Eleventh International Conference on
Computational Fluid Dynamics (ICCFD11),
Maui, Hawaii, July 11-15, 2022

ICCFD11-2022-0501

Large-Scale Investigation of 3D Discontinuous-Galerkin-Hancock Method for Hyperbolic Balance Laws with Stiff Local Sources

W. Kaufmann* and J.G. McDonald*,**
Corresponding author: wkauf057@uottawa.ca

* University of Ottawa, Ottawa, ON, Canada, K1N 6N5.
** Canadian Nuclear Laboratories, Chalk River, ON, Canada, K0J 1J0.

Abstract: A previously proposed third-order accurate, coupled space-time discontinuous-Galerkin Hancock (DGH) method is implemented in a multi-dimensional large-scale computational framework for the first time. The predictive properties of the massively parallel DGH implementation are investigated for several fluid flow governing equations including the Euler and ten-moment Gaussian closure models. The DGH method is shown to achieve third-order solution accuracy on grids with skewed elements for hyperbolic equations with stiff sources using only linear basis functions. The scheme achieves high-parallel efficiency on distributed-memory architectures with 100,000's of compute cores.

Keywords: Numerical Algorithms, Discontinuous Galerkin, High-Order Schemes, Coupled Space-Time Methods, Large-Scale Parallel Capability.

1 Introduction

Efficient approaches for the solution of first-order hyperbolic balance laws are important for a variety of physically complex flows such as multiphase, reacting, and non-equilibrium flows. The use of high-order numerical schemes (i.e., higher than second-order accuracy [1]) in conjunction with scalable and efficient parallel implementations provides a viable strategy for mitigating the computational cost of large-scale simulations of such physically complex flows.

Of particular interest and motivation for the present work are moment closures from the kinetic theory of gases. Such moment methods provide first-order hyperbolic-relaxation balance laws that offer an alternative to the Navier-Stokes equations (NSE) for viscous gas-flow prediction. As they are not derived based on a continuum assumption, moment methods offer the promise of maintaining physical validity in regimes for which the NSE provide physically incorrect predictions. Although the first-order hyperbolic nature of the resulting partial differential equations (PDEs) is associated with providing several computational advantages over numerical procedures for the solution of the NSE (e.g., improved robustness and accuracy for the treatment of wall boundaries in the presence of skewed meshes) [2], the moment-method PDEs for fluid flow often include stiff local source terms, which may require special numerical treatment. Thus, the development of efficient high-order discretizations for such PDEs needs to be considered in the context of providing accurate and robust treatment of stiff source terms.

Many hierarchies of high-order methods have been studied in the past decades for the accurate solution of hyperbolic balance laws on both structured and unstructured grids. A family of finite-volume methods known as essentially non-oscillatory (ENO) schemes were first proposed by Harten et al. [3]. These schemes are total-variation-bounded (TVB) and retain high-order accuracy in smooth regions by using an adaptive stencil. Weighted ENO (WENO), e.g., [4, 5, 6], and central ENO (CENO), e.g., [7, 8, 9], schemes were later developed, which utilize a weighted average of multiple stencils and a fixed central stencil with high-order polynomial reconstruction, respectively. Finite-element-type schemes have also been developed, among which are the families of high-order continuous-Galerkin (CG) and discontinuous-Galerkin (DG) schemes.

The family of Runge-Kutta DG (RKDG) schemes were first proposed by Cockburn et al. [10, 11, 12], yielding a TVB scheme that uses multi-step temporal integration. Later, Dumbser et al. [13]. formulated a unified framework for single-step finite-volume and DG schemes denoted $P_N P_M$ methods, in which least-squares polynomial reconstruction is applied to a DG method. Variations of the DG formulation such as spectral volume (SV) and spectral difference (SD) methods have been proposed by Wang [14] and Liu et al.. [15], which share numerous similarities with the DG method. Huynh [16] later introduced a general formulation combining these schemes in the framework of a flux reconstruction (FR) approach. More recent and extensive reviews of the status of high-order methods, their potential and applicability can be found in Wang et al. [1] and Gassner and Winters [17].

The current study investigates the capabilities of a particular member of the discontinuous-Galerkin family of schemes for a range of problems described by first-order hyperbolic PDEs with stiff local source terms. Specifically, the coupled space-time discontinuous-Galerkin Hancock (DGH) method, originally proposed by Suzuki [18] for the efficient solutions of hyperbolic balance laws resulting from non-equilibrium extended hydrodynamics is considered. For the present work, the scheme has been implemented in a massively parallel, multi-dimensional framework for practical computations—to the best of the authors' knowledge this marks the first time it has been presented as such. The current paper describes the evaluation of the scheme at large computational scale for various PDEs. Although only linear basis functions are used, the DGH scheme is shown to achieve third-order accuracy in coupled space time on skewed hexahedral elements and it has been proven to be highly efficient on large-scale, distributed-memory architectures.

The rest of the paper is organized as follows. Section 2 gives a brief review of moment methods and the application of maximum-entropy moment closures for deriving several governing fluid equations. The formulation of the DGH scheme in three dimensions (3D) is presented in Section 3. Furthermore, the application of the scheme on grids with skewed hexahedral elements is illustrated and discussed. In Section 4, numerical results to demonstrate the predictive capabilities of the DGH scheme are presented. The model equations considered are those provided by the linear convection-relaxation, the compressible Euler, and the ten-moment Gaussian closure models. Additionally, parallel performance studies with the DGH implementation are included to illustrate the near-optimum strong scalabilty of the algorithm up to 10,000's and 100,000's of compute CPU-cores on two different distributed-memory architectures. Finally, a set of conclusions are drawn and potential extensions of the scheme are discussed in Section 5.

2 Moment Methods

The efficient numerical solution of hyperbolic-relaxation balance laws describing moment closures from the kinetic theory of gases is the primary motivation for this work. A brief summary of the theory of moment methods is, therefore, provided here for the sake of completeness.

In gas-kinetic theory [19], rather than starting from the assumption that a fluid can be treated as a continuum, it is assumed that a gas is comprised of discrete particles. A distribution function, $\mathcal{F}(x_i, v_i, t)$, describes the density of particles at a position, x_i, with velocity, v_i, at time t. The evolution of the distribution function for a gas with no external acceleration fields is described by the Boltzmann equation,

$$\frac{\partial \mathcal{F}}{\partial t} + v_i \frac{\partial \mathcal{F}}{\partial x_i} = \frac{\delta \mathcal{F}}{\delta t}. \tag{1}$$

The right hand side term, $\frac{\delta \mathcal{F}}{\delta t}$, is known as the collision operator. This term models the effects of inter-particle collisions on the distribution function. This is a high dimensional equation, with three dimensions in physical space, three dimensions in velocity space, and time.

It is clear that any direct numerical solution of the Boltzmann equation is prohibitively expensive. However, the large amount of information the distribution function provides is rarely needed, as macroscopic fluid properties are usually sought. Moments provide a connection between the distribution function and macroscopic properties. To take a moment of a distribution function, the function is multiplied by an appropriate velocity-dependent weight, $W(v_i)$, and integrated over all velocity space. For example, the mass density, ρ, can be found by taking $W(v_i) = m$, where m is the mass of a particle. This is expressed as

$$\rho = \iiint\limits_{-\infty}^{\infty} m\mathcal{F}(x_i, v_i)\,\mathrm{d}v_i = \langle m\mathcal{F}\rangle \,. \tag{2}$$

The shorthand notation, $\langle W(v_i)\mathcal{F}\rangle$, is used to denote integration over all velocity space. Similarly, the momentum density of the gas is found by taking $W(v_i) = mv_i$, such that

$$\rho u_i = \langle m v_i \mathcal{F}\rangle \,, \tag{3}$$

where u_i is the bulk velocity. Moments can also be taken using weights dependent on the random velocity c_i, defined as $c_i = v_i - u_i$. For example, the second-order pressure tensor, P_{ij}, is defined as

$$P_{ij} = \langle m c_i c_j \mathcal{F}\rangle \,. \tag{4}$$

For a monatomic gas, it is related to the traditional thermodynamic pressure with $p = P_{ii}/3$, and to the deviatoric (viscous) stress tensor with $\tau_{ij} = p\delta_{ij} - P_{ij}$. Moments of arbitrarily high order can be taken, however the physical significance of a moment becomes more ambiguous as this order is increased.

Evolution equations for macroscopic properties can be obtained by taking moments of the Boltzmann equation directly. For an arbitrary velocity-dependent weight, W, this yields

$$\frac{\partial}{\partial t}\langle W\mathcal{F}\rangle + \frac{\partial}{\partial x_i}\langle v_i W\mathcal{F}\rangle = \Delta\left(W\mathcal{F}\right), \tag{5}$$

where $\Delta\left(W\mathcal{F}\right) = \left\langle W\frac{\delta\mathcal{F}}{\delta t}\right\rangle$. For a desired set of macroscopic properties, there exists a set of related weights, $\boldsymbol{W} = [W_0, W_1, \ldots, W_N]^\intercal$, and the resulting vector of moments can now be defined as $\boldsymbol{U} = \langle \boldsymbol{W}\mathcal{F}\rangle$. It is appealing to rewrite Equation (5) in balance law form by further defining the flux dyad $\boldsymbol{F}_i = \langle v_i \boldsymbol{W}\mathcal{F}\rangle$ and local source term $\boldsymbol{S} = \Delta\left(\boldsymbol{W}\mathcal{F}\right)$, yielding

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_i}{\partial x_i} = \boldsymbol{S}. \tag{6}$$

Equation (6) represents a set of first-order PDEs describing the evolution of macroscopic properties of a fluid. To close the system, the form of the distribution function is restricted to a prescribed form that both maximizes entropy and satisfies a given set of velocity moments present in the solution vector $\boldsymbol{U}$ [19, 20], such that

$$\mathcal{F} = e^{\boldsymbol{\alpha}^\intercal \boldsymbol{W}}. \tag{7}$$

Here, the closure coefficients, $\boldsymbol{\alpha}$, are the Lagrange multipliers obtained from the constrained entropy-maximization problem. Maximum-entropy moment closures lead to globally hyperbolic moment equations [21, 22].

Two members of the maximum-entropy closure hierarchy considered in the present work are the five-moment (Euler) and ten-moment (Gaussian) closures. The weight vectors for the Euler and ten-moment Gaussian models are $\boldsymbol{W}_5 = [m, mv_i, mv_iv_i]^\intercal$ and $\boldsymbol{W}_{10} = [m, mv_i, mv_iv_j]^\intercal$, respectively. The Gaussian closure is well suited to describe viscous, adiabatic flows, despite not having a treatment for heat flux. Higher-order moment closure can be derived within the framework of maximum entropy [23], and although their numerical treatment could also be considered with the DGH scheme discussed here they are considered beyond the scope of the current work.

3 The Discontinuous-Galerkin Hancock Method

3.1 Introduction

First-order hyperbolic balance laws, such as moment methods derived from the kinetic theory of gases, provide models with physical and mathematical advantages over classical techniques. A spatial discretization such as a standard finite-volume method is often used for the numerical solution of such PDEs (see[2, 24] and references therein). The discontinuous-Galerkin method is a strong alternative that allows for high-order spatial accuracy on arbitrary structured or unstructured computational domains, and the potential for high parallel efficiency.

The discontinuous-Galerkin Hancock model, developed by Suzuki and Van Leer [18], utilizes a coupled space-time approach, but can be seen as being in between semi-discrete and fully discrete due to the fact that the test functions for solution representation are only dependent on space. The method is based upon Huynh's upwind moment scheme [25], originally developed for the numerical solution of hyperbolic systems of conservation laws in fully conservative form, and extended in the work of Suzuki and Van Leer [18] to the class of hyperbolic-relaxation equations. The scheme is a one-step method with an intermediate state, providing third-order accuracy in both space and time. Due to the relatively large amount of local computational work required and low amount of communication with neighbouring elements, the DGH method is well suited to provide efficient implementations on large-scale parallel architectures, as demonstrated in the present work.

3.2 Weak Formulation

The DGH method is a coupled space-time method for the solution of hyperbolic balance laws with stiff local source terms. Specifically, weak solutions for PDEs of the form shown in Equation (6) are sought such that the product of the PDE with a chosen, finite set of test functions, integrated over the space-time domain, $\Omega(t) \times T$, should be satisfied. A scalar test function is defined in the same space as the solution $\boldsymbol{U}(x_j, t)$, denoted as $\nu(x_j, t) \in \Omega(t) \times T$. By taking the product of the PDE with an arbitrary test function and integrating over the space-time domain, the weak formulation of the PDE is obtained. This can be written as

$$\iint\limits_{\Omega(t)\times T} \nu(x_j,t)\frac{\partial \boldsymbol{U}}{\partial t}\,\mathrm{d}x_j\mathrm{d}t = -\iint\limits_{\Omega(t)\times T} \nu(x_j,t)\frac{\partial \boldsymbol{F}_i}{\partial x_i}\,\mathrm{d}x_j\mathrm{d}t + \iint\limits_{\Omega(t)\times T} \nu(x_j,t)\boldsymbol{S}\,\mathrm{d}x_j\mathrm{d}t. \tag{8}$$

Here, T is the time interval over $\left[t^n, t^{n+1}\right]$. Solving the weak formulation given by Equation (8) is equivalent to finding $\boldsymbol{U}(x_j, t)$ for a chosen set of test functions $\nu(x_j, t)$. Assuming the domain $\Omega(t)$ can be subdivided into non-overlapping subdomains or cells $\Omega(t)_k$, and constructing test functions that are non-zero over a unique computational element $\Omega(t)_k$, the following terms in Equation (8) can be integrated by parts to provide

$$\int\limits_T \nu(x_j,t)\frac{\partial \boldsymbol{U}}{\partial t}\,\mathrm{d}t = \left[\boldsymbol{U}^{n+1}\nu(x_j,t) - \boldsymbol{U}^n\nu(x_j,t)\right] - \int\limits_T \boldsymbol{U}\frac{\partial}{\partial t}\nu(x_j,t)\,\mathrm{d}t, \tag{9a}$$

$$\int\limits_{\Omega(t)_k} \nu(x_j,t)\frac{\partial \boldsymbol{F}_i}{\partial x_i}\,\mathrm{d}x_j = \int\limits_{\Gamma_k} \nu(x_j,t)\boldsymbol{F}_i\cdot\hat{n}_i\,\mathrm{d}\Gamma - \int\limits_{\Omega(t)_k} \boldsymbol{F}_i\frac{\partial}{\partial x_i}\nu(x_j,t)\,\mathrm{d}x_j, \tag{9b}$$

which removes derivatives of the solution and its flux from the space-time integrals. Here, Γ_k is the boundary of the cell, and $\hat{n}_i$ is the outward unit normal to the boundary. When it is assumed that the spatial domain over the time interval T is fixed (i.e. $\Omega(t) = \Omega$) and the test function is a function of space only, $\nu(x_j, t) = \nu(x_j)$, the weak formulation can be further simplified. By substituting Equations (9a) and (9b) into Equation (8) and applying Fubini's theorem, which allows for the alternation of the order for integration in space and time, the weak formulation becomes

$$\begin{aligned}\int\limits_{\Omega_k} \nu(x_j)\left[\boldsymbol{U}^{n+1} - \boldsymbol{U}^n\right]\,\mathrm{d}x_j = &-\iint\limits_{\Gamma_k\times T} \nu(x_j)\boldsymbol{F}_i\cdot\hat{n}_i\,\mathrm{d}\Gamma\mathrm{d}t + \iint\limits_{\Omega_k\times T} \boldsymbol{F}_i\frac{\partial}{\partial x_i}\nu(x_j)\,\mathrm{d}x_j\mathrm{d}t \\ &+ \iint\limits_{\Omega_k\times T} \nu(x_j)\boldsymbol{S}\,\mathrm{d}x_j\mathrm{d}t.\end{aligned} \tag{10}$$

3.3 Polynomial Representation of the Solution

The individual solution $\boldsymbol{U}_h(x_j, t)|_{\Omega_k}$ in element Ω_k is assumed to be a polynomial function. Considering the Legendre polynomials P up to degree $K = 1$ (corresponding to piecewise, linear functions) and a 3D problem yields

$$\boldsymbol{U}_h(x_j,t)|_{\Omega_k}, \nu(x_j)|_{\Omega_k} \in P^1\left(\Omega_k\right), \tag{11}$$

where $x_j \equiv \boldsymbol{x} = [x, y, z]$ is a position vector in a Cartesian coordinate system and

$$\begin{aligned} P^1\left(\Omega_k\right) &= \operatorname{span}\left\{\phi_0(x_j), \phi_1(x_j), \phi_2(x_j), \phi_3(x_j)\right\} \\ &= \operatorname{span}\left\{1, x - x_{c_k}, y - y_{c_k}, z - z_{c_k}\right\}. \end{aligned} \tag{12}$$

Here, $\boldsymbol{x}_{c_k} = [x_{c_k},\ y_{c_k},\ z_{c_k}]$ is the centroid of the element Ω_k. The numerical solution variables (i.e., the degrees of freedom) are then the cell average value $\overline{\boldsymbol{U}}_k$, and the average value of the x, y, and z components of the solution gradient within each cell $[(\overline{\Delta_x \boldsymbol{U}})_k,\ (\overline{\Delta_y \boldsymbol{U}})_k,\ (\overline{\Delta_z \boldsymbol{U}})_k]$. The assumed solution in cell k is therefore

$$\boldsymbol{U}_k = \overline{\boldsymbol{U}}_k + (\overline{\Delta_x \boldsymbol{U}})_k(x - x_{c_k}) + (\overline{\Delta_y \boldsymbol{U}})_k(y - y_{c_k}) + (\overline{\Delta_z \boldsymbol{U}})_k(z - z_{c_k}). \tag{13}$$

3.4 Update Formulas for the Degrees of Freedom

Inserting the solution representation from Equation (13) into the weak formulation given by Equation (10), and performing integration using each test function yields the following update formulas for the degrees of freedom:

$$\overline{\boldsymbol{U}}_k^{n+1} = \overline{\boldsymbol{U}}_k^{n} - \underbrace{\frac{1}{V_k} \iint\limits_{\partial\Gamma_k \times T} \boldsymbol{F}_i \cdot \hat{n}_i \, \mathrm{d}\Gamma \mathrm{d}t}_{\text{I}} + \underbrace{\frac{1}{V_k} \iint\limits_{\Omega_k \times T} \boldsymbol{S} \, \mathrm{d}x_j \mathrm{d}t}_{\text{II}}, \tag{14}$$

$$\begin{aligned} \begin{bmatrix} \overline{\Delta_x \boldsymbol{U}}_k^{n+1} \\ \overline{\Delta_y \boldsymbol{U}}_k^{n+1} \\ \overline{\Delta_z \boldsymbol{U}}_k^{n+1} \end{bmatrix} = \begin{bmatrix} \overline{\Delta_x \boldsymbol{U}}_k^{n} \\ \overline{\Delta_y \boldsymbol{U}}_k^{n} \\ \overline{\Delta_z \boldsymbol{U}}_k^{n} \end{bmatrix} + \boldsymbol{K}_k \Bigg(& \underbrace{- \iint\limits_{\partial\Gamma_k \times T} \begin{bmatrix} x - x_{c_k} \\ y - y_{c_k} \\ z - z_{c_k} \end{bmatrix} \boldsymbol{F}_i \cdot \hat{n}_i \, \mathrm{d}\Gamma \mathrm{d}t}_{\text{III}} \\ & + \underbrace{\iint\limits_{\Omega_k \times T} \begin{bmatrix} \boldsymbol{F}_x \\ \boldsymbol{F}_y \\ \boldsymbol{F}_z \end{bmatrix} \mathrm{d}x_j \mathrm{d}t}_{\text{IV}} + \underbrace{\iint\limits_{\Omega_k \times T} \begin{bmatrix} x - x_{c_k} \\ y - y_{c_k} \\ z - z_{c_k} \end{bmatrix} \boldsymbol{S} \, \mathrm{d}x_j \mathrm{d}t}_{\text{V}} \Bigg). \end{aligned} \tag{15}$$

Here, V_k is the volume of the cell, $\partial\Gamma_k$ represents the edges along the boundary Γ_k, and $\boldsymbol{K}_k$ is the matrix given by

$$\boldsymbol{K}_k = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}^{-1}, \tag{16}$$

where

$$I_{xx} = \iiint\limits_{\Omega_k} (x - x_{c_k})^2 \,\mathrm{d}x_j, \tag{17a}$$

$$I_{xy} = I_{yx} = \iiint\limits_{\Omega_k} (x - x_{c_k})(y - y_{c_k}) \,\mathrm{d}x_j, \tag{17b}$$

$$I_{xz} = I_{zx} = \iiint\limits_{\Omega_k} (x - x_{c_k})(z - z_{c_k}) \,\mathrm{d}x_j, \tag{17c}$$

$$I_{yy} = \iiint\limits_{\Omega_k} (y - y_{c_k})^2 \,\mathrm{d}x_j, \tag{17d}$$

$$I_{yz} = I_{zy} = \iiint\limits_{\Omega_k} (y - y_{c_k})(z - z_{c_k}) \,\mathrm{d}x_j, \tag{17e}$$

$$I_{zz} = \iiint\limits_{\Omega_k} (z - z_{c_k})^2 \,\mathrm{d}x_j, \tag{17f}$$

are the area moments of inertia of the cell.

Equations (14) and (15) represent the discrete form of Equation (10) and require the evaluation of five integrals (see the terms I-V), three of which involve a coupled volumetric-time domain, referred to as volumetric integrals, and two of which a coupled surface-time domain, referred to as surface integrals. In summary, there are two surface integrals for the boundary flux through the coupled space-time surface of the computational element (I, III), a flux volumetric integral (IV), and two volumetric integrals for the source term (II, V). The numerical approximations considered in the present work for evaluating the integrals I-V are described in Sections 3.5-3.6.

3.5 Evaluation of Volumetric Flux and Source Term Integrals

There are three volume integrals present in the update formulas for the degrees of freedom that must be evaluated; these are the source term (II), the flux (IV), and the first-order moment of the source term (V). The Radau IIA method [26] for source-term time integration is used, due to the stiffness of the source term. This method is an L-stable, two-step, third-order accurate implicit time-marching scheme written as

$$\begin{aligned}
\overline{\boldsymbol{U}}^{n+\frac{1}{3}} &= \overline{\boldsymbol{U}}^{n} + \Delta t \left[\frac{5}{12} \left(\frac{\mathrm{d}\overline{\boldsymbol{U}}}{\mathrm{d}t} \right)^{n+\frac{1}{3}} - \frac{1}{12} \left(\frac{\mathrm{d}\overline{\boldsymbol{U}}}{\mathrm{d}t} \right)^{n+1} \right], \\
\overline{\boldsymbol{U}}^{n+1} &= \overline{\boldsymbol{U}}^{n} + \Delta t \left[\frac{3}{4} \left(\frac{\mathrm{d}\overline{\boldsymbol{U}}}{\mathrm{d}t} \right)^{n+\frac{1}{3}} + \frac{1}{4} \left(\frac{\mathrm{d}\overline{\boldsymbol{U}}}{\mathrm{d}t} \right)^{n+1} \right].
\end{aligned} \tag{18}$$

Therefore, all integral spatial approximations in II, IV, and V are evaluated at the Radau IIA stages.

3.5.1 Volume Integral of the Flux

The temporal integration in integral IV is approximated by the two-point Radau IIA quadrature, such that

$$\iint\limits_{\Omega_k \times T} \boldsymbol{F}_i \,\mathrm{d}x_j \mathrm{d}t \approx \Delta t \int\limits_{\Omega_k} \left[\frac{3}{4} \boldsymbol{F}_i^{n+\frac{1}{3}} + \frac{1}{4} \boldsymbol{F}_i^{n+1} \right] \mathrm{d}x_j. \tag{19}$$

For performing the spatial integration in Equation(19), a Gaussian quadrature rule is used. Cockburn and Shu [27] proved that for a semi-discrete method with a solution representation in polynomial space P^K, a quadrature rule used for the integration in space along edges must be exact for a polynomial of at least degree $2K+1$. Therefore, to achieve third-order accuracy, a two-point Gaussian quadrature rule is sufficient for approximating the spatial integration with the desired accuracy. For a multi-dimensional space a tensor product of quadrature rule is considered. Thus, assuming a 3D canonical reference element $[\zeta, \eta, \mu] \in [-1, 1]^3$, a two-point Gaussian quadrature rule in each coordinate direction is required (i.e., a total of eight points). For illustration, Figure 1 depicts the quadrature points used for evaluating integrals in two spatial dimensions, in which case there are four Gauss quadrature points at each stage of Radau IIA scheme.

Since elements can be any convex polygonal shape, a mapping from the physical domain $[x,\ y,\ z]$ to a computational domain $[\zeta,\ \eta,\ \mu]$ is required. For skewed hexahedral elements, a trilinear mapping, as used in other previous works (e.g., [9]), is chosen for this transformation, such that

$$\begin{aligned}\boldsymbol{x}(\zeta,\eta,\mu) = {} & \frac{1-\zeta}{2}\frac{1-\eta}{2}\frac{1+\mu}{2}\boldsymbol{x}_0 + \frac{1+\zeta}{2}\frac{1-\eta}{2}\frac{1+\mu}{2}\boldsymbol{x}_1 + \frac{1+\zeta}{2}\frac{1+\eta}{2}\frac{1+\mu}{2}\boldsymbol{x}_2 \\ & + \frac{1-\zeta}{2}\frac{1+\eta}{2}\frac{1+\mu}{2}\boldsymbol{x}_3 + \frac{1+\zeta}{2}\frac{1-\eta}{2}\frac{1-\mu}{2}\boldsymbol{x}_4 + \frac{1-\zeta}{2}\frac{1-\eta}{2}\frac{1-\mu}{2}\boldsymbol{x}_5 \\ & + \frac{1-\zeta}{2}\frac{1+\eta}{2}\frac{1-\mu}{2}\boldsymbol{x}_6 + \frac{1+\zeta}{2}\frac{1+\eta}{2}\frac{1-\mu}{2}\boldsymbol{x}_7.\end{aligned} \tag{20}$$

Here, $\boldsymbol{x}_i$ with $i = [0, 1, 2, \ldots, 7]$ are the physical positions of the nodes that make up a single element. The matrix of the coordinate transform from the physical domain to the computational domain is defined as the Jacobian, $\boldsymbol{J} = \frac{\partial x_i}{\partial \zeta_j}$,

$$\boldsymbol{J}(\zeta_j) = \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \mu} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \mu}\frac{\partial z}{\partial \zeta} + \frac{\partial x}{\partial \mu}\frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \mu}\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \mu}\frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \mu}. \tag{21}$$

Each flux volumetric integral in Equation (19) can be evaluated by applying a Gauss quadrature integration over the canonical element $\hat{\Omega}_k$ to which an arbitrary computational cell, Ω_k, is mapped using the trilinear representation. Thus,

$$\int\limits_{\Omega_k} \boldsymbol{F}_i\left(\boldsymbol{U}(x_j,t)\right)\,\mathrm{d}x_j = \int\limits_{\hat{\Omega}_k} \boldsymbol{F}_i\left(\boldsymbol{U}(\zeta_j,t)\right)|\boldsymbol{J}(\zeta_j)|\,\mathrm{d}\zeta_j \approx \sum_{\chi=0}^{7} w_\chi |\boldsymbol{J}(\zeta_j)| \boldsymbol{F}_i\left(\boldsymbol{U}(\zeta_j,t)\right), \tag{22}$$

where χ is the index through the eight-point flux-volume quadrature with w_χ as the corresponding weight, and $|\boldsymbol{J}(\zeta_j)|$ is the determinant of the Jacobian.

Finally, Equation (19) becomes

$$\iint\limits_{\Omega_k \times T} \boldsymbol{F}_i\,\mathrm{d}x_j\mathrm{d}t \approx \Delta t \sum_\chi w_\chi\left|\boldsymbol{J}\left(\zeta_j\right)\right| \left[\frac{3}{4}\boldsymbol{F}_i^{n+\frac{1}{3}} + \frac{1}{4}\boldsymbol{F}_i^{n+1}\right]. \tag{23}$$

Similarly, the volume integral of the flux for the time interval T' over $[t^n, t^{n+\frac{1}{3}}]$ is given by

$$\iint\limits_{\Omega_k \times T'} \boldsymbol{F}_i\,\mathrm{d}x_j\mathrm{d}t \approx \frac{\Delta t}{3} \sum_\chi w_\chi\left|\boldsymbol{J}\left(\zeta_j\right)\right| \left[\frac{1}{2}\boldsymbol{F}_i^{n} + \frac{1}{2}\boldsymbol{F}_i^{n+\frac{1}{3}}\right]. \tag{24}$$

3.5.2 Volume Integral of the Source Term

A midpoint quadrature rule is used for approximating the integral II in Equation(14). Figure 1 depicts the location of these points for a coupled space-time element. A linearization of the source term is performed to make the update equation for the cell-averaged values decoupled from the update formulas for the solution slopes, such that

$$\boldsymbol{S}(\boldsymbol{U}_h(x_i,t)) \approx \boldsymbol{S}(\overline{\boldsymbol{U}}_k(t)) + \left(\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{U}}\right)^n \left(\begin{bmatrix}\overline{\Delta_x \boldsymbol{U}}_k^n \\ \overline{\Delta_y \boldsymbol{U}}_k^n \\ \overline{\Delta_z \boldsymbol{U}}_k^n\end{bmatrix} \cdot \begin{bmatrix} x - x_{c_k} \\ y - y_{c_k} \\ z - z_{c_k}\end{bmatrix}\right). \tag{25}$$

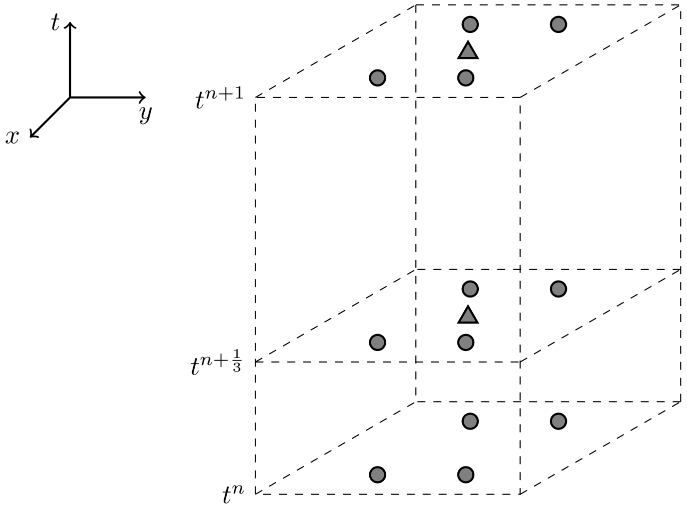


Figure 1: Schematic detailing volume integrals performed using Gaussian quadrature in space and Radau IIA in time. Quadrature points used for the source-term integration are presented as triangles (▲), while points used for flux integration are designated with bullets (•). Calculated states in time are shown with dashed lines.

Application of Radau IIA scheme in combination with the space integration for the source term yields

$$\iint\limits_{\Omega_k\times T} \boldsymbol{S}\,\mathrm{d}x_j\mathrm{d}t \approx V_k\Delta t\left[\frac{3}{4}\boldsymbol{S}(\overline{\boldsymbol{U}}_k^{n+\frac{1}{3}}) + \frac{1}{4}\boldsymbol{S}(\overline{\boldsymbol{U}}_k^{n+1})\right]. \tag{26}$$

3.5.3 Volume Integral of the Moment of the Source Term

The evaluation of integral V in Equation (15) is performed by making use of the linearization for the source term given by Equation (25). Performing analytical spatial integration, the time integration is approximated with Radau IIA quadrature, yielding

$$\begin{aligned}\iint\limits_{\Omega_k\times T} (x-x_{c_k})\,\boldsymbol{S}\,\mathrm{d}x_j\mathrm{d}t \approx{}& \frac{3\Delta t}{4}\frac{\partial\boldsymbol{S}}{\partial\boldsymbol{U}}^{n+\frac{1}{3}}\left(I_{xx}\overline{\Delta_x\boldsymbol{U}}_k^{n+\frac{1}{3}} + I_{xy}\overline{\Delta_y\boldsymbol{U}}_k^{n+\frac{1}{3}} + I_{xz}\overline{\Delta_z\boldsymbol{U}}_k^{n+\frac{1}{3}}\right)\\ &+ \frac{\Delta t}{4}\frac{\partial\boldsymbol{S}}{\partial\boldsymbol{U}}^{n+1}\left(I_{xx}\overline{\Delta_x\boldsymbol{U}}_k^{n+1} + I_{xy}\overline{\Delta_y\boldsymbol{U}}_k^{n+1} + I_{xz}\overline{\Delta_z\boldsymbol{U}}_k^{n+1}\right),\end{aligned} \tag{27a}$$

$$\begin{aligned}\iint\limits_{\Omega_k\times T} (y-y_{c_k})\,\boldsymbol{S}\,\mathrm{d}x_j\mathrm{d}t \approx{}& \frac{3\Delta t}{4}\frac{\partial\boldsymbol{S}}{\partial\boldsymbol{U}}^{n+\frac{1}{3}}\left(I_{yx}\overline{\Delta_x\boldsymbol{U}}_k^{n+\frac{1}{3}} + I_{yy}\overline{\Delta_y\boldsymbol{U}}_k^{n+\frac{1}{3}} + I_{yz}\overline{\Delta_z\boldsymbol{U}}_k^{n+\frac{1}{3}}\right)\\ &+ \frac{\Delta t}{4}\frac{\partial\boldsymbol{S}}{\partial\boldsymbol{U}}^{n+1}\left(I_{yx}\overline{\Delta_x\boldsymbol{U}}_k^{n+1} + I_{yy}\overline{\Delta_y\boldsymbol{U}}_k^{n+1} + I_{yz}\overline{\Delta_z\boldsymbol{U}}_k^{n+1}\right),\end{aligned} \tag{27b}$$

$$\begin{aligned}\iint\limits_{\Omega_k\times T} (z-z_{c_k})\,\boldsymbol{S}\,\mathrm{d}x_j\mathrm{d}t \approx{}& \frac{3\Delta t}{4}\frac{\partial\boldsymbol{S}}{\partial\boldsymbol{U}}^{n+\frac{1}{3}}\left(I_{zx}\overline{\Delta_x\boldsymbol{U}}_k^{n+\frac{1}{3}} + I_{zy}\overline{\Delta_y\boldsymbol{U}}_k^{n+\frac{1}{3}} + I_{zz}\overline{\Delta_z\boldsymbol{U}}_k^{n+\frac{1}{3}}\right)\\ &+ \frac{\Delta t}{4}\frac{\partial\boldsymbol{S}}{\partial\boldsymbol{U}}^{n+1}\left(I_{zx}\overline{\Delta_x\boldsymbol{U}}_k^{n+1} + I_{zy}\overline{\Delta_y\boldsymbol{U}}_k^{n+1} + I_{zz}\overline{\Delta_z\boldsymbol{U}}_k^{n+1}\right).\end{aligned} \tag{27c}$$

3.6 Surface Integral Evaluation and Interface-Flux Approximation

Solutions between cells are discontinuous in a DG scheme. In order to obtain the interface flux along the edge $\partial\Gamma_k$, a numerical Riemann solver is used to approximate

$$\boldsymbol{F}_i \cdot \hat{n}_i \approx \widetilde{\boldsymbol{F}}_\xi, \tag{28}$$

where $\widetilde{\boldsymbol{F}}_\xi$ is the vector of computed fluxes normal to the edge $\partial\Gamma_k$. Two examples of popular Riemann solvers (or flux functions, as they are commonly known) are Roe's flux function [28] and the Harten-Lax-Leer-Einfeldt (HLLE) flux function [29, 30].

Non-linear fluxes require that the surface integral I in Equation 14 is estimated by a quadrature rule. The application of the quadrature rule in the space domain provides

$$\iint\limits_{\partial\Gamma_k \times T} \boldsymbol{F}_i \cdot \hat{n}_i \,\mathrm{d}\Gamma\mathrm{d}t \approx \int\limits_T \sum_\xi w_\xi \widetilde{\boldsymbol{F}}_\xi \,\mathrm{d}t, \tag{29}$$

where w_ξ is the weight of the quadrature point ξ. For integration of Equation (29) in time, the midpoint rule is used as proposed by Suzuki [18], thereby evaluating fluxes at $t^{n+\frac{1}{2}}$,

$$\iint\limits_{\partial\Gamma_k \times T} \boldsymbol{F}_i \cdot \hat{n}_i \,\mathrm{d}\Gamma\mathrm{d}t \approx \Delta t \sum_\xi w_\xi \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{2}}. \tag{30}$$

The evaluation of the surface integral III (in the Equation (15) for the slope update) can be performed in the same manner as the approximation of integral I. Thus, the same numerical interface fluxes can be reused, as follows:

$$\iint\limits_{\partial\Gamma_k \times T} \begin{bmatrix} x - x_{c_k} \\ y - y_{c_k} \\ z - z_{c_k} \end{bmatrix} \boldsymbol{F}_i \cdot \hat{n}_i \,\mathrm{d}\Gamma\mathrm{d}t \approx \Delta t \begin{bmatrix} \sum_\xi w_\xi \left(x_\xi - x_{c_k}\right) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{2}} \\ \sum_\xi w_\xi \left(y_\xi - y_{c_k}\right) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{2}} \\ \sum_\xi w_\xi \left(z_\xi - z_{c_k}\right) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{2}} \end{bmatrix}. \tag{31}$$

The (x, y, z) abscissas of the Gaussian quadrature points $\boldsymbol{x}_{q_1}$ and $\boldsymbol{x}_{q_2}$ on the edge of a cell used for evaluating Equations(30) and Equations(31) are found with

$$\boldsymbol{x}_{q_1,q_2} = \boldsymbol{x}_m \pm \frac{\boldsymbol{\ell}}{2\sqrt{3}}, \tag{32}$$

where $\boldsymbol{x}_m$ is the midpoint of the edge, and $\boldsymbol{\ell}$ is the vector containing lengths of the edge. Application of the Radau IIA method require the evaluation of the solution state at the intermediate time $t^{n+\frac{1}{3}}$. Thus, flux evaluations are necessary at $t^{n+\frac{1}{6}}$ and $t^{n+\frac{1}{2}}$. All necessary quadrature points for evaluating flux surface integrals are shown in Figure 2 for the case in two spatial dimensions.

3.6.1 Hancock's Predictor Step

To complete the approximation of the surface integral, the solution at the half time steps, $\boldsymbol{U}_h(x_i, t^{n+\frac{1}{2}})$ and $\boldsymbol{U}_h(x_i, t^{n+\frac{1}{6}})$, for each update are needed as an input to the approximate Riemann solver. To obtain solution values at these points, Hancock's predictor step is used [25] .

The update formula for the cell-average value in Equation (14) can now be altered slightly by removing element-face interactions, yielding

$$\overline{\boldsymbol{U}}_k^{n+\psi} = \overline{\boldsymbol{U}}_k^n - \frac{1}{V_k} \iint\limits_{\partial\Gamma_k \times T'} \hat{\boldsymbol{F}}_i \cdot \hat{n}_i \,\mathrm{d}\Gamma\mathrm{d}t + \frac{1}{V_k} \iint\limits_{\Omega_k \times T'} \boldsymbol{S} \,\mathrm{d}x_j\mathrm{d}t. \tag{33}$$

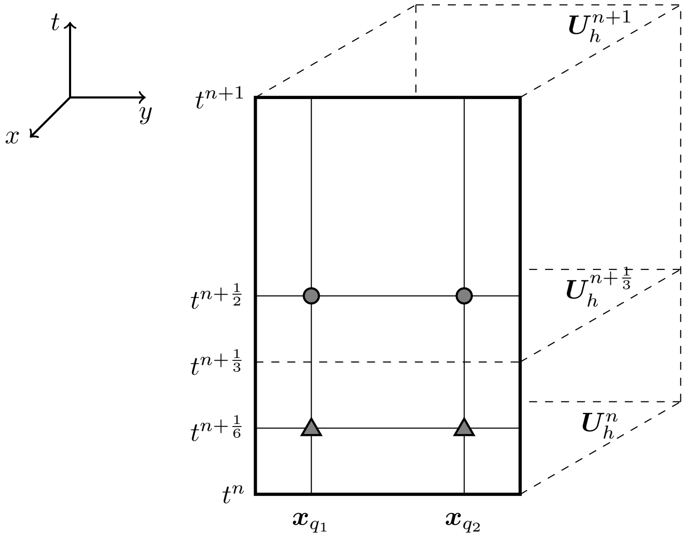


Figure 2: Schematic detailing the surface integral performed using two-point Gaussian quadrature in space and the midpoint rule in time. Quadrature points used for integration over the time step [t^n, $t^{n+\frac{1}{3}}$] are represented as triangles (▲), while bullets (●) designate quadrature points used for integration over the time step [t^n, t^{n+1}]. At each quadrature point, a Riemann solver is used to compute an interface flux. Calculated states are designated with dashed lines.

Here, $\hat{\boldsymbol{F}}_i$ is the flux vector evaluated using only the face quadrature points and interior solution within element Ω_k, and T' is the time interval $\left[t^n, t^{n+\psi}\right]$ with $\psi = \frac{1}{2}$ or $\psi = \frac{1}{6}$. No approximate Riemann solver is needed for these fluxes, as they are evaluated directly using the flux dyad of the system. This predictor step can be simplified further through the use of the implicit Euler method to approximate the temporal integration of the source integral, thus evaluating the source at time $t^{n+\frac{1}{2}}$. The flux integral is evaluated explicitly at time t^n, and Equation (33) can then be written as

$$\overline{\boldsymbol{U}}_k^{n+\psi} = \overline{\boldsymbol{U}}_k^{n} + \frac{\Delta t}{V_k}\left[-\int\limits_{\partial\Gamma_k} \hat{\boldsymbol{F}}_i\left(\boldsymbol{U}_h\left(x_j, t^n\right)\right)\cdot \hat{n}_i \,\mathrm{d}\Gamma + \int\limits_{\Omega_k} \boldsymbol{S}\left(\boldsymbol{U}_h\left(x_j, t^{n+\frac{1}{2}}\right)\right)\mathrm{d}x_j\right]. \tag{34}$$

Once the predicted cell average state at a fractional-time step $\overline{\boldsymbol{U}}_k^{n+\psi}$ is computed, the solution value at any point along the edge $\partial\Gamma_k$ can be found with

$$\boldsymbol{U}_h\left(x_j, t^{n+\psi}\right) = \overline{\boldsymbol{U}}_k^{n+\psi} + \boldsymbol{\phi}_k^n \begin{bmatrix} \overline{\Delta_x \boldsymbol{U}}_k^n \\ \overline{\Delta_y \boldsymbol{U}}_k^n \\ \overline{\Delta_z \boldsymbol{U}}_k^n \end{bmatrix} \cdot \begin{bmatrix} x - x_{c_k} \\ y - y_{c_k} \\ z - z_{c_k} \end{bmatrix}, \tag{35}$$

where $\boldsymbol{\phi}_k^n$ is a vector of slope-limiter values. Some options for slope limiting in multiple dimensions include the Venkatakrishnan limiter [31], which is the multi-dimensional extension to the one dimensional Van Albada limiter [32], or the Barth-Jesperson limiter [33], which is the multi-dimensional extension to the minmod slope limiter [34]. High-order finite-element methods (such as the DGH scheme) that rely on slopes for their assumed solution require a slope limiter to prevent spurious oscillations in regions of high gradients and/or unresolved solution content.

The slope variables used here are the values at time t^n. To evaluate the spatial integration of the source term in Equation (34), a single Gaussian quadrature point at the centroid of the element ($\boldsymbol{x}_{c_k}$, $t^{n+\frac{1}{2}}$) is used. This integral then becomes

$$\int\limits_{\Omega_k} \boldsymbol{S}\,\mathrm{d}x_j \approx V_k \boldsymbol{S}\left(\boldsymbol{U}_h\left(\boldsymbol{x}_{c_k}, t^{n+\psi}\right)\right). \tag{36}$$

From Equation (34) and (36), the state at a fractional time step $\boldsymbol{U}_h(x_i, t^{n+\psi})$ can be found as

$$\begin{aligned}\boldsymbol{U}_h(x_i, t^{n+\psi}) = \overline{\boldsymbol{U}}_k^n + \phi_k^n \begin{bmatrix} \overline{\Delta_x \boldsymbol{U}}_k^n \\ \overline{\Delta_y \boldsymbol{U}}_k^n \\ \overline{\Delta_z \boldsymbol{U}}_k^n \end{bmatrix} \cdot \begin{bmatrix} x - x_{c_k} \\ y - y_{c_k} \\ z - z_{c_k} \end{bmatrix} \\ - \frac{\Delta t}{V_k} \sum_{\partial\Gamma_k \in \partial\Omega_k} \int_{\partial\Gamma_k} \hat{\boldsymbol{F}}_i \left(\boldsymbol{U}_h \left(x_j, t^n \right) \right) \cdot \hat{n}_i \, \mathrm{d}\Gamma + \Delta t \boldsymbol{S} \left(\boldsymbol{U}_h \left(\boldsymbol{x}_{c_k}, t^{n+\psi} \right) \right).\end{aligned} \tag{37}$$

3.7 Discrete Update Formulas

The discrete form of the cell-average update is then

$$\begin{bmatrix} \overline{\boldsymbol{U}}_k^{n+\frac{1}{3}} \\ \overline{\boldsymbol{U}}_k^{n+1} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{U}}_k^n \\ \overline{\boldsymbol{U}}_k^n \end{bmatrix} - \frac{\Delta t}{V_k} \begin{bmatrix} \frac{1}{3} \sum_{\xi} w_{\xi} \widetilde{\boldsymbol{F}}^{n+\frac{1}{6}} \\ \sum_{\xi} w_{\xi} \widetilde{\boldsymbol{F}}^{n+\frac{1}{2}} \end{bmatrix} + \Delta t \begin{bmatrix} \frac{5}{12}\boldsymbol{I} & -\frac{1}{12}\boldsymbol{I} \\ \frac{3}{4}\boldsymbol{I} & \frac{1}{4}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{S}}^{n+\frac{1}{3}} \\ \overline{\boldsymbol{S}}^{n+1} \end{bmatrix}. \tag{38}$$

Though some PDEs will contain source terms that can produce implicit update formulas, and this is the case in the present work, that are simple enough to be evaluated a priori. However, Suzuki [18] advocates using Newton's method for complex, non-linear source terms in the general case. The discrete form of the slope updates is written as

$$
\begin{aligned}
\begin{bmatrix}
\overline{\Delta_x \boldsymbol{U}}_k^{n+\frac{1}{3}} \\
\overline{\Delta_y \boldsymbol{U}}_k^{n+\frac{1}{3}} \\
\overline{\Delta_z \boldsymbol{U}}_k^{n+\frac{1}{3}} \\
\overline{\Delta_x \boldsymbol{U}}_k^{n+1} \\
\overline{\Delta_y \boldsymbol{U}}_k^{n+1} \\
\overline{\Delta_z \boldsymbol{U}}_k^{n+1}
\end{bmatrix}
=
\begin{bmatrix}
\overline{\Delta_x \boldsymbol{U}}_k^{n} \\
\overline{\Delta_y \boldsymbol{U}}_k^{n} \\
\overline{\Delta_z \boldsymbol{U}}_k^{n} \\
\overline{\Delta_x \boldsymbol{U}}_k^{n} \\
\overline{\Delta_y \boldsymbol{U}}_k^{n} \\
\overline{\Delta_z \boldsymbol{U}}_k^{n}
\end{bmatrix}
&+ \Delta t \boldsymbol{\Upsilon}
\begin{bmatrix}
\frac{1}{3} \sum_{\xi} w_\xi (x_\xi - x_{c_k}) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{6}} \\
\frac{1}{3} \sum_{\xi} w_\xi (y_\xi - y_{c_k}) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{6}} \\
\frac{1}{3} \sum_{\xi} w_\xi (z_\xi - z_{c_k}) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{6}} \\
\sum_{\xi} w_\xi (x_\xi - x_{c_k}) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{2}} \\
\sum_{\xi} w_\xi (y_\xi - y_{c_k}) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{2}} \\
\sum_{\xi} w_\xi (z_\xi - z_{c_k}) \widetilde{\boldsymbol{F}}_\xi^{n+\frac{1}{2}}
\end{bmatrix} \\
&+ \Delta t \boldsymbol{\Upsilon}
\begin{bmatrix}
\frac{1}{3} \sum_{\chi} w_\chi \left| \boldsymbol{J}(\zeta_j) \right| \left[\frac{1}{2} (\boldsymbol{F}_x)_\chi^n + \frac{1}{2} (\boldsymbol{F}_x)_\chi^{n+\frac{1}{3}} \right] \\
\frac{1}{3} \sum_{\chi} w_\chi \left| \boldsymbol{J}(\zeta_j) \right| \left[\frac{1}{2} (\boldsymbol{F}_y)_\chi^n + \frac{1}{2} (\boldsymbol{F}_y)_\chi^{n+\frac{1}{3}} \right] \\
\frac{1}{3} \sum_{\chi} w_\chi \left| \boldsymbol{J}(\zeta_j) \right| \left[\frac{1}{2} (\boldsymbol{F}_z)_\chi^n + \frac{1}{2} (\boldsymbol{F}_z)_\chi^{n+\frac{1}{3}} \right] \\
\sum_{\chi} w_\chi \left| \boldsymbol{J}(\zeta_j) \right| \left[\frac{3}{4} (\boldsymbol{F}_x)_\chi^{n+\frac{1}{3}} + \frac{1}{4} (\boldsymbol{F}_x)_\chi^{n+1} \right] \\
\sum_{\chi} w_\chi \left| \boldsymbol{J}(\zeta_j) \right| \left[\frac{3}{4} (\boldsymbol{F}_y)_\chi^{n+\frac{1}{3}} + \frac{1}{4} (\boldsymbol{F}_y)_\chi^{n+1} \right] \\
\sum_{\chi} w_\chi \left| \boldsymbol{J}(\zeta_j) \right| \left[\frac{3}{4} (\boldsymbol{F}_z)_\chi^{n+\frac{1}{3}} + \frac{1}{4} (\boldsymbol{F}_z)_\chi^{n+1} \right]
\end{bmatrix} \\
&+ \Delta t
\begin{bmatrix}
\frac{5}{12} \boldsymbol{I} & -\frac{1}{12} \boldsymbol{I} \\
\frac{3}{4} \boldsymbol{I} & \frac{1}{4} \boldsymbol{I}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{U}}^{n+\frac{1}{3}} \overline{\Delta_x \boldsymbol{U}}_k^{n+\frac{1}{3}} \\
\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{U}}^{n+\frac{1}{3}} \overline{\Delta_y \boldsymbol{U}}_k^{n+\frac{1}{3}} \\
\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{U}}^{n+\frac{1}{3}} \overline{\Delta_z \boldsymbol{U}}_k^{n+\frac{1}{3}} \\
\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{U}}^{n+1} \overline{\Delta_x \boldsymbol{U}}_k^{n+1} \\
\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{U}}^{n+1} \overline{\Delta_y \boldsymbol{U}}_k^{n+1} \\
\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{U}}^{n+1} \overline{\Delta_z \boldsymbol{U}}_k^{n+1}
\end{bmatrix}.
\end{aligned}
\tag{39}
$$

Here, $\boldsymbol{I}$ is the identity matrix $\in \mathbb{R}^{3\times 3}$, and $\boldsymbol{\Upsilon}$ is the block matrix given by

$$
\boldsymbol{\Upsilon} = \begin{bmatrix}
\boldsymbol{K}_k & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{K}_k & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_k
\end{bmatrix}. \tag{40}
$$

4 Numerical Results

4.1 Linear Convection-Relaxation

To demonstrate the convergence accuracy of the third-order DGH scheme, the linear convection-relaxation equation was solved in 3D. This equation is written simply as

$$
\frac{\partial \rho}{\partial t} + v_i \frac{\partial \rho}{\partial x_i} = -\frac{\rho}{\tau}, \tag{41}
$$

Table 1: ℓ_2 error norms and convergence order obtained for solution of the linear convection-relaxation equation on a series of systematically refined computational grids.

# of Elements	ℓ_2 Error	Order
$80 \times 80 \times 80$	9.030×10^{-4}	——
$160 \times 160 \times 160$	1.231×10^{-4}	2.87
$200 \times 200 \times 200$	6.376×10^{-5}	2.95
$300 \times 300 \times 300$	1.911×10^{-5}	2.97
$350 \times 350 \times 350$	1.207×10^{-5}	2.98

where v_i is the convection velocity, and τ is the relaxation time. The exact solution to this PDE is expressed as

$$\rho(x_i,t) = \rho_o(x_i - v_i t)\exp\left(-\frac{t}{\tau}\right). \tag{42}$$

The initial condition chosen for this problem is

$$\rho(x_i,0) = \exp\left(-0.5\left[x^2+y^2+z^2\right]\right), \tag{43}$$

with the non-dimensional convection velocity components given by $v_i = [-2,-2,-2]^\intercal$, and the relaxation time set to $\tau = 1$. The solution is time marched to a final time $t = 3$ on a Cartesian grid discretizing the domain $[-10,10]^3$. The Courant–Friedrichs–Lewy (CFL) number used is 0.3. Note that for this smooth problem, the DGH method is used wihtout the application of a slope-limiter function. Table 1 shows the error norms and convergence order achieved with the linear convection-relaxation equation on a series of 3D grids. Clearly, third-order accuracy is observed.

4.2 Isentropic Vortex

This is a test case originally presented by Balsara and Shu [35], in which a mean flow is disturbed by a vortex. The compressible Euler equations were used to simulate the diagonal movement of the hydrodynamic vortex across a two-dimensional, periodic domain of $[-5,5]^2$. The compressible Euler equations describe a gas in local thermodynamic equilibrium everywhere, and are the lowest member of the maximum-entropy closure hierarchy. They are written as

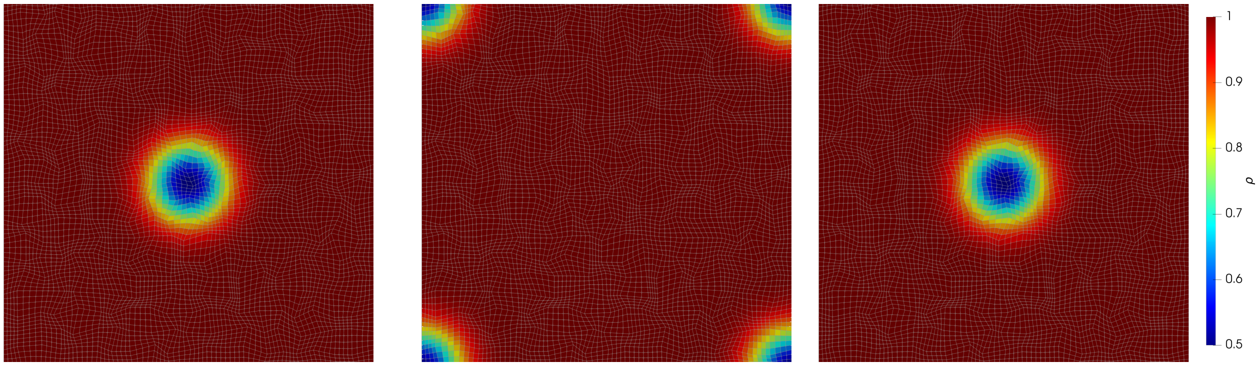
$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho u_i\right) &= 0,\\
\frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_i}\left(\rho u_i u_j + p\delta_{ij}\right) &= 0,\\
\frac{\partial}{\partial t}\left(\frac{\rho u_i u_j}{2} + \frac{p}{\gamma-1}\right) + \frac{\partial}{\partial x_i}\left(u_i\left[\frac{\rho u_j u_j}{2} + \frac{\gamma p}{\gamma-1}\right]\right) &= 0.
\end{aligned} \tag{44}$$

The primitive solution vector for the Euler equations is $\boldsymbol{W} = [\rho, u_x, u_y, p]$, and the non-dimensional initial conditions for the mean flow are given as $\boldsymbol{W}_0 = [1,1,1,1]^\intercal$. The vortex is present due to variations in the primitive variables from the mean flow, such that

$$\begin{aligned}
\delta v = \frac{\epsilon}{2\pi}e^{0.5(1-r^2)}, \quad \delta v_x = -y\delta v, \quad \delta v_y = x\delta v,\\
\delta T = -\frac{(\gamma-1)\,\epsilon^2}{8\gamma\pi^2}e^{(1-r^2)}, \quad \delta S = 0,
\end{aligned} \tag{45}$$

where the temperature and entropy are defined by $T = P/\rho$ and $S = P/\rho^\gamma$, and the heat capacity ratio is that of a diatomic gas, $\gamma = 1.4$. The strength of the vortex is designated by the parameter $\epsilon = 5$. The simulation is run to a final non-dimensional time of $t = 10$, at which point the vortex has fully traversed the grid and arrived back at its original position (see Figure 3).

This problem was simulated on a series of regular Cartesian and irregular grids. The latter grids were



(a) $t = 0$.

(b) $t = 5$.

(c) $t = 10$.

Figure 3: Predicted DGH solution for the isentropic vortex problem at the non-dimensional times $t = 0, 5$ and 10 obtained on a disturbed mesh with 80×80 computational elements. The thin white lines indicate the boundaries of the computational elements.

Table 2: ℓ_2 and ℓ_∞ error norms and solution accuracy order obtained for the isentropic vortex problem.

Mesh	# of Elements	ℓ_2 Error	ℓ_2 Order	ℓ_∞ Error	ℓ_∞ Order
Cartesian	400	1.0230×10^{-1}	——	1.2038×10^{-2}	——
	1600	2.2158×10^{-2}	2.207	6.9568×10^{-3}	0.791
	6400	3.5935×10^{-3}	2.624	1.4390×10^{-3}	2.273
	25600	4.8042×10^{-4}	2.903	1.5611×10^{-4}	3.204
	102400	6.0952×10^{-5}	2.978	1.9928×10^{-5}	2.970
	409600	7.6444×10^{-6}	2.995	2.6144×10^{-6}	2.930
Disturbed	400	1.1549×10^{-1}	——	1.6223×10^{-2}	——
	1600	2.8450×10^{-2}	2.021	1.0585×10^{-2}	0.616
	6400	5.4151×10^{-3}	2.393	3.1871×10^{-3}	1.732
	25600	8.3110×10^{-4}	2.704	5.2243×10^{-4}	2.609
	102400	1.1647×10^{-4}	2.835	9.5139×10^{-5}	2.457

obtained by randomly disturbing the nodes of the coarsest Cartesian mesh such that skewed convex quadrilaterals are generated, and then refine each element using a half-edge approach multiple times to achieve the desired mesh resolution. The simulation was run with a CFL number of 0.3, without use of a slope limiter. The ℓ_2 and ℓ_∞ errors and computed orders of accuracy are shown tabulated for simulations performed on both the Cartesian and disturbed meshes in Table 2. The inspection of Table 2 clearly shows that the ℓ_2 error norms converge with the nominal accuracy in the asymptotic limit on both Cartesian and irregular quadrilateral grids for a set of non-linear PDEs. It is worth emphasising that the DGH method achieves the third-order accuracy despite the fact only linear elements are used.

4.3 Kelvin-Helmholtz Instability

The compressible Euler equations of gas dynamics were again used to simulate a case demonstrating the Kelvin-Helmholtz instability, which occurs at the interface between two fluids with differing velocities. In this case a jump in mass densities is used to make the interface between the two streams obvious. Being able to sharply capture such interfaces in small-scale structures for this case helps demonstrate the low numerical dissipation of the DGH method. For this problem, the instability is triggered at two interfaces, which are slightly perturbed lines at $(y_B, y_T) = (0.25, 0.75) \pm 0.01 \cos(6\pi x)$. The initial conditions are given as

$$\boldsymbol{W}_0 = \begin{cases} \boldsymbol{W}_1, & y_B < y < y_T \\ \boldsymbol{W}_2, & \text{otherwise} \end{cases}, \quad \boldsymbol{W}_1 = \begin{bmatrix} 2 \\ 0.5 \\ 0 \\ 2.5 \end{bmatrix}, \; \boldsymbol{W}_2 = \begin{bmatrix} 1 \\ -0.5 \\ 0 \\ 2.5 \end{bmatrix}. \tag{46}$$

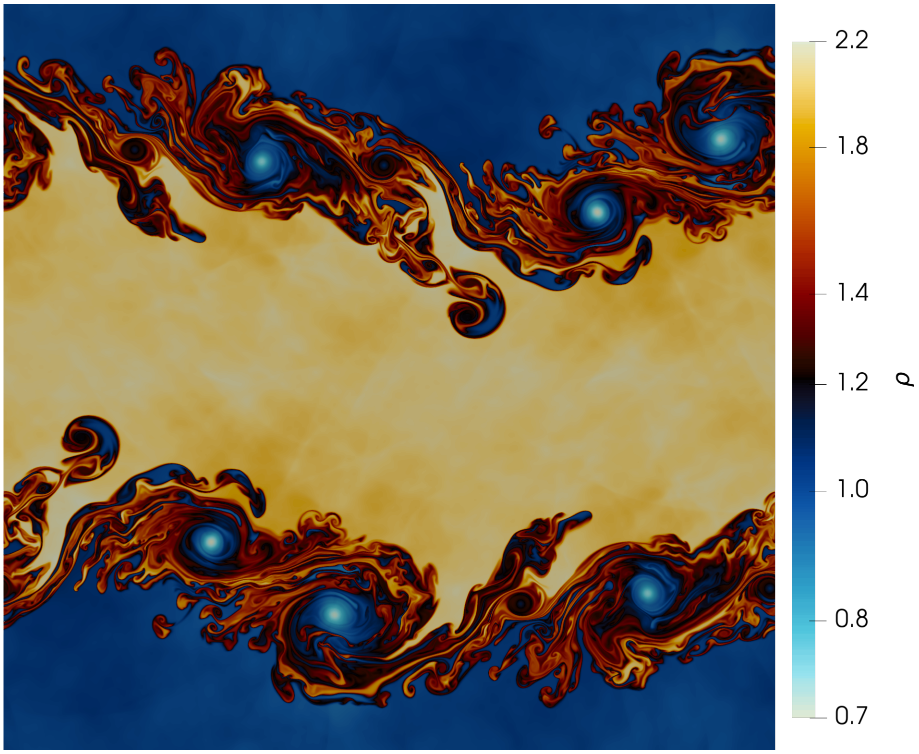


Figure 4: DGH solution for Kelvin-Helmholtz instability at time $t = 2$.

The solution is time-marched to a final non-dimensional time of $t = 2$ on a domain $(x, y) \in [0, 1]^2$ with periodic boundary conditions in both directions. The result, shown in Figure 4, was obtained on a Cartesian computational domain of $40,960,000$ elements using the Roe flux function [28], Venkatakrishnan slope limiter [31], and a CFL number of 0.3. The scheme is efficiently able to resolve small-scale structures in the flow without exhibiting excessive numerical dissipation.

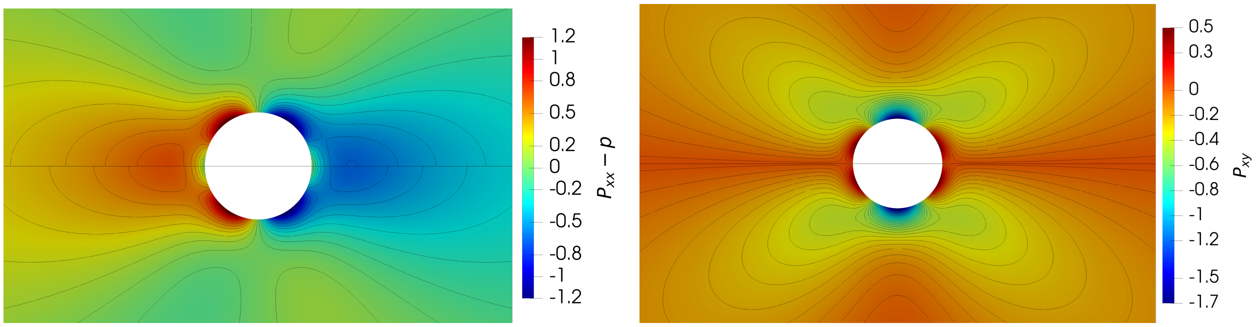
4.4 Stokes Flow

As mentioned, one of the main motivations for the development of the DGH-based implementation is the large-scale efficient solution of moment-closure models from the kinetic theory of gases. Such models offer hyperbolic-relaxation models that can be used as an alternative to the compressible Navier-Stokes equations for continuum flows, while also remaining valid for significant departures from local thermodynamic equilibrium. The ten-moment (Gaussian) closure is written in conservation form as

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_k}\left(\rho u_k\right) &= 0,\\ \frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_k}\left(\rho u_i u_k + P_{ik}\right) &= 0,\\ \frac{\partial}{\partial t}\left(\rho u_i u_j + P_{ij}\right) + \frac{\partial}{\partial x_k}\left(\rho u_i u_j u_k + u_i P_{jk} + u_j P_{ik} + u_k P_{ij}\right) &= -\frac{3P_{ij} - P_{kk}\delta_{ij}}{3\tau},\end{aligned} \tag{47}$$

where the relaxation time τ can be related to viscosity and hydrostatic pressure through $\tau = \mu/p$ [19, 36, 37, 2].

As a demonstration of such a hyperbolic model's ability to accurately describe traditional viscous flows, Figure 5 compares a moment-closure solution for low-Reynolds-number flow past a circular cylinder to the classical analytical Navier-Stokes solution. The fluid simulated is Argon with a density of $\rho = 1.784\,\mathrm{kg/m^3}$, a free-stream velocity of $0.5\,\mathrm{m/s}$, and a pressure of $101\,325\,\mathrm{Pa}$. A constant viscosity of $\mu = 2.117 \times 10^{-5}\,\mathrm{Pa \cdot s}$



(a) The deviatoric x-direction pressure, $P_{xx} - p$.

(b) The shear pressure, P_{xy}.

Figure 5: DGH prediction (top half) and exact solution (bottom half) for Stokes flow past a circular cylinder.

is used. The diameter of the cylinder is set to $1 \times 10^{-5}\,\mathrm{m}$, and the outer-diameter of the domain is 30 times this value. The corresponding Reynolds and Knudsen numbers are $\mathrm{Re} = 0.421$ and $\mathrm{Kn} = 0.0063$, respectively. A relatively uniform semi-cylindrical grid was used to discretize the computational domain for this problem, in which a typical cell covered an angle of $0.05625°$ around the cylinder, and the radial mesh spacing was about 0.45% of the inner-diameter. A reflection boundary condition was imposed at the inner-diameter surface and symmetry plane surfaces, and a free-stream flow was used at the outer boundary. The Roe flux function [28] and a CFL number of 0.3 were used for this simulation. Despite the fact that the flow Mach-number is only $\mathrm{Ma} = 0.0018$, the compressible Gaussian moment-closure solution is accurate without the need for low-Mach-number preconditioning. This, again, shows the very low level of numerical dissipation present in the DGH scheme.

4.5 Taylor-Green Vortex

The Taylor-Green vortex is a suitable problem for evaluating the capabilities of high-order methods. This test case is characterized by its transition to turbulence, and subsequent decay. The incompressible NSE are typically used for the simulation of this test case, and obtaining an accurate result with the ten-moment closure would be a novel result in demonstrating its applicability to the direct numerical simulation of turbulent flows. This subject is treated in more detail in [38].

Because a perfect compressible gas with a heat capacity ratio $\gamma = 1.4$ is simulated, the diatomic extension for the Gaussian closure developed by Hittinger [37] is used to model this problem. The Gaussian closure was originally derived for a monatomic gas with only three translational degrees of freedom, with no internal degrees of freedom (rotational or vibrational). Hittinger's extension accounts for the energy that can be present in these internal degrees of freedom for a diatomic gas. The previously shown ten-moment model is extended to

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_k}\left(\rho u_k\right) = 0, \\
\frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_k}\left(\rho u_i u_k + P_{ik}\right) = 0, \\
\frac{\partial}{\partial t}\left(P_{ij} + \rho u_i u_j\right) + \frac{\partial}{\partial x_k}\left(\rho u_i u_j u_k + u_i P_{jk} + u_j P_{ik} + u_k P_{ij}\right) \\
= -\frac{3P_{ij} - P_{kk}\delta_{ij}}{3\tau_t} - \frac{2\left(P_{kk} - 3E_{\mathrm{rot}}\right)}{15\tau_r}\delta_{ij}, \\
\frac{\partial E_{\mathrm{rot}}}{\partial t} + \frac{\partial}{\partial x_k}\left(u_k E_{\mathrm{rot}}\right) = -\frac{3E_{\mathrm{rot}} - P_{kk}}{5\tau_r}.
\end{aligned} \tag{48}$$

Here, the relaxation times can be related to the gas viscosities with

$$\tau_t = \frac{\mu}{p}, \;\; \tau_r = \frac{15\mu_B}{4p}, \tag{49}$$

where μ is the fluid viscosity, μ_B is the bulk viscosity, and $p = (P_{ii} + 2E_{\mathrm{rot}})/5$.

The domain is given by $(x, y, z) \in [0, 2\pi L]^3$, with a non-dimensional reference length $L = 1$. Periodic boundary conditions are used on all sides of the domain, and the initial conditions are given by

$$\boldsymbol{W}_0 = \begin{bmatrix} \rho \\ U \sin\left(\frac{x}{L}\right) \cos\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right) \\ -U \cos\left(\frac{x}{L}\right) \sin\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right) \\ 0 \\ \frac{1}{\gamma M^2} + \frac{1}{16}\left[\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right)\right]\left[\cos\left(\frac{2z}{L}\right) + 2\right] \end{bmatrix}. \tag{50}$$

Here, the non-dimensional velocity $U = 1$, and density is computed with the equation of state

$$\rho = \frac{\gamma p M^2}{T_{\mathrm{ref}}}, \tag{51}$$

with reference temperature $T_{\mathrm{ref}} = 1$ and a Mach number of $M = 0.1$. The Reynolds number of the flow is defined as $\mathrm{Re} = \frac{\rho_{\mathrm{ref}} U L}{\mu}$, and is equal to 1600. The reference density is taken to be $\rho_{\mathrm{ref}} = 1$, and the appropriate viscosity is then $\mu = \frac{1}{1600}$, with the bulk viscosity approximated as $\mu_B = 3\mu$. The solution is time-marched to final non-dimensional time $t = 20$, using a CFL of 0.1, the HLLE flux function and no slope limiter. To validate results, the integral of the kinetic energy as well as the integral of the enstrophy,

$$E_k = \frac{1}{\rho_{\mathrm{ref}}\Omega} \int\limits_{\Omega} \frac{1}{2} \rho u_j u_j \, \mathrm{d}\Omega, \tag{52a}$$

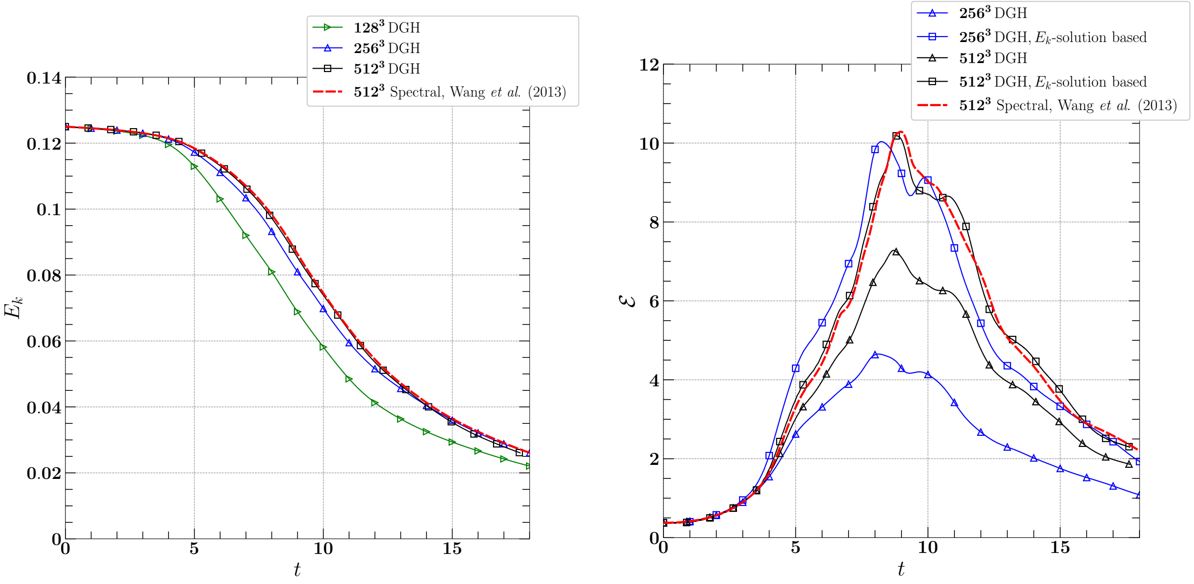
$$\mathcal{E} = \frac{1}{\rho_{\mathrm{ref}}\Omega} \int\limits_{\Omega} \frac{1}{2} \rho \left| \vec{\nabla} \times \vec{u} \right|^2 \mathrm{d}\Omega, \tag{52b}$$

are computed for the entire domain, Ω, during the calculation. For incompressible flows, the temporal evolution of the kinetic energy dissipation rate, $\epsilon = -\frac{\mathrm{d}E_k}{\mathrm{d}t}$, can be related to the enstrophy exactly as

$$\mathcal{E} = \frac{\rho_{\mathrm{ref}}\,\epsilon}{2\mu}, \tag{53}$$

and this is approximately true for compressible flows at a low Mach number.

Figure 6 depicts the predicted enstrophy and kinetic energy by the DGH scheme on a series of computational Cartesian meshes ranging in size from 128^3 to 512^3 hexahedral elements for a total of 2,097,152 and 134,217,728 cells, respectively. Additionally, the same figure provides the reference solution from Wang et al. [1] which was obtained by solving the incompressible NSE on a grid with 512^3 elements and a spectral method. An inspection of Figure 6a shows that the DGH solution appears to be converging towards the reference data of Wang et al., although some slight differences can still be noticed at the end of the dissipation dominated phase (i.e., beyond about $t = 15$). In contrast, Figure 6b shows that the estimated enstropy computed directly with the solution slopes predicted by the DGH scheme are approaching the reference data at a relatively slower rate. However, an indirect estimation of the enstrophy using Equation (53) reveals significantly improved predictions that are much closer to the reference data. This behaviour can be explained by considering that the enstrophy relies on derivatives of the velocity field whereas the kinetic energy depends directly on the velocity field. Recalling that the third-order DGH scheme relies on linear elements for the solution procedure, it is obvious that the solution slopes are approximated with lower accuracy than the solution, and this fact is reflected in the direct evaluation of the enstrophy in this case. Three-dimensional plots obtained on the 512^3 Cartesian grid are shown in Figure 7. The DGH scheme accurately captures a turbulent result, and is able to resolve small scale structures in the flow. The reader is referred to [38] for further discussions regarding the contextualization of the results in the field of direct numerical simulation of turbulent flow.



(a) The kinetic energy, E_k.

(b) The enstrophy, $\mathcal{E}$.

Figure 6: Plot of the predicted kinetic energy and enstrophy by the DGH scheme along with the reference spectral method solution from Wang et al. [1].

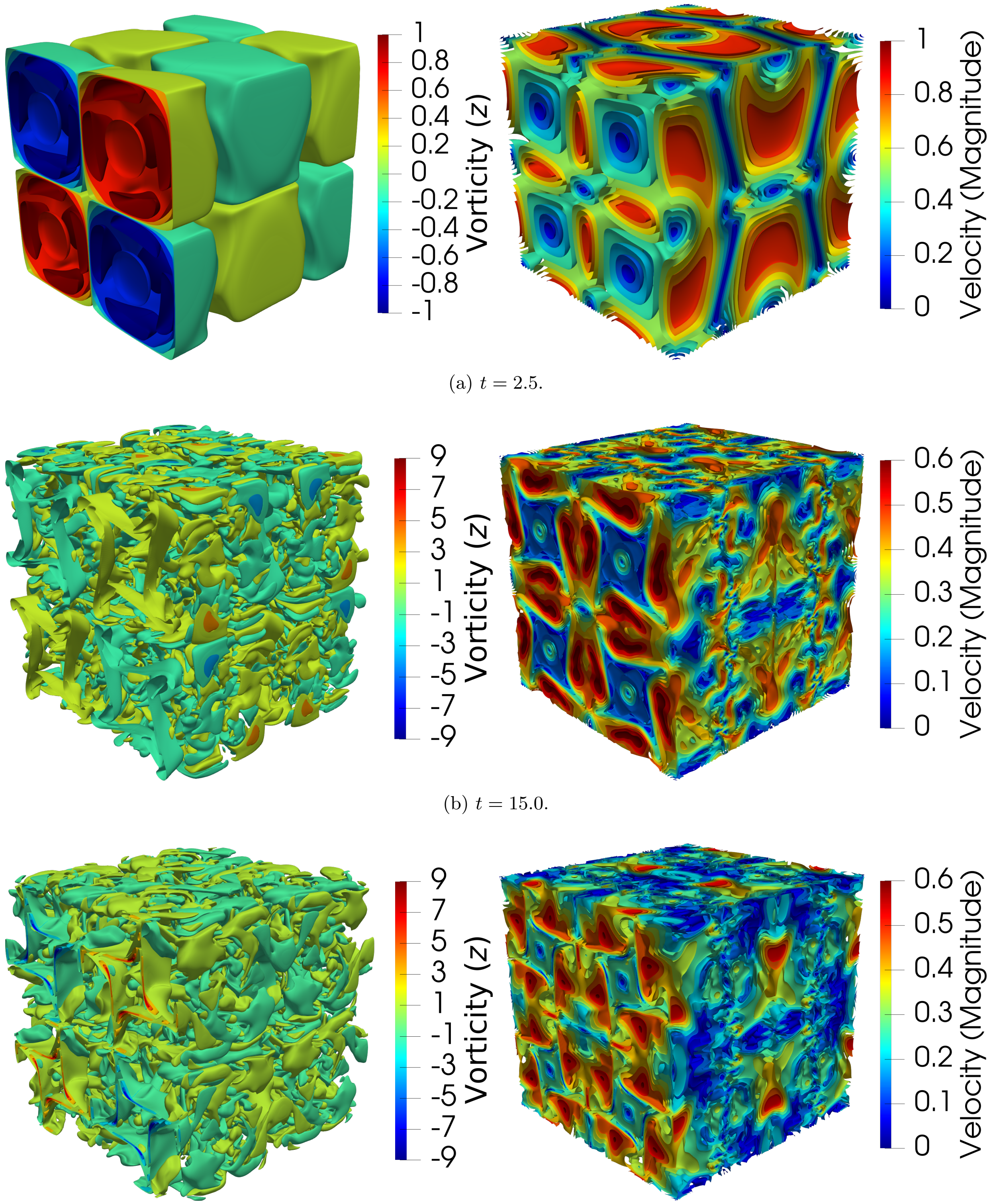
4.6 Parallel Performance

Strong scaling studies using the DGH scheme were performed using two different distributed-clusters for the solution of the Taylor-Green problem. The scheme was shown to achieve a parallel efficiency of about $E_p = 0.96$ on up to 262,144 computational cores of Intel “Knights Landing” 7230 on the Theta KNL system located at the Argonne National Laboratory. Figure 8a shows the results from three different scaling studies of varying grid sizes (64^3, 128^3, 512^3) on the Theta system. Using the 512^3 case, the scheme was also shown (Figure 8b) to achieve a parallel efficiency of about $E_p = 0.97$ on up to 16,384 computational cores of Intel “Skylake” and “Cascade Lake” on the Niagara system, located at the University of Toronto.

5 Conclusion and Future Work

The interest in and derivation of hyperbolic-relaxation equations with stiff local source terms from the kinetic theory of gases is discussed. The formulation of the coupled space-time finite-element discontinuous-Galerkin Hancock method is shown in three dimensions, and results obtained using the first large-scale implementation of the scheme are shared. The third-order accuracy of the scheme is confirmed on multi-dimensional Cartesian and non-Cartesian meshes for linear and non-linear conservation laws. The scheme is used in conjunction with a high-order member of the maximum-entropy closure hierarchy to accurately capture a classical viscous result, which is in great agreement with the exact solution obtained with the Navier-Stokes equations. The scheme and an extension to the high-order maximum-entropy closure is also used to produce a novel result for the direct numerical simulation of a turbulent case. Finally, a strong scaling study of the implementation was performed on two separate distributed clusters, and remained close to perfectly optimal on both architectures, on up to more than a quarter-million computational cores.

As future work, a direct comparison between the DGH and other high-order finite-element schemes in terms of speed and accuracy will be assessed. An extension of the DGH scheme to fourth-order (or higher) temporal and spatial accuracy will be investigated. An extension of the scheme to second-order derivative terms for the solution of PDEs with parabolic terms is also of interest.

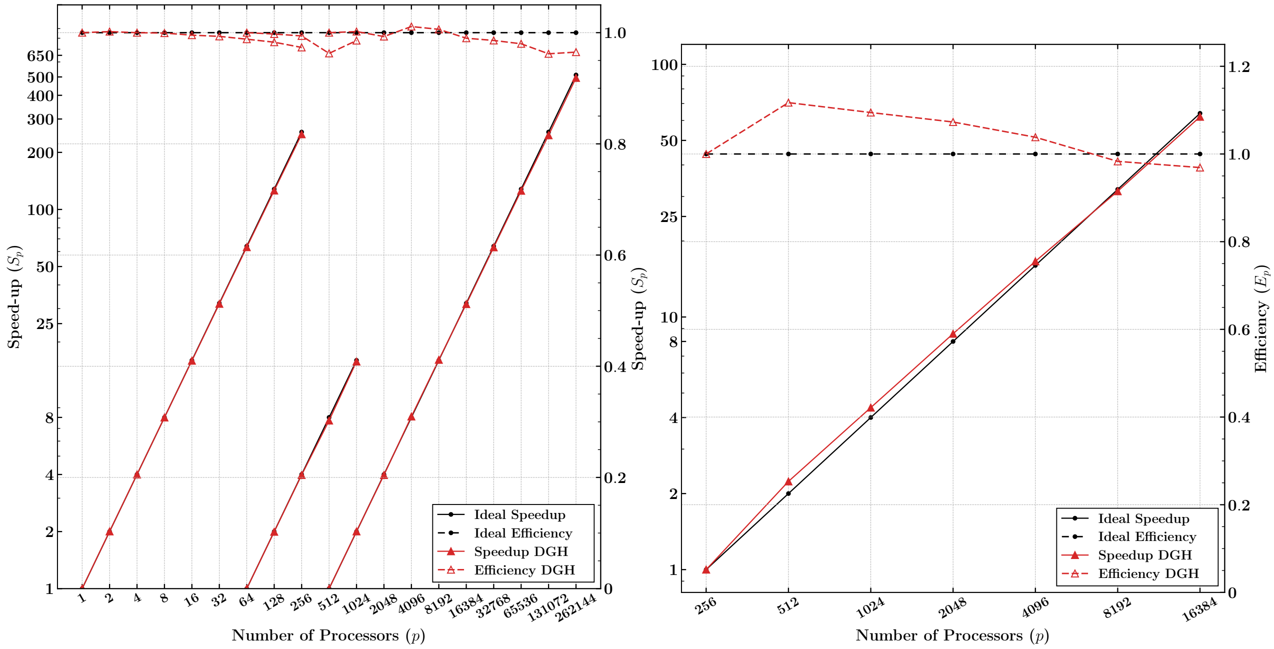


(a) $t = 2.5$.

(b) $t = 15.0$.

(c) $t = 17.5$.

Figure 7: Contour plots of the z-component of vorticity and magnitude of velocity at various non-dimensional times, using a Cartesian grid of size 512^3.



(a) Strong scaling analysis on Theta.

(b) Strong scaling analysis on Niagara.

Figure 8: Strong scaling analysis for the Taylor-Green problem, performed on two distributed clusters.

Acknowledgments

This study was funded by the Natural Science and Engineering Research Council of Canada (NSERC) through grant number RGPIN-2020-06295 and by the Atomic Energy of Canada Limited, under the auspices of the Federal Nuclear Science and Technology Program. The research was conducted at the Canadian Nuclear Laboratories. Computational resources for this work were provided by the Argonne Leadership Computing Facility (ALCF) and the Niagara supercomputer at the SciNet HPC Consortium. The ALCF is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. The SciNet HPC consortium is funded by the Canada Foundation for Innovation; the Government of Ontario; Ontario Research Fund - Research Excellence; and the University of Toronto [39, 40]. The authors are very grateful for this support.

References

[1] Z.J. Wang et al. High-order CFD methods: current status and perspective. International Journal for Numerical Methods in Fluids, 72(8):811–845, 2013.

[2] J. G. McDonald, J. S. Sachdev, and C. P. T. Groth. Application of Gaussian moment closure to micro-scale flows with moving and embedded boundaries. AIAA J., 52:1839–1857, 2014.

[3] A. Harten, B. Enquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. Journal of Computational Physics, 71:231–303, 1987.

[4] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. Journal of Computational Physics, 126:202–228, 1996.

[5] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. Journal of Computational Physics, 144:194–212, 1998.

[6] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. Journal of Computational Physics, 150:97–127, 1999.

[7] J. Balbás and E. Tadmor. Non-oscillatory central schemes for one- and two-dimensional MHD equations. II: High-order semi-discrete schemes. SIAM Journal of Scientific Computing, 28:533–560, 2006.

[8] L. Ivan and C. P.T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. Journal of Computational Physics, 257, Part A(0):830 – 862, 2014.

[9] L. Ivan, H. De Sterck, A. Susanto, and C. P. T. Groth. High-order central ENO finite-volume scheme for hyperbolic conservation laws on three-dimensional cubed-sphere grids. Journal of Computational Physics, 282:157–182, 2015.

[10] B. Cockburn and C-W. Shu. TVB Runge-Kutta local projection discontinous Galerkin finite-element method for conservation laws II: General framework. Math. Comp., 52:411, 1989.

[11] B. Cockburn, S.-Y. Lin, and C-W. Shu. TVB Runge-Kutta local projection discontinous Galerkin finite-element method for conservation laws III: One-dimensional systems. Journal of Computational Physics, 84:90, 1989.

[12] B. Cockburn, S. Hou, and C-W. Shu. TVB Runge-Kutta local projection discontinous Galerkin finite-element method for conservation laws IV: The multidimensional case. Journal of Computational Physics, 54:545, 1990.

[13] Michael Dumbser, Dinshaw S Balsara, Eleuterio F Toro, and Claus-Dieter Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. Journal of Computational Physics, 227(18):8209–8253, 2008.

[14] Z. J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids. Journal of Computational Physics, 178:210–251, 2002.

[15] Y. Liu, M. Vinokur, and Z.J. Wang. Discontinuous spectral difference method for conservation laws on unstructured grids. In Proceedings of the 3rd International Conference on Computational Fluid Dynamics, Toronto, Canada, July 12–16, 2004.

[16] H.T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. AIAA Paper AIAA 20074079, pages 1–42, 01 2007.

[17] G. J. Gassner and A. R. Winters. A Novel Robust Strategy for Discontinuous Galerkin Methods in Computational Fluid Mechanics: Why? When? What? Where? Frontiers in Physics, 8:612, January 2021.

[18] Y. Suzuki. Discontinuous Galerkin Methods for Extended Hydrodynamics. PhD thesis, University of Michigan, 2008.

[19] C. Levermore. Moment closure hierarchies for kinetic theories. Journal of Statistical Physics, 83:1021–1065, 06 1996.

[20] I. Müller and T. Ruggeri. Rational extended thermodynamics. Springer-Verlag, 1998.

[21] M. Junk. Domain of Definition of Levermore's Five-Moment System. Journal of Statistical Physics, 93(5-6):1143–1167, December 1998.

[22] M. Junk and A. Unterreiter. Maximum entropy moment systems and galilean invariance. Continuum Mechanics and Thermodynamics, 14:563–576, 01 2002.

[23] J.G. McDonald and M. Torrilhon. Affordable robust moment closures for CFD based on the maximum-entropy hierarchy. Journal of Computational Physics, 251:500 – 523, 2013.

[24] J. McDonald and C. P. T. Groth. Numerical modeling of micron-scale flows using the gaussian moment closure. Paper 2005-5035, AIAA, June 2005.

[25] H.T. Huynh. An upwind moment scheme for conservation laws. Computational Fluid Dynamics 2004, page 761–766, 2006.

[26] J.D. Lambert. Numerical methods for ordinary differential systems: The initial value problem. J. Wiley & Sons, 2000.

[27] B. Cockburn, S. Hou, and C.W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case. Mathematics of Computation, 54(190):545–581, 1990.

[28] P.L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. Journal of Computational Physics, 43(2):357 – 372, 1981.

[29] A. Harten. High resolution schemes for hyperbolic conservation laws. Journal of Computational Physics, 49(3):357–393, 1983.

[30] B. Einfeldt. On godunov-type methods for gas dynamics. Siam Journal on Numerical Analysis, 25:294–318, 04 1988.

[31] V. Venkatakrishnan. Convergence to steady state solutions of the euler equations on unstructured grids with limiters. Journal of Computational Physics, 118(1):120 – 130, 1995.

[32] G.D. van Albada, B. van Leer, and W.W. Roberts. A comparative study of computational methods in cosmic gas dynamics. Astronomy and Astrophysics, 108:76–84, 04 1982.

[33] T. Barth and D. Jespersen. The design and application of upwind schemes on unstructured meshes. 1989.

[34] B. van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. Journal of Computational Physics, 23(3):276 – 299, 1977.

[35] D.S. Balsara and C.W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. Journal of Computational Physics, 160(2):405–452, 2000.

[36] C. D. Levermore, W. J. Morokoff, and B. T. Nadiga. Moment realizability and the validity of the Navier-Stokes approximation for rarefied gas dynamics. submitted to the Physics of Fluids, October 1995.

[37] J. Hittinger. Foundations for the generalization of the Godunov method to hyperbolic systems with stiff relaxation source terms. PhD thesis, University of Michigan, 2000.

[38] L. Ivan and W. Kaufmann. Direct numerical simulation of turbulent flow using hyperbolic moment methods. In Proceedings of the 11th International Conference on Computational Fluid Dynamics, Hawaii, USA, July 11–15, 2022. Paper ICCFD11-2022-3102.

[39] M. Ponce, R. van Zon, S. Northrup, D. Gruner, J. Chen, F. Ertinaz, A. Fedoseev, L. Groer, F. Mao, B. C. Mundim, M. Nolta, J. Pinto, M. Saldarriaga, V. Slavnic, E. Spence, C.-H. Yu, and W. R. Peltier. Deploying a Top-100 supercomputer for large parallel workloads: the Niagara supercomputer. In PEARC '19: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning), 2019.

[40] C. Loken, D. Gruner, L. Groer, R. Peltier, N. Bunn, M. Craig, T. Henriques, J. Dempsey, C.-H. Yu, J. Chen, L. J. Dursi, J. Chong, S. Northrup, J. Pinto, N. Knecht, and R. van Zon. SciNet: Lessons learned from building a power-efficient Top-20 system and data centre. In J. Phys.: Conf. Ser., 2010.