Conservative explicit local time-stepping schemes for the shallow water equations

Thi-Thao-Phuong Hoang[a,d], Wei Leng[b], Lili Ju[a,*], Zhu Wang[a], Konstantin Pieper[c]

[a]*Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA*
[b]*State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences, Beijing 100190, China*
[c]*Department of Scientific Computing, Florida State University, Tallahassee, FL 32306, USA*
[d]*Department of Mathematics and Statistics, Auburn University, Auburn, AL 36849, USA*

Abstract

In this paper we develop explicit local time-stepping (LTS) schemes with second and third order accuracy for the shallow water equations. The system is discretized in space by a C-grid staggering method, namely the TRiSK scheme adopted in MPAS-Ocean, a global ocean model with the capability of resolving multiple resolutions within a single simulation. The time integration is designed based on the strong stability preserving Runge-Kutta (SSP-RK) methods, but different time step sizes can be used in different regions of the domain through the coupling of coarse-fine time discretizations on the interface, and are only restricted by respective local CFL conditions. The proposed LTS schemes are of predictor-corrector type in which the predictors are constructed based on Taylor series expansions and SSP-RK stepping algorithms. The schemes preserve some important physical quantities in the discrete sense, such as exact conservation of the mass and potential vorticity and conservation of the total energy within time truncation errors. Moreover, they inherit the natural parallelism of the original explicit global time-stepping schemes. Extensive numerical tests are presented to illustrate the performance of the proposed algorithms.

Keywords: Shallow water equations, local time-stepping, strong stability preserving Runge-Kutta, finite volume, mass conservation, potential vorticity

1. Introduction

MPAS-Ocean (Model for Prediction Across Scales) is a global ocean model recently introduced in [34] with the capability of resolving multiple resolutions within a single simulation. It, hence, can be used for both global ocean climate-change applications and coastal applications with different spatial scales between different regions of the domain. Because of its natural parallelism and low computational cost, explicit global time-stepping has been implemented and widely used in MPAS-Ocean. However, it is very restrictive due to the CFL condition controlled by the size

*Corresponding author

Email addresses: `tzh0059@auburn.edu` (Thi-Thao-Phuong Hoang), `wleng@lsec.cc.ac.cn` (Wei Leng), `ju@math.sc.edu` (Lili Ju), `wangzhu@math.sc.edu` (Zhu Wang), `kpieper@fsu.edu` (Konstantin Pieper)

of the smallest cell and results in computational inefficiency in the context of a multi-resolution approach. Therefore, the use of multi-resolution spherical centroidal Voronoi tessellation (SCVT) meshes [9, 10, 32, 22] in MPAS naturally requires the development of multiscale time-stepping algorithms, especially for the coastal-ocean modeling in which the time step sizes taken near the coast should be much smaller than those for the ocean, ensuring preservation of important physical properties such as the conservation of mass, potential vorticity, and energy. Taking this challenge, we develop conservative, explicit, local time-stepping (LTS) algorithms in this work, in which local time refinement is coupled with local spatial refinement and the time step size depends on the locally varying CFL condition, but not on the global CFL condition. Although we will restrict ourselves to explicit time-stepping schemes, one surely can enforce nonconforming time grids implicitly using the domain decomposition approach [13, 4, 18, 19].

A conservative, LTS algorithm was first introduced in [29] for one-dimensional scalar conservation laws based on the forward Euler discretization in time. The proposed scheme is first order accurate in both space and time. Extension to high resolution schemes with slope limiters for advection equations was presented in [7], and to second order in time was thoroughly analyzed for one-dimensional scalar conservation laws in [8] with two-dimensional numerical results showing that LTS schemes are very competitive to the global time-stepping with respect to the accuracy in time. In addition, numerical studies of LTS schemes for two and three dimensional problems in [11, 44, 45, 28] confirm that the schemes are most advantageous when coupled with local spatial refinement. Application of LTS schemes to the shallow water equations (SWEs) was investigated in [38] using a Godunov-type finite volume method for the spatial discretization and in [47] using Runge-Kutta discontinuous Galerkin finite element methods. Note that the LTS scheme in [38] is first order accurate in time while the one in [47] is second order in time away from the local time-stepping interface and first order at the interface. The time-stepping scheme used in [47] is the second order strong stability preserving Runge-Kutta (SSP-RK) method, also known as a total variation diminishing method introduced in [40, 43]. High-order Runge-Kutta (RK) based explicit LTS methods were introduced for conservation laws in [24, 1] and for wave propagation in [16].

Other works related to LTS include the adaptive mesh refinement (AMR) method [2, 3], the multirate time-stepping method [6] and the Implicit-Explicit (IMEX) based LTS methods [36, 17]. The AMR method involves the refinement in both space and time, i.e., small time step sizes are taken on the refined mesh and large time step sizes on the coarse mesh. It is different from our approach in the way that refined grids are placed over regions of the coarse grid and information is exchanged between the grids by means of injection and interpolation. The multirate time-stepping method allows different time step sizes in different regions but it requires buffer regions to accommodate the time scale transition between regions. An overview of LTS techniques over the last two decades can be found in [12].

As a first step toward developing efficient local time-stepping algorithms for the ocean model by MPAS-Ocean, in this work we consider discretizing the shallow water equations in space by the TRiSK scheme [46, 35]. The TRiSK scheme is a C-grid staggering, finite-volume type scheme used in MPAS-Ocean for the horizontal discretization and is up to second order accurate in space [46, 35]. It involves both primal and dual variables in a way such that many desirable properties, such as the conservation of some important nonlinear physical quantities on highly variable meshes, could be achieved. The objective of this study is to design LTS algorithms working with TRiSK spatial discretization such that the conservation properties are still preserved after time integration. We construct up to third order accurate LTS algorithms based on the SSP-RK method. For long time

2

simulations, the third order SSP-RK scheme is preferable to the second order one as it allows larger time step sizes and has a much better stability region; in particular, it is well known that the second order SSP-RK method is identical to the classical Heun's method whose stability region does not contain any purely imaginary number. Extension to even higher order in time schemes could be done in a similar way.

The proposed LTS schemes are of predictor-corrector type which was first introduced in [29] using the forward Euler time-stepping. In order to construct high order LTS algorithms, the key idea is to find high order approximations on the interface at intermediate time levels to handle the coupling between coarse and fine time steppings. Here we derive the predictors based on Taylor series expansions of the solution at the current time level and the SSP-RK stepping algorithms at each intermediate time level. The resulting LTS scheme is $(p + 1)$-th order accurate (for $p = 1, 2$) provided that the predictor is a p-th order Taylor series approximation (with a truncation error of $(p + 1)$-th order) and the SSP-RK scheme is of $(p + 1)$-th order accuracy. It should be noted that for the first order time-stepping ($p = 0$), this approach is identical to the first LTS scheme introduced in [29], where the predictor is obtained by freezing the values at the current time level. Therefore, the approach presented in this paper can be seen as a generalization of [29] to high order time discretizations. In addition, because of the well-defined predictor, the accuracy in time of the proposed schemes is consistent on the whole domain; in particular, the LTS algorithm based on SSP-RK2 is second order accurate in time even at the local time-stepping interface, instead of only first order accurate in time at the interface as in [47]. Exact conservation of mass is guaranteed through the use of the corrector which is defined based on the idea in [29]. Note that high order LTS schemes of predictor-corrector type were also introduced in [24, 1] for conservation laws. However, the predictors in [24, 1] are designed for Runge-Kutta time-steppings and obtained by using interpolating polynomials of degree $(p + 1)$ for the $(p + 1)$-th order Runge-Kutta method. In fact, it is not necessary to use interpolating polynomials of degree higher than p to construct the predictor for the $(p + 1)$-th order LTS scheme because the predicted values for the interface only contribute to the fluxes at the interface when advancing in time in regions with smaller time step sizes. These interface values are corrected later by following the current time-stepping scheme (and in such a way as to conserve fluxes). Thus, the predictors proposed in this work are essentially different from those in [24, 1]. Additionally, the proposed LTS algorithms built on the TRiSK discretization in space have some specific features due to the use of C-grids and staggered variables, for example, the introduction of two interface layers (instead of one layer as usual) to balance fluxes between regions of different time step sizes. Numerical tests confirm that there is no loss of accuracy due to local time-stepping for our LTS schemes when compared to using the globally defined CFL time step size, and that the total energy (sum of the potential and kinetic energy) is well conserved within the time truncation error. The proposed schemes also inherit the natural parallelism of the explicit global time-stepping schemes.

The paper is outlined as follows: in Section 2, we introduce the model shallow water equations at the continuous level and discuss the dynamics of potential vorticity. Then we give a brief description of the TRiSK scheme for the spatial discretization of the problem in Section 3. Our main results are then presented in Section 4 in which the second and third order local time-stepping schemes are derived and conservation of mass is proved. Numerical results on various test cases are discussed in Section 5 and some concluding remarks are finally given in Section 6. Additionally, detailed derivation of the predictors used in the LTS schemes is presented in Appendix A.

2. Nonlinear shallow water equations

We consider the nonlinear, inviscid shallow water equations in the "vector-invariant" form [35] as follows:

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

$$\frac{\partial \boldsymbol{u}}{\partial t} + \eta \boldsymbol{u}^{\perp} = -g\nabla(h+b) - \nabla K \tag{2}$$

with the initial conditions $h(\cdot, 0) = h_0$, $\boldsymbol{u}(\cdot, 0) = \boldsymbol{u}_0$ and appropriate boundary conditions if needed. The above hyperobolic system describes the motion of a layer of fluid on a two-dimensional surface in a rotating reference frame, where h represents the fluid layer thickness, $\boldsymbol{u}$ the fluid vector velocity along the surface of the sphere, $\boldsymbol{u}^{\perp} := \boldsymbol{k} \times \boldsymbol{u}$ the velocity rotated through a right angle with $\boldsymbol{k}$ the unit vector pointing in the local vertical direction such that $\boldsymbol{k} \cdot \boldsymbol{u} = 0$, the absolute vorticity η is defined as $\boldsymbol{k} \cdot \nabla \times \boldsymbol{u} + f$ and the kinetic energy K is $|\boldsymbol{u}|^2/2$. The parameters involved are gravity g, Coriolis parameter f and bottom topography b.

The fluid potential vorticity (PV), denoted by q, is related to the absolute vorticity and thickness field as

$$q = \frac{\eta}{h}. \tag{3}$$

Substituting (3) into (2), we obtain another form of the momentum equation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + q(h\boldsymbol{u}^{\perp}) = -g\nabla(h+b) - \nabla K, \tag{4}$$

where $q(h\boldsymbol{u}^{\perp})$ is the nonlinear Coriolis force that contains the quasi-linear Coriolis force $f\boldsymbol{u}^{\perp}$ and the rotational part of the nonlinear transport term. It has been observed and proven that the PV is a key quantity in the interpretation of atmosphere and ocean dynamics (e.g. [5, 21, 27]) and it is important for numerical models to faithfully represent some aspects of the PV dynamics, such as its compatibility with the momentum equation and its Lagragian behavior. In particular, one can derive the PV equation by manipulating the momentum equation (2):

$$\frac{\partial \eta}{\partial t} + \nabla \cdot [\eta \boldsymbol{u}] = 0, \qquad (\text{by taking } \boldsymbol{k} \cdot \nabla \times (2)) \tag{5}$$

or equivalently, given the relationship between the absolute vorticity and the PV,

$$\frac{\partial (hq)}{\partial t} + \nabla \cdot [hq\boldsymbol{u}] = 0. \tag{6}$$

Thus the evolution of the thickness-weighted PV (hq) depends solely on the divergence of the PV flux ($hq\boldsymbol{u}$), and so the conservation of PV is guaranteed.

Concerning the PV Lagrangian property, it means that PV evolves, in the frictionless and adiabatic limit, as $Dq/Dt = 0$ where D/Dt is the material derivative. This can be recovered from (6) and the thickness equation (1):

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \boldsymbol{u} \cdot \nabla q = 0. \tag{7}$$

A numerical scheme is considered to be consistent with respect to the Lagrangian behavior of PV if a discrete analog of (7) is satisfied.

Remark 2.1. *It is shown in Equations (2) and (4) that the nonlinear Coriolis force plays a fundamental role in both system energetics and potential vorticity dynamics. Consequently, robust construction of the discrete Coriolis force is critical to ensure the accuracy of the simulation of geophysical flows. We also can obtain from (6) the relationship between the PV-weighted thickness flux $hq\boldsymbol{u}$ and the nonlinear Coriolis force $hq\boldsymbol{u}^{\perp}$ in the momentum equation (4):*

$$\frac{\partial(hq)}{\partial t} = -\nabla \cdot [hq\boldsymbol{u}] = -\boldsymbol{k} \cdot \nabla \times (hq\boldsymbol{u}^{\perp}). \tag{8}$$

Such a relationship is a key point of the discretization of the momentum equation so that the discrete PV equation is both consistent and conservative.

3. Spatial discretization of the SWEs: the TRiSK scheme

The TRiSK scheme provides a unified approach to the conservation of total energy and a robust simulation of potential vorticity on unstructured, locally orthogonal meshes. In this section, we briefly recall the TRiSK scheme (we refer to [46, 35] for a full presentation of the scheme). We begin with a description of the spherical centroidal Voronoi tessellation (SCVT) [9, 10] as the primal mesh for the TRiSK scheme and some notations that will be used in the spatially semi-discretized system, then we present the spatially discrete operators involved in the scheme.

3.1. SCVT meshes

Let us consider a SCVT of the surface of a sphere and its dual Delaunay triangulation, which will be used as the primal and dual meshes for the TRiSK scheme. Note that the dual edges are perpendicular to and bisected by the primal edges in the SCVT as illustrated in Figure 1. Notations used in the semi-discrete system are defined in Tables 1-3 for the mesh elements and quantities, the mesh connectivity, the normal and tangential vectors respectively.
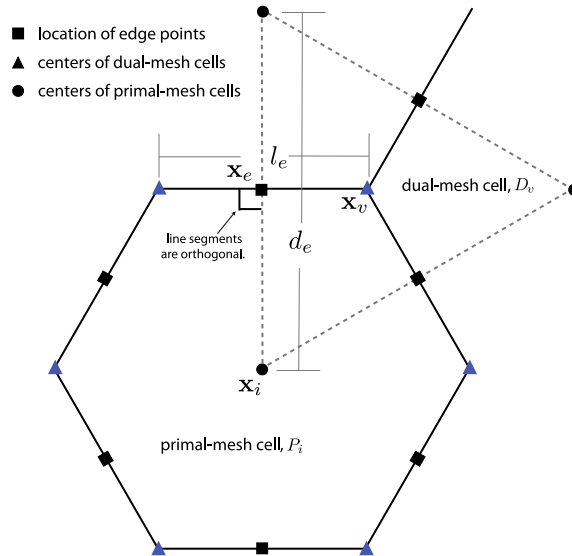

Figure 1: An illustration of the primal and dual cells used for the TRiSK scheme [35].

Notation	Definition
$\mathbf{x}_i$	centers of primal mesh cells
$\mathbf{x}_v$	centers of dual mesh cells
$\mathbf{x}_e$	the intersection point between the primal and dual edges
P_i	primal cells corresponding to $\mathbf{x}_i$
D_v	dual cells corresponding to $\mathbf{x}_v$
l_e	length of the primal edge e
d_e	length of the dual edge intersecting e
A_i	area of the primal cell P_i
A_v^{d}	area of the dual cell D_v
A_e^{e}	area associated with the primal edge e: $A_e^{\mathrm{e}} = {}^{1}\!/{}_{2}\, l_e d_e$

Table 1: List of notations used to describe the mesh elements and quantities.

Notation	Definition
$e \in EC(i)$	set of edges of the primal cell P_i
$i \in CE(e)$	two primal cells either side of primal edge e
$e \in EV(v)$	set of primal edges sharing vertex v
$i \in CV(v)$	set of primal cells having v as their vertex
$v \in VE(e)$	two endpoints of primal edge e
$e' \in ECP(e)$	set of primal edges nearby e

Table 2: List of notations used to describe the mesh connectivity.

Notation	Definition
$\boldsymbol{n}_e$	the unit vector at $\mathbf{x}_e$ normal to e in the direction corresponding to positive u_e
$\boldsymbol{t}_e$	the unit tangential vector at $\mathbf{x}_e$: $\boldsymbol{t}_e = \boldsymbol{k} \times \boldsymbol{n}_e$
$n_{e,i}$	normal indicator function

$$n_{e,i} = \begin{cases} 1 & \text{if } \boldsymbol{n}_e \text{ is an outward normal of } P_i, \\ -1 & \text{otherwise.} \end{cases} \qquad (9)$$

$t_{e,v}$	tangential indicator function

$$t_{e,v} = \begin{cases} 1 & \text{if } v \text{ is in the direction } \boldsymbol{k} \times \boldsymbol{n}_e \ , \\ -1 & \text{otherwise,} \end{cases} \qquad (10)$$

i.e., counterclockwise circulation about vertex v contributes positively to the vorticity at v.

Table 3: List of notations used to describe normal and tangential vectors.

3.2. The spatially semi-discretized system

We first rewrite the continuous SWE system in a more convenient form:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \nabla \cdot (\boldsymbol{F}) = 0, \\[2mm] \dfrac{\partial \boldsymbol{u}}{\partial t} + q(\boldsymbol{F}^{\perp}) = -\left(g\nabla(h+b) + \nabla K\right), \end{cases} \qquad (11)$$

where $\boldsymbol{F} := h\boldsymbol{u}$ is the thickness flux, and $\boldsymbol{F}^{\perp} := \boldsymbol{k} \times h\boldsymbol{u}$ is perpendicular to $\boldsymbol{F}$. By using a C-grid staggering in which the fluid thickness is stored at the primal cell centers, the normal component

of velocity stored at the primal edges and the PV stored at the dual cell centers, we obtain the discrete mass and momentum equations as follows:

$$\begin{cases} \dfrac{\partial h_i}{\partial t} = -[\nabla \cdot F_e]_i, \\[2mm] \dfrac{\partial u_e}{\partial t} = -F_e^\perp \widehat{q}_e - [g\nabla(h_i + b_i) + \nabla K_i]_e, \end{cases} \tag{12}$$

where h_i represents the mean thickness over the primal cell P_i and u_e the component of the velocity vector in the direction normal to the primal edge e. The thickness flux is defined per unit length across the primal edge e as

$$F_e := \widehat{h}_e u_e,$$

in which $\widehat{h}_e = [h]_{i\to e}$ denotes an average value of h on the primal edge e computed from its values h_i on the neighboring primal cells. We remark that this choice of average for computing $\widehat{h}_e$ is dependent on the desired physics applications. For wetting and drying cases, it is essential to use upwinding rather than averaging. Similarly, $\widehat{q}_e := [q]_{v\to e}$ is an average value of q on the primal edge e computed from its values q_v on the neighboring dual cells. Note that the discrete PV q_v represents the mean PV over the dual cell D_v and will be defined later.

As in the continuous equation (11), the flux $F_e^\perp$ is perpendicular to F_e and it is defined by using the robust flux interpolation scheme derived in [46]. This will be discussed further in Subsection 3.2.2. In the following, let us define in detail the discrete divergence $[\nabla \cdot (\cdot)]_i$, discrete gradient $[\nabla(\cdot)]_e$, discrete curl $[\nabla \times (\cdot)]_v$ and four scalar terms $F_e^\perp, \widehat{q}_e, \widehat{h}_e$ and K_i involved in the semi-discrete system (12). We note the spatial accuracy of the TRiSK scheme is usually between first and second orders depending on the quality of the meshes [46, 35].

3.2.1. Discrete operators

Using finite volume and finite difference approaches, the formulation for discrete divergence, gradient and curl operators in the TRiSK scheme are defined as follows:

i) Discrete divergence $[\nabla \cdot (\cdot)]_i$:

$$(\nabla \cdot \boldsymbol{F})_i := \frac{1}{A_i} \sum_{e \in EC(i)} n_{e,i} F_e l_e. \tag{13}$$

ii) Discrete gradient $[\nabla(\cdot)]_e$:

$$(\nabla h)_e := \frac{1}{d_e} \sum_{i \in CE(e)} -n_{e,i} h_i. \tag{14}$$

iii) Discrete curl $[\nabla \times (\cdot)]_v$:

$$\boldsymbol{k} \cdot (\nabla \times \boldsymbol{F})_v := \frac{1}{A_v^{(d)}} \sum_{e \in EV(v)} t_{e,v} F_e d_e. \tag{15}$$

3.2.2. Flux interpolation

Following the flux interpolation scheme in [46], $F_e^\perp$ is defined as a weighted combination of $F_{e'}$ at the edges e' surrounding the two cells either side of e:

$$F_e^\perp := \frac{1}{d_e} \sum_{e' \in ECP(e)} w_{e,e'} l_{e'} F_{e'} = M(F_e), \tag{16}$$

7

in which $w_{e,e'}$ is the weight to be specified in the following. Denote by $\delta_i^{\boldsymbol{F}} := (\nabla \cdot \boldsymbol{F})_i$, the divergence of $\boldsymbol{F}$ on the primal cell P_i, and by $\delta_v^{\boldsymbol{F}^\perp} := (\nabla \cdot \boldsymbol{F}^\perp)_v$, the divergence of $\boldsymbol{F}^\perp$ on the dual cell D_v.

Then the weight $w_{e,e'}$ is chosen as in [35] so that $\delta_v^{\boldsymbol{F}^\perp}$ is a convex combination of the surrounding $\delta_i^{\boldsymbol{F}}$:

$$\delta_v^{\boldsymbol{F}^\perp} = \frac{1}{A_v^{(d)}} \sum_{i \in CV(v)} R_{i,v} A_i \delta_i^{\boldsymbol{F}} = I(\delta_i^{\boldsymbol{F}}), \tag{17}$$

where $R_{i,v} \geq 0$ is the weight associated with each primal-dual mesh intersection area and satisfies

$$\sum_{v \in VC(i)} R_{i,v} = 1.$$

Explicit formulas for $w_{e,e'}$ are given in [46] as follows: $w_{e,e} = 0$ for all primal edges e and

$$w_{e,e'} = \left(\sum_v R_{i,v} - \frac{1}{2} \right) n_{e',i} t_{e,v_*}, \quad \forall\, e' \in ECP(e),\ e' \neq e, \tag{18}$$

in which $i = CE(e) \cap CE(e')$ and the sum is taken over all vertices v between the edge e' and the edge e (either clockwise or counterclockwise around cell i). In addition, v_* is the last vertex encountered when traversing from the edge e' to the edge e.

3.2.3. Discrete PV

The discrete PV is defined at the centers of the dual cells

$$q_v = \frac{\eta_v}{h_v}, \tag{19}$$

where

$$\eta_v = \boldsymbol{k} \cdot [\nabla \times u_e \boldsymbol{n}_e]_v + f_v, \quad \text{and} \quad f_v = \frac{1}{A_v^{\mathrm{d}}} \int_{D_v} f.$$

The dual-mesh thickness h_v is computed from the primal-mesh thickness using the interpolation function I (cf. Equation (17)):

$$h_v = I(h_i). \tag{20}$$

The relationship between F_e and $F_e^\perp$ is used here to obtain consistent thickness, and consistent, compatible and conservative PV as proven in [35].

The averaging operator $\widehat{q}_e$ is defined in a way such that the nonlinear Coriolis force is energetically-neutral:

$$\sum_e A_e^{\mathrm{e}} F_e F_e^\perp \widehat{q}_e = 0. \tag{21}$$

In particular, the following choice is suggested in [35]:

$$\widehat{q}_e = \frac{Q_e^\perp}{F_e^\perp}, \tag{22}$$

where

$$Q_e^\perp = \frac{1}{d_e} \left[\sum_{e' \in ECP(e)} w_{e,e'} l_{e'} F_{e'} \left(\frac{\widetilde{q}_e + \widetilde{q}_{e'}}{2} \right) \right]. \tag{23}$$

8

To define $\widetilde{q}_e$, one can use the *arithmetic mean* to guarantee the energy conservation

$$\widetilde{q}_e = \frac{1}{2} \sum_{v \in VE(e)} q_v. \tag{24}$$

Note that the choice of $\widehat{q}_e$ in (22) to obtain an energetically-neutral formulation of the nonlinear Coriolis force on C-grids might not be the optimal choice from a potential enstrophy perspective. Alternatives to (22) including the potential enstrophy conserving and potential enstrophy dissipating schemes as discussed in [35, Subsection 3.8] can be accommodated within the TRiSK framework, though in that case conservation properties are not necessarily followed.

3.2.4. Averaging operators from primal cells to primal edges and vice versa

To ensure a conservative exchange between the potential energy $gh(h/2 + b)$ and the kinetic energy $h|\boldsymbol{u}|^2/2$ in the discrete sense, the discrete thickness on the primal edges is defined as

$$\widehat{h}_e = \frac{1}{2} \sum_{i \in CE(e)} h_i, \tag{25}$$

and the discrete kinetic energy on primal cells as

$$K_i = \frac{1}{A_i} \sum_{e \in EC(i)} A_e^{(e)} \frac{u_e^2}{2}, \tag{26}$$

where $A_e^{(e)}$ the area associated with the velocity point at $\mathbf{x}_e$:

$$A_e^{(e)} = \frac{1}{2} l_e d_e.$$

With the definitions of discrete operators and scalar terms above (in particular, (13),(14) and (15) for the discrete divergence, gradient and curl operators, (16) and (18) for flux $F_e^{\perp}$, (19), (22) and (24) for $\widehat{q}_e$, (25) and (26) for $\widehat{h}_e$ and K_i), the semi-discrete system (12) used by MPAS-Ocean is fully described.

4. Conservative explicit local time-stepping schemes

For time integration of the discrete nonlinear system (12), explicit methods are usually preferred because they are easy to implement and naturally parallel, facilitating good parallel scalability. We consider in this work the explicit, strong stability preserving Runge-Kutta (SSP-RK) methods of second and third order accuracy. In the context of multi-resolution meshes, these explicit schemes could become inefficient because the global CFL condition is restricted by the size of the smallest cell. Therefore, we propose LTS schemes with spatially varying time step sizes, which only requires a local CFL condition instead of the more restrictive, global CFL condition. In this section, we first describe global time-stepping schemes in which a single time step size Δt is enforced over the whole domain and then LTS schemes in which coarse and fine time step sizes are used in different regions of the problem domain. We will also prove that the proposed LTS schemes conserve mass exactly on the whole domain.

4.1. Global time-stepping by the SSP-RK methods

Consider a partition of the time interval $[0, T]$ into N subintervals

$$0 = t^0 < t^1 < \ldots < t^n < t^{n+1} < \ldots < t^N = T, \quad \Delta t^n = t^{n+1} - t^n.$$

The fully discrete system associated with (12) is then obtained by using the SSP-RK time discretization methods [42, 41] which were introduced to solve systems of ordinary differential equations (ODEs) resulting from spatial discretization of hyperbolic conservation laws. These methods satisfy the so-called total variation diminishing (TVD) property, i.e. the total variation of the discrete solution does not increase in time, which ensures stability of the solution. A thorough description and analysis of the SSP-RK methods can be found in [42, 41, 14, 15]. For the sake of simplicity of presentation, we rewrite the semi-discrete system (12) as

$$
\begin{cases}
\dfrac{\partial h_i}{\partial t} = \mathcal{F}_i(\boldsymbol{h}, \boldsymbol{U}), \\
\dfrac{\partial u_e}{\partial t} = \mathcal{G}_e(\boldsymbol{h}, \boldsymbol{U}),
\end{cases}
\tag{27}
$$

where $\boldsymbol{h} = \{h_i\}$, $\boldsymbol{U} = \{u_e\}$, $\mathcal{F}_i(\boldsymbol{h}, \boldsymbol{U}) = -[\nabla \cdot F_e]_i$ and $\mathcal{G}_e(\boldsymbol{h}, \boldsymbol{U}) = -F_e^\perp \widehat{q}_e - [g\nabla(h_i + b_i) + \nabla K_i]_e$.

Assume that $(\boldsymbol{h}^0, \boldsymbol{U}^0)$ is given. The *second order, two-stage* SSP-RK scheme, SSP-RK2, for solving (27) is defined as follows:

i) Stage 1:
$$
\begin{cases}
\overline{h}_i^{n+1} = h_i^n + \Delta t^n \mathcal{F}_i(\boldsymbol{h}^n, \boldsymbol{U}^n), \\
\overline{u}_e^{n+1} = u_e^n + \Delta t^n \mathcal{G}_e(\boldsymbol{h}^n, \boldsymbol{U}^n),
\end{cases}
\tag{28}
$$

ii) Stage 2:
$$
\begin{cases}
h_i^{n+1} = \dfrac{1}{2} h_i^n + \dfrac{1}{2} \left(\overline{h}_i^{n+1} + \Delta t^n \mathcal{F}_i(\overline{\boldsymbol{h}}^{n+1}, \overline{\boldsymbol{U}}^{n+1}) \right), \\
u_e^{n+1} = \dfrac{1}{2} u_e^n + \dfrac{1}{2} \left(\overline{u}_e^{n+1} + \Delta t^n \mathcal{G}_e(\overline{\boldsymbol{h}}^{n+1}, \overline{\boldsymbol{U}}^{n+1}) \right).
\end{cases}
\tag{29}
$$

for $n = 0, 1, \ldots, N - 1$.

The SSP-RK2 method is identical to the Heun's method which is essentially a predictor-corrector method with forward Euler's method as predictor and trapezoidal method as corrector. By increasing the number of stages, one can enlarge the regions of absolute stability of the second order SSP-RK methods (see [25, Figure 4]); however, these regions never contain purely imaginary numbers. The accuracy and stability can be improved by using higher order SSP-RK methods, in particular, here we use the *third order, three-stage* SSP-RK scheme, SSP-RK3, as follows:

i) Stage 1:
$$
\begin{cases}
\overline{h}_i^{n+1} = h_i^n + \Delta t^n \mathcal{F}_i(\boldsymbol{h}^n, \boldsymbol{U}^n), \\
\overline{u}_e^{n+1} = u_e^n + \Delta t^n \mathcal{G}_e(\boldsymbol{h}^n, \boldsymbol{U}^n),
\end{cases}
\tag{30}
$$

ii) Stage 2:
$$
\begin{cases}
\overline{h}_i^{n+1/2} = \dfrac{3}{4} h_i^n + \dfrac{1}{4} \left(\overline{h}_i^{n+1} + \Delta t^n \mathcal{F}_i(\overline{\boldsymbol{h}}^{n+1}, \overline{\boldsymbol{U}}^{n+1}) \right), \\
\overline{u}_e^{n+1/2} = \dfrac{3}{4} u_e^n + \dfrac{1}{4} \left(\overline{u}_e^{n+1} + \Delta t^n \mathcal{G}_e(\overline{\boldsymbol{h}}^{n+1}, \overline{\boldsymbol{U}}^{n+1}) \right).
\end{cases}
\tag{31}
$$

10

iii) Stage 3:

$$\begin{cases} h_i^{n+1} = \dfrac{1}{3}h_i^n + \dfrac{2}{3}\left(\overline{h}_i^{n+1/2} + \Delta t^n \mathcal{F}_i(\overline{\boldsymbol{h}}^{n+1/2}, \overline{\boldsymbol{U}}^{n+1/2})\right), \\ u_e^{n+1} = \dfrac{1}{3}u_e^n + \dfrac{2}{3}\left(\overline{u}_e^{n+1/2} + \Delta t^n \mathcal{G}_e(\overline{\boldsymbol{h}}^{n+1/2}, \overline{\boldsymbol{U}}^{n+1/2})\right). \end{cases} \tag{32}$$

for $n = 0, \ldots, N-1$.

The above SSP-RK2 and SSP-RK3 schemes are also known as the optimal (in the sense of number of stages required and the CFL coefficient) second and third order SSP-RK method [15]. It should be noted that SSP-RK3 allows the use of larger time step sizes compared to SSP-RK2 and its region of absolute stability contains purely imaginary numbers (see [26, Figure 2]), thus it is more efficient and practically more stable for long time simulations of the shallow water equations. This will be illustrated in the numerical experiments. Next we present our local time-stepping algorithms based on these two SSP-RK schemes.

4.2. Local time-stepping (LTS) schemes

For simplicity, we consider a partition of the computational domain into two zones, one with coarse time increments and the other with fine time increments as depicted in Figure 2:

$$[t^n, t^{n+1}) = \cup_{k=0}^{M-1}[t^{n,k}, t^{n,k+1}), \quad \text{where } t^{n,0} = t^n, \text{ and } t^{n,k+1} = t^{n,k} + \frac{\Delta t^n}{M}, \tag{33}$$

for all $k = 0, \ldots, M-1$.

In order to define the LTS algorithms for the discrete nonlinear system (12) obtained by the TRiSK scheme, we need to divide the primal cells into four different sets (see Figure 3):

i) $\mathcal{F}_P$ is the set of primal cells with fine time increments ('fine' cells),

ii) $\mathcal{C}_P^{\text{IF-L1}}$ is the set of primal cells with coarse time increments and neighboring the 'fine' cells (interface-layer 1 cells),

iii) $\mathcal{C}_P^{\text{IF-L2}}$ is the set of primal cells with coarse time increments and neighboring the interface-layer 1 cells (interface-layer 2 cells), and

iv) $\mathcal{C}_P^{\text{int}}$ is the set of remaining cells with coarse time increments ('coarse' internal cells).

The edges of the cells in these sets belong to $\mathcal{F}_E$ ('fine' edges), $\mathcal{C}_E^{\text{IF-L1}}$ (interface-layer 1 edges), $\mathcal{C}_E^{\text{IF-L2}}$ (interface-layer 2 edges), and $\mathcal{C}_E^{\text{int}}$ ('coarse' internal edges) respectively. Note that the common edge between an interface cell and a 'fine' cell belongs to $\mathcal{F}_E^{\text{IF}}$, the common edge between an interface-layer 1 cell and an interface-layer 2 cell belongs to $\mathcal{C}_E^{\text{IF-L1}}$ and the common edge between an interface layer 2 cell and a 'coarse' internal cell belongs to $\mathcal{C}_E^{\text{IF-L2}}$ as depicted in Figure 3. Here we have introduced two interface layers with coarse time increments, layer 1 and layer 2, to couple the systems in the 'fine' region with small time step sizes and in the 'coarse' region with large time step sizes. These two interface layers are treated differently, in particular, the interface-layer 1 is more crucial as it neighbors the 'fine' region and the introduction of the interface-layer 2 is specific to our proposed schemes as described next.

The proposed LTS algorithms are of predictor-corrector type. In order to advance between the intermediate time levels $t^{n,k}$ in the 'fine' region, we need to first predict the solution on the interface-layer 1 at all $t^{n,k}$. We have designed a predictor particularly associated with the SSP-RK
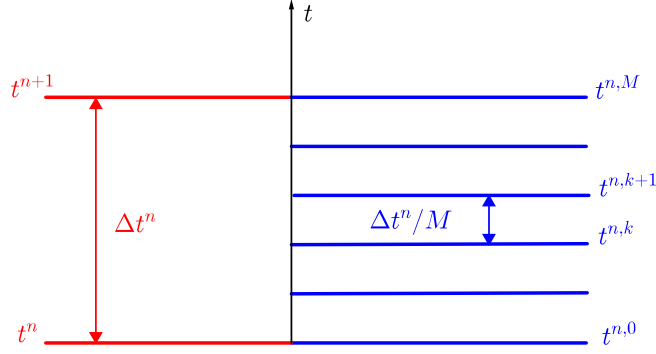
Figure 2: Coarse and fine time increments.
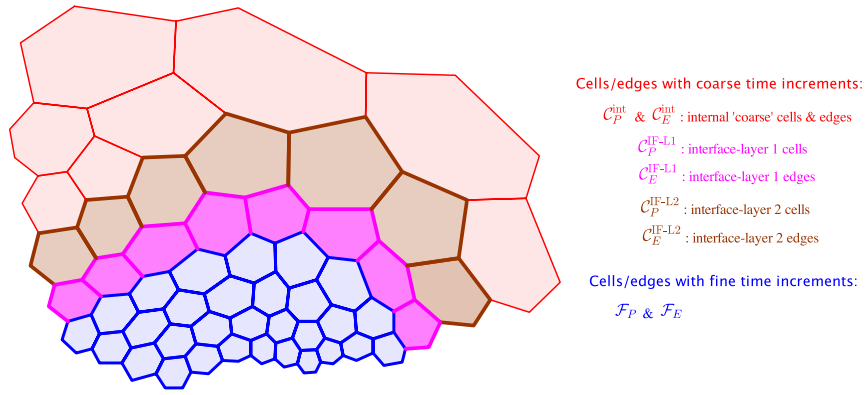
Figure 3: An illustration of different types of cells and edges with either coarse or fine time increments.

time discretization, which is based on Taylor series expansion in time of the solution at t^n on the interface-layer 1 with a coarse time step. The derivation of the predictors is presented in detail in Appendix A. Once the predictor is obtained, we can advance in time in the 'fine' region and in the internal 'coarse' region (excluding the two interface layers). The last step is correcting the interface values using the fluxes at intermediate time levels. Note that the introduction of the interface-layer 2 is necessary for exact mass conservation (cf. Theorem 4.4) in the context of the C-grid staggering spatial discretization; in particular, it helps to balance the nonlinear fluxes coming from the interface-layer 1 computed with approximate values by Taylor expansion. In the following, we give details of the algorithms based on the optimal second and third order SSP-RK schemes respectively.

4.2.1. Second order LTS algorithm

As SSP-RK2 consists of two stages, we use Taylor series expansions to predict the values at intermediate time levels on the interface-layer 1 for both stages. We also compute the solution at stage 1 on the interface-layer 2 to advance in time in the internal 'coarse' region. The predictor-corrector LTS algorithm based on SSP-RK2 is defined as follows: for $n = 0, 1, \ldots, N - 1$,

1) Interface prediction:

12

1a) First compute the values of the stage 1 of SSP-RK2 (28) on the interface-layer 1 with the coarse time step size: $\overline{h}_i^{n+1}$ for $i \in \mathcal{C}_P^{\text{IF-L1}}$ and $\overline{u}_e^{n+1}$ for $e \in \mathcal{C}_E^{\text{IF-L1}}$. Then use them to predict the values on interface cells/edges at intermediate time levels based on the first-order Taylor expansion as follows:

$$\begin{bmatrix} h_i^{n,k} \\ u_e^{n,k} \end{bmatrix} = (1 - \alpha_k) \begin{bmatrix} h_i^n \\ u_e^n \end{bmatrix} + \alpha_k \begin{bmatrix} \overline{h}_i^{n+1} \\ \overline{u}_e^{n+1} \end{bmatrix}, \tag{34}$$

$$\begin{bmatrix} \overline{h}_i^{n,k+1} \\ \overline{u}_e^{n,k+1} \end{bmatrix} = (1 - \beta_k) \begin{bmatrix} h_i^n \\ u_e^n \end{bmatrix} + \beta_k \begin{bmatrix} \overline{h}_i^{n+1} \\ \overline{u}_e^{n+1} \end{bmatrix}, \tag{35}$$

for all $i \in \mathcal{C}_P^{\text{IF-L1}}$ and $e \in \mathcal{C}_E^{\text{IF-L1}}$, where

$$\alpha_k = \frac{k}{M}, \quad \beta_k = \frac{k+1}{M}, \tag{36}$$

for $k = 0, 1, \ldots, M - 1$. Detailed derivation of these coefficients α_k and β_k is given in Appendix A.1. Remark that for SSP-RK2 the predictor (34)-(35) by Taylor expansion happens to be identical to a linear interpolant between the current and forward Euler solutions. However, the need for Taylor expansions will be clearly seen when extending to third order or even higher order SSP-RK algorithms. Note that the approximate values by (34)-(35) will be used to advance in time on 'fine' cells (and their edges) neighboring the interface cells (Step 2a).

1b) Compute the solution at stage 1 of SSP-RK2 on the interface-layer 2 with the coarse time step size: $\overline{h}_i^{n+1}$ for all $i \in \mathcal{C}_P^{\text{IF-L2}}$ and $\overline{u}_e^{n+1}$ for all $e \in \mathcal{C}_E^{\text{IF-L2}}$. Note that these values are needed to advance in time on 'coarse' internal cells (and their edges) neighboring the interface-layer 2 (Step 2b).

2) Advancing from t^n to t^{n+1} excluding the interface layers:

2a) For 'fine' cells/edges: at each intermediate time level $k = 0, 1, \ldots, M - 1$, compute the solution by SSP-RK2 with the fine time step size. Note that the evaluation of the fluxes only involves the solution on the interface-layer 1 computed at Step 1a) and in the fine region:

$$\begin{cases} \overline{h}_i^{n,k+1} = h_i^{n,k} + \dfrac{\Delta t^n}{M} \mathcal{F}_i(\boldsymbol{h}^{n,k}|_{\mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{F}_P}, \boldsymbol{U}^{n,k}|_{\mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{F}_E}), & \forall i \in \mathcal{F}_P, \\[4mm] \overline{u}_e^{n,k+1} = u_e^{n,k} + \dfrac{\Delta t^n}{M} \mathcal{G}_e(\boldsymbol{h}^{n,k}|_{\mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{F}_P}, \boldsymbol{U}^{n,k}|_{\mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{F}_E}), & \forall e \in \mathcal{F}_E, \end{cases} \tag{37}$$

and

$$\begin{cases} h_i^{n,k+1} = \dfrac{1}{2} h_i^{n,k} + \dfrac{1}{2}\left(\overline{h}_i^{n,k+1} + \dfrac{\Delta t^n}{M} \mathcal{F}_i(\overline{\boldsymbol{h}}^{n,k+1}|_{\mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{F}_P}, \overline{\boldsymbol{U}}^{n,k+1}|_{\mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{F}_E}) \right), & \forall i \in \mathcal{F}_P, \\[4mm] u_e^{n,k+1} = \dfrac{1}{2} u_e^{n,k} + \dfrac{1}{2}\left(\overline{u}_e^{n,k+1} + \dfrac{\Delta t^n}{M} \mathcal{G}_e(\overline{\boldsymbol{h}}^{n,k+1}|_{\mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{F}_P}, \overline{\boldsymbol{U}}^{n,k+1}|_{\mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{F}_E}) \right), & \forall e \in \mathcal{F}_E. \end{cases} \tag{38}$$

2b) For 'coarse' internal cells/edges: do similar calculations as SSP-RK2 with the coarse time step size. Note that only the solution on the interface-layer 2 computed at Step 1b) and in the internal coarse region is used to evaluate the fluxes.

3) Interface correction:

i) Stage 1:

$$
\begin{cases}
\widetilde{h}_i^{n+1} = h_i^n + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{F}_i(\boldsymbol{h}^{n,k}, \boldsymbol{U}^{n,k}), & \forall\, i \in \mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{C}_P^{\text{IF-L2}}, \\[4mm]
\widetilde{u}_e^{n+1} = u_e^n + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{G}_e(\boldsymbol{h}^{n,k}, \boldsymbol{U}^{n,k}), & \forall\, e \in \mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{C}_E^{\text{IF-L2}},
\end{cases}
\tag{39}
$$

in which

$$
\begin{cases}
h_i^{n,k} = h_i^n, & \text{if } i \in \mathcal{C}_P^{\text{IF-L2}} \cup \mathcal{C}_P^{\text{int}}, \\[2mm]
u_e^{n,k} = u_e^n, & \text{if } e \in \mathcal{C}_E^{\text{IF-L2}} \cup \mathcal{C}_E^{\text{int}},
\end{cases}
\quad \text{for} \quad k = 0, 1, \dots, M-1.
\tag{40}
$$

ii) Stage 2:

$$
\begin{cases}
h_i^{n+1} = \dfrac{1}{2} h_i^n + \dfrac{1}{2}\left(\widetilde{h}_i^{n+1} + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{F}_i(\overline{\boldsymbol{h}}^{n,k+1}, \overline{\boldsymbol{U}}^{n,k+1}) \right), & \forall\, i \in \mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{C}_P^{\text{IF-L2}}, \\[4mm]
u_e^{n+1} = \dfrac{1}{2} u_e^n + \dfrac{1}{2}\left(\widetilde{u}_e^{n+1} + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{G}_e(\overline{\boldsymbol{h}}^{n,k+1}, \overline{\boldsymbol{U}}^{n,k+1}) \right), & \forall\, e \in \mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{C}_E^{\text{IF-L2}},
\end{cases}
\tag{41}
$$

in which

$$
\begin{cases}
\overline{h}_i^{n,k+1} = \overline{h}_i^{n+1}, & \text{if } i \in \mathcal{C}_P^{\text{IF-L2}} \cup \mathcal{C}_P^{\text{int}}, \\[2mm]
\overline{u}_e^{n,k+1} = \overline{u}_e^{n+1}, & \text{if } e \in \mathcal{C}_P^{\text{IF-L2}} \cup \mathcal{C}_E^{\text{int}},
\end{cases}
\quad \text{for} \quad k = 0, 1, \dots, M-1.
\tag{42}
$$

It is easy to see that Steps 2a) and 2b) are the most time-consuming part of the above LTS algorithm and can be performed in parallel.

Remark 4.1. *Differently from Trahan and Dawson's LTS scheme [47] which is second order accurate in time away from the LTS interface but reduces to first order at the interface (thus destroy the overall second order temporal accuracy), the proposed second order LTS scheme here preserves the second order accuracy on the whole domain, even at the region of the LTS interface. This is obtained by the prediction of the interface solution at both stages of SSP-RK2 for each intermediate time level. In addition, our scheme has been derived in such a way that the LTS scheme recovers the corresponding global time-stepping scheme when $M = 1$, which is not the case for Trahan and Dawson's LTS scheme.*

4.2.2. Third order LTS algorithm

The third order LTS algorithm based on SSP-RK3 is constructed in a similar way to the second order one. In particular, for each $n = 0, 1, \dots, N-1$, we perform a three-step algorithm of predictor-corrector type as follows:

1) Interface prediction:

1a) First compute the values at stage 1 and stage 2 of SSP-RK3 (30)-(31) with the coarse time step size on the interface-layer 1: $\overline{h}_i^{n+1}, \overline{h}_i^{n+1/2}$ for $i \in \mathcal{C}_P^{\text{IF-L1}}$ and $\overline{u}_e^{n+1}, \overline{u}_e^{n+1/2}$ for

$e \in \mathcal{C}_E^{\text{IF-L1}}$. Then use them to predict the values on interface cells/edges at intermediate time levels based on the second-order Taylor expansion as follows:

$$\begin{bmatrix} h_i^{n,k} \\ u_e^{n,k} \end{bmatrix} = (1 - \alpha_k - \widehat{\alpha}_k) \begin{bmatrix} h_i^n \\ u_e^n \end{bmatrix} + (\alpha_k - \widehat{\alpha}_k) \begin{bmatrix} \overline{h}_i^{n+1} \\ \overline{u}_e^{n+1} \end{bmatrix} + 2\widehat{\alpha}_k \begin{bmatrix} \overline{h}_i^{n+1/2} \\ \overline{u}_e^{n+1/2} \end{bmatrix}, \quad (43)$$

$$\begin{bmatrix} \overline{h}_i^{n,k+1} \\ \overline{u}_e^{n,k+1} \end{bmatrix} = (1 - \beta_k - \widehat{\beta}_k) \begin{bmatrix} h_i^n \\ u_e^n \end{bmatrix} + (\beta_k - \widehat{\beta}_k) \begin{bmatrix} \overline{h}_i^{n+1} \\ \overline{u}_e^{n+1} \end{bmatrix} + 2\widehat{\beta}_k \begin{bmatrix} \overline{h}_i^{n+1/2} \\ \overline{u}_e^{n+1/2} \end{bmatrix}, \quad (44)$$

$$\begin{bmatrix} \overline{h}_i^{n,k+1/2} \\ \overline{u}_e^{n,k+1/2} \end{bmatrix} = (1 - \gamma_k - \widehat{\gamma}_k) \begin{bmatrix} h_i^n \\ u_e^n \end{bmatrix} + (\gamma_k - \widehat{\gamma}_k) \begin{bmatrix} \overline{h}_i^{n+1} \\ \overline{u}_e^{n+1} \end{bmatrix} + 2\widehat{\gamma}_k \begin{bmatrix} \overline{h}_i^{n+1/2} \\ \overline{u}_e^{n+1/2} \end{bmatrix}, \quad (45)$$

for all $i \in \mathcal{C}_P^{\text{IF-L1}}$ and $e \in \mathcal{C}_E^{\text{IF-L1}}$, where α_k and β_k are the same as in (36) and

$$\widehat{\alpha}_k = \frac{k^2}{M^2}, \quad \widehat{\beta}_k = \frac{k(k+2)}{M^2}, \quad \gamma_k = \frac{2k+1}{2M}, \quad \widehat{\gamma}_k = \frac{2k^2 + 2k + 1}{2M^2}, \quad (46)$$

for $k = 0, 1, \ldots, M - 1$. See Appendix A.2 for detailed derivations of these coefficients.

1b) Compute the solution at stage 1 and stage 2 of SSP-RK3 at the interface-layer 2 with the coarse time step size: $\overline{h}_i^{n+1}, \overline{h}_i^{n+1/2}$ for all $i \in \mathcal{C}_P^{\text{IF-L2}}$ and $\overline{u}_e^{n+1}, \overline{u}_e^{n+1/2}$ for all $e \in \mathcal{C}_E^{\text{IF-L2}}$.

2) Advancing from t^n to t^{n+1} excluding the interface layers:

2a) For 'fine' cells/edges: at each intermediate time level $k = 0, 1, \ldots, M - 1$, compute the solution by SSP-RK3 with the fine time step size. The evaluation of the fluxes again only involves the solution on the interface-layer 1 computed at Step 1a) and in the fine region.

2b) For 'coarse' internal cells/edges: do similar calculations as SSP-RK3 with the coarse time step size. Again only the solution on the interface-layer 2 computed at Step 1b) and in the internal coarse region is used to evaluate the fluxes.

3) Interface correction:

i) Stage 1:

$$\begin{cases} \widetilde{h}_i^{n+1} = h_i^n + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{F}_i(\boldsymbol{h}^{n,k}, \boldsymbol{U}^{n,k}), & \forall\, i \in \mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{C}_P^{\text{IF-L2}}, \\[1.2em] \widetilde{u}_e^{n+1} = u_e^n + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{G}_e(\boldsymbol{h}^{n,k}, \boldsymbol{U}^{n,k}), & \forall\, e \in \mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{C}_E^{\text{IF-L2}}, \end{cases} \quad (47)$$

in which

$$\begin{cases} h_i^{n,k} = h_i^n, & \text{if } i \in \mathcal{C}_P^{\text{IF-L2}} \cup \mathcal{C}_P^{\text{int}}, \\ u_e^{n,k} = u_e^n, & \text{if } e \in \mathcal{C}_E^{\text{IF-L2}} \cup \mathcal{C}_E^{\text{int}}, \end{cases} \quad \text{for} \quad k = 0, 1, \ldots, M - 1. \quad (48)$$

ii) Stage 2:

$$\begin{cases} \widetilde{h}_i^{n+1/2} = \dfrac{3}{4} h_i^n + \dfrac{1}{4} \left(\widetilde{h}_i^{n+1} + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{F}_i(\overline{\boldsymbol{h}}^{n,k+1}, \overline{\boldsymbol{U}}^{n,k+1}) \right), & \forall\, i \in \mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{C}_P^{\text{IF-L2}}, \\[1.4em] \widetilde{h}_e^{n+1/2} = \dfrac{3}{4} u_e^n + \dfrac{1}{4} \left(\widetilde{u}_e^{n+1} + \dfrac{\Delta t^n}{M} \displaystyle\sum_{k=0}^{M-1} \mathcal{G}_e(\overline{\boldsymbol{h}}^{n,k+1}, \overline{\boldsymbol{U}}^{n,k+1}) \right), & \forall\, e \in \mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{C}_E^{\text{IF-L2}}, \end{cases}$$
$$(49)$$

15

in which
$$\begin{cases} \overline{h}_i^{n,k+1} = \overline{h}_i^{n+1}, & \text{if } i \in \mathcal{C}_P^{\text{IF-L2}} \cup \mathcal{C}_P^{\text{int}}, \\ \overline{u}_e^{n,k+1} = \overline{u}_e^{n+1}, & \text{if } e \in \mathcal{C}_E^{\text{IF-L2}} \cup \mathcal{C}_E^{\text{int}}, \end{cases} \quad \text{for} \quad k = 0, 1, \dots, M-1. \quad (50)$$

iii) Stage 3:

$$\begin{cases} h_i^{n+1} = \dfrac{1}{3} h_i^n + \dfrac{2}{3} \left(\widetilde{h}_i^{n+1/2} + \dfrac{\Delta t^n}{M} \sum_{k=0}^{M-1} \mathcal{F}_i(\overline{\boldsymbol{h}}^{n,k+1/2}, \overline{\boldsymbol{U}}^{n,k+1/2}) \right), & \forall i \in \mathcal{C}_P^{\text{IF-L1}} \cup \mathcal{C}_P^{\text{IF-L2}}, \\ u_e^{n+1} = \dfrac{1}{3} u_e^n + \dfrac{2}{3} \left(\widetilde{h}_e^{n+1/2} + \dfrac{\Delta t^n}{M} \sum_{k=0}^{M-1} \mathcal{G}_e(\overline{\boldsymbol{h}}^{n,k+1/2}, \overline{\boldsymbol{U}}^{n,k+1/2}) \right), & \forall e \in \mathcal{C}_E^{\text{IF-L1}} \cup \mathcal{C}_E^{\text{IF-L2}}, \end{cases}$$
$$(51)$$

in which
$$\begin{cases} \overline{h}_i^{n,k+1/2} = \overline{h}_i^{n+1/2}, & \text{if } i \in \mathcal{C}_P^{\text{IF-L2}} \cup \mathcal{C}_P^{\text{int}}, \\ \overline{u}_e^{n,k+1/2} = \overline{u}_e^{n+1/2}, & \text{if } e \in \mathcal{C}_E^{\text{IF-L2}} \cup \mathcal{C}_E^{\text{int}}, \end{cases} \quad \text{for} \quad k = 0, 1, \dots, M-1. \quad (52)$$

Remark 4.2. *It is noteworthy that for the explicit first order time-stepping method (the forward Euler method), the produced LTS algorithm by following our construction idea, would be identical to Osher and Sanders' LTS scheme [29] in which the predictor is obtained by freezing the solution at t^n (i.e., the zero-order Taylor series expansion):*

$$h_i^{n,k} = h_i^n, \quad u_e^{n,k} = h_e^n, \quad \text{for all } i \in \mathcal{C}_P^{IF\text{-}L1}, \; e \in \mathcal{C}_E^{IF\text{-}L1}, \; k = 0, 1, \dots, M-1. \quad (53)$$

Thus we have proposed a unified approach to construct LTS algorithms based on the SSP-RK methods up to third order accuracy. Extension to higher order LTS schemes can be done similarly within this framework by performing higher order Taylor series approximations of the predictor [20]. It should also be noted that the predictor for SSP-RK2 is monotonic, but the predictor for SSP-RK3 is not and thus the TVD property of the third order LTS algorithm is not theoretically guaranteed. On the other hand, numerical results presented in Section 5 show that the SSP-RK3 based LTS algorithm is absolutely stable while the SSP-RK2 based LTS algorithm is not; in particular, the third-order LTS method can ensure stable energy conservation for long time horizons with time step sizes independent of the time horizons.

Remark 4.3. *Extension of the proposed LTS schemes to the case of more than two zones is possible under one condition: for any two neighboring subdomains, the time step size in one subdomain must be an integer multiple of the time step size in the other subdomain.*

4.3. Conservation of mass and energy

It has been shown in [35] that the semidiscrete solution (12) by the TRiSK scheme satisfies conservation of total energy for two reasons: firstly the nonlinear Coriolis force does not create or destroy kinetic energy and secondly the exchange between kinetic and potential energy is conservative. The proposed LTS algorithms for solving (12) inherit all the properties, including the conservation of total energy, of the TRiSK scheme. That is to say they conserve the total energy within time truncation errors. In addition, we have the following result on mass conservation.

Theorem 4.4. *The proposed LTS schemes exhibit exact conservation of the mass.*

16

PROOF. We present the proof of mass conservation for the second order LTS scheme; for the third order scheme, the proof can be done similarly. The TRiSK spatial discretization is mass conserving, thus we only need to show that mass is conserved in the region of the two interface layers. Toward this end, let P_{i_1} and P_{i_2} be two neighboring primal cells such that $i_1 \in \mathcal{C}_P^{\text{IF-L1}}$ and $i_2 \in \mathcal{F}_P$. Denote by e_{12} the common edge of P_{i_1} and P_{i_2}. Assume that no-flux is imposed on the boundary of P_{i_1} and P_{i_2}, except at e_{12}. By the definition of the discrete divergence operator (13), we have that

$$\mathcal{F}_i(\boldsymbol{h}^{n,k}, \boldsymbol{U}^{n,k}) = -[\nabla \cdot F_e^{n,k}]_i = -\frac{1}{A_i} n_{e,i} F_e^{n,k} l_e$$

and

$$\mathcal{F}_i(\overline{\boldsymbol{h}}^{n,k+1}, \overline{\boldsymbol{U}}^{n,k+1}) = -\frac{1}{A_i} n_{e,i} \overline{F}_e^{n,k+1} l_e,$$

for $i \in \{i_1, i_2\}$ and $k = 0, 1, \ldots, M - 1$. As $i_1 \in \mathcal{C}_P^{\text{IF-L1}}$, it follows from equations (39) and (41) that

$$h_{i_1}^{n+1} = \frac{1}{2} h_{i_1}^n + \frac{1}{2} \left(\widetilde{h}_{i_1}^{n+1} - \frac{\Delta t^n}{M} \sum_{k=0}^{M-1} \frac{1}{A_{i_1}} n_{e,i_1} \overline{F}_e^{n,k+1} l_e \right)$$

$$= \frac{1}{2} \left(h_{i_1}^n + h_{i_1}^n - \frac{\Delta t^n}{M} \sum_{k=0}^{M-1} \frac{1}{A_{i_1}} n_{e,i_1} F_e^{n,k} l_e - \frac{\Delta t^n}{M} \sum_{k=0}^{M-1} \frac{1}{A_{i_1}} n_{e,i_1} \overline{F}_e^{n,k+1} l_e \right)$$

$$= h_{i_1}^n - \left(\frac{\Delta t^n}{M} \right) \frac{1}{A_{i_1}} n_{e,i_1} l_e \sum_{k=0}^{M-1} \left(\frac{F_e^{n,k} + \overline{F}_e^{n,k+1}}{2} \right). \tag{54}$$

On the other hand, as $i_2 \in \mathcal{F}_P$, one can use (37) and (38) to obtain

$$h_{i_2}^{n+1} = h_{i_2}^{n,M} = \frac{1}{2} h_{i_2}^{n,M-1} + \frac{1}{2} \left(\overline{h}_{i_2}^{n,M} - \left(\frac{\Delta t^n}{M} \right) \frac{1}{A_{i_2}} n_{e,i_2} \overline{F}_e^{n,M} l_e \right)$$

$$= h_{i_2}^{n,M-1} - \left(\frac{\Delta t^n}{M} \right) \frac{1}{A_{i_2}} n_{e,i_2} l_e \left(\frac{F_e^{n,M-1} + \overline{F}_e^{n,M}}{2} \right).$$

Solving the above recursion yields

$$h_{i_2}^{n+1} = h_{i_2}^n - \left(\frac{\Delta t^n}{M} \right) \frac{1}{A_{i_2}} n_{e,i_2} l_e \sum_{k=0}^{M-1} \left(\frac{F_e^{n,k} + \overline{F}_e^{n,k+1}}{2} \right). \tag{55}$$

From (54) and (55) and the fact that $n_{e,i_1} = -n_{e,i_2}$ (cf. Equation (9)), we deduce that

$$\int_{A_{i_1} \cup A_{i_2}} h^{n+1} = \int_{A_{i_1} \cup A_{i_2}} h^n. \tag{56}$$

This also holds for any $i_1 \in \mathcal{C}_P^{\text{IF-L1}}$ and $i_2 \in \mathcal{C}_P^{\text{IF-L2}}$ such that P_{i_1} and P_{i_2} be two neighboring primal cells. The associated fluid height $h_{i_1}^{n+1}$ and $h_{i_2}^{n+1}$ can be derived as (54). The proof is then completed.

Remark 4.5. *In the TRiSK scheme [35], the spatially discrete PV equation is obtained by applying the discrete curl operator to the spatially discrete momentum equation (12) so that the resulting*

discrete PV is compatible with the discrete velocity field for all time. In addition, it has been shown in [35] that the discrete PV equation is consistent with the underlying discrete thickness equation. Using those two properties of the discrete PV and by following the same argument as in the proof of Theorem 4.4 - but using the discrete momentum equation (instead of the discrete thickness equation), one can show that the proposed LTS schemes also exhibits exact conservation of PV.

5. Numerical experiments

We carry out numerical experiments using the simplified SOMA Double-Gyre test cases [49] in 1D and 2D and the shallow water test case 5 (SWTC5) [48] to evaluate the performance of the proposed SSP-RK based LTS schemes. In Subsection 5.1, we first consider the 1D shallow water equations and describe the 1D counterpart of the TRiSK scheme as the spatial discretization of the problem. Then we present numerical results for the 1D SOMA test case which show the accuracy, order of convergence and conservation properties of LTS solutions in comparison with those of global time-stepping solutions. Results for 2D problems are given in Subsection 5.2 for the simplified SOMA test case with a uniform mesh and in Subsection 5.3 for the SWTC5 with a multi-resolution mesh of the sphere. Subsection 5.4 presents the testing results on the parallel scalability of the proposed LTS algorithms.

5.1. 1D shallow water equations

The 1D shallow water equations are written as

$$\partial_t h + \partial_x(hu) = 0, \qquad \text{on } (-L, L) \times (0, T), \qquad (57)$$

$$\partial_t u + \partial_x \left(\frac{u^2}{2} + g(h + b(x)) \right) = 0, \qquad \text{on } (-L, L) \times (0, T), \qquad (58)$$

for some $L > 0$. The system is complemented by the zero-Dirichlet boundary conditions:

$$u(-L, t) = 0, \quad u(L, t) = 0, \quad \forall t \in (0, T). \qquad (59)$$

and the initial conditions h_0 and u_0. Consider a partition of $\Omega = (0, L)$ into $N + 1$ grid points:

$$0 = x_{1/2} < x_{3/2} < \ldots < x_{N+1/2} = L,$$

and denote by x_i the midpoint of $[x_{i-1/2}, x_{i+1/2}]$, $i = 1, \ldots, N$. In addition, we make use of the following notation:

$$\Delta x_i = x_{i+1/2} - x_{i-1/2}, \quad \forall i = 1, \ldots, N, \qquad \Delta x_{i+1/2} = x_{i+1} - x_i, \quad \forall i = 1, \ldots, N - 1.$$

The TRiSK scheme in 1D space reduces to a staggered central difference scheme in which h is approximated at the midpoints x_i, and u at the nodal points $x_{i+1/2}$:

$$h_i(t) \approx h(x_i, t), \quad \forall i = 1, \ldots N,$$
$$u_{i+1/2}(t) \approx u(x_{i+1/2}, t), \quad \forall i = 0, \ldots, N.$$

Define the following average values:

$$\hat{h}_{i+1/2}(t) \; := \; \frac{1}{2} \left(h_{i+1}(t) + h_i(t) \right), \quad \forall i = 1, \ldots, N - 1, \quad \forall t \in (0, T),$$

$$\hat{u}_i^2(t) \; := \; \frac{1}{2\Delta x_i} \left(\Delta x_{i+1/2} \, u_{i+1/2}^2(t) + \Delta x_{i-1/2} \, u_{i-1/2}^2(t) \right), \quad \forall i = 1, \ldots, N, \; \forall t \in (0, T).$$

$$(60)$$

Using central differences, the semi-discretized problem of (57)-(58) then can be written as

$$
\begin{cases}
h'_i(t) &= -\dfrac{1}{\Delta x_i}\left(u_{i+1/2}(t)\hat{h}_{i+1/2}(t) - u_{i-1/2}(t)\hat{h}_{i-1/2}(t)\right),\ \forall\, i = 1,\ldots,N,\ \forall\, t \in (0,T),\\[2mm]
u'_{i+1/2}(t) &= -\dfrac{1}{\Delta x_{i+1/2}}\left[\dfrac{\hat{u}^2_{i+1}(t) - \hat{u}^2_i(t)}{2} + g\left((h_{i+1}(t) + b(x_{i+1})) - (h_i(t) + b(x_i))\right)\right],\\[2mm]
& \hspace{4cm} \forall\, i = 1,\ldots,N-1,\ \forall\, t \in (0,T),\\[2mm]
u_{1/2}(t) &= u_{N+1/2}(t) = 0,\quad \forall\, t \in (0,T),\\[1mm]
u_{i+1/2}(0) &= u_0(x_{i+1/2}),\quad \forall\, i = 0,\ldots,N,\\[1mm]
h_i(0) &= h_0(x_i),\quad \forall\, i = 1,\ldots,N.
\end{cases}
\tag{61}
$$

Note that the above system (61) clearly can be recast as an autonomous system of the variable vector $\left(h_1,\ldots,h_N,u_{1+1/2},\ldots,u_{N-1/2}\right)^\top \in \mathbb{R}^{2N-1}$.

For numerical experiments, we consider a 1D simplified version of the SOMA Double-Gyre test case [49]. The domain $\Omega = [-L, L]$ with $L = 1465.7$km. Denote by $R = 6371$km, the radius of the Earth and let $\mathbf{x}_c = (0, R)$ be a reference point on a circle with center point $(0,0)$ and radius R. For each point $\mathbf{x} = (x, \sqrt{R^2 - x^2})$ on the circle with $x \in \Omega$, define the function

$$
s(x) = 1 - \frac{d(x)}{\lambda^2},\quad d(x) = R\arccos(\mathbf{x}\cdot\mathbf{x}_c),\quad x \in \Omega.
\tag{62}
$$

The bathymetry (see Figure 4) describing the bottom topography is given by

$$
b(x) = -H_{\text{shelf}} - \frac{H_0}{2}\left[1 + \tanh\left(\frac{s}{\omega}\right)\right],\quad \forall\, \kappa > \gamma.
\tag{63}
$$

Parameter choices of $\lambda = 1250$km, $H_{\text{shelf}} = 0.1$km, $H_0 = 2.4$km, $\omega = 0.1$ and $\gamma = -0.4$ describe an ocean basin with a radius of L and a "continental shelf" that is approximately 150km wide along all margins. We consider a Gaussian (wave-like) initial condition for h (see Figure 4) and the zero initial condition for u.

Figure 4: (1D SOMA) The bottom topography of the ocean between two continental shelf (left) and the initial fluid height $dh_0 = h_0 + b$ (right).

In space, we utilize a uniform partition of $\Omega = [-L, L]$ with $N = 513$ nodes and $\Delta x = 5.725$km. Explicit time schemes require the time step size to at least satisfy the CFL condition which is governed by the maximal wave speed:

$$
\Delta t \le \frac{\Delta x}{\nu} = \alpha,\quad \nu = \max_i\left(|\hat{u}_i \pm \sqrt{gh_i}|\right).
\tag{64}
$$

For the LTS schemes, we consider a decomposition into 3 subdomains, $\Omega_1 = (-1465.7\text{km}, -600\text{km})$, $\Omega_2 = (-600\text{km}, 600\text{km})$ and $\Omega_3 = (600\text{km}, 1465.7\text{km})$. A coarse time step size is used in Ω_1 and

Ω_3: $\Delta t_1 = \Delta t_3 = \Delta t_{\text{coarse}}$, while a fine time step size is used in Ω_2: $\Delta t_2 = \Delta t_{\text{fine}}$ with

$$\Delta t_{\text{fine}} = \frac{\Delta t_{\text{coarse}}}{M}, \quad \text{for some positive integer } M. \tag{65}$$

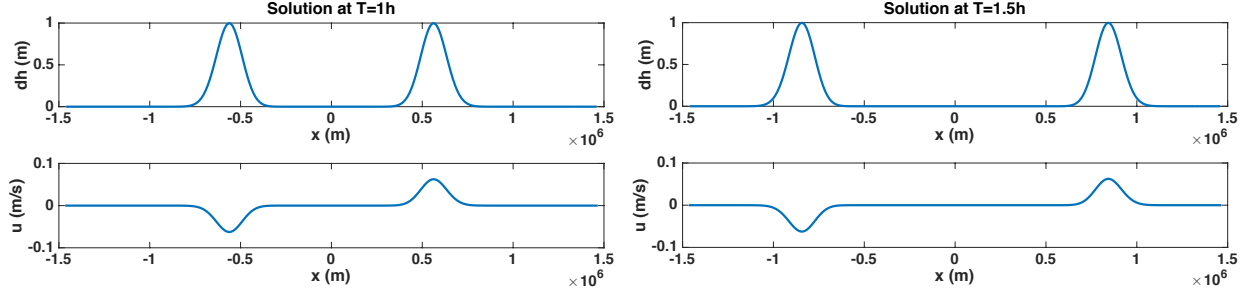The snapshots of the solution at $T = 1$ hour and $T = 1.5$ hours are shown in Figure 5.

Figure 5: (1D SOMA) Snapshots of the fluid height and velocity at $T = 1$ hour (left) and $T = 1.5$ hours (right).

We first investigate the convergence order in time of the proposed second and third order LTS algorithms and their effectiveness in terms of accuracy with different M. Toward this end, we fix the coarse time step size Δt_{coarse} and vary the fine time step size, $\Delta t_{\text{fine}} = \Delta t_{\text{coarse}}/M$, with $M \in \{1, 2, 4, 8\}$. Note that $M = 1$ corresponds exactly the global uniform time-stepping with Δt_{coarse}. We compute the L^2 relative errors at $T = 1$ hour and $T = 1.5$ hours based on the classic RK4 reference solution obtained with a very fine time step size $\Delta t_{\text{ref}} = 0.001\alpha$. The errors are expected to decrease as M increases. This is confirmed by the results given in Tables 4 and 5 for the second and third order LTS algorithms respectively. Note that at $T = 1$ hour, the wave fronts reach the LTS fine-coarse interface regions while at $T = 1.5$ hours, they completely move out of the interface regions (see Figure 5). Thus it is easy to see that at $T = 1.5$ hours, using large M does not improve very much the accuracy. On the other hand, for M fixed, it is observed that the proposed LTS schemes achieve perfectly the same orders of temporal accuracy on the whole domain as their corresponding global schemes.

Next we verify the evolution of the total energy and mass for a longer time, $T = 10$ days. We fix $M = 4$ and run the simulation using the first, second and third order LTS schemes with different time step sizes: for SSP-RK1, $\Delta t_{\text{coarse}} = 0.0025\alpha$; for SSP-RK2, $\Delta t_{\text{coarse}} \in \{0.125\alpha, 0.0625\alpha\}$ and for SSP-RK3, $\Delta t_{\text{coarse}} = 0.5\alpha$. In Figure 6, we plot the relative change of total energy as a function of time and it is observed that the first and second order LTS schemes fail (the total energy blows up due to oscillatory solutions) if the time step sizes are not sufficiently small while for the third order scheme it is stable even with a much larger time step size. This confirms the need of using higher order time-stepping methods to obtain better stability, especially for long time simulations. In Figure 7, we plot the evolution of the relative changes of total energy and mass of the second order LTS scheme with $\Delta t_{\text{coarse}} = 0.0625\alpha$ and of the third order scheme with $\Delta t_{\text{coarse}} \in \{0.125\alpha, 0.0625\alpha\}$. It is easy to find that the LTS schemes conserve the energy to within time truncation errors; in particular, for fixed $\Delta t_{\text{coarse}} = 0.0625\alpha$, the total energy computed by the third order LTS scheme is much closer to the initial total energy compared to the energy by the second order LTS scheme. In addition, it is also easy to verify both LTS schemes conserve mass numerically close to machine precision along the time (mostly caused by accumulated round-off errors).

SSP-RK2 based LTS algorithm for 1D SOMA test case									
Δt_{coarse}	M	\multicolumn{4}{c}{T=1 hour}				\multicolumn{4}{c}{T=1.5 hours}
		h	[CR]	$\boldsymbol{u}$	[CR]	h	[CR]	$\boldsymbol{u}$	[CR]
0.5α	1	2.97e-03	–	2.97e-03	–	6.72e-03	–	1.23e-02	–
	2	7.91e-04	–	7.91e-04	–	2.09e-03	–	2.09e-03	–
	4	2.77e-04	–	2.75e-04	–	1.51e-03	–	1.51e-03	–
	8	1.82e-04	–	1.80e-04	–	1.37e-03	–	1.37e-03	–
0.25α	1	7.43e-04	[2.00]	7.43e-04	[2.00]	1.11e-03	[2.60]	1.11e-03	[3.47]
	2	1.98e-04	[2.00]	1.98e-04	[2.00]	5.23e-04	[2.00]	5.23e-04	[2.00]
	4	6.92e-05	[2.00]	6.90e-05	[2.00]	3.80e-04	[1.99]	3.80e-04	[1.99]
	8	4.57e-05	[1.99]	4.52e-05	[1.99]	3.45e-04	[1.99]	3.45e-04	[1.99]
0.125α	1	1.86e-04	[2.00]	1.86e-04	[2.00]	2.79e-04	[1.99]	2.79e-04	[1.99]
	2	4.94e-05	[2.00]	4.94e-05	[2.00]	1.31e-04	[2.00]	1.31e-04	[2.00]
	4	1.73e-05	[2.00]	1.73e-05	[2.00]	9.51e-05	[2.00]	9.51e-05	[2.00]
	8	1.14e-05	[2.00]	1.13e-05	[2.00]	8.64e-05	[2.00]	8.64e-05	[2.00]
0.0625α	1	4.65e-05	[2.00]	4.65e-05	[2.00]	6.97e-05	[2.00]	6.97e-05	[2.00]
	2	1.23e-05	[2.00]	1.24e-05	[2.00]	3.27e-05	[2.00]	3.27e-05	[2.00]
	4	4.32e-06	[2.00]	4.31e-06	[1.99]	2.38e-05	[2.00]	2.38e-05	[2.00]
	8	2.86e-06	[2.00]	2.84e-06	[2.00]	2.16e-05	[2.00]	2.16e-05	[2.00]

Table 4: (1D SOMA) L^2 relative errors in h and $\boldsymbol{u}$ at different final times T between the RK4 reference solution with the very fine time step size $\Delta t_{\text{ref}} = 0.001\alpha$ and the approximate solutions obtained by the second order LTS algorithm. The rates of convergence "CR" for fixed M are shown in square brackets.
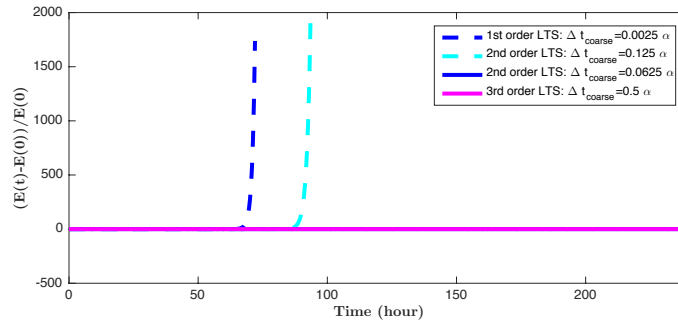

Figure 6: (1D SOMA) The relative change of total energy as a function of time computed respectively by the first order LTS scheme with $\Delta t_{\text{coarse}} = 0.0025\alpha$, by the second order LTS scheme with $\Delta t_{\text{coarse}} \in \{0.125\alpha, 0.0625\alpha\}$ and by the third order LTS scheme with $\Delta t_{\text{coarse}} = 0.5\alpha$. $M = 4$ for all cases.

5.2. Shallow water equations in planar region: the 2D SOMA test case

The LTS schemes are again tested numerically with a simplified version of the 2D SOMA Double-Gyre test case [49], where the geophysical domain and bathymetry are used, but only the shallow water equation (1)-(2) is considered, and neither wind nor bottom stress is applied. The spatial domain Ω is a circle of radius L with $L = 1465.7$km. The ocean basin is defined on the surface of the sphere, in particular, the reference point $\mathbf{x}_c = (x_c, y_c, z_c)$ is located at (35°N, 0°N) and the bathymetry $b(x, y)$ is given similarly to (62)-(63). Again, we consider a Gaussian (wave-like) initial

SSP-RK3 based LTS algorithm for 1D SOMA test case								
$\Delta t_{\mathrm{coarse}}$	M	T=1 hour				T=1.5 hours			
		h	[CR]	$\boldsymbol{u}$	[CR]	h	[CR]	$\boldsymbol{u}$	[CR]
0.5α	1	5.62e-05	–	5.62e-05	–	8.41e-05	–	8.41e-05	–
	2	8.29e-06	–	8.31e-06	–	3.22e-05	–	3.22e-05	–
	4	3.60e-06	–	3.53e-06	–	2.59e-05	–	2.59e-05	–
	8	3.34e-06	–	3.24e-06	–	2.52e-05	–	2.52e-05	–
0.25α	1	7.02e-06	[3.00]	7.02e-06	[3.00]	1.05e-05	[3.00]	1.05e-05	[3.00]
	2	1.04e-06	[3.00]	1.04e-06	[3.00]	4.04e-06	[3.00]	4.04e-06	[3.00]
	4	4.50e-07	[3.00]	4.45e-07	[2.99]	3.26e-06	[2.99]	3.26e-06	[2.99]
	8	4.17e-07	[3.00]	4.10e-07	[2.98]	3.17e-06	[2.99]	3.17e-06	[2.99]
0.125α	1	8.78e-07	[3.00]	8.78e-07	[3.00]	1.32e-06	[2.99]	1.32e-06	[2.99]
	2	1.30e-07	[3.00]	1.30e-07	[3.00]	5.05e-07	[3.00]	5.05e-07	[3.00]
	4	5.63e-08	[3.00]	5.58e-08	[3.00]	4.08e-07	[3.00]	4.08e-07	[3.00]
	8	5.22e-08	[3.00]	5.14e-08	[3.00]	3.96e-07	[3.00]	3.96e-07	[3.00]
0.0625α	1	1.10e-07	[3.00]	1.10e-07	[3.00]	1.65e-07	[3.00]	1.65e-07	[3.00]
	2	1.62e-08	[3.00]	1.62e-08	[3.00]	6.31e-08	[3.00]	6.31e-08	[3.00]
	4	7.04e-09	[3.00]	6.98e-09	[3.00]	5.10e-08	[3.00]	5.10e-08	[3.00]
	8	6.53e-09	[3.00]	6.44e-09	[3.00]	4.96e-08	[3.00]	4.96e-08	[3.00]

Table 5: (1D SOMA) L^2 relative errors in h and $\boldsymbol{u}$ at different final times T between the RK4 reference solution with the very fine time step size $\Delta t_{\mathrm{ref}} = 0.001\alpha$ and the approximate solutions obtained by the third order LTS algorithm. The rates of convergence "CR" for fixed M are shown in square brackets.
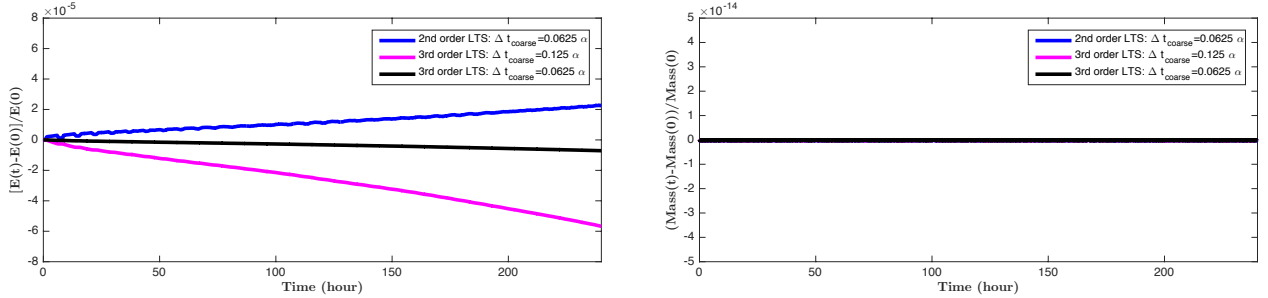

Figure 7: (1D SOMA) Evolution of the relative changes of total energy (left) and mass (right) for $T = 10$ days, computed respectively by the second order LTS scheme with $\Delta t_{\mathrm{coarse}} = 0.0625\alpha$ and by the third order scheme with $\Delta t_{\mathrm{coarse}} \in \{0.125\alpha, 0.0625\alpha\}$. $M = 4$ for all cases.

condition for h, depicted in Figure 8 (left), and the zero initial condition for u. A quasi-uniform mesh is used in this example and has $8,521$ cells. For the LTS schemes, the cells that are in the radius of 600km of the domain center are marked as fine cells, as is shown in blue in Figure 8 (right); the other cells are coarse cells. Smaller time step sizes Δt_{fine} are used in the fine cells, and larger time step sizes $\Delta t_{\mathrm{coarse}}$ in the coarse cells with $\Delta t_{\mathrm{coarse}} = M\Delta t_{\mathrm{fine}}$. The fluid height at different times is displayed in Figure 9.

To investigate the accuracy of LTS schemes, as in the 1D case we fix the coarse time step size $\Delta t_{\mathrm{coarse}}$ and vary $M \in \{1, 2, 4, 8\}$. Tables 6 and 7 show the relative errors in L^2-norm by the second
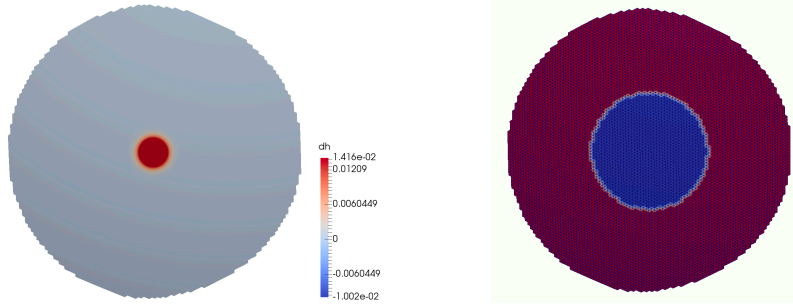
Figure 8: (2D SOMA) The initial fluid height $dh = h + b$ (left) and the LTS interface setting (right).
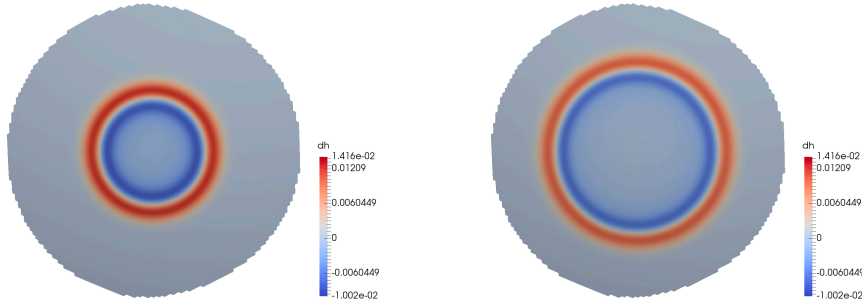

Figure 9: (2D SOMA) Snapshots of the fluid height at $T = 1$ hour (left) and $T = 1.5$ hours (right).

and third order LTS algorithms respectively. The reference solution is again given by the RK4 with the very fine time step size $\Delta t_{\text{ref}} = 0.001\alpha$. From these tables, we draw the same conclusions as in the 1D case, i.e., both algorithms reach their respective optimal temporal convergence perfectly, and the errors decrease as M increases. Finally, as shown in Figure 10 which presents the evolution of the relative changes of total energy, mass and potential vorticity for $T = 10$ days, it is observed that the third order scheme gives better energy conservation along the time than the second order scheme given the same time step size, and they conserve the mass and potential vorticity almost exactly along the time.

5.3. Shallow water equations on sphere: the SWTC5 test case

We further test our LTS schemes on the standard shallow water test case 5 (SWTC5) [48] on the whole sphere. On the sphere with a radius of $a = 6371.22$km, an isolated mountain is placed around the point with longitude and latitude $(\lambda_c, \theta_c) = (3\pi/2, \pi/6)$, with height as $h_s = h_{s_0}(1 - r/R)$, where $h_{s_0} = 2$km, $R = \pi/9$, $r^2 = \min\{R^2, (\lambda - \lambda_c)^2 + (\theta - \theta_c)^2\}$, and (λ, θ) is the latitude and longitude. The initial longitudinal and latitudinal components of velocity are $(u, v) = (u_0 \cos(\theta), 0)$, where $u_0 = 20$ms^{-1}, and the initial thickness is given by

$$h = h_0 - \frac{1}{g}(a\Omega u_0 + \frac{u_0^2}{2})(\sin(\theta))^2, \tag{66}$$

where $h_0 = 5.96$km, $\Omega = 7.292 \times 10^{-5}$s^{-1}, and $g = 9.80616$ms^{-2}.

The SWTC5 is of great interest since it has been shown in [33] that with grid refinement in certain area the solution has better convergence and details. However, the refined cells would restrict the time step size with uniform time-stepping while the proposed LTS algorithms would release such

SSP-RK2 based LTS algorithm for 2D SOMA test case									
Δt_{coarse}	M	T=1 hour				T=1.5 hours			
		h	[CR]	$\boldsymbol{u}$	[CR]	h	[CR]	$\boldsymbol{u}$	[CR]
0.5α	1	3.79e-02	–	3.54e-02	–	5.70e-02	–	5.33e-02	–
	2	9.96e-03	–	9.34e-03	–	2.55e-02	–	2.39e-02	–
	4	3.27e-03	–	3.12e-03	–	1.76e-02	–	1.65e-02	–
	8	1.96e-03	–	1.87e-03	–	1.56e-02	–	1.46e-02	–
0.25α	1	9.42e-03	[2.01]	8.80e-03	[2.01]	1.41e-02	[2.02]	1.32e-02	[2.01]
	2	2.48e-03	[2.01]	2.33e-03	[2.00]	6.35e-03	[2.01]	5.96e-03	[2.00]
	4	8.13e-04	[2.01]	7.77e-04	[2.01]	4.41e-03	[2.00]	4.14e-03	[1.99]
	8	4.92e-04	[1.99]	4.68e-04	[2.00]	3.91e-03	[2.00]	3.67e-03	[1.99]
0.125α	1	2.35e-03	[2.00]	2.20e-03	[2.00]	3.52e-03	[2.00]	3.30e-03	[2.00]
	2	6.19e-04	[2.00]	5.82e-04	[2.00]	1.59e-03	[2.00]	1.49e-03	[2.00]
	4	2.03e-04	[2.00]	1.94e-04	[2.00]	1.11e-03	[1.99]	1.04e-03	[1.99]
	8	1.23e-04	[2.00]	1.17e-04	[2.00]	9.81e-04	[1.99]	9.21e-04	[1.99]
0.0625α	1	5.88e-04	[2.00]	5.50e-04	[2.00]	8.81e-04	[2.00]	8.25e-04	[2.00]
	2	1.55e-04	[2.00]	1.45e-04	[2.00]	3.97e-04	[2.00]	3.72e-04	[2.00]
	4	5.06e-05	[2.00]	4.85e-05	[2.00]	2.76e-04	[2.01]	2.59e-04	[2.01]
	8	3.08e-05	[2.00]	2.92e-05	[2.00]	2.45e-04	[2.00]	2.30e-04	[2.00]

Table 6: (2D SOMA) L^2 relative errors in h and $\boldsymbol{u}$ at different final times T between the RK4 reference solution with the very fine time step size $\Delta t_{\text{ref}} = 0.001\alpha$ and the approximate solutions obtained by the second order LTS algorithm. The rates of convergence "CR" for fixed M are shown in square brackets.
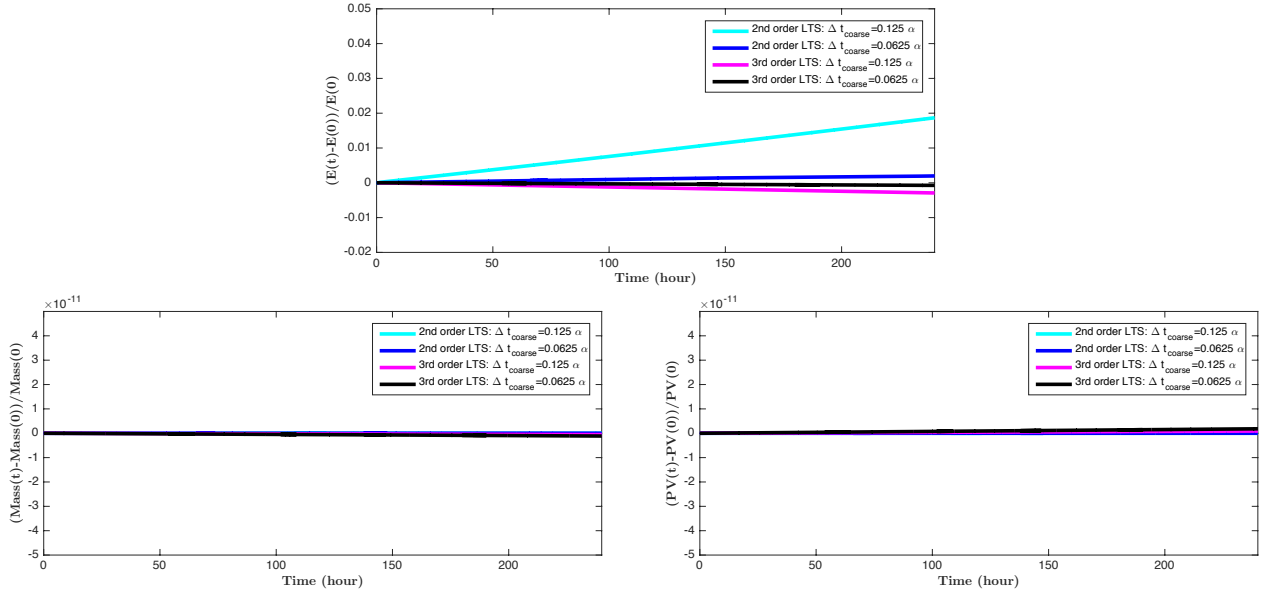

Figure 10: (2D SOMA) Evolution of the relative changes of total energy (top), mass (bottom-left) and potential vorticity (bottom-right) for $T = 10$ days, computed respectively by the second and third order schemes with $\Delta t_{\text{coarse}} \in \{0.125\alpha, 0.0625\alpha\}$. $M = 4$ for all cases.

		T=1 hour				T=1.5 hours			

| SSP-RK3 based LTS algorithm for 2D SOMA test case |

Δt_{coarse}	M	T=1 hour				T=1.5 hours			
		h	[CR]	$\boldsymbol{u}$	[CR]	h	[CR]	$\boldsymbol{u}$	[CR]
0.5α	1	1.95e-03	–	1.78e-03	–	2.75e-03	–	2.54e-03	–
	2	2.60e-04	–	2.32e-04	–	9.10e-04	–	8.42e-04	–
	4	8.84e-05	–	6.42e-05	–	6.96e-04	–	6.44e-04	–
	8	8.21e-05	–	5.69e-05	–	6.71e-04	–	6.20e-04	–
0.25α	1	2.45e-04	[2.99]	2.23e-04	[3.00]	3.46e-04	[2.99]	3.19e-04	[2.99]
	2	3.24e-05	[3.00]	2.90e-05	[3.00]	1.14e-04	[3.00]	1.05e-04	[3.00]
	4	1.07e-05	[3.05]	8.07e-06	[2.99]	8.74e-05	[2.99]	8.08e-05	[2.99]
	8	9.83e-06	[3.06]	7.21e-06	[2.98]	8.42e-05	[2.99]	7.78e-05	[2.99]
0.125α	1	3.06e-05	[3.00]	2.79e-05	[3.00]	4.33e-05	[3.00]	3.99e-05	[3.00]
	2	4.05e-06	[3.00]	3.62e-06	[3.00]	1.42e-05	[3.01]	1.32e-05	[2.99]
	4	1.31e-06	[3.03]	1.01e-06	[3.00]	1.09e-05	[3.00]	1.01e-05	[3.00]
	8	1.20e-06	[3.03]	9.06e-07	[2.99]	1.05e-05	[3.00]	9.75e-06	[3.00]
0.0625α	1	3.83e-06	[3.00]	3.49e-06	[3.00]	5.41e-06	[3.00]	4.99e-06	[3.00]
	2	5.06e-07	[3.00]	4.52e-07	[3.00]	1.78e-06	[3.00]	1.65e-06	[3.00]
	4	1.64e-07	[3.00]	1.28e-07	[2.98]	1.37e-06	[2.99]	1.27e-06	[2.99]
	8	1.55e-07	[2.95]	1.19e-07	[2.93]	1.32e-06	[2.99]	1.22e-06	[3.00]

Table 7: (2D SOMA) L^2 relative errors in h and $\boldsymbol{u}$ at different final times T between the RK4 reference solution with very fine time step size $\Delta t_{\text{ref}} = 0.001\alpha$ and the approximate solutions obtained by the third order LTS algorithm. The rates of convergence "CR" for fixed M are shown in square brackets.

issue. The orographic feature, depicted in Figure 11 (left), extends $\pi/9$ radians in latitude and longitude, thus a fine-cell region with $\pi/6$ radians is used to cover the orographic feature as in [33], and a variable-resolution SCVT mesh with 40,962 cells is generated, where the coarse cell size is approximately two times of the fine cell size (four times in term of cell area), as is shown in Figure 11 (middle). For time discretization, we apply the local time-stepping, in which a fine time-step region with $3\pi/10$ radians is used to cover the fine-cell region, as is shown in Figure 11 (right). The initial fluid height is depicted in Figure 12 (left).

We here focus the third order LTS scheme due to its nice stability for long time simulations. We compute the L^2 relative errors at the time $T = 1$ day between the approximate solution obtained by the third order LTS algorithm and the reference solution given by RK4 with the very fine time step size $\Delta t_{\text{ref}} = 0.001\alpha$. The results with fixed $M = 4$ reported in Table 8 confirm that the SSP-RK3 based LTS algorithm does obtain a temporal convergence of $O(\Delta t_{\text{coarse}}^3)$. Moreover, in this test example, the physical phenomena in fine-cell region is of interest, thus we use smaller time step sizes there by the way of increasing M (with fixed $\Delta t_{\text{coarse}} = 0.25\alpha$), to achieve better approximate solutions, as is shown in Table 8. It seems that $M = 4$ is a good choice since $M = 8$ just gives very slight improvement over $M = 4$. A simulation with the final time $T = 15$ days is also carried out by using the third order LTS scheme with $\Delta t_{coarse} = 0.5\alpha$ and $M = 4$. The final fluid height is shown in Figure 12 (right). Figure 13 presents the evolution of the relative changes of total energy, mass and potential vorticity. It is observed that the third order LTS scheme conserves the energy very well with tiny monotonic dissipations (up to about 2.93e-09 in the relative change in 15 days).
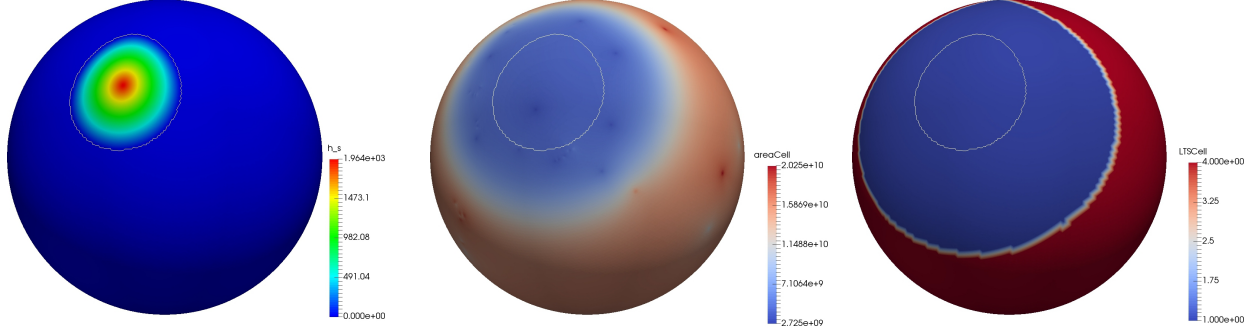
Figure 11: (SWTC5) Left: The bottom topography b in which the orographic feature (depicted by the white oval curve in the figure) extends $\pi/9$ radians in latitude and longitude; middle: the cell area of a variable-resolution SCVT mesh with 40,962 cells where the coarse cell size is approximately two times of the fine cell size; right: the LTS interface where a fine time-stepping region with $3\pi/10$ radians is used to cover the fine-cell region, the rest is coarse time-stepping region.
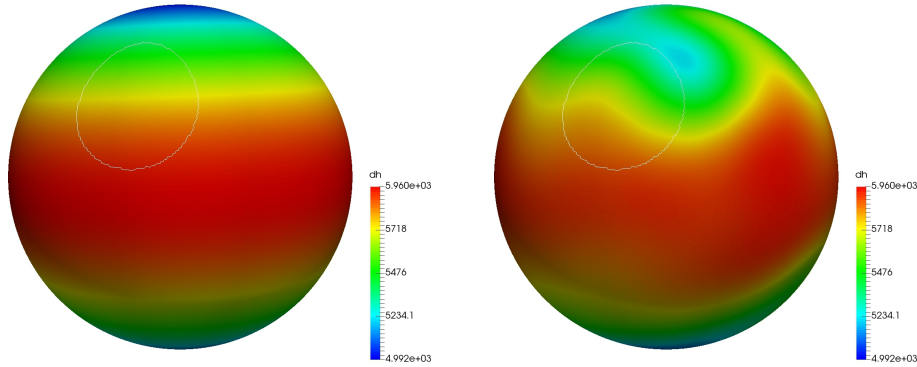

Figure 12: (SWTC5) The fluid height $dh = h + b$ at $T = 0$ (left) and at $T = 15$ days (right) measured in meters.

SSP-RK3 based LTS, M=4						SSP-RK3 based LTS, $\Delta t_{\text{coarse}} = 0.25\alpha$		
Δt_{coarse}	h	[CR]	$\boldsymbol{u}$	[CR]		M	h	$\boldsymbol{u}$
0.5α	3.38e-06	–	2.20e-05	–		1	1.69e-06	9.38e-06
0.25α	5.88e-07	[2.52]	3.27e-06	[2.75]		2	6.76e-07	3.68e-06
0.125α	7.80e-08	[2.91]	4.20e-07	[2.96]		4	5.95e-07	3.27e-06
0.0625α	1.24e-08	[2.85]	6.25e-08	[2.93]		8	5.88e-07	3.25e-06

Table 8: (SWTC5) L^2 relative errors in h and $\boldsymbol{u}$ between the RK4 reference solution with the very fine time step size $\Delta t_{\text{ref}} = 0.001\alpha$ and the approximate solutions obtained by the third order LTS algorithm at $T = 1$ day. The results are for fixed M (left) to verify the convergence order and for fixed Δt_{coarse} (right) to verify the efficiency of LTS.

5.4. Parallel scalability

The parallel scalability tests of the proposed SSP-RK based LTS algorithms are performed on the "LSSC-IV" cluster system located in the State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences, in which each node has 2 Intel Xeon Gold 6140 CPUs (2.3GHz, 18 cores) and 192GB memory and a 100GB EDR Infiniband network is used for data communication between nodes. Since different cells may have different time step sizes, the LTS
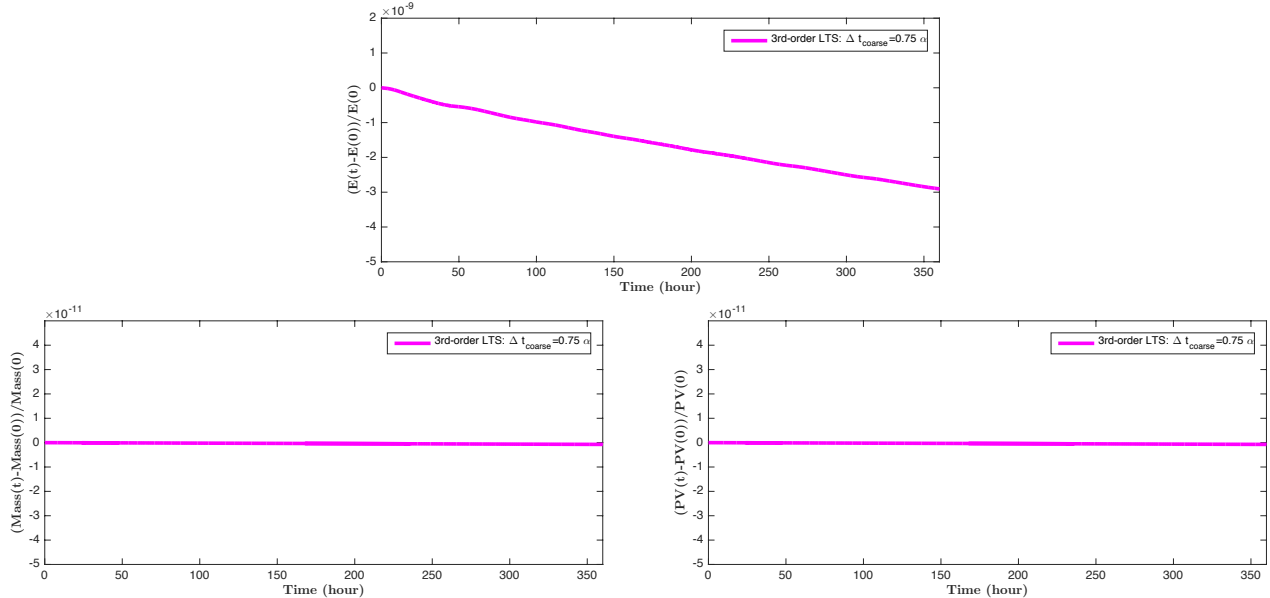
26

Figure 13: (SWTC5) Evolution of the relative changes of total energy (top), mass (bottom-left) and potential vorticity (bottom-right) for $T = 15$ days computed by the third order LTS with $\Delta t_{\text{coarse}} = 0.5\alpha$ and $M = 4$.

schemes usually bring serious computation/load imbalance problem across processors in parallel computing if quasi-uniform mesh partitions are used. To solve this problem, weighted partition tools and algorithms [23, 30, 31] have been widely used to distribute and balance the computations as much as possible among all processors/cores. The single-constraint weighted partitioning scheme based on the local time step sizes and "Metis" [23] are used for domain decomposition in our parallel implementation within the Message Passing Interface (MPI) environment. We test the scalability of the proposed SSP-RK based LTS algorithms with $\Delta t_{\text{coarse}} = 0.5\alpha$ and $M = 4$ for the SWTC5 test case with $T = 3$ hours. A sequence of three spatial meshes are used: the first one is the variable-resolution mesh with 40,962 cells in Subsection 5.3 and the other two are its consecutive uniform refinements which give us two meshes with 163,842 cells and 655,362 cells respectively. Results of the parallel scalability with different number of cores (up to 128 cores) are reported in Tables 9. It is observed that the two LTS algorithms have very similar performance in term of parallel scalability. As the subproblem size per core increases, the parallel speedup gets better as expected, and the improvements become more significant when the number of cores is large. Super-linear speedups are observed in the case of large size problems with small numbers of cores (probably due to the cache effect). With 128 cores, the SSP-RK2 based LTS algorithm achieves a parallel efficiency of 35.0% and the SSP-RK3 based LTS algorithm 44.9% for the problem of 40,962 cells, and these two numbers increase to 67.1% and 77.5% respectively for the problem of 163,842 cells, and finally reach 86.2% and 93.3% for the problem of 655,362 cells. These results demonstrate excellent scalability of the proposed LTS algorithms.

We also compare the performance in terms of running time of the proposed LTS algorithms and the original global SSP-RK algorithms for the SWTC5 test case. We consider the mesh of 655,362 cells, in which the region with a fine time step contains 216,701 cells and the one with a coarse time step contains 438,661 cells. The global SSP-RK algorithms use a uniform time step size $\Delta t = 0.125\alpha$ while the proposed LTS algorithms use different time step sizes, $\Delta t_{\text{coarse}} = 0.5\alpha$

27

No. of	40,962 Cells			163,842 Cells			655,362 Cells		
Cores	Time	Speedup	Efficiency	Time	Speedup	Efficiency	Time	Speedup	Efficency
The SSP-RK2 based LTS algorithm									
1	286.90	-	-	1208.94	-	-	5122.30	-	-
2	152.31	1.88	94.2%	605.24	2.00	99.9%	2531.42	2.02	101.1%
4	81.92	3.50	87.6%	305.91	3.95	98.8%	1290.65	3.97	99.2%
8	44.21	6.49	81.1%	158.15	7.64	95.6%	677.51	7.56	94.6%
16	24.95	11.50	71.9%	82.70	14.62	91.3%	339.14	15.10	94.4%
32	15.00	19.13	59.8%	44.74	27.02	84.4%	177.56	28.85	90.2%
64	9.37	30.63	47.9%	24.37	49.61	77.5%	87.99	58.22	91.0%
128	6.40	44.84	35.0%	14.09	85.82	67.1%	46.41	110.37	86.2%
The SSP-RK3 based LTS algorithm									
1	398.50	-	-	1704.73	-	-	7220.48	-	-
2	207.41	1.92	96.1%	838.29	2.03	101.7%	3543.05	2.04	101.9%
4	109.93	3.62	90.6%	420.18	4.06	101.4%	1745.22	4.14	103.4%
8	58.23	6.84	85.6%	213.74	7.98	99.7%	889.65	8.12	101.5%
16	31.82	12.52	78.3%	110.45	15.43	96.5%	461.69	15.64	97.7%
32	18.97	21.00	65.6%	57.51	29.64	92.6%	236.77	30.50	95.3%
64	10.86	36.70	57.4%	30.94	55.10	86.1%	115.57	62.47	97.6%
128	6.93	57.51	44.9%	17.18	99.20	77.5%	60.43	119.48	93.3%

Table 9: Results of the parallel scalability of the proposed LTS algorithms with $\Delta t_{\text{coarse}} = 0.5\alpha$ and $M = 4$ for the SWTC5 test case with $T = 3$ hours. The computer running times are measured in seconds.

and $\Delta t_{\text{fine}} = 0.125\alpha$ (i.e., $M = 4$). The running times of all algorithms with different number of cores are reported in Table 10, together with the corresponding ratios of the running times of the algorithms without LTS versus with LTS. The theoretically optimal value for the ratio is approximately $(4 \times 655362)/(1 \times 438661 + 4 \times 216701) \approx 2.01$ when the cost for interface predictions and corrections is considered to be negligible. It is observed that both the proposed second and third order LTS algorithms under our parallel implementation perform very well, and the achieved practical values for the ratio are very close to the optimal one for all tested numbers of cores. This confirms the efficiency of the proposed LTS algorithms versus the global SSP-RK time-stepping.

6. Conclusions

In this paper, we have designed conservative explicit local time-stepping algorithms for shallow water equations discretized in space by the TRiSK scheme on multi-resolution meshes. Different time-step sizes are used in different regions of the domain and the restrictive global CFL condition is replaced by a local CFL condition. Second and third order LTS algorithms based on the explicit SSP-RK methods are developed and tested on different test-cases. Specially, numerical results show that the proposed LTS algorithms are respectively second and third order accurate in time on the whole domain, even at the local time-stepping interface. The LTS algorithms also couple very well with multi-resolution meshes while preserving all the properties of the TRiSK scheme such as the conservation of some important nonlinear quantities. In addition, these algorithms inherit the natural parallelism of the original explicit global time-stepping schemes as verified by the scalability

No. of	The SSP-RK2 algorithm			The SSP-RK3 algorithm		
Cores	Without LTS	With LTS	Ratio	Without LTS	With LTS	Ratio
1	10213.82	5122.30	1.99	14476.94	7220.48	2.00
2	5036.51	2531.42	1.99	7021.38	3543.05	1.98
4	2503.81	1290.65	1.94	3348.39	1745.22	1.92
8	1287.30	677.51	1.90	1722.99	889.65	1.94
16	674.88	339.14	1.99	883.58	461.69	1.91
32	331.31	177.56	1.87	463.39	236.77	1.96
64	168.53	87.99	1.92	229.58	115.37	1.99
128	92.35	46.41	1.99	119.57	60.43	1.98

Table 10: Running times of the global SSP-RK algorithms with uniform time step size $\Delta t = 0.125\alpha$ and the proposed LTS algorithms with $\Delta t_{\mathrm{coarse}} = 0.5\alpha$ and $\Delta t_{\mathrm{fine}} = 0.125\alpha$ (i.e., $M = 4$) for the SWTC5 test case with $T = 3$ hours on the mesh of 655,362 cells. The computer running times are measured in seconds.

tests. Thus, the proposed algorithms have opened up the possibility for global ocean to coastal scale simulations at high computational efficiency. With the use of spatially variable time step sizes, coastal cells of much smaller scale (compared to those in the global ocean) no longer constrain the time steps of the overall simulation. Our next steps include further analysis of the accuracy and performance of the LTS schemes on more realistic test cases and extending the algorithms to more complex models, such as the case of multi-layers and the full ocean model [34] with tracer transport existing in MPAS-Ocean.

Acknowledgement

References

[1] A. ASHBOURNE, *Efficient Runge-Kutta Based Local Time-Stepping Methods*, Master Thesis, University of Waterloo, Canada, 2016.

[2] M. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys. **53**, 1984, pp. 484-512.

[3] M. BERGER AND R.J. LEVEQUE, *Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems*, SIAM J. Numer. Anal. **35**, 1998, pp. 2298-2316.

[4] E. BLAYO, L. HALPERN AND C. JAPHET, *Optimized Schwarz waveform relaxation algorithms with nonconforming time discretization for coupling convection-diffusion problems with discontinuous coefficients.* In Domain Decomposition Methods in Science and Engineering XVI, Lect. Notes Comput. Sci. Eng. **55**, Springer, Berlin, 2007, pp. 267-274.

[5] F.P. BRETHERTON, *Critical layer instability in baroclinic flows*, Q. J. Royal Meteorol. Soc. **92** (393), 1966.

[6] E.M. CONSTANTINESCU AND A. SANDU, *Multirate timestepping methods for hyperbolic conservation laws*, J. Sci. Comput. **33**, 2007, pp. 239-278.

[7] C. DAWSON AND KIRBY, *High resolution upwind-mixed finite element methods for advection-diffusion equations with variable time-stepping*, Numer. Methods. Partial Differ. Eqs. **11**(5), 1995, pp. 525-538.

[8] C. DAWSON AND KIRBY, *High resolution schemes for conservation laws with locally varying time steps*, SIAM J. Sci. Comput. **22**(6), 2001, pp. 2256-2281.

[9] Q. DU, V. FABRE AND M. GUNZBURGER, *Centroidal Voronoi tessellations: applications and algorithms*, SIAM Review **41**(4), 1999, pp. 637-676.

[10] Q. DU, M. GUNZBURGER AND L. .JU, *Constrained centroidal Voronoi tessellations for surfaces*, SIAM J. Sci. Comput. **24**, 2003, pp. 1488-1506.

[11] C. FUMEAUX, D. BAUMANN, P. LEUCHTMANN AND R. VAHLDIECK, *A generalized local time-step scheme for efficient fvtd simulations in strongly inhomogeneous meshes*, IEEE Trans. on Microwave Theory and Techniques **52**(3), 2004, pp. 1067-1076.

[12] M. J. GANDER AND L. HALPERN, *Techniques for locally adaptive time stepping developed over the last two decades.* In Domain Decomposition Methods in Science and Engineering XX, Lect. Notes Comput. Sci. Eng. **91**, Springer, Berlin, 2007, pp. 377-385.

[13] M. J. GANDER, L. HALPERN AND M. KERN, *A Schwarz waveform relaxation method for advection-diffusion-reaction problems with discontinuous coefficients and non-matching grids.* In Domain Decomposition Methods in Science and Engineering XVI, Lect. Notes Comput. Sci. Eng. **55**, Springer, Berlin, 2007, pp. 283-290.

[14] S. GOTTLIEB AND C.-W. SHU, *Total variation diminishing Runge Kutta schemes*, Math. Comp. **67**, 1998, pp. 73-85.

[15] S. GOTTLIEB, C.-W. SHU AND E. TADMOR, *Strong Stability-Preserving High-Order Time Discretization Methods*, SIAM Rev. **43(1)**, 2001, pp. 89-112.

[16] M.J. GROTE, M. MEHLIN AND T. MITKOVA, *Runge-Kutta-based explicit local time-stepping methods for wave propagation*, SIAM J. Sci. Comput. **37 (2)**, 2015, pp. A747-A775.

[17] S. GUPTA, B. WOHLMUTH, R. HELMIG, *Multi-rate timestepping schemes for hydro-geomechanical model for subsurface methane hydrate reservoirs*, Adv. Water Resour. **91**, 2016, pp. 78-87.

[18] L. HALPERN, C. JAPHET AND J. SZEFTEL, *Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problems*, SIAM J. Numer. Anal. **50**, 2012, pp. 2588-2611.

[19] T.T.P. HOANG, J. JAFFRÉ, C. JAPHET, M. KERN AND J.E. ROBERTS, *Space-time domain decomposition methods for diffusion problems in mixed formulations*, SIAM J. Numer. Anal. **51(6)**, 2013, pp. 3532-3559.

[20] T.T.P. Hoang, L. Ju, W. Leng and Z. Wang, *High-order Runge-Kutta discontinuous Galerkin methods with local time-stepping for conservation laws*, in preparation.

[21] B.J. Hoskins, M.E. McIntyre and A.W. Robertson, *On the use and significance of isentropic potential vorticity maps*, Q. J. Royal Meteorol. Soc. **111** (466), 1985.

[22] L. Ju, T. Ringler and M. Gunzburger, *Voronoi tessellations and their application to climate and global modeling*, Numerical techniques for global atmospheric models. In: P.H. Lauritzen, C. Jablonowski, M.A. Taylor, R.D. Nair (Eds.), Springer Lecture Notes in Computational Science and Engineering, 2010, pp. 1-30.

[23] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput. **20**(1), 1998, pp. 359-392.

[24] L. Krivodonova, *An efficient local time-stepping scheme for solution of nonlinear conservation laws*, J. Comput. Phys. **229**, 2010, pp. 8537-8551.

[25] E.J. Kubatko, C. Dawson, J. Westerink, *Time step restrictions for Runge-Kutta discontinuous Galerkin methods on triangular grids*, J. Comput. Phys. **227**, 2008, pp. 9697-9710.

[26] E. J. Kubatko, B. A. Yeager and D. I. Ketcheson, *Optimal Strong-Stability-Preserving Runge-Kutta time discretizations for discontinuous Galerkin methods*, J. Sci. Comput. **60**, 2014, pp. 313-344.

[27] J. Marshall, D. Olbers, H. Ross and D. Wolf-Gladrow, *Potential vorticity constraints on the dynamics and hydrography of the southern ocean*, J. Phys. Oceanogr. **23** (3), 1993, pp. 465-487.

[28] S. Muller and Y. Stiriba, *Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping*, J. Sci. Comput. **30**(3), 2006, pp. 493-531.

[29] S. Osher and R. Sanders, *Numerical Approximations to Nonlinear Conservation Laws with Locally Varying Time and Space Grids*, Math. Comp. **41** (164), 1983, pp. 321-336.

[30] F. Pellegrini, *SCOTCH 5.1 User's Guide*, Laboratoire Bordelais de Recherche en Informatique (LaBRI), 2008.

[31] M. Rietmann, D. Peter, O. Schenk, B. Uar and M.J. Grote, *Load-balanced local time stepping for large-scale wave propagation*, In Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International, pp. 925-935.

[32] T.D. Ringler, L. Ju and M. Gunzburger, *A multi-resolution method for climate system modeling*, Ocean Dyn. **58**, 2008, pp. 475-498.

[33] T.D. Ringler, D. Jacobsen, M. Gunzburger, L. Ju, M. Duda and W. Skamarock, *Exploring a multiresolution modeling approach within the shallow-water equations*, Monthly Weather Review **139** (11), 2011, pp. 3348-3368.

[34] T.D. Ringler, M. Petersen, R.L. Higdon, D. Jacobsen, P.W. Jones and M. Maltrud, *A multi-resolution approach to global ocean modeling*, Ocean Modelling **69**, 2013, pp. 211-232.

[35] T.D. Ringler, J. Thuburn, W.C. Skamarock and J.B. Klemp, *A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids*, J. Comput. Phys. **229**, 2010, pp. 3065-3090.

[36] I. Rybak, J. Magiera, R, Helmig and C. Rohde, *Multirate time integration for coupled saturated/unsaturated porous medium and free flow systems*, Comput Geosci **19**, 2015, pp. 299-309.

[37] R. Sadourny and C. Basdevant, *Parameterization of subgrid scale barotropic and baroclinic eddies in quasi-geostrophic models: anticipated potential vorticity method*, J. Atmos. Sci. **42** (13), 1985, pp. 1353-1363.

[38] B.F. Sanders, *Integration of a shallow water model with a local time-step*, J. Hydraul. Res. **46** (4), 2008, pp. 466-475.

[39] B. Seny, J. Lambrechts, T. Toulorge and J. Remacle , *An efficient parallel implementation of explicit multirate Runge-Kutta schemes for discontinuous Galerkin computations*, J. Comput. Phys. **256**, 2014, pp. 135-160.

[40] C.-W. Shu, *TVB uniformly high-order schemes for conservation laws*, Math. Comput. **49** (179), 1987, pp. 105-121.

[41] C.-W. Shu, *Total-Variation-Diminishing time discretizations*, SIAM J. Sci. Statist. Comput. **9**, 1988, pp. 1073-1084.

[42] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys. **77**, 1988, pp. 439-471.

[43] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, II*, J. Comput. Phys. **83** (1), 1989, pp. 32-78.

[44] Z. Tan, Z. Zhang, Y. Huang and T. Tang, *Moving mesh methods with locally varying time steps*, J. Comput. Phys. **200**, 2004, pp. 347-367.

[45] H. Tang and G. Warnecke, *A class of high resolution difference schemes for nonlinear Hamilton-Jacobi equations with varying time and space grids*, SIAM J. Sci. Comput. **26**(4), 2005, pp. 1415-1431.

[46] J. Thuburn, T.D. Ringler, W.C. Skamarock and J.B. Klemp, *Numerical representation of geostrophic modes on arbitrarily structured C-grids*, J. Comput. Phys. **228** (22), 2009, pp. 8321-8335.

[47] C.J. Trahan and C. Dawson, *Local time-stepping in Runge-Kutta discontinuous Galerkin finite element methods applied to the shallow-water equations*, Comput. Meth. Appl. Mech. Eng. **217**, 2012, pp. 139-152.

[48] D.L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, *A standard test set for numerical approximations to the shallow water equations in spherical geometry*, J. Comput. Phys. **102**, 1992, pp. 211-224.

[49] P.J. WOLFRAM, T.D. RINGLER, M.E. MALTRUD, D.W. JACOBSEN, AND M.R. PETERSEN, *Diagnosing Isopycnal Diffusivity in an Eddying, Idealized Midlatitude Ocean Basin via Lagrangian, in Situ, Global, High-Performance Particle Tracking (LIGHT)*, J. Phys. Oceanogr. **45**, 2114-2133, 2015.

Appendix A   Derivation of the predictors

Consider an autonomous system of ordinary differential equations of the form:

$$w_t = F(w), \quad t \in (0, T), \tag{A.1}$$

with an initial condition $w(0) = w_0$. Let us partition the time interval $(0, T)$ into N uniform intervals:

$$0 = t^0 < t^1 < \ldots < t^n < t^{n+1} < \ldots < t^N = T, \quad \Delta t = t^{n+1} - t^n = \frac{T}{N},$$

and each interval is then further divided into M subintervals:

$$[t^n, t^{n+1}) = \cup_{k=0}^{M-1} [t^{n,k}, t^{n,k+1}), \quad \text{with } t^{n,0} = t^n \text{ and } t^{n,k+1} = t^{n,k} + \frac{\Delta t}{M}, \ k = 0, \ldots, M-1.$$

Given the solution w^n at t^n, we construct the approximation of w at the intermediate time levels $t^{n,k}$ for the second and third order SSP-RK schemes respectively. The idea is mostly based on Taylor series expansion.

A.1   The predictor for the second order LTS algorithm

The approximate solution to (A.1) by SSP-RK2 with a coarse time step size Δt is given by

$$\overline{w}^{n+1} = w^n + \Delta t F(w^n), \tag{A.2}$$
$$w^{n+1} = \frac{1}{2} w^n + \frac{1}{2} \left(\overline{w}^{n+1} + \Delta t F(\overline{w}^{n+1}) \right).$$

Given the solution w^n at t^n, our aim is to find the second order approximation of $w(t^{n,k})$ and $\overline{w}(t^{n,k+1})$ at the intermediate time level $t^{n,k}$ (and $t^{n,k+1}$ respectively), for $k = 0, \ldots, M-1$. Performing Taylor series expansion of w at t^n yields

$$w(t) = w(t^n) + (t - t^n) w_t(t^n) + \frac{1}{2!} (t - t^n)^2 w_{tt}(t^n) + \ldots, \tag{A.3}$$

thus we can approximate $w(t^{n,k})$ by truncating (A.3) to the second term (the first-order Taylor expansion):

$$w^{n,k} = w^n + \frac{k \Delta t}{M} w_t^n, \tag{A.4}$$

which gives a truncation error of second order in time. From (A.1) and (A.2) we deduce

$$w_t^n = F(w^n) = \frac{\overline{w}^{n+1} - w^n}{\Delta t}. \tag{A.5}$$

Substituting this into (A.4) we obtain a second order approximation of $w(t^{n,k})$

$$w^{n,k} = w^n + \frac{k}{M}\left(\overline{w}^{n+1} - w^n\right) = (1 - \alpha_k)w^n + \alpha_k\,\overline{w}^{n+1}, \tag{A.6}$$

where $\alpha_k = \dfrac{k}{M}$ for $k = 0, \ldots, M - 1$. Next, applying SSP-RK2 with a fine time step size $\dfrac{\Delta t}{M}$ at $t^{n,k}$, we get

$$\overline{w}^{n,k+1} = w^{n,k} + \frac{\Delta t}{M}F(w^{n,k}),$$

where we can approximate $F(w^{n,k})$ by $F(w^n)$ so that a second order approximation of $\overline{w}^{n,k+1}$ is obtained as

$$\overline{w}^{n,k+1} = w^{n,k} + \frac{\Delta t}{M}F(w^n) = w^n + \frac{k+1}{M}\left(\overline{w}^{n+1} - w^n\right) = (1 - \beta_k)w^n + \beta_k\,\overline{w}^{n+1}, \tag{A.7}$$

where $\beta_k = \dfrac{k+1}{M}$, $k = 0, \ldots, M - 1$.

A.2 The predictor for the third order LTS algorithm

The approximate solution to (A.1) by SSP-RK3 with a coarse time step size Δt is given by

$$\overline{w}^{n+1} = w^n + \Delta t\, F(w^n), \tag{A.8}$$

$$\overline{w}^{n+1/2} = \frac{3}{4}w^n + \frac{1}{4}\left(\overline{w}^{n+1} + \Delta t\, F(\overline{w}^{n+1})\right), \tag{A.9}$$

$$w^{n+1} = \frac{1}{3}w^n + \frac{2}{3}\left(\overline{w}^{n+1/2} + \Delta t\, F(\overline{w}^{n+1/2})\right). $$

As in the second order case, we use Taylor expansion to approximate the solution $w(t^{n,k})$; however, we need truncate (A.3) to the third term (the second-order Taylor expansion):

$$w^{n,k} = w^n + \frac{k\Delta t}{M}w_t^n + \frac{1}{2}\left(\frac{k\Delta t}{M}\right)^2 w_{tt}^n, \tag{A.10}$$

which gives a truncation error of third order in time. To find the approximation of w_{tt}^n, we rewrite (A.9) as follows:

$$\overline{w}^{n+1/2} = \frac{3}{4}w^n + \frac{1}{4}\left(\overline{w}^{n+1} + \Delta t\, F(\overline{w}^{n+1})\right) = w^n + \frac{1}{4}\Delta t\left(F(w^n) + F(\overline{w}^{n+1})\right). \tag{A.11}$$

Using Taylor expansion, we get

$$F(\overline{w}^{n+1}) = F(w^n) + (\overline{w}^{n+1} - w^n)F'(w^n) + \ldots = F(w^n) + \Delta t\, w_t^n\, F'(w^n) + O((\Delta t)^2). \tag{A.12}$$

Substituting it into (A.11) results in

$$\overline{w}^{n+1/2} = w^n + \frac{1}{2}\Delta t F(w^n) + \frac{1}{4}(\Delta t)^2\, w_t^n F'(w^n) + O(\Delta t^3). \tag{A.13}$$

Differentiate (A.1) with respect to t and by the chain rule, we have

$$w_{tt}(t^n) = F'(w(t^n))w_t(t^n).$$

Thus we can approximate $w_{tt}(t^n)$ from (A.13) as follows:

$$w_{tt}^n = \frac{4}{(\Delta t)^2}\left(\overline{w}^{n+1/2} - w^n - \frac{1}{2}(\overline{w}^{n+1} - w^n)\right) = \frac{4}{(\Delta t)^2}\left(\overline{w}^{n+1/2} - \frac{1}{2}\overline{w}^{n+1} - \frac{1}{2}w^n\right).$$

Substituting this into (A.10) yields a third order approximation of $w^{n,k}$

$$
\begin{aligned}
w^{n,k} &= w^n + \frac{k}{M}(\overline{w}^{n+1} - w^n) + 2\frac{k^2}{M^2}\left(\overline{w}^{n+1/2} - \frac{1}{2}\overline{w}^{n+1} - \frac{1}{2}w^n\right) \\
&= (1 - \alpha_k - \widehat{\alpha}_k)w^n + (\alpha_k - \widehat{\alpha}_k)\overline{w}^{n+1} + 2\widehat{\alpha}_k\,\overline{w}^{n+1/2},
\end{aligned}
\tag{A.14}
$$

where $\widehat{\alpha}_k = \frac{k^2}{M^2}$, $k = 0, \ldots, M - 1$. Now for $\overline{w}^{n,k+1}$ and $\overline{w}^{n,k+1/2}$, we follow the SSP-RK3 with a fine time step size:

$$\overline{w}^{n,k+1} = w^{n,k} + \frac{1}{M}\Delta t F(w^{n,k}),$$

$$\overline{w}^{n,k+1/2} = \frac{3}{4}w^{n,k} + \frac{1}{4}\left(\overline{w}^{n,k+1} + \frac{1}{M}\Delta t F(\overline{w}^{n,k+1})\right) = w^{n,k} + \frac{1}{4M}\Delta t\left(F(w^{n,k}) + F(\overline{w}^{n,k+1})\right).$$

We use Taylor expansion (A.12) to obtain a second order approximation of $F(w^{n,k})$ and $F(\overline{w}^{n,k+1})$:

$$F(w^{n,k}) = F(w^n) + \frac{k\Delta t}{M}w_t^n\, F'(w^n),$$

$$F(\overline{w}^{n,k+1}) = F(w^n) + \frac{(k+1)\Delta t}{M}w_t^n\, F'(w^n),$$

Then we get a third order approximation of $\overline{w}^{n,k+1}$ as

$$
\begin{aligned}
\overline{w}^{n,k+1} &= w^{n,k} + \frac{\Delta t}{M}F(w^n) + \frac{k(\Delta t)^2}{M^2}w_t^n\frac{\partial F}{\partial w}(w^n) \\
&= w^{n,k} + \frac{1}{M}(\overline{w}^{n+1} - w^n) + 4\frac{k}{M^2}\left(\overline{w}^{n+1/2} - \frac{1}{2}\overline{w}^{n+1} - \frac{1}{2}w^n\right) \\
&= w^n + \frac{k+1}{M}(\overline{w}^{n+1} - w^n) + 2\frac{k^2 + 2k}{M^2}\left(\overline{w}^{n+1/2} - \frac{1}{2}\overline{w}^{n+1} - \frac{1}{2}w^n\right) \\
&= (1 - \beta_k - \widehat{\beta}_k)w^n + (\beta_k - \widehat{\beta}_k)\overline{w}^{n+1} + 2\widehat{\beta}_k\,\overline{w}^{n+1/2},
\end{aligned}
\tag{A.15}
$$

where $\widehat{\beta}_k = \frac{k(k+2)}{M^2}$, $k = 0, \ldots, M-1$. Similarly, we obtain a third order approximation of $\overline{w}^{n,k+1/2}$ as

$$
\begin{aligned}
\overline{w}^{n,k+1/2} &= w^{n,k} + \frac{(\Delta t)^2}{4M}\left(F(w^{n,k}) + F(\overline{w}^{n,k+1})\right) \\
&= w^n + \frac{2k+1}{2M}(\overline{w}^{n+1} - w^n) + \frac{2k^2 + 2k + 1}{M^2}\left(\overline{w}^{n+1/2} - \frac{1}{2}\overline{w}^{n+1} - \frac{1}{2}w^n\right) \\
&= (1 - \gamma_k - \widehat{\gamma}_k)w^n + (\gamma_k - \widehat{\gamma}_k)\overline{w}^{n+1} + 2\widehat{\gamma}_k\,\overline{w}^{n+1/2},
\end{aligned}
\tag{A.16}
$$

where

$$\gamma_k = \frac{2k+1}{2M}, \quad \widehat{\gamma}_k = \frac{2k^2 + 2k + 1}{2M^2}, \quad k = 0, \ldots, M - 1.$$