Locally Filtered Transport for computational efficiency in multi-component advection-reaction models

Hennes Hajduk a, Ben R. Hodges b,*, Vadym Aizinger c, d, Balthasar Reuter d

a Technische Universität Dortmund, Institute of Applied Mathematics (LS III), Vogelpothsweg 87, 44227 Dortmund, Germany
b University of Texas at Austin, Center for Water and the Environment, 10100 Burnet Rd, MC R8000, Austin, TX 78758, United States
c Alfred Wegener Institute, Helmholtz Centre for Polar and Marine Research, Am Handelshafen 12, 27570 Bremerhaven, Germany
d Friedrich-Alexander University Erlangen–Nürnberg, Chair of Applied Mathematics I, Cauerstraße 11, 91058 Erlangen, Germany

A R T I C L E   I N F O

Article history:
Received 20 April 2017
Received in revised form 31 October 2017
Accepted 5 January 2018

Keywords:
Discontinuous Galerkin finite element method
Shallow water equations
Multi-component advection-reaction system
NPZ model
Transport scheme

A B S T R A C T

This work introduces the Locally Filtered Transport (LFT) method for numerical transport models. Locally turning off the transport computation in areas of nearly uniform concentration is proposed as a new approach for reducing computational cost in ecosystem models that require transport of tens to hundreds of constituent concentrations. The proposed method is locally mass conservative just as the discontinuous Galerkin finite element scheme it is based on. The performance of the method is illustrated using numerical examples including an advection-reaction ecosystem simulation with a simple nitrogen, phytoplankton, and zooplankton (NPZ) model.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

This work introduces and evaluates a new computationally-efficient scheme for transport equations in multi-component advection-diffusion-reaction models. The idea behind our method is quite simple: many physical, chemical, and biological processes take place on highly-localized spatial and temporal scales such that one or more transported constituents might be at quasi-uniform or “background” concentrations over large areas. For example, outside of a localized algae bloom, the chlorophyll concentration in a water quality model is typically at some background level, such that the same small concentration is fluxed in and out of most computational cells. These computational cycles of the transport equations are wasted and cannot affect the model results until the reaction equations initiate local growth of a bloom. Thus, an ecosystem model coupled to a large-scale circulation model (e.g., regional or global ocean, climate) incurs substantial computational costs for transport in parts of the spatio-temporal domain where some or all of the constituents are not present, have only the background concentration, or do not play a significant role in the reactions. For simple ecosystem models implemented in only two dimensions (2D), such as the Nitrogen-Phytoplankton-Zooplankton (NPZ) model used herein for demonstration purposes, the increased computational costs of transporting a few scalars is generally irrelevant. However, for ecosystem models that transport different species of plankton and include chemical speciation (NO3, NH4, dissolved inorganic carbon, dissolved organic carbon, etc.) the number of transported variables can easily be several dozen or more [e.g.,Robson and Hamilton, 2004, Schwalb et al., 2015]. In three dimensions (3D) the extensive scalar transport requirements can dominate the overall computational time. Arguably, such models are computationally inefficient as they are not generally designed to identify and transport constituents only when and where they are significant. In this study, we demonstrate how to add such capability to an existing hydrodynamic/transport model with an approach we call Locally Filtered Transport (LFT).

We propose the new LFT algorithm that adaptively turns on/off the computation of certain discrete terms. The model performance
is evaluated using a conventional (hydrostatic, inviscid) 2D shallow water and transport model based on the discontinuous Galerkin (DG) finite element method (Aizinger, 2004; Aizinger and Dawson, 2002). The present work builds on the background filtering approach (Hodges, 2014), which required an unconventional mass transport algorithm. Herein, we show that localization techniques can be efficiently extended to standard concentration transport schemes. Furthermore, the computational costs associated with localization that were identified in (Hodges, 2014) are elegantly handled within the DG framework using the vertex-based slope limiter (Aizinger, 2011; Kuzmin, 2010). The utility of the proposed approach is not limited to discontinuous Galerkin methods or geophysical applications: it can be easily transferred to any numerical PDE (partial differential equation) solver containing transport equations and might achieve meaningful performance gains even in the absence of reaction terms or in situations when only a few species are transported.

This paper is structured as follows. The system of governing equations is introduced in Sec. 2 followed by a description of the LFT method in Sec. 3. An NPZ ecosystem model is presented in Sec. 4, which is used as a test case for the LFT method combined with a DG hydrodynamic/transport model in Sec. 5. For completeness and to allow others to build on the modeling approach, the details of DG discretization are provided in Appendix A. A brief discussion and conclusions section completes the paper.

2. Governing equations

The model problem for this study is the 2D shallow water equations in conservative form, eqs. (1) and (2) below, combined with a varying number of equations for advection-reaction, represented by eq. (3) below, and augmented – as needed – by the corresponding initial and boundary conditions.

\[ \partial_t \xi + \nabla \cdot (u H) = 0, \]  
\[ \partial_t (u H) + \nabla \cdot (u \otimes u H) + g H \nabla \xi + f_c \times u H + \tau_{df} u H = H F, \]  
\[ \partial_t (c_m H) + \nabla \cdot (u H c_m) = H R_{m(1, \ldots, M)} + H F_n, \quad m = 1, \ldots, M. \]

The primary unknowns in eqs. (1)–(3) are the water surface elevation (\( \xi \)) measured from a uniform datum, the depth-integrated horizontal velocity vector \( u H = [U, V]^T \), and the depth-integrated concentrations of multiple transported species \( c_m H, \quad m = 1, \ldots, M \). Given a boundary condition of spatially-varying bathymetry elevation, \( b(x, y) \), the auxiliary variable \( H \) denotes the total water depth \( \xi - b \). All equations are required to hold on some Lipschitz bounded 2D domain \( \Omega \) and on time interval \( [0, T_{\text{end}}] \). Furthermore, \( g \) denotes gravity, \( f_c \) is the Coriolis coefficient, \( k \) is the vertical unit vector pointing upwards, \( \tau_{df} \) is the coefficient of the quadratic friction law, \( F = (F_x, F_y) \) lumps together the forcing terms in the momentum equation (e.g., tidal potential), and \( R_{m, n}, \quad m = 1, \ldots, M \) are the source/sink terms and reaction rates in advection-reaction equations, respectively. With the exception of the reaction terms, the model is very similar to (Aizinger and Dawson, 2002), where the DG method was proposed for the 2D shallow water equations combined with non-reaction species transport.

Equations (1) and (2) utilize three types of boundary conditions (land, river, open sea) denoted by \( \Gamma_L, \Gamma_R, \Gamma_O \), respectively, while eq. (3) for constituent transport may have boundaries that are inflow, outflow, or wall (no-flow). In this work, a river boundary is always an inflow boundary, a land boundary is always a wall boundary, and open sea boundaries are dynamically switched between the in-/out-flow modes depending on the flow direction. These conditions are presented formally in Table 1.

3. Locally Filtered Transport

The LFT method relies on definition of an individualized active domain for each transported constituent that is a subset of the total domain \( \Omega \). Each active domain evolves over time as advection-diffusion and reactions change the constituent concentration. Outside of the active domain the advection terms are ignored.

We consider two cases: first, as discussed in the introduction, there are regions of a computational domain over which some constituent might fall below a dynamically-meaningful concentration and may safely be ignored. Second, there is the possibility of regions with nearly uniform concentrations where transport is merely moving the same concentration about with no effect on the local distribution. For example, far away from the influence of estuaries, a large-scale ocean model that is not resolving salinity effects for meso-scale features might be simply shuffling around minuscule changes in salinity over the majority of the domain. A modeler might want to retain salinity transport for estuarine input to the coastal shelf, but it could be safely excised from the majority of the domain. Both cases can be addressed by monitoring the local concentration difference. Herein we define a uniformity difference, or \( \delta_u \) for each constituent such that concentration differences between two neighboring grid elements smaller than \( \delta_u \) allow the concentration to be considered locally uniform so that no transport computation is needed.

This general idea was introduced in (Hodges, 2014) as part of an algorithm using mass transport (rather than concentration transport) to allow local subtime stepping in regions where high velocities strictly limit the local advective time step. The prior methods had relatively high computational costs due to the approach taken to identify the active domain. In the present work, we adapt the background filtering from (Hodges, 2014) to the DG algorithm using uniform time steps and a conventional concentration transport discretization. Although the LFT idea can be extended to any model, the slope-limiting DG method (Aizinger, 2011; Kuzmin, 2010) has a particular advantage in that the majority of the computational effort for localization is already required in the existing transport algorithm, namely by the slope-limiting function (see Appendix A.5), which identifies and sorts the local concentration differences to maintain monotonicity.

The discrete time advance of eq. (3) for constituent transport is described in detail in Appendix A.5. For purposes of the LFT method, the key point is that the advance from time level \( t^n \) to \( t^{n+1} \) uses an explicit Runge-Kutta scheme with a slope limiter. At time \( t^n \), the minimum and maximum values of a transported constituent for each computational element \( T_k \) that is connected to a node \( (x_i) \) are known from the slope limiter of the time advance from the \( n - 1 \) to \( n \) step, i.e. \( q_{\text{min}}^{(n)} \) and \( q_{\text{max}}^{(n)} \) are already defined at each vertex, see eq. (A.9). We use this time-lagged data to find the maximum nodal jump of scalar field \( a \) for element \( T_k \) and its nodal neighbors as

<table>
<thead>
<tr>
<th>BC type</th>
<th>Shallow water</th>
<th>Transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>land ( \Gamma_L )</td>
<td>( u \cdot n = 0 )</td>
<td>( u \cdot n c_m = 0 )</td>
</tr>
<tr>
<td>river ( \Gamma_R )</td>
<td>( H - H_0, \quad u - u_0 )</td>
<td>( c_m - (c_m)_0 )</td>
</tr>
<tr>
<td>open sea ( \Gamma_O )</td>
<td>( H - H_0 )</td>
<td>( c_m - (c_m)_0 ) if ( u \cdot n &lt; 0 )</td>
</tr>
</tbody>
</table>
$I_t := \max \left( a_i^{\text{max}} - a_i^{\text{min}} \right)$. $x_i$ is vertex of $T_k$.

It then follows that the only elements $T_k$ included in the Runge-Kutta time advance for a given constituent are those where $\Xi_k > \delta_u$.

All elements where the $\delta_u$ condition is not met will simply proceed with $c_k^{n+1} = c_k^n$. Note that because we are using an explicit time advance with a CFL < 1 limitation, it is guaranteed that the concentration error associated with neglecting transport in any element is ±$\delta_u$. That is, if we compute the transport between all elements, any element whose neighbors are within ±$\delta_u$ cannot change by more than that value in a conservative, slope-limited transport scheme.

Some observations on our implementation of the LFT scheme:

- Each species has its own active domain, however the algorithm could be modified to treat combinations of species (e.g. by considering sums of concentrations).
- Setting the $\delta_u = 0$ yields the same results as the standard transport scheme — which means our method is fully consistent.

### Table 2: NPZ-model parameters chosen as in [Wroblewski et al., 1988.]

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_0$</td>
<td>half saturation constant for PAR</td>
<td>92 W/m$^2$</td>
</tr>
<tr>
<td>$l$</td>
<td>incident PAR</td>
<td>110 W/m$^2$</td>
</tr>
<tr>
<td>$k_{ex}$</td>
<td>light extinction coefficient</td>
<td>0.1 m$^{-1}$</td>
</tr>
<tr>
<td>$V_{10}$</td>
<td>maximum phytoplankton growth rate</td>
<td>2.0 per day</td>
</tr>
<tr>
<td>$k_4$</td>
<td>nutrient uptake half saturation</td>
<td>0.2 $\mu g$ atom NO$_3$/l</td>
</tr>
<tr>
<td>$R_{eo}$</td>
<td>maximum herbivore ingestion rate</td>
<td>0.5 per day</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Ivlev’s constant for herbivore grazing</td>
<td>0.5 $\mu g$ atom NO$_3$/l</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>phytoplankton death rate</td>
<td>0.1 per day</td>
</tr>
<tr>
<td>$\delta$</td>
<td>zooplankton death rate</td>
<td>0.2 per day</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>zooplankton assimilation</td>
<td>0.3</td>
</tr>
</tbody>
</table>

---

**Fig. 1.** Simple motion problem with one plume: Initial state with increasing plume areas (upper), final solution after one orbital transit of domain (lower). The initial area of the plume doubles in each plot from left to right. The active domain for LFT with $\delta_u(c) = 10^{-6}$ is indicated by mesh.

**Fig. 2.** Simple motion problem with four circles: Initial state with increasing plume areas (upper), final solution after one orbital transit of domain (lower). The initial area of each plume doubles in each plot from left to right. The active domain for LFT with $\delta_u(c) = 10^{-6}$ is indicated by mesh.
which utilizes MATLAB. The runs were executed using MATLAB version R2017a.

Simple motion problem: Radii of the circles and mean percentages of elements in the active domain for the LFT algorithm.

<table>
<thead>
<tr>
<th># of circles</th>
<th>Test case</th>
<th>Radius</th>
<th>$\delta_u = 10^{-6}$: % Active</th>
<th>$\delta_u = 10^{-8}$: % Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>r1_1</td>
<td>13.3</td>
<td>4.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>r2_1</td>
<td>15.2</td>
<td>26.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>r3_1</td>
<td>17.7</td>
<td>29.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>r4_1</td>
<td>20.7</td>
<td>32.1</td>
<td></td>
</tr>
</tbody>
</table>

| 4            | r1_4      | 50.8   | 74.1                          |                               |
|              | r2_4      | 56.1   | 77.5                          |                               |
|              | r3_4      | 62.4   | 81.1                          |                               |
|              | r4_4      | 69.1   | 84.7                          |                               |

Fig. 3. Performance of LFT for the simple motion problem. Relative size of the active domain with different $\delta_u$. See Table 3 for the test case abbreviations in the legend.

- Implemented as shown in the Appendix, our scheme guarantees full local conservation of mass for each constituent. This can be seen using a simple observation: since our DG discretization utilizes the conservative (or divergence) form of the 2D shallow-water and transport equations, the computation of edge fluxes is the only potential source of mass error. The uniqueness of the solution to the Riemann problem (see eq. (A.5)) automatically provides the conservation for the exterior boundary edges and those interior edges that lie between two masked (active) elements. No flux computation takes place on interior edges shared by two non-mask elements. This leaves fluxes on interior edges between masked and non-masked elements as the only potential source of error. If the flux contributions from such edges are accounted for on both adjoining elements – as is the case in our implementation – the scheme is fully mass conservative; otherwise, the missing flux terms result in a mass error. This accounting for can be carried out in two different ways, both of them fully mass conserving: First, setting the fluxes to zero which is equivalent to turning off the contribution from one edge on masked elements; second, treating these fluxes in the same way as fluxes between two active elements – this would be equivalent to updating the affected non-masked elements using a single edge contribution. For performance reasons, our implementation employs the first approach.

- We use computed concentrations ($c_m$) rather than the evolved depth-integrated concentrations ($c_m(H)$) of eq. (3). If the latter were used the changes in the water depth $H$ could produce large gradients in areas of constant local concentration and vice versa.

- The ideas are readily generalized to diffusion and reaction operators, but for simplicity we confine ourselves to advection terms in this study.

- Another important generalization of the presented methodology can be made in the context of 3D modeling. Since the 3D circulation and transport is often marked by a clear anisotropy between horizontal and vertical directions, the LFT framework would greatly benefit from introducing separate masks for vertical and horizontal transport (both of them fully three-dimensional, i.e., working on the 3D mesh). This can be accomplished similarly to introducing separate masks for different species as illustrated in Sec. 5.2.

4. NPZ model

To demonstrate the effectiveness of the LFT method we solve the shallow water equations coupled to an NPZ ecosystem model of nitrogen, phytoplankton, and zooplankton dynamics (Franks, 2002; Wroblewski et al., 1988). The system is governed by eqs. (1)–(3) with $M = 3$ and reaction equations

\[ R_1 = (k(l,n) - i(p)) p - h(p) z, \]
\[ R_2 = ((1 - \gamma) h(p) - f(z)) z, \]
\[ R_3 = (\gamma h(p) + f(z)) z - (k(l,n) - i(p)) p, \]

Table 4

Simple motion problem: Runtimes (transport model without slope limiting and total) and accuracy. All presented runs were performed on a machine with an Intel Core i7-6700 processor (with 4 cores, 8 threads), and 16 GBytes of RAM. The code is implemented as part of the software package PESTUNG (see Frank et al., 2015a, 2015b; Jaust et al., 2017; Reuter et al., 2016), which utilizes MATLAB. The runs were executed using MATLAB version R2017a.

<table>
<thead>
<tr>
<th>Test case</th>
<th>No LFT</th>
<th>Transport time, sec.</th>
<th>Total time, sec.</th>
<th>$L^1$-error $t-2\pi$</th>
<th>Transport time, sec.</th>
<th>Total time, sec.</th>
<th>$L^1$-error $t-2\pi$</th>
<th>Transport time, sec.</th>
<th>Total time, sec.</th>
<th>$L^1$-error $t-2\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1_1</td>
<td>225</td>
<td>794</td>
<td>7.05e-3</td>
<td>80</td>
<td>650</td>
<td>7.00e-3</td>
<td>100</td>
<td>684</td>
<td>6.99e-3</td>
<td></td>
</tr>
<tr>
<td>r2_1</td>
<td>225</td>
<td>794</td>
<td>8.32e-3</td>
<td>83</td>
<td>650</td>
<td>8.31e-3</td>
<td>103</td>
<td>690</td>
<td>8.31e-3</td>
<td></td>
</tr>
<tr>
<td>r3_1</td>
<td>225</td>
<td>794</td>
<td>9.82e-3</td>
<td>88</td>
<td>654</td>
<td>9.96e-3</td>
<td>106</td>
<td>690</td>
<td>9.96e-3</td>
<td></td>
</tr>
<tr>
<td>r4_1</td>
<td>225</td>
<td>794</td>
<td>1.19e-2</td>
<td>92</td>
<td>659</td>
<td>1.23e-2</td>
<td>112</td>
<td>696</td>
<td>1.23e-2</td>
<td></td>
</tr>
<tr>
<td>r1_4</td>
<td>225</td>
<td>794</td>
<td>2.82e-2</td>
<td>145</td>
<td>713</td>
<td>2.81e-2</td>
<td>186</td>
<td>770</td>
<td>2.81e-2</td>
<td></td>
</tr>
<tr>
<td>r2_4</td>
<td>225</td>
<td>794</td>
<td>3.33e-2</td>
<td>154</td>
<td>722</td>
<td>3.33e-2</td>
<td>192</td>
<td>777</td>
<td>3.33e-2</td>
<td></td>
</tr>
<tr>
<td>r3_4</td>
<td>225</td>
<td>794</td>
<td>3.94e-2</td>
<td>166</td>
<td>734</td>
<td>3.99e-2</td>
<td>198</td>
<td>783</td>
<td>3.99e-2</td>
<td></td>
</tr>
<tr>
<td>r4_4</td>
<td>225</td>
<td>794</td>
<td>4.80e-2</td>
<td>178</td>
<td>745</td>
<td>4.91e-2</td>
<td>204</td>
<td>789</td>
<td>4.91e-2</td>
<td></td>
</tr>
</tbody>
</table>
where \( n, p, z \) are concentrations of nitrogen, phytoplankton, and zooplankton, respectively. Our NPZ model is similar to (Wroblewski et al., 1988) with some minor modifications, as discussed below. A summary of all model coefficients is provided in Table 2. Note that in some sophisticated models these coefficients can be functions of the ecosystem state, but herein are taken as constants for simplicity. The following provides a brief overview of the model equations. The reader is referred to (Franks, 2002; Wroblewski et al., 1988) for more in-depth discussions.

The phytoplankton response \( k(l, n) \) is the product of the responses to light \( l \) and nutrients \( n \), in this case

\[
k(l, n) = f(l) g_{MM}(n),
\]

where \( f(l) \) is the depth averaged phytoplankton response to irradiance, and \( g_{MM} \) is the Michaelis-Menten uptake. Herein we use the saturating response model for light (c.f. (Franks, 2002; Rinke et al., 2010))

\[
f(l) = \frac{l \exp(ke_x(\xi - \zeta))}{l_0 + l \exp(ke_x(\xi - \zeta))}
\]

where \( l_0 \) is the half saturation constant for photosynthetically active radiation (PAR), \( l \) is the incident PAR, and \( ke_x \) is the light extinction coefficient. Here, \( \zeta \) is the vertical coordinate with respect to datum. As is common, \( g_{MM}(n) \) is specified as

\[
g_{MM}(n) = \frac{V_m n}{k_5 + n}
\]

where \( k_5 \) is the nutrient uptake half saturation constant, and \( V_m \) is the maximum phytoplankton growth rate.

The grazing of zooplankton on phytoplankton, \( h(p) \), is given by

\[
h(p) = R_m(1 - \exp(-\lambda p))
\]

where \( R_m \) is the maximum herbivore ingestion rate, and \( \lambda \) is Ivlev’s constant for herbivore grazing.

Finally, the death rates for both plankton types are simple constants

\[
i(p) = \epsilon, \quad j(z) = \delta.
\]

In general, local phytoplankton growth depends on the local light intensity, which depends on the water depth. However, our 2D hydrodynamic model uses depth-averaged concentrations so, for consistency, we use a depth-averaged response to irradiance \( \bar{f}(l) \). For further simplification, we neglect the relatively minor motions of the free surface and use the water depth at rest for

![Fig. 4](image-url)  
*Fig. 4.* Total mass error in the \( L^1 \)-norm for the simple motion problem: standard transport (top), LFT with \( \delta_u = 10^{-8} \) (bottom). See Table 3 for the test case abbreviations in the legend.

![Fig. 5](image-url)  
*Fig. 5.* Galveston Bay model domain with computational mesh (upper); surface elevation (middle), and velocity magnitude (lower) after 10 days of spin up. All lengths are in meters, all velocities are in m/s.
the computation of \( f(I) \) similarly to (Wroblewski et al., 1988). We also do not account for seasonal and diel variations in light intensity. Although our approach is clearly too simplistic for an accurate ecosystem model, it will suffice for a demonstration of how the LFT method applies to an advection-diffusion-reaction problem.

We use Galveston Bay (Texas, USA) as a test case in Sec. 5.2 and computed the value of \( I \) in Table 2 using the NOAA long-term mean radiation value for this area during April, May, and June, which has been multiplied by 0.4 to obtain the PAR fraction of shortwave radiation (Six and Maier-Reimer, 1996). To compute the depth-averaged phytoplankton response to light, we integrate eq. (4) over the water column obtaining

\[
\bar{f}(I) = \frac{1}{b} \int_{b}^{0} f(I) \, d\zeta = -\frac{1}{k_{ex} b} \log \left( \frac{l_0 + I}{l_0 + I \exp(k_{ex} b)} \right).
\]

5. Numerical results

5.1. Simple plume motion

To demonstrate that localized transport only within the active domain does not affect the computed concentration fields, we use a

---

**Fig. 6.** Galveston Bay initial conditions for phytoplankton (upper) and zooplankton (lower). Concentration in kg/m³. The initial active domain is indicated by mesh.
standard orbital transport of a concentration without reaction terms. In this problem, a concentration plume is transported in a circular orbit around the domain center by a fixed (in time) velocity field. The domain $\Omega$ is a unit square uniformly partitioned into 8192 triangular elements and the initial plumes are circular. The forcing function $F$ on right-hand side of eq. (2) is chosen to produce a stationary velocity field given by $u(x,y) = (0.5 - y, x - 0.5)^T$ over the time interval $(0, 2\pi)$ corresponding to a full circle rotation of the initial scene.

Two different configurations are considered with the initial concentration consisting of one (Fig. 1) and four (Fig. 2) individual plumes, respectively. In addition, each configuration is further subdivided into four test cases, such that the area of the plume is doubled in each succeeding case. The initial concentration is equal to one inside the plume and zero outside of it. We compare results to control cases without the LFT algorithm and examine effects of using $\delta_u(c)$ of $10^{-6}$ and $10^{-8}$ as the discriminator for the active domain. Table 3 details the radii of the plumes and lists the average percentages of elements in the active domain in relation to the total number of elements. Note that in Figs. 1 and 2 the upper panel initial conditions show the active domain (computational mesh) is only around the edges of the plume and not in the plume center or the empty domain, which have uniform concentrations (relative to $\delta_u$). As the plume undergoes minor numerical diffusion during its orbit, the computational mesh evolves (lower panels) to include the entire plume as well as a diffusion area surrounding it where concentration gradients develop (again, relative to $\delta_u$).

Fig. 3 shows how the fraction of the domain in active computation evolves over time (as the plume diffuses during its orbit). Since the analytical solution at final time $t = 2\pi$ is equal to the initial condition, the discretization error can be computed as the difference between the final solution and the initial condition. The $L^1$-errors for all tests are listed in Table 4; they appear to be dominated by the numerical diffusion of the transport scheme and display virtually no sensitivity to the LFT approximation. Since the actual speed-up achieved by the LFT scheme is strongly dependent
on the programming language, the discrete transport scheme implementation, and data structures in the code, the runtimes displayed in Table 4 should be only used as an indicator supplementing a more rigorous metric given in Table 3 and Fig. 3 by the fraction of the domain that is active. These runtimes were obtained by averaging over three separate runs and can be considered reasonably reliable since runtime variations between runs were well below 1%. Columns titled ‘Transport’ in Table 4 detail the time spent in the transport scheme exempt from the slope limiter (that took ca. 200 s. in all runs), whereas columns under heading ‘Total’ list wall times of the entire simulation. For this test case, we observe between 10% and 55% of speed-up in the transport part depending on the size of the active domain.

To illustrate our claim that the LFT scheme is fully mass conservative we plot the development of the mass error over time in Fig. 4. The LFT mass errors (bottom) turn out to be in the round-off range just as in the transport scheme without LFT (top); the results for $\delta_u = 10^{-6}$ turned out to be nearly identical to those for $\delta_u = 10^{-8}$ and are not shown here.

5.2. Reactive transport in Galveston Bay

Galveston Bay (Texas, USA) was chosen as the setting for demonstrating the performance of the LFT method with the NPZ model of Sec. 4. This bay is challenging for hydrodynamic and transport modeling due to its complex geometry, which includes 17 islands and the narrow Houston Ship channel. This channel has steep sides and is three times deeper than typical depths throughout the bay. All simulations were performed on an unstructured triangular mesh consisting of 3397 elements as illustrated in Fig. 5. To provide meaningful initial conditions for the velocity and water elevation, a cold-start simulation (zero initial velocities and initially-flat water surface) was run for 10 days as a model spin-up period. Test simulations were all started from this 10-day data and continued for 90 days with a constant time step of 5 s.

![Fig. 8. Snapshots of modeled Galveston Bay modeled phytoplankton behavior for 15, 30, 45, 60, 75, and 90 days of simulation (left to right and top to bottom). Upper panels are phytoplankton concentration in kg/m$^3$ (exponential color scale) and active domain for phytoplankton transport shown with mesh. Lower panels are magnitude of the concentration difference between LFT and control simulations (linear color scale). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)](image)
The only river inflow we included is the San Jacinto River, which is modeled for simplicity as a constant inflow with a flux corresponding to velocity of 0.5 m s\(^{-1}\) at the northernmost river element. The southernmost boundary is considered open sea, with tidal elevations imposed for 5 tidal components (O1, K1, N2, M2, S2). The remaining boundary edges are land (no flux) boundaries. Wind forcing is neglected in all the model runs. The initial conditions for the transport model are constant 10\(^{-6}\) kg/m\(^3\) phyto/zooplankton concentrations on a few elements in the bay far away from the open sea boundary and not intersecting each other (see Fig. 6), with zero concentrations in the remainder of the domain. The initial values were chosen with the intent to prevent the maximum concentrations of both planktons exceeding 5 \times 10\(^{-6}\) kg/m\(^3\) at any time — as suggested in (Carlson, 1977). The initial nitrogen concentrations are zero everywhere with a constant nitrogen concentration of 4 \times 10\(^{-4}\) kg/m\(^3\) (as suggested in (Gruber, 2008)) as a boundary condition for the San Jacinto River inflow. Note that in this idealized case, the use of zero values as background concentrations means that the model is only representing the plankton growth/death involving the intersection of the initial patches and the river nitrogen flux. Thus, the model is not illustrative of expected real ecosystem processes across the bay. Simulations were conducted with the new LFT transport algorithm and a conventional (non-LFT) transport algorithm as a control. For the LFT simulations the \(\delta_u\) to discriminate between active and inactive regions were selected as \(\delta_u(n) = 10^{-10} \text{ kg/m}^3\), \(\delta_u(p) = 10^{-13} \text{ kg/m}^3\), and \(\delta_u(z) = 10^{-13} \text{ kg/m}^3\). The \(\delta_u\) used herein were six and seven orders of magnitude below the boundary and initial concentrations for nitrogen and plankton, respectively; i.e. close to numerical zero for the concentration magnitudes involved. The appropriate setting of \(\delta_u\) for each constituent depends on the expected accuracy and uncertainty allowable for the scalar transport, and cannot be \textit{a priori} defined from the LFT method itself. For simple passive scalars the allowable \(\delta_u\) will be associated with concentrations and concentration gradients that are deemed important by the modeler. For ecosystem models, the nonlinear interaction algorithms between constituents will need to be considered. It seems likely that \(\delta_u\) might be substantially higher than applied herein given the typical uncertainty in the coefficients of the reaction equations in ecosystem models. For example, arguably the practical \(\delta_u(p)\) can be set simply to ensure the numerical death or growth of phytoplankton attributable to error in the LFT field is one order of magnitude smaller than the uncertainty in the death/growth rates associated with uncertainties in the zooplankton grazing rate, phytoplankton death rate, and nutrient uptake coefficients. Note that increasing \(\delta_u\) serves to decrease the number of elements in the computational domain (increasing computational efficiency), but also increases the error in the concentration field. Thus, a particular application might obtain dramatic increases in computational efficiency where larger errors are acceptable. Complex ecosystem models can probably be exercised for simple test cases to determine effective settings for \(\delta_u\) values.

The upper panels of Figs. 7 and 8, as well as Fig. 9, illustrate concentrations (kg/m\(^3\)) of each species in the LFT model and the active domain after 15, 30, 45, 60, 75 and 90 days. Exponential color scales are used where appropriate. The lower panels of Figs. 7 and 8 show the magnitude of the difference between the LFT and conventional transport simulations for nitrogen and phytoplankton (the difference plot of zooplankton is qualitatively similar to that of phytoplankton and is omitted). The differences between the LFT method and the control are several orders of magnitude lower than the corresponding concentrations themselves. Thus, with appropriate selection of \(\delta_u\) the results from the LFT method are sufficiently similar to those of the conventional transport method for typical ecosystem simulations.

The performance of the LFT algorithm is quantified in Fig. 10, where the percentage of elements in the active domain and the \(L^\infty\) error for each species are plotted as functions of time. Once again, even in this particularly sensitive norm the results do not differ much from those of the control simulations using the conventional transport method. As might be expected, a constituent that is continuously and widely distributed, such as nitrogen after day 20, provides little opportunity for improving computational efficiency. However, it can be seen that the phytoplankton and zooplankton behaviors can allow dramatic improvements. Indeed, beyond day 50 (for phytoplankton) and day 60 (for zooplankton) the LFT transport computations required for these constituents are near

![Fig. 9. Snapshots of modeled Galveston Bay modeled zooplankton behavior for 15, 30, 45, 60, 75, and 90 days of simulation (left to right and top to bottom). Zooplankton concentration in kg/m\(^3\) (exponential color scale) and active domain for zooplankton transport shown with mesh. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)](image-url)
zero, as indicated by the disappearing mesh in the upper frames of Figs. 8 and 9. It follows that effective implementation of the LFT should consider the likely distribution of a given constituent and values for \( \delta_u \) that reflect the importance of gradients across the ecosystem. The small oscillations in Fig. 10 do not reflect incipient instability, but instead are the natural consequence of the small differences that occur as new cells are activated/deactivated in the computational mask.

To illustrate the effect of the LFT method on the computational efficiency in the setting of the reactive transport model, we list runtimes of main algorithm parts in Table 5 as well as the mean percentages of elements in the active domain for each species. The achieved speed-up of ca. 36% in the transport part of the model or ca. 20% of the total computation time is substantial for such a small number of species; however, the actual size of the speed-up (if any) will vary strongly with the implementation, programming framework, etc.

6. Discussion & conclusions

A new Locally Filtered Transport method is proposed and implemented in the transport routines of a Discontinuous Galerkin hydrodynamic model coupled to an NPZ ecosystem model. The LFT method limits transport to active regions, which are defined based on concentration differences between local computational elements. If, for a given constituent, the local concentration differences between element values and those of its neighbors are smaller than a pre-defined tolerance (\( \delta_u \)), the advection on the element is not computed for that constituent, i.e. advection in/out is considered sufficiently balanced and cannot change the local concentration. This approach requires time-space varying of the active transport regions (which are different for different constituents), which does not introduce significant errors for sufficiently small \( \delta_u \).

Thus, the LFT method can be used to decrease the number of computational elements where the discrete transport terms are evaluated. Both the local error induced by the LFT method and the resulting improvement in computational efficiency are directly related to the choice of \( \delta_u \). It follows that where larger errors are acceptable greater computational efficiencies can be achieved. For comprehensive ecosystem models where constituent transport dominates the computational time and there are large uncertainties in the source/sink components of reaction terms, the LFT method has the potential to significantly decrease the overall computational costs without degrading the model validity.

The LFT method does not come without some overhead; thus in situations when some or all constituents vary significantly in large parts of the computational domain and over long simulation times the overall efficiency may not get improved or can even degrade. Another important aspect related to the implementation of the LFT method is the choice of the uniformity indicator. The slope limiter as an integral part of the DG discretization offers a particularly efficient and elegant solution to this problem, but a number of alternative techniques could be easily used in its stead providing a great degree of flexibility for large classes of numerical schemes.

An extension of the proposed framework to 3D would be rather straightforward for many types of passive tracers; however, a more sophisticated approach might be called for in the case of biogeochemical or sediment transport simulations. The natural anisotropy between the horizontal and vertical transport mechanisms, different boundary conditions, stratification effects may require a clear separation between the horizontal and vertical LFT schemes.

Table 5

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynamical core time, sec.</th>
<th>Transport time, sec.</th>
<th>Slope limiter time, sec.</th>
<th>Total time, sec.</th>
<th>Active elements (time mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Nitro</td>
</tr>
<tr>
<td>No LFT</td>
<td>55,293</td>
<td>209,232</td>
<td>127,306</td>
<td>391,831</td>
<td>-</td>
</tr>
<tr>
<td>LFT</td>
<td>54,849</td>
<td>133,106</td>
<td>126,402</td>
<td>314,357</td>
<td>75%</td>
</tr>
</tbody>
</table>

Fig. 10. Performance of LFT for Galveston Bay simulations: Size of active domain for each species in % of the total number of elements (upper), absolute \( L_\infty \)-error (kg/m\(^3\), log scale) of the LFT scheme using the unfiltered transport scheme as the reference (lower).
By introducing separate vertical and horizontal masks similar to those used for different constituents of the NPZ model our methodology can be readily extended to the 3D case.

The LFT method and its predecessor, background filtering for mass transport (Hodges, 2014), are ideas that are still in early development and exploration. Herein, we have provided the mathematical foundations for a mass-conservative approach with conventional hydrodynamic and transport algorithms. Demonstrating the full potential of the method to improve computational efficiency of scalar transport remains a subject for future exploration with a more complex ecosystem application. In particular, there is a need for a consistent and effective method for selecting the uniformity difference ($\delta_u$) as a function of transport scales and reaction coefficients and predicting the speed-up that can be achieved. At the present, selecting an effective $\delta_u$ is more art than science.

Acknowledgments

Hodges was partially supported by the US National Science Foundation under Grant Nos. 1331610 and 1417433, and the Research and Development program of the Texas General Land Office Oil Spill Prevention and Response Division under Grant No. 16-098-000-9290.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.envsoft.2018.01.003.

Appendix A. Discretization

Appendix A.1. Notation

Prior to describing our DG scheme for eqs. (1)–(3) we introduce some notation. Let $\mathcal{K} = \{T\}$ be a regular family of non-overlapping partitions of $\Omega$ into $K$ closed triangles $T$ of characteristic size $h$ such that $\Omega = \bigcup_{T \in \mathcal{K}} T$. For $T \in \mathcal{K}$, let $\nu_T$ denote the unit normal on $\partial T$ exterior to $T$. Let $\mathcal{K}_0$ denote the set of interior edges, $\mathcal{K}_0 = \mathcal{K} \setminus \{e\}$ the set of boundary edges, and $\mathcal{E} := \mathcal{K}_0 \cup \mathcal{K}_0 = \{e\}$ the set of all edges (the subscript $h$ is suppressed here). For an interior edge $E \in \mathcal{K}_0$ shared by triangles $T^-$ and $T^+$, we define the one-sided values of a scalar quantity $w = w(x)$ on $E$ by

\[
\begin{align*}
\overline{w}(x) & := \lim_{\varepsilon \to 0^-} w(x - \varepsilon \nu_T^-) \quad \text{and} \quad \overline{w}(x) := \lim_{\varepsilon \to 0^+} w(x - \varepsilon \nu_T^+),
\end{align*}
\]

respectively. For an edge $E \in \mathcal{K}_0$, only the first definition is meaningful.

Appendix A.2. Conservative equations

For compact notation we introduce a new set of unknowns $(q, s) = (\xi, U, V), s^T = (c_1 H, \ldots c_M H)$ and rewrite eqs. (1)–(3) as

\[
\begin{align*}
\partial_t q + \nabla \cdot B_1(q) &= M_1(q) \quad \text{(A.1)} \\
\partial_t s + \nabla \cdot B_2(q, s) &= M_2(q, s) \quad \text{(A.2)}
\end{align*}
\]

with

\[
B_1(q) = \begin{bmatrix}
\frac{U^2}{H} + g \frac{\xi}{\frac{\xi}{2} - b} \\
\frac{U V}{H} \\
\frac{V^2}{H} + g \frac{\xi}{\frac{\xi}{2} - b}
\end{bmatrix}, \quad B_2(q, s) = \begin{bmatrix}
\frac{U}{H} \\
\frac{U V}{H} \\
\frac{V^2}{H} + g \frac{\xi}{\frac{\xi}{2} - b}
\end{bmatrix}
\]

\[
M_1(q) = \begin{bmatrix}
H F_x - g \frac{\xi}{\frac{\xi}{2} - b} - \tau_{\partial} U + f_{\partial} V \\
H F_y - g \frac{\xi}{\frac{\xi}{2} - b} - \tau_{\partial} V - f_{\partial} U
\end{bmatrix}, \quad M_2(q, s) = \begin{bmatrix}
H R_{f, 1} \left[ \frac{s}{H} \right] + H F_1 \\
H R_{f, 2} \left[ \frac{s}{H} \right] + H F_{M}
\end{bmatrix}
\]

Appendix A.3. Semi-discrete formulation

Although our DG implementation of the shallow water and transport equations is very general and currently supports polynomial discretization spaces of orders zero to four, we only use piecewise linear discretization in this work. Thus, we denote by $\mathcal{P}_1(T)$ the space of polynomials of degree at most 1 on each element $T \in \mathcal{K}_0$.

Let $\mathcal{P}_1(\mathcal{K}_0) := \{ w_h : \Omega \to \mathbb{R} : \forall T \in \mathcal{K}_0, w_h \vert_T \in \mathcal{P}_1(T)\}$ denote the broken polynomial space on the triangulation $\mathcal{K}_0$. For the semi-discrete formulation, we assume that the coefficient functions for any fixed $t \in (0, T_{\text{end}})$ or their components in case of vector-valued coefficients are approximated in this space. Similarly to (Frank et al., 2015b), we use the standard $L^2$-projection into $\mathcal{P}_1(T)$ to compute these approximations.

Because of the local nature of the DG method, we can formulate the semi-discrete system of equations on an element-by-element basis by multiplying eqs. (A.1) and (A.2) with some test functions $\psi_h \in \mathcal{P}_1(\mathcal{K}_0)^3, \phi_h \in \mathcal{P}_1(\mathcal{K}_0)^M$ and integrating the advective terms by parts over element $T \in \mathcal{K}_0$. Then our semi-discrete formulation reads:

 Seek $(q_h(t), s_h(t)) \in \mathcal{P}_1(\mathcal{K}_0)^{3+M}$ such that the following holds for any $t \in (0, T_{\text{end}})$ and $\forall T \in \mathcal{K}_0, \psi_h \in \mathcal{P}_1(\mathcal{K}_0)^3, \phi_h \in \mathcal{P}_1(\mathcal{K}_0)^M$:

\[
\int_T \psi_h \cdot \partial_t q_h \, dx - \int_T \nabla \psi_h : B_1(q_h) \, dx + \int_T \psi_h \cdot B_1(q_h) \cdot \nu_T \cdot ds = \int_T \psi_h \cdot M_1(q_h) \, dx,
\]

\[
\int_T \phi_h \cdot \partial_t s_h \, dx - \int_T \nabla \phi_h : B_2(q_h, s_h) \, dx + \int_T \phi_h \cdot B_2(q_h, s_h) \cdot \nu_T \cdot ds = \int_T \phi_h \cdot M_2(q_h, s_h) \, dx.
\]
Due to the discontinuous nature of a DG approximation, particular care has to be taken when computing the boundary fluxes. The specific way to perform this task has critical impact on the stability, accuracy, and conservation properties of the numerical scheme. In eq. (A.3), these fluxes are obtained as solutions for local Riemann problems using the Lax-Friedrichs approximation (Aizinger, 2004) combined with the Roe-Pike averaging (Roe and Pike, 1985):

\[
B_I(q_h) \cdot \nu_T = \frac{1}{2} \left( \left( B_1(q_h^0 + B_1(q_h^3)) \right) \cdot \nu_T + \left( \left( q_h^3 - q_h^0 \right) \right) \right)
\]

(A.5)

with (adding \( h \)-subscripts to indicate discrete unknowns)

\[
\hat{\lambda} := \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} \cdot \nu_T + \frac{1}{2} \left( \begin{bmatrix} H^1_h + H^2_h \\ H^2_h + H^1_h \\ H^1_h + H^2_h \\ H^2_h + H^1_h \end{bmatrix} \right) \frac{1}{2} \left( \begin{bmatrix} H^1_h + H^2_h \\ H^2_h + H^1_h \\ H^1_h + H^2_h \\ H^2_h + H^1_h \end{bmatrix} \right)
\]

where the vectors of degrees of freedom associated with \( \hat{u} \) and \( \hat{v} \) are the vectors of degrees of freedom associated with \( q_h^0 \) and \( q_h^3 \), respectively. The basis functions \( B_I(\cdot) \) are the first component of the basis function of \( \mathbb{P}_4 \) with \( h \)-subscripts.

\[ \langle B_I(q_h) \rangle \cdot \nu_T = \frac{1}{2} \left( \begin{bmatrix} H^1_h + H^2_h \\ H^2_h + H^1_h \\ H^1_h + H^2_h \\ H^2_h + H^1_h \end{bmatrix} \right) \frac{1}{2} \left( \begin{bmatrix} H^1_h + H^2_h \\ H^2_h + H^1_h \\ H^1_h + H^2_h \\ H^2_h + H^1_h \end{bmatrix} \right)
\]

The semi-discrete formulation in eqs. (A.3) and (A.4) is discretized using an orthonormal (with respect to the standard \( L^2 \)-scalar product) piecewise linear DG basis resulting in a time-dependent system of equations for the degrees of freedom. As a first step, all functions and unknowns are expressed in terms of a local basis \( \langle \phi_k(1), \phi_k(2), \phi_k(3) \rangle \) of \( \mathbb{P}_1(T_{h}) \) as described in (Frank et al., 2015b; Reuter et al., 2016; Jaust et al., 2017) for a fixed \( T_h \in \mathcal{T}_h \). The basis functions defined in the reference coordinate system \((\hat{x}, \hat{y})\) on the unit reference triangle \( \hat{T} \) are:

\[ \hat{\phi}_1(\hat{x}) = \sqrt{2}, \quad \hat{\phi}_2(\hat{x}) = 2 - 6\hat{x}^2, \quad \hat{\phi}_3(\hat{x}) = 2\sqrt{3} \left( 1 - \hat{x}^2 - 2\hat{x}^2 \right). \]

(A.6)

Thus, a scalar function \( \phi_I(\mathbf{x}) \) can be locally written as \( \phi_I(\mathbf{x})|_{T_h} = \sum_{k=1}^{3} a_{k,I} \phi_k(\mathbf{x}) \), where \( a_{k,I} \) denote the degrees of freedom of the representation of \( \phi_I(\mathbf{x}) \) in the local DG basis. The latter is obtained as affine-linear transformation of the reference basis given in eq. (A.6).

Appendix A.5. Time discretization and slope limiting

Time discretization of eqs. (A.3) and (A.4) uses an explicit Runge–Kutta method (Gottlieb and Shu, 1998) of order two that belongs to the class of SSP (strong stability preserving) methods (Gottlieb et al., 2001). This type of time-stepping method preserves the monotonicity of the solution if it is postprocessed by a slope limiter. Details on how these methods are implemented can be found in (Reuter et al., 2016).

Let \( 0 = t^0 < t^1 < \ldots < t^{n+1} \) be a not necessarily equidistant decomposition of the time interval \((0,T)\) and let \( \Delta t^n := t^{n+1} - t^n \) denote the time step size. System (A.3)–(A.4) can be compactly written in operator form as

\[ M_q \partial_t Q_h(t) = L_q(t, Q_h), \quad M_s \partial_t S_h(t) = L_s(t, Q_h, S_h), \]

(A.7)

where \( Q_h, S_h \) are the vectors of degrees of freedom associated with \( q_h, s_h, M_q, M_s \) are the corresponding mass matrices, and \( L_q, L_s \) are the non-linear operators comprising terms not involving time derivatives in eqs. (A.3) and (A.4). The update scheme of the Runge–Kutta method is then given by

\[ \begin{bmatrix} Q_h^{(0)} \\ S_h^{(0)} \end{bmatrix} = \begin{bmatrix} Q_h \\ S_h \end{bmatrix}, \]

\[ \begin{bmatrix} Q_h^{(1)} \\ S_h^{(1)} \end{bmatrix} = \begin{bmatrix} Q_h^{(0)} \\ S_h^{(0)} \end{bmatrix} + \Delta t^n \begin{bmatrix} M_q L_q \left( t^n, Q_h^{(0)} \right) \\ M_s L_s \left( t^n, Q_h^{(0)}, S_h^{(0)} \right) \end{bmatrix}, \]

\[ \begin{bmatrix} Q_h^{(2)} \\ S_h^{(2)} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} Q_h^{(1)} \\ S_h^{(1)} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} Q_h^{(0)} \\ S_h^{(0)} \end{bmatrix} + \frac{\Delta t^n}{2} \begin{bmatrix} M_q L_q \left( t^{n+1}, Q_h^{(1)} \right) \\ M_s L_s \left( t^{n+1}, Q_h^{(1)}, S_h^{(1)} \right) \end{bmatrix}, \]

\[ \begin{bmatrix} Q_h^{(n+1)} \\ S_h^{(n+1)} \end{bmatrix} = \begin{bmatrix} Q_h^{(2)} \\ S_h^{(2)} \end{bmatrix}, \]

(A.8)

where we abbreviated \( Q_h^{(n)} := Q_h(t^n) \), etc.

To preserve the monotonicity of the solution and to prevent the appearance of negative concentrations, the numerical solution has to be postprocessed using a slope limiter. The methods used here are vertex-based limiters proposed by Kuzmin in (Kuzmin, 2010) and Aizinger in (Aizinger, 2011) and further generalized in (Kuzmin, 2013; Reuter et al., 2016). Their advantages include low numerical diffusion, support of arbitrary element shapes, and intuitive extensions to higher order DG discretizations and anisotropic problems (Aizinger et al., 2017). Since the limiters play an essential role in our modified transport algorithm, a short description is in order.

The common idea underlying all limiters is to rely on the piecewise constant part of the discrete solution — guaranteed to be monotonous — to limit higher-order degrees of freedom of the DG solution. Formulating the vertex-based limiting scheme for a linear DG approximation is very simple for a hierarchical basis (e.g., Taylor
the minimum and maximum mean solution values of normal with respect to the elements containing information, particularly $\varphi_{k2}$ and $\varphi_{k3}$ are orthogonal to any constant function, and, by construction, $\varphi_{k1} = \frac{1}{|T_k|}$ holds, where $|T_k| := \int_{T_k} dx$ is the area of $T_k$. Thus

$$\sigma_k := \frac{1}{|T_k|} \int_{T_k} a_i(x) \, dx = a_i \left| \frac{1}{T_k} \right|^\frac{1}{2},$$

where $\sigma_k$ denotes the integral mean of $a_i$ on $T_k$. Therefore, any changes in the linear degrees of freedom $a_{k2}$ and $a_{k3}$ do not affect the local conservation properties of the solution since the latter only depends on $a_{k1}$; by adjusting coefficients $a_{k2}$, $a_{k3}$ in an appropriate way the slope limiting procedure produces a piecewise linear DG solution fulfilling the maximum principle.

We seek the maximum admissible value of coefficient $\theta_k$, $0 \leq \theta_k \leq 1$ in

$$a_i(x)|_{T_k} = a_{k1} \varphi_{k1}(x) + \theta_k (a_{k2} \varphi_{k2}(x) + a_{k3} \varphi_{k3}(x)).$$

Here, $\theta_k = 1$ corresponds to an unlimited solution, and $\theta_k = 0$ reduces the solution to its piecewise constant component. The correction factor $\theta_k$ is chosen as a maximum admissible value so that the above reconstruction is bounded in all vertices $x_i \in T_k$ by the minimum and maximum mean solution values of $a_i$ on all elements containing $x_i$ (see Fig. A.11):

$$a_i^{\min} \leq a_i(x)|_{T_k} \leq a_i^{\max}, \quad \forall : x_i \text{ vertex of } T_k,$$

where

$$a_i^{\min} := \min_{x_i} (\sigma_k), \quad a_i^{\max} := \max_{x_i} (\sigma_k), \quad x_i \text{ vertex of } T_k.$$

To enforce eq. (A.9), the correction factor $\theta_k$ is defined as in (Kuzmin, 2010),

$$\theta_k := \min_{x_i \text{ vertex of } T_k} \begin{cases} a_i^{\max} - \sigma_k, \quad \text{if } a_i(x)|_{T_k} > a_i^{\max}, \\ a_i^{\min} - \sigma_k, \quad \text{if } a_i(x)|_{T_k} < a_i^{\min}, \\ 1, \quad \text{otherwise}. \end{cases}$$

Alternatively, the same $\theta_k$ can be obtained by solving a one-dimensional optimization problem as shown in (Aizinger, 2011).

The fact that our slope limiting procedure must be performed on the values on concentration unknowns $c_m$, $m = 1, \ldots, M$ instead of the depth integrated concentrations $s_m$ that serve as the primary unknowns in transport eq. (A.2) suggests the following structure of each substep of the time stepping algorithm given in eq. (A.8):

$$s_h \rightarrow c_h := \Pi_1(s_h) \rightarrow s_h^{\lim} := \Pi_1(L(c_h) H_h),$$

where $\Pi_1(\cdot)$ denotes the $L^2$-projection into broken piecewise linear polynomial space $P_1(\mathbb{R}^D)$ and $L(\cdot)$ the slope limiting operator. This procedure must be performed after each time substep in eq. (A.8) and replaces values of $s_h^i$, $i = 1, 2$ there with their limited counterparts.

The values of $a_i^{\min}$ and $a_i^{\max}$ for each species calculated in the course of our limiting procedure can be then employed in the modified transport scheme described in Sec. 3. The above scheme is computationally efficient compared to the region identification approach used in the substep time stepping work of (Hodges, 2014).

References


Aizinger, V., 2011. A geometry independent slope limiter for the discontinuous


Aizinger, V., Dawson, C., 2002. A discontinuous Galerkin method for two-
dimensional flow and transport in shallow water. Adv. Water Resour. 25 (1),

67–84.


https://doi.org/10.1002/fdm.3490.


361–369.


Gottlieb, S., Shu, C.-W., 1998. Strong stability-preserving high-order time dis-

cretization methods. Math. Comp. 67 (221), 73–95.


Gruber, N., 2008. The marine nitrogen cycle: overview and challenges. Nitro-

egen. 1–50.


1178–1190.


1674–1693.


plankton in a large lake supports the nearshore shunt hypothesis and the