translated, i.e., having weights 0. Points in the volume have weights defined by the same weighting functions used to create the nodal basis, that is

$$\mathcal{W}(e_1) = \left(\frac{2\lambda_3}{2\lambda_3 + \lambda_1}\right)\left(\frac{2\lambda_2}{2\lambda_2 + \lambda_1}\right)$$

where the point is defined by the barycentric coordinates λ_i corresponding to vertices i and e_1 is the curved boundary interface defined by vertices 2 and 3. For details of this blending function, see Hesthaven and Warburton (2008).

The base mesh with three mesh refinements is shown in Fig. 3, and details of the base mesh for a curved and straight mesh boundary are shown in Fig. 4. Using our criterion $h_{\min} = \rho$, the minimum theoretical edge length for our geometry, that is a Gaussian bump defined by $\tilde{H}(x) = e^{-x^2}$, is $h_{\min} = 0.25$. The mesh shown in Fig. 3a has a minimum edge length of $h_{\min} = 0.2418$, close to the theoretical value.

4 Numerical studies and scientific implications

Biogeochemical models may contain a large number of biological or chemical components (Hofmann and

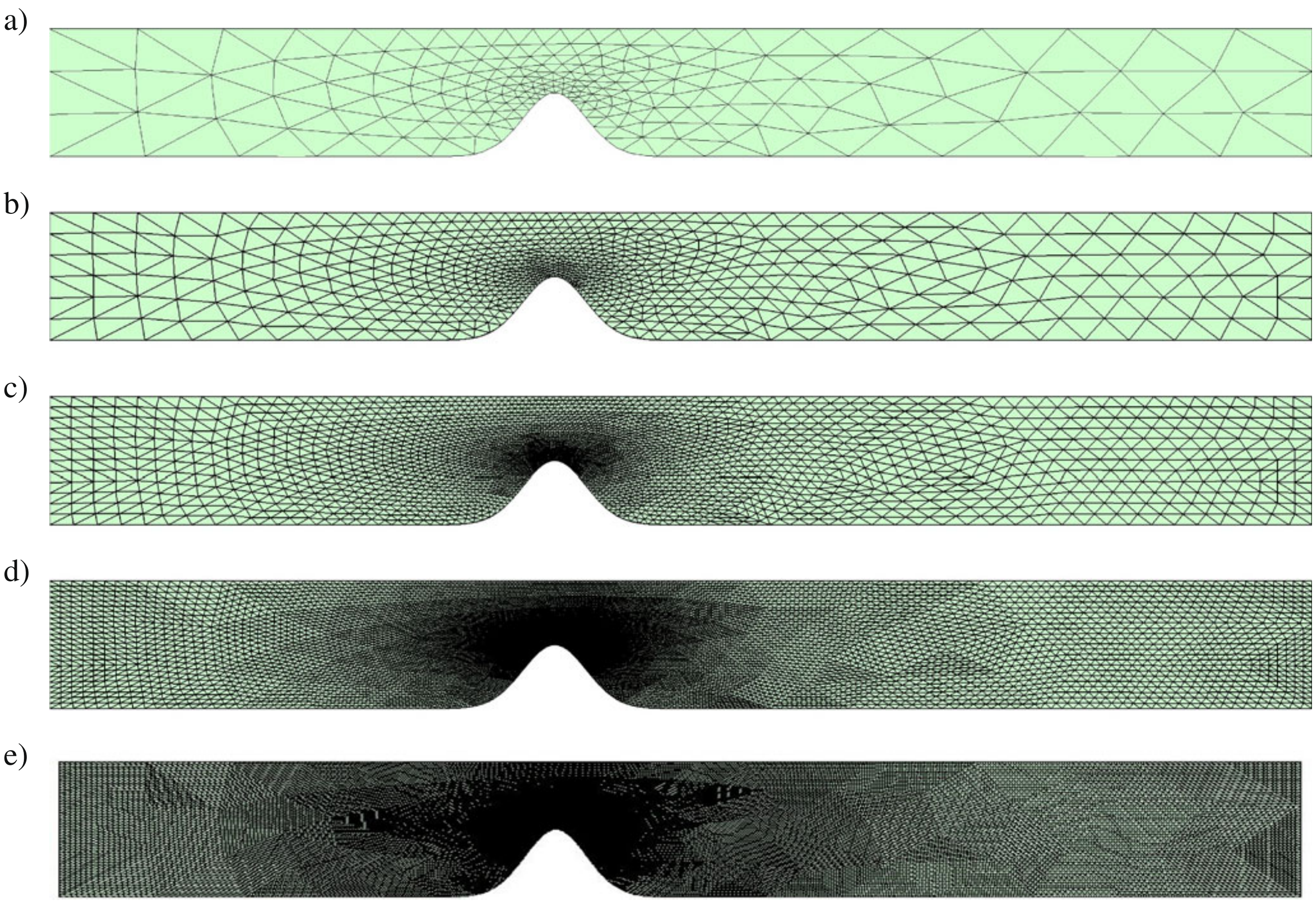

Fig. 3 **a** The base mesh ($g1$) with 350 elements. **b** First ($g2$) (1,400 elements). **c** Second ($g3$) (5,600 elements). **d** Third ($g4$) (22,400 elements) and fourth ($g5$) (89,600 elements) grid refinements. The more-refined meshes are used for lower-order schemes whereas less-refined meshes are used for higher-order schemes such that the cost of the two schemes are comparable

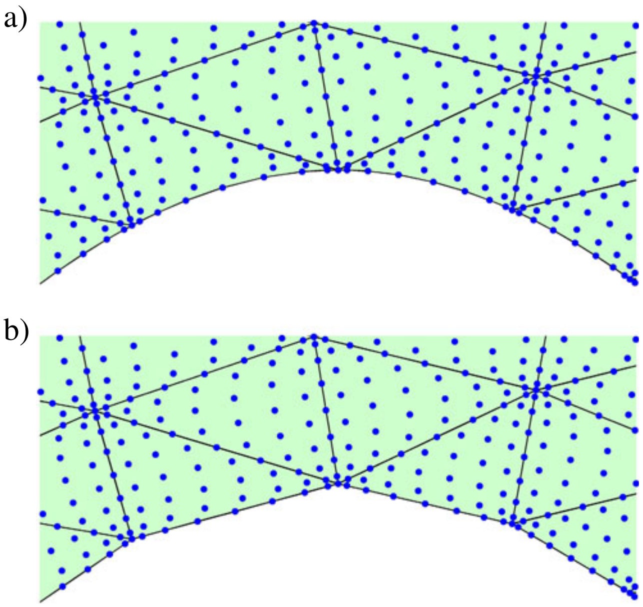

Fig. 4 Details of ($g1$) using a **a** curved and **b** straight mesh for a $p = 8$ nodal basis

Friedrichs 2002). The simplest models often only use nutrient, phytoplankton, and zooplankton as components and are commonly called NPZ models. More complicated models (Besiktepe et al. 2002) can be adaptive and contain many components. Each component requires the solution of an ADR equation of the form 1. The source terms describe the commonly nonlinear "reactions" and may lead to stationary, periodic, or chaotic dynamics. For this numerical work, a nondimensional version of a NPZ model (Flierl and McGillicuddy 2002) is used since it contains all characteristics required for our studies:

$$\begin{aligned}
&\frac{\partial \phi_N^*}{\partial t^*} + \nabla \times (\mathbf{u}^* \phi_N^*) - \nabla \times \frac{1}{P_e} \nabla \phi_N^* \\
&\quad = -\mathcal{U}^* e^{z^*/h^*} \frac{\phi_P^* \phi_N^*}{\phi_N^* + k_s^*} + d_P^* \phi_P^* + d_Z^* \phi_Z^* \\
&\qquad + (1-a) g_\nu^* \phi_Z^* (1 - e^{-\nu^* \phi_P^*}) \qquad (27)
\end{aligned}$$

$$\frac{\partial \phi^*_P}{\partial t^*} + \nabla \times \left(\mathbf{u}^* \phi^*_P\right) - \nabla \times \frac{1}{P_e} \nabla \phi^*_P$$

$$= \mathcal{U}^* e^{z^*/h^*} \frac{\phi^*_P \phi^*_N}{\phi^*_N + k^*_s} - d^*_P \phi^*_P$$

$$- g^*_\nu \phi^*_Z \left(1 - e^{-\nu^* \phi^*_P}\right) \tag{28}$$

$$\frac{\partial \phi^*_Z}{\partial t^*} + \nabla \times \left(\mathbf{u}^* \phi^*_Z\right) - \nabla \times \frac{1}{P_e} \nabla \phi^*_Z$$

$$= -d^*_Z \phi^*_Z + a g^*_\nu \phi^*_Z \left(1 - e^{-\nu^* \phi^*_P}\right) \tag{29}$$

where $\phi^*_{(N,P,Z)} = \frac{\phi_{(N,P,Z)}}{\mathcal{N}_t}$; $u^*_x = \frac{u_x}{\bar{\mathbf{u}}}$; $u^*_z = \frac{u_z L}{\bar{\mathbf{u}} H}$; $x^* = \frac{x}{L}$; $z^* = \frac{z}{H}$; $t^* = \frac{t}{\bar{\tau}}$; the parameters are explained in Table 1; the nondimensional groups with values are given in Table 2 with P_e the Peclet number and D^* the aspect ratio; the subscripts $(.)_N, (.)_P, (.)_Z$ refer to nutrients, phytoplankton, and zooplankton, respectively; $\nabla = \frac{\partial}{\partial x^*} + \frac{\partial}{\partial z^*}$; and lowercase z^* refers to the depth coordinate which is positive upward with $z^* = 0$ at the surface. Note that not all three equations (Eqs. 27–29) are required since the biological model satisfies the following conservation law for total nutrients, assuming a closed ocean system:

$$\phi^*_N = 1 - \phi^*_P - \phi^*_Z. \tag{30}$$

The first equation (Eq. 27), for example, could be eliminated in favor of Eq. 30; however, here we still use Eq. 30 to check the conservation of the numerical schemes.

The domain setup is depicted in Fig. 5 for the geometric parameter values given in Table 2. With this setup, an upwelling of nutrients is created (see Section 4.4), and the study of idealized biological blooms, which may occur in straits or sills, can be studied. In total, we consider three sets of parameter values, differing by the nondimensional parameters ν^* and k^*_s. In the absence of advection and diffusion, they lead to Eq. 29 with at most one physically relevant steady-state solution (Burton 2009). The three sets of nondimensional parameters ν^* and k^*_s correspond to biological dynamics with single stable points at all depths (bio case 1: $k^*_s = 1/30$, $\nu^* = 0.3$), with stable limit cycles for depths around $z^* = 0.4 - 0.9$ and single stable points elsewhere (bio case 2: $k^*_s = 1/50$, $\nu^* = 0.5$) and stable limit cycles everywhere in the euphotic zone (bio case 3: $k^*_s = 1/100$, $\nu^* = 0.1$). The middle parameter values, bio case 2, correspond to those values used by Flierl and McGillicuddy (2002): They are idealized and not meant to represent a specific ocean region. We note that biological models with discontinuities in stable solutions are not always representative of nature. However, biology of interest is likely to have intrinsic oscillatory or chaotic time dependence, e.g., Flierl and McGillicuddy (2002). For our purposes, we address these issues by considering three sets of parameter values and so cover a range of biological dynamics. To handle nonphysical negative concentrations due to numerics, we use

Table 1 NPZ equation parameter descriptions and units

Parameter	Description	Units
$\mathcal{U}$	Phytoplankton uptake rate	1/day
k_s	Saturation concentration of phytoplankton	μmol/L
d_P	Mortality rate of phytoplankton	1/day
d_Z	Mortality rate of zooplankton	1/day
g	Grazing rate of zooplankton	L/(μmol day
a	Assimilation (efficiency) rate	
h	e-folding depth for light (photosynthesis)	m
ν	Parameter for Ivlev form of grazing function	L/μmol
$\mathcal{N}_T$	Total biomass	μmol/L
$\bar{\mathbf{u}}$	Average inlet velocity	km/day
H	Height of bathymetry	m
D	Total maximum depth	m
L	Effective width of bathymetry	km
$\underline{\kappa}$	Diffusion tensor (vertical and horizontal diffusion different)	m^2/s

Table 2 Values of the dimensionless numbers entering the NPZ equations (Eq. 29) that are used in the examples for this manuscript

Parameter	Value
$\mathcal{U}^* = \mathcal{U}\bar{\tau}$	7.5
$k^*_s = \frac{k_s}{\mathcal{N}_T}$	$\left[\frac{1}{30}, \frac{1}{50}, \frac{1}{100}\right]$
$d^*_P = d_P \bar{\tau}$	0.2
$d^*_Z = d_Z \bar{\tau}$	1
$g^*_\nu = \frac{g\bar{\tau}}{\nu}$	12.5
$a^* = a$	0.4
$h^* = \frac{h}{H}$	0.34
$\nu^* = \mathcal{N}_T \nu$	[0.3, 0.5, 1]
$P_e = \frac{\bar{\mathbf{u}} L}{\kappa}$	∞
$D^* = \frac{D}{H}$	2

Bracketed triplets of values correspond to the three bio cases [1, 2, 3]. The other values are the same for the three cases

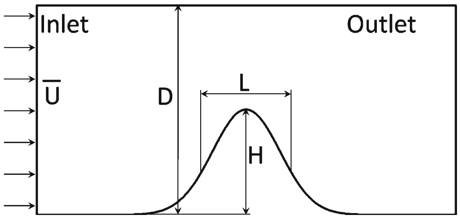

Fig. 5 Test case domain with idealized strait bottom geometry described by $\tilde{H}(x) = He^{-\frac{x^2}{L^2}}$

$\max(0, \phi^*_{(N,P,Z)})$ when evaluating the source terms. In the absence of advection, a timescale of $\bar{\tau} = 1$[days] is used, while in the presence of advection, the advective timescale $\bar{\tau}_a = L/\bar{\mathbf{u}}$ is used, where $\bar{\mathbf{u}} = \int_{\text{Inlet}} |\mathbf{u}| dz$ is the average inlet velocity.

In our numerical study, we need to characterize the three dynamical regimes and their behavior since these properties affect numerical errors. Specifically, for each dynamical regime, we study the three limiting balances of terms in Eqs. 27–29: biological terms dominating, advection terms dominating, and advection and biological terms balancing. When biological terms dominate, the advection is slow, and the problem reduces to a 1D problem, studied in Section 4.1. When advection is fast, the biology is unimportant, and we study this case in Section 4.3, with the generated flowfield studied in Section 4.2. Finally, the case where the advection and biological terms are approximately balanced is studied in Section 4.4, with the effect of biological patches demonstrated in Section 4.5. Since the timescale of biology varies in depth, the advection and biological terms can only be exactly balanced for one depth. While this results in many choices of approximately balanced terms, we will focus on one parameter set where $\bar{\tau}_a =$ 12.5 days. For more details on biological dynamics in straits, we refer to Burton (2009). Finally, capabilities of numerical filtering for higher-order schemes are examined in Section 4.6.

Results in this section will be reported for various grid and polynomial degree combinations, and the notation (grid number, polynomial degree) is used to denote this information. For example, $(g2, p4)$ refers to the second grid (Fig. 3b) with a fourth degree basis function. We provide a table with the number of DOFs and computational time estimates for the test cases we completed Table 3. In what follows, we show the results of $(g1, p6)$ and $(g2, p5)$ for our high-order simulations and compare them to $(g4, p1)$. Normalizing by the average computational time of $(g4, p1)$, these simulations have relative computational times of 0.34, 1, and 1.2 for $(g1, p6)$, $(g4, p1)$, and $(g2, p5)$, respectively (see Table 3). Following the discussion in Sections 2.3 and 3.6, we note that, in terms of efficiency, $(g4, p1)$ and $(g2, p5)$ are comparable (in fact, $(g2, p5)$ would be cheaper if fully optimized, see Section 3.6). $(g1, p6)$ is included because it is comparable in accuracy to $(g4, p1)$ and it highlights the effect of under-resolution (here $g1$) when using higher-order schemes (here $p6$).

4.1 1D biogeochemical source terms studies

In this section, we first illustrate the convergence of our numerical implementation. Following this, we examine the numerical behavior of the biological source terms using three tests: *perturbations of steady states, vertical resolution*, and *high-order bases*.

Numerical convergence in space and time Since an analytical solution to Eq. 29 does not exist, we verify the spatial implementation of the quadrature-free and quadrature-based source terms using the analytical test problem $\frac{\partial \phi}{\partial t} - 2\frac{\partial^2 \phi}{\partial z^2} = S(z, t)$ on $\Omega \in [-100, 0]$ integrating until $t = \pi/200$, with solution $\phi = \sin(t)\cos^2(\pi/50z)$ (for an appropriately chosen $S(z, t)$). We use a sufficiently small timestep, such that the errors are dominated by the spatial discretization. The results are shown in Table 4, with the norm of the error $e = \phi_h - \phi$ calculated as described in Section 3.5. From Table 4, we note that our implementation converges at the optimal rates for both the quadrature-based and quadrature-free treatments. While the error numbers are for a special case and not those for Eqs. 27–29, they show that the solution using quadratures is more accurate than the solution without quadratures, and this result will be generally expected.

Table 3 Normalized run-times and DOFs for various grids/polynomial degree basis functions, for the simulations in Section 4.4

Degree of basis	Grid 1	Grid 2	Grid 3	Grid 4	Grid 5
1	0.0014 (1,050)	0.014 (4,200)	0.12 (16,800)	**1.0** (67,200)	8.2 (268,800)
2	0.007 (2,100)	0.057 (8,400)	0.51 (33,600)	4.2 (134,400)	
3	0.026 (3,500)	0.21 (14,000)	1.8 (56,000)		
4	0.062 (5,250)	0.51 (21,000)	4.1 (84,000)		
5	0.15 (7,350)	**1.2** (29,400)			
6	**0.34** (9,800)	3.1 (39,200)			

The times are normalized by the $(g4, p1)$ run-time, and numbers in parentheses are the DOFs

Table 4 Spatial convergence of 1D DG solver used to evaluate the source terms using N_h elements

	Degree	$N_h = 10$		$N_h = 20$	
		$\|e\|_2$	Order	$\|e\|_2$	Order
Quadrature-based	$p = 1$	5.550e−003	1.9	1.409e−003	2.0
	$p = 2$	5.901e−004	2.9	7.491e−005	3.0
	$p = 3$	4.690e−005	3.9	3.019e−006	4.0
	$p = 4$	2.976e−006	4.9	9.673e−008	4.9
Quadrature-free	$p = 1$	1.340e−002	1.9	3.435e−003	2.0
	$p = 2$	1.068e−003	2.9	1.332e−004	3.0
	$p = 3$	7.464e−005	3.9	4.724e−006	4.0
	$p = 4$	4.282e−006	5.0	1.274e−007	5.1

The L^2 norm (see Section 3.5) of the error, $e = \phi_h - \phi$, is smaller for the quadrature-based scheme, but the order of convergence is the same for both. Order of convergence is computed in a standard way, e.g., Chapra and Canale (2006)

We verify the implementation of the fourth-order low-storage Runge–Kutta time integrator using the ordinary differential equation $\frac{\partial \phi}{\partial t} = \phi$ on $\Omega \in [-100, 0]$ integrating up to a time $t = 1$, with solution $\phi = \phi_0 e^t$. Here we choose $\phi_0 = 1$, such that the spatial discretization does not affect the error. The results are given in Table 5, from which we note that our implementation converges at the optimal rate for this low-storage Runge–Kutta scheme. While this test corresponds to exponential biological growth, as above, the error values are of course not those that would occur for Eqs. 27–29.

Perturbations of steady states The purpose of these studies is to examine how the biological dynamics behave as a result of perturbations away from the steady state. Perturbations will arise due to the forcing and dynamics and due to numerical reasons in the more complicated tests in Section 4.4, and it is important to understand how these perturbations will affect the biological dynamics. All three different biological regimes were examined in these perturbation tests. We focus on the behavior of the 1D dynamics for the time interval $t^* = [0, 250]$ because this corresponds to the residence time of the biology for the dynamics in Section 4.4. We initialize all tests using a perturbed or unperturbed exact steady state, which can be found by setting $\frac{\partial \Phi^*}{\partial t} + \nabla \cdot (\mathbf{u}^* \Phi^*) - \nabla \cdot \frac{1}{P_e} \nabla \Phi^* = 0$ in Eqs. 27–29. The steady-state solution is perturbed by setting $\Phi^*_{(P,Z)\text{perturb}} = (1 + \gamma)\Phi^*_{(P,Z)\text{steady}}$, where γ is some constant, and using Eq. 30. Where required, we impose $\phi^*_Z + \phi^*_P < 1$, by setting $\Phi^*_{(P,Z)\text{perturb}} = \frac{\Phi^*_{(P,Z)\text{steady}}}{(\phi^*_Z + \phi^*_P)_{\text{steady}}}$. This initialization is done numerically by setting the value of the numerical solution equal to the calculated solution *at the nodal points*.

First we ensure that the exact steady-state solution can be maintained, and then we initialize with a perturbation from the exact steady state, and the results are reported in Table 6. For these runs we used 100, second-order accurate linear elements ($p = 1$), which roughly corresponds to the resolution at the inlet for ($g5$, $p1$). We find that the steady solution can be maintained for all cases up to machine precision for the quadrature-free implementation when evaluating the error at the nodal points. This is because we initialized the numerical simulation using the exactly calculated steady state *at the nodal points*. Note that the quadrature-based scheme has a smaller difference than the quadrature-free version when evaluating the error at the quadrature points, except for the case with stable limit cycles in the euphotic zone (bio case 3). Because the quadrature version evaluates the source term at the quadrature points and the interpolation of the solution onto the quadrature points is not exactly at the analytical steady state, the source terms are nonzero, and the solution evolves. If the source-terms were polynomials of lower degree than the basis in the z direction, this would not happen.

Finally, Table 6 gives a rough description of the dynamical properties of the equations. Here the norm of the initial difference, $\|D_i\|_2$, should be compared to the norm of the final difference at quadrature points

Table 5 Temporal convergence of 1D DG solver using N_t timesteps (different values of N_t given only to show that the order does not vary with N_t but the absolute error of course changes)

Integration	$N_t = 16$		$N_t = 32$		$N_t = 64$	
scheme	$\|e\|_2$	Order	$\|e\|_2$	Order	$\|e\|_2$	Order
RK4	5.683e−006	3.9	3.646e−007	4.0	2.308e−008	4.0

Order is computed using Chapra and Canale (2006)

Table 6 Difference between analytical steady-state solution, and perturbed solution at $t^* = 250$

Stability	$\gamma \times 100\%$	$\|D_i\|_2^{\mathrm{qp}}$	$\|D_i\|_2^{\mathrm{nd}}$	$\|D_q\|_2^{\mathrm{qp}}$	$\|D_{\mathrm{qf}}\|_2^{\mathrm{qp}}$	$\|D_q\|_2^{\mathrm{nd}}$	$\|D_{\mathrm{qf}}\|_2^{\mathrm{nd}}$
Single stable points	0	0.361	0.000	0.296	0.361	0.117	0.000
	0.05	0.361	0.014	0.296	0.361	0.117	0.002
	0.50	0.377	0.126	0.295	0.363	0.122	0.015
	5.00	0.906	0.860	0.318	0.402	0.208	0.141
Stable limit cycles at bottom of euphotic zone	0	0.364	0.000	0.352	0.364	0.015	0.000
	0.05	0.366	0.017	0.353	0.365	0.019	0.011
	0.50	0.410	0.169	0.377	0.387	0.116	0.112
	5.00	1.360	1.29	1.01	0.974	0.945	0.915
Stable limit cycles in whole euphotic zone	0	0.109	0.000	0.112	0.109	0.038	0.000
	0.05	0.111	0.021	0.637	0.517	0.736	0.619
	0.50	0.234	0.205	1.40	1.20	1.51	1.36
	5.00	1.880	1.88	2.39	2.17	2.51	2.37

Here $D = \frac{\Phi-\Phi_h}{\int_\Omega 1 d\Omega} \times 100\%$ is the percent error per area in the domain. The column $\|D_i\|_2$ gives the initial difference, D_q indicates using quadratures, D_{qf} indicates quadrature-free, $\|.\|_2^{\mathrm{qp}}$ indicates the error evaluated at quadrature points, $\|.\|_2^{\mathrm{nd}}$ indicates the error evaluated at nodal points

for the quadrature-based treatment and at the nodal points for the quadrature-free implementation. For the case with single stable points (bio case 1), the initial difference of the perturbed solution to the analytical steady state is greater than the final difference, which indicates that the solution is approaching the calculated steady-state value. For the case with stable limit cycles in the euphotic zone (bio case 3), the final difference is greater than the initial perturbed difference, showing the solution is logically not approaching the steady state but instead spiraling outward toward the stable limit cycles present at each depth. Additionally, plotting the solution (Fig. 6) profile for the largest perturbation, we can see that the perturbed solution tends toward the steady state for the entire column for bio case 1, only for the top part of the water column for bio case 2 and nowhere for bio case 3. Thus, the parameter set with limit cycles in the euphotic zone (bio case 3) has the most structure in the vertical and will require the most resolution to model accurately. Also, numerical perturbations will be most important for bio case 3 because the differences will grow away from the calculated steady state, as opposed to decaying.

Vertical resolution By varying the resolution of the problem, we found that a minimum of 25 degrees of freedom were necessary to roughly capture the vertical structure of the biological model dynamics at the final time. For the tests in Section 4.4, ($g1$, $p6$) has approximately 21 degrees of freedom, indicating that it will be under-resolved.

High-order bases We find that the quadrature-based treatment of the source terms results in large jumps of the solution between elements. This is illustrated in Fig. 7 after 500 time units of integration using a 15th degree polynomial and three elements. The problem is amplified when using a uniform nodal spacing, due to a larger interpolation error. Also, increasing the number of quadrature points used for integration did not solve this problem. The problem originates from the discontinuous jump in the solution, causing oscillations known as Gibbs phenomena. Note that both simulations are initialized in the same manner, but the Gibbs oscillations can only be "seen" when evaluating the initial condition at points other than the nodes. The quadrature-based integration, then, "sees" these oscillations because the source terms are evaluated at the quadrature points. Using the quadrature-free approach for this 1D problem essentially decouples the vertical nodes, so numerically, the quadrature-free version does not "see" the oscillations. The Gibbs oscillations would have occurred in the quadrature-free scheme if the initialization was done at the quadrature points instead. This example illustrates one of the drawbacks of using increasingly higher-order schemes, that is, without special treatment, large oscillations occur for nonsmooth functions. Using lower order but on a finer discretization (more elements) can be a better strategy if special treatment is not used. This issue is further addressed in Sections 4.4 and 4.6.

In this section, we showed that, with our implementation, both the quadrature-based and quadrature-free treatment of the source terms give accurate, convergent results (see Table 4), although the absolute error of the quadrature-based implementation is smaller than the quadrature-free implementation. Then we showed that the analytical steady-state solution could be maintained and illustrated the dynamical behavior of three different biological parameter sets through perturbations of the steady-state solutions. With the vertical resolution tests, we found a minimum of 25 degrees of freedom necessary to roughly capture the vertical solution features of our particular setup. Finally, we showed that oscillations can occur solely due to numerics for a high-order discretization. While the quadrature-based algorithm was shown to be more accurate, the oscillations at element interfaces and the added numerical cost need to be considered. The additional accuracy may be warranted when a bifurcation of the solution could occur, or when the solution is under-resolved. As a whole, a key result is that, for any numerical scheme, careful numerical studies should be performed in one dimension to understand the potential errors arising from the nonlinear source term discretization before proceeding with advective models.

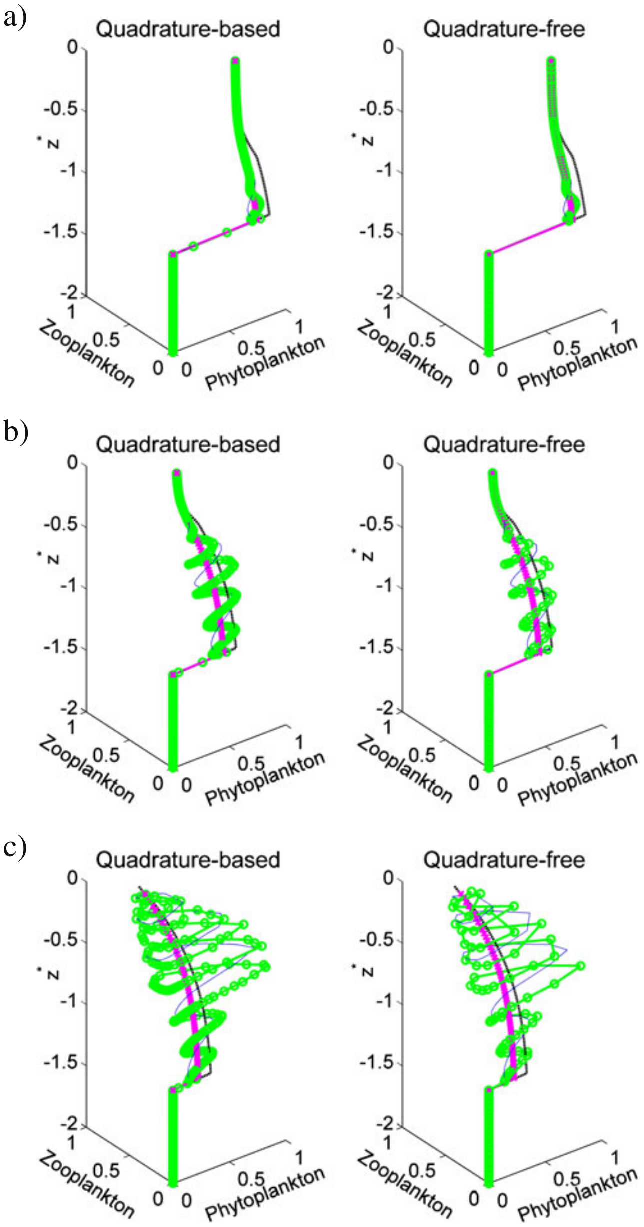

Fig. 6 Solution profiles at all depths with $\gamma = 5\%$. *Magenta crosses* show the analytical steady-state solution, the *thick black dashed lines* show the initial condition, *green circles* show the profile at $t^* = 250$, and *thin blue lines* show the profile at $t^* = 125$. Plotted for biological dynamics with **a** single stable points at all depths, **b** stable limit cycles at bottom of euphotic zone, and **c** stable limit cycles in entire euphotic zone. The quadrature-based solution is plotted at the quadrature points, whereas the quadrature-free solution is plotted at the nodal points

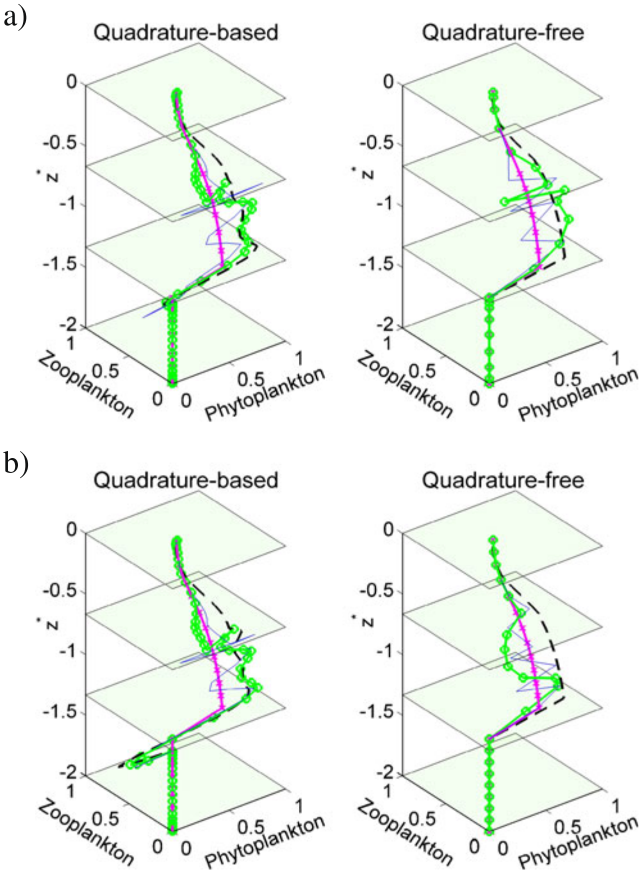

Fig. 7 Solution profiles for all depths at $t^* = 500$ using a 15th degree polynomial and three elements with $\gamma = 5\%$ for dynamics with stable limit cycles at the bottom of the euphotic zone. As in Fig. 6, the *magenta crosses* show the analytical steady-state solution, *thick black dashed lines* show the initial condition, *green circles* show the profile at $t^* = 250$, and *thin blue lines* give the profile at $t^* = 125$. The solution is plotted at the quadrature points for the quadrature version, and at the nodal points for the quadrature-free (i.e., where the source terms are evaluated). **a** Uses well-behaved (Gauss–Lobatto) nodal points, **b** uses uniform nodal points

4.2 Flow field convergence

A potential flow-field is calculated by solving Eq. 3 using HDG as described in Section 3.3. Once ψ is found, we take $\mathbf{u} = [\psi_z, -\psi_x]$. The value of ψ is specified on all boundaries. The top and bottom boundaries are specified as constants $\psi = \psi_{\text{top}}$ and $\psi = \psi_{\text{bot}}$. The inlet and outlet stream functions are specified to vary linearly in z^*, i.e., $\psi = \psi_{\text{top}} + z^*(\psi_{\text{bot}} - \psi_{\text{top}})/D^*$. For the advection-dominating cases in Section 4.3, periodic boundaries are used, in which case the values of ψ are equal at the inlet and outlet, with no need for boundary conditions. A flowfield specific to the grid and polynomial degree is used for all simulations.

We perform a convergence study on steady flowfields to verify that we are indeed obtaining near-optimal $\mathcal{O}(p+1)$ convergence (Cockburn et al. 2009; Nguyen et al. 2009), for the gradients of ψ used for the flowfield $\mathbf{u}$. To evaluate the convergence, reference solutions using $(g5, p2)$ and $(g2, p8)$ are calculated, both giving similar results. The error was evaluated by considering the point-wise solution at all interior vertices of grid 1. That is, the solution was averaged across all element-local solutions touching the vertex and compared to the reference solutions. The point-wise error calculated using $(g2, p8)$ as the true solution is plotted in Fig. 8 for multiple grids and polynomial bases. Using this point-wise error calculation, we obtained near-optimal rate of convergence. When the domain boundary nodes are included, nonoptimal (smaller and larger) rates of convergence were found for $p > 1$. This may be due to variations in the discretizations of the domain boundary for the different grids/polynomial bases. Similarly, when the $\|.\|_2$ or $\|.\|_\infty$ norms are considered, we find near-optimal convergence for $p = 1$, but not for $p > 1$. When the domain boundary is not curved, the flowfields with $p > 1$ have both the same rate of convergence and error magnitude as $p = 1$ using any of the error metrics mentioned. This shows that curved boundaries are essential for an accurate high-order solution. Also, by using HDG, we have a velocity field defined on the same grid with the same order of accuracy as our complete solution scheme for the ADR equations.

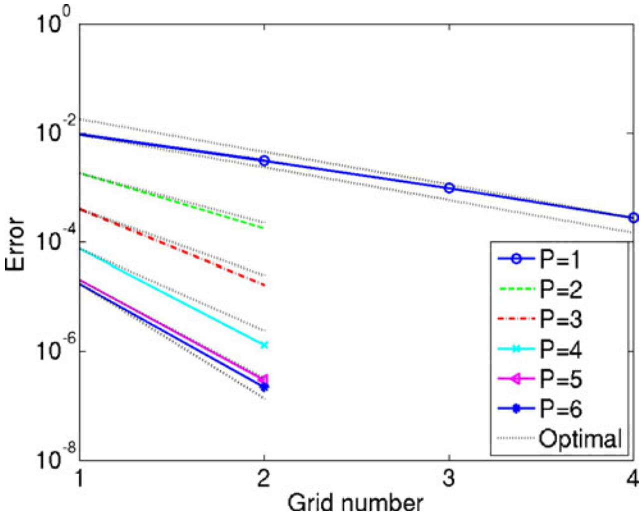

Fig. 8 Illustrating the convergence of the flow-field error. The point-wise error calculated using $(g2, p8)$ as the true solution for multiple grids and polynomial bases. Note that HDG gives near-optimal convergence for the derivative quantities $\mathbf{u} = [\psi_z, -\psi_x]$

Therefore, by using HDG with boundary-curved elements we obtain an accurate, high-order convergent potential flowfield for our geometry.

4.3 Tracer advection over bump test case results

We study tracer advection without source terms to evaluate the behavior of our numerical scheme when advection dominates. We do not illustrate here the convergence of the resolution (our results as shown in Sections 4.1 and 4.2). Instead we focus on numerical advection artifacts that may affect the behavior of the biology. We examine three cases: a uniform tracer, a tracer with a discontinuous jump in the vertical, and a tracer with a discontinuous jump in the horizontal and a linear horizontal gradient. The first case examines the divergence of the flowfield. The second case examines the behavior of the scheme in the presence of a horizontal jump, which occurs at the edge of the euphotic zone for our choices of parameters. Note that because we initialize by setting the numerical initial condition at the nodes equal to an analytical initial condition with a jump and the jump does not necessarily occur at element boundaries, the assigned numerical initial condition contains oscillations. The number of oscillations increases with the polynomial degree but is present at all orders. Also, the straightness of the interface is affected by the grid resolution. The third case examines the behavior of the scheme in the presence of a horizontal jump and horizontal gradient, which is a numerical test for frontal dynamics. Unlike the second case, the assigned initial condition for the third case does not contain oscillations because the jump occurs at element boundaries. All cases evolve in a periodic domain. In all cases, the flow is left to right, and the duration of the simulation is determined by the time the mean inlet velocity would take to travel through the domain. One of the objectives of these tests is to compare schemes at the same overall cost, for example, a lower-order scheme is used on a higher-resolution grid.

The results for the reference solution, $(g1, p5)$; low-order solution, $(g4, p1)$; and high-order solutions, $(g2, p5)$ and $(g1, p6)$ on curved meshes, are shown in

Fig. 9. In the top plot (case 1), we find that oscillations exceeding 10^{-4} are less localized for the low-order schemes than for the high-order schemes. From the middle and bottom plots (cases 2–3), we see that the higher-order schemes have larger magnitude oscillations around the jumps (as expected from the initialization). However, the sharpness of the interfaces are comparable between ($g1$, $p5$) and ($g2$, $p5$) and also between ($g4$, $p1$) and ($g1$, $p6$). Running a lower-order case ($g3$, $p1$) with 16,800 DOFs (not shown) resulted in a much more diffuse solution compared to ($g1$, $p6$), even though ($g1$, $p6$) is as under-resolved. The higher-order discretization with fewer degrees of freedom is therefore less numerically dissipative, illustrating one of the advantages of a high-order scheme.

The locations where the solution is outside of the intervals [0.9999, 1.0001], [0, 1], and [−2, 3] for cases 1, 2, and 3, respectively, are plotted in Fig. 10. From this figure, we see that the smallest magnitude and most localized errors for case 1 happen with high-order schemes using curved boundaries. The largest magnitude errors occur for high-order schemes using straight boundaries, and the least localized errors occur for low-order schemes. From case 2, we see that for both high- and low-order schemes, the initial oscillations do not remain close to the jump but spread through the domain, although the amplitude of the radiated oscillations are at least an order of magnitude smaller than the initialized oscillations. In case 3, we see that the high- and low-order schemes have similar performance, both developing numerical oscillations with the same order of magnitude around the jump. The absolute magnitude of the oscillations is approximately half the size with low order compared to high order and can be explained by the greater number of degrees of freedom with the low-order case. From these plots, we note that

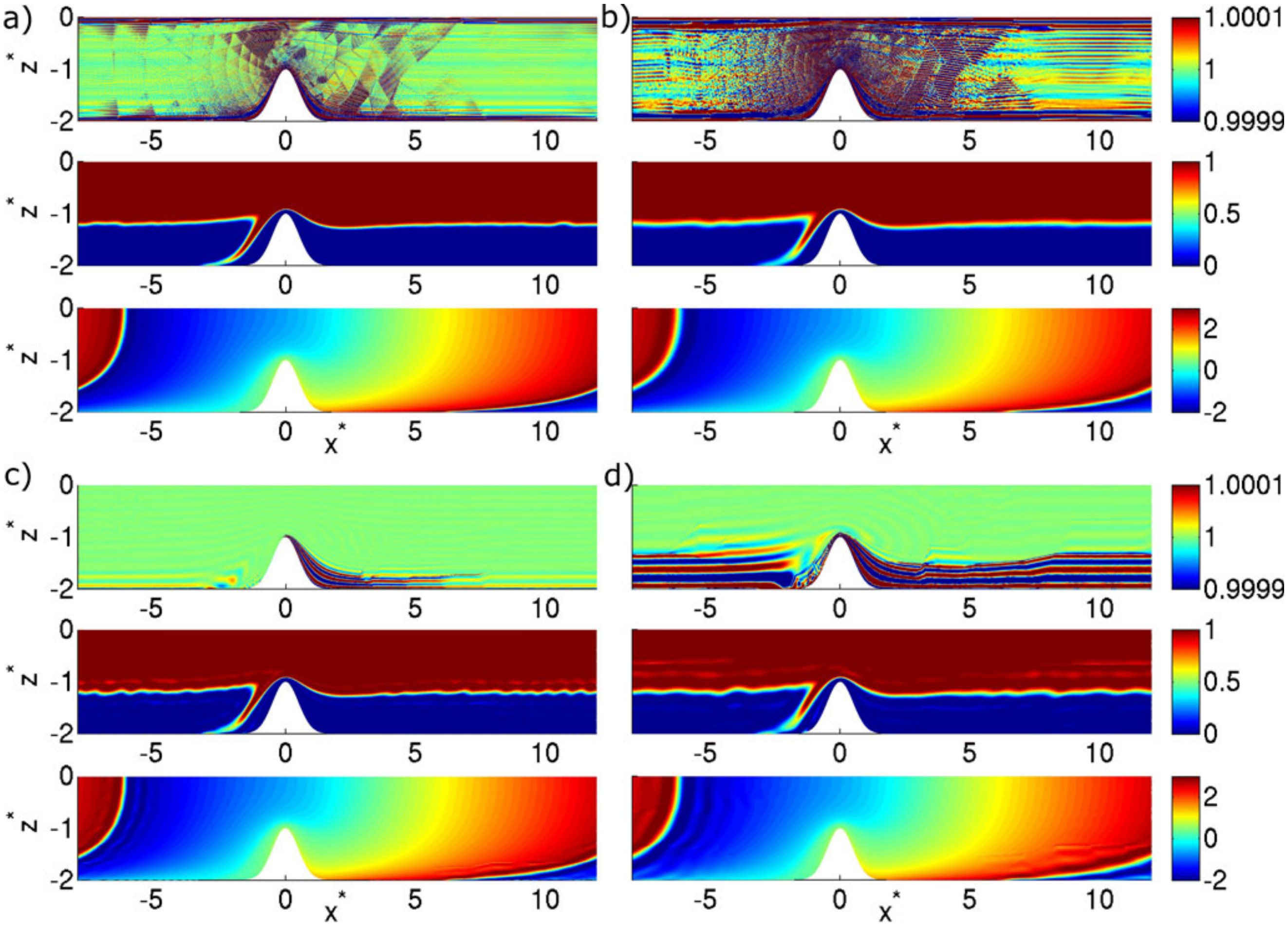

Fig. 9 Solution of three passive tracer cases, on **a** ($g5$, $p1$) with 268,800 DOFs, **b** ($g4$, $p1$) with 67,200 DOFs, **c** ($g2$, $p5$) with 29,400 DOFs, and **d** ($g1$, $p6$) with 9,800 DOFs with curved boundaries. *Top* plots show advection of a constant tracer, *middle* is advection of tracer with vertical jump, and *bottom* is advection with a horizontal jump and horizontal gradient. Note that **b** ($g4$, $p1$) and **c** ($g2$, $p5$) have similar costs while **d** ($g1$, $p6$) is much cheaper and under-resolved (see Table 3 for costs)

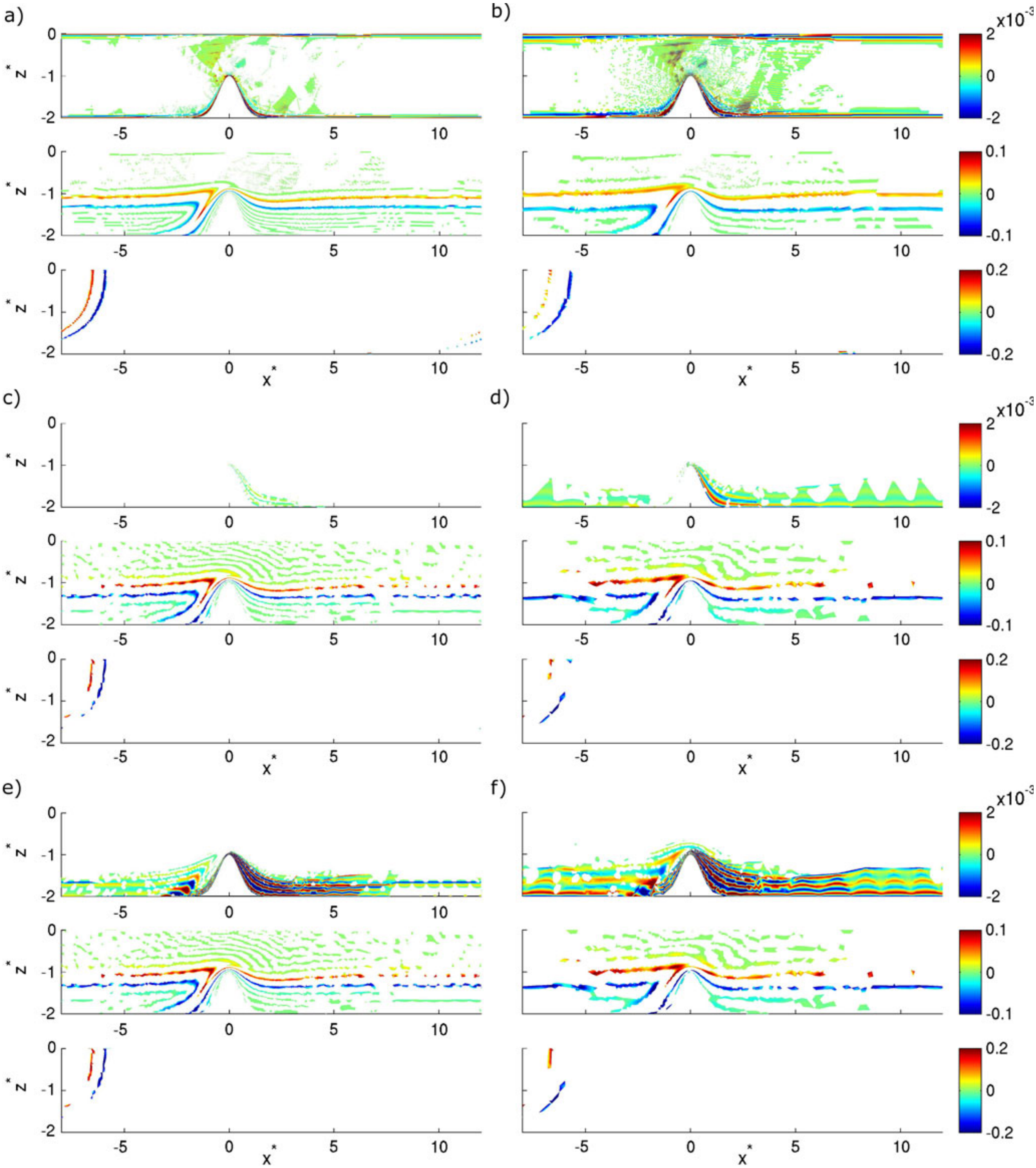

Fig. 10 Field values outside initially specified fields with **a** ($g5$, $p1$), **b** ($g4$, $p1$), **c** ($g2$, $p5$) on curved mesh, **d** ($g1$, $p6$) on curved mesh, **e** ($g2$, $p5$) on straight mesh, and **f** ($g1$, $p6$) on straight mesh. For each case (**a**–**e**), the *top* plot shows advection of a constant tracer with interval [0.9999, 1.0001], *middle* is advection of tracer with vertical jump and interval [0, 1], and *bottom* is advection with a horizontal gradient and horizontal jump with interval [−2, 3]

the magnitude of the oscillations are within 20% of the solution, and this could have a significant effect on the biology. This illustrates that discontinuous or highly nonsmooth functions caused by physics or biology are problematic. In the context of this work, the jump will be smoother for the test cases in Section 4.4 where we examine the case for approximately balanced advection and biological source terms. Alternatively, a shock-capturing (see, for example, Hoteit et al. 2004; Persson and Peraire 2006; Krivodonova 2007), filtering (for example, Hesthaven and Kirby 2008), or postprocessing technique could be used to handle the discontinuity (see, for example, Cockburn et al. 2003; Qiu and Shu 2005). In Section 4.6, we illustrate how a filtering approach can be used to damp oscillations for high-order schemes. Note that special treatment is required for both high- and low-order schemes, with a slope-limiting procedure more often used for the latter.

We note that the large oscillations in case 1 for the high-order scheme happen only down-stream from the peak. This is because we do not ensure that the definition for the discrete divergence is the same in the equations calculating the potential flow field and the tracer/biological ADR equations (see, for example, Dawson et al. 2004). Therefore, we expect to see discretization errors appearing in the ADR discrete divergence operator, which are advected downstream. Examining the normalized discrete divergence as defined in the ADR equations, $\left(\frac{\nabla\cdot\mathbf{u}}{|\mathbf{u}|}\right)$, of curved and straight meshes with $(g1, p6)$ and $(g4, p1)$, we found that the largest divergence errors occurs for the higher-order scheme on the straight mesh. The normalized divergence, in this case, was of $\mathcal{O}(1)$ near the peak, and $\left\|\frac{\nabla\cdot\mathbf{u}}{|\mathbf{u}|}\right\|_2 \approx 0.56$ for the domain. The normalized divergence for the low-order and high-order curved schemes were of $\mathcal{O}(10^{-2})$ near the peak, with the maximum error for the low-order scheme approximately half the size of the high-order scheme. However, the error for the high-order scheme was more localized, resulting in a lower volume-averaged error $\left\|\frac{\nabla\cdot\mathbf{u}}{|\mathbf{u}|}\right\|_2 \approx 0.044$, than that of the low-order scheme which was $\left\|\frac{\nabla\cdot\mathbf{u}}{|\mathbf{u}|}\right\|_2 \approx 0.11$. This reveals that it is important to use a curved mesh for higher-order schemes: The resultant flow field is then less numerically divergent than a flow field solved with a low-order finely resolved scheme. Without using a curved mesh, the divergence can be of the same order as the velocity near the geometry, which may excite nonphysical biological dynamics downstream of the peak. In our case (in Section 4.4), the peak was beneath the euphotic zone and thus using a straight boundary representation would not cause problems; however, in the general case, this result cannot be overlooked. Of course, our advection scheme is conservative, but to also be constancy preserving, the numerical flow field needs to be discretely divergence-free.

Since the $(g1, p6)$ simulations took less computational time than $(g4, p1)$ and because the $(g2, p5)$ simulation had a sharper interface, these results suggest that higher-order schemes performs better than the lower-order schemes for the advection dominated case, as long as curved boundaries are used.

4.4 Full NPZ equations

In this section, we explore the case where the advection and biological source terms are approximately balanced. We examine effects of low-order and high-order temporal discretizations in Section 4.4.1. In Section 4.4.2, we illustrate the difference between using a quadrature-based and quadrature-free scheme to discretize the nonlinear biological source terms. Finally, in Section 4.4.3, we study effects of spatial resolution, through both grid resolution and polynomial degree.

We still study the three biological parameter sets: single stable points, stable limit cycles at the bottom of the euphotic zone, and stable limit cycles for the entire euphotic zone, as given in Table 2. Since the timescale of biology varies in depth, the advection and biological source terms can only be balanced for one depth. While this results in many choices of approximately balanced parameter sets, we focus on one where $\bar{\tau}_a = 12.5$ days. For these tests, the inlet is specified as the steady-state solution with a smoothed discontinuity. The discontinuity is smoothed by fitting it with a cubic polynomial which can be resolved on $(g1, p6)$. The fit is biased such that 3/4 of the polynomial is below the euphotic zone. For the outlet boundary, we use $\frac{\partial\Phi}{\partial\hat{\mathbf{n}}} = 0$. Also these results are compared to a $(g5, p1)$ simulation, which is taken as the true solution.

The final solution fields for the three different regimes of biological dynamics (from Table 2) and using quadrature-based source terms for $(g5, p1)$ (the reference solution) is plotted in Fig. 11. The results show that idealized strait bathymetry effectively perturbs the biology away from the inlet conditions. The case with single stable points (bio case 1) adjusts back to the stable equilibrium, whereas the two cases with limit cycles show complex structures in the vertical. In all cases, a phytoplankton bloom over the bump is observed.

To qualitatively evaluate the effect of refining the grid or polynomial degree, we show the solution field for phytoplankton for $(g3, p1)$, $(g3, p2)$, $(g4, p1)$, and $(g4, p2)$ in Fig. 12, and these discretizations have

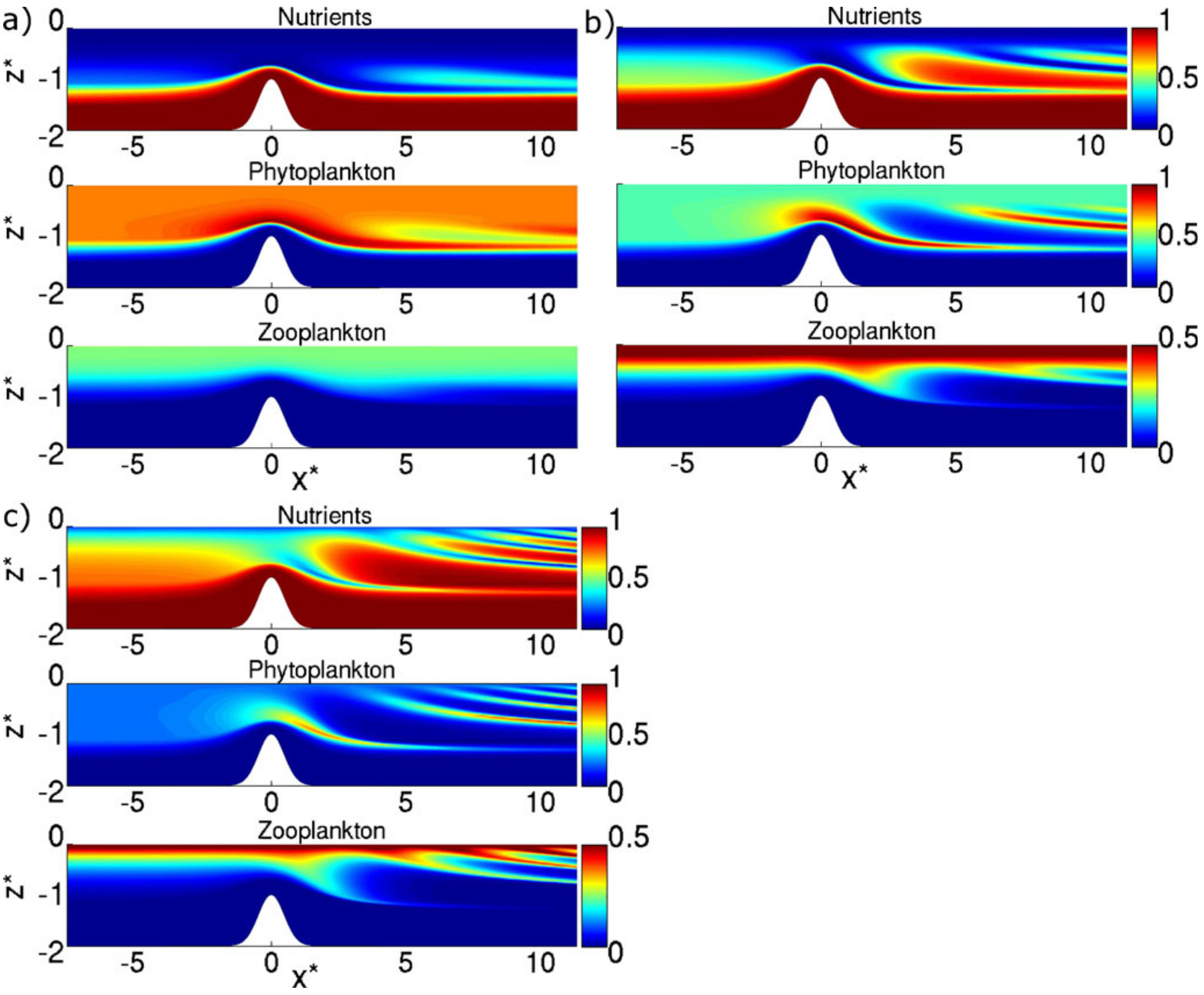

Fig. 11 Biological dynamics at $t^* = 20$ (with $\bar{\tau}_a = 12.5$ days) using ($g5$, $p1$). Biolocial dynamics with **a** single stable points, **b** stable limit cycles for depths $z^* = 0.4$–0.9, and **c** stable limit cycles in whole euphotic zone. This is the reference solution against which all other solutions are compared

16,800, 33,600, 67,200, and 134,400 DOFs, respectively. This figure shows that the solution is converging with increased resolution. More quantitative comparisons are completed next.

4.4.1 Comparing low-order and high-order temporal discretizations

We compare the solutions using fourth-order Runge–Kutta, second-order Runge–Kutta, and first-order Explicit Euler on ($g2$, $p4$) for the biology with stable limit cycles in the euphotic zone (bio case 3). The differences of the lower-order schemes compared to fourth-order Runge–Kutta at $t^* = 40$ is plotted in Fig. 13 for the phytoplankton field. Note that the timestep size for the first-order scheme is half of the second-order scheme, such that the cost of the two are the same. For this test case, we used periodic boundary conditions. From the figure, we note that the major differences occur within the euphotic zone. The stable explicit timestep for the second-order scheme is set by the Courant condition for the advection discretization, and since the largest velocity occurs in the smallest element for this discretization, the timestep size is approximately four orders of magnitude smaller than the biological timescale. Therefore, it is expected that temporal errors in the source term should be small even for the low-order scheme. Nonetheless, we still observe differences between the first-, second-, and fourth-order schemes. We

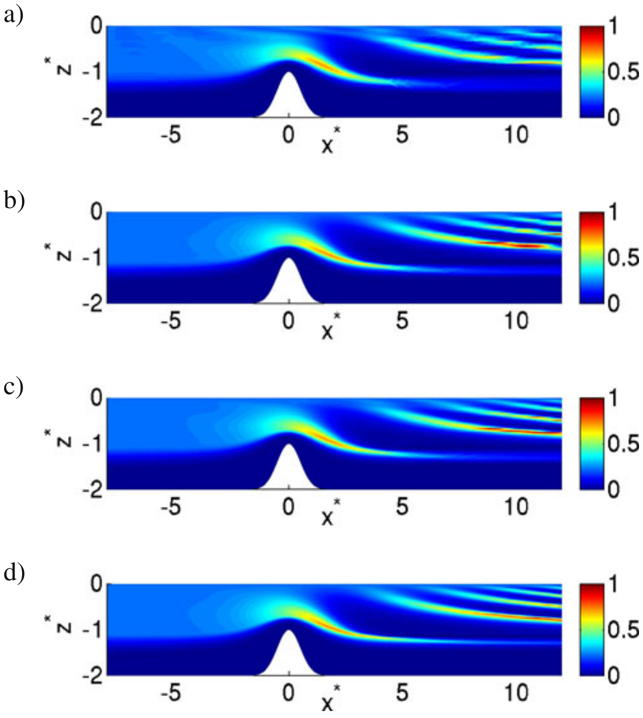

Fig. 12 Phytoplankton fields at time $t^* = 20$ (with $\bar{\tau}_a = 12.5$ days), as computed using four different spatial resolutions and order of the FE scheme: **a** ($g3$, $p1$), (16,800 DOFs) **b** ($g3$, $p2$) (33,600 DOFs), **c** ($g4$, $p1$) (67,200 DOFs), and **d** ($g4$, $p2$) (134,400 DOFs). All fields are for biological dynamics with stable limit cycles in the euphotic zone (bio case 3 in Table 2)

found that the difference at $t^* = 40$ is approximately two orders of magnitude larger than at $t^* = 20$, which indicates that the errors are growing quickly. For the first-order scheme, the maximum error is of $\mathcal{O}(1)$ at $t^* = 40$. This suggests that a low-order time discretization may result in significant errors when long integration times or fast biological timescales are involved. For example, the latter occurs in coastal applications. As another example, for stiff biogeochemical source terms, Burchard et al. (2005) found that even fourth Runge–Kutta integration is insufficient to maintain the nonnegativity of the biological components. They suggest that positivity preserving Patankar–Runge–Kutta schemes should be used to obtain a nonnegative, conservative solution.

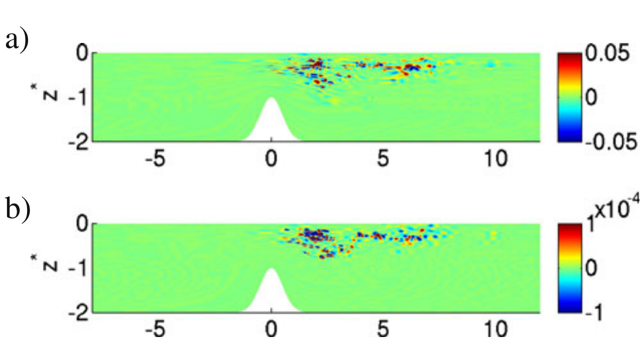

Fig. 13 Temporal discretization differences for phytoplankton field with stable limit cycles in euphotic zone at $t^* = 40$ using periodic boundary conditions and on spatial grid ($g4$, $p2$). **a** "First-order Euler" minus "fourth-order Runge–Kutta" and **b** "second-order Runge–Kutta" minus "fourth-order Runge–Kutta"

4.4.2 Comparing quadrature-based and quadrature-free source terms

In Section 4.1, we found that the greatest difference between the quadrature and quadrature-free treatment of the source terms occurred for the biological parameter set with stable limit cycles in the euphotic zone (i.e., bio case 3). Here we examine this case for full ADR dynamics using the ($g1$, $p6$) discretization. Note that we obtained the same results and conclusions with the ($g4$, $p1$) and ($g5$, $p2$) discretizations (not shown). Plotting the difference (quadrature-free minus quadrature-based) of the solution in Fig. 14 for ($g1$, $p6$), we see that the largest differences occur near the outlet of the domain where the mesh solution is under-resolved. The quadrature-based solution is more accurate in the under-resolved region because the source-term integral is more accurately evaluated, and this was verified by comparing the errors of the two schemes. However, where the solution is sufficiently resolved, the quadrature-free and quadrature-based treatments of

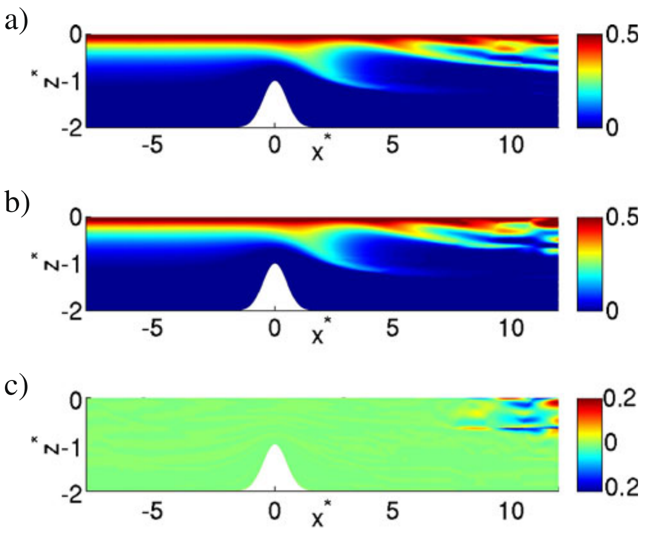

Fig. 14 Zooplankton fields at $t^* = 20$ computed using ($g1$, $p6$) and **a** quadrature-based source terms and **b** quadrature-free source terms. **c** The difference between the quadrature-free and quadrature-based source-term simulations. The biological dynamics used has stable limit cycles within the euphotic zone (bio case 3)

the source terms have similar accuracy, that is, they differ by approximately 0.1%. From the 1D studies, we found that the quadrature-free algorithm was less oscillatory at element interfaces than the quadrature-based algorithm, and we observed the same effect in these 2D simulations for $p > 7$ on $g1$, although the difference between quadrature-free and quadrature-based was less drastic. The largest differences between the quadrature-based and quadrature-free schemes did occur at element boundaries, and the quadrature-based algorithm was more accurate when under-resolved.

Using Eq. 30, we verify the conservation of the scheme. The results for the quadrature-free and quadrature-based source terms were similar up to floating point precision. Also, we find that the conservation error is dominated by the flow field divergence error. Therefore, the conservation properties of the source term discretization does not affect the choice between quadrature-free and quadrature-based algorithms.

Because the quadrature-free and quadrature-based algorithms had similar accuracy in well-resolved regions, we recommend using the quadrature-free treatment in these regions because of the improved efficiency. However, when the solution is poorly resolved, the quadrature-based treatment of the source terms is more accurate. Now, depending on the total solution cost of a particular numerical scheme, a finer resolution quadrature-free scheme may be more efficient for the same accuracy than a quadrature-based scheme.

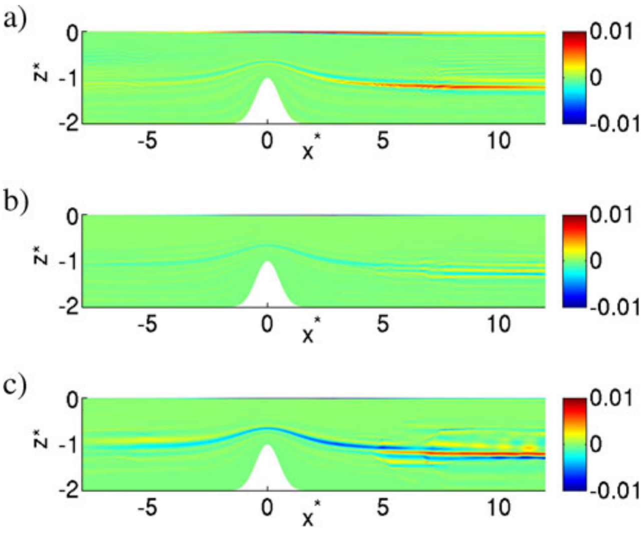

Fig. 15 Difference between zooplankton fields at $t^* = 20$ (with $\bar{\tau}_a = 12.5$ days) computed using ($g5$, $p1$) and **a** ($g4$, $p1$), **b** ($g2$, $p5$), and **c** ($g1$, $p6$). This shows the locations of the largest numerical errors for the high-order and low-order schemes. The biological dynamics used have single stable points at all depths (bio case 1)

4.4.3 Comparing low-order and high-order spatial discretizations

Figure 15 shows the differences between the reference solution and the solutions using other grids and polynomial degrees in Fig. 15 for zooplankton. We see that the ($g2$, $p5$) simulation has the smallest differences and is therefore the most accurate. This is a key result since it indicates that when the solution is resolved, for the same cost/efficiency, a higher-order scheme on a coarser grid performs better than a lower-order scheme on a finer grid. Results for the biological dynamics with limit cycles at the bottom of the euphotic zone (bio case 2) are similar, but for the biology with limit cycles in the entire euphotic zone (bio case 3), both high- and low-order schemes are under-resolved for $x^* > 7$, especially for the ($g1$, $p6$) scheme. The differences between a fine grid solution ($g5$, $p1$) and the low-order and high-order schemes are plotted in Fig. 16 for zooplankton. From Fig. 16, we note that the errors in the low-order scheme are more localized in the $x^* > 7$ region. However, the differences for ($g4$, $p1$) and ($g2$, $p5$) are similar in the $x^* > 7$ region. The ($g1$, $p6$) scheme has the least localized and largest magnitude errors in the $x^* > 7$ region. However, as plotted in Fig. 17 where the solution is smooth and the biology has less structure in the vertical, both the high-order schemes are more accurate than the low-order scheme. Particularly, note the solution near the surface for $x^* < 7$ in Fig. 17.

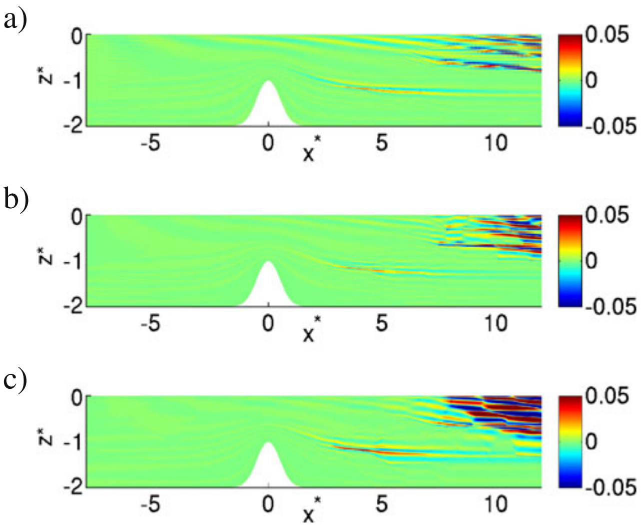

Fig. 16 As Fig. 15, but for the biological dynamics with stable limit cycles within the euphotic zone (bio case 3)

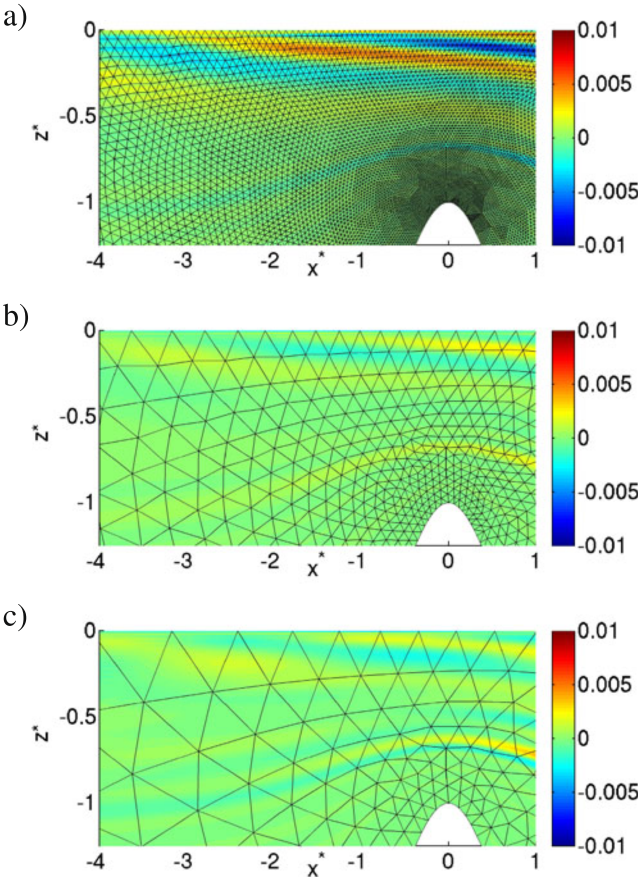

Fig. 17 As Fig. 16, but zoomed in the region above the bathymetry. The difference between zooplankton fields using $(g5, p1)$ and **a** $(g4, p1)$, **b** $(g2, p5)$, and **c** $(g1, p6)$

We examine the error characteristics of these fields more closely by considering the truncated Taylor expansions of the true solution. By the mean value theorem, the truncation error for $(g4, p1)$ is $\frac{h^2}{2!}\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)^2 \phi(\mathbf{x}_\eta)$ for some unknown point $\mathbf{x}_\eta$, and for $(g2, p5)$ and $(g1, p6)$ these terms are $\frac{h^7}{6!}\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)^6 \phi(\mathbf{x}_\zeta)$ and $\frac{h^7}{7!}\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)^7 \phi(\mathbf{x}_\xi)$ for unknown points $\mathbf{x}_\zeta$ and $\mathbf{x}_\xi$, where h is the characteristic discretization length of an element. Now, we can examine the approximate truncation error by running simulations $(g4, p2)$, $(g2, p6)$, and $(g1, p7)$ and evaluating the highest-order nonzero derivatives of the approximate solution ϕ_h. To evaluate the derivatives, we interpolate the solution onto an orthogonal modal polynomial basis, that is, we find the coefficients a_{ij} such that $\phi_h = \sum_i \phi_i \theta_i = \sum_{ij} a_{ij} P_{ij}$, where P_{ij} is a modal orthogonal polynomial with maximum degree of i on x and j on y, for a maximum total degree of $i + j$. The derivatives then evaluate as $\frac{1}{2!}\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)^2 \phi_h(\mathbf{x}_\eta) = \sum_{i+j=2} a_{ij}$, $\frac{1}{6!}\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)^6 \phi_h(\mathbf{x}_\xi) = \sum_{i+j=6} a_{ij}$, and $\frac{1}{7!}\left(\frac{\partial}{\partial x}+\frac{\partial}{\partial y}\right)^7 \phi_h(\mathbf{x}_\xi) = \sum_{i+j=7} a_{ij}$, that is, we simply need to sum the coefficients of the modal orthogonal polynomial basis which correspond to terms with total degree of 2, 6, and 7, respectively. Since the coefficients of the polynomials are evaluated on the reference element, $h \approx 1$ will be the same for all elements. Also, while this approach gives an estimate of the leading order truncated term, it does not give an exact value. In our case, we are not interested in a rigorous error estimator, but instead we only require an estimate of the error to aid the discussion.

Our approach is similar to that followed by Mavriplis (1989), where Legendre polynomials were used instead. The author proposed a smoothness estimator, where the coefficients a_{ij} are fit to the exponentially decaying function $a(i + j) = Ce^{\sigma(i+j)}$. There the author claims that $\sigma < -1$ indicates good resolution or smooth functions and $\sigma > -1$ indicates poor resolution or nonsmooth functions. The adaptive strategy used was to increase the polynomial degree for elements with $\sigma < -1$ and to refine the mesh for elements with $\sigma > -1$, if the error level in that element was insufficient. We evaluate this smoothness indicator σ on $(g1, p7)$ by doing a least squares fit of the coefficients to $Ce^{\sigma(i+j)}$. In regions where the magnitude of the solution is close to 0, that is below the euphotic zone for the zooplankton field, a_{ij} $\forall i, j$ will be small, and the smoothness indicator σ will not be accurate. The approximate size of the truncated derivative terms along with the smoothness indicator are plotted in Fig. 18. Only the smoothness indicators calculated on $(g2, p5)$ and $(g1, p7)$ are plotted since the accuracy of the smoothness indicator improves with the number of terms in the polynomial expansion and is not accurately represented on $(g4, p2)$ (Mavriplis 1989).

From Fig. 18, we note that the largest differences in Fig. 16 correspond to the regions with the largest approximate truncations errors in Fig. 18. Also, in the region $x^* > 7$ where the low-order solution is more accurate than $(g1, p6)$, we have $\sigma > -1$, which suggests that refining the elements instead of the order of accuracy is more appropriate. After one level of refinement on $(g2, p5)$, we see that the smoothness indicator shows a smaller region of nonsmooth elements. This illustrates that the smoothness is defined in terms of the numerical discretization and is not solely a function of the solution field. Also note in the region where the high-order solution is more accurate (see Fig. 17), the approximate derivative of the truncation term is small in both fields and $\sigma < -1$, suggesting that a higher degree polynomial basis is more appropriate in this region. This shows

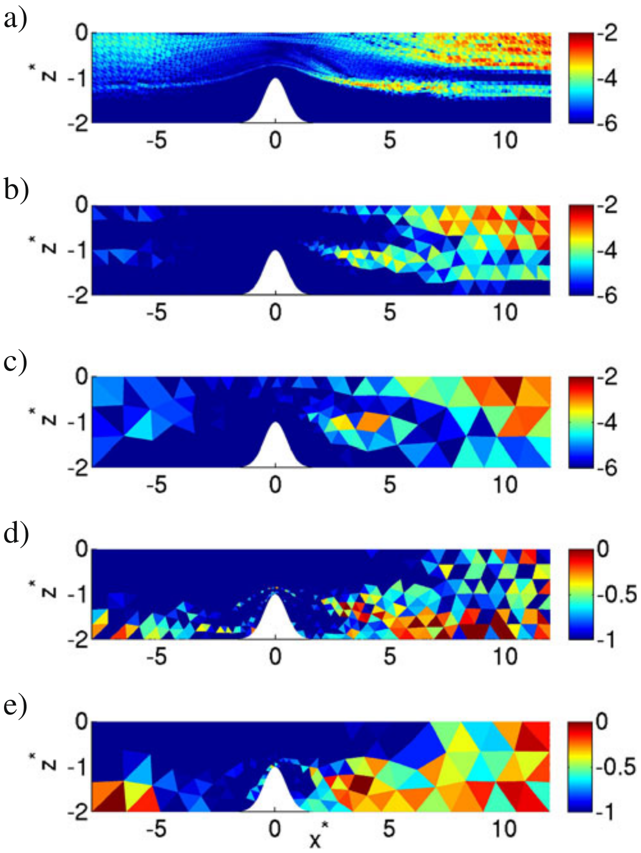

Fig. 18 Approximate truncation errors for zooplankton fields at $t^* = 20$ (with $\bar{\tau}_a = 12.5$ days). Calculated on **a** ($g4$, $p2$) using $log_{10}(\sum_{i+j=2} a_{ij})$, on **b** $g2$, $p6$ using $log_{10}(\sum_{i+j=6} a_{ij})$, and on **c** $g1$, $p7$ using $log_{10}(\sum_{i+j=7} a_{ij})$. **d**, **e** Smoothness indicator σ calculated on **d** ($g2$, $p6$) and **e** ($g1$, $p7$) . The biological dynamics used has stable limit cycles within the euphotic zone (bio case 3)

that our mesh is not optimized in terms of the solution field and highlights the importance of using both mesh refinement and polynomial basis adaptation to generate an optimal discretization for complex biological ocean dynamics. Also, this shows that whether a coarsely discretized higher-order scheme is better than a finely discretized lower-order scheme depends on the smoothness of the solution and can vary spatially across the solution. The benefit from a higher-order solution is as follows: When the solution is smooth, increasing the polynomial degree causes the error to decrease exponentially, whereas the error would only decrease algebraically if decreasing the element size. The cost of increasing the polynomial degree also scales algebraically, and because of this, a higher-order scheme performs better for smooth or well-resolved fields. Using our implementation, the ($g1$, $p6$) simulation took approximately 0.34 of the time taken by the ($g4$, $p1$) simulation. We also ran ($g3$, $p1$), which was approximately three times more efficient than ($g1$, $p6$), but this solution (not shown) was less than 1% accurate for the majority of the domain. When the solution is not resolved (i.e., not smooth for the grid resolution or polynomial degree chosen), higher-order schemes will lead to Gibbs oscillations and filtering is required (see Section 4.6), while lower-order schemes may "look good" but will be very dissipative. When the solution is resolved (i.e., smooth enough for the grid resolution or polynomial degree chosen), higher-order discretizations perform better than lower-order ones: They are more accurate and less dissipative for the same cost.

Finally, we note that the approximate truncation error and smoothness metrics were different for the different biological components. Therefore, the optimal discretization for one component is not the same as the optimal discretization for another component. Ueckermann (2009) proposed a scheme that uses a different order basis function for different biological components, but also cautions that an incurred interpolation cost needs to be considered for adaptation strategies.

4.5 Evolution of biological patch

In this section, we demonstrate how biological activity can enhance the differences between low-order and high-order discretizations beyond the effect of numerical dissipation alone. For this example, we modify bio case 1 (single stable points at all depths) from Section 4.4 by introducing a vertical column, or "patch", of biology that uses the parameters from bio case 2 (stable limit cycles at depths $z^* = 0.4$–0.9 and single stable points elsewhere). This is easily done in the dimensional form of the equations by increasing the value of $\mathcal{N}_T$ locally in the patch. Such situations occur frequently in nature, e.g., an eddy or front upwelling additional nutrients locally toward the surface. The initial condition and boundary condition is the same as in bio case 1 (the steady-state solution with smoothed discontinuity), except inside the patch where the initial conditions for bio case 2 are used instead, that is:

$$\phi^*_{\text{(patch)}} = \phi^*_{\text{(bio case 1)}} + \left[\phi^*_{\text{(bio case 2)}} - \phi^*_{\text{(bio case 1)}}\right] \cdot e^{-\frac{(x^*+6.4)^4}{2\cdot(8^4)}}, \qquad (31)$$

where $\phi^*_{\text{(patch)}}$ is the initial condition used for this example, $\phi^*_{\text{(bio case 1)}}$ is the steady state with smoothed discontinuity for bio case 1, and $\phi^*_{\text{(bio case 2)}}$ is the steady state with smoothed discontinuity for bio case 2. Note that, for this example, we nondimensionalize $\phi^* = \frac{\phi}{\mathcal{N}_{T,\text{bio case 1}}}$ by the total biomass for bio case 1. In addition, we superimpose a periodic velocity onto the mean velocity,

$$\mathbf{u}^* = \mathbf{u}^*_{\text{mean}}[1 + 5 \cdot \text{sign}\{\cos(0.16\pi t^*)\}], \tag{32}$$

where $\mathbf{u}^*$ is now the velocity used for this example and $\mathbf{u}^*_{\text{mean}}$ is the potential flowfield solved from Section 4.2. The superimposed velocity increases the distance traveled, as well as the number of time integration steps (due to the CFL condition), and therefore has the effect of increasing the numerical dissipation.

Figure 19 plots the phytoplankton fields and total biomass for $(g2, p5)$ and $(g4, p1)$ around the patch, as well as the difference of the solution, $(g2, p5)$ minus $(g4, p1)$, at $t^* = 14.4$. We do not use the $(g5, p1)$ solution (as was done in Section 4.4) for the difference plots here because we found that even $(g5, p1)$ is more dissipated than $(g2, p5)$ and therefore $(g2, p5)$ is more accurate inside the biological patch where our calculations are performed. These results show that the total biomass peak is not maintained by the low-order scheme, $(g4, p1)$, and the details in the phytoplankton fields are also dissipated. Since these simulations do not contain physical diffusion, any diffusion is due to the numerical scheme, and therefore, the $(g2, p5)$ solution is more accurate than $(g4, p1)$ because it does maintain the total biomass peak. Apart from the effects of the periodic velocity, the solution inside the patch should resemble that of Fig. 11b), and $(g2, p5)$ resembles this solution more closely than $(g4, p1)$.

While some of the differences between the $(g2, p5)$ and $(g4, p1)$ simulations can be accredited solely to the numerical dissipation, the error due to numerical dissipation is amplified by the change in biological activity. To illustrate this point, we show in Fig. 20 the relative normed difference between the total biomass of the two solutions (Q_1, Eq. 33), the sum of relative normed differences between the biological components (Q_2, Eq. 34), the relative normed difference in production

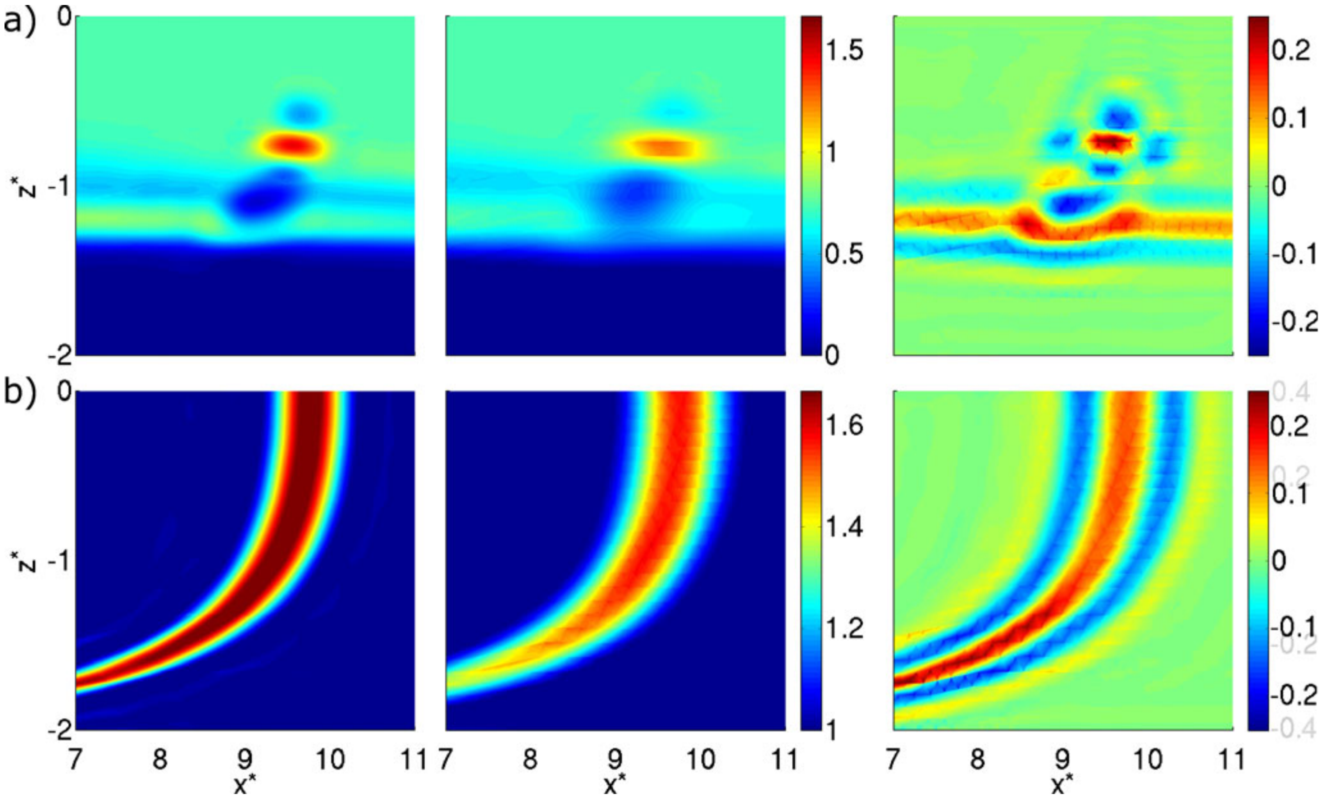

Fig. 19 Detail around the biological patch with stable limit cycles at the bottom of the euphotic zone at time $t^* = 14.4$ for **a** the phytoplankton fields and **b** the total biomass. The solution for $(g2, p5)$ is plotted on the *left*, $(g4, p1)$ in the *middle*, and the difference between the solutions, $[(g2, p5) - (g4, p1)]$, is plotted on the *right*. This shows that $(g2, p5)$ correctly maintains the full peak of the biological patch, while $(g4, p1)$ does not, leading to large differences in the phytoplankton fields

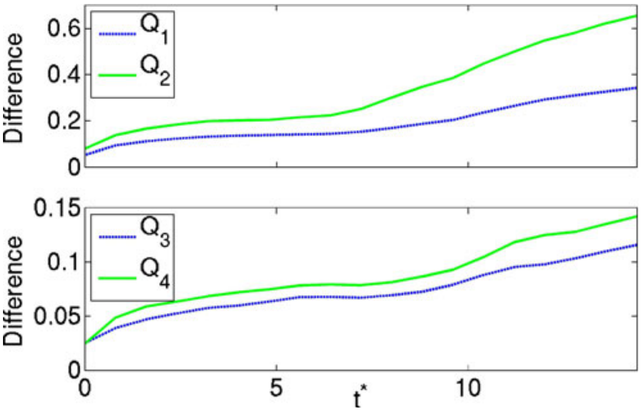

Fig. 20 The relative normed difference between the total biomass of the two solutions (Q_1, Eq. 33), the sum of relative normed differences between the biological components (Q_2, Eq. 34), the relative normed difference in production (Q_3, Eq. 35), and the relative normed difference in grazing (Q_4, Eq. 36) over time from $t^* = 0$ to $t^* = 14.4$. This shows that the difference in biological components is amplified beyond the effect of numerical dissipation due to differences in the source terms such as the production and grazing

(Q_3, Eq. 35), and the relative normed difference in grazing (Q_4, Eq. 36)

$$Q_1 = \frac{\left\| \{\phi_N^* + \phi_P^* + \phi_Z^*\}_{(g2,p5)} - \{\phi_N^* + \phi_P^* + \phi_Z^*\}_{(g1,p4)} \right\|_2^{\text{patch}}}{\left\| \{\phi_N^* + \phi_P^* + \phi_Z^* - 1\} \right\|_2^{\text{patch}}} \tag{33}$$

$$Q_2 = \frac{\sum_{I=(N,P,Z)} \left\| \phi^*_{I,(g2,p5)} - \phi^*_{I,(g1,p4)} \right\|_2^{\text{patch}}}{\left\| \{\phi_N^* + \phi_P^* + \phi_Z^* - 1\}_{(g4,p1)} \right\|_2^{\text{patch}}} \tag{34}$$

$$Q_3 = \frac{\left\| \left\{ \mathcal{U}^* e^{z^*/h^*} \frac{\phi_P^* \phi_N^*}{\phi_N^* + k_s^*} \right\}_{g2p5} - \left\{ \mathcal{U}^* e^{z^*/h^*} \frac{\phi_P^* \phi_N^*}{\phi_N^* + k_s^*} \right\}_{g4p1} \right\|_2^{\text{patch}}}{\left\| \left\{ \mathcal{U}^* e^{z^*/h^*} \frac{\phi_P^* \phi_N^*}{\phi_N^* + k_s^*} \right\}_{g4p1} \right\|_2^{\text{patch}}} \tag{35}$$

$$Q_4 = \frac{\left\| \{ag_\nu^* \phi_Z^* (1 - e^{-\nu^* \phi_P^*})\}_{g2p5} - \{ag_\nu^* \phi_Z^* (1 - e^{-\nu^* \phi_P^*})\}_{g4p1} \right\|_2^{\text{patch}}}{\left\| \{ag_\nu^* \phi_Z^* (1 - e^{-\nu^* \phi_P^*})\}_{g4p1} \right\|_2^{\text{patch}}}, \tag{36}$$

where $\|e\|_2^{\text{patch}} = \left(\int_{\text{patch}} e^2 dx^* dz^* \right)^{\frac{1}{2}}$ with the patch area determined from $(g4, p1)$ and the quantity $\|\{\phi_N^* + \phi_P^* + \phi_Z^* - 1\}\|_2^{\text{patch}}$ gives the size of the difference between the base solution and the solution inside the patch since the base number of nutrients (nondimensionalized to 1) is subtracted out.

Since our numerical scheme conserves the total biomass, the first quantity, Q_1, gives a quantitative estimate of the numerical dissipation error only. The second quantity, Q_2, should be the same as Q_1 if the only difference is due to numerical dissipation. However, from Fig. 20, we note that $Q_2 > Q_1$, which means the differences in dissipation is amplified by nonlinearities and the biology. This is explained by the differences in biological terms in the two simulations, for example, in the production and grazing terms, Q_3 and Q_4. Also, note that these differences are growing over time, and for longer integration periods, the differences will be even greater. As a final note, the initial differences between the two solutions are due to interpolation errors, since the polynomial representation and number of degrees of freedom are not the same for the two simulations.

The example shows that the numerical dissipation due to a lower-order numerical scheme can be amplified by the biological reaction terms. This is significant since for accurate biological ocean science through numerical simulations, it is important to maintain the amplitudes of biological patches. This is particularly true for biology with multiple attractors, where relatively small perturbations can lead to vastly different solutions. The conclusion is that for the same cost, higher-order schemes on coarser grids are more accurate than lower-order schemes on finer grids.

4.6 Filtering based on smoothness index

Based on the results in Section 4.4 and on the paper by Hesthaven and Kirby (2008), we created a selective exponential filter. We found that the default filter was described in Hesthaven and Kirby (2008), which is applied at every timestep, to result in a diffuse interface, even for high-order filters. This prompted us to develop the following filter. Consider a modal representation of the numerical solution, $\phi_h = \sum_{ij} a_{ij} P_{ij}$ (e.g. Ueckermann 2009). After each time integration step, the solution on every element is modified as follows:

$$\phi_h^F = \sum_{ij} \sigma(\eta_{ij}) a_{ij} P_{ij}, \tag{37}$$

where

$$\sigma = \begin{cases} \exp(-\alpha \eta_{ij}^s), & \text{if SI} > 0 \\ 1, & \text{if SI} < 0 \end{cases} \tag{38}$$

$$\text{SI} = \frac{a_{ij}}{\max_{i+j \geq 1}(a_{ij})} - \exp(-(i+j)+1) \tag{39}$$

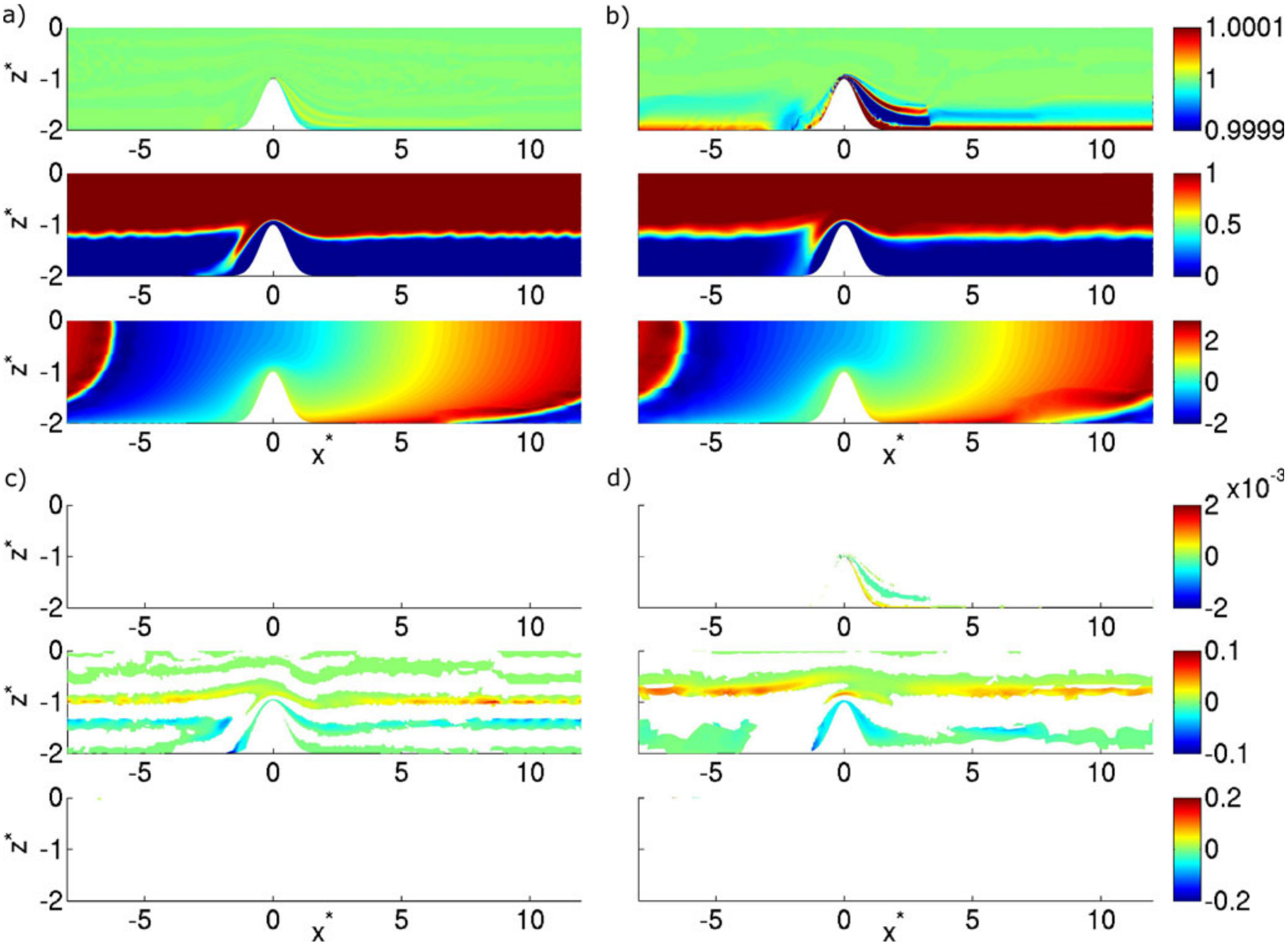

Fig. 21 Solution of three passive tracer cases as Fig. 9, for **a** ($g2$, $p5$) and **b** ($g1$, $p6$) but using our selective exponential filter (37) with $s = 5$. Field values outside initially specified fields as Fig. 10, with **c** ($g2$, $p5$) and **d** ($g1$, $p6$) using the same filter. For each case **a–d**, the filter damps the initialized oscillations of cases 1 and 2 and no oscillations are created in case 3

$$\eta = \frac{i+j}{p+1} \tag{40}$$

$$\alpha = -\log(0.01), \tag{41}$$

with s being the order of the filter and SI the smoothness index.

This results in a filter that is applied only when the smoothness index indicates that the solution is not smooth. The smoothness index relies on information about the decay rate of the modal coefficients, so it cannot be used for lower-order schemes. Using this new filter, we obtain a solution where spurious numerical oscillations are no longer generated by the numerical scheme and the interfaces are not significantly diffused. The new errors plots are shown in Fig. 21. We also verified that this filter does not affect smooth regions of the domain with other idealized test cases (not shown).

While this filter looks promising, additional verification is needed, and a number of improvements are also possible. In particular, determining the strength or order, s, of the filter in an optimal manner is an open question. Also, selectively applying the filter only in directions where the solution is nonsmooth is a topic of future research.

5 Conclusions

We completed a set of computational studies for the modeling of multiscale biogeochemical dynamics in coastal ocean regions with complex bathymetric features, utilizing recent advances in computational fluid dynamics. Specifically, we compared low- to high-order discretization schemes, both in time and space, employing standard and hybrid discontinuous Galerkin finite element methods, on both straight and curved elements. We studied the effects of a varied set of numerical properties including quadrature-free and quadrature-based discretizations of the source terms, order of the spatial discretizations of advection and diffusion operators, order of the temporal discretization in explicit schemes, and resolution of the spatial mesh, with and without our new curved elements. We verified the convergence of our numerical schemes for both the biology and flow fields, validated the codes on analytical solutions, and completed a rigorous truncation error analysis.

Our numerical analyses concentrated on the nonlinear nutrient–phytoplankton–zooplankton dynamics under advection and diffusion within an ocean strait or sill, in an idealized 2D geometry. We first nondimensionalized the PDEs, evaluated stability regions, and selected three biological dynamical regimes: single stable points at all depths, stable limit cycles at the bottom of the euphotic zone, and stable limit cycles within the whole euphotic zone (the latter two cases have limit cycles that are depth and light dependent). We evaluated the effects of numerical parameters on the three biological regimes but illustrated only the most relevant results. In addition, for each of these biological regimes, we examined three types of coupled physics–biology interactions: biological terms dominating, advection terms dominating, and advection and biological terms balancing. For the advection-dominating case, we studied the advection over a strait of a uniform tracer, horizontal front and vertical front. For the balanced situation, relatively common in the real ocean, we considered biological dynamics that were either as fast as (e.g., coastal ocean) or slower than advection time scales.

In the regime where biological terms dominate, we found that both the quadrature-based and quadrature-free treatment of the source terms give accurate, convergent results, although the quadrature-based algorithm had slightly smaller errors. We also showed that oscillations can occur solely due to numerics (Gibbs-like phenomena) for a high-order discretizations. A key result is that, for any numerical scheme, careful 1D studies should be performed to understand the potential errors from the nonlinear source-term discretization.

For the advection-dominating regime, we confirmed the flow field convergence and using passive tracers studied numerical advection artifacts that would also affect the biology. We found that for discretizations that do not resolve the solution, oscillations due to discontinuities in the tracers could be large for both low-order and high-order discretizations but can be damped using a filtering approach for the high-order case. However, our results suggested that when the solution is resolved enough, higher-order schemes on coarser grids perform better (higher accuracy, less dissipative) for the same cost than lower-order scheme on finer grids, as long as curved boundaries were used.

For the case of approximately balanced advection and biological terms, we compared low- and high-order temporal and spatial discretizations and studied quadrature-based and quadrature-free discretizations of the source terms. We found that for lower-order temporal discretizations, the errors grew rapidly and would lead to inaccurate solutions for applications with faster biological timescales or longer integration times. We also showed that the quadrature-based source-term discretization was more accurate in regions where the solution was under-resolved, but in well-resolved regions, there was only a 0.1% discrepancy, and the quadrature-free algorithm could be used for efficiency purposes. By quantitatively evaluating the truncation error and smoothness of the solution fields, we confirmed that higher-order spatial discretizations were more accurate in regions where the solution was smooth (i.e., resolved enough) but less accurate where nonsmooth (un-resolved) due to Gibbs-like oscillations. To reduce these oscillations, we developed a new numerical filter that is active only when and where the solution is not smooth locally, using a smoothness indicator. Finally, we demonstrated the importance of nondissipative numerical schemes when biological patches are present which is common in the real ocean. First, we found that effects of numerical dissipation were amplified by biological activity, causing dissipation errors to increase faster with integration time. Higher-order spatial discretizations were more accurate when modeling biological patches because they maintained the patches while lower-order schemes did not. For resolved biology (e.g., as in Fig. 15), higher-order schemes on coarser grids were for the same cost more accurate than lower-order schemes on finer grids. This conclusion is most important for longer-term simulations. It has major implications for fundamental studies of biological blooms, patchiness, and other nonlinear dynamics in coastal regions with complex bathymetric features such as straits, sills, ridges, and shelfbreaks. One can expect similar implications for longer-term eddy-resolving ecosystem studies or climate applications.

Based on our results, future research directions are to further develop schemes to reduce Gibbs-like oscillations without significant loss of accuracy and efficiency (e.g., Persson and Peraire 2006). Without oscillation limiters or filtering, the optimal performance could be obtained by using different polynomial degree basis functions in the domain, where low-order elements could be used in nonsmooth regions while high-order elements could be used in smooth regions. Because the smoothness can be determined from the discretization, an adaptive grid and polynomial degree scheme could be developed. Another possibility in this case could be to increase the grid resolution and decrease the order of schemes (e.g., medium-order schemes, i.e., ($g3$, $p3$) or ($g3$, $p4$)) up to the point when numerical oscillations reach the size of other errors. Another research direction is to develop and evaluate schemes that would preserve the nonnegativity of the biological solution. Our results can now be utilized for idealized studies of biological dynamics in straits or sills. Uncertainty quantifications (Lermusiaux 2006; Sapsis and Lermusiaux 2009) as well as adaptive model learning (Lermusiaux 2007) for biological predictions would also be useful. Finally, we are now well positioned to implement these new methods in 3D ocean modeling systems (e.g., MSEAS Group 2010) for realistic coupled biogeochemical–physical ocean science and applications.

Acknowledgements We are grateful to Dr. Patrick Haley Jr. and Ms. Lisa Burton for their collaboration on numerics and biology. We thank Dr. N.C. Nguyen for discussions on DG and HDG and Dr. J. Peraire discussions on numerics. We thank the Office of Naval Research for sustained research support under grants N00014-07-1-1061, N00014-08-1-1097, and N00014-07-1-0473 to the Massachusetts Institute of Technology. We are grateful to the Natural Sciences and Engineering Research Council (NSERC) of Canada for the Postgraduate Scholarship to the first author and to Mr. and Mrs. Pappalardo for partially supporting this research through a Pappalardo Fellowhip. We thank Dr. M. Iskandarani and the reviewers for the detailed and helpful comments and suggestions.

References

Agarwal A (2009) Statistical field estimation and scale estimation for complex coastal regions and archipelagos. S.M. thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering

Agarwal A, Lermusiaux PFJ (2010) Statistical field estimation for complex coastal regions and archipelagos. Ocean Model, Sub Judice (in press)

Anderson D, McGillicuddy D, Townsend D, Turner J (2005) The ecology and oceanography of toxic Alexandrium Fundyense blooms in the Gulf of Maine—preface. Deep-sea Res Part 2 Top Stud Oceanogr 52(19–21):2365–2368

Bassi F, Rebay S (1997) A high-order accurate discontinuous Galerkin finite element method for the numerical solution of the compressible Navier–Stokes equations. J Comput Phys 131:267–279

Bernard PE, Remacle JF, Legat V (2009) Boundary discretization for high-order discontinuous Galerkin computations of tidal flows around shallow water islands. Int J Numer Methods Fluids 59(5):535–557

Besiktepe ST, Lermusiaux PFJ, Robinson AR (2002) Coupled physical and biogeochemical data-driven simulations of Massachusetts Bay in late summer: real-time and postcruise data assimilation. J Mar Syst 40:171–212

Bourgault D, Kelley DE (2004) A laterally averaged nonhydrostatic ocean model. J Atmos Ocean Technol 21:1910–1924

Budgell WP, Oliveira A, Skogen MD (2007) Scalar advection schemes for ocean modelling on unstructured triangular grids. Ocean Dyn 57(4–5):339–361

Burchard H, Deleersnijder E, Meister A (2005) Application of modified Patankar schemes to stiff biogeochemical models for the water column. Ocean Dyn 55:326–337

Burton LJ (2009) Modeling coupled physics and biology in ocean straits with application to the San Bernardino strait in the Philippine Archipelago. S.M. thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering

Cebeci T, Shao JP, Kafyeke F, Laurendeau E (2005) Computational fluid dynamics for engineers. Springer, New York

Chapra SC, Canale RP (2006) Numerical methods for engineers, 5th edn. McGraw-Hill Higher Education, Boston

Chung TJ (2002) Computational fluid dynamics. Cambridge University Press, New York

Cockburn B (1998) An introduction to the discontinuous Galerkin method for convection dominated flows. In: Quarteroni A (ed) Advanced numerical approximation of nonlinear hyperbolic equations. Springer, New York

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

Cockburn B, Shu CW (1998a) The local discontinuous Galerkin method for time-dependent convection–diffusion systems. SIAM J Numer Anal 35(6):2440–2463

Cockburn B, Shu CW (1998b) The Runge–Kutta discontinuous Galerkin method for conservation laws V—multidimensional systems. J Comput Phys 141(2):199–224

Cockburn B, Lin SY, Shu CW (1989) TVB Runge–Kutta local projection discontinuous Galerkin finite-element method for conservation-laws. 3. One-dimensional systems. J Comput Phys 84(1):90–113

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

Cockburn B, Luskin M, Shu CW, Suli E (2003) Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. Math Comput 72(242):577–606

Cockburn B, Gopalakrishnan J, Lazarov R (2009) Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. SIAM J Numer Anal 47(2):1319–1365

Cushman-Roisin B (1987) Introduction to geophysical fluid dynamics. Prentice Hall, Upper Saddle River

Dawson C, Sun S, Wheeler MF (2004) Compatible algorithms for coupled flow and transport. Comput Methods Appl Mech Eng 193:2565–2580

Deleersnijder E, Lermusiaux PFJ (2008) Multi-scale modeling: nested grid and unstructured grid approaches. Ocean Dyn 58:335–336

Doney SC, Lima I, Moore JK, Lindsay K, Behrenfeld MJ, Mahowald TKWN, Glovera DM, Takahashi T (2009) Skill metrics for confronting global upper ocean ecosystem-biogeochemistry models against field and remote sensing data. J Mar Syst 76(1–2):95–112

Fasham MJR, Ducklow HW, McKelvie SM (1990) A nitrogen-based model of plankton dynamics in the oceanic mixed layer. J Mar Res 48(49):591–639

Fennel W, Neumann T (2004) Introduction to the modelling of marine ecosystems, Elsevier oceanography series, vol 72. Elsevier, New York

Ferziger JH, Peric M (2002) Computational methods for fluid dynamics, 3rd edn. Springer, New York

Flierl G, McGillicuddy DJ (2002) Mesoscale and submesoscale physical–biological interactions. The Sea 12:1–74

Geuzaine C, Remacle JF (2009) Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Int J Numer Methods Eng 79(11):1309–1331

Haidvogel DB, Beckmann A (1999) Numerical ocean circulation modeling. Imperial College, London. Distributed by World Scientific, River Edge

Haley PJ, Lermusiaux P, Robinson A, Leslie W, Logutov O, Cossarini G, Liang X, Moreno P, Ramp S, Doyle J, Bellingham J, Chavez F, Johnston S (2009) Forecasting and reanalysis in the Monterey Bay/California current region for the autonomous ocean sampling network-II experiment. Special issue on AOSN-II. Deep-sea Res Part II Top Stud Oceanogr 56:127–148

Haley PJ Jr, Lermusiaux PFJ (2010) Multiscale two-way embedding schemes for free-surface primitive-equations in the Multidisciplinary Simulation, Estimation and Assimilation System. Ocean Dyn (in this special issue). doi:10.1007/s10236-010-0349-4

Hanert E, Le Roux DY, Legat V, Deleersnijder E (2004) Advection schemes for unstructured grid ocean modelling. Ocean Model 7(1–2):39–58

Hecht MW, Holland WR, Rasch PJ (1995) Upwind-weighted advection schemes for ocean tracer transport—an evaluation in a passive tracer context. J Geophys Res Oceans 100(C10):20, 763–720, 778

Hesthaven JS, Kirby RM (2008) Filtering in Legendre spectral methods. Math Comput 77(263):1425–1452

Hesthaven JS, Warburton T (2008) Nodal discontinuous Galerkin methods. Texts in applied mathematics, vol 54. Springer, New York

Hofmann EE, Friedrichs MAM (2002) Predictive modeling for marine ecosystems. The Sea 12:537–565

Hofmann EE, Lascara CM (1998) Overview of interdisciplinary modelling for marine ecosystems. In: Brink KH, Robinson AR (eds) The sea, vol 10. Wiley, New York, pp 507–540

Hoteit H, Ackerer P, Mosé R, Erhel J, Philippe B (2004) New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes. Int J Numer Methods Eng 61(14):2566–2593

Iskandarani M, Levin JC, Choi BJ, Haidvogel DB (2005) Comparison of advection schemes for high-order h-p finite element and finite volume methods. Ocean Model 10(1–2):233–252

Ji R, Davis C, Chen C, Beardsley R (2008) Influence of local and external processes on the annual nitrogen cycle and primary productivity on Georges Bank: a 3-D biological–physical modeling study. J Mar Syst 73(1–2):31–47

Karniadakis GE, Sherwin SJ (2005) Spectral/hp element methods for CFD, 2nd edn. Oxford University Press, New York

Krivodonova L (2007) Limiters for high-order discontinuous Galerkin methods. J Comput Phys 226(1):276–296

Kubatko EJ, Bunya S, Dawson C, Westerink JJ, Mirabito C (2009) A performance comparison of continuous and discontinuous finite element shallow water models. J Sci Comput 40:315–339

Lalli C, Parsons T (1997) Biological oceanography: an introduction, 2nd edn. Butterworth-Heinemann, Oxford

Lambrechts J, Remacle1 JF, Hillewaert K (2010) Efficient assembly of high order continuous and discontinuous finite element operators. In: IMUM2010—9th international workshop on multiscale (un)-structured mesh numerical modeling for coastal, shelf, and global ocean dynamics. http://mseas.mit.edu/IMUM2010/

Lermusiaux P (1999) Data assimilation via error subspace statistical estimation, part II: Mid-Atlantic bight shelfbreak front simulations, and ESSE validation. Mon Weather Rev 127(8):1408–1432

Lermusiaux PFJ, Xu J (2010) Coupled ocean-acoustic prediction of transmission loss in a continental shelfbreak region: predictive skill, uncertainty quantification and dynamical sensitivities. IEEE Trans J Ocean Eng. doi:10.1109/JOE.2010.2068611

Lermusiaux PFJ (2006) Uncertainty estimation and prediction for interdisciplinary ocean dynamics. J Comput Phys 176–199 (Special issue on "Uncertainty Quantification." J. Glimm and G. Karniadakis, Eds.)

Lermusiaux PFJ (2007) Adaptive modeling, adaptive data assimilation and adaptive sampling. Physica D 230:172–196 (Special issue on "Mathematical Issues and Challenges in Data Assimilation for Geophysical Systems: Interdisciplinary Perspectives." C.K.R.T. Jones and K. Ide, Eds.)

Lermusiaux PFJ, Haley PJ Jr, Leslie WG, Logutov OG (2009) Philippines strait dynamics experiment—MSEAS home page. http://mseas.mit.edu/Research/Straits/

Lévy M, Estublier A, Madec G (2001) Choice of an advection scheme for biogeochemical models. Geophys Res Lett 28(19):3725–3728

Logutov OG, Lermusiaux PFJ (2008) Inverse barotropic tidal estimation for regional ocean applications. Ocean Model 25:17–34

Lomax H, Pulliam TH, Zingg DW (2003) Fundamentals of computational fluid dynamics (scientific computation). Springer, New York

Lynch DR, McGillicuddy DJ, Werner FE (2009) Skill assessment for coupled biological/physical models of marine systems. J Mar Syst 76(1–2):1–3

Mavriplis CA (1989) Nonconforming discretization and a posteriori error estimators for adaptive spectral element techniques. Ph.D. thesis, MIT

MSEAS-Group (2010) Multidisciplinary simulation, estimation, and assimilation systems (http://mseas.mit.edu/, http://mseas.mit.edu/codes). Reports in ocean science and engineering 6, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts

Nguyen NC, Peraire J, Cockburn B (2009) An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. J Comput Phys 228(9):3232–3254

Oschlies A, Garcon V (1998) Eddy-induced enhancement of primary production in a model of the North Atlantic Ocean. Nature 394:266–269

Pain CC, Piggott MD, Goddard AJH, Fang F, Gorman GJ, Marshall DP, Eaton MD, Power PW, de Oliveira CRE (2005) Three-dimensional unstructured mesh ocean modelling. Ocean Model 10(1–2):5–33

Pedlosky J (1987) Geophysical fluid dynamics, 2nd edn. Springer, New York

Persson PO, Peraire J (2006) Sub-cell shock capturing for discontinuous Galerkin methods. AIAA-2006-112

Persson PO, Strang G (2004) A simple mesh generator in MATLAB. SIAM Rev 46(2):329–345

Qiu J, Shu CW (2005) Runge–Kutta discontinuous Galerkin method using WENO limiters. SIAM J Sci Comput 26(3):907–929

Reed WH, Hill TR (1973) Triangular mesh methods for the neutron transport equations. Los Alamos Scientific Laboratory Report LA-UR-73-479, Fanstord University

Robinson AR, Lermusiaux PFJ (1999) Report of a workshop on the assimilation of biological data in coupled physical/ecosystem models. GLOBEC Special Contribution 3, Bologna, Italy

Robinson AR, McCarthy JJ, Rothschild BJ (2002) Biological–physical interactions in the sea. The Sea, vol 12. Wiley, New York

Rothstein LM, Cullen JJ, Abbott M, Chassignet EP, Denman K, Doney SC, Ducklow H, Fennel K, Follows M, Haidvogel D, Hoffman E, Karl DM, Kindle J, Lima I, Maltrud M, McClain C, McGillicuddy DJ, Olascoaga MJ, Spitz Y, Wiggert J, Yoder J (2006) Modeling ocean ecosystems: the paradigm program. Oceanography 19(1):22–51

Sapsis T, Lermusiaux P (2009) Dynamically orthogonal field equations for continuous stochastic dynamical systems. Physica D 238:2347–2360

Signell R (1989) Tidal dynamics and dispersion around coastal headlands. Ph.D. thesis, WHOI/MIT Joint Program

Slingo J, Bates K, Nikiforakis N, Piggott M, Roberts M, Shaffrey L, Stevens I, Vidale PL, Weller H (2009) Developing the next-generation climate system models: challenges and achievements. Philos Trans Royal Soc Math Phys Eng Sci 367(1890):815–831

Solin P, Segeth K, Dolezel I (2003) Higher-order finite element methods. Chapman & Hall/CRC, Boca Raton

Spitz Y, Allen J, Gan J (2005) Modeling of ecosystem processes on the Oregon shelf during the 2001 summer upwelling. J Geophys Res 110:C10S17

Stow CA, Jolliff J, McGillicuddy DJ, Doney SC, Allen JI, Friedrichs MAM, Rose KA, Wallhead P (2009) Skill assessment for coupled biological/physical models of marine systems. J Mar Syst 76(1–2):4–15

Strang G, Fix GJ (1973) An analysis of the finite element method. Prentice Hall, Englewood Cliffs

Ueckermann MP (2009) Towards next generation ocean models: novel discontinuous Galerkin schemes for 2D unsteady biogeochemical models. SM thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering