



HAL
open science

Space-time residual distribution on moving meshes

Matthew Hubbard, Mario Ricchiuto, Domokos Sarmany

► **To cite this version:**

Matthew Hubbard, Mario Ricchiuto, Domokos Sarmany. Space-time residual distribution on moving meshes. *Computers & Mathematics with Applications*, 2020, 10.1016/j.camwa.2019.09.019 . hal-02310394

HAL Id: hal-02310394

<https://inria.hal.science/hal-02310394>

Submitted on 10 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Space-time residual distribution on moving meshes

M.E. Hubbard^{a,*}, M. Ricchiuto^b, D. Sármany^{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. Sármany)

¹Present 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-

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

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

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

68 2. Residual Distribution on Moving Meshes

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

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

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

89 2.1. Scalar Equations

90 Consider the scalar conservation law

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

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

$$\phi_{E_t}^n = \int_{t^n}^{t^{n+1}} \int_{E(t)} \left(\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} \right) d\Omega dt. \quad (2)$$

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

$$\int_{t^n}^{t^{n+1}} \int_{\Omega(t)} \left(\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} \right) d\Omega dt = \sum_{E_t^n \in (\Omega_h^t)^n} \phi_{E_t}^n, \quad (3)$$

96 provided that the solution u is bounded and piecewise differentiable.

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

$$\frac{d}{dt} \int_{E(t)} u d\Omega = \int_{E(t)} \frac{\partial u}{\partial t} d\Omega + \int_{\partial E(t)} u \mathbf{v} \cdot \hat{\mathbf{n}} d\Gamma, \quad (4)$$

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

$$\phi_{E_t}^n = \int_{E(t^{n+1})} u(t^{n+1}) d\Omega - \int_{E(t^n)} u(t^n) d\Omega + \int_{t^n}^{t^{n+1}} \int_{\partial E(t)} (\mathbf{f} - u\mathbf{v}) \cdot \hat{\mathbf{n}} d\Gamma, \quad (5)$$

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

107 *2.1.1. Distributing the Residual*

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

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

- 126 1. In every space-time element, replace the unknown variable u with an approximation
 127 u_h that varies linearly along the edges of the element.
- 128 2. Transform the element residual into a space-time boundary integral:

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

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

131 In this work, it is assumed that the edges of each space-time element are straight
 132 lines (the mesh node velocities are assumed to be constant within each time-step)
 133 and that the surface of each lateral face is defined by the bilinear interpolant of its
 134 four vertices. The outward-pointing normal to a lateral face \mathbf{n}_t is therefore parallel to
 135 $(0, \mathbf{e}) \wedge (1, \mathbf{v})$, in which \mathbf{e} is the anticlockwise-oriented edge of the triangle obtained
 136 from the intersection of the space-time element with the \mathbf{x} -plane at $t \in [t^n, t^{n+1}]$
 137 (see Figure 1) and \mathbf{v} is the local mesh velocity, defined by linear interpolation of the
 138 velocities at the two vertices defining this edge. Since \mathbf{v} at each spatial mesh node
 139 is assumed to be constant in time, it is simple to show that

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

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

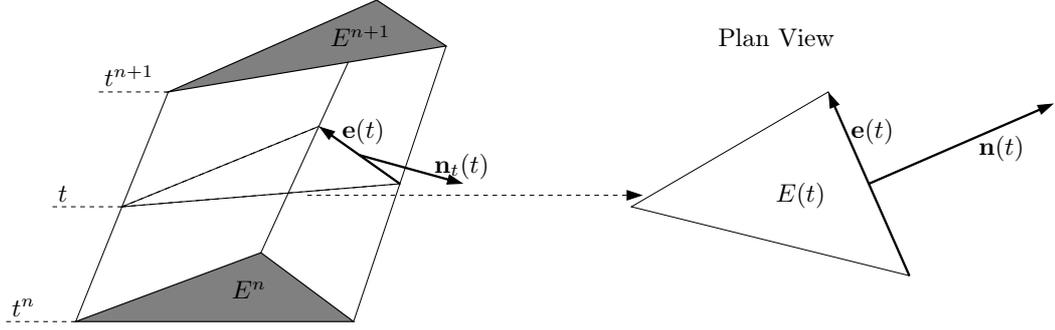


Figure 1: Geometry of the space-time element (a bilinear transformation of a right triangular prism) for the moving mesh algorithm.

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

$$\begin{aligned}
 \phi_{E_t} &= \int_{E^{n+1}} u_h^{n+1} d\Omega - \int_{E^n} u_h^n d\Omega + \int_{\text{quads}} \mathbf{f}_t \cdot \hat{\mathbf{n}}_t d\Gamma_t \\
 &= \int_{E^{n+1}} u_h^{n+1} d\Omega - \int_{E^n} u_h^n d\Omega + \int_{t^n}^{t^{n+1}} \int_{E(t)} (\mathbf{f}_h \cdot \hat{\mathbf{n}} - u_h \hat{\mathbf{n}} \cdot \mathbf{v}) d\Gamma dt \\
 &\approx \int_{E^{n+1}} u_h^{n+1} d\Omega - \int_{E^n} u_h^n d\Omega \\
 &+ \frac{\Delta t}{2} \left\{ \int_{\partial E^n} (\mathbf{f}(u_h^n) - u_h^n \mathbf{v}) \cdot \hat{\mathbf{n}} d\Gamma + \int_{\partial E^{n+1}} (\mathbf{f}(u_h^{n+1}) - u_h^{n+1} \mathbf{v}) \cdot \hat{\mathbf{n}} d\Gamma \right\} \\
 &= \int_{E^{n+1}} u_h^{n+1} d\Omega - \int_{E^n} u_h^n d\Omega + \frac{\Delta t}{2} (\phi_E^n + \phi_E^{n+1}),
 \end{aligned} \tag{8}$$

145 in which

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

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

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

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

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

155 in which the distribution coefficients are defined such that

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

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

$$\psi_E = \int_{\partial E_t} \mathbf{f}_t \cdot \hat{\mathbf{n}}_t \, d\Gamma_t = \int_{\partial E} [u_h^n] \, d\Omega = \frac{|E^n|}{3} \sum_{i \in E} [u_i^n], \quad (12)$$

160 where $|E^n|$ is the area of the triangular element E^n , $[u^n] = u^{n+} - u^{n-}$ (in which
 161 $u^{n\pm} = \lim_{t \rightarrow (t^n)\pm} u_h$) represents the jump across the triangular face and piecewise
 162 linear variation of u_h in space has been assumed. The \pm superscripts are suppressed
 163 in the majority of what follows: within a space-time element n and $n + 1$ implicitly
 164 indicate n^+ and $(n + 1)^-$, respectively, and ψ_E is evaluated at time level n .

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

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

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

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

$$\begin{aligned} \sum_{E \in D_i} (\phi_{i,n}^E + \psi_{i,n}^E) &= 0 \\ \sum_{E \in D_i} \phi_{i,n+1}^E &= 0 \end{aligned} \quad (14)$$

179 $\forall i \in \Omega_h$ at each time-step.

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

182 *2.1.2. Evaluating the Residual*

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

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

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

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

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

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

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

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

201 where, from (15),

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

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

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

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

207 In fact, for both approximations,

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

208 so either (18) or (19) can be used to define the inflow parameters in any of the standard
 209 residual distribution schemes (proof provided in Appendix C). For simplicity and clarity,
 210 we will use the trapezoidal rule approximation (19) from now on, since the overall scheme
 211 remains second-order accurate.

212 There are two important consequences of (20).

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

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

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

- 217 2. Any conservative residual distribution scheme, one for which

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

218 will satisfy a discrete geometric conservation law (DGCL) [16], *i.e.* the trivial solu-
 219 tion $u \equiv \text{constant}$ is preserved when the mesh is moved.

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

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

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

225 *2.1.3. Evaluating the Distribution Coefficients*

226 The typical design requirements for RD schemes include [36]

- 227 • *positivity*, which warrants that the numerical approximations are free of spurious
228 oscillations;
- 229 • *linearity preservation*, which ensures that a $(k-1)$ th-order polynomial representation
230 of u_h leads to a k th-order accurate scheme [26];
- 231 • *conservation*, which guarantees that discontinuities are captured correctly and that
232 the approximation converges to a weak solution of (1) as the mesh is refined;
- 233 • *compactness*, which is primarily for computational efficiency and requires that the
234 element residual be distributed to its own vertices only; and
- 235 • *continuous dependence* of the coefficients on both the solution and the advection
236 velocity, which enhances the iterative convergence of any algebraic solver applied to
237 (14).

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

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

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

$$\begin{aligned}
 u_t^{\text{in}} &= N_t \left(\sum_{j \in E} k_{j,n}^+ u_j^n + \sum_{j \in E} k_{j,n+1}^+ u_j^{n+1} - \phi_{E_t} \right), \\
 (\phi_{i,n}^E)^N &= k_{i,n}^+ (u_i^n - u_t^{\text{in}}), \quad (\phi_{i,n+1}^E)^N = k_{i,n+1}^+ (u_i^{n+1} - u_t^{\text{in}}),
 \end{aligned} \tag{24}$$

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

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

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

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

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

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

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

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

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

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

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

272 2.1.4. Solving the Discrete System

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

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

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

$$\tau_i \leq \frac{1}{\sum_{E \in D_i} k_{i,n}^+ + \sum_{E \in D_i} k_{i,n+1}^+}. \quad (29)$$

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

281 *2.2. Nonlinear Systems*

282 Consider now the nonlinear hyperbolic system of conservation laws,

$$\partial_t U + \nabla \cdot \mathbf{F} = 0 \quad \text{or} \quad \partial_t U + \mathcal{A}(U) \cdot \nabla U = 0, \quad (30)$$

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

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

286 *2.2.1. Evaluating the Residual*

287 The residuals in (31) may be approximated as

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

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

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

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

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

292 For some systems, including the shallow water equations, Φ_{E_t} and $\overline{\mathcal{A}}$ can be derived
 293 from a conservative linearisation [35], so that (32) is an exact evaluation of the discrete
 294 residual. This approach is described in more detail in [10]. **However, if the conservative**
 295 **linearisation is not known or its implementation is not practical, it is still possible to**
 296 **achieve conservation by evaluating the space-time element residual directly from (31) via**
 297 **a higher-order quadrature rule – this is one of the concepts behind the CRD schemes**

298 [41, 42]. The \mathcal{K} matrices in (34), which are required by the distribution schemes, are then
 299 evaluated using appropriate averaged states which depend on the time level at which
 300 $\overline{\mathcal{A}}$ is being evaluated. In this work, for the shallow water equations, $\overline{\mathcal{A}}^n$ and $\overline{\mathcal{A}}^{n+1}$ are
 301 evaluated using the arithmetic means of the spatial element vertex values of the primitive
 302 variables, depth and velocity, at their respective time levels [10].

303 2.2.2. Evaluating the Distribution Coefficients

304 Assuming that the inflow matrices defined in (34) are diagonalisable, we have $\mathcal{K}_{i,n} =$
 305 $\mathcal{R}\mathcal{D}\mathcal{R}^{-1}$, where \mathcal{D} is the diagonal matrix of eigenvalues, \mathcal{R}^{-1} is the matrix of the left and \mathcal{R}
 306 of the right eigenvectors. Defining $\mathcal{D}^\pm = \frac{1}{2}(\mathcal{D} \pm |\mathcal{D}|)$ with $|\mathcal{D}|$ denoting the absolute values
 307 of the entries, we can generalise the inflow parameters defined in (21) to the matrices

$$\begin{aligned} \mathcal{K}_{i,n}^+ &= (\mathcal{R}\mathcal{D}^+\mathcal{R}^{-1})_{i,n}, & \mathcal{K}_{i,n+1}^+ &= (\mathcal{R}\mathcal{D}^+\mathcal{R}^{-1})_{i,n+1}, \\ \mathcal{K}_{i,n}^- &= (\mathcal{R}\mathcal{D}^-\mathcal{R}^{-1})_{i,n}, & \mathcal{K}_{i,n+1}^- &= (\mathcal{R}\mathcal{D}^-\mathcal{R}^{-1})_{i,n+1}, \end{aligned} \quad (35)$$

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

308 These matrices are used to define upwind RD schemes along the lines of the scalar case
 309 (24)–(26). Note that the presence of the volume term in the space-time residual, which
 310 corresponds to the time derivative, ensures that the eigenvalues of the \mathcal{K} parameters can-
 311 not all vanish simultaneously [9, 43]. Consequently, \mathcal{N}_t can always be evaluated without
 312 any need for regularisation.

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

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

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

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

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

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

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

322 The matrix blending parameter Θ can be computed in a number of different ways
 323 [44, 45]. The simplest generalisation of (27) is

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

324 where the division should be understood as an element-wise operation. Another
 325 possibility, proposed in [44] and used in this work, is to choose a particular direction
 326 $\boldsymbol{\xi} = (\xi_x, \xi_y)$ and compute the decomposition $\mathcal{A} \cdot \boldsymbol{\xi} = \mathcal{R}_\xi \mathcal{D}_\xi \mathcal{R}_\xi^{-1}$. The blending (38)
 327 is then carried out on the ‘‘characteristic’’ residuals

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

328 with the blending parameter computed as

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

329 where we drop the superscript E to avoid clutter. Finally, we calculate the
 330 blended residuals based on the original variables by $\Phi_{i,n}^B = (\mathcal{R}_\xi)_n \Phi_{i,n}^B$ and $\Phi_{i,n+1}^B =$
 331 $(\mathcal{R}_\xi)_{n+1} \Phi_{i,n+1}^B$. In this work, in order to aid iterative convergence within each time-
 332 step, the blending parameter, Θ_2 , is fixed after the first pseudo-time-step.

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

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

335 and the corresponding contributions are defined as [33]

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

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

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

338 $\forall i \in \Omega_h$, which needs to be solved at each time-step. Boundary conditions are applied as
 339 described in [46].

340 *2.2.3. Solving the Discrete System*

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

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

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

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

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

347 *2.3. Application to the Shallow Water System*

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

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

350 where, in two space dimensions,

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

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

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

366 *2.3.1. Preserving Hydrostatic Balance*

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

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

$$\Phi_{E_t}^b = \Delta t \frac{g\bar{d}}{2} \sum_{i \in E} \begin{bmatrix} 0 \\ b_i \mathbf{n}_i \end{bmatrix} \approx \int_{t^n}^{t^{n+1}} \int_{E(t)} S(U) \, d\Omega \, dt, \quad (49)$$

377 where \mathbf{n}_i is the outward normal to the edge opposite vertex i scaled by the length of the
 378 edge. In order for the STN scheme (36) (and the corresponding STB schemes) to satisfy
 379 hydrostatic balance, a slightly modified version of (36) must be applied. For fixed bed
 380 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
 381 formulation, using the arithmetic mean $\bar{\eta}$ in the approximation of the spatial integral of
 382 the time derivative term. This leads to

$$V_t^{\text{in}} = \mathcal{N}_t \left(\sum_{j \in E} \mathcal{K}_{j,n}^+ V_j^n + \sum_{j \in E} \mathcal{K}_{j,n+1}^+ V_j^{n+1} - \Phi_{E_t} - \Phi_{E_t}^b \right), \quad (50)$$

$$(\Phi_{i,n}^E)^N = \mathcal{K}_{i,n}^+ (V_i^n - V_t^{\text{in}}), \quad (\Phi_{i,n+1}^E)^N = \mathcal{K}_{i,n+1}^+ (V_i^{n+1} - V_t^{\text{in}}).$$

383 A proof that this modified N scheme satisfies the C-property on fixed meshes is given in
 384 [10].

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

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

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

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

393 The details of the proof are given in Appendix A.

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

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

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

$$|E^{n+1}| \bar{b}^{n+1} - |E^n| \bar{b}^n - \frac{\Delta t}{2} \sum_{i \in E} (b_i^{n+1} \mathbf{v}_i \cdot \mathbf{n}_i^{n+1} + b_i^n \mathbf{v}_i \cdot \mathbf{n}_i^n) = 0. \quad (54)$$

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

404 2.4. The Moving Mesh Algorithm

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

448 2.4.1. Implementation Details

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

- 452 1. Update the mesh node positions, using one iteration of (55)/(56), with the chosen
 453 element weights W_E , limiting the node movement according to (57) to avoid mesh
 454 tangling.
- 455 2. Smooth the moved mesh using one relaxed Laplacian smoothing iteration, (55) with
 456 $W_E \equiv 1$, and limiting the node movement according to (57) to avoid mesh tangling.
- 457 3. Recompute the initial conditions and (if required) the bed topography at the new
 458 mesh node positions.

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

- 461 1. The following steps are repeated until the stopping criteria for the adaptation are
462 satisfied.
- 463 (a) Update the mesh node positions at the new time level, using one iteration of
464 (55)/(56), with the chosen element weights W_E , and limiting the node move-
465 ment according to (57) to avoid mesh tangling.
- 466 (b) Smooth the mesh at the new time level using one relaxed Laplacian smoothing
467 iteration, (55) with $W_E \equiv 1$, and limiting the node movement according to
468 (57) to avoid mesh tangling.
- 469 (c) If required, recompute the bed topography at the new mesh node positions.
- 470 (d) Carry out one iteration of the pseudo-time-stepping, (28) or (45), using the
471 residuals calculated on the current distorted space-time prismatic elements
472 and the chosen distribution scheme.
- 473 2. Fix the nodal positions for the new time level, but continue the pseudo-time-stepping
474 iterations, (28) or (45), using the residuals calculated on the distorted space-time
475 prismatic elements and the chosen distribution scheme, until the stopping criteria
476 for the pseudo-time-stepping are satisfied.
- 477 3. Replace the mesh at the old time level with the mesh at the new time level, including
478 the updated bed topography and solution.

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

$$\text{rel_tol} = \frac{\|(\Psi^n, \Psi^{n+1})_m^T\|_1}{\|(\Psi^n, \Psi^{n+1})_0^T\|_1} < \text{REL_TOL}, \quad (61)$$

481 where, throughout this work, $\text{REL_TOL} = 10^{-3}$ for the pseudo-time-stepping. In addi-
482 tion, there is one intermediate stopping criterion: when the mesh is moved according to
483 the weighted averaging defined by (55)/(56), the node positions are fixed when either
484 $\text{rel_tol} \leq \text{REL_TOL_MOVE}$ is satisfied or a specified number of iterations (N_{move}) has
485 been reached.

486 2.4.2. Parameter Choices

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

- 491 • In Equation (59) α governs the strength of the effect of local features. Increasing
492 its value will increase the resolution of the mesh where the first or second derivative
493 of the solution is relatively high.
- 494 • In Equation (60) β and γ govern the extent of the region where the mesh is affected
495 by local features. Increasing their values reduces the size of the region over which
496 the mesh is adapted. The adaptation becomes a pure smoothing step if $\beta, \gamma \geq 1$.

- 497 • The larger the values of N_{init} and N_{move} , the maximum numbers of mesh movement
498 iterations allowed at each stage, the closer the initial mesh and the meshes at each
499 time-step get to equidistributing the monitor function. Reducing `REL_TOL_MOVE`
500 (when it is active) has a similar effect.
- 501 • The smoothness with which the mesh changes resolution from one region to another
502 can be adjusted using the relaxation parameter μ in Equation (56). Increasing
503 its value in the smoothing iteration (where $W_E \equiv 1$) will tend to make the mesh
504 elements more uniform in size.

505 3. Numerical Results

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

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

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

515 3.1. Scalar Advection

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

$$\frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u = 0, \quad (63)$$

519 with u specified on inflow boundaries.

520 3.1.1. Circular Advection

521 This test case considers advection of a smooth (C^3) initial profile through the domain
522 $\Omega = [-1, 1] \times [-1, 1]$ with velocity $\mathbf{a} = (-y, x)$. The exact time-dependent solution profile
523 is given by

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

524 in which $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ is the radius of the non-zero region, $x_c = 0.5 \sin(t -$
525 $0.5\pi)$ and $y_c = 0.5 \cos(t - 0.5\pi)$. The initial conditions are given at $t = 0$ and the boundary
526 conditions were set to $u(x, y, t) = 0$ on the inflow sections of the boundary.

527 The accuracy and efficiency of the simulations were assessed for a sequence of fixed,
528 uniform, unstructured triangular meshes with characteristic mesh sizes $h = \frac{1}{5}, \frac{1}{10}, \frac{1}{20},$
529 $\frac{1}{40}, \frac{1}{80}$. These were also used as the initial meshes for simulations carried out using the

530 adaptive procedure described in Section 2.4 and, to demonstrate the robustness of the
 531 space-time residual distribution algorithm on moving meshes, imposing movement on
 532 each mesh node using

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

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

535 The accuracy and efficiency of the approximations obtained using the STB scheme
 536 are illustrated in Figure 2. The mesh movement used to improve the fit to the initial
 537 conditions is included in the cpu times. Results are shown for different adaptive strategies
 538 for deciding when to terminate the mesh movement iterations within each time-step:
 539 after a fixed number of iterations, $N_{\text{move}} = 5, 10$ or 20 in this case; or after the space-
 540 time residual norm has dropped below a given tolerance, $\text{REL_TOL_MOVE} = 10^{-1}$. In all
 541 adaptive cases the initial mesh is created by applying $N_{\text{init}} = 200$ iterations of the mesh
 542 movement algorithm to the fixed mesh, the monitor function parameters are chosen to be
 543 $\alpha = 100$, $\beta = \gamma = 0.2$, and the relaxation parameter for the mesh smoothing was $\mu = 0.5$.
 544 These values have been chosen to obtain an efficient adaptive strategy for a broad range
 545 of mesh sizes: they are not chosen to provide the most accurate approximations. For the
 546 time-stepping, $\text{CFL} = 2.0$ and $\text{CFL}_\tau = 0.9$.

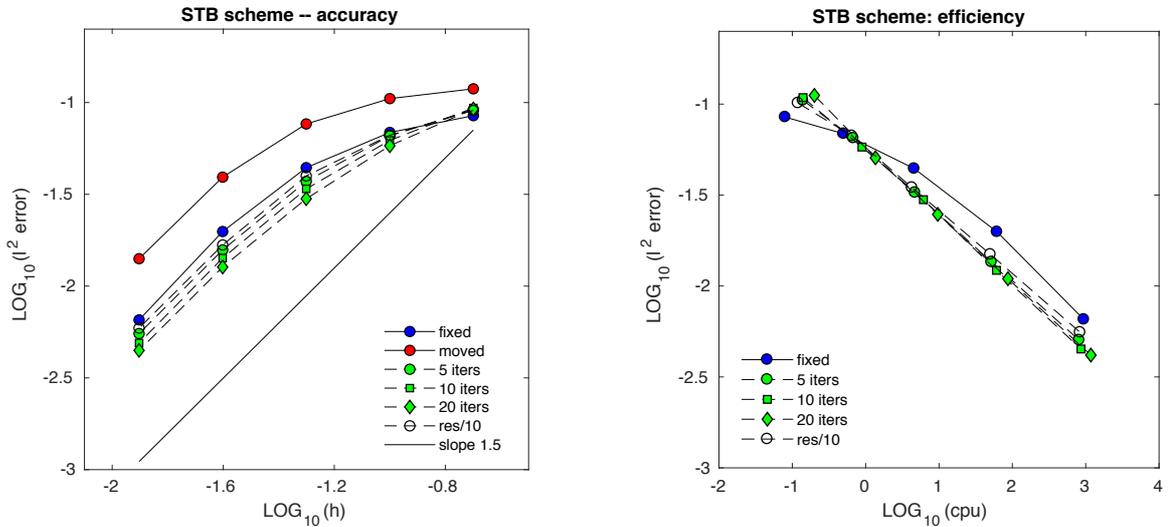


Figure 2: Comparison of accuracy and efficiency for circular advection ($t = 2\pi$), using the STB scheme on fixed meshes (blue circles, solid line), adapted meshes (green or empty symbols, dashed line) and meshes moved according to (65) (red symbols, solid line).

547 The same CFL is chosen for the time-stepping in both the fixed and moving mesh
 548 simulations. $\text{CFL} = 2$ is close to the optimal value for the efficiency of the fixed mesh
 549 computations [10]: fewer time-steps are required if the CFL is increased further, but the
 550 overall computation time decreases very little because the magnitude of the upper limit
 551 on the pseudo-time-step, Equation (29), is inversely proportional to the time-step, so
 552 the total number of pseudo-time-steps remains roughly constant. Furthermore, the error

553 starts to increase, so running the adaptive computations at a larger CFL (*e.g.* using the
 554 same time-step as the fixed mesh instead of the same CFL) doesn't necessarily improve
 555 efficiency.

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

561 Using between 10 and 20 mesh movement iterations per time-step seems to be the
 562 most efficient adaptive strategy in this case. At the finer mesh resolutions it consistently
 563 provides approximately a 30% reduction in error for a given cpu time. Adapting the
 564 meshes locally reduces the element size where the solution is non-zero to provide smaller
 565 errors but, since the CFL is fixed, the time-step sizes will also be reduced if the mesh
 566 movement increases the resolution in regions of high flow velocity, so more time-steps may
 567 be required. Table 1 suggests that is not the case here (the fastest flow is at the corners
 568 of the square domain) and that the mesh movement has a beneficial effect on the iterative
 569 convergence rates of the pseudo-time-stepping. However, each mesh movement iteration
 570 requires about 3 times as much cpu time as each residual distribution step (largely due
 571 to the expense of computing the weights and the checks for mesh tangling) so overall cpu
 572 times are not often reduced. As noted previously, increasing the CFL does not improve
 573 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}
$N_{\text{move}} = 5$	385	16204	42.09	1925	52.04	1.354×10^{-2}
$N_{\text{move}} = 10$	390	15615	40.04	3900	60.70	1.219×10^{-2}
$N_{\text{move}} = 20$	403	17855	44.31	8060	87.07	1.096×10^{-2}
$\text{REL_TOL_MOVE} = 10^{-1}$	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.

574 The node velocities and the mesh in the bottom-right quadrant of the domain at $t = \frac{\pi}{4}$,
 575 for an adaptive strategy with 20 mesh movement iterations per time-step, are shown on
 576 the left in Figure 3. Ideally, the mesh would move with the advection velocity, in which
 577 case the initial profile would be preserved by our scheme to machine accuracy. Instead
 578 the nodes are continually being recruited at the leading edge of the moving profile and
 579 discarded at the trailing edge in order to retain high resolution where the solution is
 580 non-zero: the mesh inside this region moves very little. The right-hand plots in Figure
 581 3 show the node velocities and mesh obtained at $t = \frac{\pi}{4}$ when $N_{\text{init}} = 5000$ and $N_{\text{move}} =$
 582 500, which approaches a situation where the monitor function is equidistributed (subject
 583 to smoothing) at each time-step. This simulation has a block of mesh moving in the
 584 approximate direction of the underlying advection, for which the activity in front of and
 585 behind the non-zero region is much reduced. On closer inspection, the circle of higher
 586 resolution mesh contains nodes moving at a similar velocity and in a similar direction to

587 each other, a rather different pattern to the actual advection velocity, $(-y, x)$. Therefore
 588 the method does not benefit from increased accuracy due to mesh nodes following the flow
 589 (something that can happen when methods of this type are applied in one dimension) and
 590 the error on the mesh shown on the right-hand side of Figure 3 is slightly higher: it also
 591 takes about 18 times as long to run.

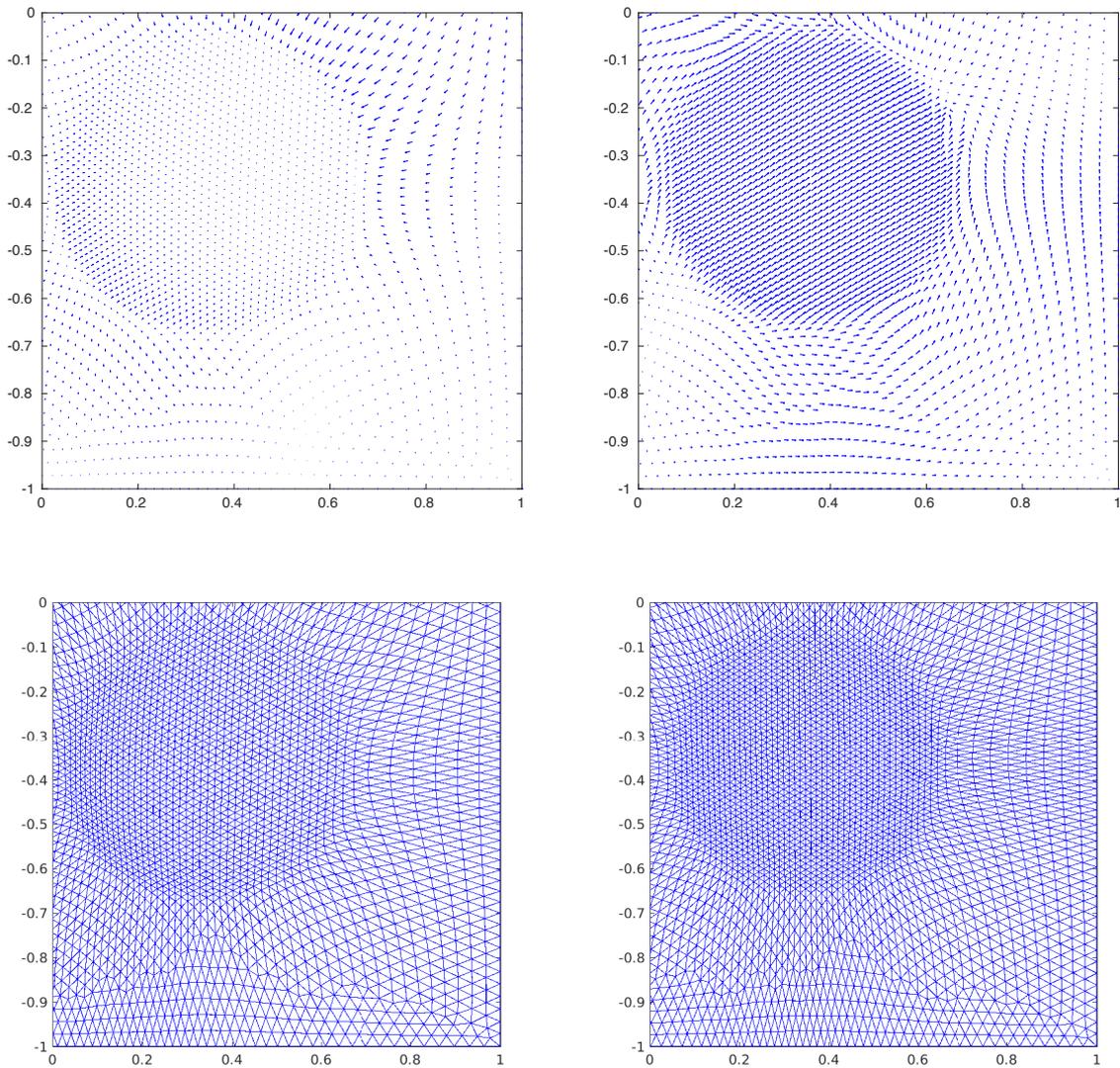


Figure 3: Comparison of node velocities (top) and meshes (bottom) in the bottom right quadrant of the computational domain at $t = \frac{\pi}{4}$ for circular advection, using the STB scheme with $N_{\text{move}} = 20$ (left) and $N_{\text{move}} = 500$ (right).

592 A single iteration of the STB scheme for a scalar equation requires little computational
 593 effort, so mesh movement iterations have a relatively significant effect on the overall
 594 computation time. It is worth noting though, that the mesh movement should tend to be
 595 more beneficial in 3D: second-order schemes, such as the ones considered here, typically

596 require 16 times as much work to reduce the error by a factor of 4 using uniform refinement
 597 in 3D, whereas it only requires 8 times as much work to reduce the error by the same
 598 amount in 2D. One might expect the mesh movement, which does not increase the size of
 599 the system to solve, to have similar cost-benefit ratios in both 2D and 3D, so the efficiency
 600 of the adaptive approach should scale much better in 3D relative to uniform refinement
 601 (assuming that the same error reduction can be achieved).

602 Figure 4 shows the effect of switching between the approximate and exact quadrature
 603 rules proposed for evaluating the flow sensors in (18) and (19). The exact quadrature
 604 (Simpson’s rule) is marginally the more efficient of the two strategies and is the one which
 605 has been used to obtain all of the other numerical results shown in this paper.

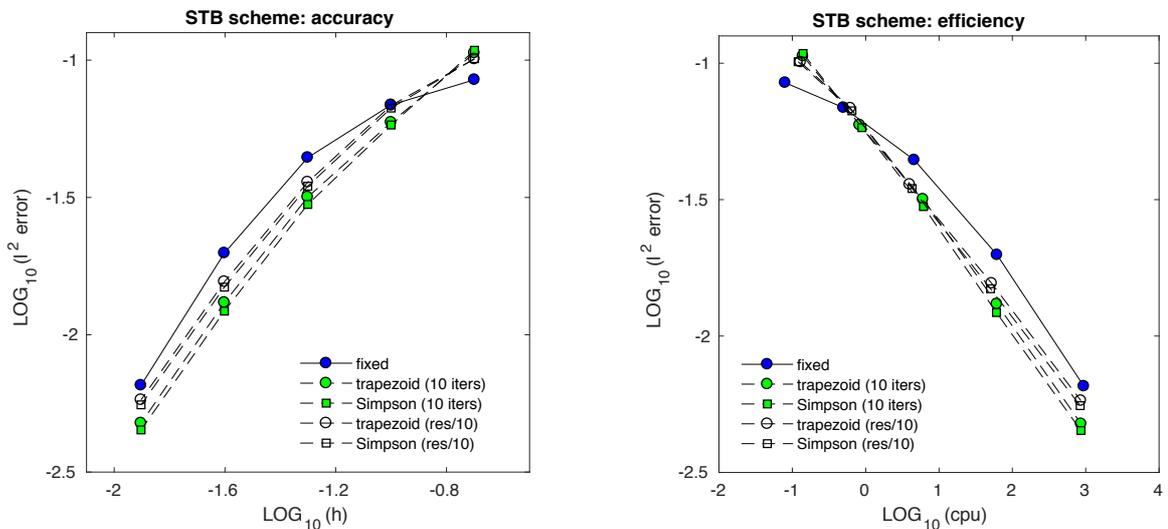


Figure 4: Comparison of accuracy and efficiency for circular advection, using the STB scheme on fixed meshes (blue circles, solid line) and adapted meshes (green or empty symbols, dashed line). Results are shown for when the trapezoid and Simpson rules are used to compute the flow sensors, Equations (19) and (18), respectively.

606 3.2. Shallow Water Flow

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

617 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 [53, 51]. Given a flat bottom

topography, the exact velocity field is expressed as $\mathbf{u}_\infty + \mathbf{u}'$, with

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

and \mathbf{u}_∞ 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

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

with

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

618 and $d_\infty = 1$.

619 For the mesh-convergence study, we set $\mathbf{u}_\infty = (6, 0)$, $\Gamma = 15$, $\omega = 4\pi$, $g = 9.80665$ and
 620 use a sequence of five unstructured triangulations of the domain $\Omega = [0, 2] \times [0, 1]$ with
 621 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
 622 of the vortex is at $(x_c, y_c) = (0.5, 0.5)$ and the time marching stops at $t = 1/6$, when
 623 $(x_c, y_c) = (1.5, 0.5)$. Freestream characteristic boundary conditions are used everywhere.

624 The accuracy and efficiency of the approximations obtained using the STB schemes
 625 are illustrated in Figure 5. Results are shown for fixed meshes and for different adaptive
 626 strategies for deciding when to terminate the mesh movement iterations within each time-
 627 step: after a fixed number of iterations, $N_{\text{move}} = 10, 30$ or 50 in this case; or after the
 628 space-time residual norm has dropped below a given tolerance, $\text{REL_TOL_MOVE} = 10^{-1}$.
 629 In all adaptive cases the initial mesh is created by applying $N_{\text{init}} = 500$ iterations of the
 630 mesh movement algorithm to the fixed mesh, the monitor function parameters are chosen
 631 to be $\alpha = 100$, $\beta = \gamma = 0.05$, and the relaxation parameter for the mesh smoothing was
 632 $\mu = 1.0$. For the time-stepping, $\text{CFL} = 4.0$, close to the optimal value found for efficiency
 633 on fixed meshes in [10], and $\text{CFL}_\tau = 0.3$.

634 In this case, the order of accuracy demonstrated on the finer meshes is slightly lower
 635 on the adapted meshes than on the fixed meshes. As a result, the mesh adaptation
 636 provides most benefit for meshes of intermediate resolution, on which improvements in
 637 efficiency of more than 60% can be obtained by moving the mesh. In this case, using
 638 30 mesh movement iterations per time-step gave the best efficiency – a higher number
 639 than for scalar advection, not only because the accuracy starts to deteriorate if more
 640 mesh movement iterations are used, but also because the computational time required for
 641 a single mesh movement iteration is lower relative to a single application of residual
 642 distribution for a nonlinear system. The total number of mesh movement iterations
 643 relative to the total number of pseudo-time-stepping iterations is shown in Table 2. The
 644 residual distribution step is about 5 times as expensive as in the scalar case, so the time
 645 required to move the mesh is relatively less significant. Moving the mesh also appears to
 646 accelerate the convergence of the pseudo-time-stepping iteration.

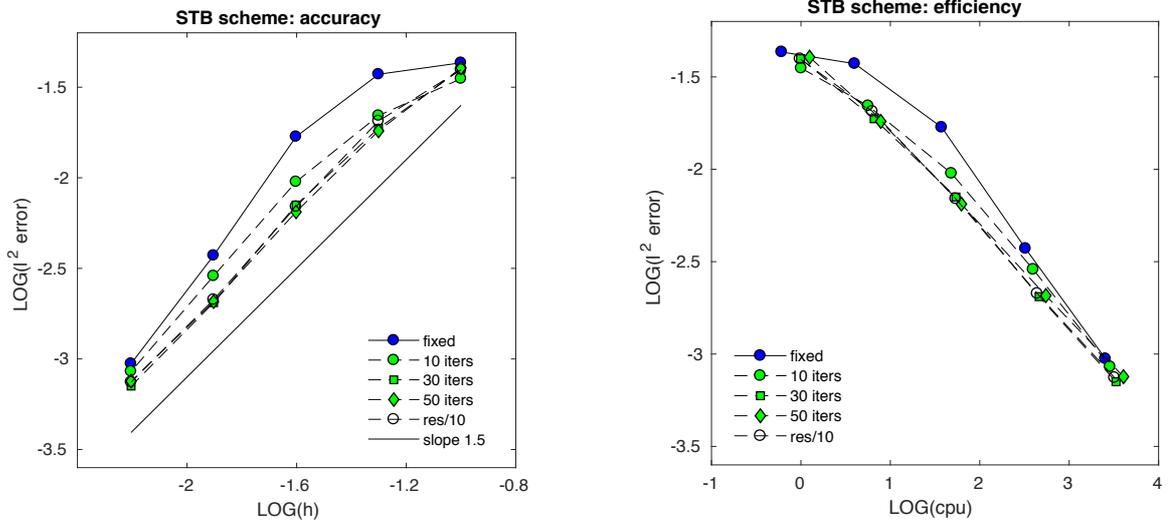


Figure 5: Comparison of accuracy and efficiency for the travelling vortex, using the STB scheme on fixed meshes (blue circles, solid line) and adapted meshes (green or empty symbols, dashed line).

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}
$N_{\text{move}} = 10$	91	8144	89.49	910	396.54	2.870×10^{-3}
$N_{\text{move}} = 30$	103	8851	85.93	3090	461.50	2.044×10^{-3}
$N_{\text{move}} = 50$	104	10036	96.50	5200	548.87	2.073×10^{-3}
$\text{REL_TOL_MOVE} = 10^{-1}$	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.

647 3.2.2. Small Perturbation of a Lake at Rest

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

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

652 and initial conditions,

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

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

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

657 are shown in Figure 8. In all adaptive cases the initial mesh is created by applying
658 $N_{\text{init}} = 1000$ iterations of the mesh movement algorithm to the fixed mesh, the monitor
659 function parameters are chosen to be $\alpha = 100$, $\beta = \gamma = 0.2$, and the relaxation parameter
660 for the mesh smoothing was $\mu = 0.7$. For the time-stepping, $\text{CFL} = 4.0$ and $\text{CFL}_r = 0.9$.

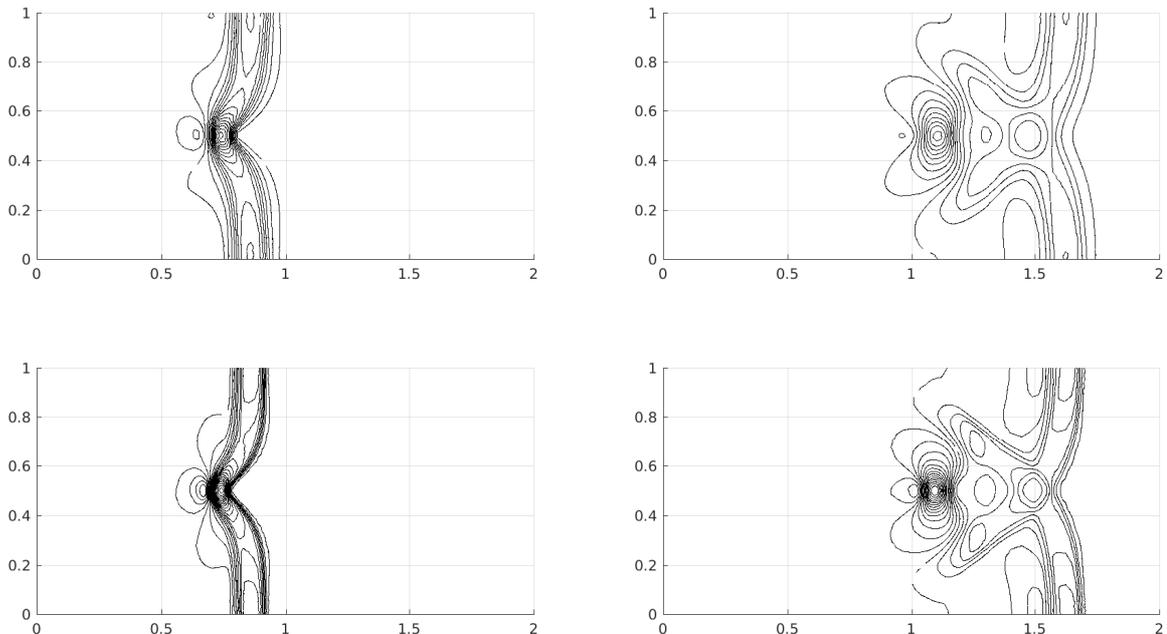


Figure 6: Comparison of isolines of free surface level η for the shallow water equations with a variable bed and a perturbation to the lake at rest equilibrium, using the STB scheme on a fixed mesh (top) and an adapted mesh with $N_{\text{move}} = 20$ (bottom): $t = 0.24$ (left), $t = 0.48$ (right). 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).

661 The mesh adaptation clearly resolves the features of the flow much more sharply. The
662 adapted solution is compared with approximations obtained on this fixed mesh and a finer
663 fixed mesh ($h \approx 0.01$, giving 23515 nodes and 69942 elements) in Figure 9. Two differ-
664 ent adaptive strategies are used, in which a fixed number of mesh movement iterations,
665 $N_{\text{move}} = 10$ or 20, are applied within each time-step. Adapting the mesh clearly improves
666 the approximation, giving estimates of local maxima and minima which are close to those
667 obtained on the fixed fine mesh for all regions except the small peak just downstream of
668 the main trough at $t = 0.48$. The improvements are greater when 20 mesh movement
669 iterations are used, but this is at the expense of computational time. The efficiency is illus-
670 trated in Table 3, which shows maximal and minimal values for different meshes alongside
671 the runtimes. These results include an intermediate mesh, for which $h \approx 0.01\sqrt{2}$ and the
672 runtime is similar to those for the adapted coarse mesh. Both adapted coarse meshes give
673 better maximal and minimal value approximations than the intermediate mesh for either
674 a similar ($N_{\text{move}} = 20$) or significantly lower ($N_{\text{move}} = 10$) runtime.

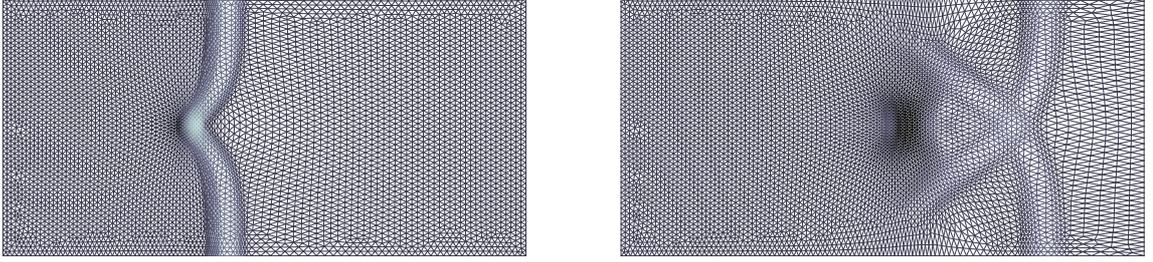


Figure 7: Adapted meshes with $N_{\text{move}} = 20$ for the shallow water equations with a variable bed and a perturbation to the lake at rest equilibrium, obtained using the STB scheme at times $t = 0.24$ (left), $t = 0.48$ (right).

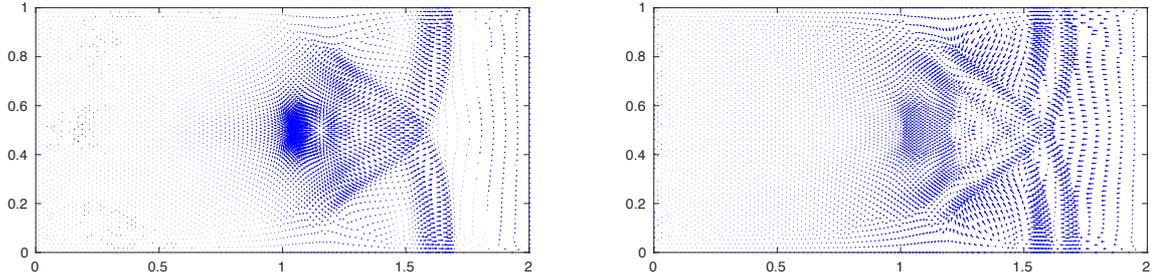


Figure 8: Fluid (left) and mesh (right) velocities obtained using the STB scheme on an adapted mesh with $N_{\text{move}} = 20$, for a perturbation to the lake at rest equilibrium, at time $t = 0.48$.

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.

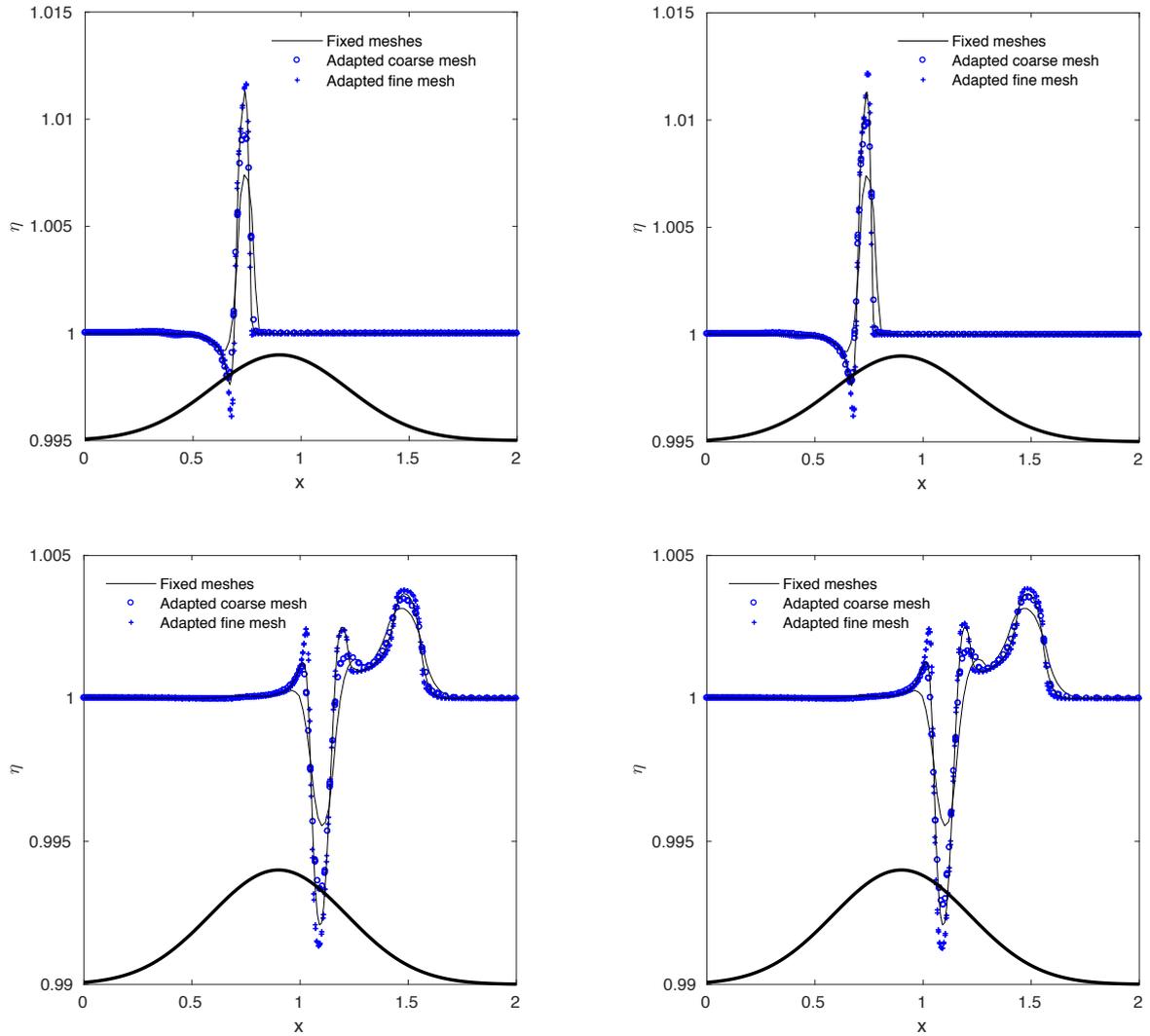


Figure 9: Comparison of profiles of free surface level η along $y = 0.5$ for the shallow water equations with a variable bed and a perturbation to the lake at rest equilibrium, using the STB scheme on coarse and fine meshes, both fixed and adapted, at times $t = 0.24$ (top) and $t = 0.48$ (bottom). Results 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.

675 *3.2.3. Circular Dam-Break over a Non-Smooth Bed*

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

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

680 and initial conditions,

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

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

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

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

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

709 **4. Concluding Remarks and Outlook**

710 In this paper we have described the generalisation of a space-time discontinuous resid-
 711 ual distribution (STDRD) scheme [10] to moving meshes. Within the STDRD framework

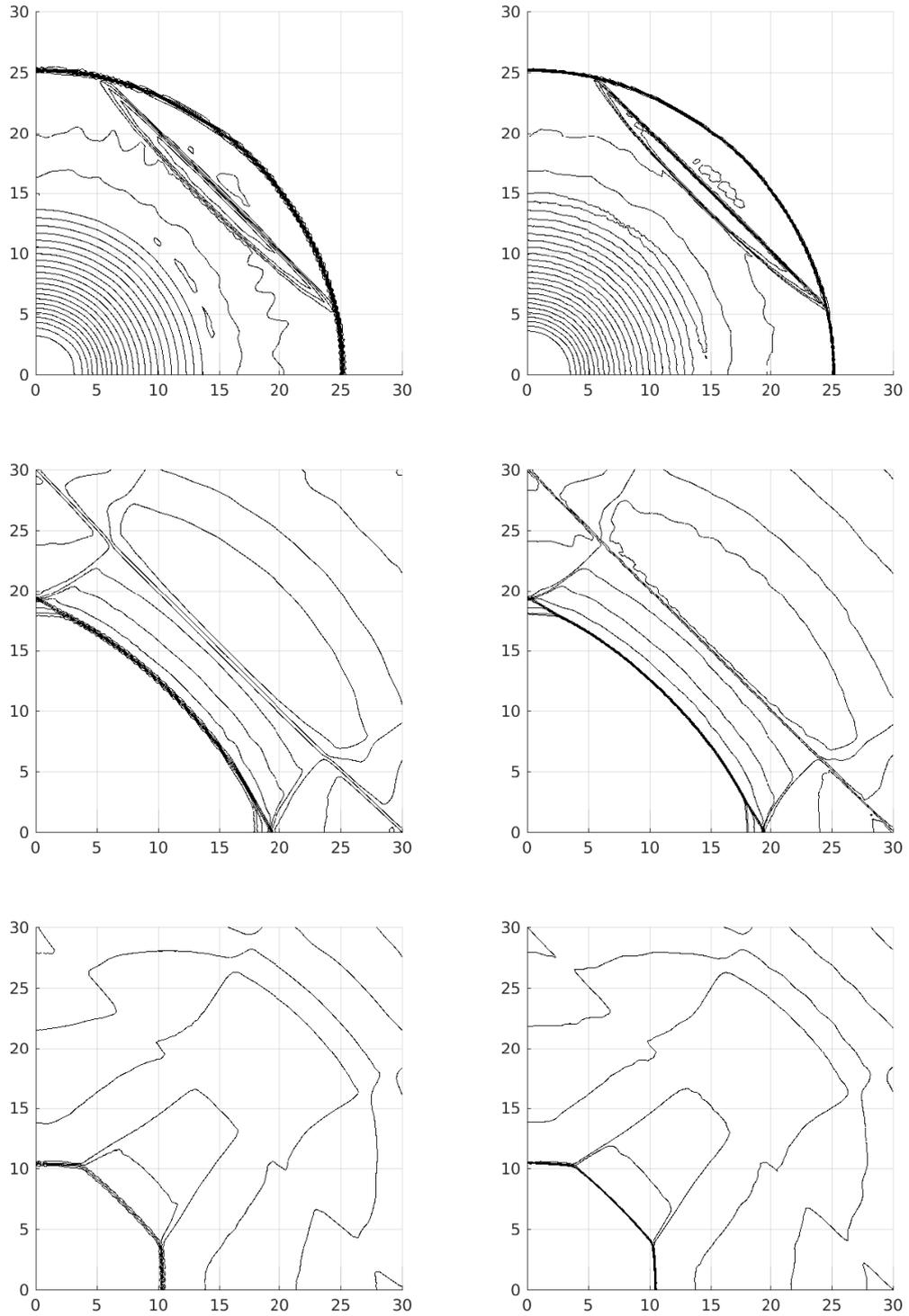


Figure 10: Comparison of isolines of free surface level η for the shallow water equations with a discontinuous bed, using the STB scheme on a fixed mesh (left) and an adapted mesh (right): $t = 3$ (top), $t = 9$ (middle), $t = 15$ (bottom). The interval between contours is 0.04 and the isolines shown are in the ranges $[0.32, 1.44]$ (top), $[0.2, 0.64]$ (middle), $[0.08, 0.4]$ (bottom).

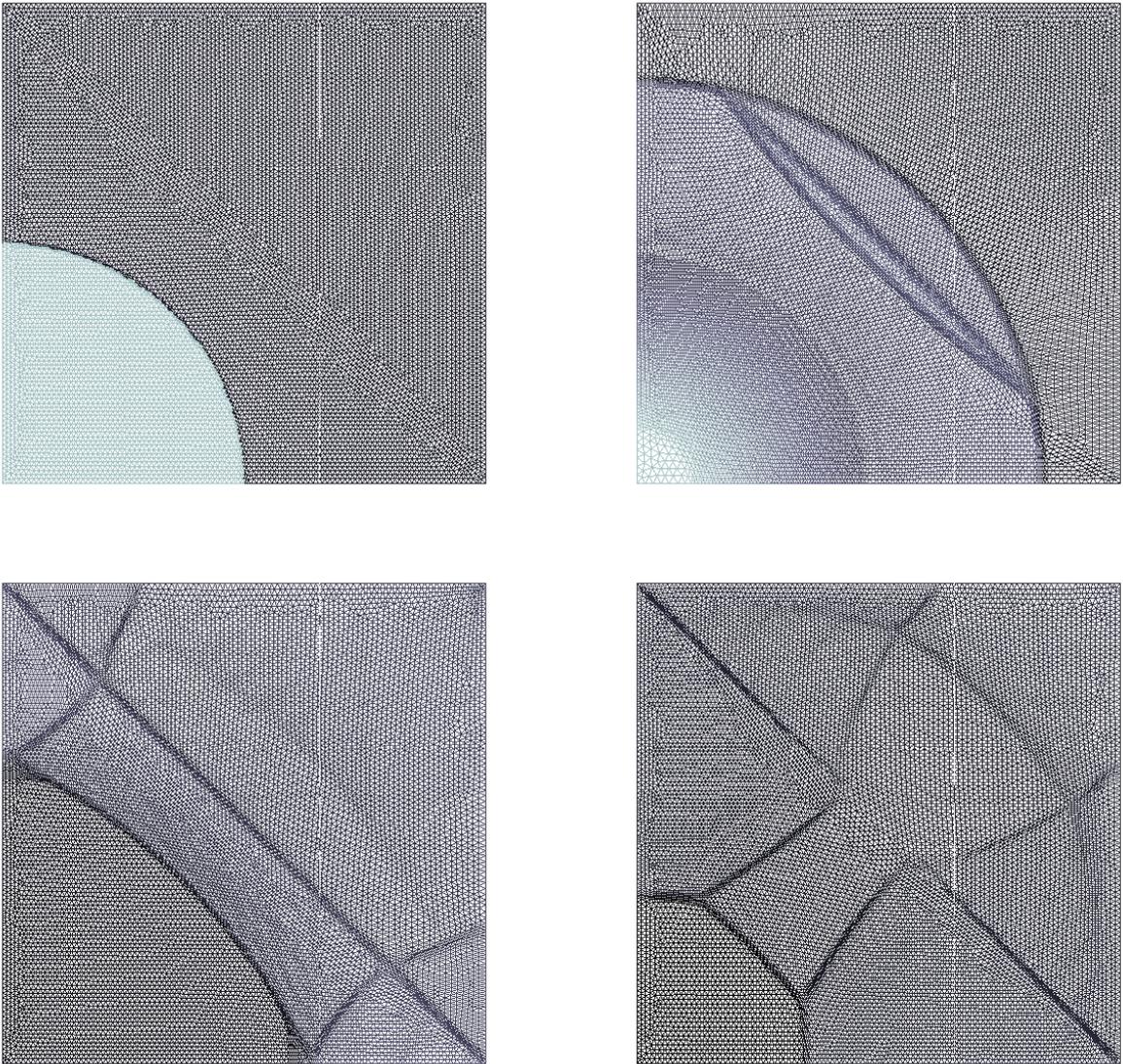


Figure 11: Adapted meshes for the shallow water equations with a discontinuous bed at times $t = 0$ (top left), $t = 3$ (top right), $t = 9$ (bottom left), $t = 15$ (bottom right). The mesh is shaded according to the local height of the free surface η .

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

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

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

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

- 740 • The mesh movement strategy is very simple. More sophisticated approaches, such as
741 the nonlinear iteration described in [11], might allow for a reduction in the number
742 of mesh movement iterations required relative to the number of pseudo-time-steps.
743 In particular, a much faster algorithm for equidistributing the initial mesh could be
744 found.
- 745 • The limit on the pseudo-time-step is inversely proportional to the size of the time-
746 step, so using a larger CFL (which is possible due to the unconditional stability of
747 the STDRD schemes) on the adapted mesh does not tend to decrease the overall
748 workload. A sophisticated nonlinear solver, for which the convergence rate was
749 independent of the time-step, would improve efficiency on both fixed and adapted
750 meshes.
- 751 • We would expect the mesh movement to have greater benefits for more complex
752 systems of nonlinear equations (for which the expense of residual distribution is
753 higher relative to the mesh movement) and in three space dimensions (where it
754 requires more work to reduce the error by the same amount using uniform mesh

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

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

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

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

769 5. Acknowledgements

770 This work was partially supported by the EPSRC [grant number EP/G003645/1].

771 **Appendix A. Proof of the C-property**

772 **Proposition**

773 *The space-time element residuals of the STDRD schemes described in this paper on moving*
 774 *meshes satisfy*

$$\Psi_E + \Phi_{E_t} + \Phi_{E_t}^b = 0 \quad (\text{A.1})$$

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

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

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

778 **Proof**

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

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

$$\begin{aligned} \Phi_{E_t} + \Phi_{E_t}^b &= \int_{E^{n+1}} \begin{bmatrix} d^{(n+1)^-} \\ \mathbf{0} \end{bmatrix} d\Omega - \int_{E^n} \begin{bmatrix} d^{n^+} \\ \mathbf{0} \end{bmatrix} d\Omega \\ &\quad + \int_{t^n}^{t^{n+1}} \int_{\partial E(t)} \begin{bmatrix} -d\mathbf{v} \cdot \hat{\mathbf{n}} \\ \frac{gd^2}{2} \hat{\mathbf{n}} \end{bmatrix} d\Gamma dt - \int_{t^n}^{t^{n+1}} \int_{E(t)} \begin{bmatrix} 0 \\ gd\nabla b \end{bmatrix} d\Omega dt, \end{aligned} \quad (\text{A.3})$$

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

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

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

788 so the residual for the mass balance equation in (A.3) can be written as

$$\int_{E^{n+1}} \eta^{(n+1)^-} d\Omega - \int_{E^n} \eta^{n^+} d\Omega - \int_{t^n}^{t^{n+1}} \int_{\partial E(t)} \eta\mathbf{v} \cdot \hat{\mathbf{n}} d\Gamma \quad (\text{A.5})$$

$$\approx |E^{n+1}| \bar{\eta}^{(n+1)-} - |E^n| \bar{\eta}^{n+} - \frac{\Delta t}{2} \left(\int_{\partial E(t^{n+1})} \eta^{(n+1)-} \mathbf{v} \cdot \hat{\mathbf{n}}^{n+1} \, d\Gamma + \int_{\partial E(t^n)} \eta^{n+} \mathbf{v} \cdot \hat{\mathbf{n}}^n \, d\Gamma \right),$$

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

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

792 It immediately follows that substituting $\eta^{(n+1)-} = \eta^{n+} = \text{const}$ in to (A.5) gives zero
 793 residual in the mass balance equation.

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

$$\int_{\partial E(t)} \frac{gd^2}{2} \hat{\mathbf{n}} \, d\Gamma = - \sum_{i \in E} \frac{g\bar{d}}{2} d_i \mathbf{n}_i. \quad (\text{A.7})$$

796 This is clearly gives zero total residual in the momentum balance equations when $\eta =$
 797 $d + b = \text{const}$ is substituted in to (A.3) and the source residual is approximated at any
 798 given time using

$$\int_{E(t)} gd \nabla b \, d\Omega = - \sum_{i \in E} \frac{g\bar{d}}{2} b_i \mathbf{n}_i, \quad (\text{A.8})$$

799 which is exact for piecewise linear d and b .

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

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

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

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

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

$$\begin{aligned} \Phi_{i,n}^N &= \mathcal{K}_{i,n}^+ \mathcal{N}_t \sum_{j \in E} \mathcal{K}_{j,n}^+ (V_i^{n+} - V_j^{n+}) \\ &\quad + \mathcal{K}_{i,n}^+ \mathcal{N}_t \sum_{j \in E} \mathcal{K}_{j,n+1}^+ (V_i^{n+} - V_j^{(n+1)-}) + \Phi_{i,n}^{LDA}, \end{aligned}$$

$$\begin{aligned}
\Phi_{i,n+1}^N &= \mathcal{K}_{i,n+1}^+ \mathcal{N}_t \sum_{j \in E} \mathcal{K}_{j,n}^+ \left(V_i^{n+1} - V_j^{n+1} \right) \\
&\quad + \mathcal{K}_{i,n+1}^+ \mathcal{N}_t \sum_{j \in E} \mathcal{K}_{j,n+1}^+ \left(V_i^{(n+1)^-} - V_j^{(n+1)^-} \right) + \Phi_{i,n+1}^{LDA}. \quad (\text{A.11})
\end{aligned}$$

808 These residuals are also clearly zero for $(V^{n+1})_0 = (V^{(n+1)^-})_0 = V^{n-} = [\text{const}, 0, 0]^T$, so
809 the STN scheme (and hence the STB scheme) also satisfies the C-property. \square

810 **Appendix B. Proof of the Discrete Maximum Principle**

811 **Proposition:**

812 *The space-time N scheme (24) on moving meshes verifies the discrete maximum principle,*

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

813 *for any time-step $\Delta t > 0$.*

814 **Proof**

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

816 First write out the space-time N scheme with discontinuities in time in full, by substituting
817 (24), (17) and (13) into (14), to give

$$\begin{aligned} & |S_i^n| (u_i^{n^+} - u_i^{n^-}) \\ & + \sum_{E \in D_i} \left[- \sum_{j \in E} k_{i,n}^+ N_t k_{j,n}^- (u_i^{n^+} - u_j^{n^+}) - \sum_{j \in E} k_{i,n}^+ N_t k_{j,n+1}^- (u_i^{n^+} - u_j^{(n+1)^-}) \right] = 0 \\ & \sum_{E \in D_i} \left[- \sum_{j \in E} k_{i,n+1}^+ N_t k_{j,n}^- (u_i^{(n+1)^-} - u_j^{n^+}) - \sum_{j \in E} k_{i,n+1}^+ N_t k_{j,n+1}^- (u_i^{(n+1)^-} - u_j^{(n+1)^-}) \right] = 0. \end{aligned} \quad (\text{B.2})$$

818 in which $D_i = \cup_{i \in E} E$, k^\pm and N_t are defined by (21) and (18) or (19), and $|S_i^n| =$
819 $\frac{1}{3} \sum_{E \in D_i} |E^n|$. This system of equations can be written in the form $AU^{new} = BU^{old}$:

$$\begin{bmatrix} A_{n^+,n^+} & A_{n^+,(n+1)^-} \\ A_{(n+1)^-,n^+} & A_{(n+1)^-, (n+1)^-} \end{bmatrix} \begin{bmatrix} U^{n^+} \\ U^{(n+1)^-} \end{bmatrix} = \begin{bmatrix} B_n^- \\ 0 \end{bmatrix} [U_{n^-}] \quad (\text{B.3})$$

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

$$\begin{aligned}
(A_{n^+,n^+})_{ii} &= |S_i^n| - \sum_{E \in D_i} \left[\sum_{j \in E, j \neq i} k_{i,n}^+ N_t k_{j,n}^- + \sum_{j \in E} k_{i,n}^+ N_t k_{j,n+1}^- \right] & \forall i \\
(A_{n^+,n^+})_{ij} &= \sum_{E \in D_i | j \in E} k_{i,n}^+ N_t k_{j,n}^- & \forall i, j | i \neq j \\
(A_{n^+, (n+1)^-})_{ij} &= \sum_{E \in D_i | j \in E} k_{i,n}^+ N_t k_{j,n+1}^- & \forall i, j \\
(A_{(n+1)^-, n^+})_{ij} &= \sum_{E \in D_i | j \in E} k_{i,n+1}^+ N_t k_{j,n}^- & \forall i, j \\
(A_{(n+1)^-, (n+1)^-})_{ij} &= \sum_{E \in D_i | j \in E} k_{i,n+1}^+ N_t k_{j,n+1}^- & \forall i, j | i \neq j \\
(A_{(n+1)^-, (n+1)^-})_{ii} &= - \sum_{E \in D_i} \left[\sum_{j \in E} k_{i,n+1}^+ N_t k_{j,n}^- + \sum_{j \in E, j \neq i} k_{i,n+1}^+ N_t k_{j,n+1}^- \right] & \forall i \\
(B_{n^-})_{ii} &= |S_i^n| & \forall i \\
(B_{n^-})_{ij} &= 0 & \forall i, j | i \neq j.
\end{aligned} \tag{B.4}$$

821 The majority of the entries in A will be zero because the set $\{E \in D_i | j \in E\}$ is empty
822 unless i and j are vertices of a common mesh element. The indices i and j both run from
823 1 to the total number of unknowns. First note that

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

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

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

825 Therefore, the matrix A is diagonally dominant for any time-step and any mesh. Hence,
826 because of the conditions in (B.5), A is an M-matrix, so A is invertible and A^{-1} is positive,
827 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.

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

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

831 This can be manipulated to give

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

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

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

834 so, from (B.8),

$$\begin{aligned} (B_n - U_{\min})_i &= |S_i^n| u_{\min}^{n^-} = \sum_{j \in D_i} (A_{n^+, n^+})_{ij} u_{\min}^{n^-} + \sum_{j \in D_i} (A_{n^+, (n+1)^-})_{ij} u_{\min}^{n^-} \\ &= A_{n^+, n^+} U_{\min} + A_{n^+, (n+1)^-} U_{\min}. \end{aligned} \quad (\text{B.10})$$

835 Now, (B.3) gives

$$A_{n^+, n^+} U^{n^+} + A_{n^+, (n+1)^-} U^{(n+1)^-} = B_n - U^{n^-}, \quad (\text{B.11})$$

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

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

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

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

838 from which it follows that

$$\begin{aligned} &(A_{(n+1)^-, (n+1)^-} U_{\min} + A_{(n+1)^-, n^+} U_{\min})_i \\ &= \sum_{j \in D_i} (A_{(n+1)^-, n^+})_{ij} u_{\min}^{n^-} + \sum_{j \in D_i} (A_{(n+1)^-, (n+1)^-})_{ij} u_{\min}^{n^-} = 0 \quad \forall i, \end{aligned} \quad (\text{B.14})$$

839 so, from (B.7) and (B.14),

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

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

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

841 in the sense that the inequality holds for each row of the system and, since A^{-1} exists
 842 and is positive, we can premultiply both sides by A^{-1} to get

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

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

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

844 A similar argument can be used to prove that

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

845

□

846 We note the following.

- 847 • This proof holds for any time-step and any meshes at the old and new time levels.
- 848 • This is a global condition, not a local condition. It therefore prohibits the creation
849 of new global extrema, but new local extrema may appear.
- 850 • A similar proof can be followed for the N scheme without the discontinuity in time
851 but now a consistency condition, $k_{i,n} \leq 0 \quad \forall i$ in every element, must be adhered to
852 (the past-shield condition). This places a restriction on the time-step.
- 853 • Note that, in the context of space-time residual distribution schemes, the CFL condi-
854 tion is defined in terms of the past-shield condition for the element residuals (which,
855 if satisfied, automatically implies that $u_i^{n+} = u_i^{n-} \quad \forall i$, even when the approximation
856 is allowed to be discontinuous in time), *i.e.* from Equation (19)

$$k_{i,n} \leq 0 \quad \Rightarrow \quad -\frac{\Delta t}{4} (\bar{\mathbf{a}}^n - \mathbf{v}_i) \cdot \mathbf{n}_i^n - \frac{|E^n|}{3} \leq 0. \quad (\text{B.20})$$

857 This imposes a limit on Δt which is different to the positivity condition for the
858 pseudo-time-stepping used to find a steady state scheme or to solve the space-time
859 system.

860 **Appendix C. Proof of the Consistency of Inflow Parameters for Residual Dis-**
861 **tribution Schemes**

862 **Proposition:**

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

$$\sum_{i \in E} k_{i,n} + \sum_{i \in E} k_{i,n+1} = 0. \quad (\text{C.1})$$

864 **Proof**

865 First note that it follows immediately from (19) that

$$\begin{aligned} \sum_{i \in E} k_{i,n} &= \left(\sum_{i \in E} \frac{\Delta t}{4} \mathbf{v}_i \cdot \mathbf{n}_i^n \right) - |E^n|, \\ \sum_{i \in E} k_{i,n+1} &= \left(\sum_{i \in E} \frac{\Delta t}{4} \mathbf{v}_i \cdot \mathbf{n}_i^{n+1} \right) + |E^{n+1}|, \end{aligned} \quad (\text{C.2})$$

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

$$\begin{aligned} \sum_{i \in E} ((\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{n}_k + (\mathbf{v}_k - \mathbf{v}_i) \cdot \mathbf{n}_j) &= \sum_{i \in E} (\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{n}_k + \sum_{i \in E} (\mathbf{v}_k - \mathbf{v}_i) \cdot \mathbf{n}_j \\ &= \sum_{i \in E} (\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{n}_k + \sum_{i \in E} (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{n}_k \\ &= 0 \end{aligned} \quad (\text{C.3})$$

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

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

870 Now, by definition,

$$\begin{aligned} |E^n| &= \frac{1}{2} (\mathbf{x}_j - \mathbf{x}_i) \wedge (\mathbf{x}_k - \mathbf{x}_i) \\ |E^{n+1}| &= \frac{1}{2} ((\mathbf{x}_j + \Delta t \mathbf{v}_j) - (\mathbf{x}_i + \Delta t \mathbf{v}_i)) \wedge ((\mathbf{x}_k + \Delta t \mathbf{v}_k) - (\mathbf{x}_i + \Delta t \mathbf{v}_i)), \end{aligned} \quad (\text{C.5})$$

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

872 anticlockwise, so

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

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

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

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

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

875

□

876 **References**

- 877 [1] R. Abgrall, Residual distribution schemes: current status and future trends, *Comput.*
878 *Fluids* 35 (7) (2006) 641–669.
- 879 [2] H. Deconinck, M. Ricchiuto, Residual distribution schemes: foundation and analysis,
880 in: *Encyclopedia of Computational Mechanics*, Vol. 3, Wiley, 2007.
- 881 [3] R. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in
882 Applied Mathematics, Cambridge University Press, Cambridge, 2002.
- 883 [4] R. H. Pletcher, J. C. Tannehill, D. A. Anderson, *Computational Fluid Mechanics*
884 *and Heat Transfer*, 3rd Edition, CRC Press, 2012.
- 885 [5] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd Edi-
886 tion, Springer-Verlag, Berlin, 2009.
- 887 [6] M. J. Baines, M. E. Hubbard, Multidimensional upwinding and grid adaptation,
888 in: *Numerical Methods for Wave Propagation*, Vol. 47 of *Fluid Mechanics and its*
889 *Applications*, Kluwer Academic Publishers, 1998, pp. 33–54.
- 890 [7] M. Ricchiuto, An explicit residual based approach for shallow water flows, *J. Comput.*
891 *Phys.* 80 (2015) 306–344.
- 892 [8] A. Warzyński, M. E. Hubbard, M. Ricchiuto, Runge-Kutta residual distribution
893 schemes, *J. Sci. Comput.* 62 (3) (2015) 772–802.
- 894 [9] Á. Csík, H. Deconinck, Space-time residual distribution schemes for hyperbolic con-
895 servation laws on unstructured linear finite elements, *Int. J. Numer. Meth. Fl.* 40 (3-4)
896 (2002) 573–581, ICFD Conference on Numerical Methods for Fluid Dynamics, Part
897 II (Oxford, 2001).
- 898 [10] D. Sármany, M. E. Hubbard, M. Ricchiuto, Unconditionally stable space-time dis-
899 continuous residual distribution for shallow-water flows, *J. Comput. Phys.* 253 (2013)
900 86–113.
- 901 [11] L. Arpaia, M. Ricchiuto, r -adaptation for shallow water flows: conservation, well-
902 balancedness, efficiency, *Comput. Fluids* 160 (2016) 175–203.
- 903 [12] J. Dobeš, H. Deconinck, An ALE formulation of the multidimensional residual dis-
904 tribution scheme for computations on moving meshes, in: Deconinck, H., Dick, E.
905 (eds) *Computational Fluid Dynamics 2006*, Springer, Berlin, Heidelberg, 2009.
- 906 [13] C. Michler, H. De Sterck, H. Deconinck, An arbitrary Lagrangian Eulerian formula-
907 tion for residual distribution schemes on moving grids, *Comput. Fluids* 32 (1) (2003)
908 59–71.
- 909 [14] H. D. Cenicerós, T. Y. Hou, An efficient dynamically adaptive mesh for potentially
910 singular solutions, *J. Comput. Phys.* 172 (2008) 609–639.

- 911 [15] T. Tang, Moving mesh methods for computational fluid dynamics, *Contemp. Math.*
912 383 (2005) 141–173.
- 913 [16] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and
914 the nonlinear stability of ALE schemes for the solution of flow problems on moving
915 grids, *J. Comput. Phys.* 174 (2) (2000) 669–694.
- 916 [17] M. J. Baines, M. E. Hubbard, P. K. Jimack, Velocity-based moving mesh methods
917 for nonlinear partial differential equations, *Commun. Comput. Phys.* 10 (3) (2011)
918 509–576.
- 919 [18] C. J. Budd, W. Huang, R. D. Russell, Adaptivity with moving grids, *Acta Numer.*
920 18 (2009) 111–241.
- 921 [19] L. Arpaia, M. Ricchiuto, A. R., An ALE formulation for explicit Runge-Kutta resid-
922 ual distribution, *J. Sci. Comput.* 63 (2015) 502–547.
- 923 [20] A. Bermudez, M. E. Vazquez-Cendon, Upwind methods for hyperbolic conservation
924 laws with source terms, *Comput. Fluids* 23 (8) (1994) 1049–1071.
- 925 [21] M. E. Hubbard, P. Garcia-Navarro, Flux difference splitting and the balancing of
926 source terms and flux gradients, *J. Comput. Phys.* 165 (1) (2000) 89–125.
- 927 [22] J. Murillo, A. Navas-Montilla, A comprehensive explanation and exercise of the
928 source terms in hyperbolic systems using Roe-type solutions. application to the 1d-2d
929 shallow water equations, *Adv. Water Resour.* 98 (2016) 70–96.
- 930 [23] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving
931 boundaries and deformable meshes, and their impact on aeroelastic computations,
932 *Comput. Method Appl. M.* 134 (1-2) (1996) 71–90.
- 933 [24] P. Wesseling, *Principles of Computational Fluid Dynamics*, Vol. 29 of Springer Series
934 in Computational Mathematics, Springer-Verlag, Berlin, 2001.
- 935 [25] R. Abgrall, A. Lerat, M. Ricchiuto, Construction of very high order residual distri-
936 bution schemes for steady inviscid flow problems on hybrid unstructured meshes, *J.*
937 *Comput. Phys.* 230 (11) (2011) 4103–4136.
- 938 [26] R. Abgrall, P. L. Roe, High order fluctuation schemes on triangular meshes, *J. Sci.*
939 *Comput.* 19 (1-3) (2003) 3–36.
- 940 [27] M. Ricchiuto, R. Abgrall, H. Deconinck, Construction of very high order residual
941 distribution schemes for unsteady advection: preliminary results, in: VKI LS 2003-
942 05, 33rd CFD Course, von Karman Institute for Fluid Dynamics, 2003.
- 943 [28] M. E. Hubbard, A. L. Laird, Achieving high-order fluctuation splitting schemes by
944 extending the stencil, *Comput. Fluids* 34 (2005) 443–459.
- 945 [29] A. R., High order schemes for hyperbolic problems using globally continuous approx-
946 imation and avoiding mass matrices, *J. Sci. Comput.* 73 (2017) 461–494.

- 947 [30] A. R., B. P., T. S., High-order residual distribution scheme for the time-dependent
948 Euler equations of fluid dynamics, *Comput. Math. Appl.* in press.
- 949 [31] N. Villedieu, High order discretisation by residual distribution schemes, Ph.D. thesis,
950 Universitat Libre de Bruxelles (2009).
- 951 [32] N. Villedieu, L. Koloszar, T. Quintino, H. Deconinck, Unsteady high order residual
952 distribution schemes with applications to linearised Euler equations, in: *Numerical
953 Mathematics and Advanced Applications*, Springer, 2009, pp. 911–919.
- 954 [33] M. E. Hubbard, M. Ricchiuto, Discontinuous upwind residual distribution: A route
955 to unconditional positivity and high order accuracy, *Comput. Fluids* 46 (1) (2011)
956 263–269.
- 957 [34] M. Ricchiuto, Contributions to the development of residual discretizations for hyper-
958 bolic conservation laws with application to shallow water flows, HDR Thesis.
- 959 [35] H. Deconinck, P. L. Roe, R. Struijs, A multidimensional generalization of Roe’s flux
960 difference splitter for the Euler equations, *Comput. Fluids* 22 (2-3) (1993) 215–222.
- 961 [36] Von Karman Institute of Fluid Dynamics, Lecture Series, High Order Discretization
962 Methods, 34th Computational Fluid Dynamics Course (November 2005).
- 963 [37] M. Ricchiuto, Á. Csík, H. Deconinck, Residual distribution for general time-
964 dependent conservation laws, *J. Comput. Phys.* 209 (1) (2005) 249–289.
- 965 [38] S. K. Godunov, A difference method for numerical calculation of discontinuous solu-
966 tions of the equations of hydrodynamics, *Mat. Sb. (N.S.)* 47 (89) (1959) 271–306.
- 967 [39] H. Deconinck, K. Sermeus, R. Abgrall, Status of multidimensional upwind residual
968 distribution schemes and applications in aeronautics, in: *AIAA Paper 2000-2328*,
969 AIAA, 2000.
- 970 [40] M. Ricchiuto, N. Villedieu, R. Abgrall, H. Deconinck, High order residual distribution
971 schemes: discontinuity capturing crosswind dissipation and extension to advection
972 diffusion, in: *VKI LS 06-01, 34th CFD Course*, von Karman Institute for Fluid
973 Dynamics, 2005.
- 974 [41] Á. Csík, M. Ricchiuto, H. Deconinck, A conservative formulation of the multidimen-
975 sional upwind residual distribution schemes for general nonlinear conservation laws,
976 *J. Comput. Phys.* 179 (1) (2002) 286–312.
- 977 [42] M. Ricchiuto, R. Abgrall, H. Deconinck, Application of conservative residual dis-
978 tribution schemes to the solution of the shallow water equations on unstructured
979 meshes, *J. Comput. Phys.* 222 (1) (2007) 287–331.
- 980 [43] A. Csík, M. Ricchiuto, H. Deconinck, Space time residual distribution schemes for
981 hyperbolic conservation laws over linear and bilinear elements, in: *VKI LS 2003-05*,
982 33rd CFD Course, von Karman Institute for Fluid Dynamics, 2003.

- 983 [44] R. Abgrall, M. Mezine, Construction of second order accurate monotone and stable
984 residual distribution schemes for unsteady flow problems, *J. Comput. Phys.* 188 (1)
985 (2003) 16–55.
- 986 [45] K. Sermeus, H. Deconinck, An entropy fix for multi-dimensional upwind residual
987 distribution schemes, *Comput. Fluids* 34 (4-5) (2005) 617–640.
- 988 [46] D. Sármany, M. E. Hubbard, Upwind residual distribution for shallow-water ocean
989 modelling, *Ocean Model.* 64 (1-2) (2013) 1–11.
- 990 [47] B. D. Rogers, A. Borthwick, P. Taylor, Mathematical balancing of flux gradient and
991 source terms prior to using Roe’s approximate Riemann solver, *J. Comput. Phys.*
992 192 (2003) 422–451.
- 993 [48] R. Abgrall, Toward the ultimate conservative scheme: following the quest, *J. Comput.*
994 *Phys.* 167 (2) (2001) 277–315.
- 995 [49] S. Rhebergen, Well-balanced r -adaptive and moving mesh space-time discontinuous
996 galerkin method for the shallow water equations, Oxford University Mathematical
997 Institute Technical Report 1757.
- 998 [50] M. Ricchiuto, On the C-property and generalized C-property of residual distribution
999 for the shallow water equations, *J. Sci. Comput.* 48 (1-3) (2011) 304–318.
- 1000 [51] M. Ricchiuto, A. Bollermann, Stabilized residual distribution for shallow water sim-
1001 ulations, *J. Comput. Phys.* 228 (4) (2009) 1071–1115.
- 1002 [52] M. E. Hubbard, Multidimensional upwinding and grid adaptation for conservation
1003 laws, Ph.D. thesis, Department of Mathematics, University of Reading (1996).
- 1004 [53] J. Dobeš, H. Deconinck, Second order blended multidimensional upwind residual
1005 distribution scheme for steady and unsteady computations, *J. Comput. Appl. Math.*
1006 215 (2) (2008) 378–389.
- 1007 [54] M. Seaïd, Non-oscillatory relaxation methods for the shallow-water equations in one
1008 and two space dimensions, *Int. J. Numer. Meth. Fl.* 46 (5) (2004) 457–484.