Kinetic Predicted-Moment Flux Reconstruction for High-Order High-Performance Fluid Simulation: *Supplementary Material*

ZIKE XU, ShanghaiTech University, China
XUAN ZHANG, ShanghaiTech University, China
XIAOPEI LIU, ShanghaiTech University, China

This supplementary material provides derivational and implementation details that complement the main paper. We follow the notation established there and restate only the definitions needed locally for clarity.

1 Spatial Discretization of Hyperbolic Conservation Laws

To solve the Navier-Stokes equations, we begin with the underlying hyperbolic framework, starting from the simplest linear advection equation. Without delving into a formal definition, hyperbolic systems can generally be written in conservation form,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \tag{1}$$

where $\mathbf{U}$ represents the state vector of interest and $\mathbf{F}$ is the flux function of $\mathbf{U}$. For example, in the one-dimensional (1D) case where $F(U) = aU$, the linear advection equation reads

$$\frac{\partial U}{\partial t} + a\frac{\partial U}{\partial x} = 0, \tag{2}$$

where a is the constant advection speed. The Euler equations describe inviscid fluid flow and form a hyperbolic system; viscous effects in the Navier-Stokes equations are incorporated later through a kinetic formulation. The governing equations must be complemented by a spatial discretization. Different numerical methods adopt distinct representations of the solution function $\mathbf{U}(x)$. In 1D, the finite-difference method (FDM) selects a set of grid points $\{x_i\}_{i=1}^{N}$ and their corresponding function values $\{\mathbf{U}_i\}_{i=1}^{N} = \mathbf{U}(x_i)$. The finite-volume method (FVM) divides the domain into N non-overlapping elements Ω_e, $e = 1, 2, \ldots, N$, and stores the average value of the solution function within each element, $\bar{\mathbf{U}}_e = \frac{1}{|\Omega_e|}\int_{\Omega_e} \mathbf{U}(x)\mathrm{d}x$. For the actual discretization, FDM interpolates neighboring samples, while the basic FVM practice is to store a piecewise-constant state within Ω_e; in practice, a piecewise linear reconstruction with a constant gradient per element is used to achieve second-order accuracy [Laney 1998; Toro 2009]. The finite-element method (FEM) and the discontinuous Galerkin method (DG) go further by choosing basis functions $\phi_i(x)$, $i = 1, 2, \ldots, K$, and representing the solution as a linear combination, i.e., in 1D, $\mathbf{U}(x) = \sum_{i=1}^{K} \mathbf{U}_i \phi_i(x)$. DG restricts each basis function to a single element Ω_e, permitting discontinuities at element interfaces [Cockburn et al. 2000; Hesthaven and Warburton 2008]. From a solution-representation perspective, DG offers notable advantages: it requires only element-local information, whereas other methods typically depend on neighboring elements or points, compromising spatial locality and complicating boundary treatments. The nodal discontinuous Galerkin method (NDG) [Hesthaven and Warburton 2008] is a variant of DG that uses Lagrange basis functions to represent the approximate solution, denoted $\mathbf{U}(x)$, consistent with the main text:

$$\mathbf{U}(x) = \sum_{j=1}^{K} \mathbf{U}_j \ell_j(x), \tag{3}$$

where the Lagrange basis function $\ell_j(x)$ is defined as

$$\ell_j(x) = \prod_{k=1, k\neq j}^{K} \frac{x - x_k}{x_j - x_k}, \tag{4}$$

and satisfies $\ell_j(x_i) = \delta_{ij}$, where δ_{ij} is the Kronecker delta function. The point set $\{x_i\}_{i=1}^{K}$ is chosen from quadrature abscissas, such as Gauss-Legendre or Gauss-Lobatto points. In this work we use Gauss-Lobatto points, defined as the roots of the Lobatto polynomials $\mathrm{Lo}_n(x) = L_n(x) - L_{n-2}(x)$ for $n \geq 2$, where $L_n(x)$ is the n-th Legendre polynomial [Huynh 2007]. The following lists the roots of Lo_4 and Lo_5:

Roots of Lo_4	$-1, -\frac{\sqrt{5}}{5}, \frac{\sqrt{5}}{5}, 1$
Roots of Lo_5	$-1, -\frac{\sqrt{21}}{7}, 0, \frac{\sqrt{21}}{7}, 1$

The choice of point set is crucial. We call the chosen points *solution points*, since $\mathbf{U}(x_i) = \mathbf{U}_i$ holds by construction. To map the global solution to a local element, we apply the coordinate transformation

$$r = \frac{2}{L}(x - x_L) - 1, \quad x = \frac{L}{2}(r + 1) + x_L, \tag{5}$$

where $r \in [-1, 1]$, x_L is the left boundary of the element, and L is the element length. We reuse the same notation for the solution function in local coordinates. After the mapping, the solution polynomial $\mathbf{U}(r)$ becomes

$$\mathbf{U}(r) = \sum_{j=1}^{K} \mathbf{U}_j \ell_j(r), \tag{6}$$

which is illustrated in Fig. 1.

The spatial derivative $\frac{\partial}{\partial x}$ is related to the local derivative $\frac{\partial}{\partial r}$ by the chain rule,

$$\frac{\partial}{\partial x} = \frac{\partial r}{\partial x}\frac{\partial}{\partial r} = \frac{2}{L}\frac{\partial}{\partial r}. \tag{7}$$

In practice, we store the solution values in a block vector of length $N \times K \times C$, where C is the number of components in $\mathbf{U}$. This provides both the mathematical representation and an implementation-level organization of the solution.

Authors' Contact Information: Zike Xu, ShanghaiTech University, Shanghai, China, kriaeth@outlook.com; Xuan Zhang, ShanghaiTech University, Shanghai, China, xuanzhang2002@gmail.com; Xiaopei Liu, ShanghaiTech University, Shanghai, China, aurorean.xp@gmail.com.

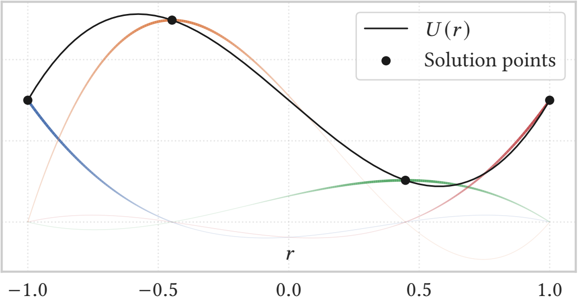

Fig. 1. **Lagrange polynomial basis in 1D.** We show an example in 1D the weighted basis functions $U_j\ell_j(r)$ and the reconstructed degree-$(K-1)$ polynomial at $K = 4$ Gauss-Lobatto solution points.

2 Flux Reconstruction for Hyperbolic Conservation Laws

To numerically solve hyperbolic conservation laws, one first discretizes the equation in space, where the only term that requires evaluation is the divergence term $\nabla \cdot \mathbf{F}(\mathbf{U})$. Note that in solving hyperbolic conservation laws, the flux function $\mathbf{F}$ is defined as a function of $\mathbf{U}$. For example, $\mathbf{F}(U) = \mathbf{a}U$ for a high-dimensional linear advection equation. Based on the element-local solution polynomial $\mathbf{U}(r)$ in Eq. (6), we need to reconstruct the flux function $\mathbf{F}(\mathbf{U})$ to properly represent a globally continuous flux field. This scheme is called *flux reconstruction* [Huynh 2007].

First, we examine a direct approach for flux function representation without reconstruction. Since we have nodal flux values $\mathbf{F}(\mathbf{U}_j)$ at the solution points, we can construct the flux function as

$$\mathbf{F}(r) = \sum_{j=1}^{K} \mathbf{F}(\mathbf{U}_j)\ell_j(r). \tag{8}$$

Differentiating the flux function in 1D, we obtain

$$\frac{\partial \mathbf{F}(r)}{\partial r} = \sum_{j=1}^{K} \mathbf{F}(\mathbf{U}_j)\frac{\partial \ell_j(r)}{\partial r}.$$

When evaluated at r_i, this becomes

$$\frac{\partial \mathbf{F}(r_i)}{\partial r} = \sum_{j=1}^{K} \mathbf{F}(\mathbf{U}_j)\frac{\partial \ell_j(r_i)}{\partial r} = \mathbf{D}\mathbf{F}(\mathbf{U}),$$

where $\mathbf{D}$ is the differentiation square matrix of size K with elements

$$\mathbf{D}_{ij} = \frac{\partial \ell_j(r_i)}{\partial r},$$

where $\mathbf{F}$ is the vector of flux values at the solution points. For fixed solution points, the matrix $\mathbf{D}$ remains constant and can be precomputed.

While this scheme is functional, it has a significant limitation. If there are multiple elements, information is not propagated across element boundaries [Huynh 2007]. To take into account the information from neighboring elements, it is proposed by Huynh [2007] to add a correction function to the reconstructed flux function, as

$$\tilde{\mathbf{F}}(r) = \mathbf{F}(r) + g_{LB}(r)\left[\mathbf{F}^*_{LB} - \mathbf{F}(-1)\right] + g_{RB}(r)\left[\mathbf{F}^*_{RB} - \mathbf{F}(1)\right], \tag{9}$$

where $g_{LB}(r)$ and $g_{RB}(r)$ are the *correction functions*, $\mathbf{F}^*_{LB}$ and $\mathbf{F}^*_{RB}$ are the common flux values at the left and right element boundaries, respectively, which are taken similarly to those in Hesthaven and Warburton [2008]. It is worth noting that the extrapolation of the flux function $\mathbf{F}(r)$ to the element boundaries is performed with a single dot product. To evaluate $\mathbf{F}(-1)$, we can write

$$\mathbf{F}(-1) = \sum_{j=1}^{K} \mathbf{F}(\mathbf{U}_j)\ell_j(-1) = \mathbf{w}_L^\top \mathbf{F}(\mathbf{U}),$$

where $\mathbf{w}_L$ can be precomputed as

$$\mathbf{w}_L = \begin{bmatrix} \ell_1(-1) \\ \ell_2(-1) \\ \vdots \\ \ell_K(-1) \end{bmatrix}, \quad \mathbf{w}_R = \begin{bmatrix} \ell_1(1) \\ \ell_2(1) \\ \vdots \\ \ell_K(1) \end{bmatrix}. \tag{10}$$

There are a few requirements for the correction functions:

- $g_{LB}(-1) = 1$, $g_{LB}(1) = 0$, $g_{RB}(-1) = 0$ and $g_{RB}(1) = 1$
- both $g_{LB}(r)$ and $g_{RB}(r)$ are degree-K polynomials, one degree higher than the solution polynomial ($P = K - 1$).

These requirements ensure that adjacent elements share the same flux value at the element boundary, i.e., $\tilde{\mathbf{F}}_e(x_R) = \tilde{\mathbf{F}}_{e+1}(x_L)$, while $\frac{\partial \tilde{\mathbf{F}}(r)}{\partial r}$ retains the same polynomial degree as $\mathbf{U}(r)$. The corrected flux function $\tilde{\mathbf{F}}(r)$ is illustrated in Fig. 2.

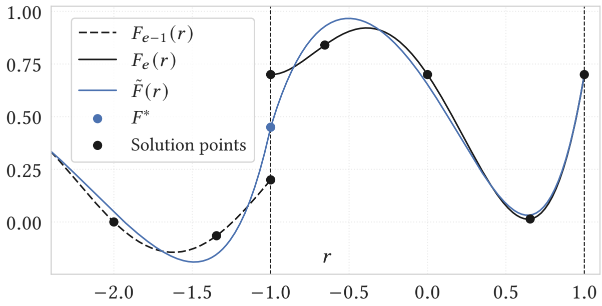

Fig. 2. **Continuous flux reconstruction.** The discontinuous element-local flux $F(r)$ is corrected at each element interface to match the common flux F^*, yielding a globally continuous corrected flux $\tilde{F}(r)$.

Some possible choices of the correction functions are provided in Huynh [2007]; Vincent et al. [2011]. In this work, we choose the *lumping for Lobatto points* (g_2) type in Huynh [2007] for the correction functions, setting $g_{LB}(r) = g_2(r)$ and $g_{RB}(r) = g_2(-r)$ by symmetry. The correction function g_2 is defined as

$$g_2(r) = \frac{K-1}{2K-1}R_{R,K}(r) + \frac{K}{2K-1}R_{R,K-1}(r),$$

where $R_{R,K}$ is the Radau polynomials of order K defined as

$$R_{R,K}(r) = \frac{(-1)^K}{2}\left(L_K(r) - L_{K-1}(r)\right),$$

where L_K is the Legendre polynomial of degree K. The whole chain gives an analytical expression of the correction function g_2 defined on $r \in [-1, 1]$; for $K = 5$,

$$g_2(r) = -\frac{7}{4}r^5 + \frac{35}{16}r^4 + \frac{5}{4}r^3 - \frac{15}{8}r^2 + \frac{3}{16}.$$

To complement this correction function, the solution points are chosen as the Gauss-Lobatto points, where $g_2'(r_i) = 0$ at every solution point except the first ($r_1 = -1$). This is illustrated in Fig. 3.

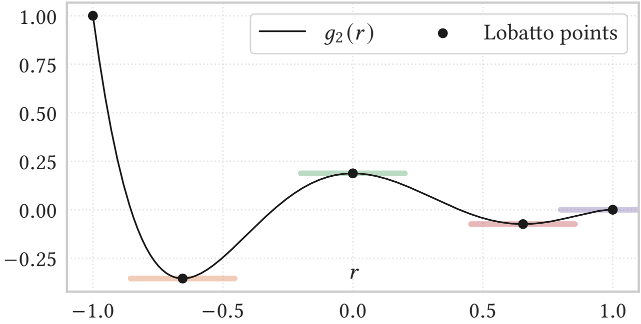

Fig. 3. **The shape of the correction function g_2.** The correction function g_2 defined on $r \in [-1, 1]$ with $K = 5$ under Gauss-Lobatto points.

As for the common flux values, we can take the general Lax-Friedrichs flux. In fluid simulations, we use an arbitrary Riemann solver [Laney 1998; Toro 2009] to compute the flux values at the element boundaries, e.g., the HLLC flux. For the Euler equations, we refer the reader to Section 4. Collecting the K nodal conservative states into a block vector $\mathbf{U} = [\mathbf{U}_1, \ldots, \mathbf{U}_K]^\top$ and similarly $\mathbf{F} = [\mathbf{F}(\mathbf{U}_1), \ldots, \mathbf{F}(\mathbf{U}_K)]^\top$, the resulting update is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{2}{L}\left[\mathbf{D}\mathbf{F} + \mathbf{w}_{LB}\left(\mathbf{F}^*_{LB} - \mathbf{w}_L^\top \mathbf{F}\right) + \mathbf{w}_{RB}\left(\mathbf{F}^*_{RB} - \mathbf{w}_R^\top \mathbf{F}\right)\right] = \mathbf{0}, \quad (11)$$

where

$$\mathbf{w}_{LB} = \begin{bmatrix} g'_{LB}(r_1) \\ g'_{LB}(r_2) \\ \vdots \\ g'_{LB}(r_K) \end{bmatrix}, \quad \mathbf{w}_{RB} = \begin{bmatrix} g'_{RB}(r_1) \\ g'_{RB}(r_2) \\ \vdots \\ g'_{RB}(r_K) \end{bmatrix}.$$

Here, $\mathbf{w}_L$ and $\mathbf{w}_R$ are the pre-computed vectors from Eq. (10). The coefficient $2/L$ arises from the chain rule in Eq. (7). This can be further simplified and denoted as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{2}{L}\left(\mathbf{Q}\mathbf{F} + \mathbf{C}\mathbf{F}^*\right) = \mathbf{0}, \quad (12)$$

where $\mathbf{Q} = \mathbf{D} - \mathbf{C}\mathbf{R}$, and

$$\mathbf{C} = \begin{bmatrix} \mathbf{w}_{LB} & \mathbf{w}_{RB} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{w}_L^\top \\ \mathbf{w}_R^\top \end{bmatrix}, \quad \mathbf{F}^* = \begin{bmatrix} \mathbf{F}^*_{LB} \\ \mathbf{F}^*_{RB} \end{bmatrix}.$$

The concrete computational process for 1D flux reconstruction can be summarized as follows:

(1) Precompute the matrices $\mathbf{w}_L$, $\mathbf{w}_R$, $\mathbf{D}$, and $\mathbf{C}$. Note that these matrices are globally fixed and only need to be computed once at the beginning.
(2) Compute the flux $\mathbf{F}(\mathbf{U}_j)$ at the solution points inside each element.
(3) Extrapolate the flux to the element boundaries $\mathbf{F}(-1)$ and $\mathbf{F}(1)$ using $\mathbf{w}_L$ and $\mathbf{w}_R$ from Eq. (10). For Gauss-Lobatto points this step is trivial since the boundary values are already stored at $\mathbf{F}_1$ and $\mathbf{F}_K$.
(4) Compute the common flux $\mathbf{F}^*$ at all interfaces using a Riemann solver.
(5) Update the solution values using Eq. (12) with a suitable time integration scheme, e.g., the Runge-Kutta method.

Boundary conditions are not addressed in this section. For implementations related to the Euler equations and the Navier-Stokes equations, we refer readers to Kou et al. [2022]; Mengaldo et al. [2014] for stable and convenient approaches.

It is worth mentioning that the flux reconstruction method is conservative with proper common flux functions. For interested readers, we refer to Gassner [2013]; Huynh [2007] for a detailed proof of conservative properties and energy stability. To update in time, we can use the third-order strong stability preserving Runge-Kutta (SSP-RK3) method [Gottlieb et al. 2001]. Denoting by $\mathbf{U}^{(n)}$ the block vector of K nodal states at time level n and by $\mathcal{L}(\mathbf{U}) = -\frac{2}{L}(\mathbf{Q}\mathbf{F} + \mathbf{C}\mathbf{F}^*)$ the spatial operator from Eq. (12), the scheme reads

$$\tilde{\mathbf{U}} = \mathbf{U}^{(n)} + \Delta t \mathcal{L}(\mathbf{U}^{(n)}) \quad (13)$$

$$\tilde{\mathbf{U}} = \frac{3}{4}\mathbf{U}^{(n)} + \frac{1}{4}\left[\tilde{\mathbf{U}} + \Delta t \mathcal{L}(\tilde{\mathbf{U}})\right] \quad (14)$$

$$\mathbf{U}^{(n+1)} = \frac{1}{3}\mathbf{U}^{(n)} + \frac{2}{3}\left[\tilde{\mathbf{U}} + \Delta t \mathcal{L}(\tilde{\mathbf{U}})\right]. \quad (15)$$

These equations are written in a form conducive to implementation: only a single additional array is needed to store the intermediate solution $\tilde{\mathbf{U}}$ if properly implemented.

3 Flux Reconstruction on Tensor-Product Elements

On a 2D quadrilateral element, the K^2 solution points form a tensor-product grid: K points in each coordinate direction r and s. Because each direction acts independently, the multi-dimensional FR update decomposes into $2K$ independent 1D FR problems, one per row and one per column of the grid, and the directional contributions are accumulated at every solution point. The same idea extends to 3D hexahedra with K^3 points and $3K^2$ independent lines.

The hyperbolic conservation laws in 2D can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{0}, \quad (16)$$

where $\mathbf{F}_x$ and $\mathbf{F}_y$ are the columns of the flux tensor $\mathbf{F}$, i.e. the directional fluxes in the x and y directions, respectively. We use tensor-product elements and represent the solution as

$$\mathbf{U}(r, s) = \sum_{i=1}^{K}\sum_{j=1}^{K} \mathbf{U}_{ij}\ell_i(r)\ell_j(s), \quad (17)$$

and the element-local flux polynomial, evaluated from these nodal values, is

$$\mathbf{F}(r, s) = \sum_{i=1}^{K}\sum_{j=1}^{K} \mathbf{F}(\mathbf{U}_{ij})\ell_i(r)\ell_j(s). \quad (18)$$

In 2D, the 1D flux reconstruction scheme is applied independently along each row and column of the tensor-product element. On a tensor-product grid, each solution point on an element edge is also a *flux point*, i.e., a location where the common flux is evaluated. Each face contains K flux points, so the correction is organized face-wise. At a corner, the same flux point contributes once in each coordinate direction. The corrected directional fluxes can be written as

$$\begin{aligned}\tilde{\mathbf{F}}_x(r,s) = \mathbf{F}_x(r,s) &+ \sum_{j=1}^{K} g_{LB}(r)\ell_j(s)\left[\mathbf{F}^*_{x,1j} - \mathbf{F}_x(r_1,s_j)\right] \\ &+ \sum_{j=1}^{K} g_{RB}(r)\ell_j(s)\left[\mathbf{F}^*_{x,Kj} - \mathbf{F}_x(r_K,s_j)\right], \\ \tilde{\mathbf{F}}_y(r,s) = \mathbf{F}_y(r,s) &+ \sum_{i=1}^{K} g_{LB}(s)\ell_i(r)\left[\mathbf{F}^*_{y,i1} - \mathbf{F}_y(r_i,s_1)\right] \\ &+ \sum_{i=1}^{K} g_{RB}(s)\ell_i(r)\left[\mathbf{F}^*_{y,iK} - \mathbf{F}_y(r_i,s_K)\right], \end{aligned} \tag{19}$$

where $\mathbf{F}^*_{x,1j}$, $\mathbf{F}^*_{x,Kj}$, $\mathbf{F}^*_{y,i1}$, and $\mathbf{F}^*_{y,iK}$ denote the common fluxes in the corresponding directions at the face flux points, and hence

$$\tilde{\mathbf{F}}(r,s) = \left[\tilde{\mathbf{F}}_x(r,s)\ \tilde{\mathbf{F}}_y(r,s)\right].$$

Along each row $s = s_j$, $\tilde{\mathbf{F}}_x$ follows the 1D FR reconstruction in the r-direction. Therefore,

$$\frac{\partial \tilde{\mathbf{F}}_x}{\partial r}(r_i,s_j) = \sum_k Q_{ik}\,\mathbf{F}_x(r_k,s_j) + \sum_{b=\pm} C_{ib}\,\mathbf{F}^*_{x,bj}. \tag{20}$$

Thus, each directional derivative is evaluated by applying the 1D FR operator independently along that coordinate line, which is precisely the sum-factorized tensor-product structure. The resulting semi-discrete update equation at solution point (i,j), for each conserved variable $c = 1,\ldots,N_c$, is:

$$\begin{aligned}\frac{\partial U_{ijc}}{\partial t} + \frac{2}{L}\Bigg[&\sum_k Q_{ik}\,F_{x,kjc} + \sum_{b=\pm} C_{ib}\,F^*_{x,bjc} \\ &+ \sum_k Q_{jk}\,F_{y,ikc} + \sum_{b=\pm} C_{jb}\,F^*_{y,ibc}\Bigg] = 0,\end{aligned} \tag{21}$$

where U_{ijc}, $F_{x,ijc}$, and $F_{y,ijc}$ are the c-th components of the conserved state and directional fluxes at solution point (i,j); $F^*_{x,bjc}$ is the c-th component of the common flux in the x direction at boundary $b \in \{-,+\}$ ($r = -1$ or $r = 1$) for row j, and $F^*_{y,ibc}$ the same for the y direction at boundary b ($s = -1$ or $s = 1$) for column i. $\mathbf{Q} \in \mathbb{R}^{K\times K}$ and $\mathbf{C} \in \mathbb{R}^{K\times 2}$ are the same scalar FR matrices as in 1D (Section 2); Q_{ik} sums along the r-axis (first line) of F_x, while Q_{jk} sums along the s-axis (second line) of F_y, reflecting the tensor-product structure. The same idea extends directly to 3D tensor-product hexahedra: there are $3K^2$ independent 1D FR lines and six face-based common-flux corrections, one pair for each coordinate direction.

4 Instantiation for the Isothermal Equations

This section concretizes the abstract FR framework of Sections 2 and 3 for the isothermal, weakly compressible Navier-Stokes equations used throughout the main text. We specify the flux functions, the rotation-invariance property that allows a single common-flux routine to serve all face directions, and the split-form technique that suppresses aliasing instabilities at high polynomial orders.

Isothermal flux. In two dimensions, the conserved state is $\mathbf{U} = [\rho,\ \rho u,\ \rho v]^\top$, and the directional fluxes are

$$\mathbf{F}_x(\mathbf{U}) = \begin{bmatrix}\rho u \\ \rho u^2 + p \\ \rho u v\end{bmatrix}, \qquad \mathbf{F}_y(\mathbf{U}) = \begin{bmatrix}\rho v \\ \rho u v \\ \rho v^2 + p\end{bmatrix}, \tag{22}$$

with the isothermal equation of state $p = \rho c_s^2$, where $c_s = \sqrt{RT}$ is the constant speed of sound.

Common flux and rotation invariance. At each element interface, a *common flux* $\mathbf{F}^*$ couples neighboring elements. Any approximate Riemann solver (e.g., Lax-Friedrichs or HLLC [Toro et al. 1994]) can provide $\mathbf{F}^*$ given the left and right states $\mathbf{U}_L$, $\mathbf{U}_R$ and a face normal $\mathbf{n}$. A Riemann solver typically operates along the x-axis. Extending it to a face with arbitrary unit normal $\mathbf{n} = (\cos\theta,\ \sin\theta)^\top$ exploits the rotation invariance of Eq. (22):

$$\mathbf{F}\cdot\mathbf{n} = \mathbf{T}^{-1}\,\mathbf{F}_x(\mathbf{T}\,\mathbf{U}), \tag{23}$$

where $\mathbf{T}$ rotates the momentum components into the face-aligned frame:

$$\mathbf{T} = \begin{bmatrix}1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta\end{bmatrix},$$

and θ is the angle between the x-axis and $\mathbf{n}$. On axis-aligned grids (as used throughout this paper), $\theta \in \{0,\ \pm\frac{\pi}{2},\ \pi\}$ and $\mathbf{T}$ reduces to a signed permutation of the momentum components.

Split-form anti-aliasing. At high polynomial orders, nonlinear aliasing errors can destabilize the simulation [Jameson et al. 2012]. The split-form remedy [Abe et al. 2018; Kuya et al. 2018] recasts the convective flux divergence as a weighted sum. Taking the x-direction flux as an example, it decomposes into convective and pressure parts:

$$\mathbf{F}_x = \rho u\,\boldsymbol{\phi} + \mathbf{F}^p, \qquad \boldsymbol{\phi} = \begin{bmatrix}1 \\ u \\ v\end{bmatrix}, \qquad \mathbf{F}^p = \begin{bmatrix}0 \\ p \\ 0\end{bmatrix}.$$

The product rule rewrites the convective divergence as

$$\frac{\partial}{\partial r}(\rho u\,\boldsymbol{\phi}) = \frac{1}{2}\frac{\partial}{\partial r}(\rho u\,\boldsymbol{\phi}) + \frac{1}{2}\boldsymbol{\phi}\,\frac{\partial}{\partial r}(\rho u) + \frac{1}{2}\rho u\,\frac{\partial}{\partial r}\boldsymbol{\phi}. \tag{24}$$

In the FR update (Eq. (12)), the term $\mathbf{QF}_x$ is replaced by

$$\mathbf{QF}_x \leftarrow \tfrac{1}{2}\mathbf{D}(\rho u\,\boldsymbol{\phi}) + \tfrac{1}{2}\boldsymbol{\phi}\,\mathbf{D}(\rho u) + \tfrac{1}{2}\rho u\,\mathbf{D}\boldsymbol{\phi} + \mathbf{DF}^p - \mathbf{CR}\,\mathbf{F}_x.$$

Although algebraically equivalent to the original divergence (products with $\boldsymbol{\phi}$ and ρu are taken node-wise across solution points), the split form is significantly more stable at high polynomial orders while preserving conservation [Kuya et al. 2018]. The y-direction follows by symmetry, replacing ρu and r with ρv and s. The kinetic flux formulation of the main text decomposes naturally into inviscid and viscous parts, making this technique directly applicable.

The framework above handles only inviscid fluxes; viscous effects can be treated using the kinetic approach developed in subsequent sections. We also refer readers to Arnold et al. [2002]; Huynh [2009] for conventional viscous treatments in FR.

5 Kinetic Flux Evaluation

This section derives the single-stage fully-discrete update and specifies the kinetic flux formulas the update requires. We adopt the isothermal BGK model throughout: the distribution function satisfies $\partial_t f + \boldsymbol{\xi} \cdot \nabla f = -(f - g)/\tau$, and the Maxwellian equilibrium is expanded to second order in $\mathbf{u}$ (low-Mach regime, $\|\mathbf{u}\| \ll \sqrt{RT}$) [Guo et al. 2013; Krüger et al. 2017]:

$$g(\boldsymbol{\xi}) = \frac{\rho}{(2\pi RT)^{D/2}} \exp\left(-\frac{\|\boldsymbol{\xi}\|^2}{2RT}\right) \times \left[1 + \frac{\boldsymbol{\xi} \cdot \mathbf{u}}{RT} + \frac{(\boldsymbol{\xi} \cdot \mathbf{u})^2}{2(RT)^2} - \frac{\|\mathbf{u}\|^2}{2RT}\right], \tag{25}$$

where $\tau = \nu/(RT)$ is the relaxation time, R the specific gas constant, T the constant temperature, and D the spatial dimension. Macroscopic variables are recovered from velocity-space moments $\langle\cdot\rangle$: $\mathbf{U} = \langle \boldsymbol{\varphi} f\rangle$, $\mathbf{F}_d = \langle \xi_d \boldsymbol{\varphi} f\rangle$, with $\boldsymbol{\varphi} = [1,\ \boldsymbol{\xi}^\top]^\top$.

Single-stage update. Integrating the semi-discrete FR system along direction d over one time step $[0, \Delta t]$ gives

$$\begin{aligned}\mathbf{U}^{(n+1)} &= \mathbf{U}^{(n)} - \frac{2}{L}\int_0^{\Delta t} \left[\mathbf{Q}\,\mathbf{F}_d(t) + \mathbf{C}\,\mathbf{F}^*_d(t)\right] \mathrm{d}t \\ &\approx \mathbf{U}^{(n)} - \frac{2\Delta t}{L}\left[\mathbf{Q}\,\mathbf{F}_d + \mathbf{C}\,\mathbf{F}^*_d\right],\end{aligned} \tag{26}$$

where the second line applies the midpoint rule. In 1D, $\mathbf{F}_d \in \mathbb{R}^{K\times C}$ is the block vector of element-local fluxes at the K solution points and $\mathbf{F}^*_d \in \mathbb{R}^{2\times C}$ collects the common fluxes at the two element boundaries, both evaluated at the half-step $h = \Delta t/2$. For tensor-product elements in 2D and 3D, the same FR update should be applied along each direction, and the resulting directional contributions should be summed as in Section 3. Because the midpoint rule evaluates the integrand at a single time level, the common flux $\mathbf{F}^*_d$ is computed only once per step, whereas a second-order Runge-Kutta integrator would require at least two evaluations. This reduction is significant: the common flux is the most expensive component of FR, not only due to the flux computation itself but also because of the complex inter-element data dependencies it entails (Section 6). Single-stage schemes that exploit this advantage have been explored extensively in the FR community [Babbar et al. 2022; Gassner et al. 2011].

The task is now to construct $\mathbf{F}_d$ and $\mathbf{F}^*_d$ from the known state $\mathbf{U}^{(n)}$. We work at the per-point level: $\mathbf{F}_d(h)$ denotes the N_c-component flux at a single solution point and $\mathbf{F}^*_d(h)$ the common flux at a single interface; their rows assemble the block vectors above. Each per-point flux is a velocity-space moment of the corresponding distribution function:

$$\mathbf{F}_d(h) = \langle \xi_d\, \boldsymbol{\varphi}\, f(h)\rangle\,, \qquad \mathbf{F}^*_d(h) = \langle \xi_d\, \boldsymbol{\varphi}\, f^*(h)\rangle\,. \tag{27}$$

Half-step distribution function. Evaluating Eq. (27) requires the distribution function $f(h)$ at the half-step. Integrating the BGK equation over $[0, h]$ along characteristic lines [Guo et al. 2013; Shu et al. 2014] yields an explicit approximation for the half-step distribution function:

$$f(h) \approx A\, f^s(h) + B\, g(h), \qquad A = \frac{\tau}{h},\ \ B = 1 - \frac{\tau}{h}, \tag{28}$$

where $f^s(h) := g(0) - h\,\boldsymbol{\xi} \cdot \nabla g(0)$ is the propagated distribution function and $g(h) = g(\mathbf{U}(h))$ is the equilibrium at the half-step macroscopic state. The only remaining unknown is $\mathbf{U}(h)$, which sets $g(h)$. Taking moments of Eq. (28) and using the compatibility condition $\langle \boldsymbol{\varphi}(f - g)\rangle = 0$ yields

$$\mathbf{U}(h) = \langle \boldsymbol{\varphi}\,[g(0) - h\,\boldsymbol{\xi} \cdot \nabla g(0)]\rangle \tag{29}$$

which determines it from $g(0)$ and $\nabla g(0)$, both of which the FR discretization provides. Note that Eq. (28) implicitly requires $\tau \le h$, so that $A \in (0, 1]$ and $f(h)$ is a convex combination. More elaborate approximations of $f(h)$ are possible [Guo et al. 2013; Xu 2001], but Eq. (28) is linear in time and the least expensive.

Common distribution function. The common distribution function $f^*(h)$, required by Eq. (27) to compute $\mathbf{F}^*_d(h)$, adopts the same A-B decomposition as Eq. (28) [Xu 2001]. Two simpler alternatives motivate this choice. First, directly upwinding the propagated distribution (KFVS [Chou and Baganoff 1997]) gives a Lax-Friedrichs-like numerical flux with excessive numerical dissipation. Second, evaluating the flux purely from $g(\mathbf{U}^*)$ for a common state $\mathbf{U}^*$ recovers only inviscid dynamics with no physical viscosity. Concretely, at each element interface with unit normal $\mathbf{n}$, each neighboring element first reconstructs its own half-step distribution, denoted by $f_L(h)$ and $f_R(h)$, from Eq. (28). The common state is therefore approximated by upwinding these full half-step distributions through half-range moments:

$$\mathbf{U}^*(h) = \langle \boldsymbol{\varphi}\, f_L(h)\rangle_+ + \langle \boldsymbol{\varphi}\, f_R(h)\rangle_-\,, \tag{30}$$

where $\langle\cdot\rangle_\pm$ denotes integration over the half-spaces $\boldsymbol{\xi} \cdot \mathbf{n} \gtrless 0$. With $\mathbf{U}^*(h)$ fixed, the common distribution is approximated by centering the equilibrium part at $\mathbf{U}^*(h)$ and selecting the propagated part upwind:

$$f^*(h) \approx B\, g(\mathbf{U}^*(h)) + A \begin{cases} f^s_L(h) & \text{if } \boldsymbol{\xi} \cdot \mathbf{n} > 0, \\ f^s_R(h) & \text{if } \boldsymbol{\xi} \cdot \mathbf{n} < 0, \end{cases} \tag{31}$$

where $f^s_L(h)$ and $f^s_R(h)$ are the propagated distributions supplied by the left and right elements, respectively. For axis-aligned grids (the scope of the main text), the interface normal coincides with a coordinate direction, so $\xi_d = \boldsymbol{\xi} \cdot \mathbf{n}$ and $\mathbf{F}^*_d(h) = \langle \xi_d\, \boldsymbol{\varphi}\, f^*(h)\rangle$ follows directly from Eq. (27).

6 Deriving Kinetic Predicted-Moment Flux Reconstruction

Building on the previous section, we establish a complete numerical formulation in which Eqs. (26) and (27) serve as the core update rule, while Eqs. (28) and (31) provide formulas for evaluating the element-local distribution function $f(h)$ and the common distribution function $f^*(h)$.

6.1 Bottlenecks

In the inviscid FR scheme, the corrected flux is built from the element-local flux and the common flux at interface flux points. The only inter-element coupling enters through this common flux, which depends only on reconstructed states. Viscosity changes this picture because the common flux must now encode viscous information in addition to the conservative variables, so its evaluation is no longer state-only.

Why kinetic. A kinetic numerical flux is attractive because the half-range upwinding of the time-evolved distribution already supplies a penalty, while the BGK evolution incorporates viscous effects within the same flux evaluation and avoids a separate lifting-based viscous treatment [Gassner et al. 2007; Xu 2004]. However, the common flux still depends on information originating from both adjacent elements, so the key bottleneck is how to represent and exchange that interfacial viscous information efficiently.

Why naive strategies are unattractive. Several straightforward strategies are possible, but none is appealing in our setting. Reconstructing neighbor gradients on the fly requires substantial additional data movement and repeated differentiation. Storing interface ($6 \times K^2$ directional points on faces) or full-element gradients avoids this recomputation but increases memory footprint and global-memory traffic. These costs are amplified under the fused single-stage tensor-product implementation presented in the main text, where local derivatives are produced and consumed on chip rather than written to global memory. A stored-gradient pipeline would forfeit this advantage (for memory-bound tensor-product FR) by reintroducing an additional global-memory write/read path between predictor and corrector stages. Likewise, discrete-velocity kinetic solvers such as LBFS [Ma et al. 2022; Shu et al. 2014] and conventional GKS [Xu 2001] both fit naturally with gradient-based common-flux evaluation, but their costs are substantial in the present high-order FR setting: LBFS requires velocity-space unrolling, while conventional GKS relies on costly time-dependent flux [Guo et al. 2008].

Why moments. A moment-based representation is attractive in FR for two reasons. (1) It transmits only the information required by the viscous part of the common flux, reducing interface data relative to full gradients. (2) It matches the kinetic decomposition $f = g + f^{\mathrm{neq}}$, allowing the predictor to encode and the corrector to reconstruct the common flux directly in moment space. We therefore formulate the common flux in terms of predicted moments. This compact representation is specifically designed for tensor-product FR, where gradient information is produced locally but is expensive to transmit across the inter-element path.

6.2 Derivations

The single-stage update (Eq. (26)) requires the element-local flux $\mathbf{F}_d$ and the common flux $\mathbf{F}^*_d$, both defined as velocity-space moments (Eq. (27)) of the half-step distribution functions $f(h)$ (Eq. (28)) and $f^*(h)$ (Eq. (31)). The moment-based formulation evaluates these in two stages. The *predictor* computes $\mathbf{U}(h)$ via Eq. (29) and encodes gradient information into the non-equilibrium stress tensor $\boldsymbol{\Pi}(h)$; these are the only quantities exchanged across element interfaces. The *corrector* then evaluates $\mathbf{F}(h)$ and $\mathbf{F}^*(h)$ entirely from the predicted moments, without explicit access to spatial gradients. The corrector exploits the split-form decomposition of Section 4: the flux tensor admits an additive decomposition into equilibrium moments (inviscid) and non-equilibrium moments (viscous).

Before proceeding, we introduce the moment notation for the Mach-number-expanded equilibrium g in Eq. (25). Since all moments are polynomial in $\mathbf{u}$, the half-range integrals required for the common flux evaluation reduce to simple algebraic expressions, avoiding the erf functions that arise from the unexpanded Maxwellian. For the moments $m_{abc} = \left\langle \xi_x^a \xi_y^b \xi_z^c g \right\rangle$, where a, b, c are non-negative integers, the first-order moments correspond to the conservative components:

$$m_{100} = \rho u, \qquad m_{010} = \rho v, \qquad m_{001} = \rho w.$$

The second-order moments are related to the momentum flux:

$$\begin{aligned} m_{200} &= \rho(u^2 + RT), & m_{020} &= \rho(v^2 + RT), & m_{002} &= \rho(w^2 + RT), \\ m_{110} &= \rho uv, & m_{011} &= \rho vw, & m_{101} &= \rho uw. \end{aligned}$$

Finally, the third-order moments are:

$$\begin{aligned} m_{300} &= 3\rho RTu, & m_{030} &= 3\rho RTv, & m_{003} &= 3\rho RTw, \\ m_{210} &= \rho RTv, & m_{120} &= \rho RTu, & m_{201} &= \rho RTw, \\ m_{021} &= \rho RTw, & m_{102} &= \rho RTu, & m_{012} &= \rho RTv, \end{aligned}$$

with $m_{111} = 0$. These moments will enter the viscous corrections below. The viscous stress tensor $\boldsymbol{\Pi}$ is defined as the second-order moment of the non-equilibrium distribution function, i.e., $\Pi_{\alpha\beta} = \left\langle \xi_\alpha \xi_\beta f^{\mathrm{neq}} \right\rangle$. Grad's closure [Grad 1949] offers various reconstruction approaches; here we approximate the non-equilibrium distribution function f^{neq} using the stress tensor as follows:

$$f^{\mathrm{neq}}(\boldsymbol{\xi}) \approx \frac{1}{(2\pi RT)^{D/2}} \exp\left(-\frac{\|\boldsymbol{\xi}\|^2}{2RT}\right) \frac{\Pi_{\alpha\beta}(\xi_\alpha \xi_\beta - RT\delta_{\alpha\beta})}{2(RT)^2}, \tag{32}$$

where $D = 3$ is the spatial dimension, $\delta_{\alpha\beta}$ is the Kronecker delta, and summation over repeated Greek indices is implied. Here, Grad's closure is used only as a decoding closure: given the transmitted stress tensor $\boldsymbol{\Pi}$, it provides a velocity-space representation of f^{neq} whose required full and half-range moments can be evaluated analytically. We *do not* solve a moment system beyond the Navier-Stokes variables or close an independent hierarchy of moment equations. With the above non-equilibrium approximation, all zeroth- and first-order moments of f^{neq} vanish, and the second-order moment recovers the stress tensor $\boldsymbol{\Pi}$. By defining $M_{abc} = \left\langle \xi_x^a \xi_y^b \xi_z^c f^{\mathrm{neq}} \right\rangle$, we have

$$\begin{gathered} M_{000} = M_{100} = M_{010} = M_{001} = 0 \\ M_{200} = \Pi_{xx}, \quad M_{020} = \Pi_{yy}, \quad M_{002} = \Pi_{zz}, \\ M_{110} = \Pi_{xy} = \Pi_{yx}, \quad M_{101} = \Pi_{xz} = \Pi_{zx}, \quad M_{011} = \Pi_{yz} = \Pi_{zy}. \end{gathered}$$

Derivation of the moment-based predictor. With these definitions, the predictor computes the macroscopic state $\mathbf{U}(h)$ and the viscous stress tensor $\boldsymbol{\Pi}(h)$. The predicted state $\mathbf{U}(h)$ follows from Eq. (29). To avoid explicit quadrature over velocity space, we approximate ∇g by finite differences using auxiliary states $\mathbf{U}_d^{\triangle} = \mathbf{U} - u_c h \nabla_d \mathbf{U}$, where $u_c = \epsilon\sqrt{RT}$ is a characteristic velocity (typically with $\epsilon = 0.1$):

$$\begin{aligned} \mathbf{U}(h) &= \langle \boldsymbol{\varphi} g(0) \rangle - h \sum_{d \in \{x,y,z\}} \langle \boldsymbol{\varphi}\, \xi_d\, \nabla_d g(0) \rangle \\ &\approx \langle \boldsymbol{\varphi} g(0) \rangle + \frac{1}{u_c} \sum_{d \in \{x,y,z\}} \left\langle \boldsymbol{\varphi}\, \xi_d \left[g(\mathbf{U}_d^{\triangle}) - g(\mathbf{U}) \right] \right\rangle. \end{aligned}$$

Here, we approximate $\nabla_d g(0)$ by $[g(\mathbf{U}) - g(\mathbf{U}_d^{\triangle})]/(u_c h)$. Defining the perturbed equilibrium moments $m_{d,abc}^{\triangle} := \left\langle \xi_x^a \xi_y^b \xi_z^c g(\mathbf{U}_d^{\triangle}) \right\rangle$ and the difference notation $\Delta_d m_{abc} := (m_{d,abc}^{\triangle} - m_{abc})/u_c$, the macroscopic predictor is obtained by summing the equilibrium moments and their corresponding gradient corrections:

$$\begin{aligned} U_0(h) &= m_{000} + \Delta_x m_{100} + \Delta_y m_{010} + \Delta_z m_{001}, \\ U_1(h) &= m_{100} + \Delta_x m_{200} + \Delta_y m_{110} + \Delta_z m_{101}, \\ U_2(h) &= m_{010} + \Delta_x m_{110} + \Delta_y m_{020} + \Delta_z m_{011}, \\ U_3(h) &= m_{001} + \Delta_x m_{101} + \Delta_y m_{011} + \Delta_z m_{002}. \end{aligned} \tag{33}$$

Substituting Eq. (28) into the definition of $\mathbf{\Pi}(h)$ gives:

$$\Pi_{\alpha\beta}(h) = \left\langle \xi_\alpha \xi_\beta \left\{ A \left[g(0) - h\,\boldsymbol{\xi} \cdot \nabla g(0) \right] + (B-1) g(h) \right\} \right\rangle .$$

With $B = 1 - A$, the predicted stress tensor $\mathbf{\Pi}(h)$ can be written as:

$$\begin{aligned} \Pi_{xx}(h) &= A \left(-\Delta_t m_{200} + \Delta_x m_{300} + \Delta_y m_{210} + \Delta_z m_{201} \right), \\ \Pi_{xy}(h) &= A \left(-\Delta_t m_{110} + \Delta_x m_{210} + \Delta_y m_{120} + \Delta_z m_{111} \right), \\ \Pi_{xz}(h) &= A \left(-\Delta_t m_{101} + \Delta_x m_{201} + \Delta_y m_{111} + \Delta_z m_{102} \right), \\ \Pi_{yy}(h) &= A \left(-\Delta_t m_{020} + \Delta_x m_{120} + \Delta_y m_{030} + \Delta_z m_{021} \right), \\ \Pi_{yz}(h) &= A \left(-\Delta_t m_{011} + \Delta_x m_{111} + \Delta_y m_{021} + \Delta_z m_{012} \right), \\ \Pi_{zz}(h) &= A \left(-\Delta_t m_{002} + \Delta_x m_{102} + \Delta_y m_{012} + \Delta_z m_{003} \right), \end{aligned} \tag{34}$$

where $\Delta_t m_{abc} := m_{abc}(\mathbf{U}(h)) - m_{abc}$ denotes the temporal difference in equilibrium moments. To enforce the traceless property, each component $\Pi_{xx}(h)$, $\Pi_{yy}(h)$, and $\Pi_{zz}(h)$ is reduced by subtracting $\operatorname{tr}(\mathbf{\Pi})/3$, so that $\Pi_{zz}(h)$ does not need to be stored explicitly. The remaining components follow from the symmetry property $\Pi_{\alpha\beta} = \Pi_{\beta\alpha}$. All terms in the predictor are thus formulated as linear combinations of moments. Structurally, the predictor is a moment-space realization of the Cauchy-Kowalewski procedure: the kinetic equation converts temporal derivatives into spatial ones. The predicted stress $\mathbf{\Pi}(h)$ encodes only time-0 spatial gradients. This first-order treatment is standard in the GKS family [Xu 2001] and preserves the formal temporal accuracy of the midpoint-rule update. A higher-order variant would evaluate Eq. (34) on the predicted state $\mathbf{U}(h)$, at the cost of additional instructions in the predictor.

Derivation of the moment-based corrector. In the corrector step, the full flux tensor $\mathbf{F}(h)$ (using a definition different from that in the main text) can be expressed in terms of the predicted macroscopic state $\mathbf{U}(h) = [\rho, \rho\mathbf{u}^\top]^\top$ and the predicted stress tensor $\mathbf{\Pi}(h)$ as:

$$\begin{aligned} \mathbf{F}(h) &= \left\langle \boldsymbol{\varphi} \boldsymbol{\xi}^\top g(\mathbf{U}(h)) \right\rangle + \left\langle \boldsymbol{\varphi} \boldsymbol{\xi}^\top f^{\mathrm{neq}}(\mathbf{\Pi}(h)) \right\rangle \\ &= \begin{bmatrix} m_{100} & m_{010} & m_{001} \\ m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{002} \end{bmatrix} + \begin{bmatrix} M_{100} & M_{010} & M_{001} \\ M_{200} & M_{110} & M_{101} \\ M_{110} & M_{020} & M_{011} \\ M_{101} & M_{011} & M_{002} \end{bmatrix} \\ &= \begin{bmatrix} \rho u & \rho v & \rho w \\ \Pi_{xx} + \rho u^2 + p & \Pi_{xy} + \rho uv & \Pi_{xz} + \rho uw \\ \Pi_{xy} + \rho uv & \Pi_{yy} + \rho v^2 + p & \Pi_{yz} + \rho vw \\ \Pi_{xz} + \rho uw & \Pi_{yz} + \rho vw & \Pi_{zz} + \rho w^2 + p \end{bmatrix}, \end{aligned} \tag{35}$$

where $p = \rho RT$, and all quantities are evaluated at the half-step. This formula exhibits a linear decomposition into an inviscid and a viscous component, recovering the familiar flux formulation of the NS equations. This decomposition also allows us to leverage the split-form technique detailed in Section 4, which is a critical component for ensuring practical numerical stability and computational efficiency.

In the KPM-FR scheme, the common-flux evaluation specified in Eq. (31) can also be cast into a concise, analytically closed form. The common state $\mathbf{U}^*(h)$ is now approximated by aggregating contributions from both sides of the interface as specified in Eq. (30). Without loss of generality, we work in the face-aligned frame and take the interface normal to be the x-direction, so that $\xi_n = \boldsymbol{\xi} \cdot \mathbf{n} = \xi_x$; the other coordinate directions follow from the rotation-invariance argument in Section 4. The half-range moments $\langle \chi f \rangle_\pm = \int_{\xi_x \gtrless 0} \chi f \,\mathrm{d}\boldsymbol{\xi}$ denote the moments analytically integrated over the velocity half-spaces defined by positive and negative normal velocities, respectively. Specifically, we define the half-range equilibrium moments as $m^\pm_{abc} = \left\langle \xi_x^a \xi_y^b \xi_z^c g \right\rangle_\pm$ and, similarly, the non-equilibrium moments as $M^\pm_{abc} = \left\langle \xi_x^a \xi_y^b \xi_z^c f^{\mathrm{neq}} \right\rangle_\pm$.

For notational brevity, we first introduce a set of constants associated with the half-range integration of the Maxwellian distribution function:

$$C_1 = \sqrt{2\pi RT}, \quad C_2 = 2\sqrt{\frac{2RT}{\pi}}, \quad C_3 = \sqrt{\frac{RT}{2\pi}}.$$

Utilizing these constants, some half-range moments associated with the equilibrium and non-equilibrium distribution functions can be expressed as:

$$\begin{aligned} m^\pm_{000} &= \rho \left(\frac{1}{2} \pm \frac{u}{C_1} \right), & M^\pm_{000} &= 0, \\ m^\pm_{100} &= \rho \left(\frac{u}{2} \pm \frac{u^2 + 2RT}{2C_1} \right), & M^\pm_{100} &= \pm \frac{\Pi_{xx}}{2C_1}, \\ m^\pm_{010} &= \rho v \left(\frac{1}{2} \pm \frac{u}{C_1} \right) = v m^\pm_{000}, & M^\pm_{010} &= \pm \frac{\Pi_{xy}}{C_1}, \\ m^\pm_{001} &= \rho w \left(\frac{1}{2} \pm \frac{u}{C_1} \right) = w m^\pm_{000}, & M^\pm_{001} &= \pm \frac{\Pi_{xz}}{C_1}, \\ m^\pm_{200} &= \frac{\rho}{2} \left(u^2 + RT \pm C_2 u \right), & M^\pm_{200} &= \frac{\Pi_{xx}}{2}, \\ m^\pm_{110} &= \rho v \left(\frac{u}{2} \pm C_3 \right), & M^\pm_{110} &= \frac{\Pi_{xy}}{2}, \\ m^\pm_{101} &= \rho w \left(\frac{u}{2} \pm C_3 \right), & M^\pm_{101} &= \frac{\Pi_{xz}}{2}. \end{aligned}$$

We also introduce the shorthand half-range moments $c_L := m^+_{000,L}$ and $c_R := m^-_{000,R}$. Consequently, the resulting approximation for $\mathbf{U}^*(h)$ is:

$$\begin{aligned} U^*_0(h) &= m^+_{000,L} + M^+_{000,L} + m^-_{000,R} + M^-_{000,R} \\ &= c_L + c_R, \\ U^*_1(h) &= m^+_{100,L} + M^+_{100,L} + m^-_{100,R} + M^-_{100,R} \\ &= (2C_1)^{-1} \left[\rho_L (2RT + C_1 u_L + u_L^2) - \rho_R (2RT - C_1 u_R + u_R^2) \right. \\ &\quad \left. + \left(\Pi_{L,xx} - \Pi_{R,xx} \right) \right], \\ U^*_2(h) &= m^+_{010,L} + M^+_{010,L} + m^-_{010,R} + M^-_{010,R} \\ &= c_L v_L + c_R v_R + C_1^{-1} (\Pi_{L,xy} - \Pi_{R,xy}), \\ U^*_3(h) &= m^+_{001,L} + M^+_{001,L} + m^-_{001,R} + M^-_{001,R} \\ &= c_L w_L + c_R w_R + C_1^{-1} (\Pi_{L,xz} - \Pi_{R,xz}). \end{aligned} \tag{36}$$

The non-equilibrium contribution is particularly simple in our case, consisting of linear functions of $\mathbf{\Pi}$.

Applying the same A-B decomposition as in Eq. (31) and using $Af^s = Ag + f^{\text{neq}}$, the common flux $\mathbf{F}^*(h)$ takes the form

$$\begin{aligned} \mathbf{F}^*(h) &= B\left\langle \xi_x \boldsymbol{\varphi} g(\mathbf{U}^*(h))\right\rangle + \left\langle \xi_x \boldsymbol{\varphi}(Ag_L(h) + f_L^{\text{neq}}(h))\right\rangle_+ \\ &\quad + \left\langle \xi_x \boldsymbol{\varphi}(Ag_R(h) + f_R^{\text{neq}}(h))\right\rangle_- , \\ &= B\left[m_{100}^*, m_{200}^*, m_{110}^*, m_{101}^*\right]^\top + \bar{\mathbf{F}}(h), \\ &= B\left[\rho^* u^*, \rho^*(u^{*2} + RT), \rho^* u^* v^*, \rho^* u^* w^*\right]^\top + \bar{\mathbf{F}}(h), \end{aligned} \tag{37}$$

where

$$\bar{\mathbf{F}}(h) := \left\langle \xi_x \boldsymbol{\varphi}(Ag_L(h) + f_L^{\text{neq}}(h))\right\rangle_+ + \left\langle \xi_x \boldsymbol{\varphi}(Ag_R(h) + f_R^{\text{neq}}(h))\right\rangle_-$$

denotes the half-range coupling contribution to the common flux. With this convention, the components of $\bar{\mathbf{F}}$ can be explicitly expressed as:

$$\begin{aligned} \bar{F}_0(h) &= Am_{100,L}^+ + Am_{100,R}^- + M_{100,L}^+ + M_{100,R}^- \\ &= \frac{A}{2C_1}\left[\rho_L(2RT + C_1 u_L + u_L^2) - \rho_R(2RT - C_1 u_R + u_R^2)\right] \\ &\quad + \frac{1}{2C_1}\left(\Pi_{L,xx} - \Pi_{R,xx}\right), \\ \bar{F}_1(h) &= Am_{200,L}^+ + Am_{200,R}^- + M_{200,L}^+ + M_{200,R}^- \\ &= \frac{A}{2}\left[\rho_L(RT + C_2 u_L + u_L^2) + \rho_R(RT - C_2 u_R + u_R^2)\right] \\ &\quad + \frac{1}{2}\left(\Pi_{L,xx} + \Pi_{R,xx}\right), \\ \bar{F}_2(h) &= Am_{110,L}^+ + Am_{110,R}^- + M_{110,L}^+ + M_{110,R}^- \\ &= A\left[\rho_L\left(C_3 v_L + \frac{u_L v_L}{2}\right) + \rho_R\left(-C_3 v_R + \frac{u_R v_R}{2}\right)\right] \\ &\quad + \frac{1}{2}\left(\Pi_{L,xy} + \Pi_{R,xy}\right), \\ \bar{F}_3(h) &= Am_{101,L}^+ + Am_{101,R}^- + M_{101,L}^+ + M_{101,R}^- \\ &= A\left[\rho_L\left(C_3 w_L + \frac{u_L w_L}{2}\right) + \rho_R\left(-C_3 w_R + \frac{u_R w_R}{2}\right)\right] \\ &\quad + \frac{1}{2}\left(\Pi_{L,xz} + \Pi_{R,xz}\right). \end{aligned} \tag{38}$$

This completes the derivation of our single-stage kinetic moment framework. While the core formulation is now established, the practical implementation incorporates two key refinements introduced in the main manuscript: zero-overhead stabilization and oscillation-free tangential transport. We restate them below in the context of our kinetic moment expressions.

To enhance numerical robustness for high-Reynolds-number simulations, we introduce a blending parameter $\lambda \in [0, 1]$ to interpolate between the baseline scheme and the more dissipative KFVS flux. This is achieved by modifying the coefficients A and B in the common-flux evaluation (Eqs. (37) and (38)) as:

$$\tilde{A} = (1-\lambda)A + \lambda, \quad \tilde{B} = (1-\lambda)B. \tag{39}$$

Setting $\lambda = 0.5$ empirically provides a useful balance between stability and accuracy, mirroring the design of the Fromm scheme. Additionally, to mitigate spurious oscillations near the inviscid limit, we adopt the oscillation-free tangential transport strategy, where tangential velocity components are treated as passive scalars advected by the kinetic mass flux m_{100}^*. The tangential flux components in Eq. (37) are replaced by:

$$\begin{aligned} F_2^*(h) &= \bar{F}_2(h) + \frac{B}{2}\left[m_{100}^*(v_L + v_R) + \left|m_{100}^*\right|(v_L - v_R)\right], \\ F_3^*(h) &= \bar{F}_3(h) + \frac{B}{2}\left[m_{100}^*(w_L + w_R) + \left|m_{100}^*\right|(w_L - w_R)\right]. \end{aligned} \tag{40}$$

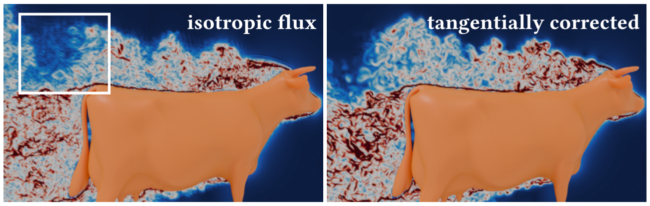

Fig. 4. **Shear-flow oscillation control.** Comparison between the standard isotropic kinetic formulation and the tangentially corrected transport scheme. The isotropic formulation produces spurious oscillations, while the corrected scheme suppresses them.

Notably, these modifications are implemented in an anisotropic manner: the blending parameter λ is only applied to the normal components of the kinetic flux, whereas the tangential terms in the formulas above retain the baseline coefficients A and B (effectively $\lambda = 0$). This preserves the kinetic penalty for diffusion. This parameter-free correction effectively suppresses shear-induced artifacts while maintaining acoustic stability via the fully kinetic normal flux, as illustrated in Fig. 4. We recommend employing both enhancements in practical simulations when robustness across cases is prioritized.

6.3 Two-dimensional formulation

For completeness, we also present the KPM-FR formulas in two dimensions. First, we introduce the moment notation for the equilibrium distribution function in 2D, $m_{ab} = \left\langle \xi_x^a \xi_y^b g\right\rangle$, where a, b are non-negative integers. The relevant moments up to the third order are listed below:

$$\begin{aligned} &m_{00} = \rho, && m_{10} = \rho u, && m_{01} = \rho v, \\ &m_{20} = \rho(u^2 + RT), && m_{02} = \rho(v^2 + RT), && m_{11} = \rho u v, \\ &m_{30} = 3\rho RT u, && m_{03} = 3\rho RT v, && \\ &m_{21} = \rho RT v, && m_{12} = \rho RT u. && \end{aligned}$$

Following the derivation in the 3D case, the non-equilibrium distribution function f^{neq} in 2D follows the same form as Eq. (32) but with $D = 2$, where the indices α, β run over $\{x, y\}$. Defining $M_{ab} = \left\langle \xi_x^a \xi_y^b f^{\text{neq}}\right\rangle$ similarly, we recover the stress tensor components:

$$\begin{gathered} M_{00} = M_{10} = M_{01} = 0, \\ M_{20} = \Pi_{xx}, \quad M_{02} = \Pi_{yy}, \quad M_{11} = \Pi_{xy} = \Pi_{yx}. \end{gathered}$$

With these definitions, the predictor computes the macroscopic state $\mathbf{U}(h)$ and the viscous stress tensor $\boldsymbol{\Pi}(h)$. Specifically, the predicted state follows from the Taylor expansion of the equilibrium distribution function:

$$\mathbf{U}(h) = \langle \boldsymbol{\varphi} g(0)\rangle - h \sum_{d \in \{x,y\}} \langle \boldsymbol{\varphi}\, \xi_d\, \nabla_d g(0)\rangle$$

$$\approx \langle \boldsymbol{\varphi} g(0) \rangle + \frac{1}{u_c} \sum_{d \in \{x,y\}} \left\langle \boldsymbol{\varphi}\, \xi_d \left[g(\mathbf{U}_d^{\triangle}) - g(\mathbf{U}) \right] \right\rangle .$$

Here, $\nabla_d g(0)$ is approximated by $[g(\mathbf{U}) - g(\mathbf{U}_d^{\triangle})]/(u_c h)$ using the auxiliary states $\mathbf{U}_d^{\triangle} = \mathbf{U} - u_c h \nabla_d \mathbf{U}$. The characteristic velocity is given by $u_c = \epsilon\sqrt{RT}$ as in the 3D formulation. Analogously, by defining the 2D perturbed equilibrium moments $m_{d,ab}^{\triangle} := \left\langle \xi_x^a \xi_y^b g(\mathbf{U}_d^{\triangle}) \right\rangle$ and the difference notation $\Delta_d m_{ab} := (m_{d,ab}^{\triangle} - m_{ab})/u_c$, the macroscopic predictor is obtained as:

$$\begin{aligned} U_0(h) &= m_{00} + \Delta_x m_{10} + \Delta_y m_{01}, \\ U_1(h) &= m_{10} + \Delta_x m_{20} + \Delta_y m_{11}, \\ U_2(h) &= m_{01} + \Delta_x m_{11} + \Delta_y m_{02}. \end{aligned} \tag{41}$$

Similarly, the predicted stress tensor is obtained by substituting the reconstructed distribution function into its definition, $\Pi_{\alpha\beta}(h) = \left\langle \xi_\alpha \xi_\beta [A(g(0) - h\, \boldsymbol{\xi} \cdot \nabla g(0)) + (B-1) g(h)] \right\rangle$. With $B = 1 - A$, the predicted stress tensor $\boldsymbol{\Pi}(h)$ can be written as:

$$\begin{aligned} \Pi_{xx}(h) &= A\left(-\Delta_t m_{20} + \Delta_x m_{30} + \Delta_y m_{21}\right), \\ \Pi_{xy}(h) &= A\left(-\Delta_t m_{11} + \Delta_x m_{21} + \Delta_y m_{12}\right), \\ \Pi_{yy}(h) &= A\left(-\Delta_t m_{02} + \Delta_x m_{12} + \Delta_y m_{03}\right), \end{aligned} \tag{42}$$

where $\Delta_t m_{ab} := m_{ab}(\mathbf{U}(h)) - m_{ab}$ denotes the temporal difference in equilibrium moments.

Proceeding to the corrector step, the full flux tensor $\mathbf{F}(h)$ takes a form analogous to the 3D case:

$$\begin{aligned} \mathbf{F}(h) &= \left\langle \boldsymbol{\varphi} \boldsymbol{\xi}^\top g(\mathbf{U}(h)) \right\rangle + \left\langle \boldsymbol{\varphi} \boldsymbol{\xi}^\top f^{\mathrm{neq}}(\boldsymbol{\Pi}(h)) \right\rangle \\ &= \begin{bmatrix} m_{10} & m_{01} \\ m_{20} & m_{11} \\ m_{11} & m_{02} \end{bmatrix} + \begin{bmatrix} M_{10} & M_{01} \\ M_{20} & M_{11} \\ M_{11} & M_{02} \end{bmatrix} \\ &= \begin{bmatrix} \rho u & \rho v \\ \Pi_{xx} + \rho u^2 + \rho RT & \Pi_{xy} + \rho u v \\ \Pi_{xy} + \rho u v & \Pi_{yy} + \rho v^2 + \rho RT \end{bmatrix}, \end{aligned} \tag{43}$$

with all quantities evaluated at the half-step.

In the KPM-FR scheme, the common-flux evaluation specified in Eq. (31) can also be cast into a concise, analytically closed form. We use the same set of constants associated with the half-range integration as in the 3D case:

$$C_1 = \sqrt{2\pi RT}, \quad C_2 = 2\sqrt{\frac{2RT}{\pi}}, \quad C_3 = \sqrt{\frac{RT}{2\pi}}.$$

Utilizing these constants, the 2D half-range moments for the equilibrium and non-equilibrium distributions take a form analogous to the 3D case:

$$\begin{aligned} m_{00}^{\pm} &= \rho\left(\frac{1}{2} \pm \frac{u}{C_1}\right), & M_{00}^{\pm} &= 0, \\ m_{10}^{\pm} &= \rho\left(\frac{u}{2} \pm \frac{u^2 + 2RT}{2C_1}\right), & M_{10}^{\pm} &= \pm\frac{\Pi_{xx}}{2C_1}, \\ m_{01}^{\pm} &= \rho v\left(\frac{1}{2} \pm \frac{u}{C_1}\right) = v m_{00}^{\pm}, & M_{01}^{\pm} &= \pm\frac{\Pi_{xy}}{C_1}, \\ m_{20}^{\pm} &= \frac{\rho}{2}\left(u^2 + RT \pm C_2 u\right), & M_{20}^{\pm} &= \frac{\Pi_{xx}}{2}, \\ m_{11}^{\pm} &= \rho v\left(\frac{u}{2} \pm C_3\right), & M_{11}^{\pm} &= \frac{\Pi_{xy}}{2}. \end{aligned}$$

We also define the shorthand half-range moments $c_L := m_{00,L}^{+}$ and $c_R := m_{00,R}^{-}$. Consequently, the resulting approximation for the common state $\mathbf{U}^*(h)$ is:

$$\begin{aligned} U_0^*(h) &= m_{00,L}^{+} + M_{00,L}^{+} + m_{00,R}^{-} + M_{00,R}^{-} \\ &= c_L + c_R, \\ U_1^*(h) &= m_{10,L}^{+} + M_{10,L}^{+} + m_{10,R}^{-} + M_{10,R}^{-} \\ &= (2C_1)^{-1}\left[\rho_L(2RT + C_1 u_L + u_L^2) - \rho_R(2RT - C_1 u_R + u_R^2) \right. \\ &\quad \left. + \ (\Pi_{L,xx} - \Pi_{R,xx})\right], \\ U_2^*(h) &= m_{01,L}^{+} + M_{01,L}^{+} + m_{01,R}^{-} + M_{01,R}^{-} \\ &= c_L v_L + c_R v_R + C_1^{-1}(\Pi_{L,xy} - \Pi_{R,xy}). \end{aligned} \tag{44}$$

As in Eq. (37), the total common flux $\mathbf{F}^*(h)$ combines the central part based on the common state and the upwinded half-range contributions:

$$\begin{aligned} \mathbf{F}^*(h) &= B\left[m_{10}^*, m_{20}^*, m_{11}^*\right]^\top + \bar{\mathbf{F}}(h) \\ &= B\left[\rho^* u^*, \rho^*(u^{*2} + RT), \rho^* u^* v^*\right]^\top + \bar{\mathbf{F}}(h), \end{aligned} \tag{45}$$

where

$$\bar{\mathbf{F}}(h) := \left\langle \xi_x \boldsymbol{\varphi}(A g_L(h) + f_L^{\mathrm{neq}}(h)) \right\rangle_+ + \left\langle \xi_x \boldsymbol{\varphi}(A g_R(h) + f_R^{\mathrm{neq}}(h)) \right\rangle_-$$

denotes the half-range coupling contribution to the common flux. With the same x-normal convention, the components of $\bar{\mathbf{F}}$ can be explicitly expressed as:

$$\begin{aligned} \bar{F}_0(h) &= A m_{10,L}^{+} + A m_{10,R}^{-} + M_{10,L}^{+} + M_{10,R}^{-} \\ &= \frac{A}{2C_1}\left[\rho_L(2RT + C_1 u_L + u_L^2) - \rho_R(2RT - C_1 u_R + u_R^2)\right] \\ &\quad + \frac{1}{2C_1}\left(\Pi_{L,xx} - \Pi_{R,xx}\right), \\ \bar{F}_1(h) &= A m_{20,L}^{+} + A m_{20,R}^{-} + M_{20,L}^{+} + M_{20,R}^{-} \\ &= \frac{A}{2}\left[\rho_L(RT + C_2 u_L + u_L^2) + \rho_R(RT - C_2 u_R + u_R^2)\right] \\ &\quad + \frac{1}{2}\left(\Pi_{L,xx} + \Pi_{R,xx}\right), \\ \bar{F}_2(h) &= A m_{11,L}^{+} + A m_{11,R}^{-} + M_{11,L}^{+} + M_{11,R}^{-} \\ &= A\left[\rho_L\left(C_3 v_L + \frac{u_L v_L}{2}\right) + \rho_R\left(-C_3 v_R + \frac{u_R v_R}{2}\right)\right] \\ &\quad + \frac{1}{2}\left(\Pi_{L,xy} + \Pi_{R,xy}\right). \end{aligned} \tag{46}$$

References

Yoshiaki Abe, Issei Morinaka, Takanori Haga, Taku Nonomura, Hisaichi Shibata, and Koji Miyaji. 2018. Stable, Non-Dissipative, and Conservative Flux-Reconstruction Schemes in Split Forms. *J. Comput. Phys.* 353 (Jan. 2018), 193–227. doi:10.1016/j.jcp.2017.10.007

Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. 2002. Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems. *SIAM J. Numer. Anal.* 39, 5 (Jan. 2002), 1749–1779. doi:10.1137/S0036142901384162

Arpit Babbar, Sudarshan Kumar Kenettinkara, and Praveen Chandrashekar. 2022. Lax-Wendroff Flux Reconstruction Method for Hyperbolic Conservation Laws. *J. Comput. Phys.* 467 (Oct. 2022), 111423. doi:10.1016/j.jcp.2022.111423

S.Y. Chou and D. Baganoff. 1997. Kinetic Flux–Vector Splitting for the Navier–Stokes Equations. *J. Comput. Phys.* 130, 2 (Jan. 1997), 217–230. doi:10.1006/jcph.1996.5579

Bernardo Cockburn, George E. Karniadakis, Chi-Wang Shu, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, and T. Schlick (Eds.). 2000. *Discontinuous Galerkin Methods: Theory, Computation and Applications.* Lecture Notes in Computational Science and Engineering, Vol. 11. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-59721-3

Gregor Gassner, Frieder Lörcher, and Claus-Dieter Munz. 2007. A Contribution to the Construction of Diffusion Fluxes for Finite Volume and Discontinuous Galerkin Schemes. *J. Comput. Phys.* 224, 2 (June 2007), 1049–1063. doi:10.1016/j.jcp.2006.11.004

Gregor J. Gassner. 2013. A Skew-Symmetric Discontinuous Galerkin Spectral Element Discretization and Its Relation to SBP-SAT Finite Difference Methods. *SIAM Journal on Scientific Computing* 35, 3 (Jan. 2013), A1233–A1253. doi:10.1137/120890144

Gregor J. Gassner, Michael Dumbser, Florian Hindenlang, and Claus-Dieter Munz. 2011. Explicit One-Step Time Discretizations for Discontinuous Galerkin and Finite Volume Schemes Based on Local Predictors. *J. Comput. Phys.* 230, 11 (May 2011), 4232–4247. doi:10.1016/j.jcp.2010.10.024

Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. 2001. Strong Stability-Preserving High-Order Time Discretization Methods. *SIAM Rev.* 43, 1 (Jan. 2001), 89–112. doi:10.1137/S003614450036757X

Harold Grad. 1949. On the Kinetic Theory of Rarefied Gases. *Communications on Pure and Applied Mathematics* 2, 4 (Dec. 1949), 331–407. doi:10.1002/cpa.3160020403

Zhaoli Guo, Hongwei Liu, Li-Shi Luo, and Kun Xu. 2008. A Comparative Study of the LBE and GKS Methods for 2D near Incompressible Laminar Flows. *J. Comput. Phys.* 227, 10 (May 2008), 4955–4976. doi:10.1016/j.jcp.2008.01.024

Zhaoli Guo, Kun Xu, and Ruijie Wang. 2013. Discrete Unified Gas Kinetic Scheme for All Knudsen Number Flows: Low-speed Isothermal Case. *Physical Review E* 88, 3 (Sept. 2013), 033305. doi:10.1103/PhysRevE.88.033305

Jan S. Hesthaven and Tim Warburton. 2008. *Nodal Discontinuous Galerkin Methods.* Texts in Applied Mathematics, Vol. 54. Springer New York, New York, NY. doi:10.1007/978-0-387-72067-8

H. T. Huynh. 2007. A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods. In *18th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Miami, Florida. doi:10.2514/6.2007-4079

H. T. Huynh. 2009. A Reconstruction Approach to High-Order Schemnes Including Discontinuous Galerkin for Diffusion. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, Orlando, Florida. doi:10.2514/6.2009-403

A. Jameson, P. E. Vincent, and P. Castonguay. 2012. On the Non-Linear Stability of Flux Reconstruction Schemes. *Journal of Scientific Computing* 50, 2 (Feb. 2012), 434–445. doi:10.1007/s10915-011-9490-6

Jiaqing Kou, Saumitra Joshi, Aurelio Hurtado-de-Mendoza, Kunal Puri, Charles Hirsch, and Esteban Ferrer. 2022. Immersed Boundary Method for High-Order Flux Reconstruction Based on Volume Penalization. *J. Comput. Phys.* 448 (Jan. 2022), 110721. doi:10.1016/j.jcp.2021.110721

Timm Krüger, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. 2017. *The Lattice Boltzmann Method: Principles and Practice*. Springer International Publishing, Cham. doi:10.1007/978-3-319-44649-3

Yuichi Kuya, Kosuke Totani, and Soshi Kawai. 2018. Kinetic Energy and Entropy Preserving Schemes for Compressible Flows by Split Convective Forms. *J. Comput. Phys.* 375 (Dec. 2018), 823–853. doi:10.1016/j.jcp.2018.08.058

Culbert B. Laney. 1998. *Computational Gasdynamics*. Cambridge University Press.

Chao Ma, Jie Wu, and Liming Yang. 2022. A Novel High-Order Solver for Simulation of Incompressible Flows Using the Flux Reconstruction Method and Lattice Boltzmann Flux Solver. *Computers & Fluids* 248 (Nov. 2022), 105673. doi:10.1016/j.compfluid.2022.105673

Gianmarco Mengaldo, Daniele De Grazia, Freddie Witherden, Antony Farrington, Peter Vincent, Spencer Sherwin, and Joaquim Peiro. 2014. A Guide to the Implementation of Boundary Conditions in Compact High-Order Methods for Compressible Aerodynamics. In *7th AIAA Theoretical Fluid Mechanics Conference*. American Institute of Aeronautics and Astronautics, Atlanta, GA. doi:10.2514/6.2014-2923

C. Shu, Y. Wang, C. J. Teo, and J. Wu. 2014. Development of Lattice Boltzmann Flux Solver for Simulation of Incompressible Flows. *Advances in Applied Mathematics and Mechanics* 6, 4 (Aug. 2014), 436–460. doi:10.4208/aamm.2014.4.s2

E. F. Toro. 2009. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* (3rd ed ed.). Springer, Dordrecht New York.

E. F. Toro, M. Spruce, and W. Speares. 1994. Restoration of the Contact Surface in the HLL-Riemann Solver. *Shock Waves* 4, 1 (July 1994), 25–34. doi:10.1007/BF01414629

P. E. Vincent, P. Castonguay, and A. Jameson. 2011. A New Class of High-Order Energy Stable Flux Reconstruction Schemes. *J. Sci. Comput.* 47, 1 (April 2011), 50–72. doi:10.1007/s10915-010-9420-z

Kun Xu. 2001. A Gas-Kinetic BGK Scheme for the Navier–Stokes Equations and Its Connection with Artificial Dissipation and Godunov Method. *J. Comput. Phys.* 171, 1 (July 2001), 289–335. doi:10.1006/jcph.2001.6790

Kun Xu. 2004. Discontinuous Galerkin BGK Method for Viscous Flow Equations: One-Dimensional Systems. *SIAM Journal on Scientific Computing* 25, 6 (Jan. 2004), 1941–1963. doi:10.1137/S1064827502416113