AIAC-2005-078

COMPUTATION OF TURBULENT FLOWS AROUND ROTATING BODIES USING UNSTRUCTURED GRIDS

PART I: CODE DEVELOPMENT

L. Oktay GÖNÇ [*] and Mehmet Ali AK[¥]
TÜBİTAK-SAGE
Ankara, Turkey

M. Haluk AKSEL[†] and İsmail H. TUNCER[§]
Middle East Technical University
Ankara, Turkey

ABSTRACT

This paper, first in a series of two, presents the development of a 3-D parallel flow solver for turbulent flows around rotating bodies using unstructured grids. Cell centered finite volume solver which employs Roe's upwind flux differencing scheme, Spalart-Allmaras turbulence model and Runge-Kutta explicit multistage time stepping scheme is presented. Arbitrary Lagrangian Eulerian formulation is implemented for moving grids. The computational grid is partitioned by METIS and PVM is used for inter-process communication. The main objective of this study is to be able to solve unsteady turbulent flows around rotating missile configurations and to evaluate the aerodynamic stability derivative coefficients.

INTRODUCTION

The accurate prediction of missile aerodynamics is a major task for the missile industry. Determination of the aerodynamic forces and moments is very important for the prediction of the motion of projectiles. The rates of change of these forces or moments with respect to linear or angular velocity components, namely "stability derivatives", play an important role in the dynamical analysis of the projectiles. So they must be obtained using various techniques.

Stability derivatives can be determined both by experimental and theoretical methods. But some of these aerodynamic forces and moments, such as Magnus effect, can not be estimated with an acceptable accuracy even by experimental methods. Besides, experimental methods are expensive and require a long lead time for the resultant data. Semi-empirical methods can be used for the estimation of such unknowns but these methods are limited to the subsonic flow region. A reasonable prediction of such force and moment coefficients can be performed by computational fluid dynamics methods (CFD).

Solution of flow fields around complex geometries directly addresses the use of unstructured grids. In order to handle the turbulent flows around projectiles at high Reynolds numbers, which have to be predicted well to obtain the stability derivatives accurately, especially at the vicinity of wall boundaries high quality computational mesh is required. Use of such unstructured grids which have large number of computational cells in the computations, requires massive computational resources. In this study rotational effects of missile configurations is also included. It seems that only parallel architecture computers offer the promise of providing orders of magnitude greater computational power for such problems.

The viscous phenomena that occur around missile configurations, such as boundary layer separations, wakes and vortices are quite important for missile aerodynamic characteristics. Also it is well known that the viscous phenomena are of primary importance in the study of unsteady flows. Shearing effects and possible instabilities result in small and large scale turbulent effects. So, prediction of the turbulent effects in the flow solutions is extremely important.

The main objective of this study is, therefore, to develop a 3-D parallel flow solver for turbulent flows

[*] Dr. in Defense Industries Research & Development Institute, Email: lgonc@sage.tubitak.gov.tr
[¥] Dr. in Defense Industries Research & Development Institute, Email: maliak@sage.tubitak.gov.tr
[†] Prof. Dr. in Mechanical Engineering Department,Email: maksel@metu.edu.tr
[§] Prof. Dr., in Aerospace Engineering Department, Email: tuncer@ae.metu.edu.tr

around rotating bodies using unstructured grids, to obtain numerical solutions of Navier-Stokes equation on rotating grids and to be able to evaluate rotational effects [1].

GOVERNING EQUATIONS

For a three-dimensional flow through a finite volume Ω moving with a speed $\vec{V}_{c.v.}$ which is enclosed by the boundary surface S and an exterior normal $\vec{n}$, integral form of the conservation equations in non-inertial frame of reference are given as

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{U}d\Omega+\oint_{S}\vec{F}\cdot d\vec{S}=\oint_{S}\vec{Q}\cdot d\vec{S} \tag{1}$$

where the column vector $\vec{U}$ represents the conservative variables, column vector $\vec{F}$ represents the convective flux vector, and column vector $\vec{Q}$ represents the viscous diffusive flux vector.

$$\vec{U}=\begin{bmatrix}\rho\\ \rho\vec{V}\\ \rho E\end{bmatrix}\qquad \vec{F}=\begin{bmatrix}\rho(\vec{V}-\vec{V}_{c.v.})\\ (\rho\vec{V}\otimes\vec{V}+P\bar{\bar{I}}-\rho\vec{V}\otimes\vec{V}_{c.v.})\\ (\rho E(\vec{V}-\vec{V}_{c.v.})+P\vec{V})\end{bmatrix}$$

$$\vec{Q}=\begin{bmatrix}0\\ \tau_{ij}\\ (\tau_{ij}\cdot\vec{V}-q)\end{bmatrix} \tag{2}$$

where the fluid velocity is : $\vec{V}=u\vec{i}+v\vec{j}+w\vec{k}$

and grid velocity is : $\vec{V}_{c.v.}=u_{c.v.}\vec{i}+v_{c.v.}\vec{j}+w_{c.v.}\vec{k}$

If the convective flux vector; $\vec{F}=F\vec{i}+G\vec{j}+H\vec{k}$ is expressed explicitly:

$$F=\begin{bmatrix}\rho(u-u_{c.v.})\\ \rho u(u-u_{c.v.})+p\\ \rho v(u-u_{c.v.})\\ \rho w(u-u_{c.v.})\\ (\rho E+p)(u-u_{c.v.})+u_{c.v.}p\end{bmatrix}$$

$$G=\begin{bmatrix}\rho(v-v_{c.v.})\\ \rho u(v-v_{c.v.})\\ \rho v(v-v_{c.v.})+p\\ \rho w(v-v_{c.v.})\\ (\rho E+p)(v-v_{c.v.})+v_{c.v.}p\end{bmatrix}$$

$$H=\begin{bmatrix}\rho(w-w_{c.v.})\\ \rho u(w-w_{c.v.})\\ \rho v(w-w_{c.v.})\\ \rho w(w-w_{c.v.})+p\\ (\rho E+p)(w-w_{c.v.})+w_{c.v.}p\end{bmatrix} \tag{3}$$

If non-dimensional viscous diffusive flux vector; $\vec{Q}=\frac{M_\infty}{Re_L}\cdot\left(Q_x\vec{i}+Q_y\vec{j}+Q_z\vec{k}\right)$ is expressed explicitly in the Cartesian coordinate system:

$$Q_x=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ \tau_{xx}u+\tau_{xy}v+\tau_{xz}w-\dot{q}_x\end{bmatrix}$$

$$Q_y=\begin{bmatrix}0\\ \tau_{yx}\\ \tau_{yy}\\ \tau_{yz}\\ \tau_{yx}u+\tau_{yy}v+\tau_{yz}w-\dot{q}_y\end{bmatrix}$$

$$Q_z=\begin{bmatrix}0\\ \tau_{zx}\\ \tau_{zy}\\ \tau_{zz}\\ \tau_{zx}u+\tau_{zy}v+\tau_{zz}w-\dot{q}_z\end{bmatrix} \tag{4}$$

where viscous shear stress tensor components are:

$$\tau_{xx}=2(\mu+\mu_t)\frac{\partial u}{\partial x}-\frac{2}{3}(\mu+\mu_t)(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z})$$

$$\tau_{yy}=2(\mu+\mu_t)\frac{\partial v}{\partial y}-\frac{2}{3}(\mu+\mu_t)(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z})$$

$$\tau_{zz}=2(\mu+\mu_t)\frac{\partial w}{\partial z}-\frac{2}{3}(\mu+\mu_t)(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z})$$

$$\tau_{xy}=\tau_{yx}=(\mu+\mu_t)(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x})$$

$$\tau_{xz}=\tau_{zx}=(\mu+\mu_t)(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x})$$

$$\tau_{yz}=\tau_{zy}=(\mu+\mu_t)(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}) \tag{5}$$

and the heat conduction terms are:

$$\dot{q}_x = -\frac{1}{(\gamma-1)}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial x}$$
$$\dot{q}_y = -\frac{1}{(\gamma-1)}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial y}$$
$$\dot{q}_z = -\frac{1}{(\gamma-1)}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial z} \quad (6)$$

Pressure is given by the equation of state for a perfect gas:

$$P = (\gamma-1)\left[\rho E - \frac{1}{2}\rho\left(u^2+v^2+w^2\right)\right] \quad (7)$$

Pechier et al. [4] state that grid movement only modifies the convection flux at finite volume surface so there is no contribution of grid movement on diffusive flux terms.

The governing flow equations have been nondimensionalized by the free stream density ρ_∞, free stream speed of sound c_∞, free stream temperature T_∞, free stream viscosity μ_∞ and reference length L.

Spalart-Allmaras Turbulence Model

One-equation turbulence model of Spalart-Allmaras is employed in this study. The Spalart-Allmaras [13] turbulence model is a relatively simple one-equation model that solves a modeled transport equation for the kinematic eddy (turbulent) viscosity. It was designed specifically for aerospace applications involving wall-bounded flows and has been shown to give good results for boundary layers subjected to adverse pressure gradients.

The transported variable in the Spalart-Allmaras model, $\tilde{\nu}$, is identical to the turbulent kinematic viscosity except in the near-wall (viscous-affected) region. The transport equation for $\tilde{\nu}$ is

$$\frac{\partial \tilde{\nu}}{\partial t} + \vec{V}\cdot\nabla\tilde{\nu} = C_{b1}(1-f_{t2})\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\left[\nabla\cdot\left\{(\nu+\tilde{\nu})\nabla\tilde{\nu}\right\} + C_{b2}\nabla\tilde{\nu}\nabla\tilde{\nu}\right] - \left(C_{w1}f_w - \frac{C_{b1}}{\kappa^2}f_{t2}\right)\left(\frac{\tilde{\nu}}{d}\right)^2 \quad (8)$$

where the right-hand-side terms represent turbulence eddy viscosity production, diffusion and near-wall turbulence destruction terms, respectively. $\sigma = \frac{2}{3}$, $C_{b1} = 0.1355$, $C_{b2} = 0.622$, $\kappa = 0.4187$ and $C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1+C_{b2}}{\sigma} = 3.2059$ are constants, d is the minimum distance from the wall and ν is the molecular kinematic viscosity.

Turbulent viscosity is defined as:

$$\mu_{turb} = \rho\tilde{\nu}f_{v1} \quad (9)$$

where $f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}$ is the viscous damping function, $\chi \equiv \frac{\tilde{\nu}}{\nu}$ and $C_{v1} = 7.1$ is a constant term.

The vorticity magnitude S which appears in the turbulence production term is modified such that $\tilde{S}$ maintains its log-layer behavior [2]:

$$\tilde{S} = S f_{v3}(\chi) + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}(\chi) \quad (10)$$

where $f_{v2}(\chi) = 1 - \frac{\chi}{1+\chi f_{v1}}$

and $f_{v3} = 1$ (11)

Modified versions of the functions $f_{v2}(\chi)$ and $f_{v3}(\chi)$ are introduced by Spalart in order to eliminate the poor convergence of the residual turbulence especially near reattachment:

$$f_{v2}(\chi) = \left(1+\frac{\chi}{C_{v2}}\right)^{-3}$$

and $f_{v3}(\chi) = \frac{(1+\chi f_{v1})(1-f_{v2}(\chi))}{\chi}$ (12)

Guillen , et al. [2] state that these forms of $f_{v2}(\chi)$ and $f_{v3}(\chi)$ functions result in a modification of the natural laminar-turbulent transition of the Spalart-Allmaras turbulence model.

The function f_{t2} is introduced into the production and destruction terms in order to make $\tilde{\nu} = 0$ a stable solution to the linearized problem [10]. This term does not allow eddy viscosity to increase in regions where it has the value corresponding to half of the laminar viscosity [3].

$$f_{t2} = C_{t3}\exp(-C_{t4}\cdot\chi^2) \quad (13)$$

where $C_{t3} = 1.3$ and $C_{43} = 0.5$ are constants.

The eddy viscosity production is related to the vorticity [3]. S is a scalar measure of the deformation tensor which is based on the magnitude of vorticity which takes the following form in two-dimensional space;

$$S = \left|\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right| \quad (14)$$

Saxena and Nair [3] state that in a boundary layer the blocking effect of the wall is felt at a distance through the pressure term, which acts as the main destruction term for the Reynolds shear stress. In order to obtain a faster decaying behavior of

destruction in the outer region of the boundary layer, a function f_w is used:

$$f_w = g\left\{\frac{1+C_{w3}{}^6}{g^6+C_{w3}{}^6}\right\}^{\frac{1}{6}} \tag{15}$$

where $g = r + C_{w2}\left(r^6 - r\right)$, and $r = \dfrac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}$ is the characteristic length and $C_{w2} = 0.3$, $C_{w3} = 2.0$ are constants.

In order to adapt Equation (8) into the integral form, divergence theorem is applied to conservative and diffusive parts of the equation. Production term, destruction term and a part of the diffusion term that is excluded from the area integral are left as source terms in the integral form of the equation.

$$\frac{\partial}{\partial t}\int_\Omega \tilde{\nu}\, d\Omega + \oint_S \tilde{\nu}\,\vec{V}\cdot\vec{n}\, dS - \oint_S \frac{1}{\sigma}(\nu+\tilde{\nu})\nabla\tilde{\nu}\cdot\vec{n}\, dS = \int_\Omega \left\{\begin{array}{l} C_{b1}(1-f_{t2})\tilde{S}\tilde{\nu} + \dfrac{C_{b2}}{\sigma}\nabla\tilde{\nu}\nabla\tilde{\nu} - \\ \left(C_{w1}f_w - \dfrac{C_{b1}}{\kappa^2}f_{t2}\right)\left(\dfrac{\tilde{\nu}}{d}\right)^2 \end{array}\right\}\cdot d\Omega \tag{16}$$

Non-dimensional form of the equation including the ALE formulation for moving bodies came out to be:

$$\frac{\partial}{\partial t}\int_\Omega \tilde{\nu}\, d\Omega + \oint_S \tilde{\nu}\,(\vec{V}-\vec{V}_{c.v.})\cdot\vec{n}\, dS - \oint_S \frac{M_\infty}{Re_\infty}\frac{1}{\sigma}(\nu+\tilde{\nu})\nabla\tilde{\nu}\cdot\vec{n}dS = \int_\Omega C_{b1}(1-f_{t2})\left\{S f_{v3}(\chi) + \frac{M_\infty}{Re_\infty}\frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}(\chi)\right\}\tilde{\nu}d\Omega + \int_\Omega \frac{M_\infty}{Re_\infty}\left\{\frac{C_{b2}}{\sigma}\nabla\tilde{\nu}\nabla\tilde{\nu} - \left(C_{w1}f_w - \frac{C_{b1}}{\kappa^2}f_{t2}\right)\left(\frac{\tilde{\nu}}{d}\right)^2\right\}\cdot d\Omega \tag{17}$$

Note that as it was for the conservation equations, grid velocity is only introduced in the convective term.

Special attention is given to the f_w term which has a dimensional term r inside. This term is non-dimensionalized as follows:

$$r = \frac{M_\infty}{Re_\infty}\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2} \tag{18}$$

where all terms in this equation are previously non-dimensionalized.

The integral compact form of Equation (17) which is suitable for numerical calculations, is given by;

$$\frac{\partial}{\partial t}\int_\Omega \vec{U}_{turb}\, d\Omega + \oint_S (\vec{F}_{turb,convective} - \vec{F}_{turb,diffusive})\cdot d\vec{S} = \int_\Omega \vec{Q}_{turb}\, d\Omega \tag{19}$$

where the column vector $\vec{U}_{turb}$ represents the passive scalar vector for working variable $\tilde{\nu}$, column vector $\vec{F}_{turb}$ represents the convective and diffusive flux terms, and column vector $\vec{Q}_{turb}$ represents the source term;

$$\vec{U}_{turb} = \left|\tilde{\nu}\right| \tag{20}$$

$$\vec{F}_{turb,convective} = \left[\tilde{\nu}(\vec{V}-\vec{V}_{c.v.})\right] \tag{21}$$

$$\vec{F}_{turb,diffusive} = \left[\frac{M_\infty}{Re_\infty}\frac{1}{\sigma}(\nu+\tilde{\nu})\nabla\tilde{\nu}\right] \tag{22}$$

$$\vec{Q}_{turb} = \left[\begin{array}{l} C_{b1}(1-f_{t2})\left\{S f_{v3}(\chi) + \dfrac{M_\infty}{Re_\infty}\dfrac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}(\chi)\right\}\cdot\tilde{\nu} \\ + \dfrac{M_\infty}{Re_\infty}\left\{\dfrac{C_{b2}}{\sigma}\nabla\tilde{\nu}\nabla\tilde{\nu} - \left(C_{w1}f_w - \dfrac{C_{b1}}{\kappa^2}f_{t2}\right)\cdot\left(\dfrac{\tilde{\nu}}{d}\right)^2\right\} \end{array}\right] \tag{23}$$

NUMERICAL METHOD

A cell centered finite volume discretization is applied to Equation (1) which is in integral form. Time rate of change of conservative variable vector $\vec{U}$ within a computational domain Ω moving with a speed $\vec{V}_{c.v.}$ is balanced by the net convective and diffusive fluxes across the boundary surface S.

In the finite-volume formulation, for a constant control volume of tetrahedron, Equation (1) becomes:

$$\Omega\frac{\partial}{\partial t}\overline{U}_i + \sum_{j=1}^{4}\left[(\vec{F}\cdot\vec{n}S)_j\right]_i - \sum_{j=1}^{4}\left[(\vec{Q}\cdot\vec{n}S)_j\right]_i = 0 \tag{24}$$

where $i = 1, 2, \cdots, \#\,\text{of cells}$

If Equation (24) is written explicitly across the four faces of a tetrahedron;

$$\Omega\frac{\partial}{\partial t}\overline{U}_i + \sum_{j=1}^{4}\left((F-Q_x)n_x + (G-Q_y)n_y + (H-Q_z)n_z\right)\cdot\Delta S_i = 0 \tag{25}$$

Inviscid Fluxes

Inviscid flux quantities $\vec{F}(\vec{U})$ (Equations 24 & 25) are computed using Roe's [18, 19] flux-difference splitting scheme across each cell face of cell centered control volumes.

For a first order scheme, the state of the primitive variables at each cell face is set to cell-centered averages on either side of the face. For the higher

order scheme, the reconstruction scheme suggested by Frink et al. is employed, [5], which is based on an analytical formulation for computing the gradient term of a Taylor series expansion within tetrahedral cells:

$$\tilde{U}_{face(1,2,3)} = \tilde{U}_{center} + \frac{1}{4}\left[\frac{1}{3}\left(\tilde{U}_{node\,1} + \tilde{U}_{node\,2} + \tilde{U}_{node\,3}\right) - \tilde{U}_{node\,4}\right] \quad (26)$$

where $\tilde{U} = [\rho \;\; u \;\; v \;\; w \;\; p]^T$ represents the primitive flow variables for a triangular cell face which is composed of nodes 1, 2 and 3.

Frink [5] states that use of such a reconstruction scheme gives acceptable results around flow discontinuities without requiring the introduction of higher order spatial discretizations with limiters.

Roe Fluxes for Moving Grids

In Arbitrary Lagrangian Eulerian (ALE) formulation the computational grid is allowed to move with a velocity different from the fluid velocity. Figure 1 shows the flow of fluid across the cell boundary between i^{th} and $i+1^{th}$ control volumes. The Eulerian flux vector $F_{eul}(U)$ passing through the cell boundary for one-dimension is given by Roe [18, 19] as:

$$F_{eul}(U) = F_{L,eul}(U) + \sum_j \alpha_j \bar{\Lambda}_{eul} R^{(j)} \quad (27)$$

where; $F_{L,eul} = \begin{bmatrix} \rho_L u_L \\ \rho_L u_L u_L + p \\ (\rho_L E_L + p) u_L \end{bmatrix}$ (28)

and eigenvalues , $\bar{\Lambda}_{eul} = \begin{bmatrix} u - c \\ u \\ u + c \end{bmatrix}$ (29)

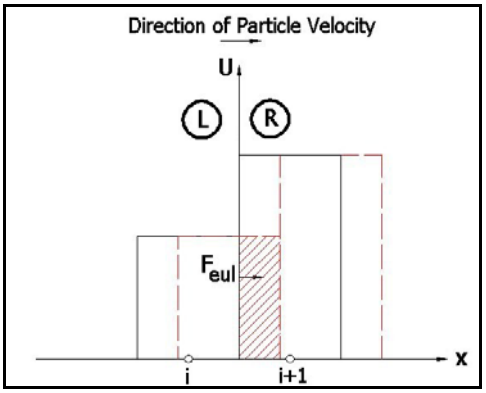

Figure 1. Motion of Fluid across the Cell Boundary

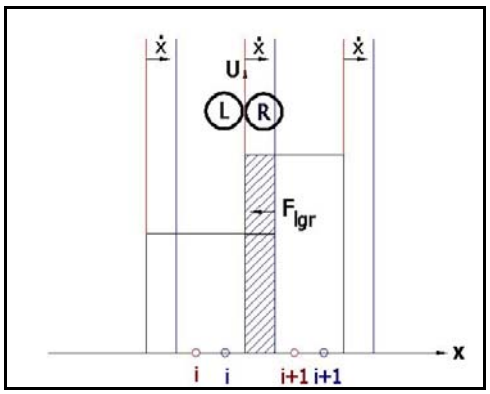

Figure 2. Motion of the Cell Boundary with respect to Stationary Fluid

In Figure 2, the movement of the boundary with the speed $\dot{x}$ to the right is presented where the fluid is thought to be stationary. The Lagrangian flux vector $F_{lgr}(U)$ passing through the cell boundary but in the opposite direction is:

$$F_{lgr}(U) = F_{L,lgr}(U) + F_{lgr}(\Delta U) \quad (30)$$

where; $F_{L,lgr} = \begin{bmatrix} \rho_L \\ \rho_L u_L \\ \rho_L E_L \end{bmatrix} \cdot \dot{x}$; (31)

$$F_{lgr}(\Delta U) = \begin{bmatrix} \Delta\rho \\ \Delta(\rho u) \\ \Delta(\rho E) \end{bmatrix} \cdot \dot{x} \; ; \quad (32)$$

and;

$$\Delta\rho = \rho_R - \rho_L \quad (33)$$

$$\Delta(\rho u) = (\rho u)_R - (\rho u)_L \quad (34)$$

$$\Delta(\rho E) = (\rho E)_R - (\rho E_L) \quad (35)$$

After proper simplifications, the net flux passing through the cell boundary becomes:

$$F_{ale}(U_{rel}) = F_{eul}(U) - F_{lgr}(U) \quad (36)$$

$$F_{ale}(U_{rel}) = F_{L,ale}(U) + \sum_j \alpha_j \bar{\Lambda}_{ale} R^{(j)} \quad (37)$$

where $\bar{\Lambda}_{ale} = \begin{bmatrix} u - c - \dot{x} \\ u - \dot{x} \\ u + c - \dot{x} \end{bmatrix}$; (38)

and $F_{L,ale} = \begin{bmatrix} \rho_L (u_L - \dot{x}) \\ \rho_L u_L (u_L - \dot{x}) + p \\ (\rho_L E_L + p)(u_L - \dot{x}) + \dot{x} p \end{bmatrix}$ (39)

The above numerical discretization of Roe [18, 19] for moving boundaries is valid for three dimensional governing equations. From a physical point of view, the grid motion only affects the convective fluxes. As shown above, to calculate the new convective terms and eigenvalues, the velocity $\vec{V}_{c.v.}$ of the face of a control volume is required. Trepanier et al. [12] state that for deforming meshes, the total volumetric increment is composed of elementary increments along each of its faces. Accordingly, the relevant facial velocity associated with this facial volume increment $\Delta\Omega$, during a time step Δt is defined by

$$\vec{V}_{c.v.} = \frac{\Delta\Omega}{S\Delta t} \tag{40}$$

Trepanier et al. ended up with a finite volume simple explicit approach noting that the grid is deforming.

$$U^{n+1} = \frac{\Omega^n}{\Omega^{n+1}}\left\{U^n + \frac{\Delta t}{\Omega^n}\sum_k F_{ale,k}(U_{rel})S_k\right\} \tag{41}$$

This is also the same as the present case although the grid is not deforming. When volume does not change, above equation reduces to:

$$(U^{n+1} - U^n)\Omega = \Delta t\sum_k F_{ale,k}(U_{rel})S_k \tag{42}$$

In this study, the computational domain is rotated at each time step without deforming the computational cells. In other words, the grid remains undistorted and follows the motion of the body. For non-deforming control volumes, the facial volumetric increments sum up to zero. So, the use of the flux vectors and eigenvalues derived at the beginning of this chapter guarantees the satisfaction of geometric conservation law. Barakos et al. [7] also states that there is no need to apply any geometric conservation law in this formulation.

Viscous Fluxes

The viscous fluxes $\vec{Q}(\vec{U})$, (Equations. 24 & 25), are approximated at the cell face centers by first computing the velocity gradients at cell centroids, then averaging the values with neighbor cell centroids at the shared cell face. The velocity gradients at the cell centroids are calculated using the divergence theorem. This theorem can be considered as defining the average of the gradient of a scalar U as a function of its values at the boundaries of the finite volume under consideration.

$$\int_\Omega \vec{\nabla}\vec{U}\cdot d\Omega = \oint_S \vec{U}\cdot d\vec{S} \tag{43}$$

$$\overline{\left(\frac{\partial U}{\partial x}\right)}_\Omega \equiv \frac{1}{\Omega}\int_\Omega \frac{\partial \vec{U}}{\partial x}\cdot d\Omega = \frac{1}{\Omega}\oint_S \vec{U}\cdot\vec{n}_x\cdot d\vec{S} \tag{44}$$

Turbulence Model

Non-dimensionalized integral form of this transport Equation (17) has been solved separately from the conservation flow equations. For the spatial discretization of the conservative part of the equation, Equation (21), HLLC approximate Riemann solver has been used by implementing the turbulent working variable $\tilde{\nu}$ as a passive scalar in the formulation, [6]. Face averaged values of the conservative flow variables are calculated using the same formulation used for Roe's upwind scheme. Face averaged value for the passive scalar turbulent working variable $\tilde{\nu}$ at left and right states are calculated by :

$$U_{turb,face,L/R} = \rho_{L/R}\left(\frac{S_{L/R} - u_{L/R}}{S_{L/R} - S_{face}}\right)\tilde{\nu}_{L/R} \tag{45}$$

The HLLC flux term at the face of the cell interface is defined as:

$$F_{turb,conv,face} = \begin{cases} F_{turb,L} & \text{if} \quad 0 \le S_L \\ F_{turb,L} + S_L\left(U_{turb,face,L} - \tilde{\nu}_L\right) & \text{if} \quad S_L \le 0 \le S_{face} \\ F_{turb,R} + S_R\left(U_{turb,face,R} - \tilde{\nu}_R\right) & \text{if} \quad S_{face} \le 0 \le S_R \\ F_{turb,R} & \text{if} \quad 0 \ge S_R \end{cases} \tag{46}$$

where S is the wave speed.

The diffusive part of the equation has been modelled by the same methodology used for the computation of viscous fluxes. A third order Runge-Kutta explicit scheme has been adapted for the temporal discretization.

Temporal Discretization

Equation (25) can be written as:

$$\Omega\frac{\partial}{\partial t}\overline{U}_i + R_i = 0 \tag{47}$$

for $i = 1, 2, \cdots, \#\,\text{of cells}$, where

$$R_i = \sum_{j=1}^{4}\left((F - Q_x)\,n_x + (G - Q_y)\,n_y + (H - Q_z)\,n_z\right)\cdot\Delta S_i \tag{48}$$

R_i is the residual which is the summation of the fluxes through the four faces of the tetrahedral computational cell. These set of equations are integrated in time using a fully explicit third order Runge-Kutta scheme developed by Jameson [20]. This scheme is used for both steady state and unsteady calculations. For unsteady calculations, time step is chosen to be global constant value which is the minimum of all computational cells. On the other hand for steady state calculations, spatially variable time stepping is used for fast steady state convergence.

Grid Rotation

In a steady coning motion of a missile, the longitudinal axis of the missile performs a rotation at a constant angular velocity about a line parallel to the free stream velocity vector and coincident with the projectile center of gravity, while oriented at a constant angle with respect to the free stream velocity vector [8]. In particular the projectile may rotate about its longitudinal axis also, Figure 3. So in order to simulate the motion of a missile realistically both rotations must be taken into account. Coning and spinning rates are given by aeroballistics analysis. With respect to the fixed coordinate system, the vertical and horizontal components of angle of attack α and β, vary in a periodic fashion as the projectile rotates about the free stream velocity vector, Figure 3. However, the total angle of attack, $\alpha_t \approx \sqrt{\alpha^2 + \beta^2}$ is constant [8]. The rotational velocity of the projectile around its axis is denoted by ω, where the rotational velocity of the projectile around the trajectory of the flight is denoted by Ω, Figure 3.

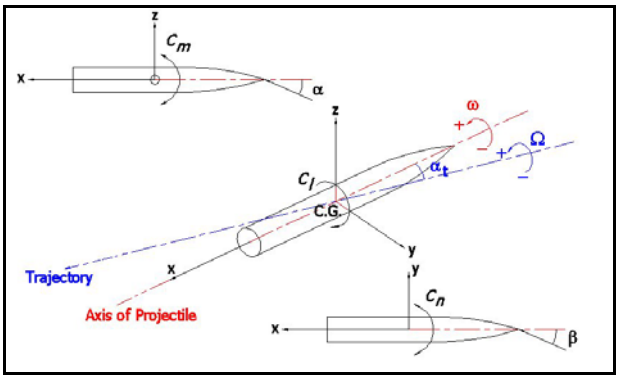

Figure 3. Schematic Figure of Rotational Motion of a Missile

The rotation of the computational grid around the missile about the axis of projectile and about the free stream velocity vector can be performed by multiplication of simple transformation matrices [16]:

$$\left[X^{n+1}\right] = \left[X^{n}\right]\cdot\left[R_{z\beta}\right]\cdot\left[R_{y\alpha}\right]\cdot \left\{\left[R_{y\alpha}\right]^{-1}\cdot\left[R_{z\beta}\right]^{-1}\right\}\cdot\left[R_{x\theta}\right]\cdot\left\{\left[R_{z\beta}\right]\cdot\left[R_{y\alpha}\right]\right\}\cdot\left[R_{x\phi}\right] \quad (49)$$

where X^n is the grid coordinates at time step n which is rotated to a new position given by X^{n+1}, and

$$\left[R_{z\beta}\right] = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (50)$$

$$\left[R_{y\alpha}\right] = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad (51)$$

$$\left[R_{x\theta}\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \text{ where } \theta = \omega\cdot\Delta t \quad (52)$$

$$\left[R_{x\phi}\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \text{ where } \phi = \Omega\cdot\Delta t \quad (53)$$

Boundary Conditions

The boundary conditions are implemented by using ghost cells at the boundaries. For flows around moving bodies, the entire computational grid is rotated instead of deforming the mesh. As the far field boundary condition for external flows, characteristic Riemann invariants corresponding to the incoming and outgoing waves traveling in characteristic directions, which are defined as normal to the boundary, are used [15, 11]. The density is computed from the entropy relation, and the pressure from the perfect gas law using the square of the speed of sound. Batina [14] states that this approach correctly accounts for wave propagation at farfield which is important for convergence rate and serves as a non-reflecting boundary condition for unsteady applications.

The viscous wall boundary condition imposes a no-slip condition of the flow, a zero pressure gradient, and the appropriate heat transfer condition (adiabatic or constant temperature) at the zone boundary (wall surface). The no-slip condition can involve a non-zero velocity if the wall is moving. Hirsch [17] and Toro [6] proposes the following approach for evaluating normal velocity component at solid wall boundary moving with a speed $u_{C.V}$:

$$\begin{aligned} u_{\text{ghost cell}} &= -\left(u_i - 2\cdot u_{C.V}\right) \\ \rho_{\text{ghost cell}} &= \rho_i \\ p_{\text{ghost cell}} &= p_i \end{aligned} \quad (54)$$

Symmetry boundary condition is the same as the condition defined for non-penetrating inviscid wall boundary.

For Spalart-Allmaras turbulence model, on the no-slip surfaces, the working variable $\tilde{\nu}$ is set to zero. For tangent-flow surfaces, in other words for the symmetry boundary condition, zero gradient of the working variable is applied. For far field boundary, it is checked whether it is inflow or outflow first and

then working variable is set to $\tilde{\nu} = 1.0$ for the inflow boundaries which corresponds to a free stream turbulent kinematic viscosity of $\nu_t = 0.02786$. For outflow boundaries the value of $\tilde{\nu}$ is extrapolated from the interior mesh. Initial value of $\tilde{\nu}$ has been taken as the same with the free stream value.

Parallel Processing

Parallel processing is based on domain decomposition. The domain decomposition is performed by "METIS" [9] which is an open source software. Communication between parallel processes is achieved by "PVM – Parallel Virtual Machine" message passing software libraries. The parallel computing environment consists of a LINUX based PC cluster.

The parallel solver is composed of two separate executables; "Master" and "Worker". The "Master" process performs followings in sequence:

1. The input data and computational grid are read.
2. Neighbor connectivity is evaluated.
3. Computational grid is partitioned by METIS.
4. "Worker" processes are spawned for each partition and partition data are sent.
5. Residuals and flow variables from each partition are received at prescribed time steps and I/O is done.
6. Program is stopped when prescribed convergence criteria is satisfied or prescribed number of time steps is reached.

Whereas, "Worker" processes perform the followings in sequence:

1. The input data are received from the "Master" process.
2. The geometric properties of cells, including the grid movement and grid velocities are computed.
3. The interface boundary conditions to/from the adjacent partitions are exchanged.
4. The flow solution is implemented for the partition.
5. The residuals and aerodynamic load coefficients are sent to the "Master" process periodically.

CONCLUSION

In this paper, first in a series of two, the parallel Navier-Stokes solver with Spalart-Allmaras turbulence model on unstructured rotating grids is presented in detail. Code verification and validation studies will be presented in the second part of the series.

ACKNOWLEDGMENTS

Authors acknowledge the funding that was received from Turkish Scientific and Technological Research Council (TÜBİTAK) for this project.

REFERENCES

1. Gönç L. Oktay., *"Computation of External Flow Around Rotating Bodies,"* Ph.D. Thesis, Middle East Technical University, 2005.
2. Deck, S., Duveau, P., D'Espiney, P., and Guillen, P. *"Development and Application of Spalart-Allmaras One Equation Turbulence Model to Three-Dimensional Supersonic Complex Configurations, "* Aerospace Science and Technology, Vol.6, pp. 171-183, 2002.
3. Saxena, S.K., and Nair, M.T., *"Implementation and Testing of Spalart-Allmaras Model in a Multi-Block Code, "* AIAA-2002-0835, January, 2002.
4. M. Pechier, P. Guillen, and R. Cayzac, *"Magnus Effect over Finned Projectiles, "* Journal of Spacecraft and Rockets, Vol.38, pp. 542-549, July-August, 2001.
5. Frink, N. T., and Pirzadeh, S. Z., *"Tetrahedral Finite-Volume Solutions to the Navier-Stokes Equations on Complex Configurations, "* International Journal for Numerical Methos in Fluids, Vol.31, pp. 175-187, 1999.
6. Toro, E. F., *"Riemann Solvers and Numerical Methods for Fluid Dynamics, "* 2nd Edition, Springer, 1999.
7. Barakos, G., and Drikakis, D., *"An Implicit Unfactored Method for Unsteady Turbulent Compressible Flows with Moving Boundaries, "* Computers & Fluids, Vol.28, pp. 899-922, 1999.
8. Weinacht, P., Sturek, W. B., and Schiff, L. B., *"Navier-Stokes Predictions of Pitch Damping for Axisymmetric Projectiles, "* Journal of Spacecraft and Rockets, Vol.34, No.6, pp. 753-761, November-December, 1997.
9. Karypis, G. and Kumar, V., *"METIS - A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 3.0., "* Manual, University of Minnesota and Army HPC Research Center, 1997
10. Ashford, G.A., and Powell, K.G., *"An Unstructured Grid Generation and Adaptive Solution Technique for High-Reynolds-Number Compressible Flows, "* von Karman Institute for Fluid Dynamics, Lecture Series 1996-06, March, 1996.
11. Frink, N.T., *"Recent Progress Toward a Three-Dimensional Unstructured Navier-Stokes Flow Solver,"* AIAA 94-0061, 1994.

12. Trepanier, J. Y., Reggio, M., Paraschivoiu, M., and Camarero R., *"Unsteady Euler Solutions for Arbitrarily Moving Bodies and Boundaries, "* AIAA Journal, Vol.31, No.10, pp. 1869-1876, October, 1993.

13. Spalart, P. and Allmaras, S. *"A One-Equation Turbulence Model for Aerodynamic Flows, "* Technical Report AIAA-92-0439, American Institute of Aeronautics and Astronautics, 1992.

14. Batina, J. T., *"Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis, "* AIAA Journal, Vol.29, No.3, pp. 327-333, March, 1991.

15. Frink, N.T., Parikh, P., and Pirzadeh, S., *"A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes,"* AIAA 91-0102, 1991.

16. Rogers, D. F., and Adams, J. A., *"Mathematical Elements for Computer Graphics, "* 2nd Edition, McGraw-Hill, 1990.

17. Hirsch, C., *"Numerical Computation of Internal and External Flows, Vol. 1&2, "* John Wiley & Sons, 1989.

18. Roe, P. L., *"Discrete Models for the Numerical Analysis of Time-Dependent Multidimensional Gas Dynamics, "* Journal of Computational Physics, Vol.63, pp. 458-476, 1986.

19. Roe, P. L., *"Approximate Riemann Solvers, Parameter Vectors and Difference Schemes, "* Journal of Computational Physics, Vol.43, pp. 357-372, 1981.

20. Jameson, A., Schmidt, W., and Turkel, E., *"Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes, "* AIAA 81-1259, 1981.