A super-parallel mixed explicit discontinuous Galerkin method for the second-order Boltzmann-based constitutive models of rarefied and microscale gases

L. Prince Raj, S. Singh, A. Karchani, and R. S. Myong*

School of Mechanical and Aerospace Engineering and ACTRC & ReCAPT, Gyeongsang National University, Jinju, Gyeongnam 52828, South Korea

Abstract: Super-parallel performance of a mixed explicit discontinuous Galerkin method is reported for the second-order Boltzmann-based nonlinear coupled constitutive models of rarefied and microscale gases. One of the challenging issues in the discontinuous Galerkin (DG) method is the higher computational cost compared with the traditional finite volume method (FVM) for a given set of grids. In the present study, we focus on the computational cost of a mixed modal explicit DG method for solving the conservation laws in conjunction with the first- and second-order Boltzmann-based constitutive models, in particular, in the context of parallelization of the implicit algebraic constitutive equations of rarefied and microscale gases in continuum and transition regimes. The computational cost of the Navier-Stokes-Fourier (NSF) and nonlinear coupled constitutive relation (NCCR) solvers is investigated in the serial and parallel frameworks. It was shown that the computational cost of the NCCR solver behaves nonlinearly with respect to the number of elements, due to the dependence of the number of iterations of the NCCR solver on the flow structure and the degree of thermal non-equilibrium. Such nonlinear dependence was clearly demonstrated from numerical solutions of three representative flows; flat plate, cylinder, and wedge. Ultimately, this nonlinear behavior of computational cost associated with nonlinear performance of the DG-NCCR solver resulted in an unexpected super-parallel performance in parallel processing.

Keywords: Rarefied and microscale gases, discontinuous Galerkin (DG), super-parallel performance, nonlinear coupled constitutive relations

* Corresponding author: Tel: +82-55-772-1645.
Email: myong@gnu.ac.kr.

1. Introduction

The study of gases in thermal non-equilibrium states has remained a fundamental and challenging topic in various scientific disciplines. The gas flows that occur at high altitude, in a rarefied state, or at a micro-scale and nano-scale are of great importance due to their increasing technological and scientific applications [1-6]. These applications include, for instance, high-speed rarefied aerodynamic flow, internal rarefied flow in the production process for organic light-emitting diodes (OLED), and micro- and nano-scale gases in MEMS devices and shale gas reservoirs. For those applications, the primary criteria used to measure the degree of thermal non-equilibrium are the Knudsen and Mach numbers. This classification stems from the fact that deviation from the local thermal equilibrium (LTE) in a macroscopic thermodynamic space is best characterized by a dimensionless composite number, $\mathrm{Kn}\cdot M$, as opposed to Kn alone, which is incompletely described in a macroscopic phase space by a previous theory, based on comparing the kinematic and collision terms in the Boltzmann kinetic equation. This composite number is defined as the ratio of the viscous stress (i.e., the direct consequence of the non-equilibrium effect) to the hydrodynamic pressure [7].

Previous theoretical and computational studies [1, 6, 8-11] have revealed that the fundamental governing laws of gases in states far from thermal equilibrium are significantly different than the conventional linear constitutive laws that are valid in flows near equilibrium. For instance, it was shown that the classical Navier and Fourier constitutive laws cannot predict the peculiar flow physics of the force-driven Poiseuille gas flow [9, 10] or the wall-driven Couette gas flow [11]. This limitation includes, for example, the non-negligible normal stress and tangential heat flux, and non-uniform tangent pressure profile in the Poiseuille flow, and the velocity gradient singularity in the Couette flow. As a consequence, simple modification of the classical Navier and Fourier laws using transport coefficients, or by introducing velocity-slip and temperature-jump boundary conditions, cannot solve the current bottleneck of problems in the classical first-order (linear uncoupled) laws. Ultimately, the problem demands a completely new development of the non-classical second-order (nonlinear coupled) laws.

In an effort to develop a *second-order* continuum-based theoretical and computational model, an implicit model based on an *algebraic* nonlinear coupled constitutive relation (called NCCR hereafter) was developed by Myong [12, 13] and extended for multi-dimensional problems by Myong *et. al* [6, 9, 10, 14-17]. Starting from the original Eu's generalized hydrodynamics [7], the NCCR theory was developed from the viewpoint of the moment method applied to the kinetic Boltzmann equation and, in particular, the so-called *balanced closure* [13]. The new closure was derived from a keen observation of the fact that the number of places for closing the moment equations derived from the Boltzmann kinetic equation is two (movement and interaction), rather than one (movement only) misled by the Maxwellian molecule assumption. Therefore, the order of approximations in handling the two terms—kinematic (movement) and dissipation (interaction) terms—must be the same; for instance, second-order for both terms. Further, in spite of its conceptual simplicity, 'balancing,' the new closure theory turned out to be extremely powerful; for example, it removed the high Mach number shock structure singularity in gas dynamics, which had remained unsolved for decades [13].

As a result of the new closure, a second-order implicit NCCR form of the stress tensor $\boldsymbol{\Pi}$ and the heat flux vector $\mathbf{Q}$ was derived [7, 9, 10, 12, 13]:

$$\begin{aligned}
&\rho\frac{D(\boldsymbol{\Pi}/\rho)}{Dt}+2\left[\boldsymbol{\Pi}\cdot\nabla\mathbf{u}\right]^{(2)}+2p\left[\nabla\mathbf{u}\right]^{(2)}=-\frac{p}{\mu}\boldsymbol{\Pi}q_{2nd}(\kappa),\\
&\rho\frac{D(\mathbf{Q}/\rho)}{Dt}+\frac{D\mathbf{u}}{Dt}\cdot\boldsymbol{\Pi}+\mathbf{Q}\cdot\nabla\mathbf{u}+\boldsymbol{\Pi}\cdot C_p\nabla T+pC_p\nabla T=-\frac{pC_p}{k}\mathbf{Q}q_{2nd}(\kappa).
\end{aligned}\tag{1.1}$$

Here the second-order dissipation term, q_{2nd}, and the first cumulant expansion term, κ, are given in terms of the hyperbolic sine form and a Rayleigh dissipation function, respectively;

$$q_{2nd}(\kappa)=\frac{\sinh\kappa}{\kappa},\quad \kappa=\frac{(mk_B)^{1/4}}{\sqrt{2}d}\frac{T^{1/4}}{p}\left(\frac{\boldsymbol{\Pi}:\boldsymbol{\Pi}}{2\mu}+\frac{\mathbf{Q}\cdot\mathbf{Q}}{kT}\right)^{1/2}.\tag{1.2}$$

In this expression, $\rho, p, T, \mathbf{u}$ and D/Dt denote the density, the pressure, the temperature, the average velocity vector, and the material time derivative, respectively. The μ, k represent the linear viscosity and

the linear thermal conductivity, respectively. The d, m, k_B denote the diameter of the molecule, the molecular mass, and the Boltzmann constant, respectively. The symbol $[\ ]^{(2)}$ stands for a traceless symmetric part of the tensor and C_p represents the heat capacity per unit mass at constant pressure. It is apparent that, when only the last two terms are retained, the second-order NCCR model (1.1) recovers the classical first-order Navier and Fourier laws near equilibrium [18, 19];

$$\mathbf{\Pi}_0 = -2\mu\left[\nabla\mathbf{u}\right]^{(2)},\ \mathbf{Q}_0 = -k\nabla T. \tag{1.3}$$

The origin of the sinh term in the second-order constitutive equation (1.1), the gist of the NCCR model, can be traced to the Boltzmann collision integral, in which the net change in the number of gas molecules is basically described by gain minus loss, so that the leading term next to the first-order becomes "sinh [11]." Note also that, in the steady-state, (1.1) is a mathematically implicit type due to the sinh form of q_{2nd}. Implicitness in this case is spawned from the fact that the dependent quantity (the viscous stress and heat flux) appears in more than two places, and one resides in the exponent of the exponential form of sinh.

Because the second-order constitutive equations (1.1) are of the implicit nonlinear type in form of sinh function, to solve them requires the introduction of the mixed type of DG method using auxiliary variables and an additional iterative algorithm. As can be seen from the following model system, which retains the essence of the original system (1.1), that is, the so-called 1-D NCCR-Burgers equation first introduced in Le *et al*. [6],

$$\begin{aligned} &\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2 + \Pi\right) = 0, \\ &\Pi = f^{-1}(-S) \ \text{ where } \ f(\Pi) = \frac{\sinh\Pi}{\Pi+1},\ S = \mu\frac{\partial u}{\partial x}, \end{aligned} \tag{1.4}$$

the NCCR model is highly nonlinear in states far from thermal equilibrium. Therefore, it is essential to employ an iterative numerical method to solve these constitutive relations together with the conservation laws, generating considerable additional computational cost.

Recently, many numerical methods have been proposed for the hyperbolic conservation laws [20, 21]. The DG method combines the main features commonly associated with the finite element method (FEM) and finite volume method (FVM) [22-24]. Similar to classical FEM, accuracy is obtained by approximating the solution based on a series of the higher-order polynomials within each element of the domain, rather than employing wide stencils, as in the case of FDM or FVM. However, in order to determine the physical solution among multiple solutions at the interfaces of the elements, the physics of wave propagation is considered, similar to finite volume method. The DG method is also conservative, stable and robust with strong mathematical supports. Further, owing to its compactness, it is suitable for unstructured grid, parallelization, and *hp*-adaptivity.

Nonetheless, there are certain challenging issues in the DG method that need to be addressed. The first is how to efficiently treat the diffusion terms required for the Navier-Stokes-Fourier equations or similar equations? Second, how can spurious oscillations be controlled in the presence of strong discontinuities or in steep regions? And, lastly, how can the high computational cost associated with large degree of freedom per element be reduced? This will be the main topic of the present study.

Richter [25] initially proposed the extension of the original DG method to the linear convection-diffusion equations. Dawson [26] developed an upwind-mixed method for the convection-diffusion equations. They used the upwind Godunov scheme for the advection terms, and the finite element approach for the second order derivatives. Lomtev *et al*. [27] solved the Navier-Stokes-Fourier equation by employing a DG spectral method and a mixed method for the convective and viscous terms, respectively. However, the DG method was not actively studied for solving the convection-diffusion equations, until Bassi and Rebay [28] presented a novel approach for solving such problems. They discretized not only the conservative variables but also their derivatives in the DG local space, and treated

the derivatives as additional dependent variables. Consequently, extra sets of equations were required to solve the linear constitutive equations in conjunction with the conservation laws. This so-called mixed DG method was shown to be easily implementable and preserved the level of accuracy in the calculation of derivatives.

In the DG method, an arbitrarily high-order approximate solution can be achieved by increasing the degree of polynomial expansion in the least square finite element space. Due to this locality feature of DG method and compactness of the stencils, parallelization of DG method is straightforward. In fact, many studies have been directed to implementations of parallel techniques in DG method for hyperbolic conservation law in the last decade [24, 29-32]. Biswas *et al*. [29] applied a third-order quadrature-based DG method to solve a scalar wave equation. Bey *et al*. [30] implemented an effective parallel algorithm based on *hp*-adaptive discontinuous Galerkin approximation for linear, scalar, hyperbolic conservation laws on structured grids. They obtained nearly optimal speed-up when the ratio of interior elements to subdomain interface elements is sufficiently large. Baggag *et al*. [31] also reported a parallel implementation of the discontinuous Galerkin method for time-dependent simulations based on unstructured grids. Luo [32] implemented a reconstruction-based parallel discontinuous Galerkin method for the solution of the compressible Navier-Stokes-Fourier equations on arbitrary grids using domain partitioning and single program multiple data (SPMD) parallel programming model. Recently, Landmann *et al*. [24] implemented an efficient parallel algorithm on high-order discontinuous Galerkin code for laminar and turbulent flows. However, very few studies were devoted to the parallelization of a mixed discontinuous Galerkin method for the conservation laws in conjunction with the second-order Boltzmann-based constitutive models of rarefied and microscale gases [33].

One of the understudied issues in the DG method is the higher computational cost in comparison with traditional FVM methods for a given set of grids. An obvious source of additional cost is the increase in degree of freedom associated with the calculation of inviscid and viscous fluxes at every Gaussian quadrature point located inside of the elements. The mixed DG method is another source, since

the mixed feature requires extra sets of equations for every degree of freedom of the primary and auxiliary variables, resulting in a system of equations with more unknowns for an equal number of elements, compared to FEM or FVM. Further, additional cost is incurred when solving the implicit algebraic constitutive equations like NCCR, where the computational cost is a nonlinear function of the number of elements. Hence, the main obstacle in adopting the mixed DG method for practical applications like rarefied and microscale gases is the lack of efficient numerical solvers to handle higher CPUs and memory requirements.

In the present study, we focus on the computational cost of a mixed modal explicit DG method for solving the NCCR and NSF equations, in particular, in the context of the parallelization of the second-order implicit algebraic constitutive equations. First, the parallelization of the DG method for the NCCR model is explained in detail. As the present work is focused on the analysis of the performance of the paralleled NCCR solver in rarefied and microscale gas conditions, rather than the study of high-order numerical schemes, the second-order (piecewise linear) DG method is employed in all simulations. The speed-up and efficiency of the DG method for piecewise constant and piecewise linear polynomial approximate solutions are then reported. After investigating the parallel performance of the mixed DG method and the flow behavior in rarefied and microscale conditions, the computational results are compared with the linear constitutive relations at various Knudsen numbers. In particular, the super-convergence of the NCCR model, which is a surprising result of the parallel DG computations, is explained in detail. Further, the origin of super-parallel performance of the NCCR solver is elucidated by simulating three representative flows; flat plate, cylinder, and wedge. Finally, the computational cost of the NCCR model is compared with that of the NSF model for both serial and parallel solvers.

2. Discontinuous Galerkin formulation

2.1. Conservation laws and constitutive relations

The conservation laws of two dimensional unsteady compressible viscous fluids in dimensionless form are given as;

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_{\text{inv}}\left(\mathbf{U}\right) + \nabla \cdot \mathbf{F}_{\text{vis}}\left(\mathbf{U}, \nabla \mathbf{U}\right) = 0, \tag{2.1}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \qquad \mathbf{F}_{\text{inv-}x} = \begin{bmatrix} \rho u \\ \rho u^2 + \dfrac{1}{N_\delta \,\text{Re}}\, p \\ \rho uv \\ \left(\rho E + \dfrac{1}{N_\delta \,\text{Re}}\, p\right) u \end{bmatrix}, \qquad \mathbf{F}_{\text{inv-}y} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + \dfrac{1}{N_\delta \,\text{Re}}\, p \\ \left(\rho E + \dfrac{1}{N_\delta \,\text{Re}}\, p\right) v \end{bmatrix},$$

$$\mathbf{F}_{\text{vis-}x} = \frac{1}{\text{Re}} \begin{bmatrix} 0 \\ \Pi_{xx} \\ \Pi_{xy} \\ \Pi_{xx} u + \Pi_{xy} v + \dfrac{1}{Ec\,\text{Pr}} Q_x \end{bmatrix}, \qquad \mathbf{F}_{\text{vis-}y} = \frac{1}{\text{Re}} \begin{bmatrix} 0 \\ \Pi_{xy} \\ \Pi_{yy} \\ \Pi_{xy} u + \Pi_{yy} v + \dfrac{1}{Ec\,\text{Pr}} Q_y \end{bmatrix}. \tag{2.2}$$

Here u and v are components of velocity in x and y directions, respectively. E stands for the total energy density, $\mathbf{\Pi}$ is the stress tensor, and $\boldsymbol{Q}$ is the heat flux vector. The dimensionless parameters such as Mach number (M), Reynolds number (Re), composite number (N_δ), Eckert number (Ec) and Prandtl number (Pr) are defined as

$$M = \frac{u_r}{\sqrt{\gamma R T_r}},\ \text{Re} = \frac{\rho_r u_r L}{\mu_r},\ N_\delta = \frac{\gamma M^2}{\text{Re}},\ Ec = (\gamma - 1) M^2,\ \text{Pr} = \frac{c_{pr} \mu_r}{k_r}, \tag{2.3}$$

where the subscript r represents the reference state, γ is the specific heat ratio of gas, R is the gas constant, and c_{pr} denotes the reference heat capacity per mass at constant pressure. The static pressure p is determined by the ideal gas law as

$$p = \gamma M^2 (\gamma - 1)\left(\rho E - \frac{1}{2}\rho\left(u^2 + v^2\right)\right). \tag{2.4}$$

2.1.1. Linear uncoupled constitutive relations

For classical linear constitutive relations (i.e., Navier-Stokes-Fourier), the components of shear stress are defined as

$$\Pi_{xx} = -\mu\left[\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}\right], \qquad \Pi_{xy} = -\mu\left[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right], \qquad \Pi_{yy} = -\mu\left[\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}\right]. \tag{2.5}$$

The heat fluxes are defined as

$$Q_x = -k\frac{\partial T}{\partial x}, \quad Q_y = -k\frac{\partial T}{\partial y}. \tag{2.6}$$

Here μ and k are the first coefficient of Chapman-Enskog viscosity and the thermal conductivity, respectively. In dimensionless form, they read as

$$\mu = T^s,\ k = T^s. \tag{2.7}$$

Here *s* stands for the index of inverse power laws of gas molecules, which is given by

$$s = \frac{1}{2} + \frac{2}{(\nu - 1)}, \tag{2.8}$$

where the parameter ν is the exponent of the inverse power law of inter-particle potential.

2.1.2. Nonlinear coupled constitutive relations

In contrast to the linear Navier-Stokes-Fourier constitutive model, the relationships in the NCCR constitutive model are a nonlinear and coupled function of velocity gradients and temperature gradients. These relationships can be developed from the second-order implicit NCCR form (1.1). They are expressed in terms of nonlinear coupled algebraic relations [12, 14] as given by

$$\begin{aligned} \hat{\mathbf{\Pi}} q_{2\mathrm{nd}}(c\hat{R}) &= \hat{\mathbf{\Pi}}_0 + \left[\hat{\mathbf{\Pi}} \cdot \nabla \hat{\mathbf{u}}\right]^{(2)}, \\ \hat{\boldsymbol{Q}} q_{2\mathrm{nd}}(c\hat{R}) &= \hat{\boldsymbol{Q}}_0 + \hat{\mathbf{\Pi}} \cdot \hat{\boldsymbol{Q}}_0. \end{aligned} \tag{2.9}$$

Here the caret (^) over a quantity represents the dimensionless quantity and the second-order term $\left[\hat{\boldsymbol{\Pi}}\cdot\nabla\hat{\mathbf{u}}\right]^{(2)}$ represents the trace-less part of the coupling between the stress and velocity gradient tensors. $\hat{\boldsymbol{\Pi}}_0$ and $\hat{\boldsymbol{Q}}_0$ values are determined by the Newtonian law of viscosity and the Fourier law of heat conduction, respectively.

$$\boldsymbol{\Pi}_0 = -2\mu\left[\nabla\mathbf{u}\right]^{(2)},\ \boldsymbol{Q}_0 = -k\nabla T\,. \tag{2.10}$$

All the variables in (2.9) and (2.10) are made dimensionless by introducing proper variables and parameters

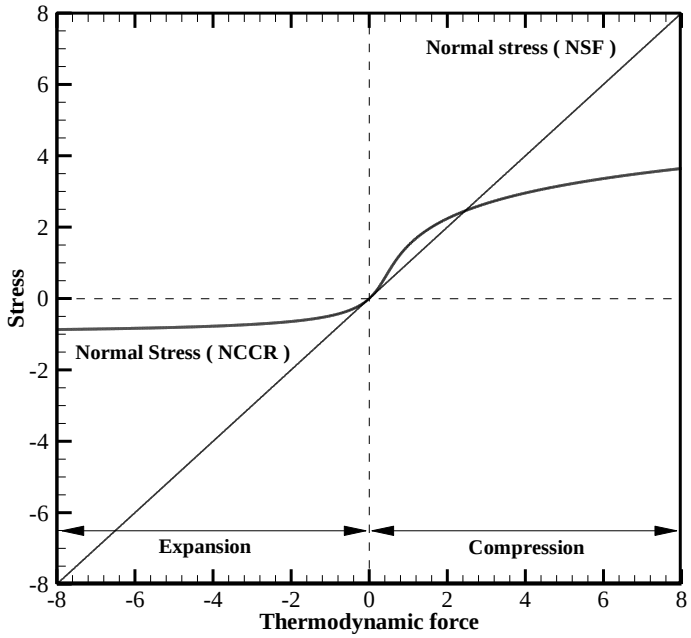
$$\hat{\boldsymbol{\Pi}} \equiv \frac{N_\delta}{p}\boldsymbol{\Pi},\ \hat{\boldsymbol{Q}} \equiv \frac{N_\delta}{p}\frac{\boldsymbol{Q}}{\sqrt{T/(2\varepsilon)}},\ \nabla\hat{\mathbf{u}} \equiv -2\eta\frac{N_\delta}{p}\nabla\mathbf{u},\ \varepsilon \equiv \frac{1}{Ec\,\mathrm{Pr}}\frac{1}{T_r/\Delta T}\ . \tag{2.11}$$

The second-order nonlinear coupling factor $q_{2\mathrm{nd}}(c\hat{R})$ and the dissipation function $\hat{R}$ are given by

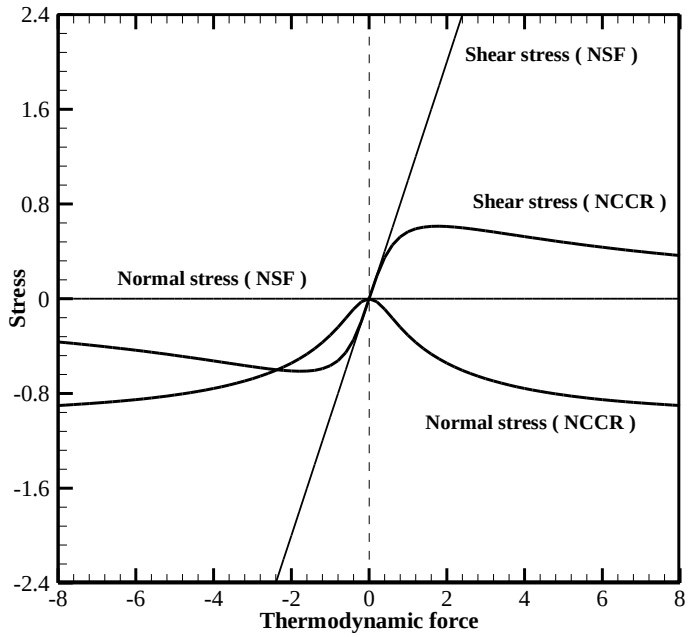
$$q_{2\mathrm{nd}}(c\hat{R}) = \frac{\sinh(c\hat{R})}{c\hat{R}},\ \ \hat{R}^2 \equiv \hat{\boldsymbol{\Pi}}:\hat{\boldsymbol{\Pi}} + \hat{\boldsymbol{Q}}\cdot\hat{\boldsymbol{Q}}. \tag{2.12}$$

Here the gas constant, $c = \left[{}^{2\sqrt{\pi}}\!/_{5}\, A_2(\nu)\Gamma\left(4-{}^{2}\!/_{\nu-1}\right)\right]^{1/2}$, has a value of 1.0179 for the argon gas molecule.

The exact solutions of (2.9) are depicted for the one-dimensional shock structure and shear flow problems in Fig. 1. The NCCR solutions are indeed well-posed (existence, uniqueness, and continuous dependence on the data) for all inputs, and remove the high Mach number shock structure singularity completely [12, 13]. The solutions also show the free-molecular asymptotic behavior with increasing degree of expansion and velocity-shear, satisfying $\hat{\Pi}_{xx} \to -1$ or $\Pi_{xx} + p \to 0$.



(a) Compression-expansion



(b) Shear flow

Figure 1. Constitutive relations in (a) compression and expansion flow, and (b) shear flow. The horizontal axis represents the thermodynamic force by velocity gradient $\hat{\Pi}_0$, while the vertical axis represents the normal stress $\hat{\Pi}$.

2.2. Iterative numerical solvers for nonlinear coupled constitutive relations

In general, the second-order constitutive equations (2.9) consist of nine nonlinear implicit algebraic equations of non-conserved variables $(\hat{\Pi}_{xx},\hat{\Pi}_{xy},\hat{\Pi}_{xz},\hat{\Pi}_{yy},\hat{\Pi}_{yz},\hat{\Pi}_{zz},\hat{Q}_x,\hat{Q}_y,\hat{Q}_z)$ for the eleven known variables $\left(p,T,\nabla\boldsymbol{u},\nabla T\right)$. Owing to the highly nonlinear terms (e.g., $q_{2\text{nd}}(c\hat{R})$ function), developing a proper numerical method for solving the nonlinear system appears to be a daunting task. Nevertheless, it was shown that it can be solved numerically by the *method of iterations* in the case of a one-dimensional problem [12, 14, 15]. Further, in the case of the two-dimensional problem, the viscous stress and heat flux components ($\hat{\Pi}_{xx},\hat{\Pi}_{xy},\hat{Q}_x$) on a line in the physical plane induced by thermodynamic forces (velocity and temperature gradients, $\partial u/\partial x,\partial v/\partial x,\partial T/\partial x)$ can be approximated as the sum of two solvers; a solver on $(\partial u/\partial x,0,\partial T/\partial x)$, and another solver on $(0,\partial v/\partial x,0)$.

The iteration procedures can be decomposed into two solvers. In the first solver of the compression and expansion of a monatomic gas, the positive $\hat{\Pi}_{xx}$ and $\hat{Q}_x$ relations are given by

$$\begin{aligned}\hat{R}_{n+1}&=\frac{1}{c}\sinh^{-1}\left[c\left(\hat{\Pi}_{xx_n}+1\right)\hat{R}_0\right],\\ \frac{\hat{Q}_{x_{n+1}}}{\hat{\Pi}_{xx_{n+1}}}&=\frac{\hat{Q}_{x_n}}{\hat{\Pi}_{xx_n}}=\frac{\hat{Q}_{x_0}}{\hat{\Pi}_{xx_0}},\end{aligned}\tag{2.13}$$

and negative $\hat{\Pi}_{xx}$ and $\hat{Q}_x$ relations are determined by

$$\begin{aligned}\hat{\Pi}_{xx_{n+1}}&=\frac{\hat{\Pi}_{xx_0}}{q_{2\text{nd}}(c\hat{R}_n)-\hat{\Pi}_{xx_0}},\\ \hat{Q}_{x_{n+1}}&=\frac{(\hat{\Pi}_{xx_n}+1)}{q_{2\text{nd}}(c\hat{R}_n)}\hat{Q}_{x_0}.\end{aligned}\tag{2.14}$$

In the second solver of the shear flow, the $\hat{\Pi}_{xx}$ can be obtained for a given $\hat{\Pi}_{xy_0}$ through the iterative relation

$$\hat{\Pi}_{xx_{n+1}}=\frac{\hat{\Pi}_{xy_0}^2}{3q_{2\text{nd}}^2(c\hat{R}_n)/2+\hat{\Pi}_{xy_0}},\ \hat{\Pi}_{xy_{n+1}}=\text{sign}(\hat{\Pi}_{xy_0})\left[-\frac{3}{2}(\hat{\Pi}_{xx_{n+1}}+1)\hat{\Pi}_{xx_{n+1}}\right]^{1/2}.\tag{2.15}$$

Here $\hat{\Pi}_{xx_1}$ and $\hat{Q}_{x_1}$ denote the initial guesses of the solution. Solutions of the iterative method for 2D NCCR equations are considered converged when $\left|\hat{R}_{n+1}-\hat{R}_n\right| \le 10^{-5}$. Finally, the converged values at the last iteration (n+1) are then implemented back into the 2D DG solver as

$$\Pi_{xx}=\frac{p}{N_\delta}\hat{\Pi}_{xx_{n+1}},\quad \Pi_{xy}=\frac{p}{N_\delta}\hat{\Pi}_{xy_{n+1}},\quad Q_x=\frac{p\sqrt{T/2\varepsilon}}{N_\delta}\hat{Q}_{x_{n+1}}. \tag{2.16}$$

2.3. Mixed DG formulation

The spatial discretization of both NSF and NCCR equations is based on the mixed DG formulation proposed by Bassi and Rebay [28]. This formulation determines the value of the second-order derivatives present in viscous terms by adding auxiliary unknowns **S**, because the second-order derivatives cannot be accommodated directly in a weak formulation using a discontinuous function space. Therefore, **S** can be defined as the derivative of either primitive or conservative variables **U**. This leads to a coupled system for **S** and **U** as

$$\begin{cases}\dfrac{\partial \mathbf{U}}{\partial t}+\nabla\cdot\mathbf{F}_{\text{inv}}\left(\mathbf{U}\right)+\nabla\cdot\mathbf{F}_{\text{vis}}\left(\mathbf{U},\mathbf{S}\right)=0,\\ \mathbf{S}-\nabla\mathbf{U}=0.\end{cases} \tag{2.17}$$

The dimensionless form of the spatial derivatives of primitive variables can then be computed by expanding the derivatives of the conservative variables as

$$\begin{aligned}
&\rho_x=\frac{\partial\rho}{\partial x},\ \rho_y=\frac{\partial\rho}{\partial y},\\
&u_x=\frac{1}{\rho}\left[\frac{\partial\rho u}{\partial x}-u\frac{\partial\rho}{\partial x}\right],\ u_y=\frac{1}{\rho}\left[\frac{\partial\rho u}{\partial y}-u\frac{\partial\rho}{\partial y}\right],\\
&v_x=\frac{1}{\rho}\left[\frac{\partial\rho v}{\partial x}-v\frac{\partial\rho}{\partial x}\right],\ v_y=\frac{1}{\rho}\left[\frac{\partial\rho v}{\partial y}-v\frac{\partial\rho}{\partial y}\right],\\
&p_x=\gamma\left(\gamma-1\right)M^2\left[\frac{\partial\rho E}{\partial x}-\left(u\frac{\partial\rho u}{\partial x}+v\frac{\partial\rho v}{\partial x}\right)+\frac{1}{2}\left(u^2+v^2\right)\frac{\partial\rho}{\partial x}\right],
\end{aligned} \tag{2.18}$$

$$p_y=\gamma(\gamma-1)M^2\left[\frac{\partial\rho E}{\partial y}-\left(u\frac{\partial\rho u}{\partial y}+v\frac{\partial\rho v}{\partial y}\right)+\frac{1}{2}\left(u^2+v^2\right)\frac{\partial\rho}{\partial y}\right],$$

$$T_x=\frac{\gamma(\gamma-1)M^2}{\rho}\left[\frac{\partial\rho E}{\partial x}-\left(u\frac{\partial\rho u}{\partial x}+v\frac{\partial\rho v}{\partial x}\right)+\frac{1}{2}\left(u^2+v^2\right)\frac{\partial\rho}{\partial x}-\frac{1}{\rho}\left(\rho E-\frac{1}{2}\rho\left(u^2+v^2\right)\right)\frac{\partial\rho}{\partial x}\right],$$

$$T_y=\frac{\gamma(\gamma-1)M^2}{\rho}\left[\frac{\partial\rho E}{\partial y}-\left(u\frac{\partial\rho u}{\partial y}+v\frac{\partial\rho v}{\partial y}\right)+\frac{1}{2}\left(u^2+v^2\right)\frac{\partial\rho}{\partial y}-\frac{1}{\rho}\left(\rho E-\frac{1}{2}\rho\left(u^2+v^2\right)\right)\frac{\partial\rho}{\partial y}\right].$$

It was noted by Le *et al*. [6] that the introduction of an extra set of equations for the auxiliary variables in (2.17) is necessary for the nonlinear implicit type of the constitutive models, such as NCCR, because it is not possible to directly combine auxiliary equations with primary equations due to the implicitness form of the viscous Jacobian matrix.

In order to discretize the mixed system (2.17) within the triangulated elements, the exact solutions of **U** and **S** are approximated by the DG polynomial approximations of $\mathbf{U}_h$ and $\mathbf{S}_h$, respectively,

$$\mathbf{U}_h(\mathbf{x},t)=\sum_{i=0}^{N_k}\hat{u}_h^i(t)\varphi^i(\mathbf{x}),\ \mathbf{S}_h(\mathbf{x},t)=\sum_{i=0}^{N_k}\hat{S}_h^i(t)\varphi^i(\mathbf{x}),\ \ \forall\mathbf{x}\in I, \tag{2.19}$$

where $\hat{u}_h^i(t)$ and $\hat{S}_h^i$ are the local degrees of freedom of **U** and **S**. $\varphi(\mathbf{x})$ is the basis function for finite element space, while N_k is the number of required basis function for the *k*-exact DG approximation. In the present study, modal basis functions proposed by Dubiner [34] are employed for triangular elements. Further, the mixed system (2.17) is multiplied with the test function, which is taken to be equal to the basis function $\varphi(\mathbf{x})$, and then integrated by parts over an element *I*. This results in the following weak formulation of the mixed system for $\mathbf{U}_h$ and $\mathbf{S}_h$

$$\begin{cases}\dfrac{\partial}{\partial t}\displaystyle\int_I\mathbf{U}_h\varphi dV-\int_I\nabla\varphi\cdot\mathbf{F}_{\text{inv}}dV+\int_{\partial I}\varphi\mathbf{F}_{\text{inv}}\cdot\mathbf{n}d\Gamma-\int_I\nabla\varphi\cdot\mathbf{F}_{\text{vis}}dV+\int_{\partial I}\varphi\mathbf{F}_{\text{vis}}\cdot\mathbf{n}d\Gamma=0,\\ \displaystyle\int_I\mathbf{S}_h\varphi dV+\int_I\nabla\varphi\mathbf{U}_h dV-\int_{\partial I}\varphi\mathbf{U}_h\mathbf{n}d\Gamma=0,\end{cases} \tag{2.20}$$

where **n** is the outward unit normal vector. *V* and Γ represent the volume and boundary of the element *I*, respectively. The number of quadrature points necessary for *k*-th order finite element space depends on

the type of quadrature rules employed in the numerical process [35]. In the present study, the Gauss-Legendre quadrature rule has been implemented for both volume and boundary integrations. Therefore, the volume and boundary integrals in (2.20) are computed using $2k$ and $2k+1$ order accurate Gauss quadrature formulas, respectively [35, 36].

The flux functions $\mathbf{F}_{\text{inv}} \cdot \mathbf{n}, \mathbf{F}_{\text{vis}} \cdot \mathbf{n}$ and $\mathbf{U} \cdot \mathbf{n}$ appearing in (2.20) are represented by a numerical flux function. The dimensionless form of the Rusanov (local Lax–Friedrichs (LLF)) flux $\mathbf{h}_{\text{inv}}$ is applied for inviscid terms. This monotone flux is commonly used in the DG method due to its efficiency in computational cost. The Rusanov (LLF) flux is also the most dissipative flux that may improve the stability of DG numerical approximation [37].

$$\begin{aligned} &\mathbf{F}_{\text{inv}} \cdot \mathbf{n} \approx \mathbf{h}_{\text{inv}}\left(\mathbf{U}^{-}, \mathbf{U}^{+}\right) = \frac{1}{2}\left[\mathbf{F}_{\text{inv}}\left(\mathbf{U}^{-}\right) + \mathbf{F}_{\text{inv}}\left(\mathbf{U}^{+}\right) - C\left(\mathbf{U}^{+} - \mathbf{U}^{-}\right)\right], \\ &C = \max\left(\left|\mathbf{u}^{-}\right| + a_S^{-}, \left|\mathbf{u}^{+}\right| + a_S^{+}\right). \end{aligned} \tag{2.21}$$

Here $a_S = \sqrt{T}/M$ is the speed of sound at an elemental interface, and the superscripts (+) and (-) denote the inside and outside sides at an elemental interface. The central flux (BR1) [28] is employed as the numerical fluxes for calculation of auxiliary and viscous fluxes at elemental interfaces;

$$\begin{aligned} &\mathbf{F}_{\text{vis}} \cdot \mathbf{n} \approx \mathbf{h}_{\text{vis}}\left(\mathbf{U}^{-}, \mathrm{S}^{-}, \mathbf{U}^{+}, \mathrm{S}^{+}\right) = \frac{1}{2}\left[\mathbf{F}_{\text{vis}}\left(\mathbf{U}^{-}, \mathrm{S}^{-}\right) + \mathbf{F}_{\text{vis}}\left(\mathbf{U}^{+}, \mathrm{S}^{+}\right)\right], \\ &\mathbf{U} \cdot \mathbf{n} \approx \mathbf{h}_{\text{aux}}\left(\mathbf{U}^{-}, \mathbf{U}^{+}, \mathbf{n}\right) = \frac{1}{2}\left[\mathbf{U}^{-} + \mathbf{U}^{+}\right]. \end{aligned} \tag{2.22}$$

In high-order numerical schemes, spurious numerical fluctuations of the solutions may result in negative density and pressure during the time evaluation, violating the physical constraint of positivity. Therefore, a positivity preserving limiter is needed to enforce positive pressure and density at every element. In this work, a positivity preserving limiter proposed by Shu and Zhang is employed [38, 39]. By assembling all the elemental contributions together, the semi-discrete DG formulation for 2D conservation laws (2.1) yields a system of ODE's in time for each element as

$$\mathbf{M}\frac{d\mathbf{U}}{dt} = \mathbf{R}(\mathbf{U}), \tag{2.23}$$

where **M** is the diagonal mass matrix and **R**(**U**) is the residual vector of the system. A third-order total variation diminishing Runge-Kutta (TVD-RK) method is employed for explicit time marching [40]. The local time step Δt for each element is determined by the following relation [41]

$$\Delta t = \frac{h}{(2k+1)} \frac{CFL}{|u| + a_s + \frac{1}{\mathrm{Re}}\frac{\mu}{h}}, \tag{2.24}$$

where *CFL* is the Courant number and h is the radius of the circumscribed circle in element I.

3. Parallelization of explicit DG solver

The DG method is compact and highly parallelizable due to the local nature of the discretization. The solution is approximated independently in each element, where inter-element data sharing is only needed among the face neighbor elements (elements sharing a common face) to calculate numerical fluxes. Therefore, inter-process communication is only required between the corresponding neighboring processes for the computations at partition boundary faces (i.e., faces having their *left* and *right* elements with different processes).

In this study, a single program multiple data (SPMD) parallel model using a message-passing-interface (MPI) library was employed to parallelize the present mixed modal explicit DG method. The MPI library guarantees the maximal flexibility of the parallel programming, portability and scalability of the distributed memory parallel architectures [42]. A shell program was developed to unify all the parallel processing steps, as illustrated in Fig. 2.

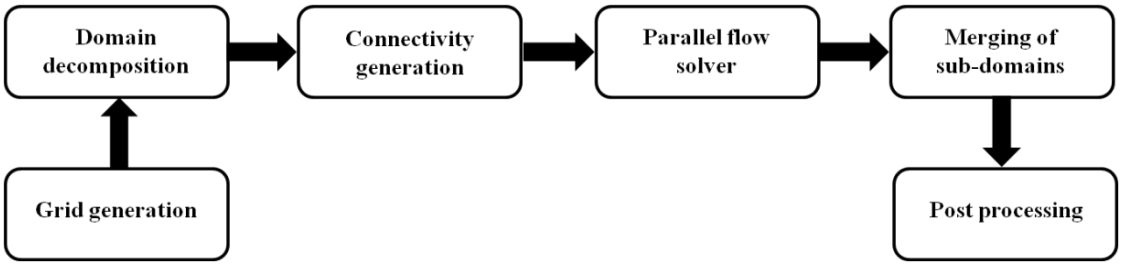


Figure 2. Illustration of unified shell program for parallel processes.

The software setup including MPICH (i.e., a high performance and widely portable implementation of the MPI standard) and 64-bit compilers with double precision accuracy was used for all the floating point operations. Moreover, a Linux cluster sharable among multiple users was established using Intel Xenon processors with ten cores at each node. This cluster is equipped with eighty cores interconnected by dual port Gigabit Ethernet.

The steps in the parallelization of the DG solver for rarefied gas flows including domain decomposition, communication process, merging of sub-domains and parallel performance measurements are described in the following sub-sections.

3.1. Domain decomposition (mesh partitioning)

Mesh partitioning is the first step in the parallel programming, where the computational domain is decomposed into several sub-domains and then individual sub-domains are assigned to each processor. Decomposition of the domain into several sub-domains was done using open source software, ParMETIS [43]. ParMETIS is an MPI-based parallel library that implements a variety of algorithms for computing fill-reducing orderings of sparse matrices, and partitioning of the unstructured graphs. It decomposes the given mesh such that each processor has approximately the same number of elements, which balances the load for the processors and the number of links cut by the decomposition is minimized. This feature is crucial to minimizing communication among the processors [43]. After the decomposition of the domain, the partitioned results, including the node and element connectivity information, are assigned to the processors. The sub-domains generated by ParMETIS for the case of flows around a cylinder and an airfoil with approximately 25,000 elements are shown in Fig. 3.

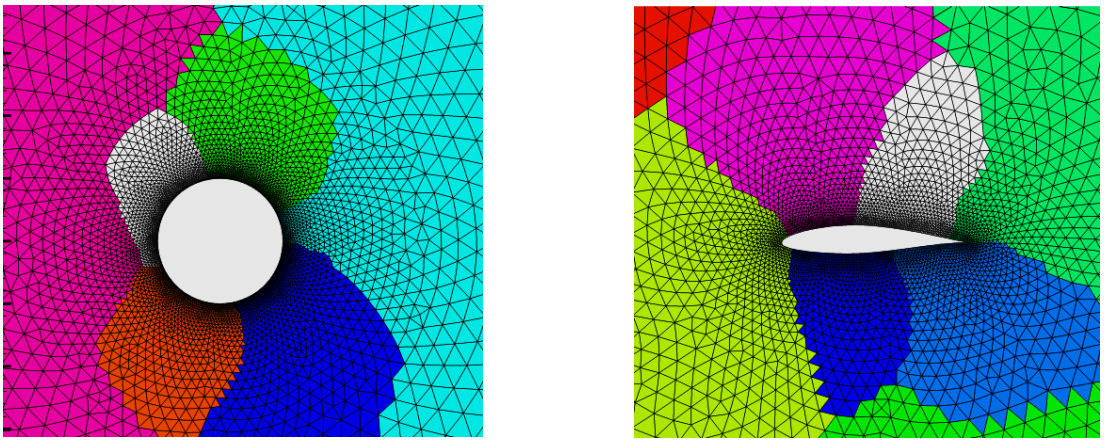


Figure 3. Triangular mesh partition for 8 processors using ParMETIS. Different colors represent sub-domains owned by different processors.

3.2. Communication process

The present parallel solver is based on a single program, multiple data (SPMD), which executes the same program in all processors with different data. The SPMD model can manage the processors to conditionally execute only certain parts of the program. Therefore, some of the processors may not necessarily need to execute the entire program. In this study, the parallelization was achieved without compromising the serial algorithm for the purpose of higher parallel performance. Moreover, the present parallel algorithm allows the MPI communications to completely overlap with the computations. This type of algorithm is usually referred to as hiding communication behind computation, which is easier to achieve in explicit time marching schemes [31] as summarized in Fig. 4. The point-to-point communication methodology of MPI was used such that the message passing operation may only occur between two different processors. While one processor is performing a *send* operation, the other processor performs a matching *receive* operation.

There are various types of *send* and *receive* routines that are available in MPI point-to-point communication. Either blocking or non-blocking routines are often used in the SPMD model due to their flexibility and for the sake of implementation. Both communication methods use a buffer to avoid data loss and confusion during the transmission of data from one processor to another. Hence, data will be copied to the buffer before it is received by the partner processor.

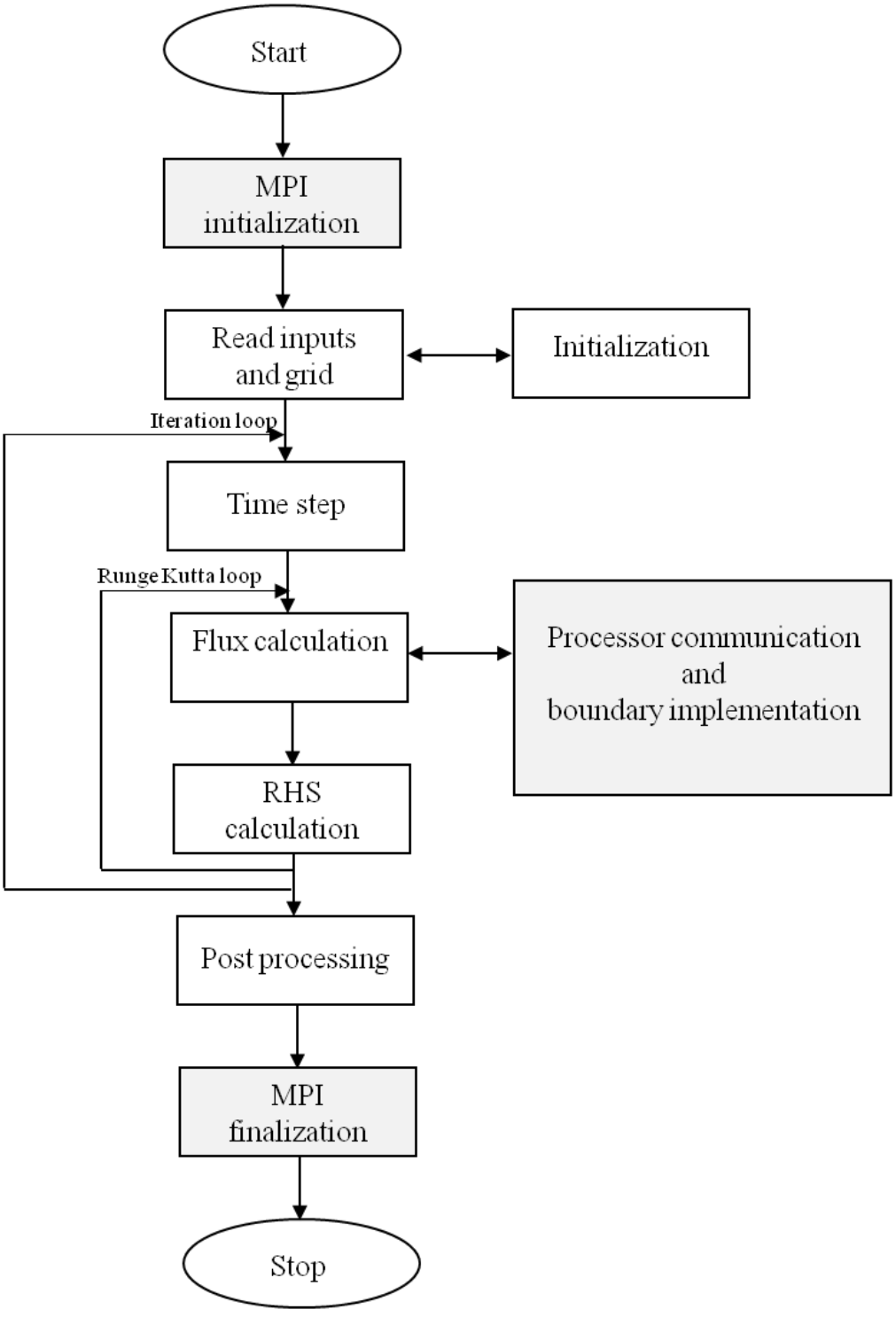


Figure 4. Flow chart of explicit DG parallel algorithm.

A buffer is a region of memory storage designed to temporarily store data during the communication process. In the blocking *send* and *receive* routines, the *send* routine will only return (block) after the completion of communication. Hence, computations cannot be done by the respective processors involved in communication until the process is completed. On the contrary, non-blocking communication functions return immediately (i.e., do not block) even if the communication is not finished. While using non-blocking communications, care should be taken to use the proper *wait* comment, to see whether the communication has finished or not. Non-blocking communications are primarily used to overlap computation with communication and exploit possible performance gains.

The communication module of the DG solver starts working by sending data (adjacent to partition boundaries) to neighbor partitions and this is followed by receiving data from a corresponding neighbor. These communications should be repeated for each of the Gaussian quadrature points on the element boundaries. However, the number of Gaussian quadrature points will increase with the increasing order of accuracy of the DG approximation, as shown in Fig. 5. As a result, the amount of data communication will also increase as the DG order of accuracy increases, as shown in Fig. 6.

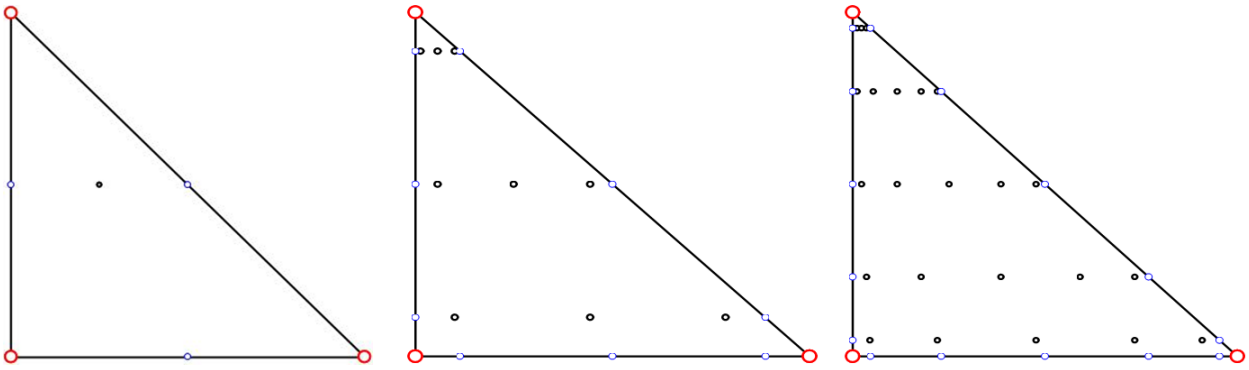


Figure 5. Number of Gaussian quadrature points required for DG-approximation with (a) k=0, (b) k=1, and (c) k=2 order of accuracy.

Non-blocking sending and receiving were used in the parallelization in order to save processor waiting time and avoid deadlock. Therefore, the application of MPI_CHECK and MPI_WAIT was

essential to confirm the completion of communication without data loss. These operations were started by calling standard MPI routines, MPI_ISEND and MPI_IRECV. Furthermore, the MPI_WAITALL routine was used to ensure the completion of the communication process. The block diagram of the DG communication algorithm is shown in Fig. 7. Once communication was completed, the data received from the neighboring processors were used for further computations.

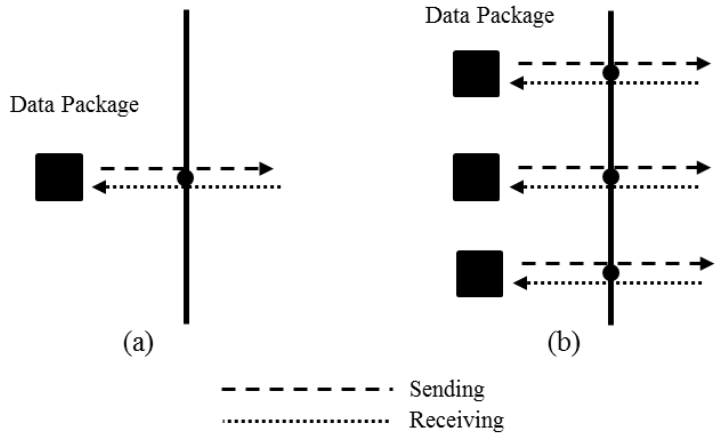


Figure 6. Data communication through Gaussian quadrature points for (a) DG piecewise constant scheme, and (b) DG piecewise linear scheme. Data package is the solution information.

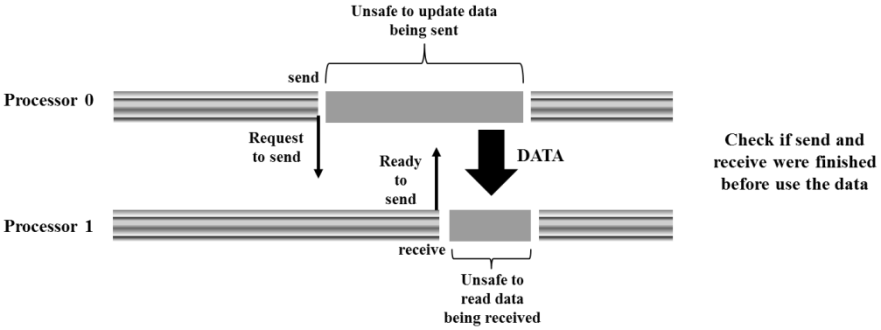


Figure 7. Non-blocking communication pattern (Isend and Irecv block diagram).

3.3. Merging of sub-domains

During the parallel computations, all the partitioned sub-domains execute the same DG solver with respective data inputs and solve the flow fields in their local domain. After the solution converges, each

of the processors plots its solution for post-processing purposes. However, it is noticeable in Fig. 8(a) that the results are visually not smooth at the boundaries of the sub-domains due to biased interpolation of the solution, and as a result of not considering all vertex neighborhoods for interpolations. A merging subroutine was devised for better post-processing of the solutions of parallel computations. In this subroutine, all subdomain results were exported into a unified single domain for the purpose of better visualization. The DG approximate solutions are sought in the finite element space and thus the solution at any point inside the computation cell can be calculated by summing up the product of moment of the solution and basis function. Although each element contains its own solution, all the solutions are needed to be interpolated to the node for post processing software like TECPLOT. Otherwise, the biased interpolation that does not consider all the neighbors of the node can result in very poor visualization as demonstrated in Fig. 8(a). In order to avoid such shortcoming, the merging of sub-domains was performed for post processing after terminating parallel processing, as shown in Fig. 8(b). In this process, as the spatial polynomial function is defined in least square space and the solution in each element is calculated locally, the results are irrespective of the number of processors.

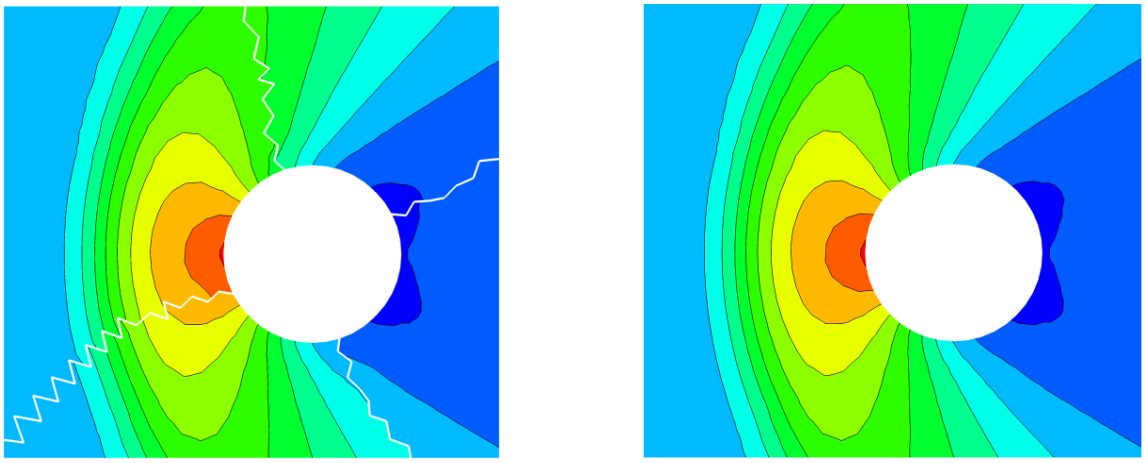


Figure 8. Merging of the sub-domains for post-processing of the solutions; (a) pressure contour of unmerged sub-domains, and (b) pressure contour of the merged domain.

3.4. Measurement of parallel performance

The measurement of parallel computation is essential for assessing the efficiency and applicability of the parallel solver. Generally, parallel performance is measured by relative speed-up, relative efficiency or scalability [44]. The definition of speed-up (S_p) was established by Amdahl's law [45]. According to this law, it is a metric for the relative improvement in performance when executing a task. However, speed-up can be used more generally to show the effect of any performance enhancement. The relative speed-up is given by

$$S_p = \frac{t_s}{t_p}, \tag{3.1}$$

where S_p is speed-up, t_s and t_p denote the elapsed time taken by a *single* processor and p processors, respectively. Relative efficiency (E) is a metric of the utilization of the resources of the improved parallelized system read as

$$E = \frac{S_p}{p}. \tag{3.2}$$

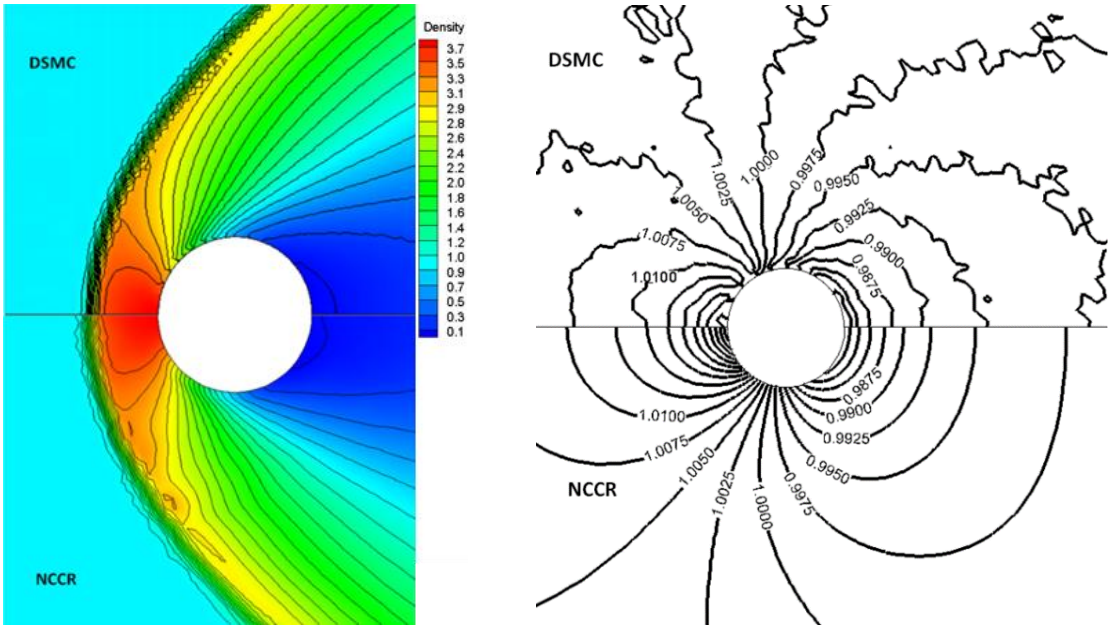
A performance analysis indicates the level of speed-up and efficiency of the parallel solver. Speed-up of the code varies with the increase in the number of processors for a fixed problem size. Linear speed-up usually remains less than p, and efficiency lies between 0 and 1. In ideal cases, elapsed time taken by p processors is equal to $t_p = t_1/p$, relative speed-up is equal to $S_p = p$, and relative efficiency is equal to E =1.

4. Results and discussions

4.1 NCCR and NSF models in rarefied and microscale conditions

The two-dimensional parallel DG code was validated for various benchmark problems of viscous compressible gas flow. A monatomic argon gas with Pr=2/3 and s=0.75 was chosen as the working gas. In the present work and our previous studies [6, 8, 46], the numerical results were compared with the DSMC results. Unstructured triangular grids with approximately 90,000 computation cells refined with a

ratio of 1.06 near the wall to capture the flow physics are used for the DG simulations. The simulation time step was calculated based on equation (2.24) with a CFL number of 0.1 and the simulations were run for approximately 100,000 iterations. The far-field boundary condition was imposed on the outer boundary of the computational domain while the Langmuir boundary condition [14, 15, 47] was applied for the solid wall. The power law model was used for calculating the transport properties [12]. On the other hand, in the DSMC simulation, the VHS inter-particle collision model and the fully diffusive wall boundary condition were implemented. Approximately 2,000,000 particles were used with at least 50 particles per each cell. The time step and the cell size were set one-tenth of mean collision time and one-fifth of the free-stream mean free path, respectively [48, 49].

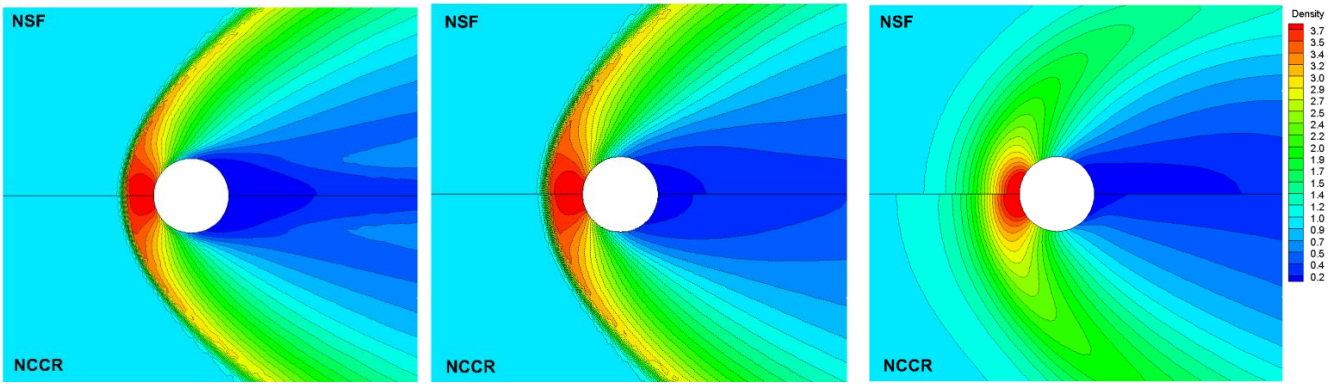


(a) (b)

Figure 9. Validation of DG-NCCR model with the DSMC in density distribution for (a) M = 5.48, Kn = 0.05, and (b) M = 0.1, Kn = 0.1.

In Fig. 9, a comparison of the DG code with DSMC solutions for rarefied and microscale gas flow around a circular cylinder is shown for both cases of high speed rarefied (M=5.48, Kn=0.05) and low speed microscale (M=0.1, Kn=0.1) problems. The results show that the numerical solutions of the NCCR model are in close agreement with the simulated solutions of DSMC. In the high speed case, the flow

consists of a compressive bow shock structure, a stagnation region near the frontal part of the cylinder, and a gaseous expansion region near the rear part of the cylinder. The density remains initially constant at the free-stream region and then experiences a rapid change across the bow shock wave, whose value is very close to the theoretical prediction given by the Rankine–Hugoniot relations. Further, the shock thickness and general flow pattern are very similar. In the low speed case, the flow consists of smooth compression and stagnation regions near the frontal part of the cylinder, and a modest expansion region near the rear part of the cylinder. The solution of NCCR is smooth in the low speed regime, whereas the DSMC solution contains non-negligible statistical fluctuations.

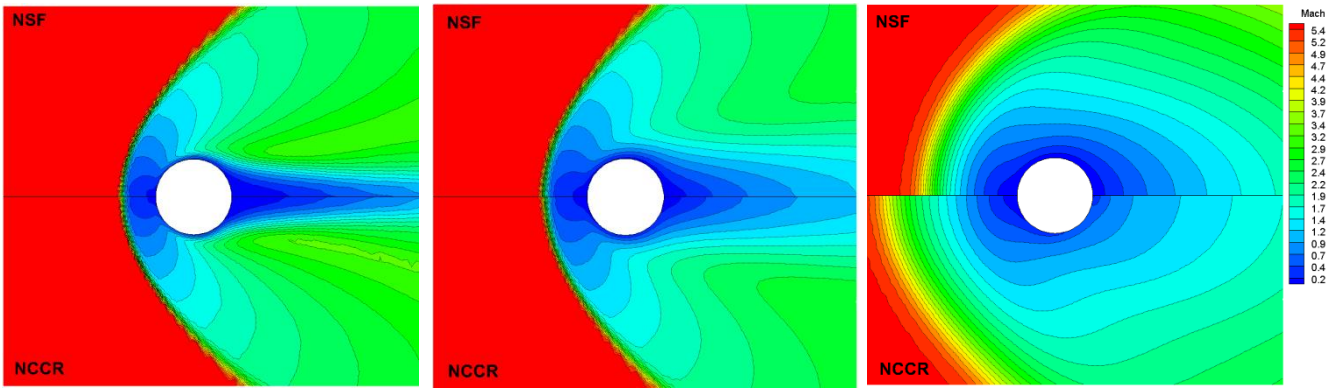


(a) (b) (c)

Figure 10. Comparison of normalized density contours for NSF and NCCR at M = 5.48 with (a) Kn = 0.005, (b) Kn = 0.05, and (c) Kn = 0.5.

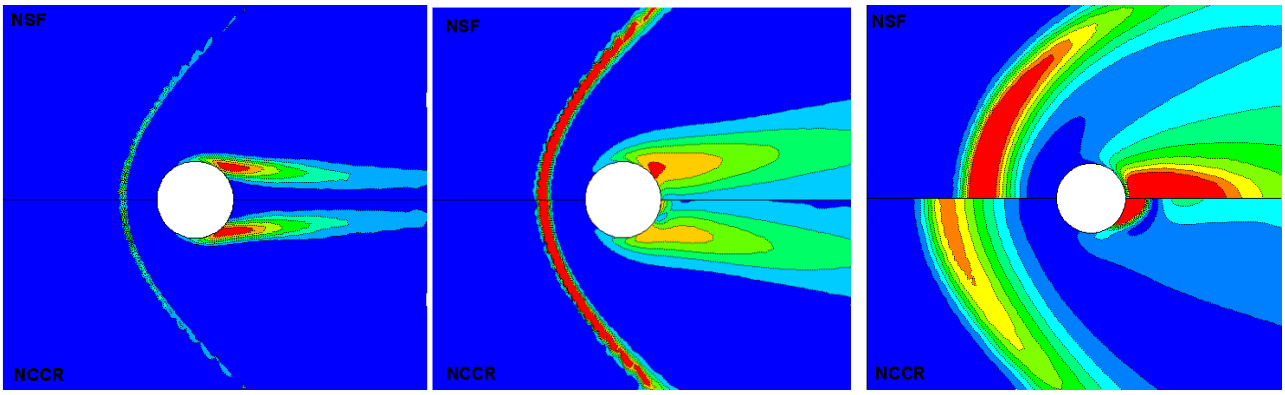
The parallel DG solver was also applied to simulate the high speed flow (M=5.48) around a circular cylinder in the continuum (Kn=0.005) and transition (Kn=0.5) regimes. The NSF results were compared with the NCCR results for three different Knudsen numbers in Figs. 10 and 11. The contours show that the flow fields vary significantly as the Knudsen number increases from 0.005 to 0.5. At Kn=0.005, the discrepancies between the NSF and NCCR results turn out to be negligible, as shown in Figs. 10(a) and 11(a). As the Knudsen number increases to 0.05, however, non-negligible deviations begin to appear, as seen in Figs. 10(b) and 11(b). As the Knudsen number increases further to 0.5, well within the transition regime, significant deviations are present all over the flow field, as shown in Figs. 10(c) and 11(c). There are two distinctive regions of compression and expansion in the frontal and rear parts of the cylinder,

respectively. A most notable difference is the structure of the stand-off shock wave in the frontal parts; a thicker and broader shock structure in the NCCR model in comparison with the NSF model.



(a) (b) (c)

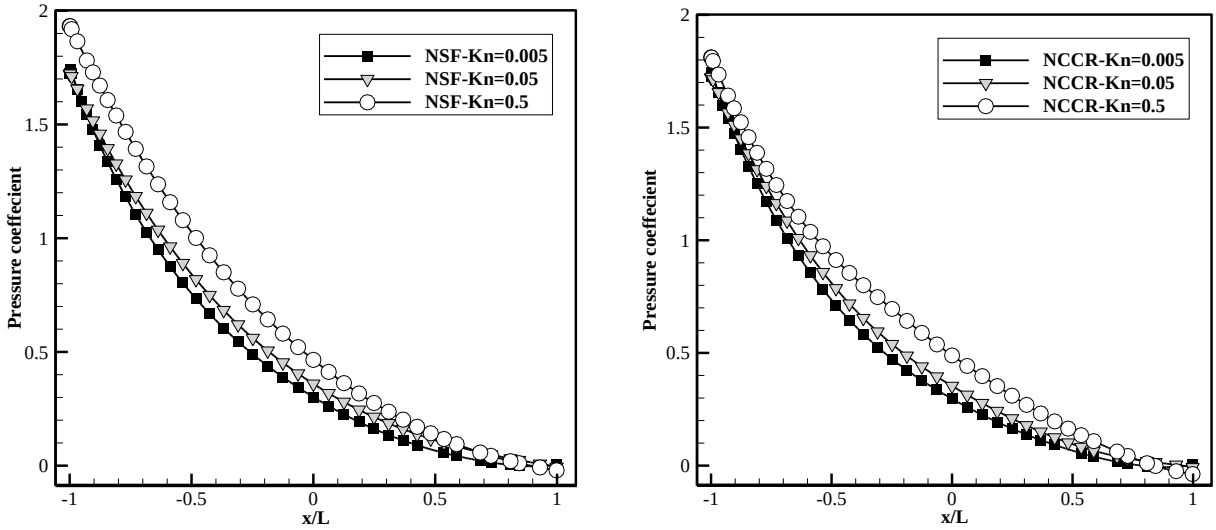
Figure 11. Comparison of Mach number contours for NSF and NCCR at M = 5.48 with (a) Kn = 0.005, (b) Kn = 0.05, and (c) Kn = 0.5.



(a) (b) (c)

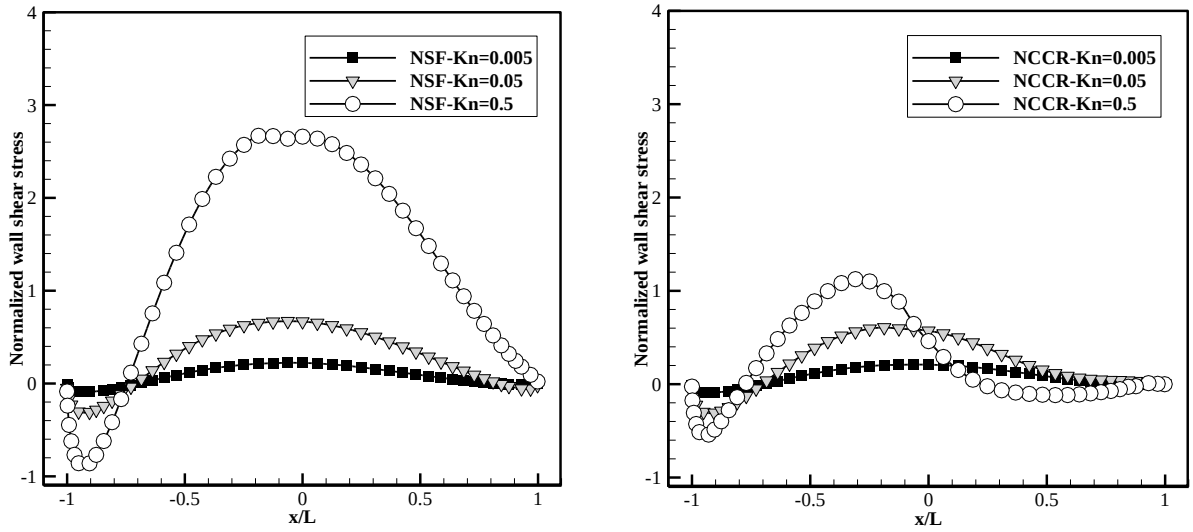
Figure 12. Comparison of degree of non-equilibrium contours for NSF and NCCR at M = 5.48 with (a) Kn = 0.005, (b) Kn = 0.05, and (c) Kn = 0.5.

Further, from the contours of the degree of thermal non-equilibrium measured by the Rayleigh–Onsager dissipation function (2.12) as described in Fig. 12, significant deviations from the LTE assumption can be observed in the compressive bow shock region and the expansive rear part of the cylinder. Because of this, most of the gaps between the NSF and NCCR models are observed in these regions. It can also be noted that the degree of gaseous expansion near the rear part of the cylinder predicted by the NSF model is considerably higher than that of the NCCR model. This over-estimation of the degree of thermal non-equilibrium in the NSF model may explain the poor performance of the NSF model in high Knudsen and Mach number flows.



(a) (b)

Figure 13. Pressure coefficient distribution around a cylinder at various Knudsen numbers for (a) NSF model, and (b) NCCR model.



(a) (b)

Figure 14. Normalized wall shear stress distribution around a cylinder at various Knudsen numbers for (a) NSF model, and (b) NCCR model.

Fig. 13 illustrates the pressure coefficient distributions around the cylinder predicted by the NSF and NCCR models at three different Knudsen numbers. The pressure coefficient values estimated by the NCCR model at Kn = 0.5 is considerably different from that of NSF, since the degree of thermal non-

equilibrium is non-negligible in one-eighth of the frontal part of the cylinder. Fig. 14 shows a comparison of the distribution of the normalized wall shear stress around the cylinder predicted by the NSF and NCCR models. A significant reduction in viscous shear stress across the surface of the cylinder is observed for the NCCR model, in particular, in the case of Kn=0.5 in the transition regime. This is due to the shear-thinning property of the second-order NCCR model in comparison with the first-order NSF model as shown in Fig. 1(b), that is, smaller effective viscosity at high Knudsen number [11, 47].

4.2 Parallel performance of the NSF solver

Figs. 15(a) and 15(b) illustrate the speed-up of the piecewise constant and piecewise linear DG approximations for elements ranging from 4,000 to 200,000, and with a range of processors from 1 to 64. The plots indicate that the speed-up increases almost linearly as the number of processors increases, and the speed-up is enhanced in the case of piecewise linear approximation. Figs. 16(a) and 16(b) show the relative efficiency of the parallel code for piecewise constant and piecewise linear DG approximations, respectively. The communication overload increases as the number of processors increases, and, as a result, the required run-time for communication between processors becomes comparable to the computational time of the simulations, for cases with a smaller number of elements. In passing, it should be mentioned that the algorithms used in the serial code are well optimized; for example, the arrays are arranged such that they follow the storage order of the compiler or programming language.

Hence, the speed-up and parallel efficiency are higher for cases with larger numbers of elements (200,000) and processors (64). Moreover, the speed-up of the piecewise linear polynomial expansion is substantially higher than that of the piecewise constant scheme due to the reduction in communication overload between processors in comparison with the numerical computation overload. Overall, the present results demonstrate that the piecewise linear DG schemes are highly parallelizable and a better choice for parallelization.

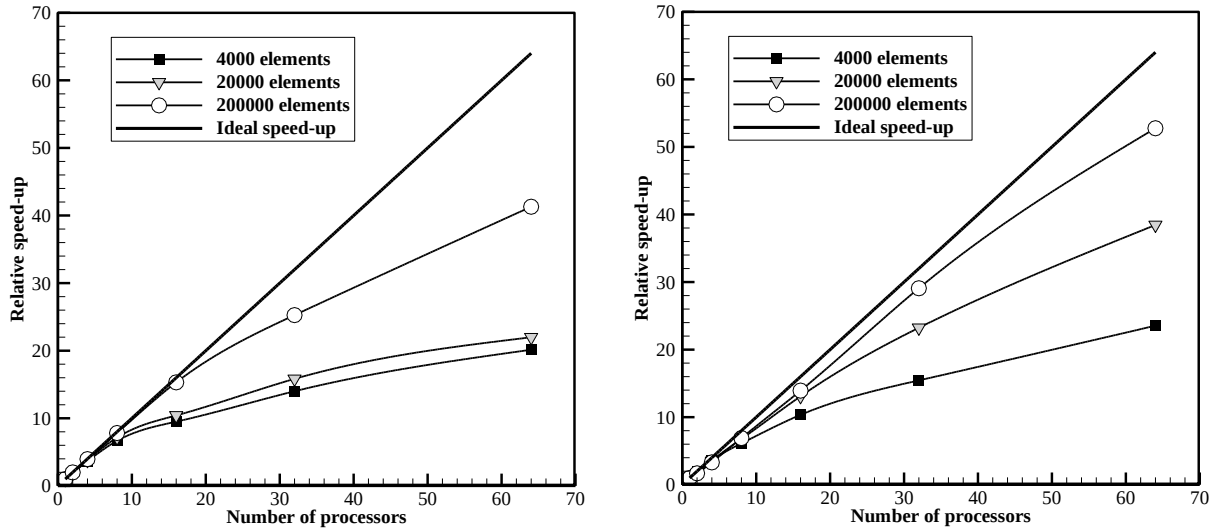


Figure 15. Parallel speed-up, S_p, for NSF (a) DG piecewise constant, and (b) DG piecewise linear scheme.

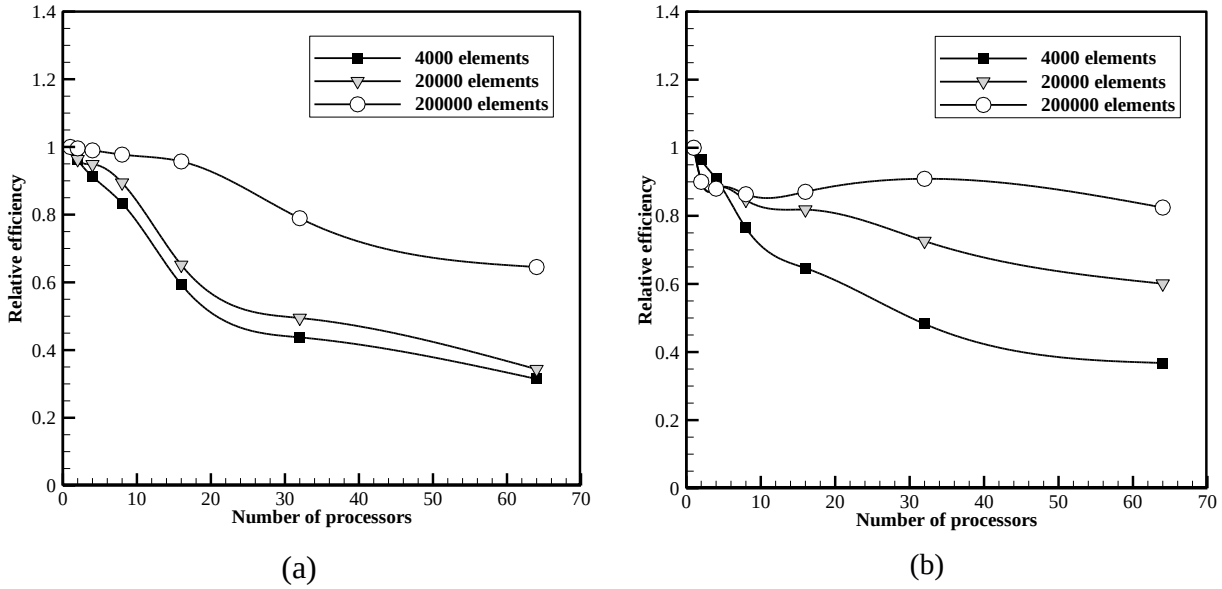


Figure 16. Parallel relative efficiency, *E*, for NSF (a) DG piecewise constant scheme, and (b) DG piecewise linear scheme.

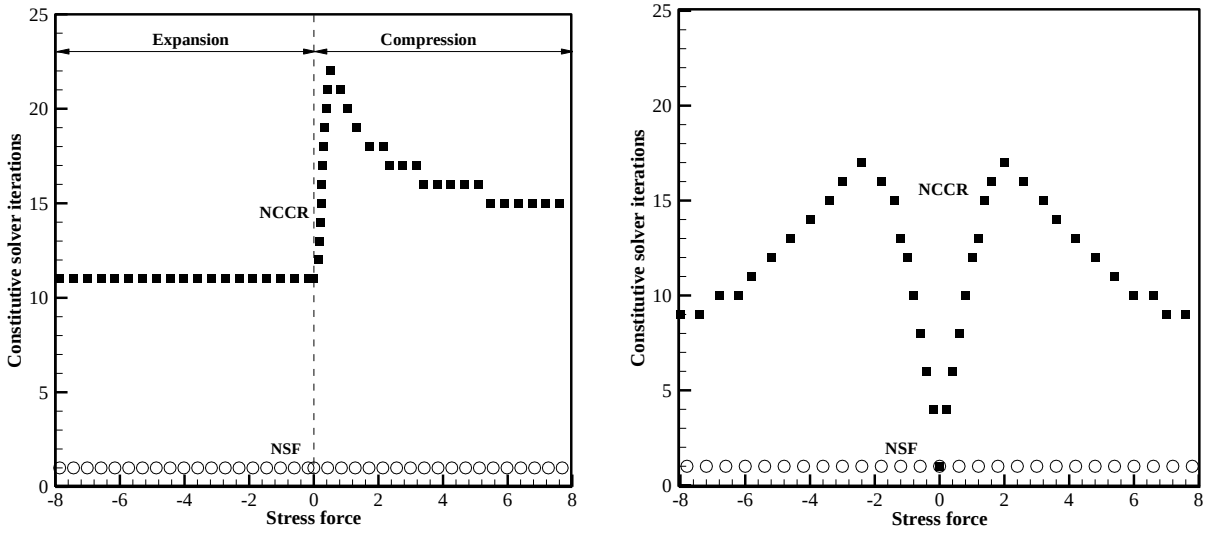
4.3 Super-parallel performance of the NCCR solver and its origin

4.3.1. Dependence of number of iterations on the flow structure

As explained in detail in subsection 2.2 of iterative numerical solvers for the second-order nonlinear coupled constitutive relations, the present method involves various iterative procedures for elementary flow structures; for example,

$$\hat{R}_{n+1} = \frac{1}{c}\sinh^{-1}\left[c\left(\hat{\Pi}_{xx_n}+1\right)\hat{R}_0\right],$$
$$\hat{\Pi}_{xx_{n+1}} = \frac{\hat{\Pi}_{xx_0}}{q_{2\text{nd}}(c\hat{R}_n)-\hat{\Pi}_{xx_0}}, \tag{4.1}$$
$$\hat{\Pi}_{xx_{n+1}} = \frac{\hat{\Pi}^2_{xy_0}}{3q^2_{2\text{nd}}(c\hat{R}_n)/2+\hat{\Pi}_{xy_0}}.$$

Therefore, the actual number of iterations in the NCCR solver varies significantly, depending on the flow structure (compression, expansion, velocity shear) and the degree of thermal non-equilibrium, as shown in Figs. 17(a) and 17(b). As seen in Fig 1(a), the normal stress is almost constant in expansion regime, and hence the number of required iteration for obtaining the NCCR solution is almost constant as shown in Fig 17(a). On the other hand, in compression regime, the normal stress predicted by the NCCR model varies with respect to the thermodynamic force, and hence the number of NCCR iterations differs significantly. Further, as shown in Fig 17 (b), the number of NCCR iterations in velocity shear case increases steeply in transition regime and then gradually reduces in free molecular regime. From these figures, one can conclude that the number of required iteration for obtaining the NCCR solution highly depends on the physics of the flow problem and it can vary in the domain depending on the local flow behavior at any target location.



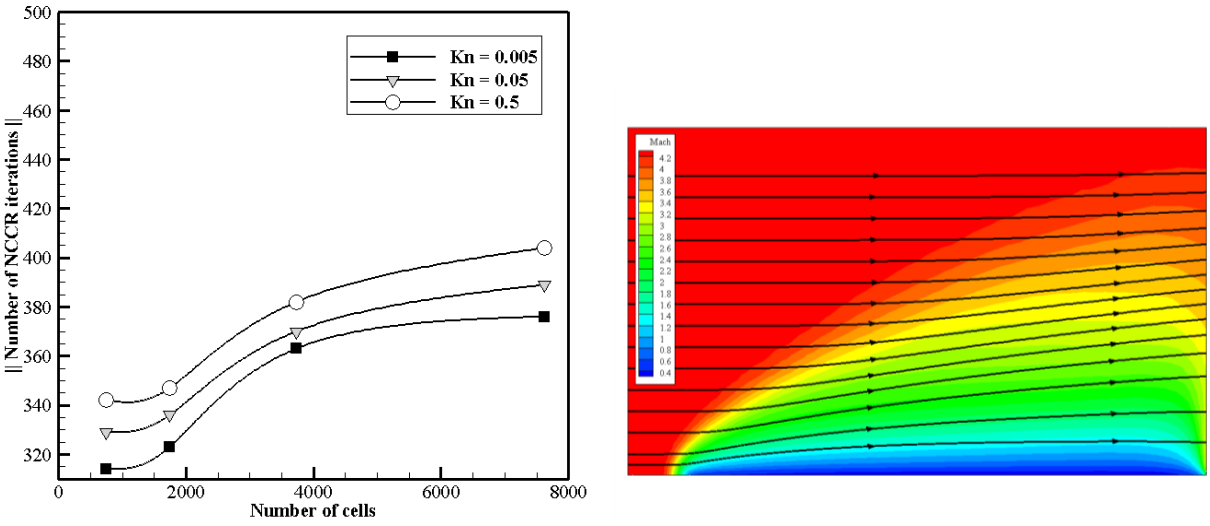
(a) (b)

Figure 17. Number of iterations for the calculation of non-conservative variables in NSF and NCCR: (a)

compression-expansion flow problem and (b) shear flow problem. The vertical axis shows the number of iterations and the horizontal axis shows the stress force.

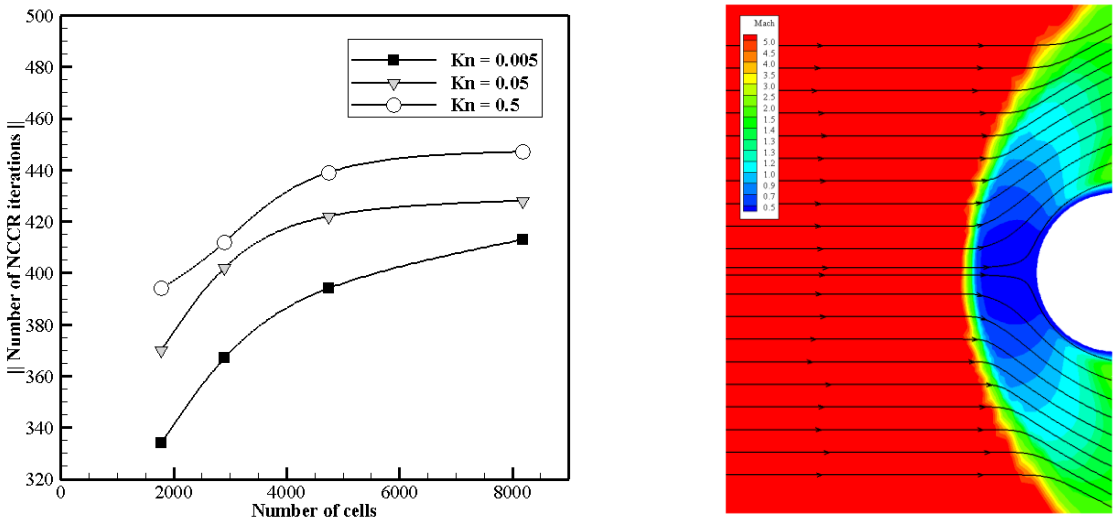
In order to illustrate the variation of number of NCCR iterations based on the flow characteristics of the problem, three benchmark problems with different flow features were considered. The flow over a flat plate, dominated with velocity-shear feature near the wall boundary, is simulated for the Mach number 4.37 at three different Knudsen numbers (0.005, 0.05 and 0.5). Unstructured meshes with uniform triangles were used for studying the computational cost of NCCR solver. The number of NCCR iterations in whole domain including the effects of all Gaussian quadrature points in a cell is shown in Fig 18(a).

Since the velocity shear is a dominant physical feature in this problem, the viscous shear stress varies considerably from tip to end of the flat plate. It is obvious in Fig. 18 that the number of required NCCR iterations increases by increasing more grid and Knudsen number (in slip and transition regimes) in the domain. The reason for this increment is that flow is more resolved by refining the grid and the NCCR solution is more deviated from the equilibrium state by increasing the Knudsen number which may cause to more iteration for finding the converged solution using the velocity-shear NCCR solver given in (2.15).



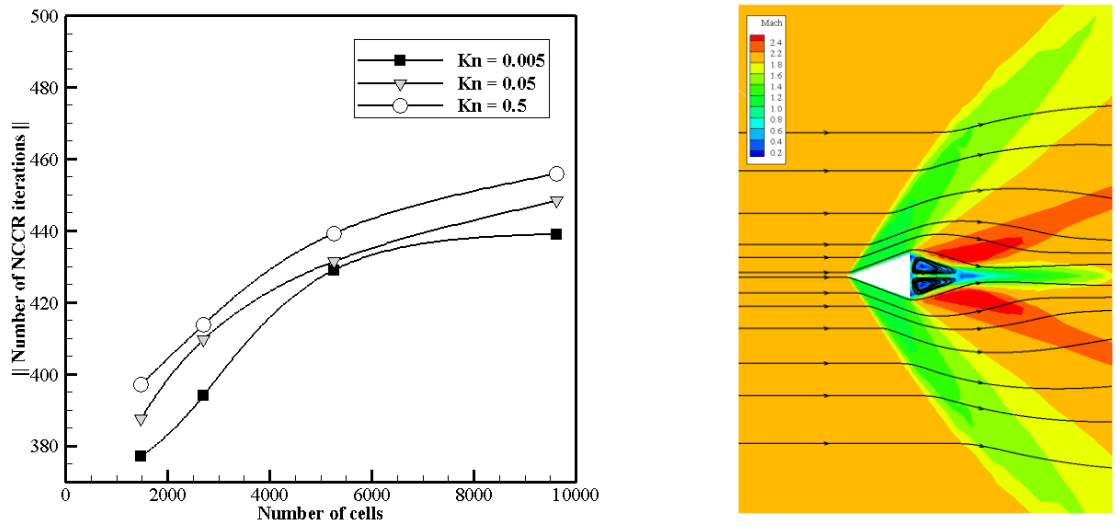
(a) (b)

Figure 18. (a) Comparison of NCCR iterations for different number of elements under different Knudsen numbers, and (b) flow field over a flat plate at M = 4.37 (and Kn = 0.05). In (a), the vertical axis represents the number of NCCR iterations in an element including all Gaussian quadrature points while the horizontal axis represents the number of elements.



(a) (b)

Figure 19. (a) Comparison of NCCR iterations for different number of elements under different Knudsen numbers, and (b) flow field over a circular cylinder at M = 5.48 (and Kn = 0.05). In (a), the vertical axis represents the number of NCCR iterations in an element including all Gaussian quadrature points while the horizontal axis represents the number of elements.



(a) (b)

Figure 20. (a) Comparison of NCCR iterations for different number of elements under different Knudsen numbers, and (b) flow field over a wedge at M = 2.0 (and Kn = 0.05). In (a), the vertical axis represents the number of NCCR iterations in an element including all Gaussian quadrature points while the horizontal axis represents the number of elements.

In Figs 19(a) and 19(b), the number of NCCR iterations and the Mach number contours in the flow over a circular cylinder are presented in case of the Mach number 5.48. The flow over a cylinder is dominated with compression flow feature associated with a bow shock wave region and the stagnation region near the frontal part. Thus, the variation in the number of NCCR iterations is closely related to the NCCR solver of compression given in (2.13). It is shown that the number of NCCR iterations increases with the cell numbers, since more cells intended to resolve the stiff region are placed in the compression region. Moreover, the comparison of the flat plate and the cylinder flow problems shows that the number of NCCR iterations of the compression dominated problem is higher than that of the velocity-shear dominated problem. The reason behind this disparity can be explained by comparing Figs. 17(a) and 17 (b) in which the number of NCCR iterations is shown the largest in case of the compression case.

Lastly, the supersonic flow over a wedge, dominated with both the compression in the frontal part and the expansion in the rear part, is simulated for the Mach number 2.0. Fig. 20(a) shows that, similar to the previous problems, the number of NCCR iterations increases with the Knudsen number. Interestingly, it can be observed that the number of NCCR iterations in case of the wedge flow varies more drastically compared to two previous problems, probably due to co-existence of the compression and expansion features in the problem.

In summary, Figs. 18 (a), 19(a), and 20(a) illustrate that the number of NCCR iterations in an element must converge to a certain value if the number of elements used in the simulation is sufficiently large to resolve the flow features accurately. They also illustrate that the number of iterations required for capturing the NCCR solution varies depending on the dominant flow structure – compression, expansion and velocity shear – and the degree of thermal non-equilibrium.

On the other hand, in the case of the NSF model, the non-conservative variables (the viscous stress and heat flux) are simply obtained from linear explicit equations (2.5) and (2.6). Hence, the computation of non-conservative variables at a point can be done in a single step (i.e., just one iteration) as shown in Figs. 17(a) and 17(b).

4.3.2. Origin of super-parallel performance of the NCCR solver

For the NSF model, when CF_{NSF}^{local}, CF_{NSF}^{global} are introduced to represent the local and global computational cost functions at a single point (e.g., Gaussian quadrature point), respectively, they may be expressed as

$$CF_{\text{NSF}}^{local}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=1, \tag{4.2}$$

$$CF_{\text{NSF}}^{global}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=\sum_{i=1}^{N_{\text{element}}}\sum_{j=1}^{N_{\text{GP}}}CF_{\text{NSF}}^{local}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right), \tag{4.3}$$

where N_{element} is the number of elements and N_{GP} is the number of Gaussian quadrature points. Since CF_{NSF}^{local} is constant, equation (4.3) can be reduced to

$$CF_{\text{NSF}}^{global}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=N_{\text{element}}N_{\text{GP}}CF_{\text{NSF}}^{local}=N_{\text{element}}N_{\text{GP}}. \tag{4.4}$$

For two different test cases with N_1 and N_2 number of elements, the cost ratio in the NSF model can be expressed as

$$\text{Cost ratio}_{\text{NSF}}=\frac{N_2}{N_1}. \tag{4.5}$$

Thus the cost ratio in the NSF solver is linear, implying that the computational cost increases linearly with the ratio of the number of elements. Fig. 21(a) illustrates the global cost of the NSF solver for various test cases with different numbers of elements. It clearly shows that the computational cost is a linear function with respect to the number of elements for both piecewise constant and piecewise linear polynomial approximations.

As explained before, however, the non-conservative variables in the NCCR model are obtained from implicit nonlinear equations solved by the method of iterations. As shown in Figs. 17(a) and 17(b), the number of iterations for convergence varies with respect to the thermodynamic force of velocity gradients in the case of compression and velocity shear, whereas it remains almost constant in the case of expansion. The local and global cost function of NCCR in the expansion case, from Fig. 17(a), can be given by

$$CF^{local}_{\mathrm{NCCR}_{\mathrm{expansion}}}(\boldsymbol{\Pi},\boldsymbol{Q}) = 11, \tag{4.6}$$

$$CF^{global}_{\mathrm{NCCR}_{\mathrm{expansion}}}(\boldsymbol{\Pi},\boldsymbol{Q}) = \sum_{i=1}^{N_{\mathrm{element}}}\sum_{j=1}^{N_{\mathrm{GP}}} CF^{local}_{\mathrm{NCCR}_{\mathrm{expansion}}}(\boldsymbol{\Pi},\boldsymbol{Q}). \tag{4.7}$$

Since $CF^{local}_{\mathrm{NCCR}_{expansion}}$ is constant, equation (4.7) can be reduced to

$$CF^{global}_{\mathrm{NCCR}_{\mathrm{expansion}}}(\boldsymbol{\Pi},\boldsymbol{Q}) = N_{\mathrm{element}}\ N_{\mathrm{GP}} CF^{local}_{\mathrm{NCCR}_{\mathrm{expansion}}} = 11\cdot N_{\mathrm{element}} N_{\mathrm{GP}}. \tag{4.8}$$

Consequently, for two different cases with N_1 and N_2 number of elements, the cost ratio of NCCR in the expansion case reduces

$$\text{Cost ratio}_{\mathrm{NCCR}_{\mathrm{expansion}}} = \frac{N_2}{N_1}. \tag{4.9}$$

Therefore, the cost ratio of NCCR in the expansion case is linear and the global computational cost follows the same trend as the NSF solver.

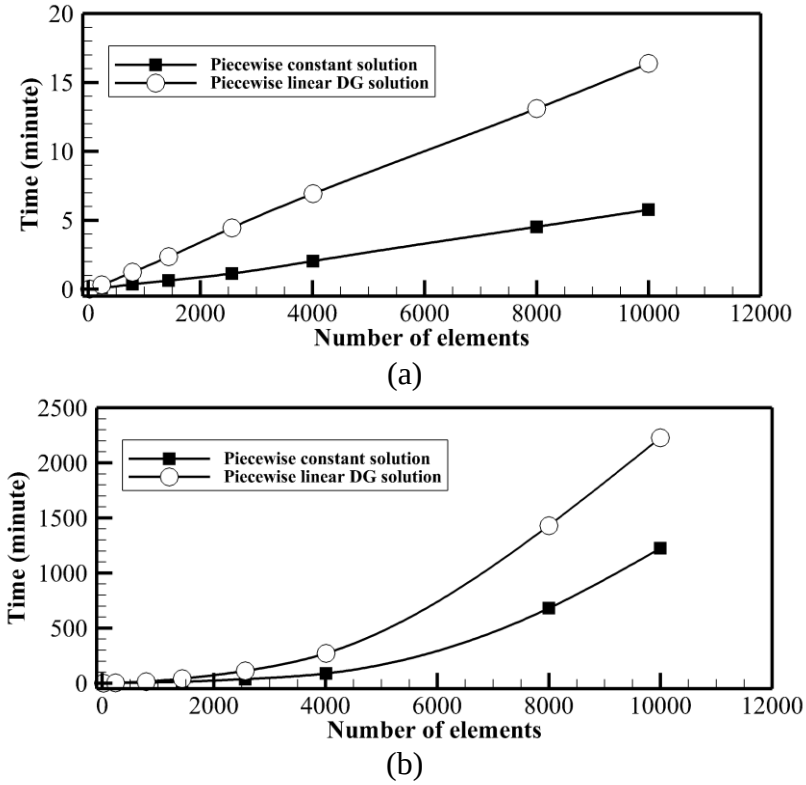


Figure 21. Computational cost of solving (a) the linear constitutive relations, and (b) the non-linear coupled constitutive relations including the cost of solving constitutive relations and the RK-DG steps, for 5000 time-marching iterations with different number of elements around a cylinder at M = 5.48, and Kn = 0.5. The vertical axis represents the serial computational time (in minutes), and the horizontal axis represents the number of elements.

Interestingly, the local cost function of NCCR in the compression case is highly nonlinear, as shown in Fig. 17(a). Therefore, the local and global cost functions of NCCR in this case should be expressed differently

$$CF^{local}_{\mathrm{NCCR}_{\mathrm{compression}}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=\Xi_{\mathrm{nonlinear}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right), \tag{4.10}$$

$$CF^{global}_{\mathrm{NCCR}_{\mathrm{compression}}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=\sum_{i=1}^{N_{\mathrm{element}}}\sum_{j=1}^{N_{\mathrm{GP}}}CF^{local}_{\mathrm{NCCR}_{\mathrm{compression}}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right). \tag{4.11}$$

For two different cases with N_1 and N_2 number of elements, the cost ratio of NCCR in the compression case then reduces

$$\mathrm{Cost\ ratio}_{\mathrm{NCCR}_{\mathrm{compression}}}=\frac{\sum_{i=1}^{N_2}\sum_{j=1}^{N_{GP}}\Xi_{\mathrm{nonlinear}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)}{\sum_{i=1}^{N_1}\sum_{j=1}^{N_{GP}}\Xi_{\mathrm{nonlinear}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)}\rightarrow\text{Nonlinear function.} \tag{4.12}$$

Similarly, the local cost function of NCCR in the velocity shear case is highly nonlinear as shown in Fig. 17(b). The local and global cost functions can be expressed as

$$CF^{local}_{\mathrm{NCCR}_{\text{velocity shear}}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=\Xi_{\mathrm{nonlinear}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right), \tag{4.13}$$

$$CF^{global}_{\mathrm{NCCR}_{\text{velocity shear}}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=\sum_{i=1}^{N_{\mathrm{element}}}\sum_{j=1}^{N_{\mathrm{GP}}}CF^{local}_{\mathrm{NCCR}_{\text{velocity shear}}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right). \tag{4.14}$$

For the two different cases with N_1 and N_2 number of elements, the cost ratio of NCCR in the velocity shear case reduces

$$\mathrm{Cost\ ratio}_{\mathrm{NCCR}_{\text{velocity shear}}}=\frac{\sum_{i=1}^{N_2}\sum_{j=1}^{N_{\mathrm{GP}}}\Xi_{\mathrm{nonlinear}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)}{\sum_{i=1}^{N_1}\sum_{j=1}^{N_{\mathrm{GP}}}\Xi_{\mathrm{nonlinear}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)}\rightarrow\text{Nonlinear function.} \tag{4.15}$$

In summary, since the global cost function is a nonlinear function of the number of elements, the cost ratio cannot be expressed explicitly, although it can be measured empirically. The total cost function of the NCCR model can be expressed as the sum of all cost functions of elementary flows (compression, expansion, velocity shear),

$$CF^{total}_{\mathrm{NCCR}}\left(\boldsymbol{\Pi},\boldsymbol{Q}\right)=CF_{\mathrm{NCCR}_{\mathrm{compression}}}+CF_{\mathrm{NCCR}_{\mathrm{expansion}}}+CF_{\mathrm{NCCR}_{\mathrm{velocity\ shear}}}. \tag{4.16}$$

Similarly, the total cost ratio of the NCCR model can be given as

$$\text{Total cost ratio}_{\mathrm{NCCR}}=\text{Cost ratio}_{\mathrm{NCCR}_{\mathrm{compression}}}+\text{Cost ratio}_{\mathrm{NCCR}_{\mathrm{expansion}}}+\text{Cost ratio}_{\mathrm{NCCR}_{\mathrm{velocity\ shear}}}. \tag{4.17}$$

This relation indicates that the total computational cost of NCCR in a flow problem is influenced by flow characteristics—a combination of three elementary flows—and depends nonlinearly on the number of elements.

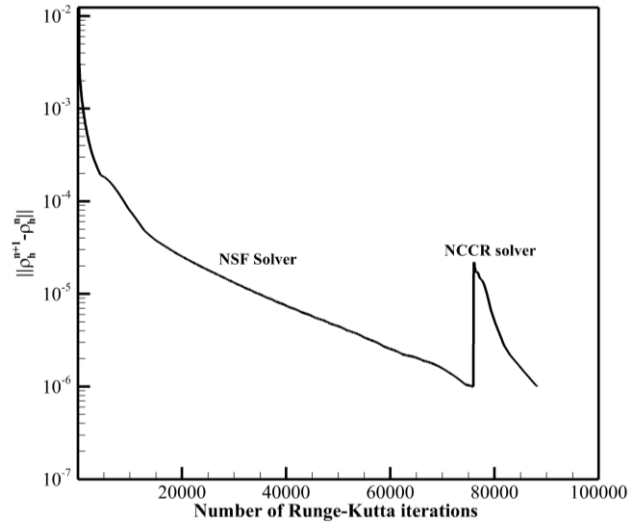


Figure 22. Comparison of required number of iterations for convergence of NSF and NCCR solutions.

Fig. 21(b) shows the global computational cost of NCCR measured empirically for various cases with different numbers of elements. It can be seen that the computational cost of NCCR does not change linearly with respect to the number of elements either for piecewise constant or piecewise linear polynomial approximations. In fact, the numerical experiment shows that the computational cost of the serial NCCR solver increases exponentially with the increasing number of elements, and it is always higher than that of the serial NSF solver. Moreover, the computational cost of the piecewise linear approximation is considerably higher than that of the piecewise constant approximation as expected. This is because a greater number of iterations are needed at each of the Gaussian quadrature points on the element interfaces, and inside the volume of the elements, as shown in Figs. 5 and 6. Consequently,

additional computing time is required to obtain the converged solution to the nonlinear constitutive relations. The NCCR solver always requires more computational time as compared to NSF solver when both of the solvers start with the same initial conditions and problem set-up. Nonetheless, it is possible to reduce the cost of NCCR for some steady-state problems in which the NCCR solver runs after convergence of NSF solver, and the NSF solution is used as the initial condition for the NCCR solver. In this case, the additional required number of Runge-Kutta iterations for obtaining a converged NCCR solution rarely exceeds 30% of the serial NSF solver as demonstrated in Fig. 22. As shown in Fig. 1 of constitutive relations, the solution of the linear (NSF) and nonlinear (NCCR) constitutive relations are similar near the origin. Thus only a small number of iterations is required to obtain the numerical solution of the NCCR if the NSF solution is used as the initial guess for the NCCR solvers. In other words, the solution of the NCCR near continuum regime will converge quickly after few RK-DG iterations when the NSF solution is used as the initial guess.

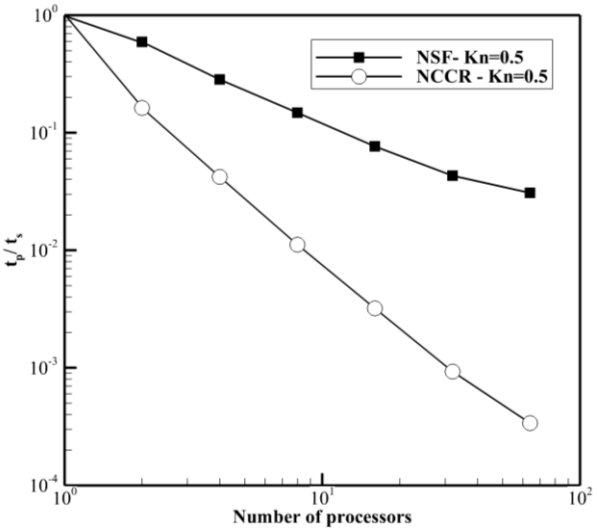


Figure 23. Comparison of the cost reduction between NSF and NCCR parallel solvers with 20,000 number of elements around a cylinder at M = 5.48, and Kn = 0.5.

Fig. 23 depicts the rate of cost reduction in the parallel NCCR and NSF solvers. In the ideal case, the NSF solver approaches a linear reduction of computational cost, which is compatible with the Amdahl's law [45]. On the other hand, very surprisingly, the computational cost of the NCCR solver reduces much

faster, almost exponentially, as the number of processors increases. This super-parallel performance is due to the nonlinear behavior of the cost of the NCCR solver, which demands less computational effort for *smaller* numbers of elements, as illustrated in Fig. 21 (b). Therefore, decomposing the domain into several sub-domains will boost the convergence rate. Put another way, the extra cost incurred from the iterative solver on implicit algebraic NCCR becomes negligible for the parallel computations with larger processors.

Fig. 24 illustrates the normalized computational cost of the paralleled NCCR solver for three different Knudsen numbers. It is shown that the higher Knudsen number causes more computational cost in any point in the domain due to requirement of more NCCR iterations for convergence of the NCCR solver. This extra computational cost does not have a significant influence on general cost behaviors and the consequent parallel performance of the NCCR solver, since it does not affect the diversity of the number of NCCR iterations throughout the domain but adds only few number of NCCR iterations in high Knudsen cases. Therefore, once the number of processor is large enough, the performance is not varying much, since the communication overhead of parallel process becomes dominant as compared with the extra computations associated with increase of the Knudsen number.

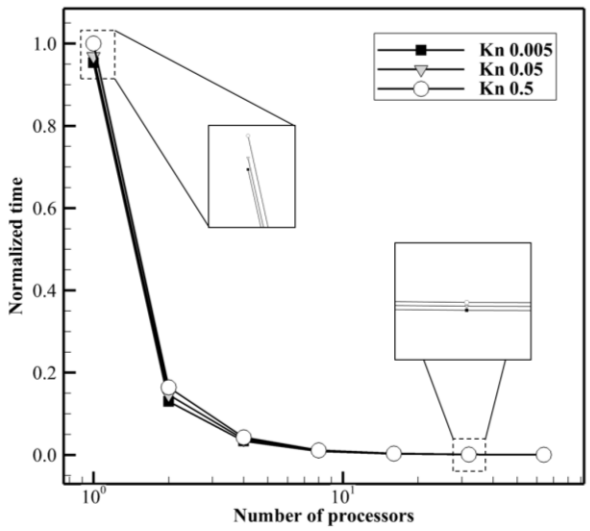


Figure 24. Normalized computational cost of parallel NCCR solvers with 20,000 number of elements around a cylinder at M = 5.48, and Kn = 0.5, 0.05, 0.005.

Finally, Fig. 25 shows the normalized computational cost of the parallel NSF and NCCR solvers for different numbers of processors. It is obvious that the cost of the NCCR solver reduces with a much higher rate than that of the NSF solver, for both the piecewise constant and piecewise linear approximations. It is also shown that the cost rate of the NSF solver decreases slowly with the increasing number of processors, and may approach to a constant rate earlier than the NCCR solver.

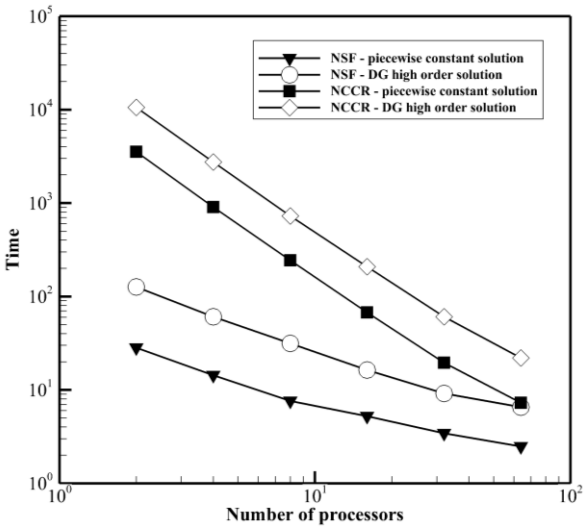


Figure 25. Comparison of the cost reduction between NSF and NCCR parallel solvers with 20,000 number of elements around a cylinder at M = 5.48, and Kn = 0.5.

In summary, because of the super-parallel performance associated with the nonlinear behavior of the cost, the NCCR solver has much higher potential for parallelization than the NSF model. According to Padua [50], the numerical methods or algorithms used can contribute to the super-parallel performance to a certain degree. Therefore, the super-parallel performance may be reduced by employing advanced numerical methods to solve the NCCR equations in less number of iterations.

5. Conclusion

An efficient SPMD-based parallel mixed explicit DG solver of the conservation laws in conjunction with first-order NSF and second-order implicit NCCR constitutive laws has been developed for simulating rarefied and microscale gases in continuum and transition regimes within a single framework.

Since a significant amount of computational work occurs locally at each element, the DG method can be developed to minimize the communication overload. The present parallel DG solver adopts domain decomposition and the MPI library for parallelization. The parallel DG solver demonstrates an acceptable level of performance, owing to its compactness, as well as the minimization of the serial parts of the coding, and the use of non-blocking parallel communication routines. The current parallel implementation of the mixed modal DG method is intended for a stand-alone PCs equipped with a large number of processing cores, as well as for clusters of PCs.

The multi-dimensional parallel explicit DG solver of the conservation laws in conjunction with NSF and NCCR constitutive laws was employed to investigate the flow field and surface properties on a cylinder in rarefied and microscale conditions. The numerical results demonstrate that the structure of the stand-off shock wave in the NCCR model is thicker in comparison with the NSF model in the transition regime, although the difference between the NSF and NCCR models is negligible in the continuum and near-continuum regimes. Moreover, due to the shear-thinning property of the second-order NCCR model in comparison with the first-order NSF model, smaller wall shear stress was observed for the NCCR model in the transition regime.

The computational cost of the NCCR and NSF solvers was investigated in the serial and parallel frameworks. It was shown that the additional number of iterations of the serial NCCR solver rarely exceeds 30% of the serial NSF solver, although the computational cost of single iteration of the NCCR solver is higher than that of the NSF solver. It was also shown that the computational cost of the NCCR solver behaves nonlinearly with respect to the number of elements, even for serial code. This is because the number of iterations in the NCCR solver varies depending on the flow structure (compression, expansion, velocity shear) and the degree of thermal non-equilibrium. Putting it differently, the diversity of the distribution of the number of NCCR iterations throughout the domain resulted in the nonlinear performance of the NCCR solver and, ultimately, a super-convergence in parallel processing. The observations also indicate that, in addition to the nonlinear behavior of NCCR cost function, the

numerical methods (or algorithms) used in the NCCR solver affect the super-parallel performance. Hence, the present super-parallel performance may be reduced when advanced numerical methods capable of solving the NCCR equations in less number of iterations are employed. However, the super-parallel performance will basically remain intact, due to the nonlinear behavior of the NCCR cost function. Furthermore, it should be mentioned that the computational cost can be nonlinearly affected by the way of grid refinement. We believe this super-parallel performance to be reported for the first time in the literature.

Overall, the parallelization of the NCCR solver in the explicit DG framework was shown to benefit from a drastic cost reduction in comparison with the NSF solver, due to its super-parallel performance. Hence, the parallel NCCR solver is much more efficient than the serial solver, and it may be employed for solving large scale industrial problems with computational cost comparable to the ordinary parallel NSF solver.

Based on the successful implementation of the parallel explicit DG solvers in the present work, applications of these methods to rarefied and microscale gas flows of complex geometry will be the next topic of the present line of research. Further, the current methodology may be applicable to the efficient computational simulation of turbulent flows, in particular, for the second-order Reynolds stress model, since similar types of extra nonlinear equations of Reynolds turbulent stress need to be calculated in conjunction with the conservation laws. We hope to report the results of studies of these problems in due course.

Acknowledgments

This work was supported by the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (NRF 2015-M1A3A3A02-010621, 2017-R1A2B2007634, and NRF-2017R1A5A1015311), South Korea.

References

[1] Reese JM, Gallis MA, Lockerby DA. New directions in fluid dynamics: non-equilibrium aerodynamic and microsystem flows. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. 2003;361:2967-88.

[2] Schmisseur JD. Hypersonics into the 21st century: A perspective on AFOSR-sponsored research in aerothermodynamics. Progress in Aerospace Sciences. 2015;72:3-16.

[3] Farber K, Farber P, Gräbel J, Krick S, Reitz J, Ueberholz P. Development and validation of a coupled Navier–Stokes/DSMC simulation for rarefied gas flow in the production process for OLEDs. Applied Mathematics and Computation. 2016;272:648-56.

[4] Rovenskaya OI. Comparative analysis of the numerical solution of full Boltzmann and BGK model equations for the Poiseuille flow in a planar microchannel. Computers & Fluids. 2013;81:45-56.

[5] Liu H, Xu K, Zhu T, Ye W. Multiple temperature kinetic model and its applications to micro-scale gas flows. Computers & Fluids. 2012;67:115-22.

[6] Le NT, Xiao H, Myong RS. A triangular discontinuous Galerkin method for non-Newtonian implicit constitutive models of rarefied and microscale gases. Journal of Computational Physics. 2014;273:160-84.

[7] Eu BC. Kinetic Theory and Irreversible Thermodynamics: J. Wiley; 1992.

[8] Xiao H, Myong RS. Computational simulations of microscale shock–vortex interaction using a mixed discontinuous Galerkin method. Computers & Fluids. 2014;105:179-93.

[9] Myong RS. A full analytical solution for the force-driven compressible Poiseuille gas flow based on a nonlinear coupled constitutive relation. Physics of Fluids. 2011;23:012002.

[10] Rana A, Ravichandran R, Park J, Myong RS. Microscopic molecular dynamics characterization of the second-order non-Navier-Fourier constitutive laws in the Poiseuille gas flow. Physics of Fluids (1994-present). 2016;28:082003.

[11] Myong RS. Theoretical description of the gaseous Knudsen layer in Couette flow based on the second-order constitutive and slip-jump models. Physics of Fluids. 2016;28:012002.

[12] Myong RS. Thermodynamically consistent hydrodynamic computational models for high-Knudsen-number gas flows. Physics of Fluids. 1999;11:2788-802.

[13] Myong RS. On the high Mach number shock structure singularity caused by overreach of Maxwellian molecules. Physics of Fluids. 2014;26:056102.

[14] Myong RS. A computational method for Eu's generalized hydrodynamic equations of rarefied and microscale Gasdynamics. Journal of Computational Physics. 2001;168:47-72.

[15] Myong RS. A generalized hydrodynamic computational model for rarefied and microscale diatomic gas flows. Journal of Computational Physics. 2004;195:655-76.

[16] Myong RS. Impact of computational physics on multi-scale CFD and related numerical algorithms. Computers & Fluids. 2011;45:64-9.

[17] Ahn JW, Kim C. An axisymmetric computational model of generalized hydrodynamic theory for rarefied multi-species gas flows. Journal of Computational Physics. 2009;228:4088-117.

[18] Navier C. Mémoire sur les lois du mouvement des fluides. Mémoires de l'Académie Royale des Sciences de l'Institut de France. 1823;6:389-440.

[19] Fourier J. Théorie Analytique Dela Chaleur. Didot, Paris. 1822.

[20] Blazek J. Computational Fluid Dynamics: Principles and Applications: Elsevier; 2005.

[21] Toro EF. Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction: Springer Science & Business Media; 2009.

[22] Cockburn B, Karniadakis GE, Shu C-W. The Development of Discontinuous Galerkin Methods: Springer; 2000.

[23] Liu H, Xu K. A Runge–Kutta discontinuous Galerkin method for viscous flow equations. Journal of Computational Physics. 2007;224:1223-42.

[24] Landmann B, Kessler M, Wagner S, Krämer E. A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows. Computers & Fluids. 2008;37:427-38.

[25] Richter GR. The discontinuous Galerkin method with diffusion. Mathematics of Computation. 1992;58:631-43.

[26] Dawson C. Godunov-mixed methods for advection-diffusion equations in multidimensions. SIAM Journal on Numerical Analysis. 1993;30:1315-32.

[27] Lomtev I, Quillen C, Karniadakis G. Spectral/hp methods for viscous compressible flows on unstructured 2D meshes. Journal of Computational Physics. 1998;144:325-57.

[28] Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. Journal of Computational Physics. 1997;131:267-79.

[29] Biswas R, Devine KD, Flaherty JE. Parallel, adaptive finite element methods for conservation laws. Applied Numerical Mathematics. 1994;14:255-83.

[30] Bey KS, Oden JT, Patra A. A parallel hp-adaptive discontinuous Galerkin method for hyperbolic conservation laws. Applied Numerical Mathematics. 1996;20:321-36.

[31] Baggag A, Atkins H, Keyes D. Parallel implementation of the discontinuous Galerkin method: Institute for Computer Applications in Science and Engineering, NASA Langley Research Center; 1999.

[32] Luo H, Luo L, Ali A, Nourgaliev R, Cai C. A parallel, reconstructed discontinuous Galerkin method for the compressible flows on arbitrary grids. Communications in Computational Physics. 2011;9:363-89.

[33] Su W, Alexeenko AA, Cai G. A parallel Runge–Kutta discontinuous Galerkin solver for rarefied gas flows based on 2D Boltzmann kinetic equations. Computers & Fluids. 2015;109:123-36.

[34] Dubiner M. Spectral methods on triangles and other domains. Journal of Scientific Computing. 1991;6:345-90.

[35] Karniadakis G, Sherwin S. Spectral/hp Element Methods for Computational Fluid Dynamics: Oxford University Press; 2013.

[36] Cockburn B, Shu C-W. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. Journal of Computational Physics. 1998;141:199-224.

[37] Karchani A, Myong RS. Performance of limiters in modal discontinuous Galerkin methods for 1-D Euler equations. Journal of Computational Fluids Engineering. 2016;21:1-11.

[38] Zhang X, Shu C-W. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. Journal of Computational Physics. 2010;229:8918-34.

[39] Zhang X, Xia Y, Shu C-W. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. Journal of Scientific Computing. 2012;50:29-62.

[40] Gottlieb S, Shu C-W. Total variation diminishing Runge-Kutta schemes. Mathematics of computation of the American Mathematical Society. 1998;67:73-85.

[41] Hesthaven JS, Warburton T. Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications: Springer Science & Business Media; 2007.

[42] Löhner R, Baum JD. Handling tens of thousands of cores with industrial/legacy codes: Approaches, implementation and timings. Computers & Fluids. 2013;85:53-62.

[43] Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing. 1998;20:359-92.

[44] Grama A. Introduction to Parallel Computing: Pearson Education; 2003.

[45] Amdahl GM. Validity of the single processor approach to achieving large scale computing capabilities. Proceedings of Spring Joint Computer Conference: ACM; 1967. p. 483-5.

[46] Bird G, Broadwell J, Rungaldier H, Vogenitz F. Theoretical and experimental study of rarefied supersonic flows about several simple shapes. AIAA Journal. 1968;6:2388-94.

[47] Myong RS. Gaseous slip models based on the Langmuir adsorption isotherm. Physics of Fluids. 2004;16:104-17.

[48] Karchani A, Myong RS. Convergence analysis of the direct simulation Monte Carlo based on the physical laws of conservation. Computers & Fluids. 2015;115:98-114.

[49] Wetzel W, Oertel H. Direct Monte Carlo simulations of hypersonic flows past blunt bodies. Progr Astronaut Aeronaut. 1989;118:432-46.

[50] Padua D. Encyclopedia of Parallel Computing: Springer Science & Business Media; 2011.