Space-time residual distribution on moving meshes

M.E. Hubbard[a,*], M. Ricchiuto[b], D. Sármány[c,1]

[a]*School of Mathematical Sciences, University of Nottingham,*
University Park, Nottingham, NG7 2RD, UK
[b]*Team CARDAMOM, INRIA Bordeaux – Sud-Ouest,*
200 Avenue de la Vieille Tour, 33405 Talence Cedex, France
[c]*School of Computing, University of Leeds, Leeds, LS2 9JT, UK*

Abstract

This article investigates the potential for an r-adaptation algorithm to improve the efficiency of space-time residual distribution schemes in the approximation of time-dependent hyperbolic conservation laws, *e.g.* scalar advection, shallow water flows, on unstructured, triangular meshes. In this adaptive framework the connectivity of the mesh, and hence the number of degrees of freedom, remain fixed, but the mesh nodes are continually "relocated" as the flow evolves so that features of interest remain resolved as they move within the domain.

Adaptive strategies of this type are well suited to the space-time residual distribution framework because, when the discrete representation is allowed to be discontinuous in time, these algorithms can be designed to be positive (and hence stable) for any choice of time-step, even on the distorted space-time prisms which arise from moving the nodes of an unstructured triangular mesh. Consequently, a local increase in mesh resolution does not impose a more restrictive stability constraint on the time-step, which can instead be chosen according to accuracy requirements. The order of accuracy of the fixed-mesh scheme is retained on the moving mesh in the majority of applications tested.

Space-time schemes of this type are analogous to conservative ALE formulations and automatically satisfy a discrete geometric conservation law, so moving the mesh does not artificially change the flow volume for pure conservation laws. For shallow water flows over variable bed topography, the so-called C-property (retention of hydrostatic balance between flux and source terms, required to maintain the steady state of still, flat, water) can also be satisfied by considering the mass balance equation in terms of free surface level instead of water depth, even when the mesh is moved.

The r-adaptation is applied within each time-step by interleaving the iterations of the nonlinear solver with updates to mesh node positions. The node movement is driven by a monitor function based on weighted approximations of the scaled gradient and Laplacian of the local solution and regularised by a smoothing iteration. Numerical results are shown in two dimensions for both scalar advection and for shallow water flow over a variable bed

---

*Corresponding author
Email addresses: `matthew.hubbard@nottingham.ac.uk` (M.E. Hubbard),
`mario.ricchiuto@inria.fr` (M. Ricchiuto), `domokos.sarmany@ecmwf.int` (D. Sármány)
[1]Present address: European Centre for Medium-Range Weather Forecasts, Shinfield, Reading, Berkshire, RG2 9AX, UK

which show that, even for this simple implementation of the mesh movement, reductions in cpu times of up to 60% can be attained without increasing the error.

Keywords: moving meshes, conservative ALE, upwind residual distribution, shallow water equations, discontinuous space-time representation, well-balanced schemes

1. Introduction

Residual distribution schemes [1, 2] have been developed as an alternative to flux-based approaches [3, 4, 5] for approximating hyperbolic conservation laws. They provide a framework within which the underlying flow physics can be represented in a genuinely multidimensional manner, unlike the approximate Riemann solvers employed in finite volume and discontinuous Galerkin schemes [5], while simultaneously avoiding the introduction of spurious numerical oscillations, especially in the vicinity of discontinuities. Robust and accurate steady-state schemes appeared in the 1980s and 1990s and were combined with adaptive mesh movement to improve their shock-capturing capabilities [6]. Since then, both explicit, Runge-Kutta [7, 8], and implicit, space-time [9, 10], schemes have been developed for simulating time-dependent problems.

Recent research has focussed on improving the efficiency of these time-dependent methods by moving the computational mesh so that the features of interest are resolved by regions in which mesh nodes are more densely clustered (r-adaptivity). The aim is to make the most efficient use of a fixed number of degrees of freedom by placing them where they will most effectively reduce the error. Explicit residual distribution schemes have been combined successfully with mesh movement in [11, 12, 13] – this paper will discuss the application of moving meshes in the space-time framework. There are particular attractions of combining mesh movement with these implicit methods because, by allowing a discontinuous-in-time representation, they can be constructed in a manner which is unconditionally stable [10], so (as long as the nonlinear implicit solver can be persuaded to converge) the size of the time-step can be chosen on the basis of accuracy – it is not limited by a CFL constraint determined by the size of the computational mesh.

The implicit nature of our scheme allows us to interleave the iterations of our nonlinear solver with adjustments to the positions of the mesh nodes at the new time level in the space-time mesh. In line with many standard approaches, *e.g.* [14, 15], the mesh movement is determined by an iteration which, at convergence, aims to equidistribute a monitor function which is chosen to indicate regions where the local error is high. We choose a monitor based on first and second derivatives of the dependent variable, to demonstrate the algorithm and give an indication of the potential of combining mesh movement with space-time residual distribution.

All mesh movement algorithms should satisfy a discrete geometric conservation law (DGCL) [16] – this guarantees that no mass is artificially created or destroyed by the movement of the mesh – and conservative space-time schemes inherit this property automatically. In fact, the space-time schemes are closely related to standard arbitrary Lagrangian-Eulerian (ALE) schemes, and can be used to derive ALE schemes which inherit the DGCL. As such, they might be classed as velocity-based moving mesh methods [17, 18], in which the node displacements from one time level to the next imply a mesh velocity field. This contrasts with the transformation-based approach, which has been ap-

2

plied to the explicit residual distribution schemes in [11, 19], though the two approaches are related.

We aim to apply space-time residual distribution with moving meshes to the shallow water equations, a nonlinear, hyperbolic, system of partial differential equations commonly used to study the hydrodynamics of coastal and river flows. This raises an additional issue, the preservation of the "lake-at-rest" steady state, in which the numerical approximations to the flux and source terms balance perfectly when the bed topography varies. This is also known as the C-property or well-balanced property, and has been the subject of much research in the last 20 years, particularly in the context of flux-based schemes [20, 21, 22]. It is actually more natural for residual distribution schemes to satisfy this property because the discrete flux and source terms which must balance are both evaluated using volume integrals, and appropriate schemes have been developed on fixed meshes. We will show how this can be extended to moving meshes for the space-time approach, linking it with recent work combining moving meshes with the explicit, Runge-Kutta, schemes [11].

The paper is organised as follows: in Section 2.1 we introduce the space-time residual distribution framework for a scalar hyperbolic conservation law and describe how it can be applied on the distorted triangular prisms that arise from moving the mesh in two space dimensions. This is supplemented by a proof that, when allowing discontinuities in time, the space-time N scheme satisfies a discrete maximum principle for any time-step. The extension to nonlinear systems of equations is described in Section 2.2 and its application to the shallow water equations is given in Section 2.3, with particular attention paid to the issues relating to producing a scheme which is both conservative and well-balanced when the mesh is moved over a variable bed topography. A simple moving mesh iteration is described in Section 2.4, along with an outline of how it is combined with the nonlinear implicit solver. Numerical results which demonstrate the potential of the adaptive scheme when applied to scalar conservation laws and the shallow water equations in two space dimensions are provided in Section 3 and concluding remarks are given in Section 4.

2. Residual Distribution on Moving Meshes

In this section, we present a description for space-time discontinuous residual distribution (STDRD) schemes for hyperbolic conservation laws with zero right-hand side on distorted triangular prisms, ones in which the two triangular faces are parallel (each is fixed at a constant time level) but may otherwise be different. A detailed description of STDRD schemes on right triangular prisms can be found in [10]. We will show how applying this scheme on distorted prisms relates to ALE-type approaches [13, 19] which are more commonly considered when the spatial mesh is allowed to change from one time-step to the next.

Consider a two-dimensional spatial domain $\Omega \subset \mathbb{R}^2$, its triangular tessellations at time levels n and $n + 1$, Ω_h^n and Ω_h^{n+1} respectively, and the corresponding space-time mesh, $(\Omega_h^t)^n$, between these two time levels. Let E^n denote a given triangular element of the spatial mesh at time level n, $E_t{}^n$ denote the corresponding space-time element, defined by joining the corresponding vertices of E^n and E^{n+1} with straight lines (see Figure 1), and define $D_i = \cup_{i \in E} E$ to be the patch of elements with vertex i in common

(the superscript indicating the time level has been dropped because the spatial mesh connectivity is assumed to remain unchanged throughout the computation). Note that the lateral faces of $E_t{}^n$ can be constructed using bilinear transformations of those of a right prism, forming a "distorted prism" which will not necessarily be a polyhedron because the vertices of each lateral face are not constrained to be coplanar. The residual distribution schemes described in the following sections generalise straightforwardly to three space dimensions.

2.1. Scalar Equations

Consider the scalar conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{f}(u) = 0 \qquad \text{or} \qquad \frac{\partial u}{\partial t} + \boldsymbol{a}(u) \cdot \nabla u = 0 \tag{1}$$

with appropriate initial conditions and Dirichlet boundary conditions at the inflow part of the domain. Here $\boldsymbol{f}(u)$ represents the conservative flux vector and $\boldsymbol{a}(u) = \partial \boldsymbol{f}/\partial u$ is the corresponding wave speed. The associated residual over the space-time element $E_t{}^n$ is given by

$$\phi_{E_t}^n = \int\limits_{t^n}^{t^{n+1}} \int\limits_{E(t)} \left(\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{f} \right) \, \mathrm{d}\Omega \, \mathrm{d}t \,. \tag{2}$$

For the original (non-discretised) equation (1), it therefore holds that

$$\int\limits_{t^n}^{t^{n+1}} \int\limits_{\Omega(t)} \left(\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{f} \right) \, \mathrm{d}\Omega \, \mathrm{d}t = \sum_{E_t^n \in (\Omega_h^t)^n} \phi_{E_t}^n \,, \tag{3}$$

provided that the solution u is bounded and piecewise differentiable.

It is also worth recalling here that, by the Reynolds transport theorem,

$$\frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{E(t)} u \, \mathrm{d}\Omega = \int\limits_{E(t)} \frac{\partial u}{\partial t} \, \mathrm{d}\Omega + \int\limits_{\partial E(t)} u \boldsymbol{v} \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma \,, \tag{4}$$

in which $\hat{\boldsymbol{n}}$ is the outward-pointing unit vector normal to the element boundary $\partial E(t)$ and $\boldsymbol{v}$ is the velocity of this moving boundary. Applying the divergence theorem, integrating with respect to time, and combining with Equation (2) then gives

$$\phi_{E_t}^n = \int\limits_{E(t^{n+1})} u(t^{n+1}) \, \mathrm{d}\Omega - \int\limits_{E(t^n)} u(t^n) \, \mathrm{d}\Omega + \int\limits_{t^n}^{t^{n+1}} \int\limits_{\partial E(t)} (\boldsymbol{f} - u\boldsymbol{v}) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma \, \mathrm{d}t \,, \tag{5}$$

i.e. solving $\phi_{E_t}^n = 0$ for all elements is equivalent to updating the solution according to the Reynolds transport theorem. This observation is presented to emphasise the close relationship between the residual distribution method derived in this paper by integrating the residual over a space-time element and the family of conservative ALE methods based on discretising the Reynolds transport theorem [23, 24]. From now on the superscript $\cdot^n$ will be dropped from the notation where the time level is obvious from the context.

4

2.1.1. Distributing the Residual

For STDRD, only right prisms have been considered so far in the literature [9, 10], *i.e.* when the spatial coordinates of E^n and E^{n+1} are exactly the same. In this work, the positions of the mesh nodes at time level t^{n+1} are allowed to differ from their positions at time level t^n. Even though the STDRD framework allows for discontinuities in time, the spatial coordinates of E^n are fixed at the values obtained in the previous time-step to avoid having to apply a conservative reconstruction of the dependent variable on the new mesh. For simplicity, it is also assumed that the shape of the domain Ω remains unchanged, and that boundary nodes only move along the boundary, not perpendicular to it. This condition could be relaxed, as long as the movement of the boundary can be predicted accurately and an appropriate boundary condition can be defined on the moving boundary.

The derivation of the STDRD numerical discretisation to be used here follows precisely that presented in [10] for fixed meshes. In particular, it is assumed that the dependent variable has a piecewise linear continuous representation in space at any given time, but discontinuities are allowed in time. This leads to a family of schemes which are second-order accurate and unconditionally stable – higher-order schemes have been derived on fixed meshes [25, 26, 27, 28, 29, 30, 31, 32]. The discretisation consists of the following general steps:

1. In every space-time element, replace the unknown variable u with an approximation u_h that varies linearly along the edges of the element.

2. Transform the element residual into a space-time boundary integral:

$$
\phi_{E_t} = \int\limits_{t^n}^{t^{n+1}} \int\limits_{E(t)} \left(\frac{\partial u_h}{\partial t} + \nabla \cdot \boldsymbol{f}_h \right) \, \mathrm{d}\Omega \, \mathrm{d}t = \int\limits_{E_t} \nabla_t \cdot \boldsymbol{f}_t \, \mathrm{d}\Omega_t = \int\limits_{\partial E_t} \boldsymbol{f}_t \cdot \hat{\boldsymbol{n}}_t \, \mathrm{d}\Gamma_t, \quad (6)
$$

   where $\nabla_t = (\partial/\partial t, \partial/\partial x, \partial/\partial y)$, $\boldsymbol{f}_t = (u_h, \boldsymbol{f}_h)$, and $\hat{\boldsymbol{n}}_t$ is the outward-pointing unit vector normal to the surface of the space-time element.

   In this work, it is assumed that the edges of each space-time element are straight lines (the mesh node velocities are assumed to be constant within each time-step) and that the surface of each lateral face is defined by the bilinear interpolant of its four vertices. The outward-pointing normal to a lateral face $\boldsymbol{n}_t$ is therefore parallel to $(0, \boldsymbol{e}) \wedge (1, \boldsymbol{v})$, in which $\boldsymbol{e}$ is the anticlockwise-oriented edge of the triangle obtained from the intersection of the space-time element with the **x**-plane at $t \in [t^n, t^{n+1}]$ (see Figure 1) and $\boldsymbol{v}$ is the local mesh velocity, defined by linear interpolation of the velocities at the two vertices defining this edge. Since $\boldsymbol{v}$ at each spatial mesh node is assumed to be constant in time, it is simple to show that

$$
(0, \boldsymbol{e}) \wedge (1, \boldsymbol{v}) = (-\boldsymbol{n} \cdot \boldsymbol{v}, \boldsymbol{n}). \quad (7)
$$

   Furthermore, $\hat{\boldsymbol{n}}_t$ in Equation (6) is the unit vector normal to the surface of the space-time element, with direction $\boldsymbol{n}_t = (-\boldsymbol{n} \cdot \boldsymbol{v}, \boldsymbol{n})$ on the lateral faces and $\boldsymbol{n}_t = (\pm 1, \boldsymbol{0})$ on the triangular faces.

5
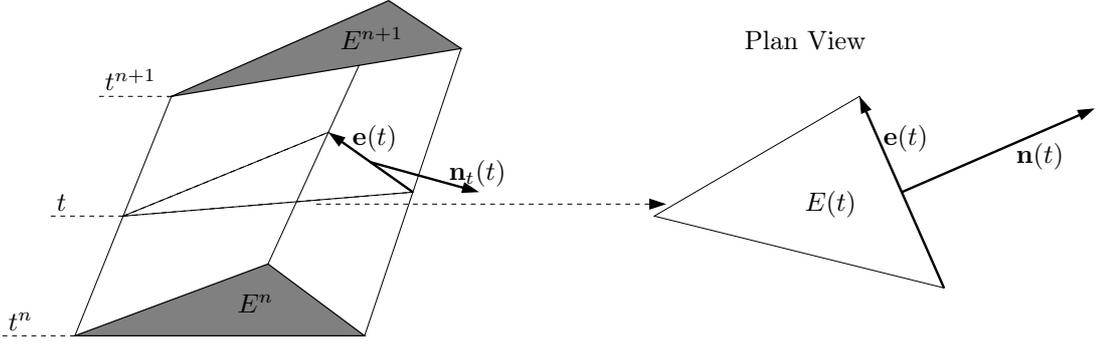
Figure 1: Geometry of the space-time element (in which the lateral faces are bilinear transformations of those of a right triangular prism) for the moving mesh algorithm.

3. Integrate over the two triangular faces and apply the trapezoidal rule in time to the lateral faces, giving

$$
\phi_{E_t} = \int_{E^{n+1}} u_h^{n+1} \, \mathrm{d}\Omega - \int_{E^n} u_h^n \, \mathrm{d}\Omega + \int_{\text{lateral}} \boldsymbol{f}_t \cdot \hat{\boldsymbol{n}}_t \, \mathrm{d}\Gamma_t
$$

$$
= \int_{E^{n+1}} u_h^{n+1} \, \mathrm{d}\Omega - \int_{E^n} u_h^n \, \mathrm{d}\Omega + \int_{t^n}^{t^{n+1}} \int_{\partial E(t)} (\boldsymbol{f}_h \cdot \hat{\boldsymbol{n}} - u_h \boldsymbol{v} \cdot \hat{\boldsymbol{n}}) \, \mathrm{d}\Gamma \, \mathrm{d}t
$$

$$
\approx \int_{E^{n+1}} u_h^{n+1} \, \mathrm{d}\Omega - \int_{E^n} u_h^n \, \mathrm{d}\Omega \tag{8}
$$

$$
+ \frac{\Delta t}{2} \left\{ \int_{\partial E^n} (\boldsymbol{f}(u_h^n) - u_h^n \boldsymbol{v}) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma + \int_{\partial E^{n+1}} (\boldsymbol{f}(u_h^{n+1}) - u_h^{n+1} \boldsymbol{v}) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma \right\}
$$

$$
= \int_{E^{n+1}} u_h^{n+1} \, \mathrm{d}\Omega - \int_{E^n} u_h^n \, \mathrm{d}\Omega + \frac{\Delta t}{2} (\phi_E^n + \phi_E^{n+1}),
$$

in which

$$
\phi_E = \int_E (\nabla \cdot \boldsymbol{f}_h - \nabla \cdot (u_h \boldsymbol{v})) \, \mathrm{d}\Omega = \int_{\partial E} (\boldsymbol{f}_h - u_h \boldsymbol{v}) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma. \tag{9}
$$

This is immediately recognisable as a conservative ALE formulation [23] (see Equation (5)), as has already been applied within the RD framework in [11, 13]. This formulation may also be derived in terms of mappings between the meshes at the two time levels [19]. This is a second-order accurate approximation of the space-time element residual, as long as the spatial residuals, ϕ_E^n and ϕ_E^{n+1}, are computed to be at least one order more accurate than the discretisation itself.

4. Distribute the space-time element residual, ϕ_{E_t} in Equation (8), to the six vertices of the element in a conservative manner, i.e. the fractions of the residual sent to

6

vertex with spatial index i at time levels n and $n+1$ are defined as

$$\phi_{i,n}^E = \beta_{i,n}^E \phi_{E_t} \quad \text{and} \quad \phi_{i,n+1}^E = \beta_{i,n+1}^E \phi_{E_t} \,, \tag{10}$$

in which the distribution coefficients are defined such that

$$\sum_{i \in E} \beta_{i,n}^E + \sum_{i \in E} \beta_{i,n+1}^E = 1 \,. \tag{11}$$

5. Integrate across the discontinuity in time, to produce additional residuals which, since the initial spatial mesh for the new time-step is constrained to be the same as the final spatial mesh at the previous time-step, can be viewed as space-time residuals evaluated in the limit $\Delta t \to 0$, $i.e.$

$$\psi_E = \lim_{\Delta t \to 0} \int_{\partial E_t} \boldsymbol{f}_t \cdot \hat{\boldsymbol{n}}_t \, \mathrm{d}\Gamma_t = \int_E [u_h^n] \, \mathrm{d}\Omega = \frac{|E^n|}{3} \sum_{i \in E} [u_i^n] \,, \tag{12}$$

where $|E^n|$ is the area of the triangular element E^n, $[u^n] = u^{n^+} - u^{n^-}$ (in which $u^{n^\pm} = \lim_{t \to (t^n)^\pm} u_h$) represents the jump across the triangular face and piecewise linear variation of u_h in space has been assumed. The $\pm$ superscripts are suppressed in the majority of what follows: within a space-time element n and $n+1$ implicitly indicate n^+ and $(n+1)^-$, respectively, and ψ_E is evaluated at time level n.

Distribute this residual to the three vertices of the element E^n in a conservative manner: the simple vertex-centred distribution

$$\psi_{i,n}^E = \frac{|E^n|}{3} [u_i^n] \,, \tag{13}$$

was shown in [10, 33, 34] to have all the properties one might ask of a residual distribution scheme, and this is the formulation that is used throughout this article. No modification is required to accommodate the moving meshes. We emphasise that there is a distinction between Equation (13) and a pure Galerkin distribution of the interface residual. In fact, the distribution defined by (13) is purely upwind in time. Furthermore, the spatial distribution of the residual (12) is derived from applying mass lumping to the contribution of the integral of the time derivative to the space-time residual [34]. This leads to a weak, pointwise, imposition of the solution at the previous time-step. The proof in Appendix B demonstrates that this choice will not produce spurious numerical oscillations.

6. Solve the nonlinear algebraic system derived by distributing the residuals, (10) and (13), to the vertices of the space-time elements:

$$\sum_{E \in D_i} \left(\phi_{i,n}^E + \psi_{i,n}^E \right) = 0$$
$$\sum_{E \in D_i} \phi_{i,n+1}^E = 0 \tag{14}$$

$\forall i \in \Omega_h$ at each time-step.

It remains to evaluate the spatial residual, ϕ_E in (8), and the distribution coefficients, β_i^E in (10).

2.1.2. Evaluating the Residual

Since both u_h and $\boldsymbol{v}$ are assumed to vary linearly along each edge of the spatial mesh, ϕ_E in (9) can be evaluated exactly using Simpson's rule, for appropriate fluxes, leading to

$$
\begin{aligned}
\phi_E &= \int_{\partial E} \left(\boldsymbol{f}(u_h) - u_h \boldsymbol{v} \right) \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma \\
&= \sum_{i \in E} \left(-\frac{1}{2}\overline{\boldsymbol{a}} \cdot \boldsymbol{n}_i + \frac{1}{2} \left(\boldsymbol{v}_i \cdot \boldsymbol{n}_i - \frac{1}{3} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_k - \frac{1}{3} \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_j \right) \right) u_i,
\end{aligned}
\tag{15}
$$

in which ϕ_E can be evaluated at any time t and j, k are the vertices of E which are not i. It is assumed here that a conservative linearisation [35] is available for the PDE (1) and that $\overline{\boldsymbol{a}}$ is the corresponding averaged velocity, *e.g.* for divergence-free scalar advection this is the arithmetic mean of the vertex values.

The form of the residual is simplified somewhat if the trapezoidal rule (an approximation which retains second-order accuracy) is used to evaluate the integrals along the mesh edges, giving

$$
\phi_E \approx \sum_{i \in E} \left(-\frac{1}{2} \left(\overline{\boldsymbol{a}} - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_i \right) u_i.
\tag{16}
$$

When $\nabla \cdot \boldsymbol{v} = 0$ the two forms for the residual, (15) and (16), are equivalent for linearly varying u and $\boldsymbol{v}$. However, the terms corresponding to the mesh movement cannot be incorporated within the conservative linearisation in either formulation because the mesh velocity field does not, in general, satisfy $\nabla \cdot \boldsymbol{v} = 0$. Previous implementations [13] have included an element-averaged value of $\boldsymbol{v}$ in the linearisation but added a source term to compensate for the mesh velocity-divergence.

Having evaluated ϕ_E, it is now possible to write the discrete residual over the space-time element in the form

$$
\phi_{E_t} = \sum_{i \in E} k_{i,n} u_i^n + \sum_{i \in E} k_{i,n+1} u_i^{n+1}.
\tag{17}
$$

where, from (15),

$$
\begin{aligned}
k_{i,n} &= -\frac{\Delta t}{4}\overline{\boldsymbol{a}}^n \cdot \boldsymbol{n}_i^n \\
&\quad + \frac{\Delta t}{4} \left(\boldsymbol{v}_i \cdot \boldsymbol{n}_i^n - \frac{1}{3} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_k^n - \frac{1}{3} \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_j^n \right) - \frac{|E^n|}{3}, \\
k_{i,n+1} &= -\frac{\Delta t}{4}\overline{\boldsymbol{a}}^{n+1} \cdot \boldsymbol{n}_i^{n+1} \\
&\quad + \frac{\Delta t}{4} \left(\boldsymbol{v}_i \cdot \boldsymbol{n}_i^{n+1} - \frac{1}{3} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_k^{n+1} - \frac{1}{3} \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_j^{n+1} \right) + \frac{|E^{n+1}|}{3},
\end{aligned}
\tag{18}
$$

$\boldsymbol{n}_i$ being the outward-pointing normal vector (in space only) opposite node i, at the specified time level, scaled by the length of the edge opposite node i. Note that $\boldsymbol{v}$ is assumed to remain unchanged during a time-step so the superscripts denoting the time

level have been dropped. If, as in (16), the residuals are approximated instead using the trapezoidal rule, then

$$
\begin{aligned}
k_{i,n} &= -\frac{\Delta t}{4} \left(\overline{\boldsymbol{a}}^n - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_i^n - \frac{|E^n|}{3}\,, \\
k_{i,n+1} &= -\frac{\Delta t}{4} \left(\overline{\boldsymbol{a}}^{n+1} - \boldsymbol{v}_i \right) \cdot \boldsymbol{n}_i^{n+1} + \frac{|E^{n+1}|}{3}\,.
\end{aligned}
\tag{19}
$$

In fact, for both approximations,

$$
\sum_{i \in E} k_{i,n} + \sum_{i \in E} k_{i,n+1} = 0\,,
\tag{20}
$$

so either (18) or (19) can be used to define the inflow parameters in any of the standard residual distribution schemes (proof provided in Appendix C).

There are two important consequences of (20).

1. The space-time N scheme (detailed in [2] and outlined in the next section) may be used in its standard form with the new definitions of the 'inflow' parameters $k_{i,n}$ and $k_{i,n+1}$ taken from either (18) or (19). It also allows us to define the quantities

$$
\begin{aligned}
k_{i,n}^+ &= \max(0, k_{i,n})\,, & k_{i,n}^- &= \min(0, k_{i,n})\,, \\
k_{i,n+1}^+ &= \max(0, k_{i,n+1})\,, & k_{i,n+1}^- &= \min(0, k_{i,n+1})\,, \\
N_t &= \frac{1}{\sum_{i \in E} k_{i,n}^+ + \sum_{i \in E} k_{i,n+1}^+}\,,
\end{aligned}
\tag{21}
$$

which will be used in the distribution schemes described in the next section.

2. Any conservative residual distribution scheme, *i.e.* one for which

$$
\sum_{i \in E} \beta_{i,n}^E + \sum_{i \in E} \beta_{i,n+1}^E = 1\,,
\tag{22}
$$

will satisfy a discrete geometric conservation law (DGCL) [16], *i.e.* the trivial solution $u \equiv$ constant is preserved when the mesh is moved.

This is easy to show because, for all of the schemes defined in Section 2.1.3, Equations (17) (or (6)) and (21) give

$$
u^{n+1} = u^{n^+} = u^{n^-} = K \quad \Rightarrow \quad \phi_{i,n}^E = \psi_{i,n}^E = \phi_{i,n+1}^E = 0\,.
\tag{23}
$$

Thus $u^{n+1} = K$ satisfies (14) and is a possible solution at the new time level. Moreover, it is the solution that would be found by the pseudo-time-stepping iteration described in Section 2.1.4.

2.1.3. Evaluating the Distribution Coefficients

The typical design requirements for RD schemes include [36]

- *positivity*, which warrants that the numerical approximations are free of spurious oscillations;

- *linearity preservation*, which ensures that a $(k-1)^{\text{th}}$-order polynomial representation of u_h leads to a kth-order accurate scheme [26];

- *conservation*, which guarantees that discontinuities are captured correctly and that the approximation converges to a weak solution of (1) as the mesh is refined;

- *compactness*, which is primarily for computational efficiency and requires that the element residual be distributed to its own vertices only; and

- *continuous dependence* of the coefficients on both the solution and the advection velocity, which enhances the iterative convergence of any algebraic solver applied to (14).

The schemes are also often designed to be *upwind*, *i.e.* the discretised model propagates information in the same direction and at the same velocity as its non-discretised counterpart. Since this provides positivity and improved numerical stability relative to centred schemes, we choose to consider upwind schemes in this work, though other successful RD schemes have been developed [2].

We briefly describe here the space-time upwind schemes that are the focus of this investigation. As before, the use of n and $n+1$ within an element residual implies n^+ and $(n+1)^-$.

- **The space-time N** (STN) scheme is here defined in a form suitable for the conservative residual distribution (CRD) formulation [37] (which will later be used in the approximation of the shallow water equations), *i.e.*

$$u_t^{\text{in}} = N_t \left(\sum_{j \in E} k_{j,n}^+ u_j^n + \sum_{j \in E} k_{j,n+1}^+ u_j^{n+1} - \phi_{E_t} \right),$$

$$\left(\phi_{i,n}^E \right)^N = k_{i,n}^+ \left(u_i^n - u_t^{\text{in}} \right), \qquad \left(\phi_{i,n+1}^E \right)^N = k_{i,n+1}^+ \left(u_i^{n+1} - u_t^{\text{in}} \right),$$

(24)

using the inflow parameter definitions given by Equation (21). This is a linear scheme that has all the desired properties except linearity preservation.

A proof of the unconditional positivity of this scheme on the space-time elements generated by the mesh movement is provided in Appendix B. In addition we note that positivity is only formally guaranteed if (17) is used for the computation of ϕ_{E_t}, though it is not always possible or desirable to evaluate the residual in this way. Nevertheless, oscillation-free behaviour is also often observed in computational experiments [2] for the CRD scheme when an appropriate quadrature rule is used to evaluate ϕ_{E_t} directly from (6). The advantage of the CRD formulation is that it becomes simple to guarantee conservation, even when a conservative linearisation is not available or is prohibitively expensive to compute.

10

- **The space-time LDA** (STLDA) [37] scheme is defined as

$$\left(\phi_{i,n}^E\right)^{LDA} = k_{i,n}^+ N_t\,\phi_{E_t}\,, \qquad \left(\phi_{i,n+1}^E\right)^{LDA} = k_{i,n+1}^+ N_t\,\phi_{E_t}\,, \tag{25}$$

which is also a linear scheme and has all the desired properties except positivity.

- By Godunov's theorem [24, 38], only nonlinear schemes can satisfy the conditions for both positivity and linearity preservation. Therefore, in order to obtain all of the desired properties, nonlinear schemes must be considered.

  The space-time blended (STB) scheme combines two linear schemes, the STN and STLDA schemes, for example, through a nonlinear blending coefficient θ,

$$\begin{aligned}
\left(\phi_{i,n}^E\right)^B &= \theta\left(\phi_{i,n}^E\right)^N + (1-\theta)\left(\phi_{i,n}^E\right)^{LDA}, \\
\left(\phi_{i,n+1}^E\right)^B &= \theta\left(\phi_{i,n+1}^E\right)^N + (1-\theta)\left(\phi_{i,n+1}^E\right)^{LDA}.
\end{aligned} \tag{26}$$

The blending coefficient determines how 'well' the required properties, especially positivity, are satisfied. The choice adopted in this article, taken from [39], is

$$\theta = \frac{|\phi_{E_t}|}{\sum_{i\in E}\left|\left(\phi_{i,n}^E\right)^N\right| + \sum_{i\in E}\left|\left(\phi_{i,n+1}^E\right)^N\right|}\,. \tag{27}$$

Note also that the STN scheme can be written as the STLDA scheme plus additional dissipation [40], so the STB scheme may be considered in these terms (as it can be in the space-only case).

2.1.4. Solving the Discrete System

Once the distribution of the residual has been determined, the system (14) is solved using the following simple pseudo-time-stepping algorithm:

$$\begin{pmatrix} u_i^{n^+} \\ u_i^{(n+1)^-} \end{pmatrix}_{m+1} = \begin{pmatrix} u_i^{n^+} \\ u_i^{(n+1)^-} \end{pmatrix}_m - \tau_i \sum_{E\in D_i} \begin{pmatrix} \phi_{i,n}^E + \psi_{i,n}^E \\ \phi_{i,n+1}^E \end{pmatrix}_m, \tag{28}$$

where i is the spatial mesh node index. The subscript m is the index for the pseudo-time-step and the iteration is fully explicit. In order to provide an iteration which is positive at each step (and hence will not diverge), the local pseudo-time-step τ_i is chosen to satisfy

$$\tau_i \le \frac{1}{\sum_{E\in D_i} k_{i,n}^+ + \sum_{E\in D_i} k_{i,n+1}^+}\,. \tag{29}$$

In this work τ_i is chosen to be CFL_τ times its maximum value, where $\mathrm{CFL}_\tau \le 1$. The value of τ is allowed to change from node to node in order to accelerate convergence of the iteration.

2.2. Nonlinear Systems

Consider now the nonlinear hyperbolic system of conservation laws,

$$\frac{\partial U}{\partial t} + \nabla \cdot \boldsymbol{F} = 0 \qquad \text{or} \qquad \frac{\partial U}{\partial t} + \boldsymbol{\mathcal{A}}(U) \cdot \nabla U = 0 \,, \tag{30}$$

where $\boldsymbol{\mathcal{A}}(U) = [\mathcal{A}_x, \mathcal{A}_y] = [\partial F_x/\partial U, \partial F_y/\partial U] = \partial \boldsymbol{F}/\partial U$ is the wave-speed tensor. Following the same approximation procedure as for the scalar equation (Section 2.1.1) gives a space-time residual of

$$
\begin{aligned}
\Phi_{E_t} \approx &\int\limits_{E^{n+1}} U_h^{n+1} \ \mathrm{d}\Omega - \int\limits_{E^n} U_h^n \ \mathrm{d}\Omega \\
&+ \frac{\Delta t}{2} \left(\int\limits_{\partial E^n} \left(\boldsymbol{F}(U_h^n) - U_h^n \boldsymbol{v} \right) \cdot \hat{\boldsymbol{n}} \ \mathrm{d}\Gamma + \int\limits_{\partial E^{n+1}} \left(\boldsymbol{F}(U_h^{n+1}) - U_h^{n+1} \boldsymbol{v} \right) \cdot \hat{\boldsymbol{n}} \ \mathrm{d}\Gamma \right) .
\end{aligned}
\tag{31}
$$

2.2.1. Evaluating the Residual

The residuals in (31) may be approximated as

$$\Phi_{E_t} \approx \sum_{i \in E} \mathcal{K}_{i,n} U_i^n + \sum_{i \in E} \mathcal{K}_{i,n+1} U_i^{n+1} \tag{32}$$

in which the $\mathcal{K}$ matrices are defined by, *cf.* Equation (18),

$$
\begin{aligned}
\mathcal{K}_{i,n} \approx &-\frac{\Delta t}{4} \overline{\boldsymbol{\mathcal{A}}}^n \cdot \boldsymbol{n}_i^n \\
&+ \frac{\Delta t}{4} \left(\boldsymbol{v}_i \cdot \boldsymbol{n}_i^n - \frac{1}{3} (\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot \boldsymbol{n}_k^n - \frac{1}{3} (\boldsymbol{v}_k - \boldsymbol{v}_i) \cdot \boldsymbol{n}_j^n \right) \mathcal{I} - \frac{|E^n|}{3} \mathcal{I} \,, \\
\mathcal{K}_{i,n+1} \approx &-\frac{\Delta t}{4} \overline{\boldsymbol{\mathcal{A}}}^{n+1} \cdot \boldsymbol{n}_i^{n+1} \\
&+ \frac{\Delta t}{4} \left(\boldsymbol{v}_i \cdot \boldsymbol{n}_i^{n+1} - \frac{1}{3} (\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot \boldsymbol{n}_k^{n+1} - \frac{1}{3} (\boldsymbol{v}_k - \boldsymbol{v}_i) \cdot \boldsymbol{n}_j^{n+1} \right) \mathcal{I} + \frac{|E^{n+1}|}{3} \mathcal{I} \,,
\end{aligned}
\tag{33}
$$

where $\mathcal{I}$ is the identity matrix and $\overline{\boldsymbol{\mathcal{A}}}$ represents an averaged state of the flux Jacobian $\boldsymbol{\mathcal{A}}$. In the simpler situation where the trapezoidal rule is used to approximate the spatial integrals, *cf.* Equation (19), the $\mathcal{K}$ matrices would be defined by

$$
\begin{aligned}
\mathcal{K}_{i,n} &\approx -\frac{\Delta t}{4} \overline{\boldsymbol{\mathcal{A}}}^n \cdot \boldsymbol{n}_i^n + \frac{\Delta t}{4} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^n \mathcal{I} - \frac{|E^n|}{3} \mathcal{I} \,, \\
\mathcal{K}_{i,n+1} &\approx -\frac{\Delta t}{4} \overline{\boldsymbol{\mathcal{A}}}^{n+1} \cdot \boldsymbol{n}_i^{n+1} + \frac{\Delta t}{4} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^{n+1} \mathcal{I} + \frac{|E^{n+1}|}{3} \mathcal{I} \,.
\end{aligned}
\tag{34}
$$

For some systems, including the shallow water equations, Φ_{E_t} and $\overline{\boldsymbol{\mathcal{A}}}$ can be derived from a conservative linearisation [35], so that (32) is an exact evaluation of the discrete residual. This approach is described in more detail in [10]. However, if the conservative linearisation is not known or its implementation is not practical, it is still possible to achieve conservation by evaluating the space-time element residual directly from (31) via

a higher-order quadrature rule – this is one of the concepts behind the CRD schemes [41, 42]. The $\mathcal{K}$ matrices in (34), which are required by the distribution schemes, are then evaluated using appropriate averaged states which depend on the time level at which $\overline{\mathcal{A}}$ is being evaluated. In this work, for the shallow water equations, $\overline{\mathcal{A}}^n$ and $\overline{\mathcal{A}}^{n+1}$ are evaluated using the arithmetic means of the spatial element vertex values of the primitive variables, depth and velocity, at their respective time levels [10].

2.2.2. Evaluating the Distribution Coefficients

Assuming that the inflow matrices defined in (34) are diagonalisable, we have $\mathcal{K}_{i,n} = \mathcal{R}\mathcal{D}\mathcal{R}^{-1}$, where $\mathcal{D}$ is the diagonal matrix of eigenvalues, $\mathcal{R}^{-1}$ is the matrix of the left and $\mathcal{R}$ of the right eigenvectors. Defining $\mathcal{D}^{\pm} = \frac{1}{2}\left(\mathcal{D} \pm |\mathcal{D}|\right)$ with $|\mathcal{D}|$ denoting the absolute values of the entries, we can generalise the inflow parameters defined in (21) to the matrices

$$
\begin{aligned}
\mathcal{K}_{i,n}^{+} &= \left(\mathcal{R}\mathcal{D}^{+}\mathcal{R}^{-1}\right)_{i,n}, \quad \mathcal{K}_{i,n+1}^{+} = \left(\mathcal{R}\mathcal{D}^{+}\mathcal{R}^{-1}\right)_{i,n+1}, \\
\mathcal{K}_{i,n}^{-} &= \left(\mathcal{R}\mathcal{D}^{-}\mathcal{R}^{-1}\right)_{i,n}, \quad \mathcal{K}_{i,n+1}^{-} = \left(\mathcal{R}\mathcal{D}^{-}\mathcal{R}^{-1}\right)_{i,n+1}, \\
\mathcal{N}_t &= \left(\sum_{i \in E} \mathcal{K}_{i,n}^{+} + \sum_{i \in E} \mathcal{K}_{i,n+1}^{+}\right)^{-1}.
\end{aligned}
\tag{35}
$$

These matrices are used to define upwind RD schemes along the lines of the scalar case (24)–(26). Note that the presence of the volume term in the space-time residual, which corresponds to the time derivative, ensures that the eigenvalues of the $\mathcal{K}$ parameters cannot all vanish simultaneously [9, 43]. Consequently, $\mathcal{N}_t$ can always be evaluated without any need for regularisation.

- **The space-time N** (STN) scheme for systems [2, 37] is defined as

$$
U_t^{\text{in}} = \mathcal{N}_t \left(\sum_{j \in E} \mathcal{K}_{j,n}^{+} U_j^n + \sum_{j \in E} \mathcal{K}_{j,n+1}^{+} U_j^{n+1} - \Phi_{E_t}\right),
\tag{36}
$$
$$
\left(\Phi_{i,n}^{E}\right)^{N} = \mathcal{K}_{i,n}^{+}\left(U_i^n - U_t^{\text{in}}\right), \qquad \left(\Phi_{i,n+1}^{E}\right)^{N} = \mathcal{K}_{i,n+1}^{+}\left(U_i^{n+1} - U_t^{\text{in}}\right).
$$

  The CRD formulation is again used here, where Φ_{E_t} is computed by integrating Equation (31) with an appropriate quadrature rule. If a conservative linearisation exists, and (32) is used to evaluate Φ_{E_t}, then an alternative form must be used for U in (32) and (36) to retain conservation: this is described in more detail for the shallow water equations in [10].

- **The space-time LDA** (STLDA) scheme for systems [2, 37] is defined as

$$
\left(\Phi_{i,n}^{E}\right)^{LDA} = \mathcal{K}_{i,n}^{+}\mathcal{N}_t\,\Phi_{E_t}, \qquad \left(\Phi_{i,n+1}^{E}\right)^{LDA} = \mathcal{K}_{i,n+1}^{+}\mathcal{N}_t\,\Phi_{E_t}.
\tag{37}
$$

- **The space-time blended** (STB) scheme for systems [2] is still defined as a combination of STN and the STLDA schemes,

$$
\begin{aligned}
\left(\Phi_{i,n}^{E}\right)^{B} &= \Theta\left(\Phi_{i,n}^{E}\right)^{N} + (\mathcal{I} - \Theta)\left(\Phi_{i,n}^{E}\right)^{LDA}, \\
\left(\Phi_{i,n+1}^{E}\right)^{B} &= \Theta\left(\Phi_{i,n+1}^{E}\right)^{N} + (\mathcal{I} - \Theta)\left(\Phi_{i,n+1}^{E}\right)^{LDA}.
\end{aligned}
\tag{38}
$$

13

The matrix blending parameter Θ can be computed in a number of different ways [44, 45]. The simplest generalisation of (27) is

$$\Theta_1 = \text{diag}\left(\frac{|\Phi_{E_t}|}{\sum_{i\in E}\left|\left(\Phi_{i,n}^E\right)^N\right| + \sum_{i\in E}\left|\left(\Phi_{i,n+1}^E\right)^N\right|}\right), \qquad (39)$$

where the division should be understood as an element-wise operation. Another possibility, proposed in [44] and used in this work, is to choose a particular direction $\boldsymbol{\xi} = (\xi_x, \xi_y)$ and compute the decomposition $\boldsymbol{\mathcal{A}}\cdot\boldsymbol{\xi} = \mathcal{R}_\xi \mathcal{D}_\xi \mathcal{R}_\xi^{-1}$. The blending (38) is then carried out on the "characteristic" residuals

$$\begin{aligned}
\Phi_{i,n}^N &= (\mathcal{R}_\xi^{-1})_n \Phi_{i,n}^N, & \Phi_{i,n+1}^N &= (\mathcal{R}_\xi^{-1})_{n+1} \Phi_{i,n+1}^N, \\
\Phi_{i,n}^{LDA} &= (\mathcal{R}_\xi^{-1})_n \Phi_{i,n}^{LDA}, & \Phi_{i,n+1}^{LDA} &= (\mathcal{R}_\xi^{-1})_{n+1} \Phi_{i,n+1}^{LDA},
\end{aligned} \qquad (40)$$

with the blending parameter computed as

$$\Theta_2 = \text{diag}\left(\frac{\left|\sum_{i\in E}\Phi_{i,n}^N + \sum_{i\in E}\Phi_{i,n+1}^N\right|}{\sum_{i\in E}\left|\Phi_{i,n}^N\right| + \sum_{i\in E}\left|\Phi_{i,n+1}^N\right|}\right), \qquad (41)$$

where we drop the superscript E to avoid clutter. Finally, we calculate the blended residuals based on the original variables by $\Phi_{i,n}^B = (\mathcal{R}_\xi)_n \Phi_{i,n}^B$ and $\Phi_{i,n+1}^B = (\mathcal{R}_\xi)_{n+1} \Phi_{i,n+1}^B$. In this work, in order to aid iterative convergence within each time-step, the blending parameter, Θ_2, is fixed after the first pseudo-time-step.

The treatment of the discontinuity in time is directly analogous to (13). The residual due to the discontinuity in time is given by

$$\Psi_E = \int_E [U_h^n]\,\mathrm{d}\Omega = \frac{|E^n|}{3}\sum_{i\in E}[U_i^n], \qquad (42)$$

and the corresponding contributions are defined as [33]

$$\Psi_i^E = \frac{|E^n|}{3}[U_i^n]. \qquad (43)$$

Combining this with the distribution of the element residuals leads to a system of nonlinear algebraic equations of the form

$$\begin{aligned}
\sum_{E\in D_i}\left(\Phi_{i,n}^E + \Psi_{i,n}^E\right) &= 0 \\
\sum_{E\in D_i}\Phi_{i,n+1}^E &= 0
\end{aligned} \qquad (44)$$

$\forall i \in \Omega_h$, which needs to be solved at each time-step. Boundary conditions are applied as described in [46].

2.2.3. Solving the Discrete System

In this work a simple pseudo-time-stepping algorithm is used to solve Equation (44):

$$\begin{pmatrix} U_i^{n^+} \\ U_i^{(n+1)^-} \end{pmatrix}_{m+1} = \begin{pmatrix} U_i^{n^+} \\ U_i^{(n+1)^-} \end{pmatrix}_m - \tau_i \sum_{E \in D_i} \begin{pmatrix} \Phi_{i,n}^E + \Psi_{i,n}^E \\ \Phi_{i,n+1}^E \end{pmatrix}_m . \tag{45}$$

The local pseudo-time-step τ_i is chosen to satisfy

$$\tau_i \leq \frac{1}{\sum_{E \in D_i} \varrho(\mathcal{K}_{i,n}^+) + \sum_{E \in D_i} \varrho(\mathcal{K}_{i,n+1}^+)} , \tag{46}$$

in which $\varrho(\mathcal{M})$ denotes the spectral radius of a given matrix $\mathcal{M}$, so $\varrho(\mathcal{K}_i^+) = \max \operatorname{diag} \mathcal{D}_i^+$, and the $\mathcal{K}_i$ are given by (34). As in the scalar case, $\mathrm{CFL}_\tau \leq 1$ is chosen. The constraint provided by (46) is chosen so that in the scalar case the iteration reduces to one which satisfies a provable discrete maximum principle.

2.3. Application to the Shallow Water System

The space-time RD framework has been applied to the frictionless shallow water equations with variable bed topography [10],

$$\frac{\partial U}{\partial t} + \nabla \cdot \boldsymbol{F}(U) + S(U) = 0 \qquad \text{on} \quad \Omega_t , \tag{47}$$

where, in two space dimensions,

$$U = \begin{bmatrix} d \\ du \\ dv \end{bmatrix} , \quad \boldsymbol{F} = \begin{bmatrix} F_x & F_y \end{bmatrix} = \begin{bmatrix} du & dv \\ du^2 + \frac{gd^2}{2} & duv \\ duv & dv^2 + \frac{gd^2}{2} \end{bmatrix} , \quad S = - \begin{bmatrix} 0 \\ gd\frac{\partial b(x,y)}{\partial x} \\ gd\frac{\partial b(x,y)}{\partial y} \end{bmatrix} , \tag{48}$$

in which d is the water height, $\boldsymbol{u} = (u, v)$ is the flow velocity and b is the height of the bed topography. The level of the free surface is defined as $\eta = d + b$. One of the main challenges of discretising this system is the retention of the hydrostatic balance property (the C-property [20]) satisfied by the underlying equations, within the framework of a conservative scheme. In [11] this is simplified by approximating a "pre-balanced" form of the equations [47]: in this work we simply modify the mass balance equation, as described in Section 2.3.1.

The CRD formulation, which guarantees conservation, is also applied [41], using the arithmetic means $\overline{d}$, $\overline{u}$, $\overline{v}$ of the values at the vertices of the triangular spatial elements as the averaged states in (34). The positivity of the STN scheme is formally lost, but we have typically observed oscillation-free solutions in our numerical experiments. There is no guarantee that this will always be the case, as shown in [48], in which a blending coefficient is derived using more rigorous arguments. In order to preserve hydrostatic balance, as described in the following section, the element-averaged free surface and bed levels, $\overline{\eta}$ and $\overline{b}$ respectively, must be evaluated in the same way as $\overline{d}$ wherever they appear.

2.3.1. Preserving Hydrostatic Balance

The shallow water system (47)–(48) is, by construction, in hydrostatic balance. Over a general non-flat bed, however, care is needed to ensure exact preservation of the C-property by the numerical algorithm. Some progress has been made on the development of well-balanced space-time discontinuous Galerkin schemes for the shallow water equations on moving meshes [49].

It was proved in [42] that, when the water height d and the bottom topography b are both assumed to be linearly varying, the linearity preserving CRD schemes satisfy the C-property [42, 50, 51] on *fixed* meshes, as long as the contribution of the source term to the augmented space-time element residual, $\Phi_{E_t} + \Phi_{E_t}^b$ at a given time level is computed as

$$\Phi_{E_t}^b = \Delta t \frac{g\overline{d}}{2} \sum_{i \in E} \begin{bmatrix} 0 \\ b_i \boldsymbol{n}_i \end{bmatrix} \approx \int\limits_{t^n}^{t^{n+1}} \int\limits_{E(t)} S(U) \, \mathrm{d}\Omega \, \mathrm{d}t \,, \tag{49}$$

where $\boldsymbol{n}_i$ is the outward normal to the edge opposite vertex i scaled by the length of the edge. In order for the STN scheme (36) (and the corresponding STB schemes) to satisfy hydrostatic balance, a slightly modified version of (36) must be applied. For fixed bed topography, $\frac{\partial d}{\partial t} = \frac{\partial \eta}{\partial t}$, so $U = [d, du, dv]^T$ can be replaced by $V = [\eta, du, dv]^T$ in the CRD formulation, using the arithmetic mean $\overline{\eta}$ in the approximation of the spatial integral of the time derivative term. This leads to

$$V_t^{\mathrm{in}} = \mathcal{N}_t \left(\sum_{j \in E} \mathcal{K}_{j,n}^+ V_j^n + \sum_{j \in E} \mathcal{K}_{j,n+1}^+ V_j^{n+1} - \Phi_{E_t} - \Phi_{E_t}^b \right) ,$$
$$\left(\Phi_{i,n}^E \right)^N = \mathcal{K}_{i,n}^+ \left(V_i^n - V_t^{\mathrm{in}} \right) , \qquad \left(\Phi_{i,n+1}^E \right)^N = \mathcal{K}_{i,n+1}^+ \left(V_i^{n+1} - V_t^{\mathrm{in}} \right) . \tag{50}$$

A proof that this modified N scheme satisfies the C-property on fixed meshes is given in [10].

The proofs that the C-property is satisfied generalise straightforwardly to moving meshes because the STN and STLDA schemes take precisely the same general forms, (36) and (37) respectively: only the definitions of the $\mathcal{K}$ matrices and the residuals Φ are different. The forms of the $\mathcal{K}$ matrices do not affect the proofs, so it only remains to construct the form of $\Phi_{E_t}^b$ (a generalisation of Equation (49)) which, when combined with the residual in Equations (31) and (42), satisfies

$$\Psi_E + \Phi_{E_t} + \Phi_{E_t}^b = 0 \qquad \text{when} \qquad V = [\eta, du, dv]^T = [\mathrm{const}, 0, 0]^T . \tag{51}$$

In order to satisfy the C-property on a moving mesh, the contribution of the source term to the space-time element residual should be computed in our schemes as

$$\Phi_{E_t}^b = \frac{\Delta t}{2} \left(\frac{g\overline{d}^n}{2} \sum_{i \in E} \begin{bmatrix} 0 \\ b_i^n \boldsymbol{n}_i^n \end{bmatrix} + \frac{g\overline{d}^{n+1}}{2} \sum_{i \in E} \begin{bmatrix} 0 \\ b_i^{n+1} \boldsymbol{n}_i^{n+1} \end{bmatrix} \right) . \tag{52}$$

The details of the proof are given in Appendix A.

As a consequence of choosing to approximate the residual using $\frac{\partial \eta}{\partial t}$ instead of $\frac{\partial d}{\partial t}$, the resulting numerical scheme locally conserves the quantity

$$\int \eta_h \, \mathrm{d}\Omega - \int b(\mathbf{x}) \, \mathrm{d}\Omega = \int d_h \, \mathrm{d}\Omega + \int b_h - b(\mathbf{x}) \, \mathrm{d}\Omega \tag{53}$$

at convergence of the pseudo-time-stepping iteration described in Section 2.2.3. Here η_h, d_h and b_h are the piecewise linear approximations obtained by interpolating the nodal values and $b(\mathbf{x})$ is the exact bed topography. This is equivalent to mass conservation ($\int d_h \, d\Omega$ remains constant) in the special case where the mesh movement over the bed topography satisfies

$$|E^{n+1}|\overline{b}^{n+1} - |E^n|\overline{b}^n - \frac{\Delta t}{2} \sum_{i \in E} \left(b_i^{n+1} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^{n+1} + b_i^n \boldsymbol{v}_i \cdot \boldsymbol{n}_i^n \right) = 0 \,. \tag{54}$$

The left-hand side of this equation is simply the residual of the equation $b_t = 0$ approximated on the moving mesh. Equation (54) is satisfied when the mesh is fixed or when the bed topography is constant.

2.4. The Moving Mesh Algorithm

The mesh movement is implemented in a simple manner by interleaving the pseudo-time-stepping iteration used to solve the systems of equations, given by (28) in the scalar case and (45) for nonlinear systems, with another iteration which updates the node positions at the new time level. It would be possible to select a different approach to moving the mesh, *e.g.* the variational approaches of Huang and Russell, described in the review paper [18] and the references therein, but instead of using the techniques they propose we observe that it is natural to embed the relaxation towards an equidistributed mesh within the existing nonlinear iteration. In this paper we show that r-adaptivity is potentially beneficial, and future work will seek to improve how to link the mesh movement with the PDE solver.

In this work, one iteration of the mesh movement consists of moving each mesh node to a weighted average of the positions of the centroids of its adjacent elements, *i.e.*

$$(\boldsymbol{\delta}_i^{n+1})_m = \frac{\sum_{E \in D_i} W_E[(\boldsymbol{x}_E^{n+1})_m - (\boldsymbol{x}_i^{n+1})_m]}{\sum_{E \in D_i} W_E} \qquad \text{where} \qquad \boldsymbol{x}_E = \frac{1}{3} \sum_{j \in E} \boldsymbol{x}_j \,. \tag{55}$$

At this stage, it is possible to relax the nodal movement by updating the nodal positions using

$$(\boldsymbol{x}_i^{n+1})_{m+1} = (\boldsymbol{x}_i^{n+1})_m + \mu (\boldsymbol{\delta}_i^{n+1})_m \,, \tag{56}$$

having initialised the iteration with $(\boldsymbol{x}_i^{n+1})_0 = \boldsymbol{x}_i^n$. For simplicity, μ is chosen to have the same value for all nodes. However, during each iteration the displacements of each node may be restricted geometrically, if necessary, to avoid mesh tangling. A simple limit, which guarantees no tangling and depends only on the original mesh is given by retaining the direction of the displacement, but adjusting $\boldsymbol{x}_i^{n+1}$ for each node, so that [6, 52]

$$|(\boldsymbol{x}_i^{n+1})_{m+1} - \boldsymbol{x}_i^n| \leq \min_{E \in D_i} \left(\frac{|E^n|}{\max l^{E^n}} \right) \,, \tag{57}$$

where l^E are the lengths of the edges of the triangular element E. Boundary nodes are restricted to remain on the boundary by ignoring the component of the movement perpendicular to the boundary. A more sophisticated treatment of boundary nodes will be necessary for concave domains, so that this projection does not induce mesh tangling.

Note also that the constraint (57) inhibits the rotation of spatial mesh elements and, in all cases investigated here, has ensured that the distortion of the space-time element does not cause it to collapse. If this did occur then a smaller time-step would be needed to reduce the mesh movement and avoid tangling.

The mesh velocity corresponding to the movement of each node, required for the subsequent residual distribution iteration, is calculated using the displacement relative to the nodal positions at the start of the iteration and given by

$$(\boldsymbol{v}_i)_{m+1} = \frac{(\boldsymbol{x}_i^{n+1})_{m+1} - \boldsymbol{x}_i^n}{\Delta t} . \tag{58}$$

There are many possible choices for the element weights, W_E (see, for example, [15, 18] and references therein). In this work, following [11], they are chosen so that nodes are moved towards regions of higher local solution gradients and Laplacians. The specific choice made here is given by

$$W_E(\chi) = \sqrt{1 + \alpha(\max(||\nabla\chi||^*, ||\nabla^2\chi||^*))^2} , \tag{59}$$

where the $||\cdot||^*$ are defined by

$$||\nabla\chi||^* = \min\left(1, \frac{||\nabla\chi||_2}{\beta \max ||\nabla\chi||_2}\right) \qquad \text{and} \qquad ||\nabla^2\chi||^* = \min\left(1, \frac{||\nabla^2\chi||_2}{\gamma \max ||\nabla^2\chi||_2}\right), \tag{60}$$

in which $||\cdot||_2$ indicates a discrete l^2 norm. For the scalar equations, $\chi = u$, and for the shallow water equations, $\chi = \eta$. The quantities α, β and γ are free constant parameters, chosen according to the test case. These weights may be scaled locally by the initial element sizes if a predetermined mesh distribution, for example one in which the nodes are clustered around complex boundary geometry, is to be retained.

On its own, this weighted averaging can produce distorted meshes which are detrimental to the quality of the approximate solution, so the algorithm also makes use of a Laplacian smoothing step, which is achieved by taking $W_E \equiv 1$ in Equation (55). The relaxation parameter μ can be chosen separately in this step.

2.4.1. Implementation Details

Given an initial mesh and initial conditions and before initiating the time-stepping, the complete algorithm first adapts the mesh to the initial conditions. In this work a fixed number of iterations (N_{init}) is used, each of which consists of the following steps.

1. Update the mesh node positions, using one iteration of (55)/(56), with the chosen element weights W_E, limiting the node movement according to (57) to avoid mesh tangling.

2. Smooth the moved mesh using one relaxed Laplacian smoothing iteration, (55) with $W_E \equiv 1$, and limiting the node movement according to (57) to avoid mesh tangling.

3. Recompute the initial conditions and (if required) the bed topography at the new mesh node positions.

18

Once the initial conditions have been adapted to, each time-step involves both mesh movement and pseudo-time-stepping iterations, which are initially interleaved.

1. The following steps are repeated until the stopping criteria for the adaptation are satisfied.

   (a) Update the mesh node positions at the new time level, using one iteration of (55)/(56), with the chosen element weights W_E, and limiting the node movement according to (57) to avoid mesh tangling.

   (b) Smooth the mesh at the new time level using one relaxed Laplacian smoothing iteration, (55) with $W_E \equiv 1$, and limiting the node movement according to (57) to avoid mesh tangling.

   (c) If required, recompute the bed topography at the new mesh node positions.

   (d) Carry out one iteration of the pseudo-time-stepping, (28) or (45), using the residuals calculated on the current distorted space-time prismatic elements and the chosen distribution scheme.

2. Fix the nodal positions for the new time level, but continue the pseudo-time-stepping iterations, (28) or (45), using the residuals calculated on the distorted space-time prismatic elements and the chosen distribution scheme, until the stopping criteria for the pseudo-time-stepping are satisfied.

3. Replace the mesh at the old time level with the mesh at the new time level, including the updated bed topography and solution.

In all the iterations, we use a stopping criterion which combines a maximum number of iterations with a drop tolerance on the residual norm,

$$\texttt{rel\_tol} = \frac{\| \, (\Psi^n, \Psi^{n+1})_m^T \, \|_1}{\| \, (\Psi^n, \Psi^{n+1})_0^T \, \|_1} < \texttt{REL\_TOL} \,, \tag{61}$$

where, throughout this work, $\texttt{REL\_TOL} = 10^{-3}$ for the pseudo-time-stepping. In addition, there is one intermediate stopping criterion: when the mesh is moved according to the weighted averaging defined by (55)/(56), the node positions are fixed when either $\texttt{rel\_tol} \leq \texttt{REL\_TOL\_MOVE}$ is satisfied or a specified number of iterations (N_{move}) has been reached.

2.4.2. Parameter Choices

The mesh movement algorithm has introduced a number of free parameters which can be modified to control the mesh movement. The optimal choices will be problem-dependent, as is widely observed (see, for example, [11, 18] and the references therein), but varying each parameter affects the approximation in a particular way.

- In Equation (59) α governs the strength of the effect of local features. Increasing its value will increase the resolution of the mesh where the first or second derivative of the solution is relatively high.

- In Equation (60) β and γ govern the extent of the region where the mesh is affected by local features. Increasing their values reduces the size of the region over which the mesh is adapted. The adaptation becomes a pure smoothing step if $\beta, \gamma \geq 1$.

- The larger the values of N_{init} and N_{move}, the maximum numbers of mesh movement iterations allowed at each stage, the closer the initial mesh and the meshes at each time-step (in the absence of any smoothing iteration) get to equidistributing the monitor function. Reducing `REL_TOL_MOVE` (when it is active) has a similar effect.

- The smoothness with which the mesh changes resolution from one region to another can be adjusted using the relaxation parameter μ in Equation (56). Increasing its value in the smoothing iteration (where $W_E \equiv 1$) will tend to make the mesh elements more uniform in size.

3. Numerical Results

For all computations the CFL specified for the time-stepping is relative to the past-shield condition, whereby no residual can be distributed backwards in time, *i.e.*

$$k_{i,n}^+ = 0 \quad \forall i \qquad \text{or} \qquad \mathcal{K}_{i,n}^+ = 0 \quad \forall i \,, \tag{62}$$

for scalar equations and systems of equations, respectively. The CFL specified for the pseudo-time-stepping, CFL_τ, is relative to the maximum pseudo-time-step size specified, *i.e.* (29) or (46), depending on the system being solved. In all cases, only l^2 estimates of the errors are shown: typically adapting the mesh shows greater benefits for the l^∞ norm, which typically selects a node in the region where the mesh resolution has been increased, but less benefits for the l^1 norm which includes more significant contributions from nodes where the mesh resolution has been decreased.

3.1. Scalar Advection

In this section we consider the scalar conservation law (1) with $\boldsymbol{f} = \boldsymbol{a}u$. For the test case studied here, $\boldsymbol{a}$ varies linearly in space with $\nabla \cdot \boldsymbol{a} = 0$, so the conservation law is equivalent to the advection equation,

$$\frac{\partial u}{\partial t} + \boldsymbol{a} \cdot \nabla u = 0 \,, \tag{63}$$

with u specified on inflow boundaries.

3.1.1. Circular Advection

This test case considers advection of a smooth (C^3) initial profile through the domain $\Omega = [-1, 1] \times [-1, 1]$ with velocity $\boldsymbol{a} = (-y, x)$. The exact time-dependent solution profile is given by

$$u(x, y, t) = \begin{cases} \cos^4(1.25\pi r) & \text{if} \quad r < 0.4 \,, \\ 0 & \text{otherwise} \,, \end{cases} \tag{64}$$

in which $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ is the radius of the non-zero region, $x_c = 0.5\sin(t - 0.5\pi)$ and $y_c = 0.5\cos(t - 0.5\pi)$. The initial conditions are given at $t = 0$ and the boundary conditions were set to $u(x, y, t) = 0$ on the inflow sections of the boundary.

The accuracy and efficiency of the simulations were assessed for a sequence of fixed, uniform, unstructured triangular meshes with characteristic mesh sizes $h = \frac{1}{5},\ \frac{1}{10},\ \frac{1}{20},\ \frac{1}{40},\ \frac{1}{80}$. These were also used as the initial meshes for simulations carried out using the adaptive procedure described in Section 2.4 and, to demonstrate the robustness of the space-time residual distribution algorithm on moving meshes, imposing movement on each mesh node using

$$
\begin{aligned}
x &= X + 0.1 \sin(2\pi X) \sin(\pi Y) \sin(t)\,, \\
y &= Y + 0.2 \sin(2\pi X) \sin(\pi Y) \sin(2t)\,,
\end{aligned}
\tag{65}
$$

in which $\boldsymbol{X} = (X, Y)$ represents the initial (reference) coordinates. At $t = 2\pi$, the end time for these numerical experiments, the mesh nodes return to their original positions.

The accuracy and efficiency of the approximations obtained using the STB scheme are illustrated in Figure 2. The mesh movement used to improve the fit to the initial conditions is included in the cpu times. Results are shown for different adaptive strategies for deciding when to terminate the mesh movement iterations within each time-step: after a fixed number of iterations, $N_{\text{move}} = 5$, 10 or 20 in this case; or after the space-time residual norm has dropped below a given tolerance, `REL_TOL_MOVE` $= 10^{-1}$. In all adaptive cases the initial mesh is created by applying $N_{\text{init}} = 200$ iterations of the mesh movement algorithm to the fixed mesh, the monitor function parameters are chosen to be $\alpha = 100$, $\beta = \gamma = 0.2$, and the relaxation parameter for the mesh smoothing was $\mu = 0.5$. These values have been chosen to obtain an efficient adaptive strategy for a broad range of mesh sizes: they are not chosen to provide the most accurate approximations. For the time-stepping, CFL $= 2.0$ and CFL$_\tau = 0.9$.
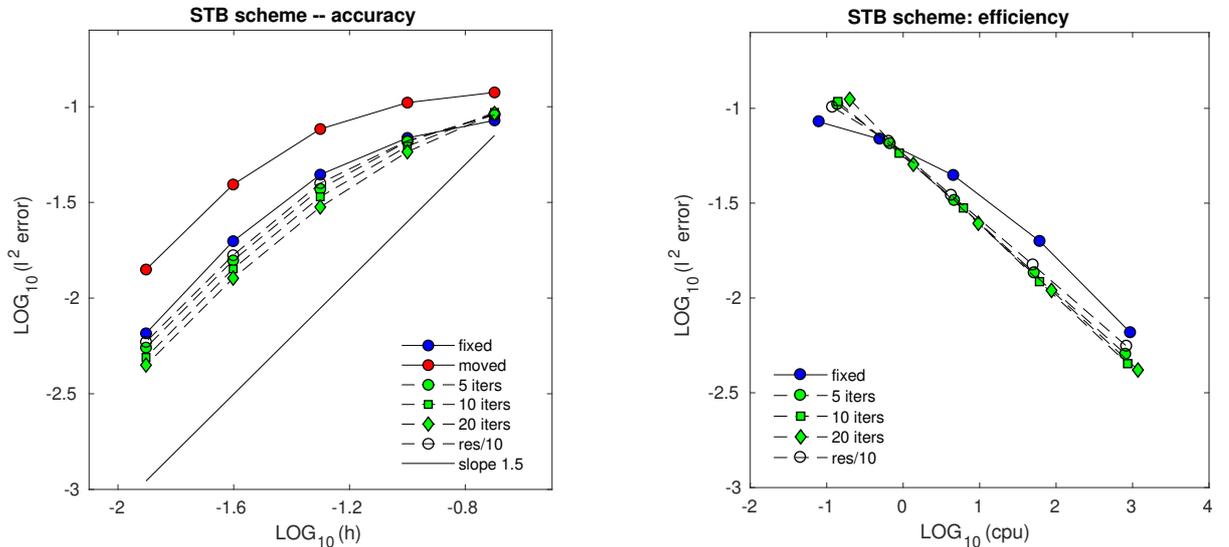
Figure 2: Comparison of accuracy and efficiency for circular advection ($t = 2\pi$), using the STB scheme on fixed meshes (blue circles, solid line), adapted meshes (green or empty symbols, dashed line) and meshes moved according to (65) (red symbols, solid line).

The same CFL is chosen for the time-stepping in both the fixed and moving mesh simulations. CFL $= 2$ is close to the optimal value for the efficiency of the fixed mesh computations [10]: fewer time-steps are required if the CFL is increased further, but the

overall computation time decreases very little because the magnitude of the upper limit on the pseudo-time-step, Equation (29), is inversely proportional to the time-step, so the total number of pseudo-time-steps remains roughly constant. Furthermore, the error starts to increase, so running the adaptive computations at a larger CFL (*e.g.* using the same time-step as the fixed mesh instead of the same CFL) doesn't necessarily improve efficiency.

Each of the adaptive meshing strategies provides more accurate approximations than the corresponding fixed mesh. The errors can be reduced further by increasing N_{move} but the rate at which it decreases is outweighed by the additional time it takes for the computations to run, so the results are not presented here. The order of accuracy on the moving meshes remains similar to that on the fixed meshes.

Using between 10 and 20 mesh movement iterations per time-step seems to be the most efficient adaptive strategy in this case. At the finer mesh resolutions it consistently provides approximately a 30% reduction in error for a given cpu time. Adapting the meshes locally reduces the element size where the solution is non-zero to provide smaller errors but, since the CFL is fixed, the time-step sizes will also be reduced if the mesh movement increases the resolution in regions of high flow velocity, so more time-steps may be required. Table 1 suggests that is not the case here (the fastest flow is at the corners of the square domain) and that the mesh movement has a beneficial effect on the iterative convergence rates of the pseudo-time-stepping. However, each mesh movement iteration requires about 3 times as much cpu time as each residual distribution step (largely due to the expense of computing the weights and the checks for mesh tangling) so overall cpu times are not often reduced. As noted previously, increasing the CFL does not improve the efficiency.

Movement	#ts	#pts	#pts/#ts	#move	cpu time (s)	l^2 error
Fixed	407	24950	61.30	0	61.91	1.983×10^{-2}
$N_{\mathrm{move}} = 5$	385	16204	42.09	1925	52.04	1.354×10^{-2}
$N_{\mathrm{move}} = 10$	390	15615	40.04	3900	60.70	1.219×10^{-2}
$N_{\mathrm{move}} = 20$	403	17855	44.31	8060	87.07	1.096×10^{-2}
`REL_TOL_MOVE` $= 10^{-1}$	383	16820	43.92	1148	50.32	1.491×10^{-2}

Table 1: Comparison of numbers of time-steps (#ts), numbers of pseudo-time-steps (#pts), number of mesh movement iterations (#move) and runtimes for fixed and adapted meshes for the circular advection test case on the mesh with $h = \frac{1}{40}$ in the fixed case using the STB scheme.

The node velocities and the mesh in the bottom-right quadrant of the domain at $t = \frac{\pi}{4}$, for an adaptive strategy with 20 mesh movement iterations per time-step, are shown on the left in Figure 3. Ideally, the mesh would move with the advection velocity, in which case the initial profile would be preserved by our scheme to machine accuracy. Instead the nodes are continually being recruited at the leading edge of the moving profile and discarded at the trailing edge in order to retain high resolution where the solution is non-zero: the mesh inside this region moves much less. The right-hand plots in Figure 3 show the node velocities and mesh obtained at $t = \frac{\pi}{4}$ when $N_{\mathrm{init}} = 5000$ and $N_{\mathrm{move}} = 500$, which approaches a situation where the monitor function is equidistributed (subject to smoothing) at each time-step. This simulation has a block of mesh moving in the

approximate direction of the underlying advection, for which the activity in front of and behind the non-zero region is much reduced. On closer inspection, the circle of higher resolution mesh contains nodes moving at a similar velocity and in a similar direction to each other, a rather different pattern to the actual advection velocity, $(-y, x)$. Therefore the method does not benefit from increased accuracy due to mesh nodes following the flow (something that can also happen when methods of this type are applied in one dimension and is inherent in methods which attempt to maintain equidistribution of a monitor function from one time-step to the next) and the error on the mesh shown on the right-hand side of Figure 3 is slightly higher: it also takes about 18 times as long to run.
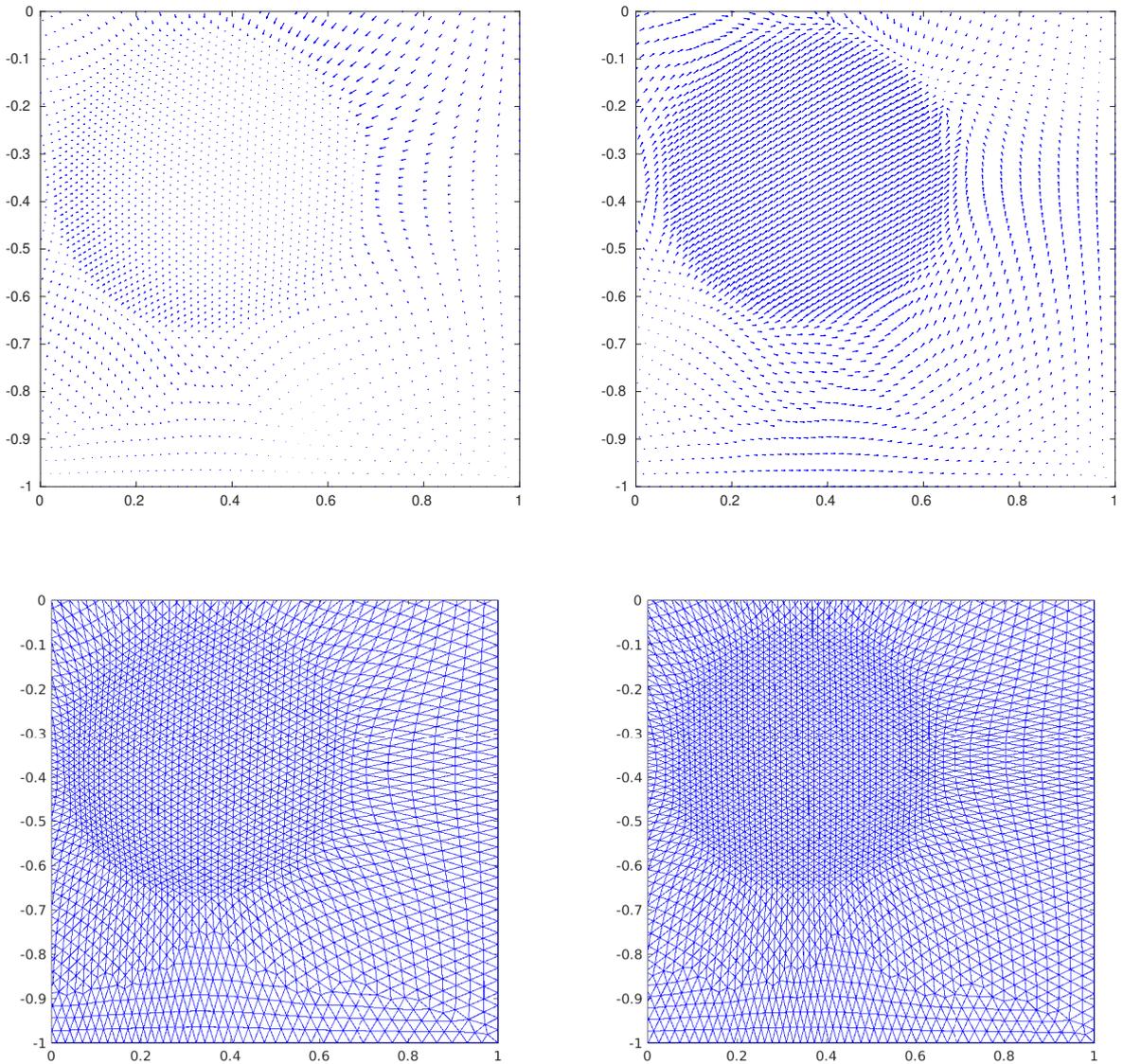
Figure 3: Comparison of node velocities (top) and meshes (bottom) in the bottom right quadrant of the computational domain at $t = \frac{\pi}{4}$ for circular advection, using the STB scheme with $N_{\text{move}} = 20$ (left) and $N_{\text{move}} = 500$ (right).

A single iteration of the STB scheme for a scalar equation requires little computational effort, so mesh movement iterations have a relatively significant effect on the overall computation time. It is worth noting though, that the mesh movement should tend to be more beneficial in 3D: second-order schemes, such as the ones considered here, typically require 16 times as much work to reduce the error by a factor of 4 using uniform refinement in 3D, whereas it only requires 8 times as much work to reduce the error by the same amount in 2D. One might expect the mesh movement, which does not increase the size of the system to solve, to have similar cost-benefit ratios in both 2D and 3D, so the efficiency of the adaptive approach should scale much better in 3D relative to uniform refinement (assuming that the same error reduction can be achieved).

Figure 4 shows the effect of switching between the approximate and exact quadrature rules proposed for evaluating the flow sensors in (18) and (19). The exact quadrature (Simpson's rule) is marginally the more efficient of the two strategies and is therefore the one which has been used to obtain all of the other numerical results shown in this paper.
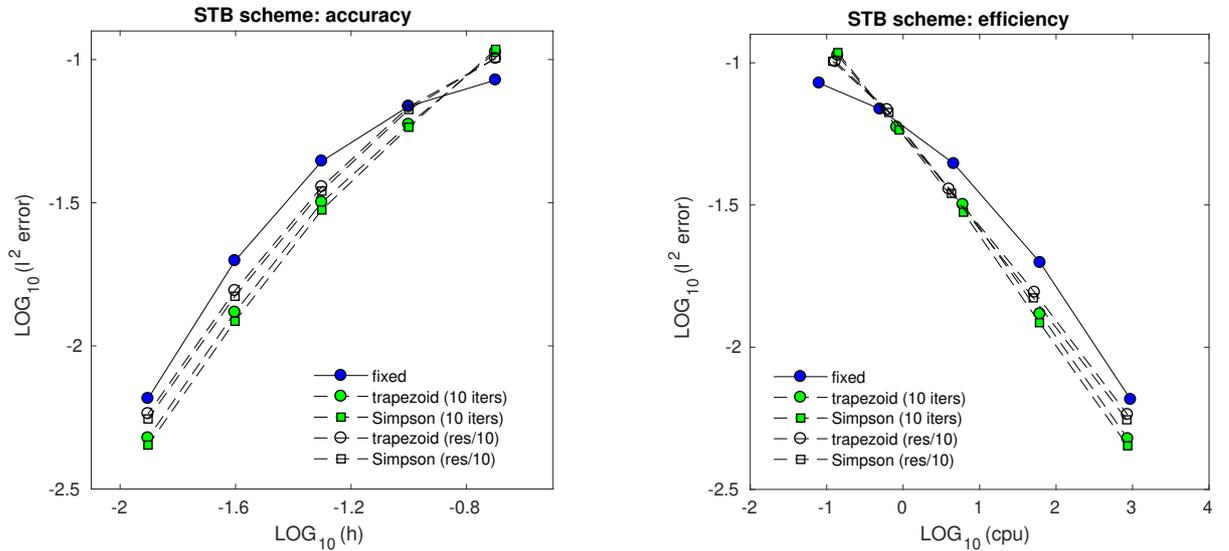
Figure 4: Comparison of accuracy and efficiency for circular advection, using the STB scheme on fixed meshes (blue circles, solid line) and adapted meshes (green or empty symbols, dashed line). Results are shown for when the trapezoid and Simpson rules are used to compute the flow sensors, Equations (19) and (18), respectively.

3.2. Shallow Water Flow

In this section we consider simulations of the two-dimensional shallow water equations with variable bed topography. The mesh is adapted to the free surface level, not the depth, so the bed topography only influences the mesh through its effects on the flow. Although we do not present the results of the simulations, we note here that it was confirmed that all of the mesh movement strategies employed preserved the "lake at rest" solution on a variable bed to machine precision. For all shallow water simulations, the largest value of the blending parameter, $\theta_2^{max} = \max \Theta_2$ (where Θ_2 is defined in Equation (41)), is applied to all variables of the residual in order to achieve an additional stabilising effect. Furthermore, to improve iterative convergence, its value is frozen after the first pseudo-time-step in each time-step.

3.2.1. Travelling Vortex

To evaluate the accuracy and (mesh) convergence properties of the STB schemes, we simulate a travelling vortex with known exact solution [51, 53]. Given a flat bottom topography, the exact velocity field is expressed as $\boldsymbol{u}_\infty + \boldsymbol{u}'$, with

$$\boldsymbol{u}' = \begin{cases} \Gamma \left(1 + \cos(\omega r_c)\right)(y_c - y, x - x_c) & \text{if} \quad \omega r_c < \pi\,, \\ (0, 0) & \text{otherwise}\,, \end{cases}$$

and $\boldsymbol{u}_\infty$ being constant. The constant Γ is the vortex intensity parameter, (x_c, y_c) are the coordinates of the centre of the vortex, r_c is the distance from the centre of the vortex, and ω is the angular wave frequency associated with the diameter of the vortex. The water height is then given as

$$d(r_c) = d_\infty + \begin{cases} \frac{1}{g}\left(\frac{\Gamma}{\omega}\right)^2 (\kappa(\omega r_c) - \kappa(\pi)) & \text{if} \quad \omega r_c < \pi\,, \\ 0 & \text{otherwise}\,, \end{cases}$$

with

$$\kappa(x) = 2\cos(x) + 2x\sin(x) + \frac{1}{8}\cos(2x) + \frac{x}{4}\sin(2x) + \frac{3}{4}x^2$$

and $d_\infty = 1$.

For the mesh-convergence study, we set $\boldsymbol{u}_\infty = (6, 0)$, $\Gamma = 15$, $\omega = 4\pi$, $g = 9.80665$ and use a sequence of five unstructured triangulations of the domain $\Omega = [0, 2] \times [0, 1]$ with characteristic mesh sizes $h = \frac{1}{10}, \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}$, respectively. At the initial state the centre of the vortex is at $(x_c, y_c) = (0.5, 0.5)$ and the time marching stops at $t = 1/6$, when $(x_c, y_c) = (1.5, 0.5)$. Freestream characteristic boundary conditions are used everywhere.

The accuracy and efficiency of the approximations obtained using the STB schemes are illustrated in Figure 5. Results are shown for fixed meshes and for different adaptive strategies for deciding when to terminate the mesh movement iterations within each time-step: after a fixed number of iterations, $N_{\text{move}} = 10$, 30 or 50 in this case; or after the space-time residual norm has dropped below a given tolerance, `REL_TOL_MOVE` $= 10^{-1}$. In all adaptive cases the initial mesh is created by applying $N_{\text{init}} = 500$ iterations of the mesh movement algorithm to the fixed mesh, the monitor function parameters are chosen to be $\alpha = 100$, $\beta = \gamma = 0.05$, and the relaxation parameter for the mesh smoothing was $\mu = 1.0$. For the time-stepping, CFL $= 4.0$, close to the optimal value found for efficiency on fixed meshes in [10], and $\text{CFL}_\tau = 0.3$.

In this case, the order of accuracy demonstrated on the finer meshes is slightly lower on the adapted meshes than on the fixed meshes. As a result, the mesh adaptation provides most benefit for meshes of intermediate resolution, on which improvements in efficiency of more than 60% can be obtained by moving the mesh. In this case, using 30 mesh movement iterations per time-step gave the best efficiency – a higher number than for scalar advection, not only because the accuracy starts to deteriorate if more mesh movement iterations are used, but also because the computational time required for a single mesh movement iteration is lower relative to a single application of residual distribution for a nonlinear system. The total number of mesh movement iterations relative to the total number of pseudo-time-stepping iterations is shown in Table 2. The residual distribution step is about 5 times as expensive as in the scalar case, so the time required to move the mesh is relatively less significant. Moving the mesh also appears to accelerate the convergence of the pseudo-time-stepping iteration.
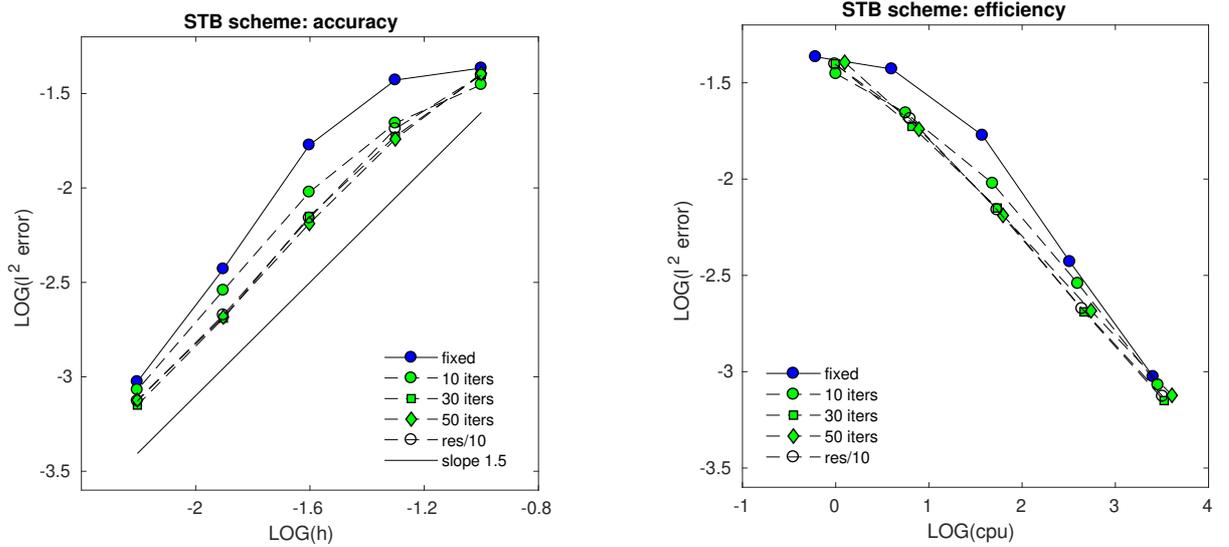
Figure 5: Comparison of accuracy and efficiency for the travelling vortex, using the STB scheme on fixed meshes (blue circles, solid line) and adapted meshes (green or empty symbols, dashed line).

Movement	#ts	#pts	#pts/#ts	#move	cpu time (s)	l^2 error in d
Fixed	66	7322	110.94	0	325.20	3.724×10^{-3}
$N_{\mathrm{move}} = 10$	91	8144	89.49	910	396.54	2.870×10^{-3}
$N_{\mathrm{move}} = 30$	103	8851	85.93	3090	461.50	2.044×10^{-3}
$N_{\mathrm{move}} = 50$	104	10036	96.50	5200	548.87	2.073×10^{-3}
`REL_TOL_MOVE` $= 10^{-1}$	103	8539	82.90	2622	439.72	2.124×10^{-3}

Table 2: Comparison of numbers of time-steps (#ts), numbers of pseudo-time-steps (#pts), number of mesh movement iterations (#move) and runtimes for fixed and adapted meshes for the travelling vortex test case on the mesh with $h = \frac{1}{80}$ in the fixed case using the STB scheme.

3.2.2. Small Perturbation of a Lake at Rest

In order to test the adaptive method in a situation with smoothly varying bed topography, this test case considers the flow of a small perturbation to the lake at rest solution over an elliptic exponential bump [54]. The computational domain is given by $[0, 2] \times [0, 1]$ with bed topography

$$b(x, y) = 0.8 \exp(-5(x - 0.9)^2 - 50(y - 0.5)^2) \tag{66}$$

and initial conditions,

$$\eta(x, y) = \begin{cases} 1.01 & \text{if} \quad 0.05 < x < 0.15\,, \\ 1 & \text{otherwise}\,, \end{cases} \qquad u \equiv v \equiv 0\,. \tag{67}$$

Solid-wall boundary conditions are used at the top and bottom boundaries while free-flow boundary conditions are imposed at the remaining boundaries.

Numerical results obtained using a mesh with $h \approx 0.02$ (with 5953 nodes and 17556 elements) are presented in Figures 6 and 7. Snapshots of the fluid and mesh velocities

26

are shown in Figure 8. In all adaptive cases the initial mesh is created by applying $N_{\text{init}} = 1000$ iterations of the mesh movement algorithm to the fixed mesh, the monitor function parameters are chosen to be $\alpha = 100$, $\beta = \gamma = 0.2$, and the relaxation parameter for the mesh smoothing was $\mu = 0.7$. For the time-stepping, CFL = 4.0 and $\text{CFL}_\tau = 0.9$.
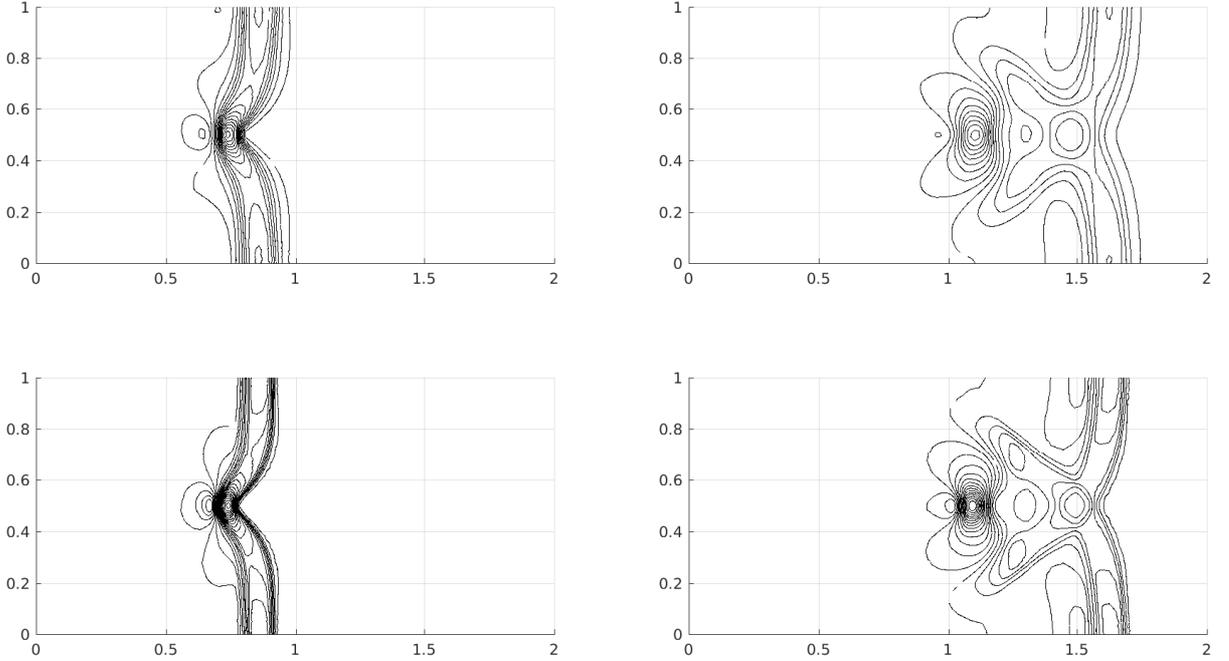
Figure 6: Comparison of isolines of free surface level η for the shallow water equations with a variable bed and a perturbation to the lake at rest equilibrium, using the STB scheme on a fixed mesh (top) and an adapted mesh with $N_{\text{move}} = 20$ (bottom): $t = 0.24$ (left), $t = 0.48$ (right). The interval between contours is 0.0005 and the isolines shown are in the ranges $[0.99925, 1.0725]$ (top left), $[0.99575, 1.00275]$ (top right), $[0.99825, 1.00975]$ (bottom left), $[0.99325, 1.00325]$ (bottom right).

The mesh adaptation clearly resolves the features of the flow much more sharply. The adapted solution is compared with approximations obtained on this fixed mesh and a finer fixed mesh ($h \approx 0.01$, giving 23515 nodes and 69942 elements) in Figure 9. Two different adaptive strategies are used, in which a fixed number of mesh movement iterations, $N_{\text{move}} = 10$ or 20, are applied within each time-step. Adapting the mesh clearly improves the approximation, giving estimates of local maxima and minima which are close to those obtained on the fixed fine mesh for all regions except the small peak just downstream of the main trough at $t = 0.48$. The improvements are greater when 20 mesh movement iterations are used, but this is at the expense of computational time. The efficiency is illustrated in Table 3, which shows maximal and minimal values for different meshes alongside the runtimes. These results include an intermediate mesh, for which $h \approx 0.01\sqrt{2}$ and the runtime is similar to those for the adapted coarse mesh. Both adapted coarse meshes give better maximal and minimal value approximations than the intermediate mesh for either a similar ($N_{\text{move}} = 20$) or significantly lower ($N_{\text{move}} = 10$) runtime.
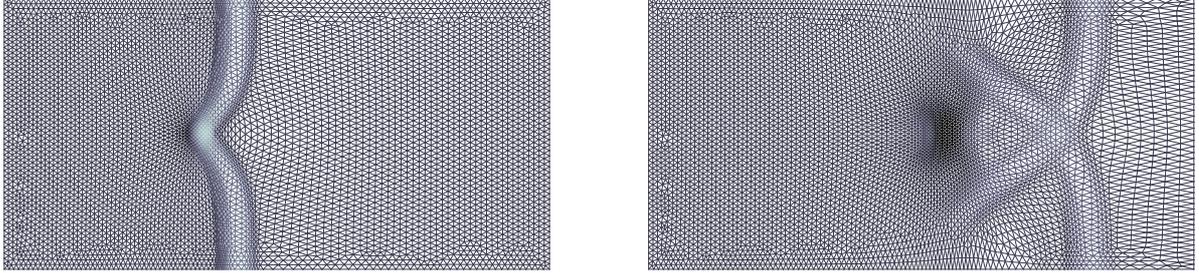
Figure 7: Adapted meshes with $N_{\text{move}} = 20$ for the shallow water equations with a variable bed and a perturbation to the lake at rest equilibrium, obtained using the STB scheme at times $t = 0.24$ (left), $t = 0.48$ (right).
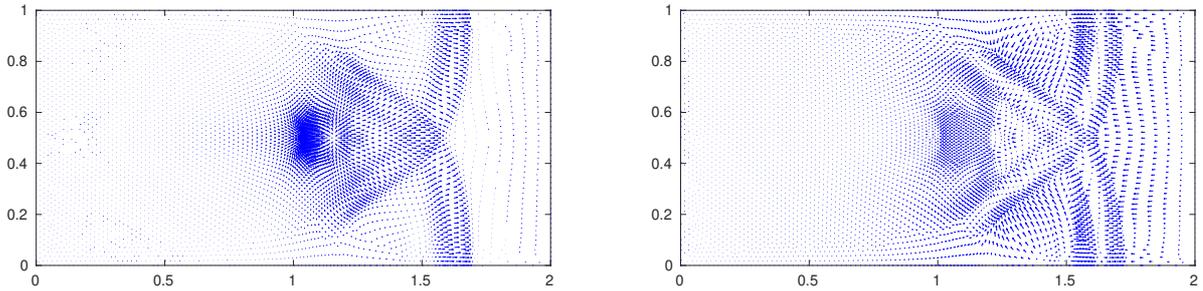
Figure 8: Fluid (left) and mesh (right) velocities obtained using the STB scheme on an adapted mesh with $N_{\text{move}} = 20$, for a perturbation to the lake at rest equilibrium, at time $t = 0.48$.

Mesh	# nodes	Movement	cpu time (s)	$\min(\eta)$	$\max(\eta)$
Coarse	5953	None	47.18	0.995548	1.003152
		$N_{\text{move}} = 10$	82.74	0.993329	1.003473
		$N_{\text{move}} = 20$	140.04	0.992780	1.003531
Intermediate	11924	None	139.67	0.993751	1.003426
Fine	23515	None	381.76	0.992074	1.003663

Table 3: Comparison of maximum/minimum values of η and runtimes for fixed and adapted meshes for the perturbed lake at rest when $t = 0.48$ using the STB scheme.
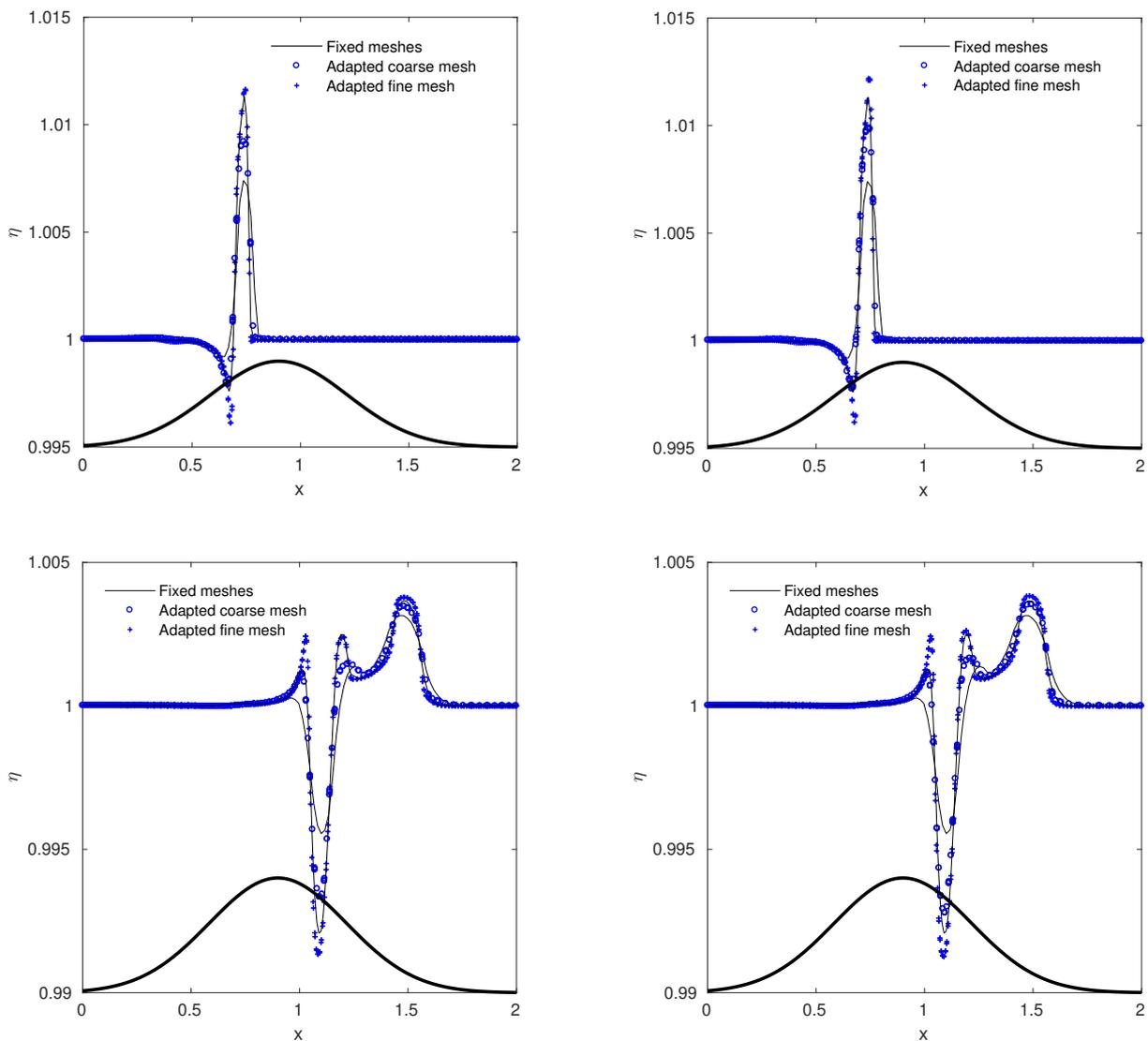
Figure 9: Comparison of profiles of free surface level η along $y = 0.5$ for the shallow water equations with a variable bed and a perturbation to the lake at rest equilibrium, using the STB scheme on coarse and fine meshes, both fixed and adapted, at times $t = 0.24$ (top) and $t = 0.48$ (bottom). Results are shown for 10 (left) and 20 (right) mesh movement iterations per time-step. The thick black line illustrates the shape of the bed topography: this is *not* drawn to scale.

3.2.3. Circular Dam-Break over a Non-Smooth Bed

The final test case has been chosen to illustrate the ability of the moving mesh to follow sharp features which interact in geometrically complex ways. It simulates an initially radially-symmetric dam-break over a discontinuous bed topography [10]. The computational domain is given by $[0, 30] \times [0, 30]$ with bed topography

$$b(x, y) = \begin{cases} 0 & \text{if } x + y < 30 \,, \\ 0.2 & \text{otherwise} \,, \end{cases} \tag{68}$$

and initial conditions,

$$\eta(x, y) = \begin{cases} 1.461837 & \text{if } r < 15 \,, \\ 0.308732 & \text{otherwise} \,, \end{cases} \quad u \equiv v \equiv 0 \,, \tag{69}$$

where $r = \sqrt{x^2 + y^2}$ is the radius of the dam. Solid-wall boundary conditions are used at the left and bottom boundaries while characteristic boundary conditions are imposed at the remaining boundaries. The characteristic mesh size is $h \approx 0.3$ (giving a mesh of 11831 nodes and 35090 elements). The simulation follows the wave hitting the underwater wall, then partially reflecting from it and partially moving forward and exiting the domain. There is also a stationary shock wave along the discontinuity of the bed, which slowly weakens as time progresses.

In all adaptive cases the initial mesh is created by applying $N_{\text{init}} = 1000$ iterations of the mesh movement algorithm to the fixed mesh, the monitor function parameters are chosen to be $\alpha = 100$, $\beta = \gamma = 0.05$, and the relaxation parameter for the mesh smoothing was $\mu = 0.7$. A fixed number of mesh movement iterations, $N_{\text{move}} = 20$, are applied within each time-step. For the time-stepping, CFL $= 2.0$ and CFL$_\tau = 0.9$.

Snapshots of the numerical results on fixed and adapted meshes are compared in Figure 10. Once more, the adapted meshes (shown in Figure 11) capture the flow features much more sharply, following them throughout the simulation without distorting their shapes. The mesh movement parameters (α, β, γ, μ, N_{init} and N_{move}) can be adjusted to fit more or less strongly to these features, though it should always be borne in mind that reducing the element size increases the number of time-steps required to run at the same CFL and increasing the number of mesh movement iterations per time-step will typically increase the overall number of pseudo-time-stepping iterations that are required.

We also note that the combination of the monitor choice and the application of smoothing produces meshes which tend to align with lower-dimensional features such as shocks, but vary smoothly throughout the domain. In fact, away from the discontinuities, the choices made for β and γ ensure that the mesh tends towards a uniform distribution of nodes. This combines well with the STB residual distribution schemes which can capture these discontinuities relatively sharply, without inducing spurious oscillations. Without the smoothing the meshes for all of the test cases would be of much lower quality and might be better paired with a more grid-insensitive approach to approximating the PDE.

4. Concluding Remarks and Outlook

In this paper we have described the generalisation of a space-time discontinuous residual distribution (STDRD) scheme [10] to moving meshes. Within the STDRD framework
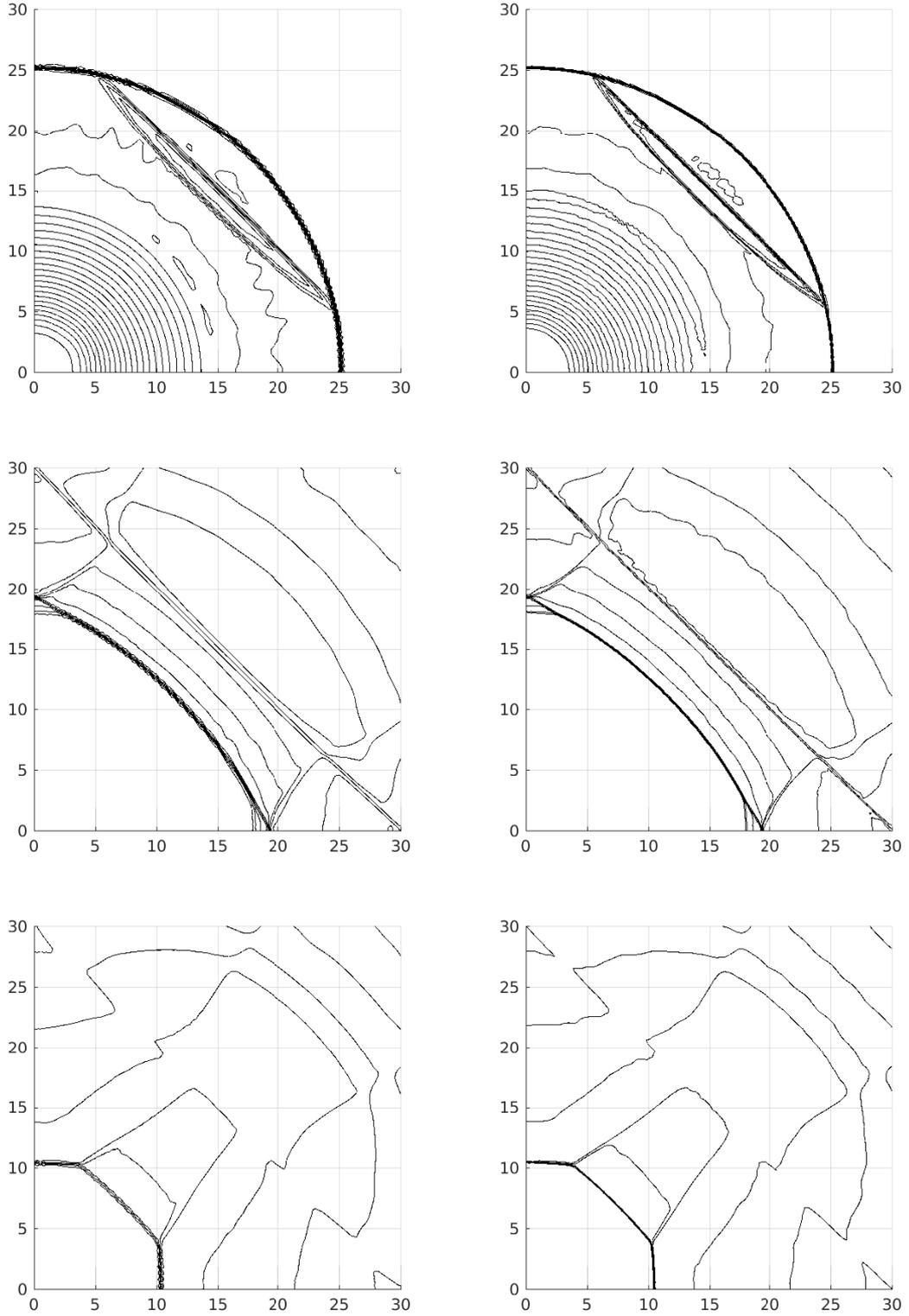
Figure 10: Comparison of isolines of free surface level η for the shallow water equations with a discontinuous bed, using the STB scheme on a fixed mesh (left) and an adapted mesh (right): $t = 3$ (top), $t = 9$ (middle), $t = 15$ (bottom). The interval between contours is 0.04 and the isolines shown are in the ranges $[0.32, 1.44]$ (top), $[0.2, 0.64]$ (middle), $[0.08, 0.4]$ (bottom).
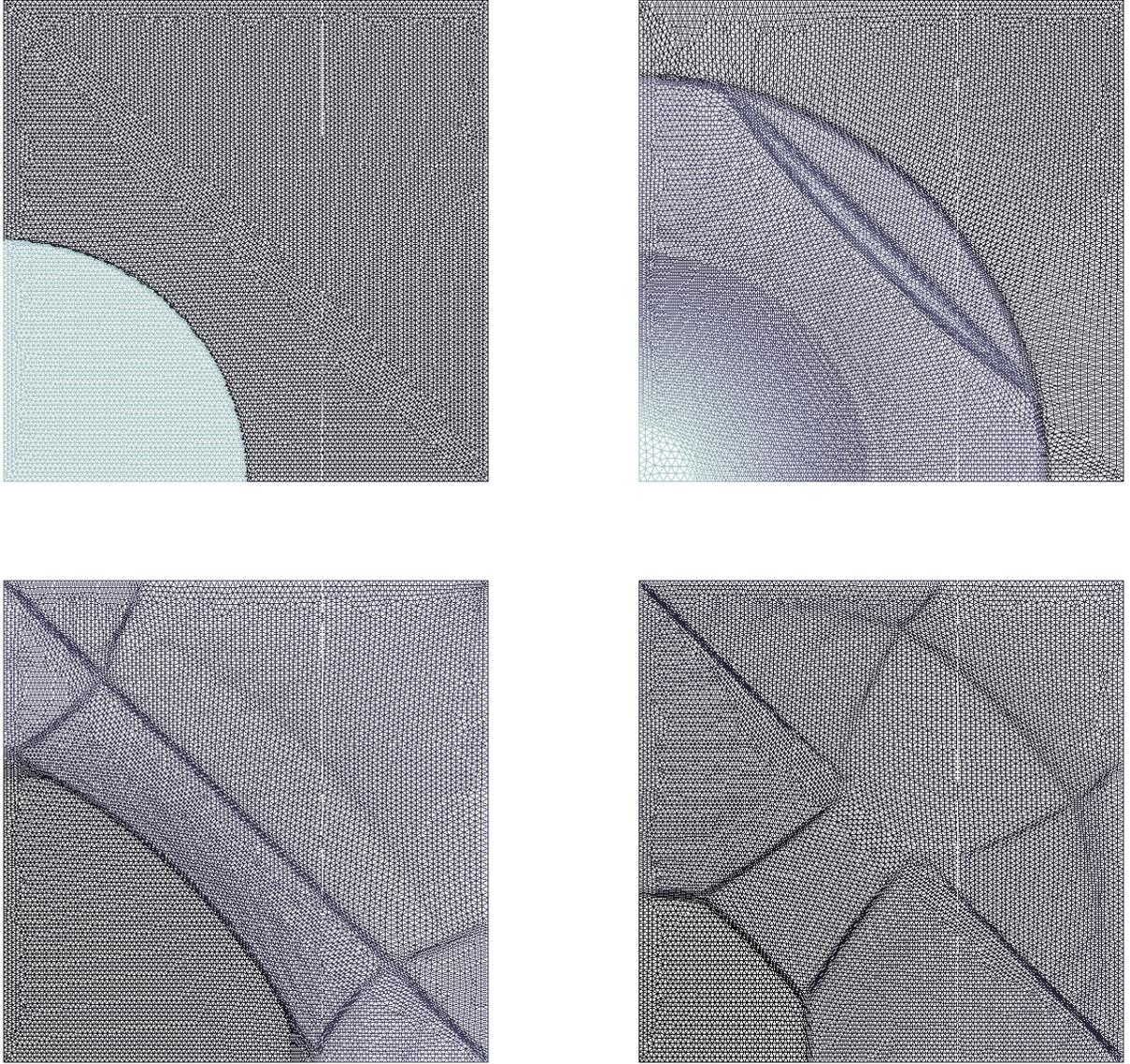
Figure 11: Adapted meshes for the shallow water equations with a discontinuous bed at times $t = 0$ (top left), $t = 3$ (top right), $t = 9$ (bottom left), $t = 15$ (bottom right). The mesh is shaded according to the local height of the free surface η.

this entails the design of schemes for distributing the residual on distorted space-time prisms, and Section 2 describes how schemes developed for prisms aligned with time can be extended to allow the spatial mesh to change between time-steps, while retaining the order of accuracy and positivity properties of the original distribution. Furthermore, the modified schemes inherently guarantee that the mesh movement satisfies a discrete geometric conservation law (DGCL), so adapting the mesh does not disturb a constant solution.

The new algorithm is applied to the two-dimensional scalar advection equation and the two-dimensional shallow water equations with variable bed topography. In the latter case, the schemes are designed so that the C-property is satisfied, *i.e.* still, flat water is maintained as a steady-state solution over a variable bed even when the mesh is being moved. The STDRD schemes are combined with a simple mesh adaptation algorithm to demonstrate that (i) more accurate approximations can be obtained by moving the mesh to follow features of the flow and (ii) there is the potential for the mesh movement to reduce the computational time required to provide approximations to within a specified level of accuracy.

The numerical results show that the mesh movement does improve the accuracy of the approximations compared to fixed, uniform meshes with the same number of nodes and connectivity. The accuracy and the efficiency of the computations both depend on the choices of a number of user-specified parameters. It is not yet clear what the optimal choices are, but it has been shown that moving the mesh can be beneficial, with reductions in cpu times of up to 60% achieved, compared to fixed mesh computations with the same accuracy. However in some cases the reduction in error is counterbalanced by the additional computational effort required and mesh movement is not currently effective.

This work has demonstrated that mesh movement can be used to improve the efficiency of STDRD schemes, without losing any of their fixed-mesh properties, even with a simple mesh movement strategy. However, there remain many areas in which both the fixed and moving mesh algorithms might be improved.

- The mesh movement strategy is very simple. More sophisticated approaches, such as the nonlinear iteration described in [11], might allow for a reduction in the number of mesh movement iterations required relative to the number of pseudo-time-steps. In particular, a much faster algorithm for equidistributing the initial mesh could be found.

- The limit on the pseudo-time-step is inversely proportional to the size of the time-step, so using a larger CFL (which is possible due to the unconditional stability of the STDRD schemes) on the adapted mesh does not tend to decrease the overall workload. A sophisticated nonlinear solver, for which the convergence rate was independent of the time-step, would improve efficiency on both fixed and adapted meshes.

- We would expect the mesh movement to have greater benefits for more complex systems of nonlinear equations (for which the expense of residual distribution is higher relative to the mesh movement) and in three space dimensions (where it requires more work to reduce the error by the same amount using uniform mesh

refinement). These are applications which should be investigated, particularly in situations where anisotropic mesh refinement is likely to be beneficial.

- Moving meshes would naturally be appropriate for moving boundary problems. In the context of the shallow water equations it would be of interest to consider problems with wetting and drying.

- Moving the mesh might affect the best choice of blending function to use in the STB scheme. There was no clear pattern visible in the components of Θ_2 in (41) generated by the test cases shown in this paper, so one possibility might be to use machine learning techniques to determine an appropriate sensor.

Even with these improvements, there will remain situations in which it is not appropriate to apply mesh movement and hp-refinement will have more effect. However, it is likely to be beneficial in situations where the accuracy with which the quantities of interest can be predicted depends on the ability of the method to track and align the mesh with sharp local features.

5. Acknowledgements

Appendix A. Proof of the C-property

Proposition

The space-time element residuals of the STDRD schemes for the shallow water equations on moving meshes, defined by Equations (31), and (42), satisfy

$$\Psi_E + \Phi_{E_t} + \Phi_{E_t}^b = 0 \tag{A.1}$$

when $V = [\eta, du, dv]^T = [\text{const}, 0, 0]^T$ and

$$\Phi_{E_t}^b = \frac{\Delta t}{2} \left(\frac{g\overline{d}^{n^+}}{2} \sum_{i \in E} \begin{bmatrix} 0 \\ b_i^{n^+} \boldsymbol{n}_i^{n^+} \end{bmatrix} + \frac{g\overline{d}^{(n+1)^-}}{2} \sum_{i \in E} \begin{bmatrix} 0 \\ b_i^{(n+1)^-} \boldsymbol{n}_i^{(n+1)^-} \end{bmatrix} \right). \tag{A.2}$$

Hence, the STN, STLDA and STB schemes based on the use of V instead of U all satisfy the C-property on moving meshes.

Proof

The superscripts $\cdot^{n^-}$, $\cdot^{n^+}$ and $\cdot^{(n+1)^-}$ are used explicitly here for completeness, although the $\pm$ are suppressed in the main body of the paper where the meaning should be clear from the context.

First note that $\Psi_E = 0$ when $V^{n^+} = V^{n^-}$, and there is no contribution from the source term when integrating across the temporal discontinuity, so we need only consider the balance within the space-time element. Now, when $u = v = 0$,

$$\Phi_{E_t} + \Phi_{E_t}^b = \int_{E^{n+1}} \begin{bmatrix} d^{(n+1)^-} \\ \boldsymbol{0} \end{bmatrix} \mathrm{d}\Omega - \int_{E^n} \begin{bmatrix} d^{n^+} \\ \boldsymbol{0} \end{bmatrix} \mathrm{d}\Omega$$

$$+ \int_{t^n}^{t^{n+1}} \int_{\partial E(t)} \begin{bmatrix} -d\boldsymbol{v} \cdot \hat{\boldsymbol{n}} \\ \frac{gd^2}{2}\hat{\boldsymbol{n}} \end{bmatrix} \mathrm{d}\Gamma \, \mathrm{d}t - \int_{t^n}^{t^{n+1}} \int_{E(t)} \begin{bmatrix} 0 \\ gd\nabla b \end{bmatrix} \mathrm{d}\Omega \, \mathrm{d}t, \tag{A.3}$$

in which $\hat{\boldsymbol{n}}$ is the unit outward-pointing normal to $\partial E(t)$.

Mass balance: Since $\frac{\partial b}{\partial t} = 0$ in the exact case, integrating the Reynolds transport theorem for the bed level b on a moving mesh over a single time-step gives

$$\int_{E^{n+1}} b^{(n+1)^-} \, \mathrm{d}\Omega - \int_{E^n} b^{n^+} \, \mathrm{d}\Omega - \int_{t^n}^{t^{n+1}} \int_{\partial E(t)} b\boldsymbol{v} \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma = 0, \tag{A.4}$$

so, since $\eta = d + b$, the residual for the mass balance equation in (A.3) can be written as

$$\int_{E^{n+1}} \eta^{(n+1)^-} \, \mathrm{d}\Omega - \int_{E^n} \eta^{n^+} \, \mathrm{d}\Omega - \int_{t^n}^{t^{n+1}} \int_{\partial E(t)} \eta\boldsymbol{v} \cdot \hat{\boldsymbol{n}} \, \mathrm{d}\Gamma \tag{A.5}$$

$$\approx \quad |E^{n+1}|\overline{\eta}^{(n+1)^-} - |E^n|\overline{\eta}^{n^+} - \frac{\Delta t}{2}\left(\int\limits_{\partial E(t^{n+1})} \eta^{(n+1)^-}\boldsymbol{v}\cdot\hat{\boldsymbol{n}}^{n+1}\,\mathrm{d}\Gamma + \int\limits_{\partial E(t^n)} \eta^{n^+}\boldsymbol{v}\cdot\hat{\boldsymbol{n}}^n\,\mathrm{d}\Gamma \right),$$

in which we have used the approximation to the residual adopted in the STDRD schemes. Now, since the nodal velocities $\boldsymbol{v}_i$ are assumed constant within each time-step, it follows from geometric arguments applied to the space-time element that

$$|E^{n+1}| - |E^n| - \frac{\Delta t}{2}\sum_{i\in E}\left(\boldsymbol{v}_i\cdot\boldsymbol{n}_i^{n+1} + \boldsymbol{v}_i\cdot\boldsymbol{n}_i^n \right) \equiv 0. \tag{A.6}$$

It immediately follows that substituting $\eta^{(n+1)^-} = \eta^{n^+} = $ const in to (A.5) gives zero residual in the mass balance equation when η is piecewise linear.

Momentum balance: Our STDRD schemes evaluate the residual using Simpson's rule to approximate the surface integrals. This is exact for piecewise linear d, so

$$\int\limits_{\partial E(t)} \frac{gd^2}{2}\hat{\boldsymbol{n}}\,\mathrm{d}\Gamma = -\sum_{i\in E}\frac{g\overline{d}}{2}d_i\boldsymbol{n}_i. \tag{A.7}$$

This is clearly gives zero total residual in the momentum balance equations when $\eta = d + b = $ const is substituted in to (A.3) and the source residual is approximated at any given time using

$$\int\limits_{E(t)} gd\nabla b\,\mathrm{d}\Omega = -\sum_{i\in E}\frac{g\overline{d}}{2}b_i\boldsymbol{n}_i, \tag{A.8}$$

which is exact for piecewise linear d and b.

Full system: It follows from the arguments above that $\eta^{(n+1)^-} = \eta^{n^+} = $ const gives zero residual in the full system, having applied the trapezoid rule in time in (A.3), when

$$\Phi_{E_t}^b = \frac{\Delta t}{2}\left(\frac{g\overline{d}^{n^+}}{2}\sum_{i\in E}\begin{bmatrix}0\\b_i^{n^+}\boldsymbol{n}_i^{n^+}\end{bmatrix} + \frac{g\overline{d}^{(n+1)^-}}{2}\sum_{i\in E}\begin{bmatrix}0\\b_i^{(n+1)^-}\boldsymbol{n}_i^{(n+1)^-}\end{bmatrix} \right). \tag{A.9}$$

Therefore $[\eta, du, dv]^T = [\text{const}, 0, 0]^T$ is a solution to the discrete problem. Moreover, since the pseudo-time-stepping is initialised with $(V^{n^+})_0 = (V^{(n+1)^-})_0 = V^{n^-} = [\text{const}, 0, 0]^T$, it follows that

$$(\Psi_{E_t} + \Phi_{E_t} + \Phi_{E_t}^b)_0 = 0 \tag{A.10}$$

and the total residual remains zero for any subsequent pseudo-time-steps. Hence, the C-property is satisfied for any linearity preserving STDRD scheme, such as STLDA. Finally, from [10], the residuals for the STN scheme are

$$\Phi_{i,n}^N = \mathcal{K}_{i,n}^+\mathcal{N}_t\sum_{j\in E}\mathcal{K}_{j,n}^+\left(V_i^{n^+} - V_j^{n^+} \right)$$
$$+ \mathcal{K}_{i,n}^+\mathcal{N}_t\sum_{j\in E}\mathcal{K}_{j,n+1}^+\left(V_i^{n^+} - V_j^{(n+1)^-} \right) + \Phi_{i,n}^{LDA},$$

$$\Phi_{i,n+1}^{N} = \mathcal{K}_{i,n+1}^{+} \mathcal{N}_{t} \sum_{j \in E} \mathcal{K}_{j,n}^{+} \left(V_{i}^{(n+1)^{-}} - V_{j}^{n^{+}} \right)$$

$$+ \mathcal{K}_{i,n+1}^{+} \mathcal{N}_{t} \sum_{j \in E} \mathcal{K}_{j,n+1}^{+} \left(V_{i}^{(n+1)^{-}} - V_{j}^{(n+1)^{-}} \right) + \Phi_{i,n+1}^{LDA} . \quad \text{(A.11)}$$

These residuals are also clearly zero for $(V^{n^{+}})_{0} = (V^{(n+1)^{-}})_{0} = V^{n^{-}} = [\text{const}, 0, 0]^{T}$, so the STN scheme (and hence the STB scheme) also satisfies the C-property. $\square$

Appendix B. Proof of the Discrete Maximum Principle

Proposition:

When a conservative linearisation exists for the scalar advection equation (so the residual can be evaluated exactly using (17)) the space-time N scheme (24) combined with discontinuities in time verifies the discrete maximum principle,

$$u_{\min}^{n^-} = \min_j u_j^{n^-} \leq u_i^{n^+}, \, u_i^{(n+1)^-} \leq \max_j u_j^{n^-} = u_{\max}^{n^-} \qquad \forall i \,, \tag{B.1}$$

for any time-step $\Delta t > 0$ on moving meshes when there is no mesh tangling.

Proof

This proof follows closely the structure of the proof of Proposition 3.8 in [2].

First write out the space-time N scheme with discontinuities in time in full, by substituting (24), (17) and (13) into (14), to give

$$|S_i^n|(u_i^{n^+} - u_i^{n^-})$$
$$+ \sum_{E \in D_i} \left[-\sum_{j \in E} k_{i,n}^+ N_t k_{j,n}^- (u_i^{n^+} - u_j^{n^+}) - \sum_{j \in E} k_{i,n}^+ N_t k_{j,n+1}^- (u_i^{n^+} - u_j^{(n+1)^-}) \right] = 0$$

$$\sum_{E \in D_i} \left[-\sum_{j \in E} k_{i,n+1}^+ N_t k_{j,n}^- (u_i^{(n+1)^-} - u_j^{n^+}) - \sum_{j \in E} k_{i,n+1}^+ N_t k_{j,n+1}^- (u_i^{(n+1)^-} - u_j^{(n+1)^-}) \right] = 0 \,.$$

(B.2)

in which $D_i = \cup_{i \in E} E$, $k^{\pm}$ and N_t are defined by (21) and (18) or (19), and $|S_i^n| = \frac{1}{3} \sum_{E \in D_i} |E^n|$. This system of equations can be written in the form $AU^{new} = BU^{old}$:

$$\begin{bmatrix} A_{n+,n+} & A_{n+,(n+1)-} \\ A_{(n+1)-,n+} & A_{(n+1)-,(n+1)-} \end{bmatrix} \begin{bmatrix} U^{n^+} \\ U^{(n+1)^-} \end{bmatrix} = \begin{bmatrix} B_{n-} \\ 0 \end{bmatrix} \begin{bmatrix} U^{n^-} \end{bmatrix} \tag{B.3}$$

in which U represents a vector of nodal values of u at the specified time level and

$$(A_{n^+,n^+})_{ii} = |S_i^n| - \sum_{E \in D_i} \left[\sum_{j \in E, j \neq i} k_{i,n}^+ N_t k_{j,n}^- + \sum_{j \in E} k_{i,n}^+ N_t k_{j,n+1}^- \right] \qquad \forall\, i$$

$$(A_{n^+,n^+})_{ij} = \sum_{E \in D_i | j \in E} k_{i,n}^+ N_t k_{j,n}^- \qquad \forall\, i, j \,|\, i \neq j$$

$$(A_{n^+,(n+1)^-})_{ij} = \sum_{E \in D_i | j \in E} k_{i,n}^+ N_t k_{j,n+1}^- \qquad \forall\, i, j$$

$$(A_{(n+1)^-,n^+})_{ij} = \sum_{E \in D_i | j \in E} k_{i,n+1}^+ N_t k_{j,n}^- \qquad \forall\, i, j$$

$$(A_{(n+1)^-,(n+1)^-})_{ij} = \sum_{E \in D_i | j \in E} k_{i,n+1}^+ N_t k_{j,n+1}^- \qquad \forall\, i, j \,|\, i \neq j$$

$$(A_{(n+1)^-,(n+1)^-})_{ii} = -\sum_{E \in D_i} \left[\sum_{j \in E} k_{i,n+1}^+ N_t k_{j,n}^- + \sum_{j \in E, j \neq i} k_{i,n+1}^+ N_t k_{j,n+1}^- \right] \qquad \forall\, i$$

$$(B_{n^-})_{ii} = |S_i^n| \qquad \forall\, i$$

$$(B_{n^-})_{ij} = 0 \qquad \forall\, i, j \,|\, i \neq j \,. $$
(B.4)

The majority of the entries in A will be zero because the set $\{E \in D_i | j \in E\}$ is empty unless i and j are vertices of a common mesh element. The indices i and j both run from 1 to the total number of unknowns. First note that

$$\begin{aligned}
(A_{n^+,n^+})_{ii} \geq 0 && (A_{(n+1)^-,(n+1)^-})_{ii} \geq 0 && \forall\, i \\
(A_{n^+,n^+})_{ij} \leq 0 && (A_{(n+1)^-,(n+1)^-})_{ij} \leq 0 && \forall\, i, j \,|\, i \neq j \\
(A_{n^+,(n+1)^-})_{ij} \leq 0 && (A_{(n+1)^-,n^+})_{ij} \leq 0 && \forall\, i, j \,,
\end{aligned}$$
(B.5)

since k^+, $N_t \geq 0$ and $k^- \leq 0$. Moreover,

$$|(A_{n^+,n^+})_{ii}| - \sum_{j \in D_i | j \neq i} |(A_{n^+,n^+})_{ij}| - \sum_{j \in D_i} |(A_{n^+,(n+1)^-})_{ij}| = |S_i^n| > 0$$

$$|(A_{(n+1)^-,(n+1)^-})_{ii}| - \sum_{j \in D_i | j \neq i} |(A_{(n+1)^-,(n+1)^-})_{ij}| - \sum_{j \in D_i} |(A_{(n+1)^-,n^+})_{ij}| = 0 \,. $$
(B.6)

Therefore, the matrix A is diagonally dominant *for any time-step and any mesh* as long as $|S_i^n| > 0$ (no tangling). Hence, because of the conditions in (B.5), A is an M-matrix, so A is invertible and A^{-1} is positive, *i.e.* $(A^{-1})_{ij} \geq 0$ $\forall\, i, j$. Note also that $(B_{n^-})_{ij} \geq 0$ $\forall\, i, j$ for any time-step and any mesh as long as the mesh has not tangled.

Consider now the vector $U_{\min}$, which has the same length as $U^{(n+1)^-}$, U^{n^+} and U^{n^-}, but with elements all equal to $u_{\min}^{n^-}$. It follows from $(B_{n^-})_{ij} \geq 0$ $\forall\, i, j$ and $u_i^{n^-} \geq u_{\min}^{n^-}$ $\forall\, i$ that

$$(B_{n^-} U^{n^-})_i \geq (B_{n^-} U_{\min})_i \qquad \forall\, i \,. $$
(B.7)

This can be manipulated to give

$$(B_{n^-} U_{\min})_i = \sum_{j \in D_i} (B_{n^-})_{ij} u_{\min}^{n^-} = |S_i^n| u_{\min}^{n^-} \,, $$
(B.8)

using the definitions given in (B.4). Summing the elements in the rows of the top half of A gives

$$\sum_{j \in D_i} (A_{n^+,n^+})_{ij} + \sum_{j \in D_i} (A_{n^+,(n+1)^-})_{ij} = |S_i^n| \qquad \forall i \,, \tag{B.9}$$

so, from (B.8),

$$(B_{n^-} U_{\min})_i = |S_i^n| u_{\min}^{n^-} = \sum_{j \in D_i} (A_{n^+,n^+})_{ij} u_{\min}^{n^-} + \sum_{j \in D_i} (A_{n^+,(n+1)^-})_{ij} u_{\min}^{n^-} \tag{B.10}$$

$$= A_{n^+,n^+} U_{\min} + A_{n^+,(n+1)^-} U_{\min} \,.$$

Now, (B.3) gives

$$A_{n^+,n^+} U^{n^+} + A_{n^+,(n+1)^-} U^{(n+1)^-} = B_{n^-} U^{n^-} \,, \tag{B.11}$$

so it follows from (B.7), (B.10) and (B.11) that

$$(A_{n^+,n^+} U^{n^+} + A_{n^+,(n+1)^-} U^{(n+1)^-})_i \geq (A_{n^+,n^+} U_{\min} + A_{n^+,(n+1)^-} U_{\min})_i \qquad \forall i \,. \tag{B.12}$$

Summing elements in the rows of the bottom half of A gives

$$\sum_{j \in D_i} (A_{(n+1)^-,n^+})_{ij} + \sum_{j \in D_i} (A_{(n+1)^-,(n+1)^-})_{ij} = 0 \qquad \forall i \,, \tag{B.13}$$

from which it follows that

$$(A_{(n+1)^-,(n+1)^-} U_{\min} + A_{(n+1)^-,n^+} U_{\min})_i$$

$$= \sum_{j \in D_i} (A_{(n+1)^-,n^+})_{ij} u_{\min}^{n^-} + \sum_{j \in D_i} (A_{(n+1)^-,(n+1)^-})_{ij} u_{\min}^{n^-} = 0 \qquad \forall i \,, \tag{B.14}$$

so, from (B.3) and (B.14),

$$(A_{(n+1)^-,n^+} U^{n^+} + A_{(n+1)^-,(n+1)^-} U^{(n+1)^-})_i = 0 = (A_{(n+1)^-,n^+} U_{\min} + A_{(n+1)^-,(n+1)^-} U_{\min})_i \qquad \forall i \,. \tag{B.15}$$

Putting together (B.12) and (B.15),

$$\begin{bmatrix} A_{n^+,n^+} & A_{n^+,(n+1)^-} \\ A_{(n+1)^-,n^+} & A_{(n+1)^-,(n+1)^-} \end{bmatrix} \begin{bmatrix} U^{n^+} \\ U^{(n+1)^-} \end{bmatrix} \geq \begin{bmatrix} A_{n^+,n^+} & A_{n^+,(n+1)^-} \\ A_{(n+1)^-,n^+} & A_{(n+1)^-,(n+1)^-} \end{bmatrix} \begin{bmatrix} U_{\min} \\ U_{\min} \end{bmatrix}, \tag{B.16}$$

in the sense that the inequality holds for each row of the system and, since A^{-1} exists and is positive when there is no mesh tangling, we can premultiply both sides by A^{-1} to get

$$\begin{bmatrix} U^{n^+} \\ U^{(n+1)^-} \end{bmatrix} \geq \begin{bmatrix} U_{\min} \\ U_{\min} \end{bmatrix}. \tag{B.17}$$

The positivity of A^{-1} therefore implies that

$$u_{\min}^{n^-} = \min_j u_j^{n^-} \leq u_i^{n^+}, \, u_i^{(n+1)^-} \qquad \forall i \,. \tag{B.18}$$

A similar argument can be used to prove that

$$u_{\max}^{n^-} = \max_j u_j^{n^-} \geq u_i^{n^+}, \, u_i^{(n+1)^-} \qquad \forall i \,. \tag{B.19}$$

$$\square$$

We also note the following.

- This proof holds for any time-step and any meshes at the old and new time levels, as long as there is no mesh tangling, *i.e.* element areas remain positive.

- This is a global condition, not a local condition. It therefore prohibits the creation of new global extrema, but new local extrema may appear.

- A similar proof can be followed for the N scheme without the discontinuity in time but now a consistency condition, $k_{i,n} \leq 0 \;\; \forall i$ in every element, must be adhered to (the past-shield condition). This places a restriction on the time-step.

- Note that, in the context of space-time residual distribution schemes, the CFL condition is defined in terms of the past-shield condition for the element residuals (which, if satisfied, automatically implies that $u_i^{n^+} = u_i^{n^-} \;\; \forall i$, even when the approximation is allowed to be discontinuous in time), *i.e.* from Equation (19)

$$k_{i,n} \leq 0 \quad \Rightarrow \quad -\frac{\Delta t}{4}\left(\overline{\boldsymbol{a}}^n - \boldsymbol{v}_i\right) \cdot \boldsymbol{n}_i^n - \frac{|E^n|}{3} \leq 0 \,. \tag{B.20}$$

This imposes a limit on Δt which is different to the positivity condition for the pseudo-time-stepping used to find a steady-state scheme or to solve the space-time system given by (28).

Appendix C. Proof of the Consistency of Inflow Parameters for Residual Distribution Schemes

Proposition:

Under the definitions given by (18) *or* (19)

$$\sum_{i \in E} k_{i,n} + \sum_{i \in E} k_{i,n+1} = 0 \,. \tag{C.1}$$

Proof

First note that it follows immediately from (19) that

$$\sum_{i \in E} k_{i,n} = \left(\sum_{i \in E} \frac{\Delta t}{4} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^n \right) - |E^n| \,,$$

$$\sum_{i \in E} k_{i,n+1} = \left(\sum_{i \in E} \frac{\Delta t}{4} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^{n+1} \right) + |E^{n+1}| \,, \tag{C.2}$$

for any time level, since $\sum_{i \in E} \boldsymbol{n}_i = \boldsymbol{0}$. Furthermore, the definitions of the inflow parameters obtained using Simpson's rule, given in (18), also lead to (C.2) because

$$\sum_{i \in E} \left((\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot \boldsymbol{n}_k + (\boldsymbol{v}_k - \boldsymbol{v}_i) \cdot \boldsymbol{n}_j \right) = \sum_{i \in E} (\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot \boldsymbol{n}_k + \sum_{i \in E} (\boldsymbol{v}_k - \boldsymbol{v}_i) \cdot \boldsymbol{n}_j$$

$$= \sum_{i \in E} (\boldsymbol{v}_j - \boldsymbol{v}_i) \cdot \boldsymbol{n}_k + \sum_{i \in E} (\boldsymbol{v}_i - \boldsymbol{v}_j) \cdot \boldsymbol{n}_k \qquad \text{(C.3)}$$

$$= 0$$

This assumes that for vertex i of the element, vertices j and k are the other vertices, labelled anticlockwise. It therefore follows that, in both cases,

$$\sum_{i \in E} k_{i,n} + \sum_{i \in E} k_{i,n+1} = |E^{n+1}| - |E^n| + \frac{\Delta t}{4} \left(\sum_{i \in E} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^n + \sum_{i \in E} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^{n+1} \right) \,. \tag{C.4}$$

Now, by definition,

$$|E^n| = \frac{1}{2} (\boldsymbol{x}_j - \boldsymbol{x}_i) \wedge (\boldsymbol{x}_k - \boldsymbol{x}_i)$$

$$|E^{n+1}| = \frac{1}{2} \left((\boldsymbol{x}_j + \Delta t \boldsymbol{v}_j) - (\boldsymbol{x}_i + \Delta t \boldsymbol{v}_i) \right) \wedge \left((\boldsymbol{x}_k + \Delta t \boldsymbol{v}_k) - (\boldsymbol{x}_i + \Delta t \boldsymbol{v}_i) \right) \,, \tag{C.5}$$

where i is any vertex of the element and j and k are again the other vertices, labelled

anticlockwise, so

$$
\begin{aligned}
|E^{n+1}| - |E^n| &= \frac{1}{2} \left((\boldsymbol{x}_j - \boldsymbol{x}_i) + \Delta t \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \right) \wedge \left((\boldsymbol{x}_k - \boldsymbol{x}_i) + \Delta t \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \right) \\
&\quad - \frac{1}{2} \left(\boldsymbol{x}_j - \boldsymbol{x}_i \right) \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_i \right) \\
&= \frac{\Delta t}{2} \left(\boldsymbol{x}_j - \boldsymbol{x}_i \right) \wedge \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) + \frac{\Delta t}{2} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_i \right) \\
&\quad + \frac{\Delta t^2}{2} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \wedge \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \\
&= \frac{\Delta t}{2} \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \wedge \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) + \frac{\Delta t}{2} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_i \right) \\
&\quad + \frac{\Delta t^2}{2} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \wedge \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \\
&= \frac{\Delta t}{2} \boldsymbol{v}_i \wedge \left(\boldsymbol{x}_j - \boldsymbol{x}_k \right) + \frac{\Delta t}{2} \boldsymbol{v}_j \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_i \right) + \frac{\Delta t}{2} \boldsymbol{v}_k \wedge \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \\
&\quad + \frac{\Delta t^2}{2} \left(\boldsymbol{v}_j - \boldsymbol{v}_i \right) \wedge \left(\boldsymbol{v}_k - \boldsymbol{v}_i \right) \\
&= \frac{\Delta t}{2} \left(\sum_{i \in E} \boldsymbol{v}_i \wedge \left(\boldsymbol{x}_j - \boldsymbol{x}_k \right) \right) + \frac{\Delta t^2}{2} \left(\boldsymbol{v}_i \wedge \boldsymbol{v}_j + \boldsymbol{v}_j \wedge \boldsymbol{v}_k + \boldsymbol{v}_k \wedge \boldsymbol{v}_i \right)
\end{aligned}
\tag{C.6}
$$

Similar manipulation, using $\boldsymbol{n}_i \perp \boldsymbol{x}_k - \boldsymbol{x}_j$, leads to

$$
\begin{aligned}
&\frac{\Delta t}{4} \left(\sum_{i \in E} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^n + \sum_{i \in E} \boldsymbol{v}_i \cdot \boldsymbol{n}_i^{n+1} \right) \\
&= \frac{\Delta t}{4} \left(\sum_{i \in E} \boldsymbol{v}_i \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_j \right) + \sum_{i \in E} \boldsymbol{v}_i \wedge \left(\left(\boldsymbol{x}_k + \Delta t \boldsymbol{v}_k \right) - \left(\boldsymbol{x}_j + \Delta t \boldsymbol{v}_j \right) \right) \right) \\
&= \frac{\Delta t}{4} \left(\sum_{i \in E} \boldsymbol{v}_i \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_j \right) + \sum_{i \in E} \boldsymbol{v}_i \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_j \right) + \sum_{i \in E} \Delta t \boldsymbol{v}_i \wedge \left(\boldsymbol{v}_k - \boldsymbol{v}_j \right) \right) \\
&= \frac{\Delta t}{2} \left(\sum_{i \in E} \boldsymbol{v}_i \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_j \right) \right) + \frac{\Delta t^2}{4} \left(\sum_{i \in E} \boldsymbol{v}_i \wedge \left(\boldsymbol{v}_k - \boldsymbol{v}_j \right) \right) \\
&= \frac{\Delta t}{2} \left(\sum_{i \in E} \boldsymbol{v}_i \wedge \left(\boldsymbol{x}_k - \boldsymbol{x}_j \right) \right) - \frac{\Delta t^2}{2} \left(\boldsymbol{v}_i \wedge \boldsymbol{v}_j + \boldsymbol{v}_j \wedge \boldsymbol{v}_k + \boldsymbol{v}_k \wedge \boldsymbol{v}_i \right)
\end{aligned}
\tag{C.7}
$$

Hence, substituting (C.6) and (C.7) into (C.4) gives

$$
\sum_{i \in E} k_{i,n} + \sum_{i \in E} k_{i,n+1} = 0 \,.
\tag{C.8}
$$

$\square$

References

[1] R. Abgrall, Residual distribution schemes: current status and future trends, Comput. Fluids 35 (7) (2006) 641–669.

[2] H. Deconinck, M. Ricchiuto, Residual distribution schemes: foundation and analysis, in: Encyclopedia of Computational Mechanics, Vol. 3, Wiley, 2007.

[3] R. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2002.

[4] R. H. Pletcher, J. C. Tannehill, D. A. Anderson, Computational Fluid Mechanics and Heat Transfer, 3rd Edition, CRC Press, 2012.

[5] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, 3rd Edition, Springer-Verlag, Berlin, 2009.

[6] M. J. Baines, M. E. Hubbard, Multidimensional upwinding and grid adaptation, in: Numerical Methods for Wave Propagation, Vol. 47 of Fluid Mechanics and its Applications, Kluwer Academic Publishers, 1998, pp. 33–54.

[7] M. Ricchiuto, An explicit residual based approach for shallow water flows, J. Comput. Phys. 80 (2015) 306–344.

[8] A. Warzyński, M. E. Hubbard, M. Ricchiuto, Runge-Kutta residual distribution schemes, J. Sci. Comput. 62 (3) (2015) 772–802.

[9] Á. Csík, H. Deconinck, Space-time residual distribution schemes for hyperbolic conservation laws on unstructured linear finite elements, Int. J. Numer. Meth. Fl. 40 (3-4) (2002) 573–581, ICFD Conference on Numerical Methods for Fluid Dynamics, Part II (Oxford, 2001).

[10] D. Sármány, M. E. Hubbard, M. Ricchiuto, Unconditionally stable space-time discontinuous residual distribution for shallow-water flows, J. Comput. Phys. 253 (2013) 86–113.

[11] L. Arpaia, M. Ricchiuto, r-adaptation for shallow water flows: conservation, well-balancedness, efficiency, Comput. Fluids 160 (2016) 175–203.

[12] J. Dobeš, H. Deconinck, An ALE formulation of the multidimensional residual distribution scheme for computations on moving meshes, in: Deconinck, H., Dick, E. (eds) Computational Fluid Dynamics 2006, Springer, Berlin, Heidelberg, 2009.

[13] C. Michler, H. De Sterck, H. Deconinck, An arbitrary Lagrangian Eulerian formulation for residual distribution schemes on moving grids, Comput. Fluids 32 (1) (2003) 59–71.

[14] H. D. Ceniceros, T. Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, J. Comput. Phys. 172 (2008) 609–639.

[15] T. Tang, Moving mesh methods for computational fluid dynamics, Contemp. Math. 383 (2005) 141–173.

[16] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids, J. Comput. Phys. 174 (2) (2000) 669–694.

[17] M. J. Baines, M. E. Hubbard, P. K. Jimack, Velocity-based moving mesh methods for nonlinear partial differential equations, Commun. Comput. Phys. 10 (3) (2011) 509–576.

[18] C. J. Budd, W. Huang, R. D. Russell, Adaptivity with moving grids, Acta Numer. 18 (2009) 111–241.

[19] L. Arpaia, M. Ricchiuto, R. Abgrall, An ALE formulation for explicit Runge-Kutta residual distribution, J. Sci. Comput. 63 (2015) 502–547.

[20] A. Bermudez, M. E. Vazquez-Cendon, Upwind methods for hyperbolic conservation laws with source terms, Comput. Fluids 23 (8) (1994) 1049–1071.

[21] M. E. Hubbard, P. Garcia-Navarro, Flux difference splitting and the balancing of source terms and flux gradients, J. Comput. Phys. 165 (1) (2000) 89–125.

[22] J. Murillo, A. Navas-Montilla, A comprehensive explanation and exercise of the source terms in hyperbolic systems using Roe-type solutions. application to the 1d-2d shallow water equations, Adv. Water Resour. 98 (2016) 70–96.

[23] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, Comput. Method Appl. M. 134 (1-2) (1996) 71–90.

[24] P. Wesseling, Principles of Computational Fluid Dynamics, Vol. 29 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2001.

[25] R. Abgrall, A. Lerat, M. Ricchiuto, Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes, J. Comput. Phys. 230 (11) (2011) 4103–4136.

[26] R. Abgrall, P. L. Roe, High order fluctuation schemes on triangular meshes, J. Sci. Comput. 19 (1-3) (2003) 3–36.

[27] M. Ricchiuto, R. Abgrall, H. Deconinck, Construction of very high order residual distribution schemes for unsteady advection: preliminary results, in: VKI LS 2003-05, 33rd CFD Course, von Karman Institute for Fluid Dynamics, 2003.

[28] M. E. Hubbard, A. L. Laird, Achieving high-order fluctuation splitting schemes by extending the stencil, Comput. Fluids 34 (2005) 443–459.

[29] R. Abgrall, High order schemes for hyperbolic problems using globally continuous approximation and avoiding mass matrices, J. Sci. Comput. 73 (2017) 461–494.

[30] R. Abgrall, P. Bacigaluppi, S. Tokareva, High-order residual distribution scheme for the time-dependent Euler equations of fluid dynamics, Comput. Math. Appl. 78 (2) (2019) 274–297.

[31] N. Villedieu, High order discretisation by residual distribution schemes, Ph.D. thesis, Universitat Libre de Bruxelles (2009).

[32] N. Villedieu, L. Koloszar, T. Quintino, H. Deconinck, Unsteady high order residual distribution schemes with applications to linearised Euler equations, in: Numerical Mathematics and Advanced Applications, Springer, 2009, pp. 911–919.

[33] M. E. Hubbard, M. Ricchiuto, Discontinuous upwind residual distribution: A route to unconditional positivity and high order accuracy, Comput. Fluids 46 (1) (2011) 263–269.

[34] M. Ricchiuto, Contributions to the development of residual discretizations for hyperbolic conservation laws with application to shallow water flows, HDR Thesis (2011).

[35] H. Deconinck, P. L. Roe, R. Struijs, A multidimensional generalization of Roe's flux difference splitter for the Euler equations, Comput. Fluids 22 (2-3) (1993) 215–222.

[36] Von Karman Institute of Fluid Dynamics, Lecture Series, High Order Discretization Methods, 34th Computational Fluid Dynamics Course (November 2005).

[37] M. Ricchiuto, Á. Csík, H. Deconinck, Residual distribution for general time-dependent conservation laws, J. Comput. Phys. 209 (1) (2005) 249–289.

[38] S. K. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, Mat. Sb. (N.S.) 47 (89) (1959) 271–306.

[39] H. Deconinck, K. Sermeus, R. Abgrall, Status of multidimensional upwind residual distribution schemes and applications in aeronautics, in: AIAA Paper 2000-2328, AIAA, 2000.

[40] M. Ricchiuto, N. Villedieu, R. Abgrall, H. Deconinck, High order residual distribution schemes: discontinuity capturing crosswind dissipation and extension to advection diffusion, in: VKI LS 06-01, 34th CFD Course, von Karman Institute for Fluid Dynamics, 2005.

[41] Á. Csík, M. Ricchiuto, H. Deconinck, A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws, J. Comput. Phys. 179 (1) (2002) 286–312.

[42] M. Ricchiuto, R. Abgrall, H. Deconinck, Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes, J. Comput. Phys. 222 (1) (2007) 287–331.

[43] A. Csík, M. Ricchiuto, H. Deconinck, Space time residual distribution schemes for hyperbolic conservation laws over linear and bilinear elements, in: VKI LS 2003-05, 33rd CFD Course, von Karman Institute for Fluid Dynamics, 2003.

[44] R. Abgrall, M. Mezine, Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems, J. Comput. Phys. 188 (1) (2003) 16–55.

[45] K. Sermeus, H. Deconinck, An entropy fix for multi-dimensional upwind residual distribution schemes, Comput. Fluids 34 (4-5) (2005) 617–640.

[46] D. Sármány, M. E. Hubbard, Upwind residual distribution for shallow-water ocean modelling, Ocean Model. 64 (1-2) (2013) 1–11.

[47] B. D. Rogers, A. Borthwick, P. Taylor, Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Reimann solver, J. Comput. Phys. 192 (2003) 422–451.

[48] R. Abgrall, Toward the ultimate conservative scheme: following the quest, J. Comput. Phys. 167 (2) (2001) 277–315.

[49] S. Rhebergen, Well-balanced r-adaptive and moving mesh space-time discontinuous Galerkin method for the shallow water equations, Oxford University Mathematical Institute Technical Report 1757 (2013).

[50] M. Ricchiuto, On the C-property and generalized C-property of residual distribution for the shallow water equations, J. Sci. Comput. 48 (1-3) (2011) 304–318.

[51] M. Ricchiuto, A. Bollermann, Stabilized residual distribution for shallow water simulations, J. Comput. Phys. 228 (4) (2009) 1071–1115.

[52] M. E. Hubbard, Multidimensional upwinding and grid adaptation for conservation laws, Ph.D. thesis, Department of Mathematics, University of Reading (1996).

[53] J. Dobeš, H. Deconinck, Second order blended multidimensional upwind residual distribution scheme for steady and unsteady computations, J. Comput. Appl. Math. 215 (2) (2008) 378–389.

[54] M. Seaïd, Non-oscillatory relaxation methods for the shallow-water equations in one and two space dimensions, Int. J. Numer. Meth. Fl. 46 (5) (2004) 457–484.