

THE REGIONALLY IMPLICIT DISCONTINUOUS GALERKIN
METHOD: IMPROVING THE STABILITY OF DG-FEM*PIERSON T. GUTHREY[†] AND JAMES A. ROSSMANITH[‡]

Abstract. Discontinuous Galerkin (DG) methods for hyperbolic partial differential equations with explicit time-stepping schemes, such as strong stability-preserving Runge–Kutta, suffer from time-step restrictions that are significantly worse than what a simple Courant–Friedrichs–Lewy argument requires. In particular, the maximum stable time-step scales inversely with the highest degree in the DG polynomial approximation space and becomes progressively smaller with each added spatial dimension. In this work we introduce a novel approach that we have dubbed the regionally implicit discontinuous Galerkin (RIDG) method to overcome these small time-step restrictions. The RIDG method is based on an extension of the Lax–Wendroff DG method, which previously had been shown to be equivalent to a predictor–corrector approach, where the predictor is a locally implicit spacetime method (i.e., the predictor is something like a block–Jacobi update for a fully implicit spacetime DG method). The corrector is an explicit method that uses the spacetime reconstructed solution from the predictor step. In this work we modify the predictor to include not just local information, but also neighboring information. With this modification we show that the stability is greatly enhanced; in particular, we show that we are able to remove the polynomial degree dependence of the maximum time-step and show how this extends to multiple spatial dimensions. A semianalytic von Neumann analysis is presented to theoretically justify the stability claims. Convergence and efficiency studies for linear and nonlinear problems in multiple dimensions are accomplished using a MATLAB code that can be freely downloaded.

Key words. discontinuous Galerkin, hyperbolic conservation laws, Courant–Friedrichs–Lewy condition, time-stepping, numerical stability

AMS subject classifications. 65M12, 65M60, 35L03

DOI. 10.1137/17M1156174

1. Introduction. Hyperbolic conservation laws model phenomena characterized by waves propagating at finite speeds; examples include the shallow water (gravity waves), compressible Euler (sound waves), Maxwell (light waves), magnetohydrodynamic (magnetoacoustic and Alfvén waves), and Einstein (gravitational waves) equations. In recent years, the discontinuous Galerkin (DG) finite element method (FEM) has become a standard approach for solving hyperbolic conservation laws alongside other methods such as weighted essentially nonoscillatory schemes (e.g., see Shu [20]) and various finite volume methods (e.g., see LeVeque [14]). The DG method was first introduced by Reed and Hill [18] for neutron transport, and then fully developed for time-dependent hyperbolic conservation laws in a series of papers by Cockburn, Shu, and collaborators (see [3] and references therein for details). An important feature of DG methods is that they can, at least in principle, be made arbitrarily high-order in space by increasing the polynomial order in each element; and therefore, the DG method is an example of a spectral element method (e.g., see Chapter 7.5 of Karniadakis and Sherwin [11]).

*Received by the editors November 9, 2017; accepted for publication (in revised form) March 7, 2019; published electronically June 6, 2019.

<http://www.siam.org/journals/sinum/57-3/M115617.html>

Funding: The work of the author was partially supported by the National Science Foundation grant DMS-1620128.

[†]Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824 (piersonguthrey@gmail.com).

[‡]Department of Mathematics, Iowa State University, Ames, IA 50011 (rossmani@iastate.edu).

If DG is only used to discretize the spatial part of the underlying PDE, it remains to also introduce a temporal discretization. Many time-stepping methods are possible, including various explicit and implicit schemes. In general, one time-step of an explicit scheme is significantly cheaper than an implicit one; the trade-off is that implicit schemes usually allow for larger time-steps. In many applications involving hyperbolic conservation laws, however, it is necessary to resolve the fastest time scales, in which case explicit methods are more efficient and easier to implement than implicit ones.

An upper bound on the largest allowable time-step for explicit schemes is provided by the Courant–Friedrichs–Lewy (CFL) condition, which requires that the domain of dependence of the numerical discretization subsumes the domain of dependence of the continuous PDE [4]. For example, a one-dimensional (1D) hyperbolic PDE for which information propagates at a maximum wave speed of λ_{\max} , on a uniform mesh of elements of size $h = \Delta x$, and with a time-stepping method that updates the solution on the element T_i^h only using existing solution values from T_{i-1}^h , T_i^h , and T_{i+1}^h , has the following constraint on Δt :

$$(1) \quad \nu := \frac{\lambda_{\max} \Delta t}{\Delta x} \leq 1.$$

This has a clear physical interpretation: a wave that emanates from the boundaries of element i that is traveling at the maximum speed, λ_{\max} , is not allowed to propagate further than one element width. If we wanted to allow the wave to travel more than one element width, we would need to widen the numerical stencil.

The CFL condition as described above is a *necessary* condition for stability (and therefore convergence), but it is not *sufficient*. For high-order DG methods with explicit time-stepping, a fact that is well-known in the literature is that the actual maximum linearly stable value of the CFL number, $\nu = \lambda_{\max} \Delta t / \Delta x$, is significantly smaller than what the CFL condition predicts (see, for example, Liu et al. [16] and sections 4.7 and 4.8 of Hesthaven and Warburton [10]).

Two popular explicit time-stepping schemes for DG are strong-stability-preserving Runge–Kutta (SSP-RK) DG [7, 8] and Lax–Wendroff [6, 17]. SSP-RK time-steps are one-step multistage Runge–Kutta methods that can be written as convex combinations of forward Euler steps. Lax–Wendroff utilizes the Cauchy–Kovalevskaya [23] procedure to convert temporal derivatives into spatial derivatives; the name Lax–Wendroff is due to the paper of Lax and Wendroff [13]. In Table 1 we illustrate for both the SSP-RK and Lax–Wendroff DG methods the gap between the CFL condition, a necessary but not sufficient condition for stability, and the semianalytically computed maximum CFL number needed for linear stability. Shown are the methods with space and time order $k = 1, 2, 3, 4$. The SSP-RK DG numbers are reported from Liu et al. [16], while the Lax–Wendroff DG numbers are from von Neumann analysis done in this paper. Note that the maximum CFL number from the CFL condition for SSP-RK DG grows with k due to the fact that the number of Runge–Kutta stages grows with k ; and therefore, the numerical domain of dependence is increased. For both sets of methods, the clear trend is that the maximum CFL numbers are much smaller than what a simple CFL domain of dependence argument would dictate. In particular, the relationship between the maximum CFL number and the order of the method is roughly $\nu_{\max} \propto 1/k$.

The goal of this paper is to develop an alternative time discretization for DG that allows for a linearly stable time-step that is closer to what is predicted by the CFL condition. The starting point of this work is the interpretation of the Lax–Wendroff DG method developed by Gassner et al. [6], where it was shown that Lax–Wendroff

TABLE 1

Shown here are the maximum CFL numbers for the SSP-RK DG and Lax-Wendroff DG methods with the same time and space order of accuracy. The line labeled “CFL cond.” is the upper bound of the CFL number as predicted just by looking at the domain of dependence of the numerical method. The line labeled “Neumann” is the numerically calculated maximum linearly stable CFL number. The SSP-RK DG numbers are reported from Liu et al. [16], while the Lax-Wendroff DG numbers are from von Neumann analysis done in this paper. For both sets of methods, the clear trend is that the maximum linearly stable CFL numbers are much smaller than what a simple CFL domain of dependence argument would dictate.

	SSP-RK(k) with P^{k-1} DG				P^{k-1} Lax-Wendroff DG			
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
CFL cond.	1.00	2.00	3.00	4.00	1.000	1.000	1.000	1.000
Neumann	1.00	0.33	0.13	0.10	1.000	0.333	0.171	0.104

DG can be formulated as a predictor-corrector method. The predictor is a local version of a spacetime DG method [12, 22] (i.e., the predictor is something like a block-Jacobi update for a fully implicit spacetime DG method), and the corrector is an explicit method that uses the spacetime reconstructed solution from the predictor step. In this work we modify the predictor to include not just local information, but also neighboring information. The name that we are giving to this new approach is the *regionally implicit* discontinuous Galerkin (RIDG) scheme, which contrasts with the *locally implicit* (LIDG) formulation of the Lax-Wendroff DG scheme developed by Gassner et al. [6]. In this new formulation, we are able to achieve all of the following:

- develop RIDG schemes for 1D, 2D, and 3D advection;
- show that RIDG has larger maximum CFL numbers than explicit SSP-RK and Lax-Wendroff DG;
- show that the maximum linearly stable CFL number is bounded below by a constant that is independent of the polynomial order;
- demonstrate experimentally the correct convergence rates on 1D, 2D, and 3D advection examples;
- demonstrate experimentally the correct convergence rates on 1D and 2D nonlinear examples.

All of the methods described in this work are written in a MATLAB code that can be freely downloaded [9].

The organization of this paper is as follows. In section 2 we briefly review how space is discretized in the DG method. In section 3 we review the Lax-Wendroff DG scheme then develop the 1D version of the proposed RIDG scheme and carry out von Neumann stability analysis for both methods. The generalization to multiple dimensions is done in section 4. In section 5 we carry out numerical convergence tests to validate the new approach and to quantify the computational efficiency of RIDG relative to the Lax-Wendroff method. Finally, in section 6 we show how to extend the method to a nonlinear scalar problem: the 1D and 2D Burgers equation. In this case we compare the efficiency and accuracy of our method against the fourth-order Runge-Kutta discontinuous Galerkin (RKDG) scheme.

2. DG-FEM spatial discretization. Consider hyperbolic conservation laws of the form

$$(2) \quad \underline{q}_{,t} + \nabla \cdot \underline{\underline{F}}(\underline{q}) = 0,$$

where $\underline{q}(t, \underline{x}) : \mathbb{R}^+ \times \mathbb{R}^{M_{\text{dim}}} \mapsto \mathbb{R}^{M_{\text{eqn}}}$ is the vector of conserved variables, $\underline{\underline{F}}(\underline{q}) :$

$\mathbb{R}^{M_{\text{eqn}}} \mapsto \mathbb{R}^{M_{\text{eqn}} \times M_{\text{dim}}}$ is the flux function, M_{dim} is the number of spatial dimensions, and M_{eqn} is the number of conserved variables. We assume that the system is hyperbolic, which means that the flux Jacobian

$$(3) \quad \underline{\underline{A}}(\underline{q}; \underline{n}) = \frac{\partial (\underline{n} \cdot \underline{\underline{F}})}{\partial \underline{q}}$$

for all $\underline{q} \in \mathcal{S} \subset \mathbb{R}^{M_{\text{eqn}}}$, where \mathcal{S} is some physically meaningful convex subset of $\mathbb{R}^{M_{\text{eqn}}}$, and for all directions $\underline{n} \in \mathbb{R}^{M_{\text{dim}}}$ such that $\|\underline{n}\| = 1$, must be diagonalizable with only real eigenvalues (e.g., see Chapter 18 of LeVeque [14]).

Next consider discretizing system (2) in space via the DG method, which was first introduced by Reed and Hill [18] for neutron transport, and then fully developed for time-dependent hyperbolic conservation laws in a series of papers by Bernardo Cockburn, Chi-Wang Shu, and collaborators (see [3] and references therein for details). We define $\Omega \subset \mathbb{R}^{M_{\text{dim}}}$ to be a polygonal domain with boundary $\partial\Omega$, and discretize Ω using a finite set of nonoverlapping elements, \mathcal{T}_i , such that $\cup_{i=1}^{M_{\text{elem}}} \mathcal{T}_i = \Omega$, where M_{elem} is the total number of elements. Let $\mathbb{P}(M_{\text{deg}}, M_{\text{dim}})$ denote the set of polynomials from $\mathbb{R}^{M_{\text{dim}}}$ to \mathbb{R} with maximal polynomial degree M_{deg} .¹ On the mesh of M_{elem} elements we define the *broken* finite element space:

$$(4) \quad \mathcal{W}^h := \left\{ \underline{w}^h \in [L^\infty(\Omega)]^{M_{\text{eqn}}} : \underline{w}^h|_{\mathcal{T}_i} \in [\mathbb{P}(M_{\text{deg}}, M_{\text{dim}})]^{M_{\text{eqn}}} \quad \forall \mathcal{T}_i \right\},$$

where h is the grid spacing, M_{dim} is the number of spatial dimensions, M_{eqn} is the number of conserved variables, and M_{deg} is the maximal polynomial degree in the finite element representation. The above expression means that $\underline{w} \in \mathcal{W}^h$ has M_{eqn} components, each of which when restricted to some element \mathcal{T}_i is a polynomial in $\mathbb{P}(M_{\text{deg}}, M_{\text{dim}})$, and no continuity is assumed across element faces.

Let $\varphi_k(\underline{x})$ for $k = 1, \dots, M_{\text{basis}}$ be an appropriate basis that spans $\mathbb{P}(M_{\text{deg}}, M_{\text{dim}})$ over \mathcal{T}_i (e.g., Legendre or Lagrange polynomials). In order to get the DG semidiscretization, we multiply (2) by $\varphi_k \in \mathbb{P}(M_{\text{deg}}, M_{\text{dim}})$, integrate over the element \mathcal{T}_i , use integration-by-parts in space, and replace the true solution, \underline{q} , by the following ansatz:

$$(5) \quad \underline{q}^h(t, \underline{x}) \Big|_{\mathcal{T}_i} = \sum_{\ell=1}^{M_{\text{basis}}} \underline{Q}_i^\ell(t) \varphi_\ell(\underline{x}).$$

All of these steps result in the following semidiscrete system:

$$(6) \quad \sum_{\ell=1}^{M_{\text{basis}}} \left[\int_{\mathcal{T}_i} \varphi_k \varphi_\ell \, d\underline{x} \right] \frac{d \underline{Q}_i^\ell}{dt} = \int_{\mathcal{T}_i} \underline{\underline{F}}(\underline{q}^h) \cdot \nabla \varphi_k \, d\underline{x} - \oint_{\partial \mathcal{T}_i} \varphi_k \underline{\mathcal{F}}(\underline{q}_+^h, \underline{q}_-^h; \underline{n}) \, ds,$$

where \underline{n} is an outward-pointing normal vector to $\partial \mathcal{T}_i$, \underline{q}_+^h and \underline{q}_-^h are the states on either side of the boundary $\partial \mathcal{T}_i$, and $\underline{\mathcal{F}}$ is the numerical flux, which must satisfy the following two conditions:

¹In one dimension (i.e., $M_{\text{dim}} = 1$), this definition is unambiguous. In higher dimensions, $\mathbb{P}(M_{\text{deg}}, M_{\text{dim}})$ could refer the set of polynomials that have a total degree $\leq M_{\text{deg}}$ (we refer to this as the $\mathcal{P}(M_{\text{deg}}, M_{\text{dim}})$ basis), it could refer to the set of polynomials that have degree $\leq M_{\text{deg}}$ in each independent variable (we refer to this as the $\mathcal{Q}(M_{\text{deg}}, M_{\text{dim}})$ basis), or it could be something in between.

- consistency: $\underline{\mathcal{F}}(\underline{q}, \underline{q}; \underline{n}) = \underline{\mathcal{F}}(\underline{q}) \cdot \underline{n}$;
- conservation: $\underline{\mathcal{F}}(\underline{q}_-^h, \underline{q}_+^h; \underline{n}) = -\underline{\mathcal{F}}(\underline{q}_+^h, \underline{q}_-^h; -\underline{n})$.

Equation (6) represents a large system of coupled ordinary differential equations in time.

3. RIDG in one space dimension. We present in this section the proposed RIDG method as applied to a 1D advection equation. Each RIDG time-step is comprised of two key steps: a predictor and a corrector. The predictor is a truncated version of an implicit spacetime DG approximation, which is not consistent, at least by itself, with the PDE that it endeavors to approximate. The corrector is a modified forward Euler step that makes use of the predicted solution; this step restores consistency, and indeed, high-order accuracy, with the underlying PDE.

In the subsections below we begin with a brief description of the advection equation in subsection 3.1. We then review the Lax–Wendroff (aka locally implicit) prediction step in subsection 3.2, which provides the motivation for RIDG. The RIDG prediction step is developed in subsection 3.3. The correction step for both predictors is detailed in subsection 3.4. Finally, we carry out semianalytic von Neumann analysis for both schemes in subsection 3.5 and demonstrate the improved stability of RIDG over Lax–Wendroff DG.

3.1. 1D advection equation. We consider here the 1D advection equation for $(t, x) \in [0, T] \times \Omega$, along with some appropriate set of boundary conditions:

$$(7) \quad q_{,t} + u q_{,x} = 0.$$

Next, we introduce a uniform Cartesian spacetime mesh with spacetime elements,

$$(8) \quad \mathcal{S}_i^{n+1/2} = [t^n, t^n + \Delta t] \times [x_i - \Delta x/2, x_i + \Delta x/2],$$

which can be written in local coordinates, $[\tau, \xi] \in [-1, 1]^2$, where

$$(9) \quad t = t^{n+1/2} + \tau (\Delta t/2) \quad \text{and} \quad x = x_i + \xi (\Delta x/2).$$

In these local coordinates the advection equation (7) becomes

$$(10) \quad q_{,\tau} + \nu q_{,\xi} = 0, \quad \text{where} \quad \nu = \frac{u \Delta t}{\Delta x},$$

and $|\nu|$ is the CFL number.

3.2. Lax–Wendroff DG (aka LIDG) prediction step. We review here the prediction step for the LIDG scheme as formulated by Gassner et al. [6]. In order to contrast with the proposed RIDG method, we will refer to this method as the LIDG method.

We fix the largest polynomial degree to M_{\deg} in order to eventually achieve an approximation that has an order of accuracy $\mathcal{O}(\Delta x^{M_{\deg}+1} + \Delta t^{M_{\deg}+1})$. At the old time, $t = t^n$, we are given the following approximate solution on each space element, $\mathcal{T}_i = [x_i - \Delta x/2, x_i + \Delta x/2]$:

$$(11) \quad q(t^n, x) \Big|_{\mathcal{T}_i} \approx q_i^n := \underline{\Phi}^T \underline{Q}_i^n,$$

where $\underline{Q}_i^n \in \mathbb{R}^{M_C}$, $\underline{\Phi} \in \mathbb{R}^{M_C}$, $M_C := M_{\deg} + 1$, and

$$(12) \quad \underline{\Phi} = \left(1, \sqrt{3}\xi, \frac{\sqrt{5}}{2} (3\xi^2 - 1), \dots \right) \quad \text{s.t.} \quad \frac{1}{2} \int_{-1}^1 \underline{\Phi} \underline{\Phi}^T d\xi = \underline{\mathbb{I}} \in \mathbb{R}^{M_C \times M_C}$$

are the orthonormal space Legendre polynomials.

In order to compute a predicted solution on each spacetime element (8) we make the following ansatz:

$$(13) \quad q(t, x) \Big|_{\mathcal{S}_i^{n+1/2}} \approx w_i^{n+1/2} := \underline{\Psi}^T \underline{W}_i^{n+1/2},$$

where $\underline{W}_i^{n+1/2} \in \mathbb{R}^{M_P}$, $\underline{\Psi} \in \mathbb{R}^{M_P}$, $M_P := (M_{\text{deg}} + 1)(M_{\text{deg}} + 2)/2$, and

$$(14) \quad \underline{\Psi} = (1, \sqrt{3}\tau, \sqrt{3}\xi, \dots) \quad \text{s.t.} \quad \frac{1}{4} \int_{-1}^1 \int_{-1}^1 \underline{\Psi} \underline{\Psi}^T d\tau d\xi = \underline{\underline{I}} \in \mathbb{R}^{M_P \times M_P}$$

are the spacetime Legendre basis functions. Next, we premultiply (10) by $\underline{\Psi}$ and integrate over $\mathcal{S}_i^{n+1/2}$ to obtain

$$(15) \quad \frac{1}{4} \int_{-1}^1 \int_{-1}^1 \underline{\Psi} [q_{,\tau} + \nu q_{,\xi}] d\tau d\xi = \underline{0}.$$

We then replace the exact solution, q , by (13). We integrate-by-parts in time, first forward, then backward, which introduces a jump term at the old time $t = t^n$. No integration-by-parts is done in space—this is what gives the local nature of the predictor step. All of this results in the following equation:

$$(16) \quad \iint \underline{\Psi} [\underline{\Psi}_{,\tau} + \nu \underline{\Psi}_{,\xi}]^T \underline{W}_i^{n+1/2} d\tau d\xi + \int \underline{\Psi}|_{\tau=-1} \left[\underline{\Psi}_{|\tau=-1}^T \underline{W}_i^{n+1/2} - \underline{\Phi}^T \underline{Q}_i^n \right] d\xi = \underline{0},$$

where all 1D integrals are over $[-1, 1]$, which can be written as

$$(17) \quad \underline{\underline{L}}^0 \underline{W}_i^{n+1/2} = \underline{\underline{T}} \underline{Q}_i^n,$$

$$(18) \quad \underline{\underline{L}}^0 = \frac{1}{4} \int_{-1}^1 \int_{-1}^1 \underline{\Psi} [\underline{\Psi}_{,\tau} + \nu \underline{\Psi}_{,\xi}]^T d\tau d\xi + \frac{1}{4} \int_{-1}^1 \underline{\Psi}|_{\tau=-1} \underline{\Psi}_{|\tau=-1}^T d\xi \in \mathbb{R}^{M_P \times M_P},$$

$$(19) \quad \underline{\underline{T}} = \frac{1}{4} \int_{-1}^1 \underline{\Psi}|_{\tau=-1} \underline{\Phi}^T d\xi \in \mathbb{R}^{M_P \times M_C}.$$

As is evident from the formulas above, the predicted spacetime solution as encoded in the coefficients $\underline{W}_i^{n+1/2}$ is entirely local—the values only depend on the old values from the same element: \underline{Q}_i^n . Therefore, we refer to this prediction step as *locally implicit*.

Remark 1. Gassner et al. [6] argued that the locally implicit prediction step as presented above produces the key step in the Lax–Wendroff DG scheme [17]. We briefly illustrate this point here.

Lax–Wendroff [13] (aka the Cauchy–Kovalevskaya [23] procedure) begins with a Taylor series in time. All time derivatives are then replaced by spatial derivatives using the underlying PDE—in this case (7). For example, if we kept all the time

derivatives up to the third derivative we would get

$$(20) \quad \begin{aligned} q^{n+1} &\approx q^n + 2q_{,\tau}^n + 2q_{,\tau,\tau}^n + \frac{4}{3}q_{,\tau,\tau,\tau}^n = q^n - 2\nu q_{,\xi}^n + 2\nu^2 q_{,\xi,\xi}^n - \frac{4}{3}\nu^3 q_{,\xi,\xi,\xi}^n \\ &= q^n - 2\nu \left\{ q^n - \nu q_{,\xi}^n + \frac{2}{3}\nu^2 q_{,\xi,\xi}^n \right\}_{,\xi} = q^n - 2\nu \mathcal{G}_{,\xi}^{\text{LxW}}. \end{aligned}$$

Using the third-order DG approximation,

$$(21) \quad q^n := \varphi_1 Q_1^n + \varphi_2 Q_2^n + \varphi_3 Q_3^n = Q_1^n + \sqrt{3}\xi Q_2^n + \frac{\sqrt{5}}{2}(3\xi^2 - 1)Q_3^n,$$

we obtain

$$(22) \quad \mathcal{G}^{\text{LxW}} = \varphi_1 \left(Q_1^n - \sqrt{3}\nu Q_2^n + 2\sqrt{5}\nu^2 Q_3^n \right) + \varphi_2 \left(Q_2^n - \sqrt{15}\nu Q_3^n \right) + \varphi_3 Q_3^n.$$

Alternatively, the time-averaged flux, \mathcal{G}^{LxW} , can be directly obtained from the locally implicit predictor described above. We first calculate the predicted spacetime solution, $w^{n+1/2}$, via (17), (18), and (19). From this we compute the time averaged flux:

$$(23) \quad \mathcal{G}^{\text{LxW}} = \frac{1}{2} \int_{-1}^1 w^{n+1/2}(\tau, \xi) d\tau = \frac{1}{2} \left\{ \int_{-1}^1 \underline{\Psi}(\tau, \xi) d\tau \right\}^T (\underline{\underline{L}}^0)^{-1} \underline{\underline{T}} \underline{\underline{Q}}^n.$$

A straightforward calculation shows that (23) with (21) and the corresponding $\mathcal{P}(2, 2)$ spacetime basis² is exactly the same as (22).

3.3. RIDG prediction step. As we have argued in section 1 and with Table 1, the locally implicit predictor described above in subsection 3.2 will result in a scheme that has a maximum linearly stable CFL number that is both small and becomes progressively smaller with increasing order of accuracy. We introduce a modified prediction step here to remedy these shortcomings.

The starting point is the same as in subsection 3.2, ansatz (11) and (13), but now with the full spacetime $\mathcal{Q}(M_{\text{deg}}, M_{\text{dim}} + 1)$ basis in the prediction step:

$$(24) \quad M_C = M_{\text{deg}} + 1 \quad \text{and} \quad M_P = (M_{\text{deg}} + 1)^2.$$

We integrate the advection equation in spacetime to get (15), but this time we integrate-by-parts in both space and time, which yields

$$(25) \quad \begin{aligned} &\iint \underline{\Psi} (\underline{\Psi}_{,\tau} + \nu \underline{\Psi}_{,\xi})^T \underline{W}_i^{n+1/2} d\tau d\xi + \int \underline{\Psi}_{|\tau=-1} \left[\underline{\Psi}_{|\tau=-1}^T \underline{W}_i^{n+1/2} - \underline{\Phi}^T \underline{Q}_i^n \right] d\xi - \\ &\int \left\{ \underline{\Psi}_{|\xi=1} \left[\nu \underline{\Psi}_{|\xi=1}^T \underline{W}_i^{n+1/2} - \mathcal{F}_{i+\frac{1}{2}}^* \right] - \underline{\Psi}_{|\xi=-1} \left[\nu \underline{\Psi}_{|\xi=-1}^T \underline{W}_i^{n+1/2} - \mathcal{F}_{i-\frac{1}{2}}^* \right] \right\} d\tau = 0, \end{aligned}$$

where all 1D integrals are over $[-1, 1]$, $\underline{\Phi}$ is the Legendre basis (12), $\underline{\Psi}$ is the spacetime Legendre basis (14), and \mathcal{F}^* are some appropriately defined numerical fluxes.

²Actually, for LIDG the result is the same whether we use $\mathcal{P}(M_{\text{deg}}, M_{\text{dim}} + 1)$, $\mathcal{Q}(M_{\text{deg}}, M_{\text{dim}} + 1)$, or something in between.

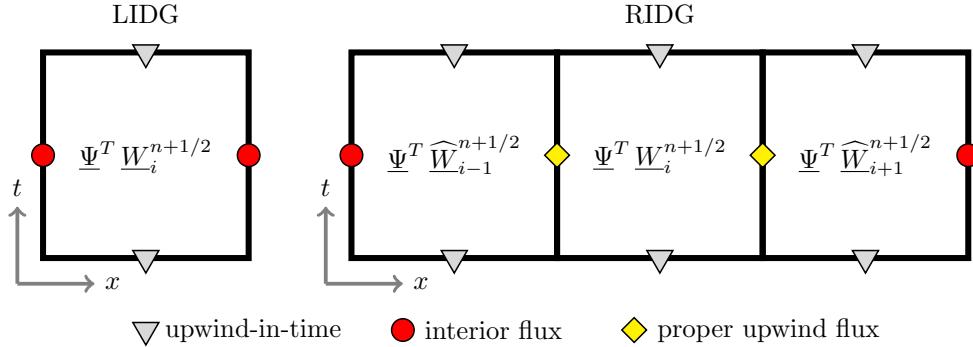


FIG. 1. Shown are the domains of dependence for the LIDG (left) and RIDG (right) prediction steps for spacetime element $S_i^{n+1/2}$ in one spatial dimension. Note that on the $t = t^n$ and $t = t^{n+1} = t^n + \Delta t$ faces, the “proper upwind flux” values are always on the “past” side of the face—we refer to these as the “upwind-in-time” values. The LIDG prediction step is purely local in space—there is no spatial communication with neighboring cells. The RIDG prediction step computes the proper upwind flux on the $x = x_i \pm \Delta x/2$ faces, but nowhere else. In the RIDG prediction step the states to the immediate left and right of element i are only temporary variables and will be discarded once the predicted solution on element i has been computed—to make note of this we place hats over the temporary variables.

The crux of the idea of the RIDG scheme in one spatial dimension can be summarized as follows:

- We define a *region* to be the current spacetime element, $S_i^{n+1/2}$, and its immediate neighbors: $S_{i-1}^{n+1/2}$ and $S_{i+1}^{n+1/2}$. This is illustrated in Figure 1.
- For $S_i^{n+1/2}$, we use the correct upwind fluxes to define the numerical fluxes, \mathcal{F}^* , on its faces.
- For the immediate neighbors, $S_{i-1}^{n+1/2}$ and $S_{i+1}^{n+1/2}$, we again use the correct upwind fluxes on the faces that are shared with $S_i^{n+1/2}$, but on the outer faces we use one-sided interior fluxes. See Figure 1.
- We use the $\mathcal{Q}(M_{\text{deg}}, M_{\text{dim}} + 1)$ spacetime basis in the prediction step (i.e., the full tensor product spacetime basis). Numerical experimentation showed us that using the $\mathcal{Q}(M_{\text{deg}}, M_{\text{dim}} + 1)$ basis for the prediction step, rather than the $\mathcal{P}(M_{\text{deg}}, M_{\text{dim}} + 1)$ basis, produces significantly more accurate results; in the case of linear equations, this creates little additional computational expense since all the relevant matrices can be precomputed.

The result of this is a collection of three elements with solutions that are coupled to each other, but that are completely decoupled from all remaining elements. This RIDG setup is depicted in Figure 1, where we also show the LIDG setup as a point of comparison.

The precise form of the fluxes for the RIDG prediction step on spacetime element $S_i^{n+1/2}$ can be written as follows:

$$(26) \quad \mathcal{F}_{i-3/2}^* = \nu \underline{\Psi}_{|\xi=-1}^T \underline{W}_{i-1}^{n+1/2}, \quad \mathcal{F}_{i-1/2}^* = \nu^+ \underline{\Psi}_{|\xi=1}^T \underline{W}_{i-1}^{n+1/2} + \nu^- \underline{\Psi}_{|\xi=-1}^T \underline{W}_i^{n+1/2},$$

$$(27) \quad \mathcal{F}_{i+1/2}^* = \nu^+ \underline{\Psi}_{|\xi=1}^T \underline{W}_i^{n+1/2} + \nu^- \underline{\Psi}_{|\xi=-1}^T \underline{W}_{i+1}^{n+1/2}, \quad \mathcal{F}_{i+3/2}^* = \nu \underline{\Psi}_{|\xi=1}^T \underline{W}_{i+1}^{n+1/2}.$$

Combining (25) with numerical fluxes (26) and (27) yields the following block 3×3

system:

$$(28) \quad \begin{bmatrix} \underline{\underline{L}}^0 + \underline{\underline{L}}^- & \underline{\underline{X}}^- & \\ \underline{\underline{X}}^+ & \underline{\underline{L}}^0 + \underline{\underline{L}}^- + \underline{\underline{L}}^+ & \underline{\underline{X}}^- \\ & \underline{\underline{X}}^+ & \underline{\underline{L}}^0 + \underline{\underline{L}}^+ \end{bmatrix} \begin{bmatrix} \widehat{W}_{i-1}^{n+1/2} \\ W_i^{n+1/2} \\ \widehat{W}_{i+1}^{n+1/2} \end{bmatrix} = \begin{bmatrix} \underline{T} Q_{i-1}^n \\ \underline{T} Q_i^n \\ \underline{T} Q_{i+1}^n \end{bmatrix},$$

where $\underline{\underline{L}}^0$ is given by (18), \underline{T} is given by (19), and

$$(29) \quad \underline{\underline{L}}^+ = \frac{\nu^+}{4} \int_{-1}^1 \underline{\Psi}_{|\xi=-1} \underline{\Psi}_{|\xi=-1}^T d\tau, \quad \underline{\underline{L}}^- = -\frac{\nu^-}{4} \int_{-1}^1 \underline{\Psi}_{|\xi=1} \underline{\Psi}_{|\xi=1}^T d\tau,$$

$$(30) \quad \underline{\underline{X}}^+ = -\frac{\nu^+}{4} \int_{-1}^1 \underline{\Psi}_{|\xi=-1} \underline{\Psi}_{|\xi=1}^T d\tau, \quad \underline{\underline{X}}^- = \frac{\nu^-}{4} \int_{-1}^1 \underline{\Psi}_{|\xi=1} \underline{\Psi}_{|\xi=-1}^T d\tau,$$

where $\underline{\underline{L}}^-, \underline{\underline{L}}^+, \underline{\underline{X}}^-, \underline{\underline{X}}^+ \in \mathbb{R}^{M_P \times M_P}$, $\nu^+ = \max(\nu, 0)$, and $\nu^- = \min(\nu, 0)$. Note that the states to the immediate left and right of the current spacetime element $S_i^{n+1/2}$ are only temporary variables and will be discarded once the predicted solution in element i has been computed—to make note of this we place hats over the temporary variables. This also means that we have to solve a block 3×3 system of the form (28) on every single element $S_i^{n+1/2}$.

3.4. Correction step for both LIDG and RIDG. In order to go from the predictor to the corrector step, we multiply (10) by $\underline{\Phi} \in \mathbb{R}^{M_C}$ and integrate in space-time:

$$(31) \quad \underline{Q}_i^{n+1} = \underline{Q}_i^n + \frac{\nu}{2} \int_{-1}^1 \int_{-1}^1 \underline{\Phi}_{,\xi} q d\tau d\xi - \frac{1}{2} \int_{-1}^1 \left[\underline{\Phi}_{|\xi=1} \mathcal{F}_{i+1/2} - \underline{\Phi}_{|\xi=-1} \mathcal{F}_{i-1/2} \right] d\tau,$$

where \mathcal{F} is the numerical flux. Next we replace q by the predicted solution from either subsection 3.2 or 3.3 and use the upwind flux,

$$(32) \quad \mathcal{F}_{i-1/2} = \nu^+ \underline{\Psi}_{|\xi=1}^T \underline{W}_{i-1}^{n+1/2} + \nu^- \underline{\Psi}_{|\xi=-1}^T \underline{W}_i^{n+1/2},$$

which results in

$$(33) \quad \underline{Q}_i^{n+1} = \underline{Q}_i^n + \underline{\underline{C}}^- \underline{W}_{i-1}^{n+1/2} + \underline{\underline{C}}^0 \underline{W}_i^{n+1/2} + \underline{\underline{C}}^+ \underline{W}_{i+1}^{n+1/2},$$

$$(34) \quad \underline{\underline{C}}^0 = \frac{\nu}{2} \int_{-1}^1 \int_{-1}^1 \underline{\Phi}_{,\xi} \underline{\Psi}^T d\tau d\xi - \frac{1}{2} \int_{-1}^1 \left[\nu^+ \underline{\Phi}_{|\xi=1} \underline{\Psi}_{|\xi=1}^T - \nu^- \underline{\Phi}_{|\xi=-1} \underline{\Psi}_{|\xi=-1}^T \right] d\tau,$$

$$(35) \quad \underline{\underline{C}}^- = \frac{\nu^+}{2} \int_{-1}^1 \underline{\Phi}_{|\xi=-1} \underline{\Psi}_{|\xi=1}^T d\tau, \quad \underline{\underline{C}}^+ = -\frac{\nu^-}{2} \int_{-1}^1 \underline{\Phi}_{|\xi=1} \underline{\Psi}_{|\xi=-1}^T d\tau,$$

where $\underline{\underline{C}}^0, \underline{\underline{C}}^-, \underline{\underline{C}}^+ \in \mathbb{R}^{M_C \times M_P}$.

3.5. Von Neumann stability analysis for both LIDG and RIDG. The Lax–Wendroff DG scheme (aka LIDG) with the prediction step detailed in subsection 3.2 and the correction step given by (33), (34), and (35) uses a stencil involving three elements: \mathcal{T}_{i-1} , \mathcal{T}_i , and \mathcal{T}_{i+1} ; the RIDG scheme detailed in subsection 3.3 with the same correction step as LIDG uses a stencil involving five elements: \mathcal{T}_{i-2} , \mathcal{T}_{i-1} , \mathcal{T}_i , \mathcal{T}_{i+1} , and \mathcal{T}_{i+2} . This resulting CFL conditions for the LIDG and RIDG schemes are

$$(36) \quad |\nu| = \frac{|u|\Delta t}{\Delta x} \leq 1 \quad \text{and} \quad |\nu| = \frac{|u|\Delta t}{\Delta x} \leq 2,$$

repectively. In reality, the CFL number, $|\nu|$, for which linear stability is achieved is smaller than what this CFL argument provides; we investigate this in more detail here.

In order to study linear stability, we employ the technique of von Neumann stability analysis (e.g., see Chapter 10.5 of LeVeque [15]). In particular, we assume the following Fourier ansatz:

$$(37) \quad \underline{Q}_i^{n+1} = \tilde{\underline{Q}}^{n+1} e^{I\omega i} \quad \text{and} \quad \underline{Q}_i^n = \tilde{\underline{Q}}^n e^{I\omega i},$$

where $I = \sqrt{-1}$ and $0 \leq \omega \leq 2\pi$ is the wave number. After using this ansatz, the next step is to write the resulting update in the form

$$(38) \quad \tilde{\underline{Q}}^{n+1} = \underline{\underline{M}}(\nu, \omega) \tilde{\underline{Q}}^n$$

for some matrix $\underline{\underline{M}} \in \mathbb{R}^{M_C \times M_C}$. If we apply ansatz (37) to LIDG and RIDG, assuming without loss of generality that $\nu \geq 0$, we obtain the following:

$$(39) \quad \begin{aligned} \underline{\underline{M}}_{\text{LIDG}}(\nu, \omega) &= \left(\underline{\underline{I}} + \underline{\underline{C}}^0 (\underline{\underline{L}}^0)^{-1} \underline{\underline{T}} \right) + e^{-I\omega} \underline{\underline{C}}^- (\underline{\underline{L}}^0)^{-1} \underline{\underline{T}}, \\ \underline{\underline{M}}_{\text{RIDG}}(\nu, \omega) &= \left(\underline{\underline{I}} + \underline{\underline{C}}^0 (\underline{\underline{L}}^0 + \underline{\underline{L}}^+)^{-1} \underline{\underline{T}} \right) \\ (40) \quad &\quad + e^{-I\omega} \left(\underline{\underline{C}}^- (\underline{\underline{L}}^0 + \underline{\underline{L}}^+)^{-1} \underline{\underline{T}} - \underline{\underline{C}}^0 (\underline{\underline{L}}^0 + \underline{\underline{L}}^+)^{-1} \underline{\underline{X}}^+ (\underline{\underline{L}}^0)^{-1} \underline{\underline{T}} \right) \\ &\quad - e^{-2I\omega} \underline{\underline{C}}^- (\underline{\underline{L}}^0 + \underline{\underline{L}}^+)^{-1} \underline{\underline{X}}^+ (\underline{\underline{L}}^0)^{-1} \underline{\underline{T}}, \end{aligned}$$

where $\underline{\underline{I}} \in \mathbb{R}^{M_C \times M_C}$ is the identity matrix.

The final step in the stability analysis is to study the spectral properties $\underline{\underline{M}}$ as a function of the CFL number ν . In particular, to find the largest ν for which LIDG or RIDG are linearly stable, we define the following function:

$$(41) \quad f(\nu) := \max_{0 \leq \omega \leq 2\pi} \rho(\underline{\underline{M}}(\nu, \omega)) - 1,$$

where $\rho(\underline{\underline{M}})$ is the spectral radius of $\underline{\underline{M}}$. For both LIDG and RIDG, the function $f(\nu)$ satisfies $f(0) = 0$, and there exists a finite range of ν for which $f(\nu) \approx 0$, and there exists a value of ν for which $f(\nu)$ transitions from being approximately zero to rapidly increasing with increasing ν . We illustrate this point in Figure 2 for both LIDG and RIDG for the case $M_{\text{deg}} = 5$ ($M_C = 6$, LIDG: $M_P = 21$, RIDG: $M_P = 36$); for each scheme we note approximately where the linear stability transition occurs. In order to numerically estimate the location of the linear stability transition we look for the value of ν that satisfies $f(\nu) = \varepsilon$. We do this via a simple bisection method where we set $\varepsilon = 0.0005$ and we replace the true maximization in (41) over the maximization of 2001 uniformly spaced wave numbers over $0 \leq \omega \leq 2\pi$.

The result of this bisection procedure for both LIDG and RIDG for $M_{\text{deg}} = 0, 1, 2, 3, 4, 5$ is summarized in Table 2. In all cases we have also run the full numerical method at various grid resolution to verify that the simulations are indeed stable at the various CFL numbers shown in Table 2. There are three key take-aways from Table 2:

- Both methods give stability regions smaller than their CFL conditions (36).
- The LIDG CFL number degrades roughly as the inverse of the method order.
- The RIDG CFL number is roughly one, independent of the method order.

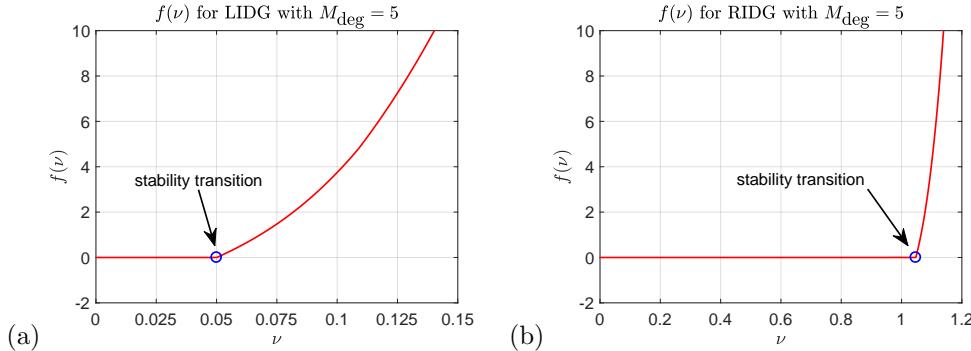


FIG. 2. Plot of the stability function $f(\nu)$ defined by (41) for $M_{\text{deg}} = 5$ for (a) LIDG and (b) RIDG schemes. There is a clear dichotomy between the linearly stable region and the unstable region.

TABLE 2

Numerically estimated maximum CFL numbers for the LIDG (aka Lax–Wendroff DG) and RIDG schemes in one dimension.

1D	$M_{\text{deg}} = 0$	$M_{\text{deg}} = 1$	$M_{\text{deg}} = 2$	$M_{\text{deg}} = 3$	$M_{\text{deg}} = 4$	$M_{\text{deg}} = 5$
LIDG	1.000	0.333	0.171	0.104	0.070	0.050
RIDG	1.000	1.168	1.135	1.097	1.066	1.047

4. Generalization to higher dimensions. We present in this section the generalization of the proposed RIDG method to the case of the 2D and 3D versions of the advection equation. The key innovation beyond what was developed in section 3 for the 1D case is the inclusion of *transverse* cells in the prediction step. With these inclusions, the prediction gives enhanced stability for waves propagating at all angles to the element faces.

4.1. RIDG method in two dimensions. We consider here the 2D advection equation for $(t, x, y) \in [0, T] \times \Omega$ with appropriate boundary conditions:

$$(42) \quad q_{,t} + u_x q_{,x} + u_y q_{,y} = 0.$$

We define a uniform Cartesian mesh with grid spacings Δx and Δy in each coordinate direction. On each spacetime element,

$$(43) \quad \mathcal{S}_{ij}^{n+1/2} = [t^n, t^n + \Delta t] \times [x_i - \Delta x/2, x_i + \Delta x/2] \times [y_j - \Delta y/2, y_j + \Delta y/2],$$

we define the local coordinates, $[\tau, \xi, \eta] \in [-1, 1]^3$, such that

$$(44) \quad t = t^{n+1/2} + \tau (\Delta t/2), \quad x = x_i + \xi (\Delta x/2), \quad \text{and} \quad y = y_j + \eta (\Delta y/2).$$

In these local coordinates, the advection equation is given by

$$(45) \quad q_{,\tau} + \nu_x q_{,\xi} + \nu_y q_{,\eta} = 0, \quad \nu_x = \frac{u_x \Delta t}{\Delta x}, \quad \nu_y = \frac{u_y \Delta t}{\Delta y},$$

where $|\nu_x|$ and $|\nu_y|$ are the CFL numbers in each coordinate direction and the multi-dimensional CFL number is

$$(46) \quad |\nu| := \max \{|\nu_x|, |\nu_y|\}.$$

At the old time, $t = t^n$, we are given the following approximate solution on each space element, $\mathcal{T}_{ij} = [x_i - \Delta x/2, x_i + \Delta x/2] \times [y_j - \Delta y/2, y_j + \Delta y/2]$:

$$(47) \quad q(t^n, x, y) \Big|_{\mathcal{T}_{ij}} \approx q_{ij}^n := \underline{\Phi}^T \underline{Q}_{ij}^n,$$

where $\underline{Q}_{ij}^n \in \mathbb{R}^{M_C}$, $\underline{\Phi} \in \mathbb{R}^{M_C}$, $M_C := (M_{\text{deg}} + 1)(M_{\text{deg}} + 2)/2$, and

$$(48) \quad \underline{\Phi} = (1, \sqrt{3}\xi, \sqrt{3}\eta, \dots) \quad \text{s.t.} \quad \frac{1}{4} \int_{-1}^1 \int_{-1}^1 \underline{\Phi} \underline{\Phi}^T d\xi d\eta = \underline{\underline{\mathbb{I}}} \in \mathbb{R}^{M_C \times M_C}$$

are the orthonormal space Legendre polynomials. In order to compute a predicted solution on each spacetime element we make the following ansatz:

$$(49) \quad q(t, x) \Big|_{\mathcal{S}_{ij}^{n+1/2}} \approx w_{ij}^{n+1/2} := \underline{\Psi}^T \underline{W}_{ij}^{n+1/2},$$

where $\underline{W}_i^{n+1/2} \in \mathbb{R}^{M_P}$, $\underline{\Psi} \in \mathbb{R}^{M_P}$, $M_P := (M_{\text{deg}} + 1)^3$, and

$$(50) \quad \underline{\Psi} = (1, \sqrt{3}\tau, \sqrt{3}\xi, \sqrt{3}\eta, \dots) \quad \text{s.t.} \quad \frac{1}{8} \iiint \underline{\Psi} \underline{\Psi}^T d\tau d\xi d\eta = \underline{\underline{\mathbb{I}}} \in \mathbb{R}^{M_P \times M_P}$$

are the spacetime Legendre basis functions, where all 1D integrals are over $[-1, 1]$.

Remark 2. Just as in the 1D case outlined in section 3, for the RIDG scheme we make use of the $\mathcal{P}(M_{\text{deg}}, M_{\text{dim}})$ spatial basis for the correction step (i.e., $M_C = (M_{\text{deg}} + 1)(M_{\text{deg}} + 2)/2$), and the $\mathcal{Q}(M_{\text{deg}}, M_{\text{dim}} + 1)$ spacetime basis for the prediction step (i.e., $M_P = (M_{\text{deg}} + 1)^3$).

We integrate the advection equation over a spacetime element and apply integrate-by-parts in all three independent variables, τ , ξ , η , which yields

$$(51) \quad \begin{aligned} & \iiint \underline{\Psi} \mathcal{R}(\underline{\Psi})^T \underline{W}_{ij}^{n+1/2} d\underline{\mathcal{S}} + \iint \underline{\Psi}|_{\tau=-1} \left[\underline{\Psi}|_{\tau=-1}^T \underline{W}_{ij}^{n+1/2} - \underline{\Phi}^T \underline{Q}_{ij}^n \right] d\underline{\mathcal{S}}_\tau - \\ & \iint \left\{ \underline{\Psi}|_{\xi=1} \left[\nu_x \underline{\Psi}|_{\xi=1}^T \underline{W}_{ij}^{n+1/2} - \mathcal{F}_{i+\frac{1}{2}, j}^* \right] - \underline{\Psi}|_{\xi=-1} \left[\nu_x \underline{\Psi}|_{\xi=-1}^T \underline{W}_{ij}^{n+1/2} - \mathcal{F}_{i-\frac{1}{2}, j}^* \right] \right\} d\underline{\mathcal{S}}_\xi - \\ & \iint \left\{ \underline{\Psi}|_{\eta=1} \left[\nu_y \underline{\Psi}|_{\eta=1}^T \underline{W}_{ij}^{n+1/2} - \mathcal{G}_{i, j+\frac{1}{2}}^* \right] - \underline{\Psi}|_{\eta=-1} \left[\nu_y \underline{\Psi}|_{\eta=-1}^T \underline{W}_{ij}^{n+1/2} - \mathcal{G}_{i, j-\frac{1}{2}}^* \right] \right\} d\underline{\mathcal{S}}_\eta = 0, \end{aligned}$$

where $\mathcal{R}(\underline{\Psi}) = \underline{\Psi}_{,\tau} + \nu_x \underline{\Psi}_{,\xi} + \nu_y \underline{\Psi}_{,\eta}$, $d\underline{\mathcal{S}} = d\tau d\xi d\eta$, $d\underline{\mathcal{S}}_\tau = d\xi d\eta$, $d\underline{\mathcal{S}}_\xi = d\tau d\eta$, $d\underline{\mathcal{S}}_\eta = d\tau d\xi$, and all 1D integrals are over $[-1, 1]$.

The crux of the idea of the RIDG scheme in two spatial dimensions can be summarized as follows:

- We define a *region* to be the current spacetime element, $\mathcal{S}_{ij}^{n+1/2}$, and the eight neighbors that have a face that shares at least one point in common with one of the faces of $\mathcal{S}_{ij}^{n+1/2}$. This is illustrated in Figure 3.
- For the current spacetime element, $\mathcal{S}_{ij}^{n+1/2}$, we use the correct upwind fluxes to define \mathcal{F}^* and \mathcal{G}^* .
- For the remaining eight elements we use the correct upwind fluxes on all faces that are interior to the region and one-sided fluxes on all faces that are on the boundary of the region. See Figure 3.

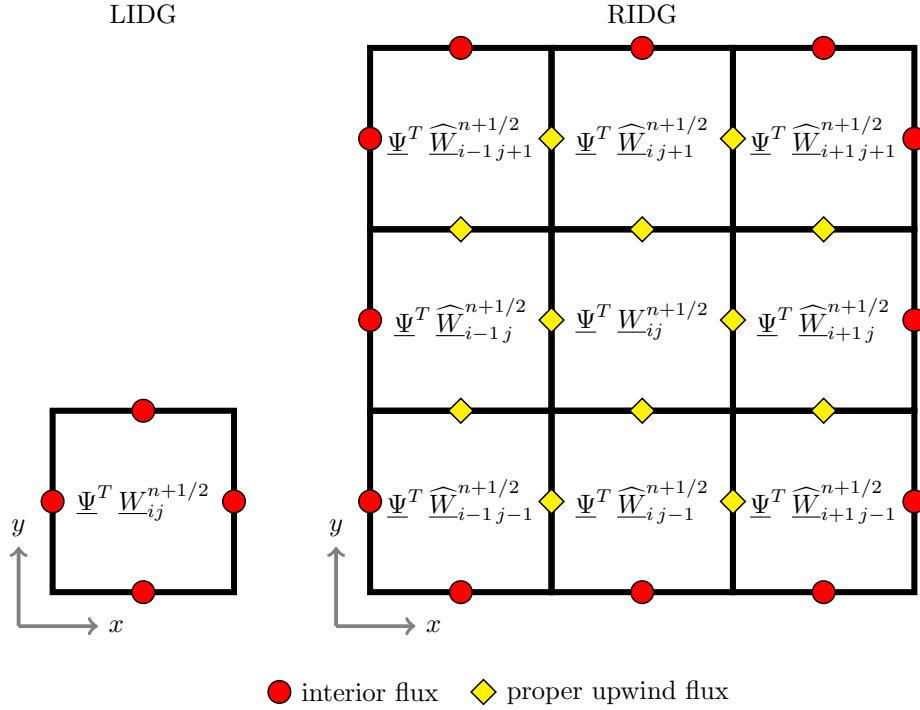


FIG. 3. Shown are the stencils for the LIDG (left) and RIDG (right) prediction steps in two spatial dimensions. The LIDG prediction step is purely local—there is no communication with neighboring cells. The RIDG prediction step computes the proper upwind flux on the spacetime faces shared with immediate neighbors, as well as the four corner elements. In the RIDG prediction step, all of the states, excepting only the one belonging to the middle element, are only temporary variables and will be discarded once the predicted solution in element ij has been computed—to make note of this we place hats over the temporary variables.

- We use the $\mathcal{Q}(M_{\deg}, M_{\dim} + 1)$ spacetime basis in the prediction step (i.e., the full tensor product spacetime basis). Numerical experimentation showed us that using the $\mathcal{Q}(M_{\deg}, M_{\dim} + 1)$ basis for the prediction step, rather than the $\mathcal{P}(M_{\deg}, M_{\dim} + 1)$ basis, produces significantly more accurate results; in the case of linear equations, this creates little additional computational expense since all the relevant matrices can be precomputed.

The result of this is a collection of nine elements with solutions that are coupled to each other, but that are completely decoupled from all remaining elements. This RIDG setup is depicted in Figure 3, where we also show the LIDG setup as a point of comparison.

Remark 3. One of the key innovations in going from the 1D RIDG scheme to its 2D counterpart is the inclusion of the *transverse* elements in the prediction step: $\mathcal{S}_{i-1,j-1}^{n+1/2}$, $\mathcal{S}_{i+1,j-1}^{n+1/2}$, $\mathcal{S}_{i-1,j+1}^{n+1/2}$, and $\mathcal{S}_{i+1,j+1}^{n+1/2}$. Without these transverse cells, the maximum allowable 2D CFL number (46) remains small for any waves traveling transverse to the mesh. In the recent literature, there exist several variants of genuinely multidimensional Riemann solvers (e.g., Balsara [2]); by including the transverse elements, the current work can be viewed as an example of a novel type of multidimensional Riemann solver.

Applying all of the above principles to (51) for all of the nine elements that are in the current region yields a block 9×9 linear system. The left-hand side of this system can be written as

$$(52) \quad \begin{bmatrix} \underline{\underline{L}}^{1010} & \underline{\underline{X}}^- & & \underline{\underline{Y}}^- & & & & & \widehat{W}_{i-1j-1}^{n+1/2} \\ \underline{\underline{X}}^+ & \underline{\underline{L}}^{1110} & \underline{\underline{X}}^- & & \underline{\underline{Y}}^- & & & & \widehat{W}_{ij-1}^{n+1/2} \\ & \underline{\underline{X}}^+ & \underline{\underline{L}}^{0110} & & \underline{\underline{Y}}^- & & & & \widehat{W}_{i+1j-1}^{n+1/2} \\ & & \underline{\underline{L}}^{1011} & \underline{\underline{X}}^- & & \underline{\underline{Y}}^- & & & \widehat{W}_{i-1j}^{n+1/2} \\ \underline{\underline{Y}}^+ & & & \underline{\underline{L}}^{1111} & \underline{\underline{X}}^- & & \underline{\underline{Y}}^- & & \underline{\underline{W}}_{ij}^{n+1/2} \\ & \underline{\underline{Y}}^+ & & \underline{\underline{X}}^+ & \underline{\underline{L}}^{1111} & \underline{\underline{X}}^- & & \underline{\underline{Y}}^- & \widehat{W}_{i+1j}^{n+1/2} \\ & & \underline{\underline{Y}}^+ & & \underline{\underline{L}}^{0111} & & \underline{\underline{X}}^- & & \widehat{W}_{i-1j+1}^{n+1/2} \\ & & & \underline{\underline{Y}}^+ & & \underline{\underline{X}}^+ & \underline{\underline{L}}^{1101} & \underline{\underline{X}}^- & \widehat{W}_{ij+1}^{n+1/2} \\ & & & & \underline{\underline{Y}}^+ & & \underline{\underline{X}}^+ & \underline{\underline{L}}^{0101} & \underline{\underline{X}}^- & \widehat{W}_{i+1j+1}^{n+1/2} \end{bmatrix},$$

and the right-hand side can be written as

$$(53) \quad \left[\underline{\underline{T}} Q_{i-1j-1}^n, \underline{\underline{T}} Q_{ij-1}^n, \dots, \underline{\underline{T}} Q_{ij}^n, \dots, \underline{\underline{T}} Q_{ij+1}^n, \underline{\underline{T}} Q_{i+1j+1}^n \right]^T,$$

where

$$(54) \quad \underline{\underline{X}}^\pm = \mp \frac{\nu_x^\pm}{8} \iint \Psi_{|\xi=\mp 1} \Psi_{|\xi=\pm 1}^T d\underline{\mathcal{S}}_\xi, \quad \underline{\underline{Y}}^\pm = \mp \frac{\nu_y^\pm}{8} \iint \Psi_{|\eta=\mp 1} \Psi_{|\eta=\pm 1}^T d\underline{\mathcal{S}}_\eta,$$

$$(55) \quad \underline{\underline{L}}^{\alpha\beta\gamma\delta} = \underline{\underline{L}}^0 + \alpha \underline{\underline{L}}_x^- + \beta \underline{\underline{L}}_x^+ + \gamma \underline{\underline{L}}_y^- + \delta \underline{\underline{L}}_y^+, \quad \alpha, \beta, \gamma, \delta \in \{0, 1\},$$

$$(56) \quad \underline{\underline{L}}^0 = \frac{1}{8} \iiint \Psi \mathcal{R}(\Psi)^T d\underline{\mathcal{S}} + \frac{1}{8} \iint \Psi_{|\tau=-1} \Psi_{|\tau=-1}^T d\underline{\mathcal{S}}_\tau, \quad \underline{\underline{T}} = \frac{1}{8} \iint \Psi_{|\tau=-1} \Phi^T d\underline{\mathcal{S}}_\tau,$$

$$(57) \quad \underline{\underline{L}}_x^\pm = \pm \frac{\nu_x^\pm}{8} \iint \Psi_{|\xi=\mp 1} \Psi_{|\xi=\mp 1}^T d\underline{\mathcal{S}}_\xi, \quad \underline{\underline{L}}_y^\pm = \pm \frac{\nu_y^\pm}{8} \iint \Psi_{|\eta=\mp 1} \Psi_{|\eta=\mp 1}^T d\underline{\mathcal{S}}_\eta,$$

where $\underline{\underline{T}} \in \mathbb{R}^{M_P \times M_C}$ and $\underline{\underline{X}}^\pm, \underline{\underline{Y}}^\pm, \underline{\underline{L}}^0, \underline{\underline{L}}_x^\pm, \underline{\underline{L}}_y^\pm \in \mathbb{R}^{M_C \times M_P}$.

The correction step can be written as

$$(58) \quad Q_{ij}^{n+1} = Q_{ij}^n + \underline{\underline{C}}_x^- W_{i-1j}^{n+\frac{1}{2}} + \underline{\underline{C}}_y^- W_{ij-1}^{n+\frac{1}{2}} + \underline{\underline{C}}^0 W_{ij}^{n+\frac{1}{2}} + \underline{\underline{C}}_x^+ W_{i+1j}^{n+\frac{1}{2}} + \underline{\underline{C}}_y^+ W_{ij+1}^{n+\frac{1}{2}},$$

$$(59) \quad \begin{aligned} \underline{\underline{C}}^0 &= \frac{1}{4} \iiint \mathcal{U}(\Phi) \Psi^T d\underline{\mathcal{S}} - \frac{1}{4} \iint \left[\nu_x^+ \Phi_{|\xi=1} \Psi_{|\xi=1}^T - \nu_x^- \Phi_{|\xi=-1} \Psi_{|\xi=-1}^T \right] d\underline{\mathcal{S}}_\xi \\ &\quad - \frac{1}{4} \iint \left[\nu_y^+ \Phi_{|\eta=1} \Psi_{|\eta=1}^T - \nu_y^- \Phi_{|\eta=-1} \Psi_{|\eta=-1}^T \right] d\underline{\mathcal{S}}_\eta, \end{aligned}$$

$$(60) \quad \underline{\underline{C}}_x^\mp = \pm \frac{\nu_x^\pm}{4} \iint \Phi_{|\xi=\mp 1} \Psi_{|\xi=\pm 1}^T d\underline{\mathcal{S}}_\xi, \quad \underline{\underline{C}}_y^\mp = \pm \frac{\nu_y^\pm}{4} \iint \Phi_{|\eta=\mp 1} \Psi_{|\eta=\pm 1}^T d\underline{\mathcal{S}}_\eta,$$

where $\mathcal{U}(\Phi) = \nu_x \Phi_{,\xi} + \nu_y \Phi_{,\eta}$, $\underline{\underline{C}}^0, \underline{\underline{C}}_x^\pm, \underline{\underline{C}}_y^\pm \in \mathbb{R}^{M_C \times M_P}$.

4.2. RIDG method in three dimensions. We consider here the 3D advection equation for $(t, x, y, z) \in [0, T] \times \Omega$ with appropriate boundary conditions:

$$(61) \quad q_{,t} + u_x q_{,x} + u_y q_{,y} + u_z q_{,z} = 0.$$

TABLE 3

Numerically estimated maximum CFL numbers, $|\nu|$, for the LIDG (aka Lax–Wendroff DG) and RIDG schemes in two and three dimensions.

2D	$M_{\text{deg}} = 0$	$M_{\text{deg}} = 1$	$M_{\text{deg}} = 3$	$M_{\text{deg}} = 5$	$M_{\text{deg}} = 7$	$M_{\text{deg}} = 9$
LIDG	0.50	0.23	0.08	0.04	0.025	0.01
RIDG	1.00	1.00	0.80	0.75	0.75	0.75
3D	$M_{\text{deg}} = 0$	$M_{\text{deg}} = 1$	$M_{\text{deg}} = 3$	$M_{\text{deg}} = 5$	$M_{\text{deg}} = 7$	$M_{\text{deg}} = 9$
LIDG	0.33	0.10	0.03	0.025	0.02	0.01
RIDG	1.00	0.80	0.60	0.60	0.60	0.60

We define a uniform Cartesian mesh with grid spacings Δx , Δy , and Δz in each coordinate direction. On each spacetime element,

$$(62) \quad \mathcal{S}_{ijk}^{n+1/2} = \mathcal{S}_{ij}^{n+1/2} \times [z_k - \Delta z/2, z_k + \Delta z/2],$$

where $\mathcal{S}_{ij}^{n+1/2}$ is defined by (43), we define the local coordinates, $[\tau, \xi, \eta, \zeta] \in [-1, 1]^4$, such that

$$(63) \quad t = t^{n+1/2} + \tau (\Delta t/2), \quad x = x_i + \xi (\Delta x/2), \quad y = y_j + \eta (\Delta y/2), \quad z = z_k + \zeta (\Delta z/2).$$

In these local coordinates, the advection equation is given by

$$(64) \quad q_{,\tau} + \nu_x q_{,\xi} + \nu_y q_{,\eta} + \nu_z q_{,\zeta} = 0, \quad \nu_x = \frac{u_x \Delta t}{\Delta x}, \quad \nu_y = \frac{u_y \Delta t}{\Delta y}, \quad \nu_z = \frac{u_z \Delta t}{\Delta z},$$

where $|\nu_x|$, $|\nu_y|$, and $|\nu_z|$ are the CFL numbers in each coordinate direction and the multidimensional CFL number is

$$(65) \quad |\nu| := \max \{|\nu_x|, |\nu_y|, |\nu_z|\}.$$

The development of the RIDG scheme in three dimensions is completely analogous to the 2D RIDG scheme from subsection 4.1. In one dimension the prediction step requires a stencil of three elements, in two dimensions we need $3^2 = 9$ elements, and in three dimensions we need $3^3 = 27$ elements. For the sake of brevity we omit the details.

4.3. Von Neumann stability analysis for both LIDG and RIDG. Linear stability analysis proceeds in two and three dimensions in the same manner as in one dimension. We take the numerical update and make the Fourier ansatz

$$(66) \quad \underline{Q}_{ijk}^{n+1} = \underline{\tilde{Q}}^{n+1} e^{I(\omega_x i + \omega_y j + \omega_z k)} \quad \text{and} \quad \underline{Q}_{ijk}^n = \underline{\tilde{Q}}^n e^{I(\omega_x i + \omega_y j + \omega_z k)},$$

where $I = \sqrt{-1}$ and $0 \leq \omega_x, \omega_y, \omega_z \leq 2\pi$ are the wave numbers in each coordinate direction. After using this ansatz, we again write the resulting update in the form

$$(67) \quad \underline{\tilde{Q}}^{n+1} = \underline{\mathcal{M}}(\nu_x, \nu_y, \nu_z, \omega_x, \omega_y, \omega_z) \underline{\tilde{Q}}^n$$

for some matrix $\underline{\mathcal{M}} \in \mathbb{R}^{M_C \times M_C}$. Finally, we define the function

$$(68) \quad f(\nu_x, \nu_y, \nu_z) := \max_{0 \leq \omega_x, \omega_y, \omega_z \leq 2\pi} \rho(\underline{\mathcal{M}}(\nu_x, \nu_y, \nu_z, \omega_x, \omega_y, \omega_z)) - 1,$$

where $\rho(\underline{\mathcal{M}})$ is the spectral radius of $\underline{\mathcal{M}}$.

Just as in one dimension, we estimate the maximum CFL numbers of LIDG and RIDG by studying the values of (68). Our numerically obtained estimates for the maximum value of $|\nu|$ as defined by (46) and (65) are summarized in Table 3. Again we see the following:

- LIDG: the maximum stable CFL number tends to zero as the polynomial degree is increased; and
- RIDG: the maximum stable CFL number has a finite lower bound with increasing polynomial degree (approximately 0.75 in two dimensions and 0.60 in three dimensions).

To get a more detailed view of the stability function (68) in two dimensions, we show false color plots of $f(\nu_x, \nu_y) + 1$ in Figure 4 for both LIDG and RIDG for various method orders. The transverse elements that were included in the prediction step for RIDG (see subsection 4.1) are critically important in achieving a stability region that does not significantly degrade in going from one dimension to two dimensions.

5. Numerical convergence studies. In this section we present convergence studies in one, two, and three dimensions for both LIDG and RIDG and compare the errors and runtimes for the two methods. In all cases, we compute an approximate order of accuracy using the following approximation:

$$(69) \quad \text{error}(h) = ch^M + \mathcal{O}(h^{M+1}) \implies M \approx \frac{\log(\text{error}(h_1)/\text{error}(h_2))}{\log(h_1/h_2)}.$$

5.1. 1D convergence tests. We consider the 1D advection equation (7) with $u = 1$, $\Omega = [-1, 1]$, periodic boundary conditions, and initial condition

$$(70) \quad q(t = 0, x) = \sin(16\pi x).$$

We run the code [9] to $t = 2$ with $M_{\text{deg}} = 3$ (LIDG: $\nu = 0.104$; RIDG: $\nu = 0.9$) and $M_{\text{deg}} = 5$ (LIDG: $\nu = 0.04$; RIDG: $\nu = 0.9$) and compare runtimes and errors; the results are shown in Table 4. We see that both methods exhibit the expected convergence rates in L^1 , L^2 , and L^∞ . For a fixed number of elements, usage of the RIDG method leads to smaller errors. We also notice that for a fixed number of elements the experiment runtime for the RIDG method is shorter than that of the LIDG method—this is due to the increase in the maximum linearly stable CFL number from LIDG to RIDG.

5.2. 2D convergence tests. We consider the 2D advection equation (42) with $u_x = u_y = 1$, $\Omega = [-1, 1]^2$, double periodic boundary conditions, and initial condition

$$(71) \quad q(t = 0, x, y) = \sin(16\pi x) \sin(16\pi y).$$

We run the code [9] to $t = 2$ with $M_{\text{deg}} = 3$ (LIDG: $\nu = 0.05$; RIDG: $\nu = 0.75$) and $M_{\text{deg}} = 5$ (LIDG: $\nu = 0.03$; RIDG: $\nu = 0.75$) and compare runtimes and errors; the results are shown in Table 5. We again see that both methods exhibit the expected convergence rates in L^1 , L^2 , and L^∞ . For a fixed number of elements, usage of the RIDG method leads to slightly smaller errors. We also notice that for a fixed number of elements the experiment runtime for the RIDG method is shorter than that of the LIDG method—this is due to the increase in the maximum linearly stable CFL number from LIDG to RIDG.

5.3. 3D convergence tests. We consider the 3D advection equation (61) with $u_x = u_y = u_z = 1$, $\Omega = [-1, 1]^3$, triple periodic boundary conditions, and initial condition

$$(72) \quad q(t = 0, x, y, z) = \sin(2\pi x) \sin(2\pi y) \sin(2\pi z).$$

We run the code [9] to $t = 2$ with $M_{\text{deg}} = 3$ (LIDG: $\nu = 0.03$; RIDG: $\nu = 0.6$) and $M_{\text{deg}} = 5$ (LIDG: $\nu = 0.025$; RIDG: $\nu = 0.6$) and compare the error properties of the

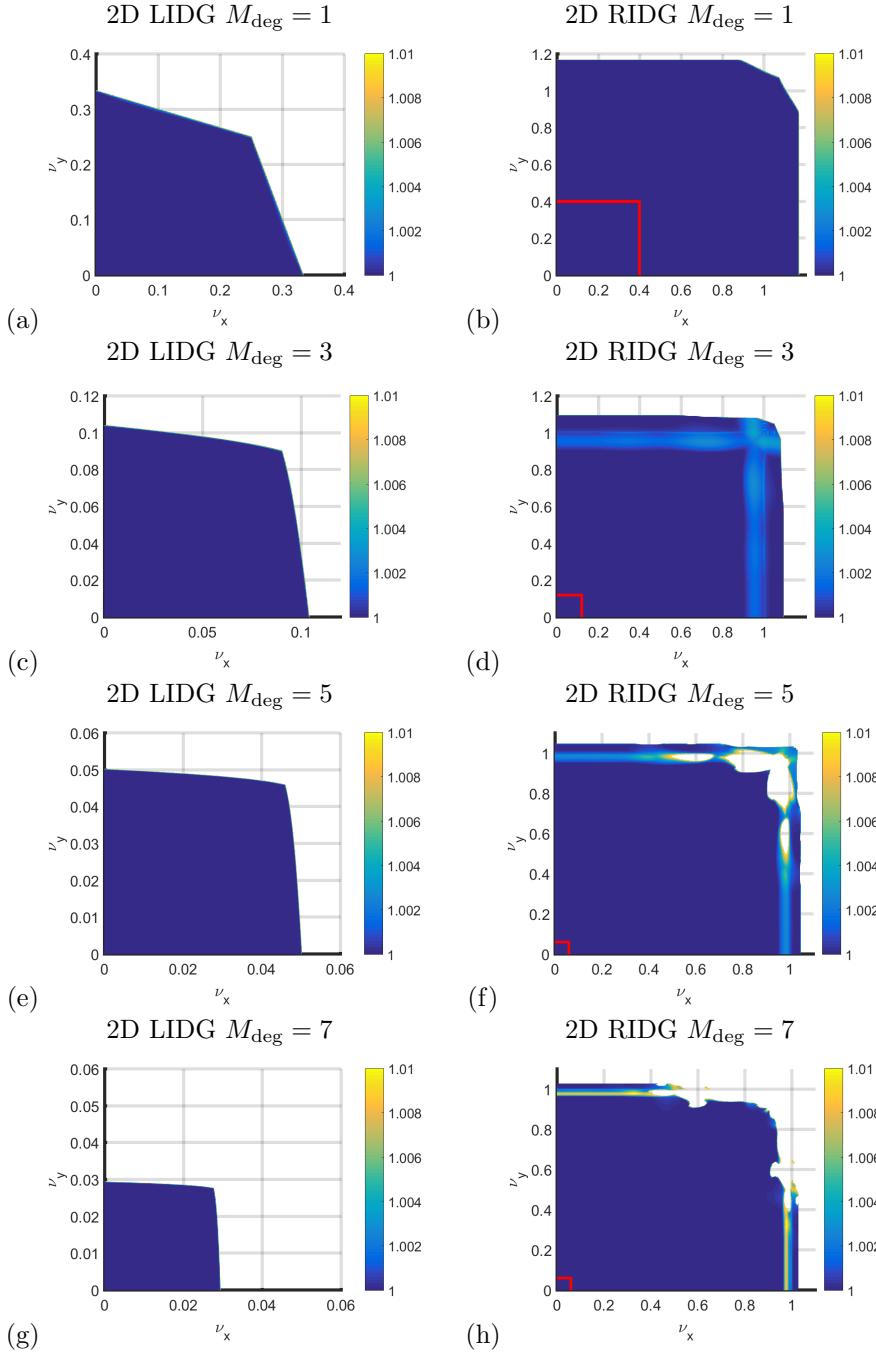


FIG. 4. Stability plots for the 2D LIDG and RIDG methods for various polynomial orders. Shown is a false color plot of $f(\nu_x, \nu_y) + 1$ as defined by (68). Note the different horizontal and vertical scales on the plots of the LIDG and RIDG schemes. The red box in each RIDG plot demonstrates the plotting bounds for the LIDG stability region for the same method order. From these plots we can estimate the maximum CFL number $|\nu|$ as defined by (46): (a) $|\nu| \lesssim 0.23$ ($M_{deg} = 1$, LIDG), (b) $|\nu| \lesssim 1.00$ ($M_{deg} = 1$, RIDG), (c) $|\nu| \lesssim 0.08$ ($M_{deg} = 3$, LIDG), (d) $|\nu| \lesssim 0.80$ ($M_{deg} = 3$, RIDG), (e) $|\nu| \lesssim 0.04$ ($M_{deg} = 5$, LIDG), (f) $|\nu| \lesssim 0.75$ ($M_{deg} = 5$, RIDG), (g) $|\nu| \lesssim 0.025$ ($M_{deg} = 7$, LIDG), and (h) $|\nu| \lesssim 0.75$ ($M_{deg} = 7$, RIDG).

TABLE 4

Convergence and runtime study for the 1D LIDG and RIDG methods with $M_{\text{deg}} = 3$ and $M_{\text{deg}} = 5$ on 1D advection equation (7) with initial condition (70). Shown for various mesh sizes are the runtimes (T_r) measured in seconds, the relative errors in L^1 , L^2 , and L^∞ , as well as the estimated convergence rates according to formula (69). We see that for any fixed number of elements, the RIDG method has a shorter runtime (i.e., computational cost) and smaller error.

1D LIDG: ($M_{\text{deg}} = 3, \nu = 0.104$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
40	0.221	1.83e-1	—	1.83e-1	—	1.92e-1	—
80	0.784	1.08e-2	4.08	1.07e-2	4.09	1.13e-2	4.09
160	3.073	6.52e-4	4.05	6.46e-4	4.05	6.66e-4	4.09
320	12.246	4.01e-5	4.02	4.00e-5	4.01	4.10e-5	4.02
640	48.928	2.49e-6	4.01	2.50e-6	4.00	2.79e-6	3.88
1D RIDG: ($M_{\text{deg}} = 3, \nu = 0.9$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
40	0.032	8.46e-2	—	8.77e-2	—	1.02e-1	—
80	0.121	3.67e-3	4.53	3.72e-3	4.56	4.68e-3	4.45
160	0.473	1.51e-4	4.61	1.52e-4	4.62	1.76e-4	4.73
320	1.885	7.96e-6	4.24	8.02e-6	4.24	8.95e-6	4.30
640	7.618	4.75e-7	4.07	4.77e-7	4.07	5.57e-7	4.01
1D LIDG: ($M_{\text{deg}} = 5, \nu = 0.04$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
40	0.518	1.11e-3	—	1.11e-3	—	1.25e-3	—
80	2.041	1.74e-5	6.00	1.76e-5	5.98	1.88e-5	6.06
160	8.060	2.73e-7	5.99	2.72e-7	6.02	2.86e-7	6.04
320	32.196	4.24e-9	6.01	4.23e-9	6.01	4.36e-9	6.03
640	129.172	6.61e-11	6.00	6.61e-11	6.00	6.78e-11	6.01
1D RIDG: ($M_{\text{deg}} = 5, \nu = 0.9$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
40	0.033	1.50e-4	—	1.65e-4	—	4.64e-4	—
80	0.123	2.68e-6	5.81	2.79e-6	5.89	5.19e-6	6.48
160	0.482	3.91e-8	6.10	4.05e-8	6.11	4.89e-8	6.73
320	1.936	5.85e-10	6.06	6.12e-10	6.05	8.37e-10	5.87
640	7.733	8.94e-12	6.03	9.46e-12	6.02	1.36e-11	5.94

solution produced by the LIDG and RIDG methods; the results are shown in Table 6. We again see that both methods exhibit the expected convergence rates in L^1 , L^2 , and L^∞ . As in the 1D and 2D settings, the RIDG method exhibits better error and runtime properties than the LIDG method.

6. Nonlinear RIDG. We show in this section how to extend the RIDG scheme to nonlinear problems. We show computational comparisons of the proposed RIDG scheme to a standard RKDG scheme on the 1D and 2D Burgers equation.

6.1. Burgers equation in one dimension. We consider the nonlinear inviscid Burgers equation in one dimension:

$$(73) \quad q_{,t} + \frac{1}{2} (q^2)_{,x} = 0,$$

where $(t, x) \in [0, T] \times [0, 2\pi]$ and periodic boundary conditions are assumed. T is chosen as some time before shock formation occurs in the exact solution. The initial conditions are taken to be

$$(74) \quad q(t=0, x) = 1 - \cos(x).$$

Unlike in the linear advection case, nonlinear conservation laws will require us to solve nonlinear algebraic equations in each of the regions depicted in Figure 1. These

TABLE 5

Convergence and runtime study for the 2D LIDG and RIDG methods with $M_{\text{deg}} = 3$ and $M_{\text{deg}} = 5$ on 2D advection equation (42) with initial condition (71). Shown for various mesh sizes are the runtimes (T_r) measured in seconds, the relative errors in L^1 , L^2 , and L^∞ , as well as the estimated convergence rates according to formula (69). We see that for any fixed number of elements, the RIDG method has a shorter runtime (i.e., computational cost) and slightly smaller error.

2D LIDG: ($M_{\text{deg}} = 3, \nu = 0.05$)							
Mesh	T_r (s)	L^1 Error	(69)	L^2 Error	(69)	L^∞ error	(69)
40^2	22.2	$8.75e-1$	—	$7.87e-1$	—	$7.93e-1$	—
80^2	176.8	$6.37e-2$	3.78	$5.72e-2$	3.78	$6.54e-2$	3.6
160^2	1415.7	$1.98e-3$	5.01	$1.81e-3$	4.98	$2.94e-3$	4.48
320^2	11335.2	$7.66e-5$	4.69	$7.09e-5$	4.67	$1.61e-4$	4.19
2D RIDG: ($M_{\text{deg}} = 3, \nu = 0.75$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
40^2	3.5	$6.29e-1$	—	$5.58e-1$	—	$5.62e-1$	—
80^2	27.1	$2.81e-2$	4.49	$2.54e-2$	4.46	$3.45e-2$	4.03
160^2	218.6	$1.04e-3$	4.75	$9.58e-4$	4.73	$1.76e-3$	4.29
320^2	1740.0	$5.37e-5$	4.28	$4.95e-5$	4.27	$1.09e-4$	4.02
2D LIDG: ($M_{\text{deg}} = 5, \nu = 0.03$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
40^2	42.0	$2.25e-2$	—	$2.24e-2$	—	$5.25e-2$	—
80^2	338.1	$2.94e-4$	6.26	$2.77e-4$	6.34	$6.80e-4$	6.27
160^2	2704.0	$2.81e-6$	6.71	$2.75e-6$	6.66	$1.05e-5$	6.02
320^2	21730.0	$3.53e-8$	6.31	$3.50e-8$	6.30	$1.67e-7$	5.98
2D RIDG: ($M_{\text{deg}} = 5, \nu = 0.75$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
40^2	4.1	$5.76e-3$	—	$5.86e-3$	—	$3.30e-2$	—
80^2	32.9	$1.62e-4$	5.15	$1.54e-4$	5.25	$6.30e-4$	5.71
160^2	254.5	$2.25e-6$	6.18	$2.16e-6$	6.15	$9.18e-6$	6.10
320^2	2035.1	$3.04e-8$	6.21	$3.00e-8$	6.17	$1.51e-7$	5.93

TABLE 6

Convergence and runtime study for the 3D LIDG and RIDG methods with $M_{\text{deg}} = 3$ and $M_{\text{deg}} = 5$ on 3D advection equation (61) with initial condition (72). Shown for various mesh sizes are the runtimes (T_r) measured in seconds, the relative errors in L^1 , L^2 , and L^∞ , as well as the estimated convergence rates according to formula (69). We see that for any fixed number of elements, the RIDG method has a shorter runtime (i.e., computational cost) and slightly smaller error.

3D LIDG: ($M_{\text{deg}} = 3, \nu = 0.03$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
20^3	123.9	$1.21e-3$	—	$1.20e-3$	—	$6.16e-3$	—
40^3	2027.4	$6.82e-5$	4.15	$6.95e-5$	4.11	$3.92e-4$	3.97
80^3	32632.4	$4.21e-6$	4.02	$4.31e-6$	4.01	$2.50e-5$	3.97
3D RIDG: ($M_{\text{deg}} = 3, \nu = 0.6$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
20^3	28.5	$9.24e-4$	—	$9.86e-4$	—	$5.02e-3$	—
40^3	457.8	$5.85e-5$	3.98	$6.21e-5$	3.99	$3.15e-4$	3.99
80^3	7299.2	$3.68e-6$	3.99	$3.89e-6$	6.20	$1.96e-5$	4.01
3D LIDG: ($M_{\text{deg}} = 5, \nu = 0.025$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
20^3	318.8	$1.41e-5$	—	$1.34e-5$	—	$9.19e-5$	—
40^3	4960.1	$1.91e-7$	6.21	$1.79e-7$	6.23	$1.53e-6$	5.90
80^3	77582.5	$2.65e-9$	6.17	$2.50e-9$	6.16	$2.48e-8$	5.95
3D RIDG: ($M_{\text{deg}} = 5, \nu = 0.6$)							
Mesh	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
20^3	61.6	$1.01e-5$	—	$9.77e-6$	—	$6.91e-5$	—
40^3	960.2	$1.37e-7$	6.20	$1.36e-7$	6.16	$1.08e-6$	6.00
80^3	15510.1	$1.81e-9$	6.24	$1.88e-9$	6.18	$1.72e-8$	5.98

nonlinear algebraic equations can be written in terms of a nonlinear residual defined on each region:

$$(75) \quad \underline{\mathcal{R}} = \begin{pmatrix} \underline{\mathcal{R}}_1 \\ \underline{\mathcal{R}}_2 \\ \underline{\mathcal{R}}_3 \end{pmatrix},$$

where

$$(76) \quad \begin{aligned} \underline{\mathcal{R}}_1 &= \int \underline{\Psi}_{|\tau=1} \underline{\Psi}_{|\tau=1}^T \underline{W}_{i-1}^{n+1/2} d\underline{\mathcal{S}}_\tau - \int \underline{\Psi}_{|\tau=-1} \underline{\Psi}_{|\tau=1}^T \underline{W}_{i-1}^n d\underline{\mathcal{S}}_\tau \\ &\quad + \nu_x \int \underline{\Psi}_{|\xi=1} \tilde{f} \left(\underline{\Psi}_{|\xi=1}^T \underline{W}_{i-1}^{n+1/2}, \underline{\Psi}_{|\xi=-1}^T \underline{W}_i^{n+1/2} \right) d\underline{\mathcal{S}}_\xi \\ &\quad - \nu_x \int \underline{\Psi}_{|\xi=-1} f \left(\underline{\Psi}_{|\xi=-1}^T \underline{W}_{i-1}^{n+1/2} \right) d\underline{\mathcal{S}}_\xi \\ &\quad - \iint \underline{\Psi}_{|\tau} \underline{\Psi}^T \underline{W}_i^{n+1/2} d\underline{\mathcal{S}} - \nu_x \iint \underline{\Psi}_{|\xi} f \left(\underline{\Psi}^T \underline{W}_i^{n+1/2} \right) d\underline{\mathcal{S}}, \end{aligned}$$

$$(77) \quad \begin{aligned} \underline{\mathcal{R}}_2 &= \int \underline{\Psi}_{|\tau=1} \underline{\Psi}_{|\tau=1}^T \underline{W}_i^{n+1/2} d\underline{\mathcal{S}}_\tau - \int \underline{\Psi}_{|\tau=-1} \underline{\Psi}_{|\tau=1}^T \underline{W}_i^n d\underline{\mathcal{S}}_\tau \\ &\quad + \nu_x \int \underline{\Psi}_{|\xi=1} \tilde{f} \left(\underline{\Psi}_{|\xi=1}^T \underline{W}_i^{n+1/2}, \underline{\Psi}_{|\xi=-1}^T \underline{W}_{i+1}^{n+1/2} \right) d\underline{\mathcal{S}}_\xi \\ &\quad - \nu_x \int \underline{\Psi}_{|\xi=-1} \tilde{f} \left(\underline{\Psi}_{|\xi=-1}^T \underline{W}_{i-1}^{n+1/2}, \underline{\Psi}_{|\xi=-1}^T \underline{W}_i^{n+1/2} \right) d\underline{\mathcal{S}}_\xi \\ &\quad - \iint \underline{\Psi}_{|\tau} \underline{\Psi}^T \underline{W}_i^{n+1/2} d\underline{\mathcal{S}} - \nu_x \iint \underline{\Psi}_{|\xi} f \left(\underline{\Psi}^T \underline{W}_i^{n+1/2} \right) d\underline{\mathcal{S}}, \end{aligned}$$

$$(78) \quad \begin{aligned} \underline{\mathcal{R}}_3 &= \int \underline{\Psi}_{|\tau=1} \underline{\Psi}_{|\tau=1}^T \underline{W}_{i+1}^{n+1/2} d\underline{\mathcal{S}}_\tau - \int \underline{\Psi}_{|\tau=-1} \underline{\Psi}_{|\tau=1}^T \underline{W}_{i+1}^n d\underline{\mathcal{S}}_\tau \\ &\quad + \nu_x \int \underline{\Psi}_{|\xi=1} f \left(\underline{\Psi}_{|\xi=1}^T \underline{W}_{i+1}^{n+1/2} \right) d\underline{\mathcal{S}}_\xi \\ &\quad - \nu_x \int \underline{\Psi}_{|\xi=-1} \tilde{f} \left(\underline{\Psi}_{|\xi=-1}^T \underline{W}_i^{n+1/2}, \underline{\Psi}_{|\xi=-1}^T \underline{W}_{i+1}^{n+1/2} \right) d\underline{\mathcal{S}}_\xi \\ &\quad - \iint \underline{\Psi}_{|\tau} \underline{\Psi}^T \underline{W}_{i+1}^{n+1/2} d\underline{\mathcal{S}} - \nu_x \iint \underline{\Psi}_{|\xi} f \left(\underline{\Psi}^T \underline{W}_{i+1}^{n+1/2} \right) d\underline{\mathcal{S}}. \end{aligned}$$

For such general nonlinear problems, we use the Rusanov [19] numerical flux in lieu of the upwinded fluxes in the prediction step (seen in Figure 1) and again for the time-averaged fluxes in the correction step. For a scalar conservation law with flux function $f(q)$ and flux Jacobian $f'(q)$, the Rusanov flux is

$$(79) \quad \tilde{f}(q_\ell, q_r) = \frac{1}{2} (f(q_\ell) + f(q_r)) - \frac{\lambda(q_\ell, q_r)}{2} (q_r - q_\ell),$$

where for scalar conservation laws

$$(80) \quad \lambda(q_\ell, q_r) = \max \{ |f'(q_\ell)|, |f'((q_\ell + q_r)/2)|, |f'(q_r)| \}.$$

For Burgers equation (73), the numerical flux (79) becomes

$$(81) \quad \tilde{f}(q_\ell, q_r) = \frac{1}{4} q_\ell^2 + \frac{1}{4} q_r^2 - \frac{\max \{ |q_\ell|, |q_r| \}}{2} (q_r - q_\ell).$$

The goal in each region in each time-step is to minimize residual (75) with respect to the unknown space-time Legendre coefficients of the approximate solution. We accomplish this by utilizing a Newton iteration. When forming the Newton iteration Jacobian (not to be confused with the flux Jacobian of the hyperbolic conservation law), one must compute the Jacobian of (75) by differentiating with respect to each coefficient. That is, we must compute

$$(82) \quad \underline{\underline{\mathcal{J}}} = \begin{pmatrix} \frac{\partial \mathcal{R}_1}{\partial \underline{W}_{i-1}^{n+1/2}} & \frac{\partial \mathcal{R}_1}{\partial \underline{W}_i^{n+1/2}} & 0 \\ \frac{\partial \mathcal{R}_2}{\partial \underline{W}_{i-1}^{n+1/2}} & \frac{\partial \mathcal{R}_2}{\partial \underline{W}_i^{n+1/2}} & \frac{\partial \mathcal{R}_2}{\partial \underline{W}_{i+1}^{n+1/2}} \\ 0 & \frac{\partial \mathcal{R}_3}{\partial \underline{W}_i^{n+1/2}} & \frac{\partial \mathcal{R}_3}{\partial \underline{W}_{i+1}^{n+1/2}} \end{pmatrix},$$

which is analogous to the coefficient matrix in the linear advection case (e.g., see (28)). When computing the entries in (82), one must deal with the fact that the wave speed that appears in the Rusanov flux (79) is not a smooth function of the coefficients; in order to handle this issue we impose in the computation of Jacobian (82) the following condition: $\frac{\partial}{\partial W} \lambda = 0$. This assumption seems to work well in practice, as evidenced in the results below, though other assumptions may be considered in the future. The stopping criterion for the Newton iteration that seems to be most effective at producing efficient solutions is the following:

- stop if the residual of the region's main cell (the cell for which we are forming a prediction) is below a certain tolerance ($TOL = 10^{-4}$);
- stop if a maximum number of iterations is reached ($N_{\text{iters}} = 3$).

Unlike in the linear case, for nonlinear conservation laws we cannot completely precompute the prediction update. However, we are able to leverage a so-called quadrature-free implementation [1] to increase the efficiency of the quadrature part of the prediction step. We review this methodology here: when computing the space-time quadrature, the most computationally expensive pieces are associated with the term in the residual and Jacobian where we integrate in space-time over the cell.

For example, one such term for the Burgers equation is

$$(83) \quad \begin{aligned} \underline{\mathcal{R}}_1|_{\text{volume}} &= -\nu_x \iint \Psi_{,\xi} f \left(\Psi^T \underline{W}_{i-1}^{n+1/2} \right) d\underline{S} \\ &= -\nu_x \iint \Psi_{,\xi} \frac{1}{2} \left(\Psi^T \underline{W}_{i-1}^{n+1/2} \right)^2 d\underline{S}. \end{aligned}$$

The contribution to the Jacobian matrix from this term is

$$(84) \quad \frac{\partial \mathcal{R}_1|_{\text{volume}}}{\partial \underline{W}_{i-1}^{n+1/2}} = -\nu_x \iint \Psi_{,\xi} \Psi^T \underline{W}_{i-1}^{n+1/2} \Psi^T d\underline{S}.$$

Each entry of this matrix has the form

$$(85) \quad \left[\frac{\partial \mathcal{R}_1|_{\text{volume}}}{\partial \underline{W}_{i-1}^{n+1/2}} \right]_{ab} = \iint \psi_{,\xi}^{(a)} \sum_{\ell=1}^{\theta_T} \psi^{(\ell)} Q_\ell \psi^{(b)} d\underline{S} = \left(\sum_{\ell=1}^{\theta_T} \iint \psi_{,\xi}^{(a)} \psi^{(\ell)} \psi^{(b)} d\underline{S} \right) Q_\ell.$$

Notice that the expression within the parentheses found in (85) can be precomputed using exact expressions (e.g., with a symbolic toolbox if the exact expressions are too laborious to derive by hand). This allows us to forego an expensive quadrature routine

TABLE 7

Convergence and runtime study for the 1D RKDG $M_{\text{deg}} = 3$ and the RIDG methods for $M_{\text{deg}} = 3, 5, 7$ on the inviscid Burgers equation (73) with initial conditions (74). Shown for various mesh sizes are the number of time-steps taken, N_T , the runtimes in seconds, T_r , and the relative errors as measured in the L^1 , L^2 , and L^∞ norms.

1D RKDG: ($M_{\text{deg}} = 3, \nu = 0.1$)								
Mesh	N_T	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
39	30	0.404	1.45E-07	-	2.39E-07	-	1.30E-06	-
52	39	0.657	4.69E-08	3.91	7.69E-08	3.94	4.26E-07	3.89
65	48	0.879	1.95E-08	3.94	3.18E-08	3.95	1.77E-07	3.93
77	57	1.243	9.93E-09	3.98	1.63E-08	3.96	9.08E-08	3.95
91	66	1.732	5.11E-09	3.98	8.39E-09	3.96	4.68E-08	3.97
105	76	2.283	2.94E-09	3.86	4.76E-09	3.97	2.66E-08	3.96
158	114	5.138	5.72E-10	4.01	9.36E-10	3.98	5.23E-09	3.98
1D RIDG: ($M_{\text{deg}} = 3, \nu = 0.9$)								
Mesh	N_T	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
39	3	0.125	1.47E-07	-	2.35E-07	-	1.48E-06	-
52	4	0.217	4.70E-08	3.97	7.55E-08	3.94	4.85E-07	3.88
65	5	0.330	1.93E-08	4.00	3.12E-08	3.96	2.01E-07	3.95
77	6	0.501	9.69E-09	4.05	1.61E-08	3.93	1.06E-07	3.79
91	7	0.578	4.95E-09	4.03	8.24E-09	4.00	5.65E-08	3.75
105	8	0.810	2.82E-09	3.93	4.69E-09	3.94	3.24E-08	3.89
158	12	2.136	5.54E-10	3.98	9.26E-10	3.97	6.50E-09	3.93
1D RIDG: ($M_{\text{deg}} = 5, \nu = 0.9$)								
Mesh	N_T	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
13	1	0.085	4.03E-08	-	6.79E-08	-	4.98E-07	-
26	2	0.303	6.90E-10	5.87	1.20E-09	5.82	9.38E-09	5.73
39	3	0.653	6.73E-11	5.74	1.22E-10	5.64	1.35E-09	4.79
53	4	0.958	1.03E-11	6.11	1.75E-11	6.31	1.81E-10	6.54
66	5	1.437	2.68E-12	6.15	4.77E-12	5.93	5.33E-11	5.57
1D RIDG: ($M_{\text{deg}} = 7, \nu = 0.9$)								
Mesh	N_T	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
3	1	0.046	1.31E-05	-	2.43E-05	-	1.06E-04	-
8	1	0.192	9.53E-09	7.37	1.48E-08	7.55	1.22E-07	6.90

in favor of an exact expansion of the coefficients Q_ℓ for forming the Newton iteration Jacobian. In [21] this idea was effectively expanded to certain types of nonpolynomial flux functions, indicating that this idea can be generalized. Thus we can avoid space-time quadrature of the volume integrals by integrating the expressions such as in (85) analytically.

Using this approach, we compared the nonlinear RIDG methods of various orders to the fourth-order RKDG method as discussed in [21]. Shown in Table 7 are the computed errors and runtimes for the RKDG method with $M_{\text{deg}} = 3$ (i.e., the fourth-order RKDG method) and the RIDG method $M_{\text{deg}} = 3, 5, 7$; both methods were applied to Burgers equation (73) with initial conditions (74), and run out to time $T = 0.4$ (i.e., before shockwaves form). We see that all methods exhibit the expected convergence rates in the L^1 , L^2 , and L^∞ norms. We note the following:

- For any fixed error that we consider for the RKDG method, the RIDG method of $M_{\text{deg}} = 3$ can obtain a solution of similar accuracy about 2.5 to 3 times faster. Furthermore, the RIDG method of $M_{\text{deg}} = 5$ can obtain a solution of similar accuracy 15 times faster.
- For any fixed error that we consider for the RKDG method, the RIDG method of $M_{\text{deg}} = 3$ can obtain a solution of similar accuracy while taking an order of magnitude fewer time-steps. Furthermore, the RIDG method of $M_{\text{deg}} = 5$

TABLE 8

Convergence and runtime study for the 2D RKDG method, the 2D RIDG method for $M_{\text{deg}} = 3$, and the RIDG method for $M_{\text{deg}} = 5$ on the inviscid Burgers equation (86) with initial conditions (87). Shown for various mesh sizes are the number of time-steps taken, N_T , the runtimes in seconds, T_r , and the relative errors as measured in the L^1 , L^2 , and L^∞ norms.

2D RKDG: ($M_{\text{deg}} = 3, \nu = 0.005$)								
Mesh	N_T	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
11 ²	18	2.793	3.03e-05	—	4.44e-05	—	3.73e-04	—
22 ²	33	18.166	1.93e-06	3.97	2.97e-06	3.90	2.39e-05	3.96
33 ²	49	61.442	4.00e-07	3.88	6.11e-07	3.90	5.40e-06	3.67
44 ²	64	144.332	1.31e-07	3.88	1.98e-07	3.92	1.72e-06	3.98
55 ²	80	284.119	5.50e-08	3.89	8.21e-08	3.94	7.01e-07	4.01
66 ²	95	485.266	2.70e-08	3.91	3.99e-08	3.95	3.47e-07	3.85
122 ²	174	3075.613	2.37e-09	3.96	3.50e-09	3.96	2.99e-08	3.99
2D RIDG: ($M_{\text{deg}} = 3, \nu = 0.75$)								
Mesh	N_T	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
11 ²	1	6.021	2.89e-05	—	4.26e-05	—	2.76e-04	—
22 ²	2	45.500	1.85e-06	3.97	2.89e-06	3.88	1.87e-05	3.89
33 ²	3	142.633	3.93e-07	3.83	6.03e-07	3.86	4.38e-06	3.58
44 ²	4	315.483	1.29e-07	3.86	1.96e-07	3.91	1.33e-06	4.15
55 ²	5	585.140	5.43e-08	3.89	8.17e-08	3.92	5.84e-07	3.68
66 ²	6	984.351	2.66e-08	3.91	3.98e-08	3.94	2.75e-07	4.14
122 ²	12	6499.030	2.35e-09	3.95	3.53e-09	3.94	2.68e-08	3.78
2D RIDG: ($M_{\text{deg}} = 5, \nu = 0.75$)								
Mesh	N_T	T_r (s)	L^1 error	(69)	L^2 error	(69)	L^∞ error	(69)
11 ²	1	158.842	1.38e-07	—	2.42e-07	—	1.82e-06	—
22 ²	2	1190.925	2.35e-09	5.87	4.57e-09	5.73	4.53e-08	5.33
33 ²	3	3750.531	2.29e-10	5.74	4.41e-10	5.77	6.28e-09	4.87

can obtain a solution of similar accuracy while taking almost two orders of magnitude fewer time-steps.

6.2. Burgers equation in two dimensions. Now we consider the nonlinear inviscid Burgers equation in two dimensions:

$$(86) \quad q_t + \left(\frac{1}{2} q^2 \right)_{,x} + \left(\frac{1}{2} q^2 \right)_{,y} = 0,$$

where $(t, x, y) \in [0, T] \times [0, 2\pi]^2$. T is some time before the shock forms in the solution. We consider the initial conditions

$$(87) \quad q(t=0, x) = \frac{1}{4}(1 - \cos(x))(1 - \cos(y)).$$

We again use the Rusanov numerical flux for both the space-time surface integrals in the prediction step and the time-averaged fluxes in the correction step. In Table 8 we compare the performance of the RKDG method to that of the RIDG method for $M_{\text{deg}} = 3, 5$ for $T = 0.4$. We observe the following:

- For all mesh sizes, the RKDG and RIDG methods with $M_{\text{deg}} = 3$ have similar error. The RIDG $M_{\text{deg}} = 3$ solutions take about 2 times longer to obtain, yet are obtained in 16 times fewer time-steps.
- For a fixed error of $\mathcal{O}(10^{-9})$, RKDG method takes 2.6 times longer to run than the RIDG method $M_{\text{deg}} = 5$. Furthermore, the RIDG method $M_{\text{deg}} = 5$ obtains solutions with almost two orders of magnitude fewer time-steps.

We conclude that with respect to serial code execution in two dimensions, the RIDG method $M_{\text{deg}} = 3$ is not as efficient as the RKDG method, but the RIDG method

$M_{\text{deg}} = 5$ is more efficient than the RKDG method. This demonstrates the fact that the RIDG methods do not experience an analogous *Butcher barrier*, which causes efficiency deterioration for Runge–Kutta methods of orders higher than 4. Furthermore, with respect to minimizing the number of time-steps (such as in the context of distributed memory programming), the RIDG method $M_{\text{deg}} = 3$ requires an order of magnitude fewer time-steps for a fixed error than the RKDG method, while the RIDG method $M_{\text{deg}} = 5$ requires almost two orders of magnitude fewer time-steps for a fixed error.

7. Conclusions. The purpose of this work was to develop a novel time-stepping method for high-order discontinuous Galerkin methods that has improved stability properties over traditional approaches (e.g., explicit SSP-RK and Lax–Wendroff). The name we gave to this new approach is the RIDG scheme, due to the fact that the prediction for a given cell is formed via an implicit method using information from small *regions* of cells around a cell, juxtaposed with the *local* predictor that forms a prediction for a given cell using only past information from that cell. More exactly, the RIDG method is comprised of a semilocalized version of a spacetime DG method, and a corrector step, which is an explicit method that uses the solution from the predictor step. In this sense, the stencils of the RIDG schemes are slightly larger than similar explicit methods, and yet are able to take significantly larger time-steps.

With this new scheme we achieved all of the following:

- developed RIDG schemes for 1D, 2D, and 3D advection;
- demonstrated experimentally the correct convergence rates on 1D, 2D, and 3D advection examples;
- showed that the maximum linearly stable CFL number is bounded below by a constant that is independent of the polynomial order (one dimension: 1.00; two dimensions: 0.75; three dimensions: 0.60);
- developed RIDG schemes for 1D and 2D nonlinear scalar equations;
- demonstrated experimentally the correct convergence rates on 1D and 2D nonlinear examples.
- showed that RIDG has larger maximum CFL numbers than explicit SSP-RKDG and Lax–Wendroff DG;
- demonstrated the efficiency of the RIDG schemes for nonlinear problems in one and two dimensions. Namely, we showed that the RIDG methods become more efficient as you increase the method order, as opposed to RKDG methods, whose efficiency deteriorates as you move beyond fourth-order accuracy due to the Butcher barrier.

All of the methods described in this work were written in a MATLAB code that can be freely downloaded [9].

There are many directions for future work for this class of methods, including the following:

- Exploring different methods for finding solutions to the nonlinear rootfinding problem that forms the predictions in each time-step. This includes the possibility of constrained optimization so that the predictions fit some desired criterion such as maintaining positivity.
- Extending the RIDG method to systems of equations while maintaining efficiency. A crucial development will be to extend existing limiter technology, including nonoscillatory limiters and positivity-preserving limiters, to the case of the RIDG scheme. Since the RIDG method takes orders of magnitude fewer time-steps when compared to SSP-RKDG and Lax–Wendroff DG, lim-

iters that are used once or twice per time-step have a reduced effect on the overall runtime of the scheme.

- Implementing domain decomposition schemes and demonstrating efficient many-core scaling for RIDG. The RIDG stencil is small (almost like nearest neighbors) and would need to communicate only twice per time-step. The RKDG, Lax–Wendroff DG, and ADER-DG methods are known to be efficient on many-core systems [5], and so the RIDG method is similar enough to these methods that we expect similar results. However, since the RIDG method takes orders of magnitude fewer time-steps when compared to these other methods, communication costs are minimized and so even greater efficiency might be achieved.

Acknowledgment. We would like to thank the anonymous referees for their thoughtful comments and suggestions that helped to improve this paper.

REFERENCES

- [1] H. ATKINS AND C.-W. SHU, *Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations*, AIAA Journal, 36 (1998), pp. 775–782.
- [2] D. BALSARA, *Multidimensional Riemann problem with self-similar internal structure. Part I—Application to hyperbolic conservation laws on structured meshes*, J. Comput. Phys., 277 (2014), pp. 163–200.
- [3] B. COCKBURN AND C.-W. SHU, *The Runge–Kutta discontinuous Galerkin method for conservation laws V*, J. Comput. Phys., 141 (1998), pp. 199–224.
- [4] R. COURANT, K. FRIEDRICHS, AND H. LEWY, *Über die partiellen Differenzengleichungen der mathematischen Physik*, Math. Ann., 100 (1928), pp. 32–74.
- [5] M. DUMBSER, F. FAMBRI, M. TAVELLI, M. BADER, AND T. WEINZIERL, *Efficient implementation of ADER discontinuous Galerkin schemes for a scalable hyperbolic PDE engine*, Axioms, 7 (2018), p. 63.
- [6] G. GASSNER, M. DUMBSER, F. HINDENLANG, AND C.-D. MUNZ, *Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors*, J. Comput. Phys., 230 (2011), pp. 4232–4247.
- [7] S. GOTTLIEB AND C.-W. SHU, *Total variation diminishing Runge–Kutta schemes*, Math. Comput., 67 (1998), pp. 73–85.
- [8] S. GOTTLIEB, C.-W. SHU, AND E. TADMOR, *Strong stability-preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112.
- [9] P. GUTHREY, *Regionally-Implicit Discontinuous Galerkin (RIDG) Code*, <https://github.com/pguthrey/regionally-implicit-dg>.
- [10] J. HESTHAVEN AND T. WARBURTON, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Springer, New York, 2007.
- [11] G. KARNIADAKIS AND S. SHERWIN, *Spectral/HP Element Methods for Computational Fluid Dynamics*, 2nd ed., Oxford University Press, Oxford, 2013.
- [12] C. KLAJ, J. VAN DER VEGT, AND H. VAN DER VEN, *Space-time discontinuous Galerkin method for the compressible Navier–Stokes equations*, J. Comput. Phys., 217 (2006), pp. 589–611.
- [13] P. LAX AND B. WENDROFF, *Systems of conservation laws*, Comm. Pure Appl. Math., 13 (1960), pp. 217–237.
- [14] R. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.
- [15] R. LEVEQUE, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady State and Time Dependent Problems*, SIAM, Philadelphia, 2007.
- [16] Y. LIU, C.-W. SHU, E. TADMOR, AND M. ZHANG, *L^2 stability analysis of central discontinuous Galerkin method and a comparison between central and regular discontinuous Galerkin methods*, ESAIM Math. Model. Numer. Anal., 42 (2008), pp. 593–607.
- [17] J. QIU, M. DUMBSER, AND C.-W. SHU, *The discontinuous Galerkin method with Lax–Wendroff type time discretizations*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4528–4543.
- [18] W. REED AND T. HILL, *Triangular Mesh Methods for the Neutron Transport Equation*, Technical report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

- [19] V. RUSANOV, *Calculation of interaction of non-steady shock waves with obstacles*, J. Comput. Math. Phys. USSR, 1 (1961), pp. 267–279.
- [20] C.-W. SHU, *High order weighted essentially nonoscillatory schemes for convection dominated problems*, SIAM Rev., 51 (2009), pp. 82–126.
- [21] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
- [22] J. SUDIRHAM, J. VAN DER VEGT, AND R. VAN DAMME, *Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains*, Appl. Numer. Math., 56 (2006), pp. 1491–1518.
- [23] S. VON KOWALESKY, *Zur Theorie der partiellen Differentialgleichungen*, J. Reine Angew. Math., 80 (1875), pp. 1–32.