International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:5, No:7, 2011

Parallel Computation in Hypersonic Aerodynamic Heating Problem

Ding Guo-hao, Li Hua, and Wang Wen-long

***Abstract*—**A parallel computational fluid dynamics code has been developed for the study of aerodynamic heating problem in hypersonic flows. The code employs the 3D Navier-Stokes equations as the basic governing equations to simulate the laminar hypersonic flow. The cell centered finite volume method based on structured grid is applied for spatial discretization. The AUSMPW+ scheme is used for the inviscid fluxes, and the MUSCL approach is used for higher order spatial accuracy. The implicit LU-SGS scheme is applied for time integration to accelerate the convergence of computations in steady flows. A parallel programming method based on MPI is employed to shorten the computing time. The validity of the code is demonstrated by comparing the numerical calculation result with the experimental data of a hypersonic flow field around a blunt body.

***Keywords*—**Aerodynamic Heating, AUSMPW+, MPI, Parallel Computation

I. Introduction

DUE to the rapid development of both numerical schemes and computer architectures, computational fluid dynamics (CFD) has played an important role in engineering applications, especially in the design of fight vehicle configuration, and it has been recognized as a indispensable tool to compute aerodynamic and aerothermodynamic environment for the hypersonic flight vehicles.

The accurate prediction of aerodynamic heating has always been a key issue in hypersonic flow simulations, but it is still challenging since the numerical schemes for hypersonic flow computation require a high level of robustness, accuracy and efficiency. Hypersonic flow problems generally include severe viscous dissipation in boundary layers and strong shock waves leading to the large gradient of flow properties, and yet the accurate calculation of shock discontinuity and boundary layer are contradictory in many schemes. For example, Roe's Flux Difference Splitting (FDS) [1] is a widely used scheme, however it has the so-called "carbuncle phenomena" in hypersonic flow around a blunt body. The entropy fix is introduced to remove the instability, but unfortunately it also brings an excessive numerical dissipation which can contaminate physical dissipation in boundary layers. Though the dissipative effect of entropy fix in boundary layers can be reduced by decreasing the entropy fix parameter [2], it is difficult to be extended to general conditions. To solve this problem, much effort was made to combine the accuracy of FDS in boundary layers and the robustness of FVS in the capturing of strong discontinuities. The AUSM (Advection Upstream Splitting Method) scheme was proposed by Liou and Steffen to meet this concern, and successively updated AUSM-type schemes [3]-[8] such as AUSMD/V, AUSM+, AUSMPW and AUSMPW+ were developed to overcome the intrinsic drawbacks. Because the AUSM-type schemes are based on the form of scalar dissipation, the amount of calculation for each time iteration step is smaller, i.e. the AUSM-type schemes have higher computational efficiency. The additional advantage is the facility for extending to other hyperbolic equation systems, such as turbulence and real gas calculations.

For accurate calculation of aerodynamic heating in hypersonic flow, the grid systems are commonly required to be denser than those for subsonic or supersonic flow, because the boundary layer regions where the grid spacing normal to the wall surface must meet the specification of cell Reynolds number, and the stiff gradient regions require more grids. Thus the time step is limited due to the CFL condition and then more additional computational time for a solution to converge is required. As the complexity of the vehicle figuration and the solving model increases, the amount of grids become massive and the calculating process of physical properties become difficult, thus the amount of calculation increase several times over. The development of single CPU performance can not meet the increasing need in engineering applications since the improvement of serial code enhances the efficiency quite limited, so that the parallel computation becomes an inevitable trend in development to reduce the runtime and expand the solving scale.

This paper presents an efficient parallel solver based on message passing model for speeding up the computation of hypersonic aerodynamic heating.

II. Governing Equations

The dimensionless three-dimensional Navier-Stokes equations neglecting the source term in a conservation form [9] can be expressed as

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{\boldsymbol{Q}}d\Omega+\iint_{\partial\Omega}(\vec{\boldsymbol{F}}_c-\vec{\boldsymbol{F}}_v)dS=0\cdot \quad (1)$$

The vector of conservative variables consists of the

DING Guo-hao is with College of Aerospace and Materials Engineering, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: dingguohao@hotmail.com).

LI Hua, is with College of Aerospace and Materials Engineering, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: lihuakd@tom.com).

WANG Wen-long is with College of Aerospace and Materials Engineering, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: wenxinwenxin_2008@163.com).

This work is financially support by National Natural Science Foundation of China (Grant No. 91016010).

following five components:

$$\vec{\boldsymbol{Q}} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{2}$$

where ρ is density, u, v, w are the velocity component in Cartesian coordinate system, and E is the total energy per unit mass, which is obtained by adding the internal energy per unit mass, e, to the kinetic energy per unit mass, $|\vec{\boldsymbol{v}}|^2/2$. Thus it is written as

$$E = e + \frac{|\vec{\boldsymbol{v}}|^2}{2} = e + \frac{u^2+v^2+w^2}{2}. \tag{3}$$

The vector of convective fluxes is

$$\vec{\boldsymbol{F}}_c = \begin{bmatrix} \rho V \\ \rho u V + n_x p \\ \rho v V + n_y p \\ \rho w V + n_z p \\ \rho H V \end{bmatrix} \tag{4}$$

with the static pressure p and the contravariant velocity V - the velocity normal to the surface element dS - being defined as the scalar product of the velocity vector and the unit exterior normal vector, i.e. ,

$$V \equiv \vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{n}} = n_x u + n_y v + n_z w. \tag{5}$$

The total enthalpy H is given as

$$H = h + \frac{|\vec{\boldsymbol{v}}|^2}{2} = E + \frac{p}{\rho}. \tag{6}$$

The vector of viscous fluxes is

$$\vec{\boldsymbol{F}}_v = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{yx} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{zx} + n_y\tau_{zy} + n_z\tau_{zz} \\ n_x\Theta_x + n_y\Theta_y + n_z\Theta_z \end{bmatrix} \tag{7}$$

where

$$\begin{aligned}
\tau_{xx} &= -\frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) + 2\mu\frac{\partial u}{\partial x} \\
\tau_{yy} &= -\frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) + 2\mu\frac{\partial v}{\partial y} \\
\tau_{zz} &= -\frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) + 2\mu\frac{\partial w}{\partial z} \\
\tau_{xy} &= \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\tau_{yz} &= \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\
\tau_{zx} &= \tau_{xz} = \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)
\end{aligned} \tag{8}$$

in which μ denotes the dynamic viscosity coefficient and

$$\begin{aligned}
\Theta_x &= u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x} \\
\Theta_y &= u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y} \\
\Theta_z &= u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z}
\end{aligned} \tag{9}$$

in which T represent the static temperature and k denotes the thermal conductivity coefficient.

In order to close the equation, the calorically perfect gas assumption is adopted, for which the equation of state takes the form

$$p = \rho R T \tag{10}$$

where R denotes the specific gas constant.

The enthalpy results from

$$h = c_p T \tag{11}$$

and

$$R = c_p - c_v, \quad \gamma = \frac{c_p}{c_v} \tag{12}$$

where c_p and c_v represent specific heat at constant pressure and at constant volume, respectively. γ is the specific heat ratio.

Finally the pressure can be obtained by

$$p = (\gamma - 1)\rho\left[E - \frac{u^2+v^2+w^2}{2}\right]. \tag{13}$$

The dynamic viscosity μ can be obtained by the Sutherland formula. The result for air is

$$\frac{\mu}{\mu_0} \approx \left(\frac{T}{T_0}\right)^{1.5}\left(\frac{T_0 + T_s}{T + T_s}\right) \tag{14}$$

where $T_0 = 273.16\text{K}$, $T_s = 124\text{K}$, $\mu_0 = 1.7161\times10^{-5}\text{kg/(m}\cdot\text{s)}$.

The relationship

$$\kappa = \frac{\mu c_p}{Pr} \tag{15}$$

is generally used for air, where Pr denotes the Prandtl number.

In addition, it is commonly assumed that the Prandtl number and the specific heat ratio are constant in the entire flow field. For air, $Pr = 0.72$, $\gamma = 1.4$.

III. Numerical Methods

In the present study, a structured grid system is adopted to discretize the computational space, and the finite volume method which directly utilizes the integral formulation of the Navier-Stokes equations is also used. Then considering a particular volume $\Omega_{i,j,k}$, as displayed in Fig. 1, the semi-discrete form of the equations [9] is as follows:

$$\frac{d}{dt}\vec{\boldsymbol{Q}}_{i,j,k} = \frac{1}{\Omega_{i,j,k}}\left[\sum_{m=1}^{N_F}[(\vec{\boldsymbol{F}}_c - \vec{\boldsymbol{F}}_v)_m \Delta S_m]\right]. \tag{16}$$

The term in square brackets on the right-hand side of (16) is also generally termed the residual. It is denoted here by $\vec{\boldsymbol{R}}_{i,j,k}$. Hence, Equation (16) can be abbreviated as

$$\frac{d}{dt}\vec{\boldsymbol{Q}}_{i,j,k} = \frac{1}{\Omega_{i,j,k}}\vec{\boldsymbol{R}}_{i,j,k}. \tag{17}$$

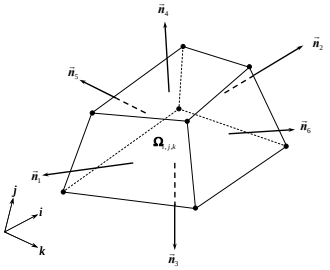


Fig. 1 Control volume and associated interface unit normal vectors for a structured grid

The convective fluxes are discretized using AUSMPW+ scheme that will be presented in detail in the following section, and the viscous fluxes are discretized using Gauss theorem [9] to evaluate the first derivatives of the velocity components and of temperature.

A. AUSMPW+ Scheme

The underlying idea of the ASUM scheme is based on the observation that the vector of convective fluxes consists of two physically distinct parts, namely the convective and the pressure part

$$\vec{\boldsymbol{F}}_c = V\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho H\end{bmatrix} + \begin{bmatrix}0\\ n_x p\\ n_y p\\ n_z p\\ 0\end{bmatrix}. \tag{18}$$

In AUSMPW+ [7], the convective flux at the interface ($I+1/2$) of the control volume in Fig. 2 can be recast as

$$\vec{\boldsymbol{F}}_{c,I+\frac{1}{2}} = \bar{M}_{\mathrm{L}}^{+} c_{I+\frac{1}{2}} \vec{\boldsymbol{\Psi}}_{\mathrm{L}} + \bar{M}_{\mathrm{R}}^{-} c_{I+\frac{1}{2}} \vec{\boldsymbol{\Psi}}_{\mathrm{R}} + \left(P_{\mathrm{L}}^{+}\vec{\boldsymbol{P}}_{\mathrm{L}} + P_{\mathrm{R}}^{-}\vec{\boldsymbol{P}}_{\mathrm{R}}\right) \tag{19}$$

where $\vec{\boldsymbol{\Psi}} = [\rho,\ \rho u,\ \rho v,\ \rho w,\ \rho H]^{\mathrm{T}}$, $\vec{\boldsymbol{P}} = [0,\ n_x p,\ n_y p,\ n_z p,\ 0]^{\mathrm{T}}$.

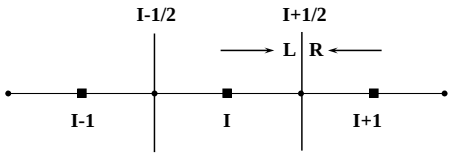


Fig. 2 Left and right state at cell interface

$\bar{M}_{\mathrm{L,R}}^{\pm}$ is the Mach number interpolation function that is written as follows:

(i) for $m_{I+\frac{1}{2}} \geq 0$

$$\begin{aligned}\bar{M}_{\mathrm{L}}^{+} &= M_{\mathrm{L}}^{+} + M_{\mathrm{R}}^{-}\cdot[(1-w)\cdot(1+f_{\mathrm{R}}) - f_{\mathrm{L}}]\\ \bar{M}_{\mathrm{R}}^{-} &= M_{\mathrm{R}}^{-}\cdot w\cdot(1+f_{\mathrm{R}})\end{aligned} \tag{20}$$

(ii) for $m_{I+\frac{1}{2}} < 0$

$$\begin{aligned}\bar{M}_{\mathrm{L}}^{+} &= M_{\mathrm{L}}^{+}\cdot w\cdot(1+f_{\mathrm{L}})\\ \bar{M}_{\mathrm{R}}^{-} &= M_{\mathrm{R}}^{-} + M_{\mathrm{L}}^{+}\cdot[(1-w)\cdot(1+f_{\mathrm{L}}) - f_{\mathrm{R}}]\end{aligned}$$

where $m_{1/2} = M_{\mathrm{L}}^{+} + M_{\mathrm{R}}^{-}$.

The Mach number splitting function $M_{\mathrm{L,R}}^{\pm}$ and the pressure splitting function $P_{\mathrm{L,R}}^{\pm}$ are given by

$$M_{\mathrm{L,R}}^{\pm} = \begin{cases}\pm\frac{1}{4}(M_{\mathrm{L,R}}\pm 1)^2, & |M_{\mathrm{L,R}}|\leq 1\\ \frac{1}{2}(M_{\mathrm{L,R}}\pm|M_{\mathrm{L,R}}|), & |M_{\mathrm{L,R}}|>1\end{cases} \tag{21}$$

$$P_{\mathrm{L,R}}^{\pm} = \begin{cases}\frac{1}{4}(M_{\mathrm{L,R}}\pm 1)^2(2\mp M_{\mathrm{L,R}}), & |M_{\mathrm{L,R}}|\leq 1\\ \frac{1}{2}(1\pm sign(M_{\mathrm{L,R}})), & |M_{\mathrm{L,R}}|>1\end{cases} \tag{22}$$

where $M_{\mathrm{L,R}} = V_{\mathrm{L,R}}/c_{1/2}$ and $c_{1/2}$ is defined as follows:

$$\begin{aligned}&\text{(i)}\ \ \frac{1}{2}(U_{\mathrm{L}}+U_{\mathrm{R}})\geq 0: \quad c_{\frac{1}{2}} = c_s^2/\max(|U_{\mathrm{L}}|, c_s)\\ &\text{(ii)}\ \ \frac{1}{2}(U_{\mathrm{L}}+U_{\mathrm{R}})<0: \quad c_{\frac{1}{2}} = c_s^2/\max(|U_{\mathrm{R}}|, c_s)\end{aligned} \tag{23}$$

Here, $c_s = \sqrt{2(\gamma-1)/(\gamma+1)H_{\mathrm{normal}}}$

and $H_{\mathrm{normal}} = \min(H_{\mathrm{total,L}} - 0.5\times V_{\mathrm{L}}^2,\ H_{\mathrm{total,R}} - 0.5\times V_{\mathrm{R}}^2)$.

The function f and w are pressure and Mach number based weighting functions, respectively.

$$f_{\mathrm{L,R}} = \begin{cases}\left(\frac{p_{\mathrm{L,R}}}{p_s}-1\right)\min\left(1, \frac{\min(p_{n,\mathrm{L/R}})}{\min(p_{\mathrm{L}}, p_{\mathrm{R}})}\right)^2, & p_s \neq 0\\ 0, & \textit{elsewhere}\end{cases} \tag{24}$$

where $p_{n,\mathrm{L/R}}$ stands for the value at the same direction interface of the adjacent grids and where $p_s = P_{\mathrm{L}}^{+}p_{\mathrm{L}} + P_{\mathrm{R}}^{+}p_{\mathrm{R}}$.

$$w(p_{\mathrm{L}}, p_{\mathrm{R}}) = 1 - \min\left(\frac{p_{\mathrm{L}}}{p_{\mathrm{R}}}, \frac{p_{\mathrm{R}}}{p_{\mathrm{L}}}\right)^3. \tag{25}$$

B. MUSCL approach and limiter function

In the discretization of convective fluxes, the left and right state values of a cell interface are utilized to compute the convective flux through the interface. The MUSCL (Monotone Upstream-Centered Schemes for Conservation Laws) approach is a higher-order linear scheme to interpolate the left and right state values [9], and the formulae are

$$\begin{aligned}\vec{\boldsymbol{\Phi}}_{\mathrm{R}} &= \vec{\boldsymbol{\Phi}}_{I+1} - \frac{1}{4}[(1+\hat{\kappa})\Delta_{-} + (1-\hat{\kappa})\Delta_{+}]\vec{\boldsymbol{\Phi}}_{I+1}\\ \vec{\boldsymbol{\Phi}}_{\mathrm{L}} &= \vec{\boldsymbol{\Phi}}_{I} + \frac{1}{4}[(1+\hat{\kappa})\Delta_{+} + (1-\hat{\kappa})\Delta_{-}]\vec{\boldsymbol{\Phi}}_{I}\end{aligned} \tag{26}$$

where the primitive variables are $\vec{\boldsymbol{\Phi}} = [\rho, u, v, w, p]^{\mathrm{T}}$ and the forward (Δ_{+}) and the backward (Δ_{-}) difference operators are defined as

$$\begin{aligned}\Delta_{+}\vec{\boldsymbol{\Phi}}_I &= \vec{\boldsymbol{\Phi}}_{I+1} - \vec{\boldsymbol{\Phi}}_I\\ \Delta_{-}\vec{\boldsymbol{\Phi}}_I &= \vec{\boldsymbol{\Phi}}_I - \vec{\boldsymbol{\Phi}}_{I-1}\end{aligned} \tag{27}$$

and the value of $\hat{\kappa}$ determines the interpolation precision,

$$\begin{cases} \hat{\kappa} = -1 & : \text{ fully upwind scheme} \\ \hat{\kappa} = 0 & : \text{ Fromm scheme} \\ \hat{\kappa} = 1/3 & : \text{ 3rd-order upwind-biased scheme} \\ \hat{\kappa} = 1 & : \text{ 3-point central difference scheme} \end{cases} \quad (28)$$

However, the MUSCL approach is no longer monotonicity preserving. It is therefore necessary to employ non-linear limiter functions in order to suppress overshoot or undershoot near discontinuities and construct a monotonicity preserving or TVD discretization. The purpose of a limiter is to reduce the slopes used to interpolate a primitive variable to the interface of a control volume. At strong discontinuities, the limiter has to reduce slopes to zero to prevent the generation of a new extremum [9]. There are many different limiter functions which are well-established and proven in practice. van Albada limiter [10] is employed in this paper, and the formulae are written as

$$\begin{aligned} \vec{\boldsymbol{\Phi}}_{\mathrm{R}} &= \vec{\boldsymbol{\Phi}}_{I+1} - \frac{s}{4}[(1+\hat{\kappa}s)\Delta_{-} + (1-\hat{\kappa}s)\Delta_{+}]_{I+1} \\ \vec{\boldsymbol{\Phi}}_{\mathrm{L}} &= \vec{\boldsymbol{\Phi}}_{I} \quad + \frac{s}{4}[(1+\hat{\kappa}s)\Delta_{+} + (1-\hat{\kappa}s)\Delta_{-}]_{I} \end{aligned} \quad (29)$$

where

$$s = \frac{2\Delta_{+}\Delta_{-} + \varepsilon}{\Delta_{+}^2 + \Delta_{-}^2 + \varepsilon}, \quad \varepsilon = 1.0\times 10^{-6} \quad (30)$$

ε is a small number that prevents division by zero and controls the sensitivity of limiter.

C. Time discretization

In the present study, the implicit LU-SGS (Lower-Upper Symmetric Gauss-Seidel) scheme introduced by Yoon and Jameson [11] is adopt to solve the Navier-Stokes equations. The LU-SGS scheme employs an implicit Newton iteration technique to solve the finite-volume approximation of the steady-state version of the governing equations, and the formula is expressed as

$$(\boldsymbol{D}+\boldsymbol{L})\boldsymbol{D}^{-1}(\boldsymbol{D}+\boldsymbol{U})\Delta\vec{\boldsymbol{Q}}^{n}_{i,j,k} = -\vec{\boldsymbol{R}}^{n}_{i,j,k} \quad (31)$$

The factors are constructed such that $\boldsymbol{L}$ consists only of the terms in the strictly lower triangular matrix, $\boldsymbol{U}$ of terms in the strictly upper triangular matrix and $\boldsymbol{D}$ of diagonal terms.

The system matrix of the LU-SGS scheme can be inverted in two steps – a forward and a backward sweep, i.e. ,

$$\begin{aligned} (\boldsymbol{D}+\boldsymbol{L})\Delta\vec{\boldsymbol{Q}}^{(1)}_{i,j,k} &= -\vec{\boldsymbol{R}}^{n}_{i,j,k} \\ (\boldsymbol{D}+\boldsymbol{U})\Delta\vec{\boldsymbol{Q}}^{n}_{i,j,k} &= \boldsymbol{D}\Delta\vec{\boldsymbol{Q}}^{(1)}_{i,j,k} \end{aligned} \quad (32)$$

with $\vec{\boldsymbol{Q}}^{n+1}_{i,j,k} = \vec{\boldsymbol{Q}}^{n}_{i,j,k} + \Delta\vec{\boldsymbol{Q}}^{n}_{i,j,k}$.

The operators $\boldsymbol{L}$, $\boldsymbol{U}$ and $\boldsymbol{D}$ are defined as

$$\begin{aligned} \boldsymbol{L} &= (\boldsymbol{A}^{+}_{c,i-1,j,k}\Delta S_{i-1/2,j,k} + \boldsymbol{A}^{+}_{c,i,j-1,k}\Delta S_{i,j-1/2,k} + \boldsymbol{A}^{+}_{c,i,j,k-1}\Delta S_{i,j,k-1/2}) \\ \boldsymbol{U} &= (\boldsymbol{A}^{-}_{c,i+1,j,k}\Delta S_{i+1/2,j,k} + \boldsymbol{A}^{-}_{c,i,j+1,k}\Delta S_{i,j+1/2,k} + \boldsymbol{A}^{-}_{c,i,j,k+1}\Delta S_{i,j,k+1/2}) \\ \boldsymbol{D} &= \frac{\Omega}{\Delta t}\boldsymbol{I} + (\boldsymbol{A}^{-}_{c,i,j,k}\Delta S_{i-1/2,j,k} + \boldsymbol{A}^{-}_{c,i,j,k}\Delta S_{i,j-1/2,k} + \boldsymbol{A}^{-}_{c,i,j,k}\Delta S_{i,j,k-1/2} \\ &\quad + \boldsymbol{A}^{+}_{c,i,j,k}\Delta S_{i+1/2,j,k} + \boldsymbol{A}^{+}_{c,i,j,k}\Delta S_{i,j+1/2,k} + \boldsymbol{A}^{+}_{c,i,j,k}\Delta S_{i,j,k+1/2}) \end{aligned} \quad (33)$$

where the split convective Jacobians $\boldsymbol{A}^{\pm}_{c}$ are constructed in such a way that the eigenvalues of the (+) matrices are all non-negative, and of the (-) matrices are all non-positive. In general, the matrices are defined as

$$\boldsymbol{A}^{\pm}_{c}\Delta S = \frac{1}{2}(\boldsymbol{A}_{c}\Delta S \pm r_{A}\boldsymbol{I}), \quad r_{A} = \omega\hat{\Lambda}_{c} \quad (34)$$

where $\boldsymbol{A}_{c}$ stands for the convective Jacobian and $\hat{\Lambda}_{c}$ represents the spectral radius of the convective flux Jacobian, respectively. ω is a constant number which is larger than 1.0.

$$\begin{aligned} \boldsymbol{A}_{c} &= \frac{\partial\vec{\boldsymbol{F}}_{c}}{\partial\vec{\boldsymbol{Q}}} = \boldsymbol{T}\boldsymbol{\Lambda}_{c}\boldsymbol{T}^{-1} \\ \hat{\Lambda}_{c} &= \max(|\boldsymbol{\Lambda}_{c}|) \end{aligned} \quad (35)$$

IV. Parallel Computation

The basic idea of parallel computation in CFD application is domain decomposition method, which can be described as follows: The computational space is split into a number of topologically simpler ones with the adoption of multi-block methodology, then the corresponding computational tasks are assigned to different CPUs, the geometrical quantities and the initial state of each block are loaded into the memory of each CPU. Subsequently the computational processes are started, instructions from each part execute simultaneously on different CPUs, which are scheduled by the main process. When the computation of the current iteration step is completed, the physical quantities or fluxes are exchanged at the interface between grid blocks by the communication between the CPUs, and the main process gathers the data of entire flow field to judge the convergence according to corresponding criterion. Writing disk or other operation is followed as necessary. In view of the above mentioned, the problems on the subdomains are regarded independent, which makes domain decomposition method suitable for parallel computing.

According to the above strategy of parallel computation, the approach of dummy cells is introduced to implement the data exchange between blocks and extend the stencil of the spatial discretization scheme beyond the boundaries (see Fig. 3). Then the computation of the fluxes, gradients, dissipation, etc. along the boundaries is simplified.

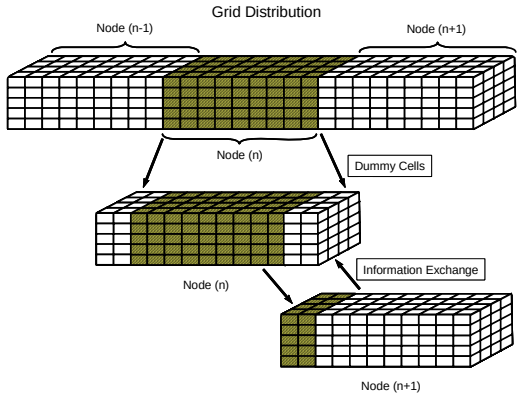


Fig. 3 Schematic of decomposed domain method using dummy cells

Message passing model is the a common used parallel programming method in CFD, which exchanges data through communications by sending and receiving messages between

tasks. Message Passing Interface (MPI) is an API specification of this model, and is a language-independent communications protocol. Message passing programs generally run the same code on multiple processors, which then communicate with one another via library calls.

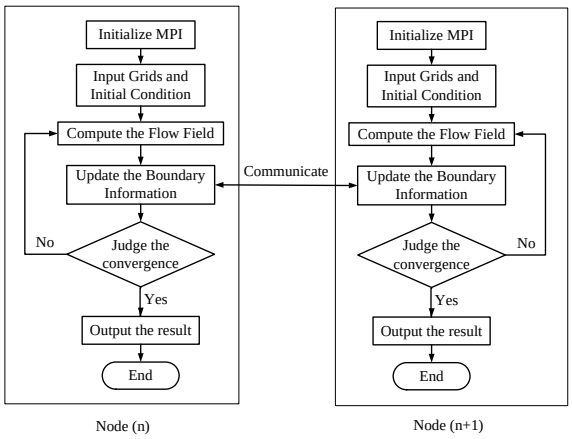


Fig. 4 Flow chart schematic model of the parallel program

Here, the parallel code is programmed using the MPI library functions. Fig. 4 shows that the key modification is adding a module to realize the communication between processors, thus the data structure and program modules remain as much as possible.

V. Result

Considering the high heat transfer, most of the high-speed flight vehicle configurations should employ a blunt leading edge, thus the hypersonic viscous flow field around a blunt cylinder is analyzed. The cylinder radius is 0.038m, and the flow parameters are as follows:

$M_\infty = 16.34, \quad p_\infty = 82.95\text{N/m}^2, \quad T_\infty = 52\text{K}, \quad T_w = 294.4\text{K}$

These represent a low enthalpy flow for which previous computations validate the perfect gas approximation.

Fig. 5 is a 301×101×11 grid, which has the cell Reynolds number at the stagnation point is 3.9.

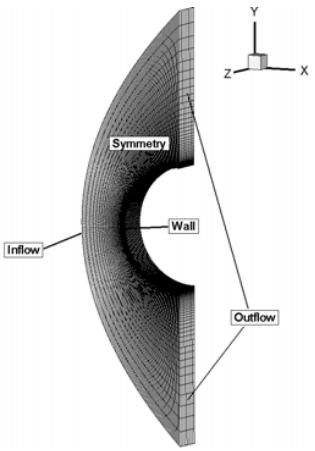
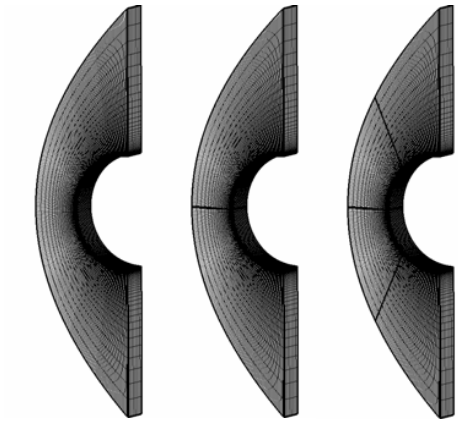


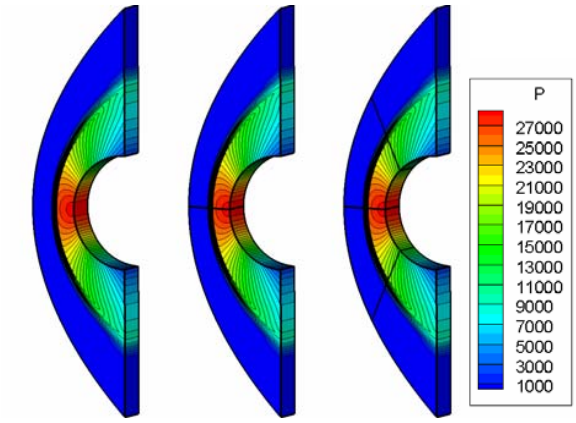
Fig. 5 Computational grid and boundary conditions

Boundary Conditions of this case are set as that in Fig. 5. Inflow boundary conditions are fixed with freestream values, and outflow boundary conditions are obtained using the 1st-order extrapolations. Wall boundary conditions are set as non-slip, isothermal and zero pressure gradient normal to the surface. Symmetric boundary conditions are used for the 3D code to solve the 2D problem.

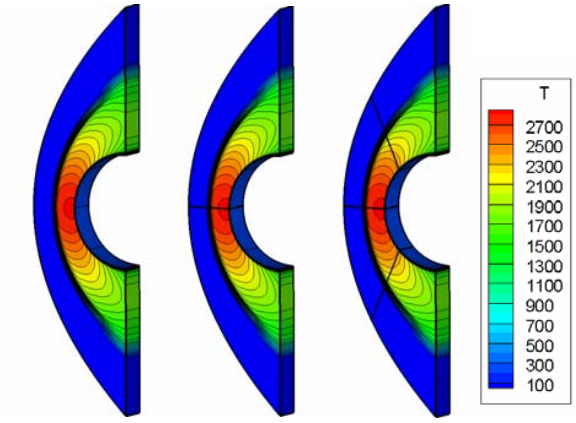
Computations are performed with 2nd order accuracy using MUSCL and van Albada limiter.



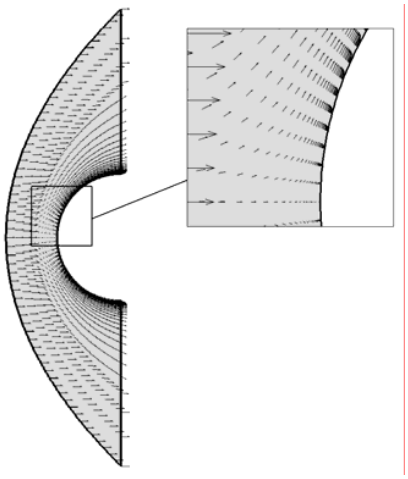
(a) Grids



(b) Pressure contours



(c) Temperature contours



(d) Velocity vectors

Fig. 6 Computational results of flow field in different partitions

The grid is split into small partitions, which are used for parallel computations on a computer with a quad-core processor. The results of flow field in different partitions are shown in Fig. 6, the flow field structures are almost the same, the strong detached bow shock, wide subsonic flow region near the stagnation line and thin boundary layer can be seen very clearly and the contour lines of multi-blocks are continuous and smooth.

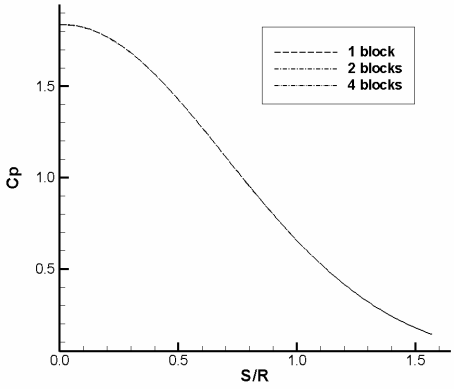


Fig. 7 Wall pressure coefficient distributions

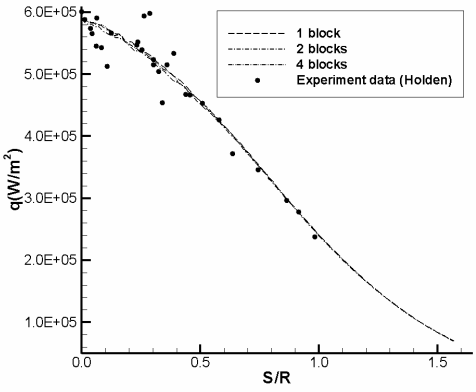


Fig. 8 Wall heat transfer distributions

Computational results are compared with experimental data by Holden et al [12]. As shown in Fig. 7, the wall pressure distributions of different partitions are almost the same and the value at the stagnation point is 1.836. The heat transfer computational results coincide with the experiment data very well in Fig. 8, and the stagnation value is $5.88\times10^5 \mathrm{W/m^2}$. Therefore the solving accuracy for the aerodynamic heating problem is credible. However, some numerical oscillations in temperature appear at the stagnation region near the wall, causing the oscillation of the curves in Fig. 8.

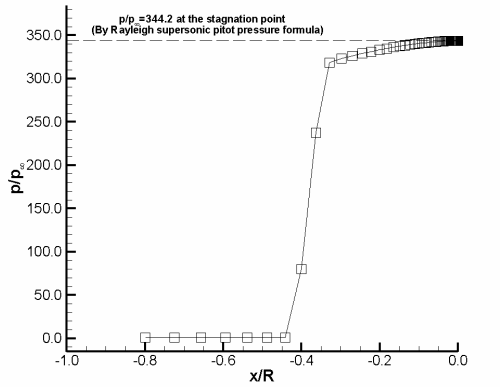


Fig. 9 Pressure distribution along the stagnation line

Fig. 9 shows that the normalized stagnation pressure is close to the value obtained form the Rayleigh supersonic pitot tube formula, which is 344.2. Shock distance calculated is in good agreement with the Billig's correlations for hypersonic shock-wave shapes [13]. The correlated shock distance is $0.393R$.

TABLE I
THE COMPARISON OF PARALLAL EFFICIENCY IN DIFFERENT PARTITIONS

CPU Number	Average wall time interval (sec per 1000 steps)	Speedup ratio	Efficiency
1	1334.4	1.00	100.0%
2	710.2	1.88	93.9%
4	425.7	3.13	78.4%

Table I shows that as the partition number grows, the parallel computational efficiency decrease distinctly due to the small granularity. The grid number is just 0.3 million, which is such a small-scale task for the current advanced computer architecture, especially when it is divided into smaller ones. Meanwhile, the total amount of communication increases, the ratio of computation and communication decrease consequently and finally affect the efficiency.

VI. Conclusion

As shown in the results, the AUSMPW+ scheme is able to predict the flow properties in boundary layer accurately, in good agreement with corresponding experiment results, and does not exhibit oscillation near the shock. It is suitable for the simulation of aerodynamic heating. The parallel computational method based on MPI is feasible and easy to handle, which can be applied to large-scale problem for higher efficiency.

REFERENCES

[1] P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," Journal of Computational Physics, vol. 135, 1997, pp. 250-258.
[2] J. H. Lee, O. H. Rho, "Numerical analysis of hypersonic viscous flow around a blunt body using Roe's FDS and AUSM+ schemes," AIAA paper 97-2054, 1997.
[3] M. S. Liou, C. J. Steffen, "A new flux splitting scheme," Journal of Computational Physics, vol. 107, 1993, pp. 23-39.
[4] M. S. Liou, "Progress towards an improved CFD method: AUSM+," AIAA paper 95-1701, 1995.
[5] M. S. Liou, "A sequel to AUSM: AUSM+", Journal of Computational Physics, vol. 129, 1996, pp. 364-382.
[6] M. S. Liou, "A further development of the AUSM+ scheme towards robust and accurate solutions for all speeds," AIAA paper 2003-4116, 2003.
[7] K. H. Kim, C. Kim, O. H. Rho, "Methods for the accurate computations of hypersonic flows I. AUSMPW+ Scheme," Journal of Computational Physics, vol. 174, 2001, pp. 38-80.
[8] K. H. Kim, C. Kim, "Accurate, efficient and monotonic numerical methods for multi-dimensional compressible flows Part I: Spatial discretization," Journal of Computational Physics, vol. 208, 2005, pp. 527-569.
[9] J. Blazek, "Computational fluid dynamics: principles and applications," Elsevier, 2001, pp. 5-25, 76-77, 93-95, 110-113, 116-119.
[10] G. D. van Albada, B. van Leer, W. W. Roberts, "A Comparative study of computational methods in cosmic gas dynamics," Astronomy and Astrophysics, vol. 108, 1982, pp. 76-84.
[11] S. Yoon, A. Jameson, "Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes equations," AIAA paper 87-0600, 1987.
[12] M. S. Holden, A. R. Moselle, F. R. Riddell, "Studies of aerothermal loads generated in regions of shock/shock interaction in hypersonic flow," AIAA paper 88-0477, 1988.
[13] F. S. Billig, "Shock-wave shapes around spherical- and cylindrical-nosed bodies," Journal of Spacecraft and Rockets, vol.4, no.6, 1967, pp. 822-823.