A class of residual distribution schemes and their relation to relaxation systems

James A. Rossmanith *

Department of Mathematics, University of Wisconsin, 480 Lincoln Drive, Madison, WI 53706-1388, USA

ARTICLE INFO

ABSTRACT

Residual distributions (RD) schemes are a class of high-resolution finite volume methods for unstructured grids. A key feature of these schemes is that they make use of genuinely multidimensional (approximate) Riemann solvers as opposed to the piecemeal 1D Riemann solvers usually employed by finite volume methods. In 1D, LeVeque and Pelanti [R.J. LeVeque, M. Pelanti, A class of approximate Riemann solvers and their relation to relaxation schemes, J. Comput. Phys. 172 (2001) 572] showed that many of the standard approximate Riemann solver methods (e.g., the Roe solver, HLL, Lax-Friedrichs) can be obtained from applying an exact Riemann solver to relaxation systems of the type introduced by Jin and Xin [S. Jin, Z.P. Xin, Relaxation schemes for systems of conservation-laws in arbitrary space dimensions, Commun. Pure Appl. Math. 48 (1995) 235]. In this work we extend LeVeque and Pelanti's results and obtain a multidimensional relaxation system from which multidimensional approximate Riemann solvers can be obtained. In particular, we show that with one choice of parameters the relaxation system yields the standard N-scheme. With another choice, the relaxation system yields a new Riemann solver, which can be viewed as a genuinely multidimensional extension of the local Lax-Friedrichs scheme. This new Riemann solver does not require the use Roe–Struijs–Deconinck averages, nor does it require the inversion of an $m \times m$ matrix in each computational grid cell, where m is the number of conserved variables. Once this new scheme is established, we apply it on a few standard cases for the 2D compressible Euler equations of gas dynamics. We show that through the use of linear-preserving limiters, the new approach produces numerical solutions that are comparable in accuracy to the N-scheme, despite being computationally less expensive.

1. Introduction

In the last few decades intense research into shock-capturing schemes has resulted in several numerical methods for solving partial differential equations (PDEs) that admit discontinuous weak solutions in the form of shock-waves. Examples of such schemes include WENO (weighted essentially non-oscillatory) [16], central [18], MUSCL (monotone upstream-centered schemes for conservation laws) [27], and wave propagation schemes [19]. One difficulty with these methods is that in general they do not trivially extend to problems in complex geometries. In order to handle application problems where complex geometry is of great importance, three broad classes of strategies have been considered: (1) Cartesian cut-cell methods [15],

* Tel.: +1 608 262 3852; fax: +1 608 263 8891.
  E-mail address: rossmani@math.wisc.edu

(2) overlapping meshes [9], and (3) unstructured grid methods. We will focus in this work on the last approach, thus eliminating the problem of cut-cells (1st approach) and interpolation between different grid patches (2nd approach), but requiring some efficient grid generation tool.

On unstructured grids, the two main classes of methods that have been developed are discontinuous Galerkin (DG) [10,22] and residual distribution (RD) [1,13] schemes. The discontinuous Galerkin approach is based on defining a piecewise polynomial approximation that is continuous inside element interiors, but discontinuous across element boundaries. Local 1D Riemann problems are solved across element boundaries to construct the necessary numerical fluxes. Residual distribution schemes can be viewed as a finite volume method where the finite volumes are defined by a grid that is dual to the original triangulation.

Although DG schemes have their own particular advantages, the focus of this work will be on RD schemes and, in particular, the aspect of RD schemes that separates them from all other methods: RD schemes are based on solving genuinely multi-dimensional Riemann problems. This aspect allows one to obtain methods that are positivity preserving for scalar conservation laws and essentially non-oscillatory for systems. This same feature, however, presents a challenge: how can these multi-dimensional Riemann problems be solved efficiently? The standard answer to this question is the so-called systems N-scheme [26] (see also [1,3]), which is a generalization of Roe's approximate Riemann solver for 1D systems [23]. One goal of this work is to develop an alternative to this approach.

LeVeque and Pelanti [21] showed how several of the standard approximate Riemann solvers can be interpreted as exact Riemann solvers for a perturbed system of hyperbolic equations known as *relaxation systems*. Their work was motivated by Jin and Xin's earlier paper [17] on a class of numerical methods known as *relaxation schemes*. What LeVeque and Pelanti essentially showed is that Jin and Xin's "new" class of methods could actually be thought of as a reinterpretation of various pre-existing approximate Riemann solvers; these results are reviewed in Section 2. After reviewing RD schemes in Section 3, we focus in this work on the continuation of LeVeque and Pelanti's reasoning and show how the N-scheme can be also be derived from a relaxation system. Furthermore, using this interpretation we derive a novel genuinely multi-dimensional Riemann solver that can be viewed as a multidimensional extension of the 1D local Lax-Friedrichs scheme [24]. Both of these results are presented in Section 4. Finally, we compare the numerical accuracy of the N-scheme and the newly derived scheme on several examples in Section 5. What we find is that when the appropriate limiters are applied, the novel scheme has comparable accuracy to the N-scheme, although it tends to be slightly more diffusive – this result is of course consistent with well-known 1D results comparing local Lax-Friedrichs versus Roe-type approximate Riemann solvers. On the other hand, this loss of accuracy is compensated by the fact that the new scheme is less computational expensive. This gain in computational efficiency will become significant for problems involving complicated equations such as the relativistic Euler or MHD equations.

2. Review of 1D relaxation systems

We briefly review in this section the results of LeVeque and Pelanti [21] for the case of 1D conservation laws. For simplicity we consider for the moment a scalar conservation laws of the form

$$q_{,t} + f(q)_{,x} = 0, \tag{1}$$

where $x \in \mathbb{R}$ is the spatial coordinate, $t \in \mathbb{R}^+$ is the time coordinate, $q \in \mathbb{R}$ is the conserved variable, and $f(q) : \mathbb{R} \to \mathbb{R}$ is the flux function. We assume that this conservation law is hyperbolic, meaning that $f'(q) \in \mathbb{R}$ for all q in the solution domain.

2.1. Finite volume methods in 1D

Using the idea of relaxation, we will construct in this section numerical methods for approximating (1). All of these methods are in the general class of finite volume methods [20], which we briefly recall in this subsection.

Let $\mathcal{T}_{\Delta x}$ be the numerical grid with grid cells centered at $x = x_i$ and spanning the interval $[x_i - \Delta x/2, x_i + \Delta x/2]$, where

$$x_i = a + (i - 1/2)\Delta x. \tag{2}$$

Here i is an integer ranging from 1 to N, a and b are the left and right end points of the domain, respectively, and $\Delta x = (b - a)/N$ is the grid spacing. In each grid cell x_i and at each time level $t = t^n$ we seek an approximation to the cell average of the exact solution $q(x,t)$:

$$Q_i^n \approx \frac{1}{\Delta x} \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} q(\xi, t^n)\, d\xi. \tag{3}$$

Integrating (1) over the grid cell centered at x_i and from $t = t^n$ to $t = t^{n+1}$ results in a numerical update formula for Q_i^n that can be written in the following *fluctuation splitting form*:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}[\mathcal{A}^- \Delta Q_{i+1/2}^n + \mathcal{A}^+ \Delta Q_{i-1/2}^n], \tag{4}$$

where $\mathcal{A}^- \Delta Q_{i+1/2}^n$ and $\mathcal{A}^+ \Delta Q_{i-1/2}^n$ are left- and right-going *fluctuations*, which measure the amount of flux that enters into grid cell x_i through the grid interfaces at $x = x_i + \Delta x/2$ and $x = x_i - \Delta x/2$, respectively. In order for this update to be numerically conservative these fluctuations must satisfy

$$\mathcal{A}^- \Delta Q_{i+1/2}^n + \mathcal{A}^+ \Delta Q_{i+1/2}^n = f(Q_{i+1}^n) - f(Q_i^n). \tag{5}$$

Note that in update (4) we collect the left-going fluctuation from the grid interface at $x_i + \Delta x/2$ and the right-going fluctuation from the grid interface at $x_i - \Delta x/2$, while in expression (5) we are adding the left- and right-going fluctuations at the same grid interface.

For first-order accurate methods, the fluctuations in update (4) are obtained by first assuming that the approximate solution has a constant value, Q_i^n, in each grid cell, and then solving at each grid interface, $x_{i-1/2} \equiv x_i - \Delta x/2$, the initial value problem for (1) with the piecewise constant initial data:

$$q(x, 0) = \begin{cases} Q_{i-1}^n & \text{if } x < x_{i-1/2}, \\ Q_i^n & \text{if } x < x_{i-1/2}. \end{cases} \tag{6}$$

This initial value problem is referred to as the *Riemann problem*. One of the pieces of information that can be obtained from solving the Riemann problem is how much of the initial flux difference, $f(Q_i^n) - f(Q_{i-1}^n)$, is carried to the left and how much to the right. It is precisely this information that is stored in the fluctuations, $\mathcal{A}^- \Delta Q$ and $\mathcal{A}^+ \Delta Q$.

2.2. Relaxation method framework in 1D

The relaxation schemes introduced by Jin and Xin [17] are based on the idea of approximating the quasilinear equation (1) by a linear system with a cleverly chosen source term. The role of this source term is to force the linear system to *relax* in the limit as $t \to \infty$ towards the original equation. By "hiding" the nonlinearity in the source term, relatively complicated quasilinear Riemann problems can be replaced by simpler linear Riemann problems.

There are many kinds of relaxation systems that one could develop in order to create an approximate solution to (1) (see pp. 26–48 of Bouchut [7] for a discussion of several different approaches). In this work we follow the approach of [21] and consider the following relaxation system:

$$\begin{bmatrix} q \\ \mu \end{bmatrix}_{,t} + \underbrace{\begin{bmatrix} 0 & 1 \\ -cd & c+d \end{bmatrix}}_{\text{coefficient matrix}} \begin{bmatrix} q \\ \mu \end{bmatrix}_{,x} = \frac{1}{\varepsilon} \underbrace{\begin{bmatrix} 0 \\ f(q) - \mu \end{bmatrix}}_{\text{source term}}, \tag{7}$$

where $c, d \in \mathbb{R}$ are parameters that will be adjusted in the next few subsections in order to arrive at various approximate Riemann solvers. Without loss of generality we will assume that $c \leqslant d$. The key observation is that by taking $\varepsilon \to 0$, the right-hand side forces $\mu \to f(q)$. Since the first equation in the above system is $q_{,t} + \mu_{,x} = 0$, $\mu \to f(q)$ will cause the relaxed system solution to approach the original conservation law solution.

In order to make this statement more precise, we will carry out a so-called *Chapman-Enskog* expansion, which in this case is simply a Taylor series expansion in ε applied to system (7). Omitting the algebra, this expansion to $\mathcal{O}(\varepsilon^2)$ yields the following equation for $q(x,t)$:

$$\underbrace{q_{,t} + f(q)_{,x}}_{\text{original cons. law}} = \varepsilon \underbrace{\left[\left(\frac{\partial f}{\partial q} - c \right) \left(d - \frac{\partial f}{\partial q} \right) q_{,x} \right]_{,x}}_{\text{diffusive correction}} + \mathcal{O}(\varepsilon^2). \tag{8}$$

This approximation is stable for $\varepsilon > 0$ if the values c and d are chosen to produce positive (or at least non-negative) diffusion; this occurs if

$$c \leqslant \frac{\partial f}{\partial q} \leqslant d. \tag{9}$$

The above statement is often referred as the *sub-characteristic condition* (see for example [8,17,21]), since it requires that the eigenvalues of the coefficient matrix, which are just c and d, enclose the characteristic speed of the original conservation law, $\partial f / \partial q$.

From relaxation system (7), LeVeque and Pelanti [21] showed that various classical approximate Riemann solvers could be derived. Following the philosophy of operator splitting (see pp. 380–390 of LeVeque [20] for a review), system (7) is first rewritten as two sub-problems:

$$\begin{bmatrix} q \\ \mu \end{bmatrix}_{,t} + \begin{bmatrix} 0 & 1 \\ -cd & c+d \end{bmatrix} \begin{bmatrix} q \\ \mu \end{bmatrix}_{,x} = 0, \tag{10}$$

$$\mu_{,t} = \frac{1}{\varepsilon} (f(q) - \mu). \tag{11}$$

Using this interpretation, LeVeque and Pelanti's [21] procedure for obtaining different approximate Riemann solvers can be summarized as follows:

(1) Choose values for the parameters c and d.
(2) Exactly solve the Riemann problem for the homogeneous linear system (10).
(3) Approximate the effect of equation (11) on the solution calculated in Step (2) by directly setting $\mu = f(q)$. In other words, instantaneously relax the solution from Step (2) to the $\varepsilon \to 0$ limit.

We will simply refer to this as the *relaxation procedure*. In the next four subsections, we will apply this strategy for various values of c and d. Each time we carry out step (2) of the above procedure we will exactly solve the initial value problem (i.e., $-\infty < x < \infty$) for system (10) using the generic *Riemann data*:

$$q(x,0) = \begin{cases} Q_\ell & \text{if } x < 0, \\ Q_r & \text{if } x > 0, \end{cases} \quad \text{and} \quad \mu(x,0) = \begin{cases} f(Q_\ell) & \text{if } x < 0, \\ f(Q_r) & \text{if } x > 0, \end{cases} \tag{12}$$

where Q_ℓ and Q_r are constants. Note that we are allowed to take $\mu = f(q)$ in the initial conditions at any arbitrary time step, since in the previous time-step we set $\mu = f(q)$ in Step (3) of the relaxation procedure.

2.3. Local Lax-Friedrichs (LLF) for scalar equations

The local Lax-Friedrichs or Rusanov method [24] is obtained by applying the relaxation procedure with the choice

$$c = -s \quad \text{and} \quad d = s, \tag{13}$$

where $s \geqslant |f'(q)|$ in order to satisfy the sub-characteristic condition. With this choice the Riemann solution is obtained by splitting the jump between the left and right states, $(Q_\ell, f(Q_\ell))$ and $(Q_r, f(Q_r))$, along the eigenvectors of the coefficient matrix:

$$\begin{bmatrix} Q_r - Q_\ell \\ f(Q_r) - f(Q_\ell) \end{bmatrix} = \alpha^1 \begin{bmatrix} 1 \\ -s \end{bmatrix} + \alpha^2 \begin{bmatrix} 1 \\ s \end{bmatrix}, \tag{14}$$

where the corresponding eigenvalues are $\lambda^1 = -s$ and $\lambda^2 = s$. From this expression we obtain the following fluctuations:

$$\mathcal{A}^- \Delta Q = \lambda^1 \alpha^1 = \frac{1}{2}(f(Q_r) - f(Q_\ell)) - \frac{s}{2}(Q_r - Q_\ell), \tag{15}$$

$$\mathcal{A}^+ \Delta Q = \lambda^2 \alpha^2 = \frac{1}{2}(f(Q_r) - f(Q_\ell)) + \frac{s}{2}(Q_r - Q_\ell). \tag{16}$$

2.4. Harten, Lax, and van Leer (HLL) for scalar equations

The HLL method of [14] is obtained by applying the above procedure with the choice

$$c = s_\ell \quad \text{and} \quad d = s_r, \tag{17}$$

where $s_\ell \leqslant f'(q) \leqslant s_r$ in order to satisfy the sub-characteristic condition. With this choice the Riemann solution is obtained by splitting the jump between the left and right states, $(Q_\ell, f(Q_\ell))$ and $(Q_r, f(Q_r))$, along the eigenvectors of the coefficient matrix:

$$\begin{bmatrix} Q_r - Q_\ell \\ f(Q_r) - f(Q_\ell) \end{bmatrix} = \alpha^1 \begin{bmatrix} 1 \\ s_\ell \end{bmatrix} + \alpha^2 \begin{bmatrix} 1 \\ s_r \end{bmatrix}, \tag{18}$$

where the corresponding eigenvalues are $\lambda^1 = s_\ell$ and $\lambda^2 = s_r$. From this expression we obtain the following fluctuations:

$$\mathcal{A}^\pm \Delta Q = s_\ell^\pm \alpha^1 + s_r^\pm \alpha^2 = \left(\frac{s_r^\pm - s_\ell^\pm}{s_r - s_\ell} \right)(f(Q_r) - f(Q_\ell)) - \left(\frac{s_r^\pm s_\ell - s_\ell^\pm s_r}{s_r - s_\ell} \right)(Q_r - Q_\ell). \tag{19}$$

In the above expressions we have made use of the following notation:

$$s^+ = \max(0,s) \quad \text{and} \quad s^- = \min(0,s). \tag{20}$$

We will make use of this notation throughout the remainder of this paper.

2.5. Roe's approximate Riemann solver for scalar equations

Roe's approximate Riemann solver [23] is obtained by applying the above procedure with the choice

$$c = d = s \equiv \frac{f(Q_r) - f(Q_\ell)}{Q_r - Q_\ell}. \tag{21}$$

With this choice the coefficient matrix becomes deficient, since only one linearly independent eigenvector exists. Therefore, the jump between the left and right states, $(Q_\ell, f(Q_\ell))$ and $(Q_r, f(Q_r))$, can be written as

$$\begin{bmatrix} Q_r - Q_\ell \\ f(Q_r) - f(Q_\ell) \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ s \end{bmatrix}, \tag{22}$$

where the corresponding eigenvalue is $\lambda = s$ (algebraic multiplicity 2, geometric multiplicity 1). Although this seems like an over-determined system for α, there exists a unique solution: $\alpha = Q_r - Q_\ell$. This results in the following fluctuations:

$$\mathcal{A}^\pm \Delta Q = s^\pm (Q_r - Q_\ell). \tag{23}$$

2.6. LLF and HLL for systems

Finally, we briefly explain how these three interpretations can be applied to a system of conservation laws of the form (1), now with $q \in \mathbb{R}^m$ and $f(q): \mathbb{R}^m \to \mathbb{R}^m$. We again assume hyperbolicity, which implies that the $m \times m$ matrix, $\partial f / \partial q$, has m real eigenvalues and m linearly independent eigenvectors for all q in the solution domain.

The systems LLF and HLL methods are obtained by considering the following relaxation system:

$$\begin{bmatrix} q \\ \mu \end{bmatrix}_{,t} + \begin{bmatrix} 0\mathbb{I} & \mathbb{I} \\ -cd\mathbb{I} & (c+d)\mathbb{I} \end{bmatrix} \begin{bmatrix} q \\ \mu \end{bmatrix}_{,x} = \frac{1}{\varepsilon} \begin{bmatrix} 0 \\ f(q) - \mu \end{bmatrix}, \tag{24}$$

where $\mathbb{I}$ is the $m \times m$ identity matrix, $0\mathbb{I}$ is the $m \times m$ matrix with zeros in every entry, $\mu \in \mathbb{R}^m$, and $c, d \in \mathbb{R}$.

The systems LLF method is obtained by taking

$$s = d = -c, \quad \text{where} \quad s \geqslant \max_{p=1,\dots,m} |\lambda^p|, \tag{25}$$

and λ^p is the pth eigenvalue of $\partial f / \partial q$. With this choice we again arrive at formula (15), which is now applied to each component of the solution vector.

Similarly, the systems HLL method is obtained by taking

$$s_\ell = c, \quad s_r = d, \quad \text{where} \quad s_\ell \leqslant \min_{p=1,\dots,m} (\lambda^p) \quad \text{and} \quad s_r \geqslant \max_{p=1,\dots,m} (\lambda^p). \tag{26}$$

With this choice we again arrive at formula (19), which is now applied to each component of the solution vector.

2.7. Roe's approximate Riemann solver for systems

Roe's approximate Riemann solver does not follow from working with relaxation system (24), but instead from

$$\begin{bmatrix} q \\ \mu \end{bmatrix}_{,t} + \begin{bmatrix} 0\mathbb{I} & \mathbb{I} \\ -(\hat{J})^2 & 2\hat{J} \end{bmatrix} \begin{bmatrix} q \\ \mu \end{bmatrix}_{,x} = \frac{1}{\varepsilon} \begin{bmatrix} 0 \\ f(q) - \mu \end{bmatrix}. \tag{27}$$

In the above expression,

$$\hat{J} = \frac{\partial f}{\partial q}(\widehat{Q}), \tag{28}$$

where $\widehat{Q}$ is the Roe average [23] and satisfies

$$\frac{\partial f}{\partial q}(\widehat{Q})(Q_r - Q_\ell) = f(Q_r) - f(Q_\ell). \tag{29}$$

With this choice the Riemann solution is obtained by splitting the jump between the left and right states, $(Q_\ell, f(Q_\ell))$ and $(Q_r, f(Q_r))$, along the m distinct eigenvectors of $\hat{J}$ in the following manner:

$$\begin{bmatrix} Q_r - Q_\ell \\ f(Q_r) - f(Q_\ell) \end{bmatrix} = \alpha^1 \begin{bmatrix} r^1 \\ s^1 r^1 \end{bmatrix} + \cdots + \alpha^m \begin{bmatrix} r^m \\ s^m r^m \end{bmatrix}, \tag{30}$$

where s^p and r^p are the pth eigenvalue and right eigenvector of the Roe matrix $\hat{J}$, respectively. Just as in the scalar case, it seems as though the parameters α are overdetermined. However, since $\hat{J}$ satisfies the constraint (29), it can easily be shown that the first set of m equations involving $Q_r - Q_\ell$ are identical to the second set of m equations involving $f(Q_r) - f(Q_\ell)$. In other words, there are only m distinct equations for m values of α; and therefore, a unique solution exists:

$$\alpha^p = \ell^p \cdot (Q_r - Q_\ell), \quad \text{for} \quad p = 1, \dots, m, \tag{31}$$

where ℓ^p is the pth left eigenvector of $\hat{J}$. This results in the following fluctuations:

$$\mathcal{A}^{\pm}\Delta Q = \sum_{p=1}^{m}[s^p]^{\pm}\{\ell^p \cdot (Q_r - Q_{\ell})\}r^p. \tag{32}$$

Note that conservation follows from (29).

3. Residual distribution schemes

We will describe in this section the basic residual distribution method for solving hyperbolic conservation laws in multidimensions. For further details we refer the reader to articles by Abgrall [1,2] and Abgrall and Mezine [4,5]. We consider a conservation law of the form

$$q_{,t} + \nabla \cdot \vec{f}(q) = q_{,t} + f(q)_x + g(q)_y = 0, \tag{33}$$

where $(x,y) \in \mathbb{R}^2$ are the spatial coordinates, $t \in \mathbb{R}$ is the time coordinate, $q \in \mathbb{R}^m$ is the vector of conserved variable, and $\vec{f}(q) : \mathbb{R}^m \to \mathbb{R}^{m \times 2}$ is the flux function. We will assume that this equation is hyperbolic, meaning that the $m \times m$ flux Jacobian matrix,

$$J(\vec{n}) \equiv \vec{n} \cdot \frac{\partial \vec{f}(q)}{\partial q}, \tag{34}$$

is diagonalizable with real eigenvalues for all $\vec{n} \in \mathbb{R}^2$ such that $\|\vec{n}\| = 1$ and for all q in the solution domain.

The first step in approximately solving (33) in some domain $\Omega \subset \mathbb{R}^2$ is to mesh the domain with a finite number of triangles. We will refer to this triangulation as $\mathcal{T}_h$, where h refers to a representative triangle radius, which in this work we just take to be the square root of the triangle area: $h \equiv \sqrt{|\mathcal{T}|}$. Associated with this triangulation is a *dual grid*, which is constructed by connecting triangle centers to edge centers. A an example triangulation along with its dual grid is shown in Fig. 1.

Unlike the discontinuous Galerkin approach [10,22], approximate solutions are centered on triangle nodes (i.e., centers of the dual grid) rather than triangle centers. In order to obtain an update for these node centered values, we integrate (33) over the median dual cell C_i and from $t = t^n$ to $t = t^{n+1}$:

$$\iint_{C_i} q(\vec{x},t^{n+1})\,\mathrm{d}\vec{x} = \iint_{C_i} q(\vec{x},t^n)\,\mathrm{d}\vec{x} - \int_{t^n}^{t^{n+1}}\iint_{C_i} \nabla \cdot \vec{f}(q)\,\mathrm{d}\vec{x}\mathrm{d}t = \iint_{C_i} q(\vec{x},t^n)\,\mathrm{d}\vec{x} - \sum_{\mathcal{T}:i\in\mathcal{T}}\int_{t^n}^{t^{n+1}}\oint_{\partial(C_i\cap\mathcal{T})} \vec{f}(q)\cdot\mathrm{d}\vec{s}\mathrm{d}t.$$

Next we define the median dual cell average and the time-averaged fluctuations through $\partial(C_i \cap \mathcal{T})$:

$$Q_i^n \equiv \frac{1}{|C_i|}\iint_{C_i} q(\vec{x},t^n)\,\mathrm{d}\vec{x}, \tag{35}$$

$$\Phi_i^{\mathcal{T}} \equiv \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\oint_{\partial(C_i\cap\mathcal{T})} \vec{f}(q)\cdot\mathrm{d}\vec{s}\mathrm{d}t. \tag{36}$$

Using these definitions, the update formula for a generic residual distribution scheme can be written as follows:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{|C_i|}\sum_{\mathcal{T}:i\in\mathcal{T}} \Phi_i^{\mathcal{T}}. \tag{37}$$

○ Triangle center   - - - Grid lines of median dual cells
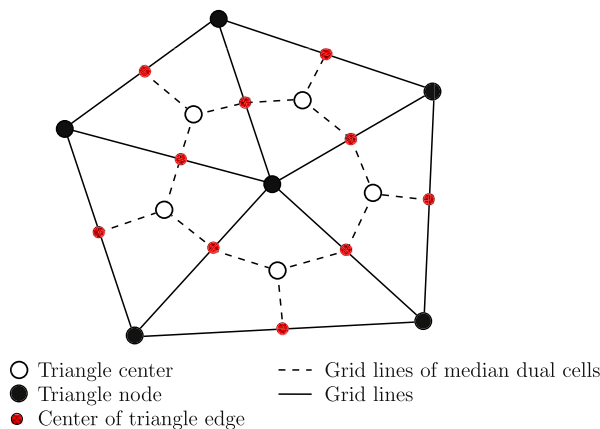● Triangle node   —— Grid lines
● Center of triangle edge

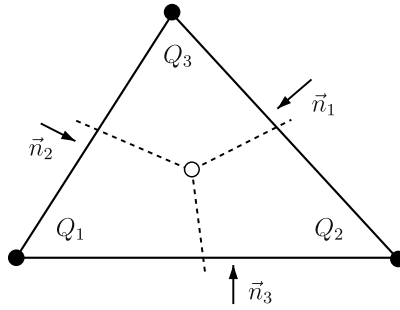**Fig. 1.** Sample triangulation and dual grid.

Fig. 2. A depiction of the multidimensional Riemann problem that must be solved in each triangle. The numerical solution is piecewise constant on each median dual cell. For example, the approximate solution on the three dual cells that overlap the triangle shown in this figure are Q_1, Q_2, and Q_3. Note that the area of each of the three sections is the same, the midpoint where the dashed lines meet is $(\vec{x}_1 + \vec{x}_2 + \vec{x}_3)/3$, $\vec{n}_k$ are the inward-pointing normal vectors to each edge, and the magnitude of $\vec{n}_k$ is equal to the length of the edge to which it is orthogonal.

In the remainder of this work, we will approximate the exact solution, $q(\vec{x}, t)$, with a piecewise constant representation, Q_i^n, that is constant on each median dual cell. We note that this view of RD schemes is slightly different than the standard view (e.g. [1]), where the approximate solution is usually viewed to be piecewise linear on each triangle $\mathcal{T}$. Although these descriptions seem contradictory, in the case for first-order accuracy in time, both interpretations yield the same numerical schemes. The advantage of viewing the solution as being piecewise constant on each medial dual cell is that this naturally sets up a series of multidimensional Riemann problems in each triangle (see Fig. 2), which can be solved to construct the fluctuations[1] $\Phi_i^{\mathcal{T}}$. In this way, we can then view approximate constructions of $\Phi_i^{\mathcal{T}}$ as approximate Riemann solvers.

Computing the fluctuations $\Phi_i^{\mathcal{T}}$ is generally done using the following framework (again, the two interpretations, piecewise constant on each dual cell vs. piecewise linear on each primal cell, make use of the exactly the same framework):

(1) On each triangle $\mathcal{T}$ construct a total residual:

$$\Phi^{\mathcal{T}} = \iint_{\mathcal{T}} \nabla \cdot \vec{f}^h \, d\vec{x} = \oint_{\partial \mathcal{T}} \vec{f}^h \cdot d\vec{s}, \tag{38}$$

where $\vec{f}^h$ denotes an interpolant that passes through the three nodal values

$(\vec{x}_i, \vec{f}(Q_i))$ for $i = 1, 2, 3$.

For example if we simply use linear interpolation, the total residual can be written as[2]

$$\Phi^{\mathcal{T}} = \frac{1}{2} \sum_{i=1}^{3} \vec{f}(Q_i) \cdot \vec{n}_i. \tag{39}$$

Here $\vec{n}_i$ represents the inward pointing normal vectors to the three edges of the triangle $\mathcal{T}$. The length of $\vec{n}_i$ is equal to the length of the edge to which it is perpendicular. If the three nodes of triangle $\mathcal{T}$ are given by (x_i, y_i) for $i = 1, 2, 3$, then the three scaled normals can be written as

$$\vec{n}_1 = (y_2 - y_3, x_3 - x_2)^t,$$
$$\vec{n}_2 = (y_3 - y_1, x_1 - x_3)^t,$$
$$\vec{n}_3 = (y_1 - y_2, x_2 - x_1)^t.$$

(2) Once this total residual has been calculated, it is then distributed to each of the nodes of the triangle:

$$\Phi^{\mathcal{T}} \quad \rightarrow \quad \Phi_1^{\mathcal{T}}, \Phi_2^{\mathcal{T}}, \Phi_3^{\mathcal{T}}.$$

The detailed strategy for how this distribution is accomplished yields a specific numerical method.

3.1. Design principles for scalar conservation laws

We first focus on design principles for scalar equations; in a subsequent subsection we explain how to extend this to the systems case. In order to obtain a numerical update that produces a stable and accurate approximation to (33), we will need the distribution strategy to satisfy certain properties:

---

[1] In this work, the terms *distributed residual* and *fluctuation* mean the same thing and are used interchangeably.

[2] For the standard N-scheme, which we will describe shortly, this is not the definition of the total residual that is used.

(1) *Numerical conservation:* Since the interpolation of the numerical solution is continuous across triangle edges, conservation simply reduces to the following constraint:

$$\sum_{i=1}^{3} \Phi_i^{\mathcal{T}} = \Phi^{\mathcal{T}}. \tag{40}$$

In other words, in a given triangle, the sum of the distributed residuals must equal the total residual.

(2) *Monotonicity preserving:* This condition makes sure that the numerical update satisfies a local maximum principle, which is needed to guarantee that the update does not generate any new spurious maxima or minima. If we write

$$\Phi_i^{\mathcal{T}} = \sum_{j=1}^{3} c_{ij}^{\mathcal{T}} (Q_i^n - Q_j^n), \tag{41}$$

then the monotonicity requirement can be written as (see [13]):

$$c_{ij}^{\mathcal{T}} \geqslant 0 \quad \forall i, j \in (1, 2, 3). \tag{42}$$

(3) *Linear preserving*: The order of accuracy of update (37) in the steady-state depends, among other things, on how accurately the total residual (38) is calculated on each triangle [6]. If we use formula (39), then $\Phi^{\mathcal{T}} = \mathcal{O}(h^3)$ in the steady-state, which is the correct order of accuracy if want an approximate solution, Q_i, that is $\mathcal{O}(h^2)$ accurate in the steady-state. What we actually need in order to get an $\mathcal{O}(h^2)$ accurate steady-state solution is that not only that $\Phi^{\mathcal{T}} = \mathcal{O}(h^3)$, but that each distributed residual also satisfies $\Phi_i^{\mathcal{T}} = \mathcal{O}(h^3)$. The distributed residuals can be written as

$$\Phi_1^{\mathcal{T}} = \beta_1 \Phi^{\mathcal{T}}, \quad \Phi_2^{\mathcal{T}} = \beta_2 \Phi^{\mathcal{T}}, \quad \Phi_3^{\mathcal{T}} = \beta_3 \Phi^{\mathcal{T}}, \tag{43}$$

where β_i measures the fraction of the total residual that is distributed to node i. To ensure that the distributed residuals are of the same order as the total residual, we need to make sure that the β_i's remain bounded as $h \to 0$. Therefore, the $\mathcal{O}(h^2)$ accuracy condition, or more commonly referred to as the linear preserving condition, can be writtten as follows (see [13]):

$$\beta_i \quad \text{for } i = 1, 2, 3, \text{ is uniformly bounded independent of the mesh}. \tag{44}$$

3.2. Scalar N-scheme

Modern finite volume methods for hyperbolic PDEs are typically based on solving, either exactly or approximately, a Riemann problem between neighboring states. For multidimensional problems, a standard approach is to solve local 1D Riemann problems and then use the information from the Riemann solutions to construct numerical fluxes or fluctuations (see Chapters 19–21 of LeVeque [20]).

In the RD framework, however, a multidimensional Riemann problem is solved. In an arbitrary triangle $\mathcal{T}$, we consider the Riemann problem between three constant states: Q_1, Q_2, and Q_3 (see Fig. 2 for a depiction). Exact solutions to multidimensional Riemann problems are at best expensive to evaluate, and in general not well-understood for many hyperbolic systems such as the Euler equations from gas dynamics [25]. Therefore, in practice an approximate method such as the N-scheme (the "N" stands for Narrow) [12,13] is utilized; this approach can be viewed as a multidimensional generalization of Roe's approximate Riemann solver [23].

Just as in the 1D case, we define a Roe-average (henceforth called the Roe–Struijs–Deconinck average [12]):

$$\vec{u}^{\mathcal{T}} \equiv \frac{\partial \vec{f}}{\partial q}(\widehat{Q}), \tag{45}$$

where $\widehat{Q}$ is an average of the three nodal values Q_i for $i = 1, 2, 3$ on the current triangle $\mathcal{T}$. The Roe–Struijs–Deconinck average satisfies the following constraint, which generalizes the 1D constraint given by (29):

$$\Phi^{\mathcal{T}} \equiv \frac{1}{2} \sum_{i=1}^{3} (\vec{u}^{\mathcal{T}} \cdot \vec{n}_i) Q_i = \oint_{\partial \mathcal{T}} \vec{f}(q^h) \cdot d\vec{s}, \tag{46}$$

where q^h is the linear interpolant passing through $(\vec{x}_i, Q_i)$ for $i = 1, 2, 3$. If the flux, $f(q)$, is at most a quadratic function of q, then

$$\vec{u}^{\mathcal{T}} = \frac{1}{3}(\vec{f}'(Q_1) + \vec{f}'(Q_2) + \vec{f}'(Q_3)). \tag{47}$$

The approximate Riemann solution gives rise to the following set of fluctuations:

$$\text{N-scheme}: \quad \Phi_i^{\mathcal{T}} = \frac{1}{2}[\vec{u}^{\mathcal{T}} \cdot \vec{n}_i]^+ (Q_i - Q_\star), \tag{48}$$

where $Q_\star$ is the so-called *upwind parameter*. In 1D the the upwind parameter relative to state Q_i is always either Q_{i-1} if $u > 0$ or Q_{i+1} if $u < 0$. In multidimensions, $Q_\star$ is obtained by enforcing the conservation constraint (40):

$$Q_\star = \left(\sum_{i=1}^{3} [\vec{u}^T \cdot \vec{n}_i]^- Q_i \right) \Bigg/ \left(\sum_{j=1}^{3} [\vec{u}^T \cdot \vec{n}_j]^- \right), \tag{49}$$

where we have made use of the following two identities:

$$\sum_{i=1}^{3} (\vec{u}^T \cdot \vec{n}_i) Q_i = \sum_{i=1}^{3} [\vec{u}^T \cdot \vec{n}_i]^+ Q_i + \sum_{i=1}^{3} [\vec{u}^T \cdot \vec{n}_i]^- Q_i,$$

$$\sum_{i=1}^{3} (\vec{u}^T \cdot \vec{n}_i) = \sum_{i=1}^{3} [\vec{u}^T \cdot \vec{n}_i]^+ + \sum_{i=1}^{3} [\vec{u}^T \cdot \vec{n}_i]^- = 0.$$

In order to demonstrate that the N-scheme is monotonicity preserving, we rewrite (48) in the form (41) with

$$c_{ij}^T = \frac{[\vec{u}^T \cdot \vec{n}_i]^+ [\vec{u}^T \cdot \vec{n}_j]^-}{\sum_k [\vec{u}^T \cdot \vec{n}_k]^-} \geqslant 0. \tag{50}$$

From this it can be shown that update (37) is monotone under the following CFL condition:

$$\Delta t \leqslant \min_i \left\{ \frac{2|C_i|}{\sum_{T : i \in T} [\vec{u}^T \cdot \vec{n}_i]^+} \right\}. \tag{51}$$

3.3. Linear preserving limiters

The N-scheme described so far is both conservative and monotonicity preserving; however, it is not yet linear preserving. The problem with the previously described N-scheme is that the weights (43) are not uniformly bounded independent of the mesh. In order to modify the N-scheme to achieve uniformly bounded β_i's, Abgrall and Mezine [5] introduced a nonlinear limiting procedure. The limiting process takes the original β_i and replaces them with limited versions, denoted $\tilde{\beta}_i$. The simpler of the two approaches discussed in [5] yields the following formulas:

$$\beta_i = \frac{\Phi_i^T}{\Phi^T} \quad \rightarrow \quad \tilde{\beta}_i = \frac{[\beta_i]^+}{\sum_j [\beta_j]^+}, \tag{52}$$

which guarantees that $0 \leqslant \tilde{\beta}_i \leqslant 1$. The limited residuals are then given by

$$\text{Limited N-scheme}: \quad \Phi_i^T = \tilde{\beta}_i \Phi^T. \tag{53}$$

It is clear that this scheme is both linear preserving and conservative. Furthermore, Abgrall and Mezine [5] proved that the limited N-scheme retains the monotonicity properties of the original N-scheme with the same CFL condition (51).

3.4. Extension to systems

Following [12], the N-scheme is extended to systems of conservation laws by first defining the following averaged flux Jacobians:

$$\widehat{J}^1 \equiv \frac{\partial f}{\partial q}(\widehat{Z}), \quad \widehat{J}^2 \equiv \frac{\partial g}{\partial q}(\widehat{Z}), \quad \text{and} \quad \vec{J} \equiv (\widehat{J}^1, \widehat{J}^2), \tag{54}$$

where Z is a parameterization of Q and

$$\widehat{Z} = \frac{1}{3}(Z_1 + Z_2 + Z_3). \tag{55}$$

In order to achieve a conservative linearization we must find a parameter vector, Z, such that the following constraint is satisfied:

$$\sum_{i=1}^{3} (\vec{n}_i \cdot \vec{J}) \widehat{Q}_i = \oint_{\partial T} \vec{f}(z^h) \cdot d\vec{s}, \tag{56}$$

where z^h is the linear interpolant that passes through the points $(\vec{x}_i, Z_i)$ for $i = 1, 2, 3$ and

$$\widehat{Q}_i \equiv \frac{\partial q}{\partial z}(\widehat{Z}) Z_i. \tag{57}$$

As was shown in [12] (see also [1,11]), constraint (56) will in general only be satisfied if we are able to find a parameterization, $z = (z^1(q), z^2(q), \ldots, z^m(q))$, such that the flux, $\vec{f}(q(z))$, depends at most quadratically on each z^p for $p = 1, 2, \ldots, m$. For

the Euler equations from gas dynamics, as well as related systems, such a parameterization is known [12]. Assuming that a parameterization has been found, we proceed by diagonalizing the flux Jacobian:

$$\vec{n}_i \cdot \vec{J} = R_i \Lambda_i R_i^{-1},$$

where R_i is the matrix of right eigenvectors and Λ_i is the diagonal matrix of eigenvalues. Following the philosophy of Roe's approximate Riemann solver [23], the systems N-scheme is obtained by applying the scalar N-scheme to each characteristic component. This results in the following method:

$$\text{Systems N-scheme}: \quad \Phi_i^{\mathcal{T}} = \frac{1}{2} R_i \Lambda_i^+ R_i^{-1} (\widehat{Q}_i - Q_\star). \tag{58}$$

The upwind parameter can be recovered by enforcing local conservation:

$$Q_\star = \left\{ \sum_{i=1}^{3} R_i \Lambda_i^- R_i^{-1} \right\}^{-1} \left\{ \sum_{j=1}^{3} R_j \Lambda_j^- R_j^{-1} \widehat{Q}_j \right\}. \tag{59}$$

Note that solving for $Q_\star$ involves inverting an $m \times m$ matrix. Finally, we note that although it is based on a generalization of the monotone scalar N-scheme, the systems N-scheme is in general only approximately non-oscillatory for nonlinear systems of conservation laws. In practice, however, this scheme has been shown to work quite well for steady-state shock computations for systems such as the Euler equations from gas dynamics [1].

The systems N-scheme described so far is not linear preserving. In order that the limiting procedure developed for scalar equations can be re-used for systems, Abgrall and Mezine [5] proposed to project the distributed residuals into the eigenspace of the Roe–Struijs–Deconinck averaged flux Jacobian in some direction $\vec{n}$. In practice, the direction $\vec{n}$ is chosen from physical considerations. For example, in the case of the shallow water equations or the Euler equation from gas dynamics, an approach that gives good results in practice is to take $\vec{n}$ to be the local Roe–Struijs–Deconinck averaged fluid velocity: $\vec{n} = \vec{u}$. Once a limiting direction has been chosen, the limiting procedure can be summarized as follows:

$$\text{for } p = 1 \ldots m$$
$$\quad \text{for } i = 1, 2, 3: \quad \text{set} \quad \Theta_i^p = \ell^p \cdot \Phi_i^{\mathcal{T}};$$
$$\quad \text{for } i = 1, 2, 3: \quad \text{set} \quad \beta_i^p = \frac{\Theta_i^p}{\sum_{j=1}^3 \Theta_j^p};$$
$$\quad \text{for } i = 1, 2, 3: \quad \text{set} \quad \tilde{\beta}_i^p = \frac{[\beta_i^p]^+}{\sum_{j=1}^3 [\beta_j^p]^+};$$
$$\text{end}$$
$$\quad \text{for } i = 1, 2, 3: \quad \text{set} \quad \Phi_i^{\mathcal{T}} = \sum_{p=1}^m \tilde{\beta}_i^p \Theta_i^p r^p,$$

where ℓ^p and r^p are the pth left and right eigenvectors of $\vec{n} \cdot \vec{J}$, respectively.

3.5. A correction for improved convergence

As was pointed out by Abgrall [2], the N-scheme in conjunction with the limiting procedure outlined in Section 3.4 has one major drawback: the method does not in general converge to a steady-state solution. The problem is not with the N-scheme itself, since this method does converge to a steady-state, but instead the problem lies in how the N-scheme interacts with the limiting procedure. In the same paper, Abgrall [2] also provided a cure for this problem. He arrived at the following distributed residual:

$$\Phi_i^{\mathcal{T}} = \underbrace{B_i \Phi^{\mathcal{T}}}_{\text{limited N-scheme}} + \underbrace{\theta |\mathcal{T}|^{-1/2} K_i \Phi^{\mathcal{T}}}_{\text{correction}}, \tag{60}$$

where $K_i = (\vec{n}_i \cdot \vec{J})/2$, $|\mathcal{T}|$ is the area of triangle $\mathcal{T}$, and θ is a grid and solution dependent parameter. Notice that conservation is not affected by this correction term. In order to produce a scheme that converges to a steady-state solution, θ needs to be chosen so that the correction is relatively small near shocks ($\theta = \mathcal{O}(|\mathcal{T}|^{1/2})$) and relatively large in smooth regions ($\theta = 1$). Abgrall [2] proposed the following formula:

$$\theta = \min\left(1, \frac{|\mathcal{T}|}{|\varphi^{\mathcal{T}}| + 10^{-10}}\right), \tag{61}$$

where $\varphi^{\mathcal{T}}$ is the projection of $\Phi^{\mathcal{T}}$ onto some important eigen-direction. In the case of the compressible Euler equations, $\varphi^{\mathcal{T}}$ should be taken to be the projection of $\Phi^{\mathcal{T}}$ onto the entropy wave. In Section 5, in which we consider several numerical examples, we will refer to the limited and corrected N-scheme as the N\LC-scheme.

4. Multidimensional relaxation systems

Having reviewed the relaxation scheme paradigm in Section 2 and residual distribution schemes in Section 3, we now turn to develop a multidimensional relaxation system framework. We again begin with the case of a scalar conservation law and introduce the following relaxation system:

$$
\begin{bmatrix} q \\ \mu^1 \\ \mu^2 \end{bmatrix}_{,t} + A^1 \begin{bmatrix} q \\ \mu^1 \\ \mu^2 \end{bmatrix}_{,x} + A^2 \begin{bmatrix} q \\ \mu^1 \\ \mu^2 \end{bmatrix}_{,y} = \frac{1}{\varepsilon} \begin{bmatrix} 0 \\ f(q) - \mu^1 \\ g(q) - \mu^2 \end{bmatrix},
\tag{62}
$$

where

$$
A^1 \equiv \begin{bmatrix} 0 & 1 & 0 \\ -c^1 d^1 & c^1 + d^1 & 0 \\ -c^2 d^1 - \frac{1}{4}(c^1 - d^1)(c^2 - d^2) & \frac{1}{2}(c^2 + d^2) & \frac{1}{2}(c^1 + d^1) \end{bmatrix},
\tag{63}
$$

$$
A^2 \equiv \begin{bmatrix} 0 & 0 & 1 \\ -c^1 d^2 - \frac{1}{4}(c^1 - d^1)(c^2 - d^2) & \frac{1}{2}(c^2 + d^2) & \frac{1}{2}(c^1 + d^1) \\ -c^2 d^2 & 0 & c^2 + d^2 \end{bmatrix}.
\tag{64}
$$

In these expressions, we assume that $\vec{c}, \vec{d} \in \mathbb{R}^2$. Just as in the 1D case, we will separate the effects of the hyperbolic left-hand side of this equation from the relaxation source term on the right-hand side by by viewing (62) as being comprised of the following two sub-problems:

$$
\begin{bmatrix} q \\ \mu^1 \\ \mu^2 \end{bmatrix}_{,t} + A^1 \begin{bmatrix} q \\ \mu^1 \\ \mu^2 \end{bmatrix}_{,x} + A^2 \begin{bmatrix} q \\ \mu^1 \\ \mu^2 \end{bmatrix}_{,y} = 0,
\tag{65}
$$

$$
\begin{bmatrix} \mu^1 \\ \mu^2 \end{bmatrix}_{,t} = \frac{1}{\varepsilon} \begin{bmatrix} f(q) - \mu^1 \\ g(q) - \mu^2 \end{bmatrix}.
\tag{66}
$$

In subsequent discussion we will make use of the following matrix:

$$
A(\vec{n}) = n^1 A^1 + n^2 A^2,
\tag{67}
$$

where $\vec{n} \in \mathbb{R}^2$ such that $\|\vec{n}\| \neq 0$. The three eigenvalues of $A(\vec{n})$ are given by

$$
\lambda^{1,3} = \frac{1}{2}\left(\vec{c} \cdot \vec{n} + \vec{d} \cdot \vec{n}\right) \pm \frac{1}{2}\sqrt{\left(n^1 d^1 - n^1 c^1\right)^2 + \left(n^2 d^2 - n^2 c^2\right)^2},
\tag{68}
$$

$$
\lambda^2 = \frac{1}{2}\left(\vec{c} \cdot \vec{n} + \vec{d} \cdot \vec{n}\right).
\tag{69}
$$

We note that these eigenvalues are strictly real.

Next we define a *relaxation procedure* that is analogous to the 1D case; the main difference is that in 2D a genuinely multidimensional Riemann problem such as the one depicted in Fig. 2 must be solved. Instead of attempting to solve this exactly, we solve it with the standard N-scheme. The full procedure can then be summarized as follows:

(1) Choose values for the parameters c^1, c^2, d^1, and d^2.
(2) On an arbitrary triangle, $\mathcal{T}$, approximately solve the multidimensional Riemann problem associated with (65) by applying the standard N-scheme.
(3) Approximate the effect of Eq. (66) on the solution calculated in Step (2) by directly setting $\mu^1 = f(q)$ and $\mu^2 = g(q)$. In other words, instantaneously relax the solution from Step (2) to the $\varepsilon \to 0$ limit.

4.1. The N-scheme

The first scheme that we will produce with the relaxation procedure is the N-scheme applied to the original scalar conservation law. We set

$$
\vec{c} = \vec{d} = \vec{u},
\tag{70}
$$

where $\vec{u}$ is the Roe–Struijs–Deconinck average that satisfies (46). The above choice for $\vec{c}$ and $\vec{d}$ results in the following coefficient matrix for the relaxation system:

$$A(\vec{n}) = \begin{bmatrix} 0 & n^1 & n^2 \\ -u^1(\vec{u} \cdot \vec{n}) & n^1 u^1 + \vec{u} \cdot \vec{n} & n^2 u^1 \\ -u^2(\vec{u} \cdot \vec{n}) & n^1 u^2 & n^2 u^2 + \vec{u} \cdot \vec{n} \end{bmatrix}; \tag{71}$$

this matrix has eigenvalues given by

$$\lambda^{1,2,3} = \vec{u} \cdot \vec{n}, \tag{72}$$

and, as in the 1D case, has an incomplete set of eigenvectors (i.e., the eigenvalues have algebraic multiplicity 3, but geometric multiplicity of only 2).

Because $A(\vec{n})$ only has two linearly independent eigenvectors it cannot be diagonalized; and instead, we reduce it to Jordan canonical form via the following similarity transformation:

$$A(\vec{n}) = SMS^{-1}, \tag{73}$$

where

$$M = \begin{bmatrix} \vec{n} \cdot \vec{u} & 1 & 0 \\ 0 & \vec{n} \cdot \vec{u} & 0 \\ 0 & 0 & \vec{n} \cdot \vec{u} \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} 1 & -\frac{1}{\vec{n} \cdot \vec{u}} & 0 \\ u^1 & -n^2 & -n^2 \\ u^2 & n^1 & n^1 \end{bmatrix}. \tag{74}$$

In approximately solving the Riemann problem via the N-scheme (Step (2) of the relaxation procedure), we will need to make sense of the expression $[A(\vec{n})]^+$. Without a full set of eigenvectors, we do this in the following way:

$$[A(\vec{n})]^+ \equiv SM^+S^{-1}, \quad \text{where} \quad M^+ = \begin{bmatrix} [\vec{n} \cdot \vec{u}]^+ & 1 & 0 \\ 0 & [\vec{n} \cdot \vec{u}]^+ & 0 \\ 0 & 0 & [\vec{n} \cdot \vec{u}]^+ \end{bmatrix}. \tag{75}$$

This results in two possibilities:

(1) $[\vec{n} \cdot \vec{u}]^+ = (\vec{n} \cdot \vec{u}) \quad \Rightarrow \quad [A(\vec{n})]^+ = A(\vec{n})$;

(2) $[\vec{n} \cdot \vec{u}]^+ = 0 \quad \Rightarrow \quad [A(\vec{n})]^+ = \begin{bmatrix} -(\vec{n} \cdot \vec{u}) & n^1 & n^2 \\ -u^1(\vec{n} \cdot \vec{u}) & u^1 n^1 & u^1 n^2 \\ -u^2(\vec{n} \cdot \vec{u}) & u^2 n^1 & u^2 n^2 \end{bmatrix}$.

The first of these two expressions is exactly the result one should expect; the second expression, however, is somewhat troubling. We should expect that $[A(\vec{n}_i)]^+(U_i - U_\star) = 0$ if $(\vec{n}_i \cdot \vec{u}) \leqslant 0$, where

$$U_i \equiv (Q_i, \vec{f}(Q_i)) \quad \text{and} \quad U_\star \equiv (Q_\star, \vec{\mu}_\star).$$

Instead, we are currently stuck with the following result when $(\vec{n} \cdot \vec{u}) \leqslant 0$:

$$[A(\vec{n}_i)]^+(U_i - U_\star) = \left\{ -(\vec{u} \cdot \vec{n})(Q_i - Q_\star) + \vec{n} \cdot (\vec{f}(Q_i) - \vec{\mu}_\star^1) \right\} \begin{bmatrix} 1 \\ u^1 \\ u^2 \end{bmatrix}.$$

In order to clean up this result, we are forced to slightly modify the relaxation procedure for the N-scheme. We will leave the sub-problem (65) alone, but replace sub-problem (66) with the the following system of ODEs:

$$\begin{bmatrix} \mu^1 \\ \mu^2 \end{bmatrix}_{,t} = \frac{1}{\varepsilon} \begin{bmatrix} \frac{\partial f}{\partial q}(\hat{q})q - \mu^1 \\ \frac{\partial g}{\partial q}(\hat{q})q - \mu^2 \end{bmatrix}, \tag{76}$$

where $\hat{q}$ is a piecewise constant function in space that is constant on each triangle $\mathcal{T}$; the value of this constant is $\widehat{Q}$, the multidimensional Roe–Struijs–Deconinck average (46). This results in the following modification of Step (3) in the relaxation procedure:

(3) On each triangle $\mathcal{T}$ approximate the effect of (76) on the solution calculated in Step (2) by directly setting $\mu^1 = u^1 q$ and $\mu^2 = u^2 q$. In other words, instantaneously relax the solution from Step (2) to the $\varepsilon \to 0$ limit.

Note that in general

$$\frac{\partial f}{\partial q}(\hat{q})q \neq f(q) \quad \text{and} \quad \frac{\partial g}{\partial q}(\hat{q})q \neq g(q);$$

and therefore, replacing (66) with (76) will yield a different numerical scheme. As we will demonstrate below, it is the scheme based on (76) that will reproduce the N-scheme.

The solution value at each node is now given by

$$U_i \equiv (Q_i, u^1 Q_i, u^2 Q_i). \tag{77}$$

Additionally, we enforce the condition:

$$U_\star \equiv (Q_\star, u^1 Q_\star, u^2 Q_\star). \tag{78}$$

With these modifications it is now true that $[A(\vec{n}_i)]^+ (U_i - U_\star) = 0$ if $(\vec{n}_i \cdot \vec{u}) \leqslant 0$.

In order to proceed with the relaxation procedure, we solve a Riemann problem between three states of the form (77) with $i = 1,2,3$. Solving this Riemann problem via the N-scheme tells us that the residuals distributed to each node are given by

$$\varphi_i^{\mathcal{T}} = \frac{1}{2}[A(\vec{n}_i)]^+ (U_i - U_\star) \equiv \begin{cases} \frac{1}{2}A(\vec{n}_i)(U_i - U_\star) & \text{if } \vec{u} \cdot \vec{n}_i > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{79}$$

where $U_\star$ has the form (78). In order to determine $U_\star$ in terms of the U_i values, we add the three residuals given by (79) and enforce that this sum yields the total residual in triangle $\mathcal{T}$:

$$\sum_{i=1}^{3} \varphi_i^{\mathcal{T}} = \frac{1}{2} \sum_{i=1}^{3} A(\vec{n}_i) U_i, \tag{80}$$

$$\Rightarrow \quad \frac{1}{2} \sum_{i=1}^{3} [A(\vec{n}_i)]^+ (U_i - U_\star) = \frac{1}{2} \sum_{i=1}^{3} \begin{bmatrix} (\vec{n}_i \cdot \vec{u}) Q_i \\ u^1 (\vec{n}_i \cdot \vec{u}) Q_i \\ u^2 (\vec{n}_i \cdot \vec{u}) Q_i \end{bmatrix}. \tag{81}$$

If $\|\vec{u}\| > 0$, then we note the following result on each triangle $\mathcal{T}$:

(1) $\exists k \in (1,2,3)$ such that $\vec{u} \cdot \vec{n}_k > 0$,
(2) $\exists k \in (1,2,3)$ such that $\vec{u} \cdot \vec{n}_k < 0$.

This result implies that there are two possibilities whenever $\|\vec{u}\| > 0$: the 1-target case – $\exists$ exactly one k s.t. $\vec{u} \cdot \vec{n}_k > 0$, and the 2-target case – $\exists$ exactly two k s.t. $\vec{u} \cdot \vec{n}_k > 0$. Without loss of generality, let us assume that $\vec{u} \cdot \vec{n}_1 > 0$ and $\vec{u} \cdot \vec{n}_3 < 0$, which yields one of the two possibilities:

1-target solution: $\vec{u} \cdot \vec{n}_1 > 0, \quad \vec{u} \cdot \vec{n}_2 \leqslant 0, \quad \vec{u} \cdot \vec{n}_3 < 0,$ (82)
2-target solution: $\vec{u} \cdot \vec{n}_1 > 0, \quad \vec{u} \cdot \vec{n}_2 > 0, \quad \vec{u} \cdot \vec{n}_3 < 0.$ (83)

We consider each of these two cases below.

4.1.1. The 1-target solution

The 1-target case is easy to analyze: the total residual is completely distributed to the lone node that is downwind of the flow, which we have taken without loss of generality to be node 1. If we let Φ denote the component of the residual corresponding to Q, then the 1-target case results in the following residual distribution:

$$\Phi_1^{\mathcal{T}} = \frac{1}{2} \sum_{i=1}^{3} (\vec{u} \cdot \vec{n}_i) Q_i, \quad \Phi_2^{\mathcal{T}} = 0, \quad \Phi_3^{\mathcal{T}} = 0, \tag{84}$$

which is the same result that one would obtain with the N-scheme on the original scalar conservation law.

4.1.2. The 2-target solution

The 2-target case involves distribution to two nodes, which we have taken without loss of generality to be the 1 and 2 nodes. From equation (80) we arrive at the following linear system that must be solved in order to obtain the upwind parameter $U_\star$:

$$-(A(\vec{n}_1) + A(\vec{n}_2)) U_\star = A(\vec{n}_3) U_3. \tag{85}$$

However, it is not difficult to show that

$$A(\vec{n}_1) + A(\vec{n}_2) + A(\vec{n}_3) = 0 \quad \Rightarrow \quad A(\vec{n}_3) = -(A(\vec{n}_1) + A(\vec{n}_2)). \tag{86}$$

This implies that $U_\star \equiv U_3$. Therefore the 2-target case results in the following residual distribution corresponding to Q:

$$\Phi_1^{\mathcal{T}} = \frac{1}{2}(\vec{n}_1 \cdot \vec{u})(Q_1 - Q_3), \quad \Phi_2^{\mathcal{T}} = \frac{1}{2}(\vec{n}_2 \cdot \vec{u})(Q_1 - Q_3), \quad \Phi_3^{\mathcal{T}} = 0. \tag{87}$$

This result is again identical to the original scalar conservation law.

4.2. The RXN-scheme: genuinely multidimensional local Lax-Friedrichs

One of the main difficulties with the N-scheme is that computing the upwind parameter $Q_\star$ for complicated systems of conservation laws can become prohibitively expensive. Despite this fact, few alternatives have been developed in the RD literature. One such alternative was introduced by Abgrall [2], who considered a local Lax-Friedrichs-type method that was obtained, in analogy to the 1D case, by taking the unstable "centered" residual and adding the appropriate numerical viscosity. In this work, we construct a new method based on the idea of relaxation systems; this scheme can be viewed as a different multidimensional generalization of the 1D LLF method. For brevity we will call this method the RXN-scheme, which stands for "relaxation N-scheme[3]." In analogy with the 1D LLF method as derived in Section 2.3, we make the following choice for the parameters $\vec{c}$ and $\vec{d}$ in (62)–(64):

$$\vec{d} = -\vec{c} = (s^T, s^T). \tag{88}$$

Note that each triangle can have a different value of s^T; this is why we call it a 'local' Lax-Friedrichs. This choice results in a coefficient matrix of the form

$$A(\vec{n}) = \begin{bmatrix} 0 & n^1 & n^2 \\ n^1(s^T)^2 & 0 & 0 \\ n^2(s^T)^2 & 0 & 0 \end{bmatrix}, \tag{89}$$

which has an eigenvector decomposition given by

$$
\begin{aligned}
A(\vec{n}) &= SMS^{-1} \\
&= \begin{bmatrix} -1 & 0 & 1 \\ \frac{s^T n^1}{\|\vec{n}\|} & -\frac{n^2}{\|\vec{n}\|} & \frac{s^T n^1}{\|\vec{n}\|} \\ \frac{s^T n^2}{\|\vec{n}\|} & \frac{n^1}{\|\vec{n}\|} & \frac{s^T n^2}{\|\vec{n}\|} \end{bmatrix} \begin{bmatrix} -\|\vec{n}\|s^T & & \\ & 0 & \\ & & \|\vec{n}\|s^T \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & \frac{n^1}{2\|\vec{n}\|s^T} & \frac{n^2}{2\|\vec{n}\|s^T} \\ 0 & -\frac{n^2}{\|\vec{n}\|} & \frac{n^1}{\|\vec{n}\|} \\ \frac{1}{2} & \frac{n^1}{2\|\vec{n}\|s^T} & \frac{n^2}{2\|\vec{n}\|s^T} \end{bmatrix}.
\end{aligned}
\tag{90}
$$

The Chapman-Enskog expansion for this relaxation system can be written to first order as

$$q_{,t} + f(q)_x + g(q)_y \approx \varepsilon \nabla \cdot \left(\begin{bmatrix} (s^T)^2 - (f'(q))^2 & -f'(q)g'(q) \\ -f'(q)g'(q) & (s^T)^2 - (g'(q))^2 \end{bmatrix} \nabla q \right). \tag{91}$$

The eigenvalues of the diffusion matrix in the above expression are

$$\lambda^1 = (s^T)^2, \quad \lambda^2 = (s^T)^2 - (f'(q)^2 + g'(q)^2), \tag{92}$$

which results in the following restriction on the choice of the lone parameter s^T:

$$s^T \geqslant \|\vec{f}'(q)\|, \tag{93}$$

for all $q \in \mathcal{T}$.

Applying the N-scheme to the relaxation system with coefficient matrix (89), yields the following residual

$$\varphi_i = \frac{1}{2} S_i M_i^+ S_i^{-1} (U_i - U_\star), \tag{94}$$

where $U_\star = (Q_\star, \mu_\star^1, \mu_\star^2)$. Simplifying this expression gives

$$\varphi_i = \frac{1}{4} \left\{ s^T \|\vec{n}_i\|(Q_i - Q_\star) + \vec{n}_i \cdot (\vec{f}(Q_i) - \vec{\mu}_\star) \right\} \begin{bmatrix} 1 \\ s^T \frac{n_i^1}{\|\vec{n}_i\|} \\ s^T \frac{n_i^2}{\|\vec{n}_i\|} \end{bmatrix}. \tag{95}$$

In order to calculate $U_\star$, we must enforce conservation:

$$\sum_{i=1}^3 \varphi_i = \varphi^T = \frac{1}{2} \sum_{i=1}^3 \begin{bmatrix} \vec{n}_i \cdot \vec{f}(Q_i) \\ n_i^1 (s^T)^2 Q_i \\ n_i^2 (s^T)^2 Q_i \end{bmatrix}, \tag{96}$$

which results in the following linear system for the upwind parameters $(Q_\star, \vec{\mu}_\star)$:

---

[3] The words "N-scheme" appear here because we make use of the N-scheme to solve the homogeneous part of the relaxation system.

$$\sum_{i=1}^{3} \begin{bmatrix} s^{\mathcal{T}} \|\vec{n}_i\| & 0 & 0 \\ 0 & \frac{n_i^1 n_i^1}{\|\vec{n}_i\|} & \frac{n_i^1 n_i^2}{\|\vec{n}_i\|} \\ 0 & \frac{n_i^1 n_i^2}{\|\vec{n}_i\|} & \frac{n_i^2 n_i^2}{\|\vec{n}_i\|} \end{bmatrix} \begin{bmatrix} Q_\star \\ \mu_\star^1 \\ \mu_\star^2 \end{bmatrix} = \sum_{j=1}^{3} \begin{bmatrix} s^{\mathcal{T}} \|\vec{n}_i\| Q_j - \vec{n}_j \cdot \vec{f}(Q_j) \\ n_j^1 \left(\frac{\vec{n}_j}{\|\vec{n}_j\|} \cdot \vec{f}(Q_j) - s^{\mathcal{T}} Q_j \right) \\ n_j^2 \left(\frac{\vec{n}_j}{\|\vec{n}_j\|} \cdot \vec{f}(Q_j) - s^{\mathcal{T}} Q_j \right) \end{bmatrix}. \tag{97}$$

The solution to this linear system can be written as

$$Q_\star = \frac{\sum\limits_{j=1}^{3} \left(s^{\mathcal{T}} \|\vec{n}_j\| Q_j - \vec{n}_j \cdot \vec{f}(Q_j) \right)}{\sum\limits_{i=1}^{3} s^{\mathcal{T}} \|\vec{n}_i\|}, \tag{98}$$

$$\mu_\star^1 = \frac{1}{N} \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{n_i^2}{\|\vec{n}_i\|} \left(\frac{\vec{n}_j}{\|\vec{n}_j\|} \cdot \vec{f}(Q_j) - s^{\mathcal{T}} Q_j \right) (n_i^2 n_j^1 - n_j^2 n_i^1), \tag{99}$$

$$\mu_\star^2 = \frac{1}{N} \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{n_i^1}{\|\vec{n}_i\|} \left(\frac{\vec{n}_j}{\|\vec{n}_j\|} \cdot \vec{f}(Q_j) - s^{\mathcal{T}} Q_j \right) (n_i^1 n_j^2 - n_j^1 n_i^2), \tag{100}$$

where

$$N = \sum_{i=1}^{3} \sum_{j=1}^{3} \left\{ \frac{n_i^1 n_i^1 n_j^2 n_j^2 - n_i^1 n_j^1 n_i^2 n_j^2}{\|\vec{n}_i\| \|\vec{n}_j\|} \right\}. \tag{101}$$

Let us now take a moment to reflect on what just happened. Although the original coefficient matrix, (89), for this method was comically simple, after applying the N-scheme to this system on an arbitrary triangle, the resulting upwind parameters are somewhat complicated. On the other hand, we see from equation (97) that the parameter $Q_\star$ is completely decoupled from $\vec{\mu}_\star$. We make use of this last fact to construct an alternative scheme in the following way: instead of computing the components of $\vec{\mu}_\star$ from (99)–(101), we enforce

$$\vec{\mu}_\star \equiv \vec{f}(Q_\star) \tag{102}$$

by again invoking the $\varepsilon \to 0$ relaxation limit. Although this direct enforcement clearly gives a different scheme than if we had used (99)–(101), what we achieve with this approach is a very simple method that we refer to as the RXN-scheme (relaxation N-scheme). In terms of the residual distributed to node i in the Q-variable, we now obtain the following expression:

$$\text{RXN-scheme}: \quad \Phi_i^{\mathcal{T}} = \frac{1}{4} s^{\mathcal{T}} \|\vec{n}_i\| (Q_i - Q_\star) + \frac{1}{4} \vec{n}_i \cdot (\vec{f}(Q_i) - \vec{f}(Q_\star)), \tag{103}$$

where $Q_\star$ is given by (98). Note that this method is automatically conservative since $Q_\star$ still satisfies the first equation in linear system (97).

Theorem 4.1. (*Monotonicity*) If there exists a $\widetilde{Q}$ such that

$$\sum_{i=1}^{3} \vec{n}_i \cdot \vec{f}(Q_i) = \sum_{i=1}^{3} \vec{n}_i \cdot \vec{f}'(\widetilde{Q}) Q_i, \tag{104}$$

and

$$s^{\mathcal{T}} \geqslant \max \left\{ \|f'(\widetilde{Q})\|, \|f'(\overline{Q}_1)\|, \|f'(\overline{Q}_2)\|, \|f'(\overline{Q}_3)\| \right\}, \tag{105}$$

where for each j

$$\vec{n}_j \cdot (\vec{f}(Q_j) - \vec{f}(Q_\star)) = \vec{n}_j \cdot \vec{f}'(\overline{Q}_j)(Q_j - Q_\star) \tag{106}$$

from the Rankine-Hugoniot conditions, then the RXN-scheme as defined by (103) and (98) satisfies the following condition:

$$\Phi_i^{\mathcal{T}} = \sum_{j=1}^{3} c_{ij}^{\mathcal{T}} (Q_i - Q_j), \tag{107}$$

where $c_{ij}^{\mathcal{T}} \geqslant 0$ for all $i,j = 1,2,3$. This condition along with the following CFL constraint on the time step:

$$\Delta t \leqslant \min_i \left\{ \frac{2|C_i|}{\sum\limits_{\mathcal{T}:i\in\mathcal{T}} \|\vec{n}_i\| s^{\mathcal{T}}} \right\}, \tag{108}$$

is enough to guarantee that the RXN-scheme with forward Euler time-stepping (37) is monotonicity preserving.

Proof

(1) Using the Rankine-Hugoniot conditions (106), we rewrite the RXN-scheme as

$$\Phi_i = P_i^{\mathcal{T}}(Q_i - Q_\star), \quad \text{where} \quad P_i^{\mathcal{T}} \equiv \frac{1}{4}(s^{\mathcal{T}}\|\vec{n}_i\| + \vec{n}_i \cdot \vec{f}'(\overline{Q}_i)). \tag{109}$$

Similarly, we rewrite (98) as follows

$$Q_\star = \frac{\sum_{j=1}^{3} N_j^{\mathcal{T}} Q_j}{\sum_{j=1}^{3} N_j^{\mathcal{T}}}, \quad \text{where} \quad N_j^{\mathcal{T}} \equiv s^{\mathcal{T}}\|\vec{n}_j\| - \vec{n}_j \cdot \vec{f}'(\widetilde{Q}). \tag{110}$$

Note that the above expression was obtained by making use of (104) and the identity: $\sum_{k=1}^{3} \vec{n}_k \cdot \vec{f}'(\widehat{Q}) = 0$. Combining expressions (109) and (110) yields (107) with

$$c_{ij}^{\mathcal{T}} \equiv \frac{P_i^{\mathcal{T}} N_j^{\mathcal{T}}}{\sum_{k=1}^{3} N_k^{\mathcal{T}}}. \tag{111}$$

We note that $c_{ij}^{\mathcal{T}} \geqslant 0 \ \forall i,j \in (1,2,3)$, because (105) implies that $P_i^{\mathcal{T}} \geqslant 0 \ \forall \ i \in (1,2,3)$ and $N_j^{\mathcal{T}} \geqslant 0 \ \forall j \in (1,2,3)$.

(2) We now insert expression (107) into (37) and simplify:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{|C_i|} \sum_{\mathcal{T}:i\in\mathcal{T}} \sum_{j\in\mathcal{T}} c_{ij}^{\mathcal{T}} \left(Q_i^n - Q_j^n\right),$$

$$\Rightarrow Q_i^{n+1} = \left\{ 1 - \frac{\Delta t}{|C_i|} \sum_{\mathcal{T}:i\in\mathcal{T}} \sum_{j\in\mathcal{T}} c_{ij}^{\mathcal{T}} \right\} Q_i^n + \frac{\Delta t}{|C_i|} \sum_{\mathcal{T}:i\in\mathcal{T}} \sum_{j\in\mathcal{T}} c_{ij}^{\mathcal{T}} Q_j^n.$$

Monotonicity is achieved if Q_i^{n+1} is a convex average of all of the surrounding Q_j^n. Since each $c_{ij}^{\mathcal{T}} \geqslant 0$, we obtain a convex average provided that

$$\frac{\Delta t}{|C_i|} \left(\sum_{\mathcal{T}:i\in\mathcal{T}} \sum_{j\in\mathcal{T}} c_{ij}^{\mathcal{T}} \right) \leqslant 1 \quad \Rightarrow \quad \Delta t_i \leqslant \frac{|C_i|}{\left(\sum_{\mathcal{T}} \sum_{j} c_{ij}^{\mathcal{T}} \right)} = \frac{|C_i|}{\left(\sum_{\mathcal{T}} \sum_{j} \frac{P_i^{\mathcal{T}} N_j^{\mathcal{T}}}{\sum_k N_k^{\mathcal{T}}} \right)},$$

$$\Rightarrow \quad \Delta t \leqslant \min_i \left\{ \frac{|C_i|}{\sum_{\mathcal{T}:i\in\mathcal{T}} P_i^{\mathcal{T}}} \right\}.$$

The time restriction is clearly satisfied if we take (108). □

In practice the time step presented in the above theorem is overly restrictive. In the numerical simulations presented in Section 5, we instead use the same time-step as used with the N-scheme: 85% of the maximum CFL number given by expression (51).

4.3. Systems N-scheme

The systems generalization of coefficient matrix (89) for a system of m conserved variables is the following $3m \times 3m$ matrix:

$$A(\vec{n}) = \begin{bmatrix} 0\mathbb{I} & n^1\mathbb{I} & n^2\mathbb{I} \\ -(\vec{n}\cdot\vec{J})\widehat{J}^1 & n^1\widehat{J}^1 + \vec{n}\cdot\vec{J} & n^2\widehat{J}^1 \\ -(\vec{n}\cdot\vec{J})\widehat{J}^2 & n^1\widehat{J}^2 & n^2\widehat{J}^2 + \vec{n}\cdot\vec{J} \end{bmatrix}, \tag{112}$$

where $\mathbb{I}$ is again the $m \times m$ identity matrix, $0\mathbb{I}$ is the $m \times m$ matrix with zeros in every entry, and $\vec{J} = (\widehat{J}^1, \widehat{J}^2)^t$ is the flux Jacobian matrix evaluated at the Roe–Struijs–Deconinck average [12]. The systems generalization of (77)-(78) are the following $3m \times 1$ vectors:

$$U_i \equiv (Q_i, \widehat{J}^1 Q_i, \widehat{J}^2 Q_i) \quad \text{and} \quad U_\star \equiv (Q_\star, \widehat{J}^1 Q_\star, \widehat{J}^2 Q_\star). \tag{113}$$

In order to calculate the appropriate residuals in the relaxation procedure, we need to again understand how to create the matrices $[A(\vec{n})]^+$ and $[A(\vec{n})]^-$. As in the scalar case this is complicated by the fact that $A(\vec{n})$ does not have a full set of eigenvectors. In particular, the Jordan canonical form of this matrix can be written as

$$A(\vec{n}) = S \begin{bmatrix} D^1 & & \\ & \ddots & \\ & & D^m \end{bmatrix} S^{-1}, \quad \text{where} \quad D^p = \begin{bmatrix} \lambda^p & 1 & 0 \\ 0 & \lambda^p & 0 \\ 0 & 0 & \lambda^p \end{bmatrix}. \tag{114}$$

Here λ^p is the pth eigenvalue of the $m \times m$ matrix $\vec{n} \cdot \vec{J}$. We omit the complicated expression for the matrix S. In order to obtain an expression for $[A(\vec{n})]^+$, one has to replace each λ^p in the above expression with $[\lambda^p]^+$. Carrying this out results in the following matrix:

$$[A(\vec{n})]^+ = \begin{bmatrix} -\widehat{J}^- & n^1 \mathbb{I} & n^2 \mathbb{I} \\ -\widehat{J}^+\widehat{J}^1 - \widehat{J}^1\widehat{J}^- & n^1\widehat{J}^1 + \widehat{J}^+ & n^2\widehat{J}^1 \\ -\widehat{J}^+\widehat{J}^2 - \widehat{J}^2\widehat{J}^- & n^1\widehat{J}^2 & n^2\widehat{J}^2 + \widehat{J}^+ \end{bmatrix}, \tag{115}$$

where $\widehat{J}^\pm = (\vec{n} \cdot \vec{J})^\pm = R\Lambda^\pm R^{-1}$, Λ is the diagonal matrix of eigenvalues of $\vec{n} \cdot \vec{J}$, and R is the corresponding matrix right-eigenvectors. An analogous formula for $[A(\vec{n})]^-$ can also be readily constructed. From the above expression we find that

$$[A(\vec{n})]^\pm U_i = [(\vec{n} \cdot \widehat{J})^\pm Q_i, \widehat{J}^1(\vec{n} \cdot \widehat{J})^\pm Q_i, \widehat{J}^2(\vec{n} \cdot \widehat{J})^\pm Q_i]^t. \tag{116}$$

Having established expressions for $[A(\vec{n})]^\pm$, we now proceed by applying the N-scheme to the relaxation system:

$$\varphi_i = \frac{1}{2}[A(\vec{n}_i)]^+(U_i - U_\star) \Rightarrow \sum_{i=1}^3 [A(\vec{n}_i)]^- U_i = \left(-\sum_{i=1}^3 [A(\vec{n})]^+\right) U_\star, \tag{117}$$

where

$$-\sum_{i=1}^3 [A(\vec{n}_i)]^+ = \begin{bmatrix} \sum_i \widehat{J}_i^- & 0 & 0 \\ -\sum_i [\widehat{J}_i^- \widehat{J}^1 + \widehat{J}^1 \widehat{J}_i^-] & \sum_i \widehat{J}_i^- & 0 \\ -\sum_i [\widehat{J}_i^- \widehat{J}^2 + \widehat{J}^2 \widehat{J}_i^-] & 0 & \sum_i \widehat{J}_i^- \end{bmatrix} \tag{118}$$

and $\widehat{J}_i = (\vec{n}_i \cdot \vec{J})$. The unique solution to the linear system in (117) is $U_\star = (Q_\star, \widehat{J}^1 Q_\star, \widehat{J}^2 Q_\star)$ with

$$Q_\star = \left\{\sum_{i=1}^3 (\vec{n}_i \cdot \vec{J})^-\right\}^{-1} \left\{\sum_{i=1}^3 (\vec{n}_i \cdot \vec{J})^- Q_i\right\}, \tag{119}$$

and the component of the residual φ_i associated with Q can be written as

$$\Phi_i^T = \frac{1}{2}(\vec{n}_i \cdot \vec{J})^+(Q_i - Q_\star). \tag{120}$$

This result shows that this relaxation scheme identically reproduces the systems N-scheme (58)–(59).

4.4. Systems RXN-scheme

Just as the LLF method in the one-dimensional case, the RXN-scheme extends to systems of conservation laws in a simple manner. All that we have to do is apply the scalar version of the scheme to each component of the vector conserved variable. The coefficient matrix in the relaxation procedure can be written as

$$A(\vec{n}) = \begin{bmatrix} 0\mathbb{I} & n^1\mathbb{I} & n^2\mathbb{I} \\ n^1 s^2\mathbb{I} & 0\mathbb{I} & 0\mathbb{I} \\ n^2 s^2\mathbb{I} & 0\mathbb{I} & 0\mathbb{I} \end{bmatrix}, \tag{121}$$

where $\mathbb{I}$ is the $m \times m$ identity matrix. In order to satisfy the sub-characteristic condition we require that

$$s \geqslant \max_{p=1,\ldots,m} \sqrt{[\lambda^{p,x}(q)]^2 + [\lambda^{p,y}(q)]^2}, \tag{122}$$

over all $q \in \mathcal{T}$. In the above expression $\lambda^{p,x}$ and $\lambda^{p,y}$ are the pth eigenvalue of $\partial f/\partial q$ and $\partial g/\partial q$, respectively.

We note that the systems RXN-scheme, and in particular, the version of this scheme with limiters (Section 3.4) and onvergence corrections (Section 3.5), provides an alternative to the systems N-scheme that does not require the inversion of an $m \times m$ matrix in each element at each time level, nor does it require any special entropy fixes or special treatment near

stagnation points. Since this method is also simpler than the N-scheme, it should also yield some gains in computational efficiency. The systems N-scheme and RXN-scheme are compared in detail in Section 5.

4.5. RXN-scheme in d-dimensions

The above procedure for obtaining the 2D RXN-scheme can be generalized to any space dimension. In the d-dimensional case we arrive at the following scheme:

$$\text{RXN}_d\text{-scheme}: \quad \Phi_i^{\mathcal{T}} = \frac{s^{\mathcal{T}} \|\vec{n}_i\| (Q_i - Q_\star) + \vec{n}_i \cdot (\vec{f}(Q_i) - \vec{f}(Q_\star))}{2d}, \tag{123}$$

where

$$Q_\star = \frac{\sum_{j=1}^{d+1} \left(s^{\mathcal{T}} \|\vec{n}_j\| Q_j - \vec{n}_j \cdot \vec{f}(Q_j) \right)}{\sum_{i=1}^{d+1} s^{\mathcal{T}} \|\vec{n}_i\|}. \tag{124}$$

In particular, we note that for $d = 1$, this scheme exactly reduces to the 1D local Lax-Friedrichs method [24]. We also mote that the $1/d$ geometric factor comes from the d-dimensional N-scheme; see for example Eq. (7) in [11].
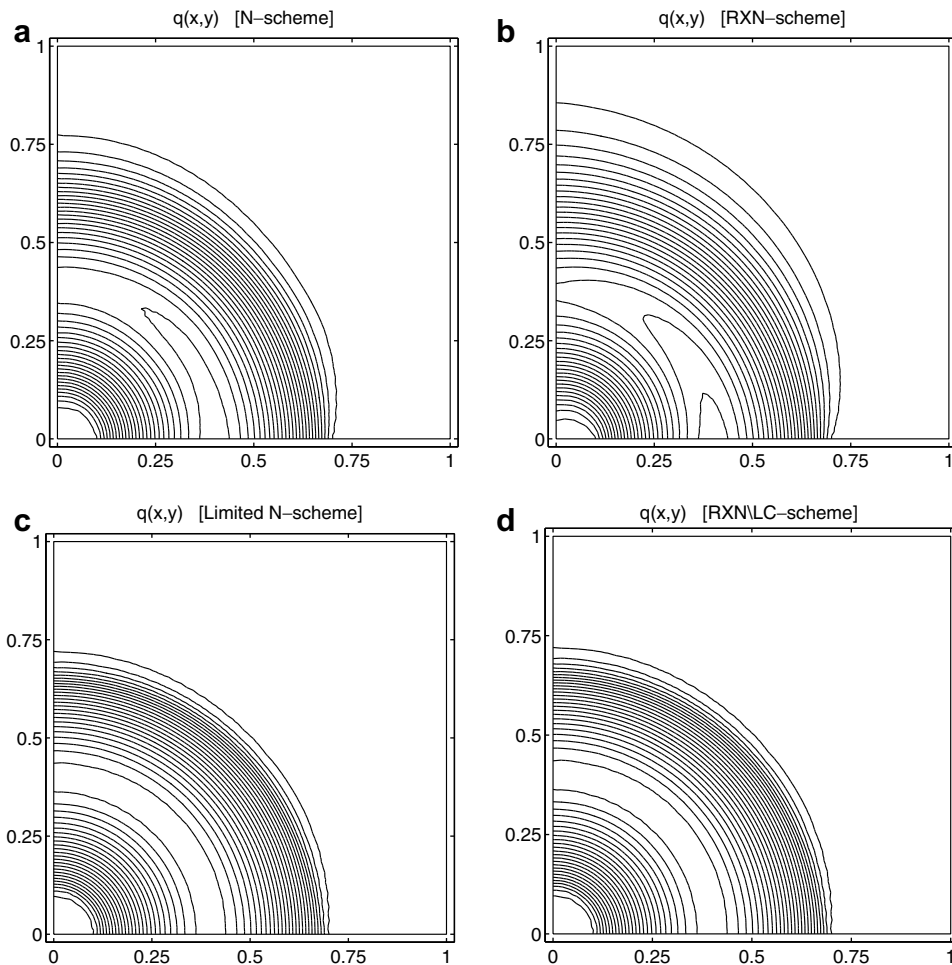
Fig. 3. Advection equation example. Shown in these panels are (a) the basic N-scheme, (b) the basic RXN-scheme, (c) the limited N-scheme (no convergence correction is needed for the limited N-scheme on scalar equations), and (d) the RXN\LC-scheme (convergence corrections are needed for the limited RXN-scheme, even for scalar problems). These results show that the basic RXN scheme is far more diffusive than the N-scheme. However, with limiting and convergence corrections, the RXN\LC gives results comparable to the limited N-scheme.
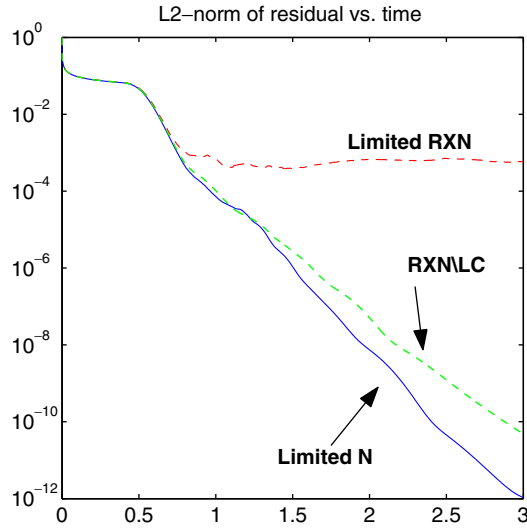
Fig. 4. Advection equation example. L_2-norms of the total residual for the limited N-scheme, the limited RXN-scheme, and the RXN\LC-scheme. This plot shows that the limited N-scheme does not need additional convergence corrections for scalar equations, but the limited RXN-scheme clearly does.

5. Numerical examples

In this section we compare the N-scheme and the newly proposed RXN scheme on several numerical examples. We will refer to the versions of the N-scheme and RXN-scheme that have been limited according to the procedure outlined in Section 3.4 and corrected according to the procedure outlined in Section 3.5 as the N\LC-scheme and RXN\LC-scheme, respectively.

5.1. Steady-state advection

First, we consider the advection equation on $[0,1] \times [0,1]$:

$$q_{,t} + \vec{u} \cdot \nabla q = 0, \tag{125}$$

with non-divergent velocity and boundary conditions given by

$$\vec{u}(x,y) = (-\pi y, \pi x),$$

$$q(1,y) = 0, \quad q(x,0) = \begin{cases} \sin\left(\pi\left(\frac{0.7-x}{0.6}\right)\right) & \text{if } 0.1 < x < 0.7, \\ 0 & \text{otherwise}. \end{cases}$$
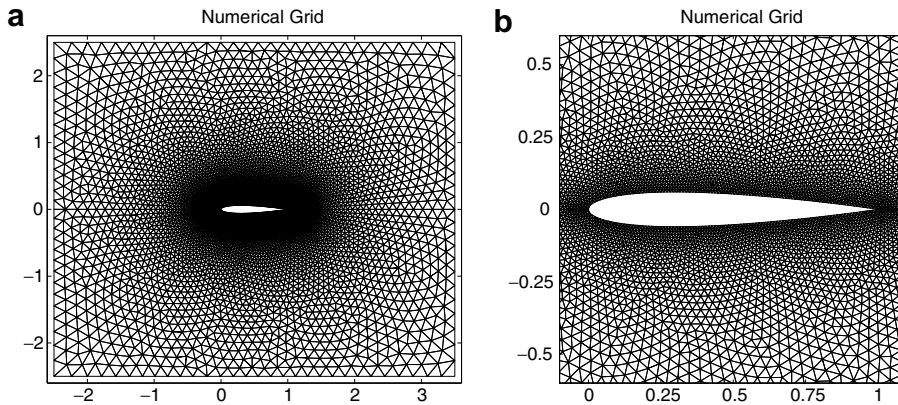
Fig. 5. The numerical grid for the NACA 0012 example. Panel (a) shows the entire domain, while Panel (b) shows a zoomed-in view of the airfoil. The grid has a total of 14,284 elements and 7298 nodes. The smallest grid elements near the airfoil have a triangle radius h that is roughly 30 times smaller than that of the largest grid elements on the outer boundaries.

This same problem was considered in [2].

For a non-divergent velocity field, an elegant way to solve the advection equation using the N-scheme is through the introduction of a streamfunction:

$$\psi(x, y) = -\frac{\pi}{2}(x^2 + y^2),$$

such that $\vec{u} = (\partial \psi / \partial y, -\partial \psi / \partial x)$. The N-scheme can then be written as

$$\Phi_i^{\mathrm{T}} = k_i^+ (Q_i - Q_\star),$$

where
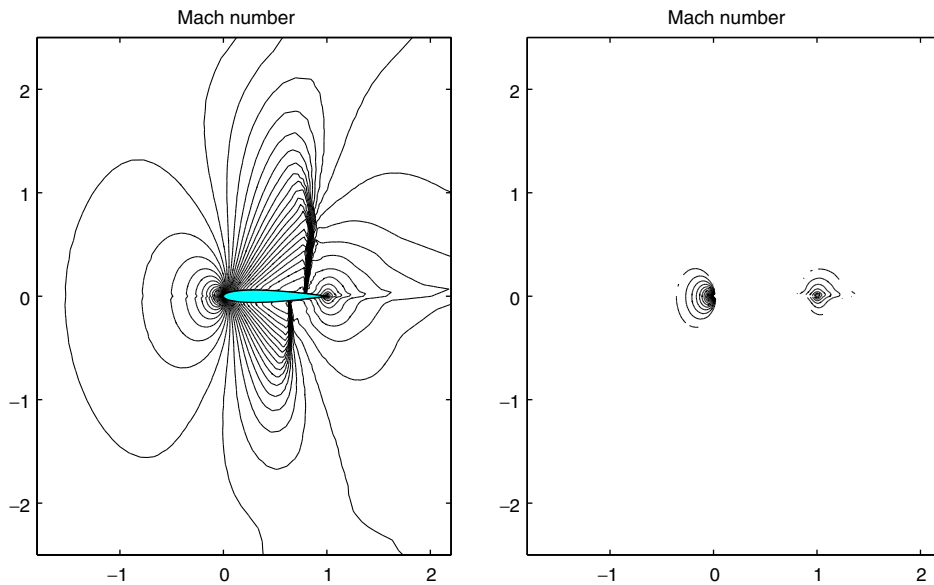
$$k_1 = \frac{1}{2}(\psi(x_2, y_2) - \psi(x_3, y_3)),$$

$$k_2 = \frac{1}{2}(\psi(x_3, y_3) - \psi(x_1, y_1)),$$

$$k_3 = \frac{1}{2}(\psi(x_1, y_1) - \psi(x_2, y_2)).$$

The advantage of this formulation is that we achieve numerical conservation, even though the equations are solved in advective form.

For the RXN scheme, we use residual (103) where the flux functions are given by

$$\vec{f}(Q_i) = \vec{u}_i Q_i, \quad \vec{f}(Q_\star) = \vec{u}_\star Q_\star, \quad \vec{u}_\star = \frac{\|\vec{n}_1\|\vec{u}_1 + \|\vec{n}_2\|\vec{u}_2 + \|\vec{n}_3\|\vec{u}_3}{\|\vec{n}_1\| + \|\vec{n}_2\| + \|\vec{n}_3\|},$$

and $Q_\star$ is given by (97) as usual.
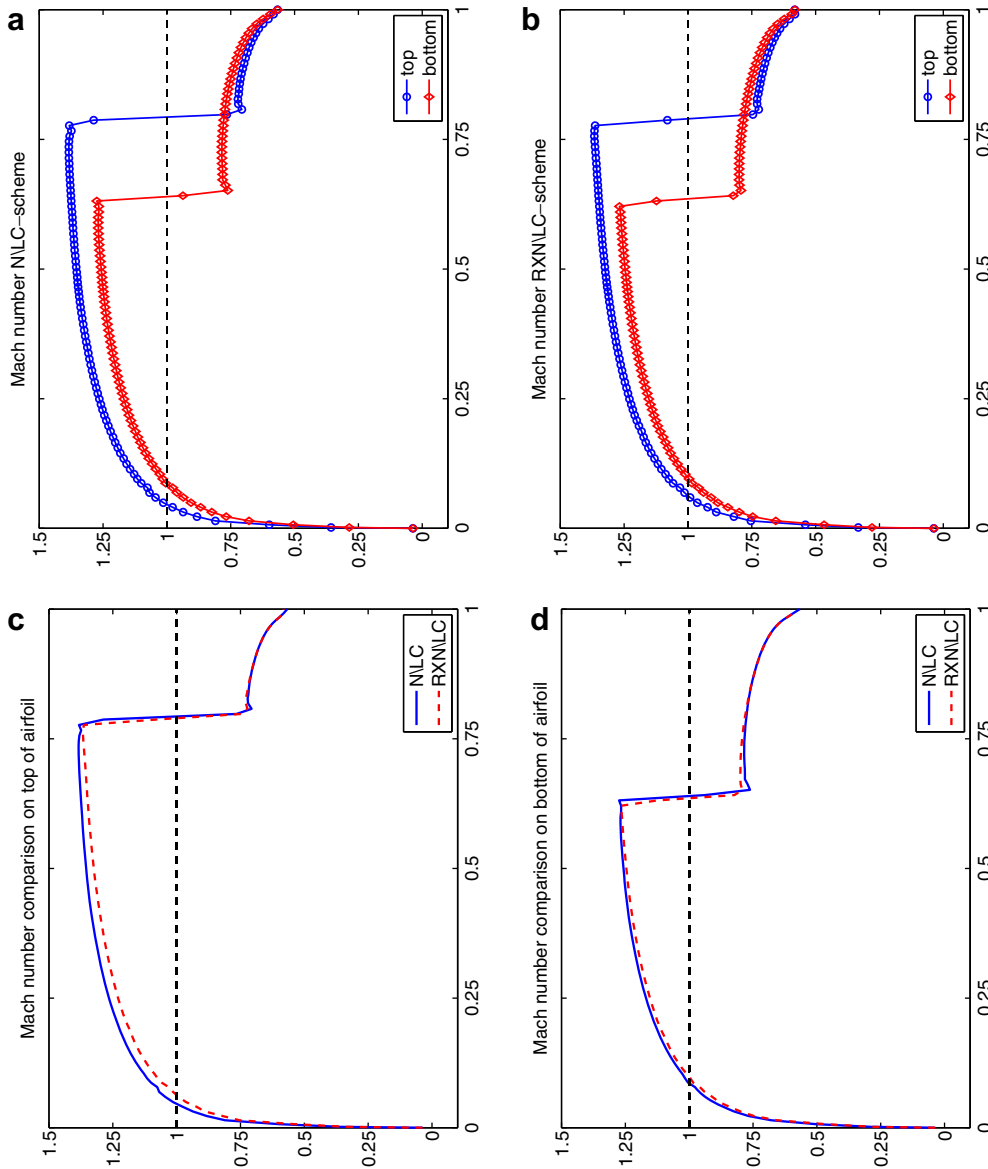
Fig. 7. The Mach number along the top and bottom edges of the NACA 0012 airfoil. Panel (a) is the N\LC-scheme solution, while Panel (b) is the RXN\LC-scheme solution. In Panels (c) and (d) we directly compare the Mach number profiles for each method: Panel (c) shows the Mach number on the top portion of the airfoil and Panel (d) shows the Mach number on the bottom portion of the airfoil. These results show that the RXN\LC-scheme produces comparable results to the N\LC-scheme.

Results on a grid with 5592 elements and 2903 nodes is shown in Fig. 3; displayed in each panel are (a) the basic N-scheme, (b) the basic RXN-scheme, (c) the limited N-scheme (no convergence correction is needed for the limited N-scheme on scalar equations), and (d) the RXN\LC-scheme (convergence corrections are needed for the limited RXN-scheme, even for scalar problems). These results show that the basic RXN scheme is far more diffusive than the N-scheme. However, with limiting and convergence corrections, the RXN\LC-scheme gives results comparable to the limited N-scheme. Convergence histories for the limited N-scheme, limited RXN-scheme, and the RXN\LC-scheme are shown in Fig. 4.

Clearly there is no advantage in using the RXN\LC-scheme over the limited N-scheme for a scalar problem, since the two methods have the same computational cost and the scalar limited N-scheme does not require convergence corrections. However, for hyperbolic systems such as the Euler equations, the RXN\LC-scheme provides a simpler algorithm with lower computational cost than the N\LC-scheme.
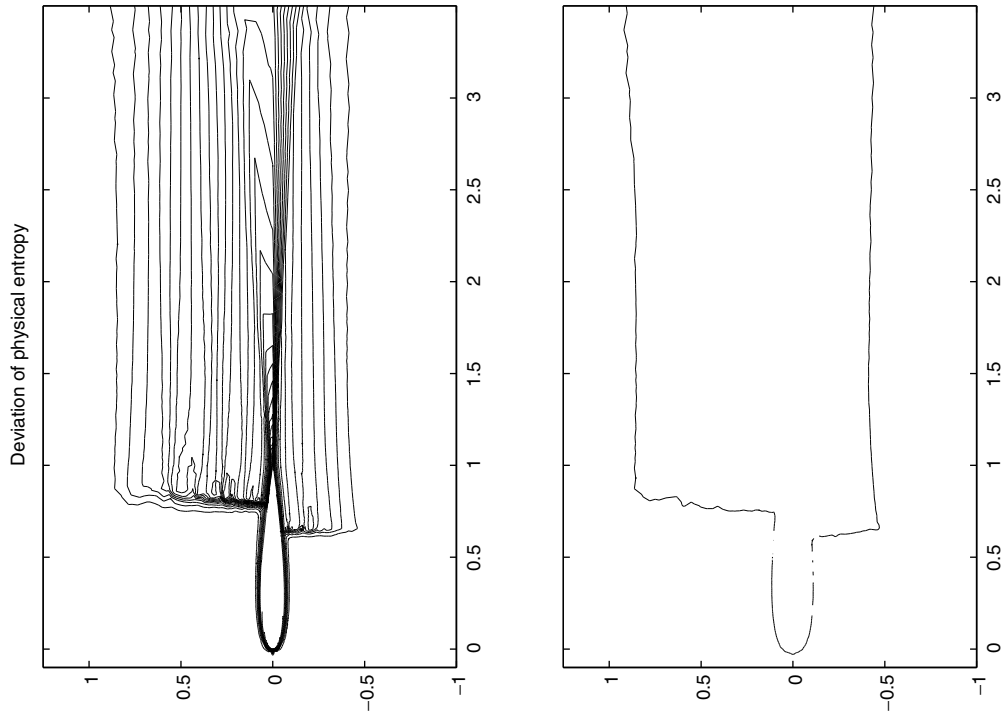
5.2. Transonic flow around the NACA 0012 airfoil

Next we consider transonic flow around the NACA 0012 airfoil using the compressible Euler equations from gas dynamics as our model. The Euler equations can be written as

$$
\begin{bmatrix} \rho \\ \rho\vec{u} \\ \mathcal{E} \end{bmatrix}_{,t} + \nabla \cdot \begin{bmatrix} \rho\vec{u} \\ \rho\vec{u}\vec{u} + p\mathbb{I} \\ \vec{u}(\mathcal{E} + p) \end{bmatrix} = 0, \tag{126}
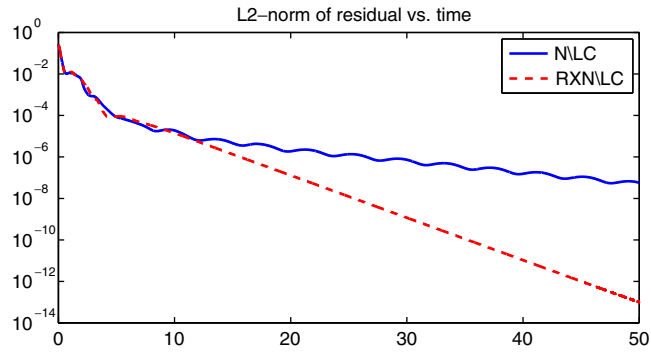$$
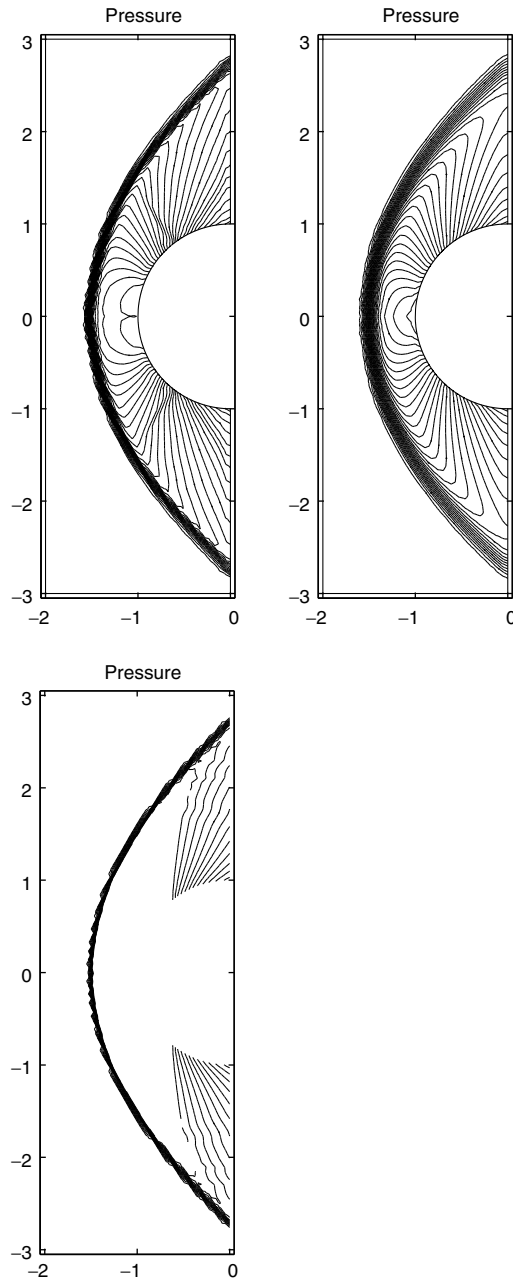
Fig. 11. L_2-norm of the total residual as a function of time for supersonic flow past a cylinder.
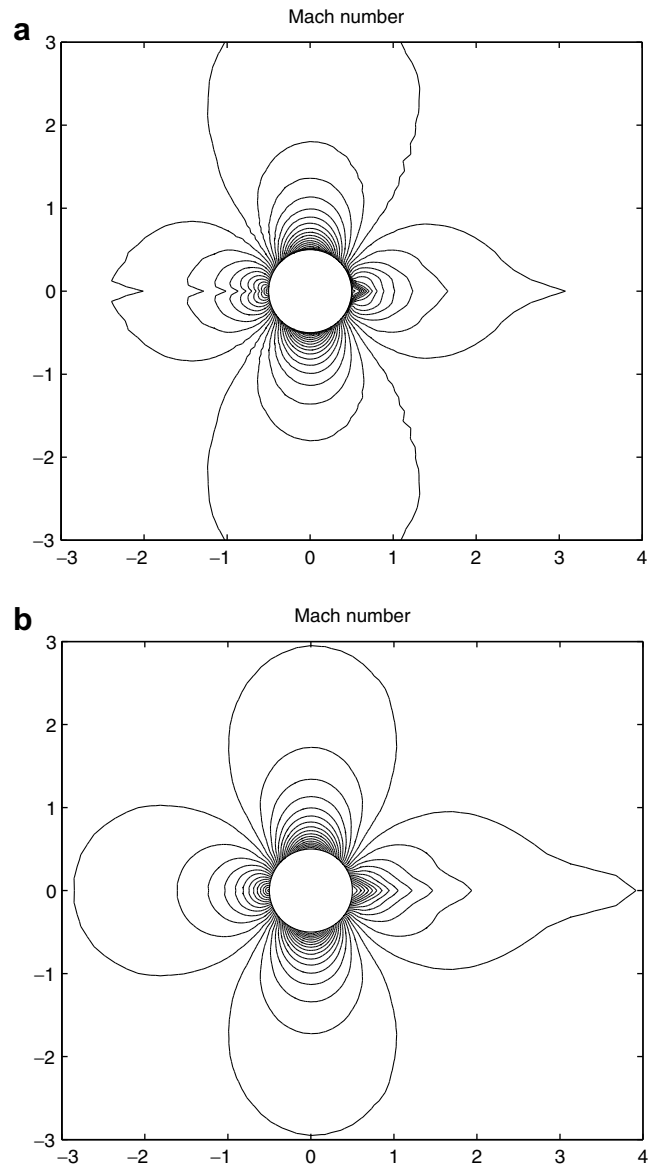
Fig. 12. Subsonic flow past a cylinder with $\mathcal{M}_\infty = 0.35$. Shown in these panels are isolines of the Mach number for the (a) N\LC and (b) RXN\LC schemes. Near the cylinder both methods produce comparable results. Away from the cylinder the grid resolution becomes coarser; and therefore, visible differences in the two methods appear. In these regions the RXN\LC scheme produces slightly more diffused contours than the N\LC scheme.

where ρ is the fluid density, $\vec{u} = (u^1, u^2)$ is the fluid velocity, $\mathcal{E}$ is the total energy, and p is the fluid pressure. The ideal gas law closes the system by relating the pressure to the other fluid variables:

$$\mathcal{E} = \frac{p}{\gamma - 1} + \frac{1}{2} \rho \|\vec{u}\|^2, \tag{127}$$

where γ is the ideal gas constant. In this example we take $\gamma = 1.4$.

The computational domain and numerical grid is shown in Fig. 5(a). In Fig. 5(b), we show a zoomed-in view of the numerical grid near the airfoil. The boundary conditions are such that subsonic flow with Mach number 0.85 enters from the left boundary at an angle of +1° from the horizontal axis. As the flow impinges on the airfoil, two supersonic bubbles are created above and below the airfoil. The supersonic flow is decelerated to the ambient subsonic flow through the creation of two shock waves (again, one above the airfoil and one below). This problem has been considered in several papers including [1,2].

Isolines of the Mach number and the pressure are shown in Fig. 6. We also plot in Fig. 7 the Mach number along the top and bottom edges of the airfoil. From this figure we note that the location of the shocks is in very good agreement between the two methods, while the location at the front of the airfoil where the solution goes from subsonic to supersonic is slightly different for each method. Furthermore, the RXN\LC-scheme is more diffusive than the N\LC-scheme, which results in slightly more entropy production near the airfoil for RXN\LC than N\LC. This can be seen both in Fig. 6, where we see bending of the Mach isolines near the airfoil, as well as in Fig. 8. Overall, however, both of these figures indicate remarkable agreement between the two solutions. In particular, the RXN\LC solution is far closer to the N\LC solution than the MUSCL-type scheme that was presented in [1]. We also note that the RXN\LC-scheme runs twice as fast as the N\LC-scheme.

Finally, in Fig. 9 we show the L_2-norm of the total residual as a function of time. We note that the fix of Abgrall [2] (see Section 3.5) is critically important in bringing both methods to convergence. Without this fix both methods stall at a total residual of only about 10^{-2}. We also find that the RXN\LC-scheme has a slightly better convergence rate than the N\LC-scheme.

5.3. Supersonic flow around a cylinder

Next we consider flow around a cylinder with Mach number $\mathcal{M}_\infty = 5$. The computational domain is $[-2,0] \times [-3,3]$ with a cylinder of unit radius centered at (0,0). In this example we found that we needed to run all the schemes at a CFL number of 0.4 in order to obtain well-converged results. The steady-state pressure on a grid with 5144 elements and 2656 nodes is shown in Fig. 10 for the following schemes: (a) the basic N-scheme, (b) the basic RXN-scheme, (c) the N\LC-scheme, and (d) the RXN\LC-scheme. The basic RXN-scheme solution is far more diffusive than the basic N-scheme, but once limiters and the convergence corrections are added, the N\LC and RXN\LC schemes produce comparable results. In fact, the RXN\LC-scheme converges faster than the N\LC-scheme, as can be seen in the convergence plot in Fig. 11. Finally, we note that the RXN\LC-scheme again runs about twice as fast as the N\LC-scheme.
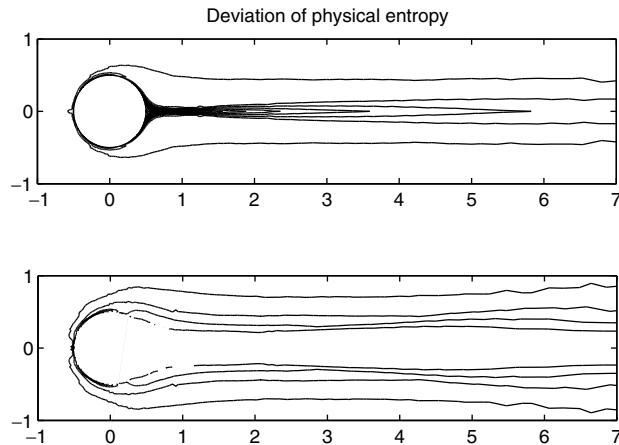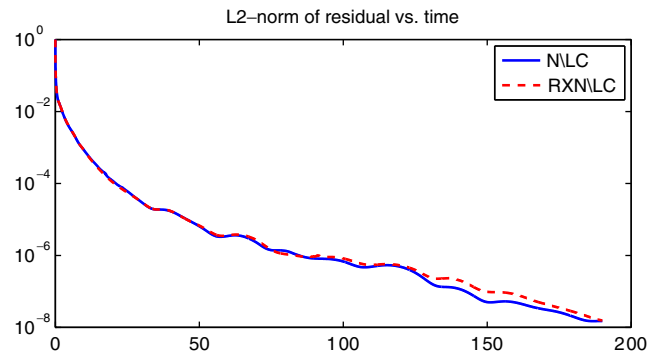
Fig. 14. L_2-norm of the total residual as a function of time for subsonic flow past a cylinder. Both methods give essentially the same convergence rates for this example.

5.4. Subsonic flow around a cylinder

Finally, we consider flow around a cylinder with Mach number $\mathcal{M}_\infty = 0.35$. This problem has been considered in several papers including [1,2]. The computational domain is $[-7,7] \times [-7,7]$ with a cylinder of radius 1/2 centered at (0,0). The steady-state Mach number on a grid with 12552 elements and 6404 nodes is shown in Fig. 12 for the (a) N\LC and (b) RXN\LC schemes. Near the cylinder both methods produce comparable results. Away from the cylinder the grid resolution becomes coarser; and therefore, visible differences in the two methods appear. In these regions the RXN\LC scheme produces slightly more diffused contours than the N\LC scheme.

Shown in Fig. 13 are the deviation of the physical entropy, $s = \log(p/\rho^\gamma)$, from the ambient entropy, $s_\infty = \log(1/\gamma^\gamma)$: $\Sigma = (s - s_\infty)/|s_\infty|$. Panel (a) is the N\LC-scheme and panel (b) is the RXN\LC-scheme. The minimum and maximum values of Σ for the N\LC and the RXN\LC schemes are $(-4.101 \times 10^{-3}, 4.324 \times 10^{-2})$ and $(-1.471 \times 10^{-3}, 1.291 \times 10^{-2})$, respectively. Each panel consists of 31 contour lines ranging from the minimum to the maximum Σ for each scheme. Therefore, these results show that the RXN\LC-scheme has a smaller entropy deviations, but that this error is more spread out behind the cylinder, while the N\LC-scheme has larger entropy deviations, but that this error is more concentrated near the x-axis.

The total residual as a function of time is shown in Fig. 14. Both methods give essentially the same convergence rates for this example. Finally, we note that the RXN\LC-scheme again runs about twice as fast as the N\LC-scheme.

6. Conclusions

In this work we have extended the results of LeVeque and Pelanti [21] to genuinely multidimensional residual distribution schemes. Specifically, we have shown that the N-scheme, both the scalar and the systems version, can be derived from a relaxation principle. Furthermore, using a genuinely multidimensional extension of the 1D local Lax-Friedrichs relaxation principle, we have derived a novel residual distribution scheme. The main benefit of this approach is that it does not require the inversion of an $m \times m$ matrix, where m is the number of conserved variables, at each time step in each grid element. The new method also does not require the use of Roe–Struijs–Deconinck averages. Using several examples of the 2D Euler equations from gas dynamics, including an example of transonic flow around the NACA 0012 airfoil, supersonic flow around a cylinder, and subsonic flow around a cylinder, we have compared the limited and corrected N-scheme (N\LC) with the newly proposed scheme (RXN\LC). These comparisons show that despite being computationally less expensive, the new method is capable of producing results comparable to those of the N\LC-scheme, although often with slightly more numerical diffusion. For more complicated equations such as magnetohydrodynamics or the general relativistic Einstein equations, we believe that the benefit of a simpler and computationally less expensive algorithm will far outweigh the slight increase in numerical dissipation. We will consider some of these more complicated systems in future work.

Finally, we would like to point out that the numerical code used in this work, including all of the numerical grids, will be made publicly available as part of the REDPACK software project. For more information see

http://www.math.wisc.edu/∼rossmani/software.html.

Acknowledgements

References

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

[2] R. Abgrall, Essentially non-oscillatory residual distribution schemes for hyperbolic problems, J. Comput. Phys. 214 (2006) 773–808.

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

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

[5] R. Abgrall, M. Mezine, Construction of second-order accurate monotone and stable residual distribution schemes for steady problems, J. Comput. Phys. 195 (2004) 474–507.

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

[7] F. Bouchut, Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources, Birkhäuser Verlag, 2005.

[8] G.Q. Chen, C.D. Levermore, T.P. Liu, Hyperbolic conservation-laws with stiff relaxation terms and entropy, Commun. Pure Appl. Math. 47 (1994) 787–830.

[9] G. Chesshire, W.D. Henshaw, Composite overlapping meshes for the solution of partial-differential equations, J. Comput. Phys. 90 (1990) 1–64.

[10] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.

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

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

[13] H. Deconinck, G. Struijs, G. Bourgeois, P. Roe, Compact advection schemes on unstructured meshes, in: VKI Lecture Series 1993–04, Computational Fluid Dynamics, 1993.

[14] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 35–61.

[15] C. Helzel, A high-resolution rotated grid method for conservation laws with embedded geometries, SIAM J. Sci. Comput. 26 (2005) 785–809.

[16] G.S. Jiang, C.-W. Shu, Efficient implementation of weighted eno schemes, J. Comput. Phys. 126 (1996) 202–228.

[17] S. Jin, Z.P. Xin, Relaxation schemes for systems of conservation-laws in arbitrary space dimensions, Commun. Pure Appl. Math. 48 (1995) 235–276.

[18] A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, J. Comput. Phys. 160 (2000) 241–282.

[19] R.J. LeVeque, Wave propagation algorithms for multi-dimensional hyperbolic systems, J. Comput. Phys. 131 (1997) 327–335.

[20] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[21] R.J. LeVeque, M. Pelanti, A class of approximate Riemann solvers and their relation to relaxation schemes, J. Comput. Phys. 172 (2001) 572–591.

[22] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[23] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.

[24] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.

[25] C.W. Schulzrinne, J.P. Collins, H.M. Glaz, Numerical-solution of the Riemann problem for 2-dimensional gas-dynamics, SIAM J. Sci. Comput. 14 (1993) 1394–1414.

[26] E. van der Weide, H. Deconinck, Positive matrix distribution schemes for hyperbolic systems, in: Comput. Fluid Dynam., Wiley, 1996, pp. 747–753.

[27] B. van Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.