Theory Guide

Isaiah M. Helt*
Northrop Grumman, 1575 S Price Rd, Chandler, AZ 85286

These instructions give you guidelines for preparing papers for AIAA Technical Papers using LaTeX. Define all symbols used in the abstract. Do not cite references in the abstract. The footnote on the first page should list the Job Title and AIAA Member Grade for each author, if known Authors do not have to be AIAA members.

I. Nomenclature

u_i = i^{th}-velocity component
μ = dynamic viscosity
ν = kinematic viscosity
ν_T = kinematic Eddy viscosity

*Aerodynamicist, Launch Vehicles Division, ihelt3@gatech.edu

II. Introduction

III. Mesh

IV. Boundary Conditions

A. Viscous Wall

B. Freestream

C. Inlet

D. Outlet

V. Turbulence

This section will detail turbulence theory and modeling in the LUNA code framework.

A. Reynolds-Averaged Navier Stokes (RANS) Models

Reynolds-Averaged Navier Stokes models are based on either solving for the mean flow using the eddy viscosity hypothesis, or close the Reynolds stress tensors, $u_i u_j$. At the moment, LUNA only implements eddy viscosity models.

The turbulent viscosity hypothesis, or eddy viscosity hypothesis, assumes that the turbulent effects can be modeled via a dissipative eddy viscosity term, denoted $\nu_T = \mu_T/\rho$ in the Navier Stokes equations. Therefore, the key to solving the closure problem in eddy viscosity models is determining the eddy viscosity. Note that ν_T is a property of the flow, not the fluid (like the standard viscosity ν).

$$\langle u_i u_j \rangle = \frac{2}{3} k \delta_{ij} - \nu_T \left[\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i} \right] \tag{1}$$

The eddy viscosity model makes two major assumptions:

1) The Deviatoric/anisotropic part of the Reynolds stress tensor is treated as a function of local mean strain rates only (i.e. in eq. 1, LHS = RHS for all i, j in isotropic turbulence)
2) The relationship between the Reynolds stress and the mean strain rates are linearly related by a single scalar

1. Spalart-Allmaras

Spalart-Allmaras is a low-Reynolds number one equation RANS turbulence model. In other words, one transport equation is required to be solved to determine the eddy viscosity. For a general compressible flow, the eddy viscosity is given as

$$\mu_T = \rho f_{v1} \tilde{\nu} \tag{2}$$

where

- ρ is the density
- f_{v1} is a damping function
- $\tilde{\nu}$ is a modified diffusivity

The damping function is:

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{3}$$

where:

$$\chi = \frac{\tilde{\nu}}{\nu} \tag{4}$$

The transport equation for the modified diffusivity $\tilde{\nu}$ is:

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \frac{\partial \rho \tilde{\nu} u_j}{\partial x_j} = \frac{1}{\sigma_{\tilde{\nu}}} \frac{\partial}{\partial x_j} \left[(\mu + \rho \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right] + P_{\tilde{\nu}} + S_{\tilde{\nu}} \tag{5}$$

where:

- u_i is the mean velocity
- $\sigma_{\tilde{\nu}}$ is a model coefficient
- μ is the dynamic viscosity
- $P_{\tilde{\nu}}$ is the production term
- $S_{\tilde{\nu}}$ is source term

Note that using the continuity equation, eq. 5 can be simplified to

$$\rho \left[\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} \right] = \frac{1}{\sigma_{\tilde{\nu}}} \frac{\partial}{\partial x_j} \left[(\mu + \rho \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right] + P_{\tilde{\nu}} + S_{\tilde{\nu}} \tag{6}$$

VI. Solver Algorithms

A. SIMPLE: Semi-Implicit Method for Pressure Linked Equations

1. Theory

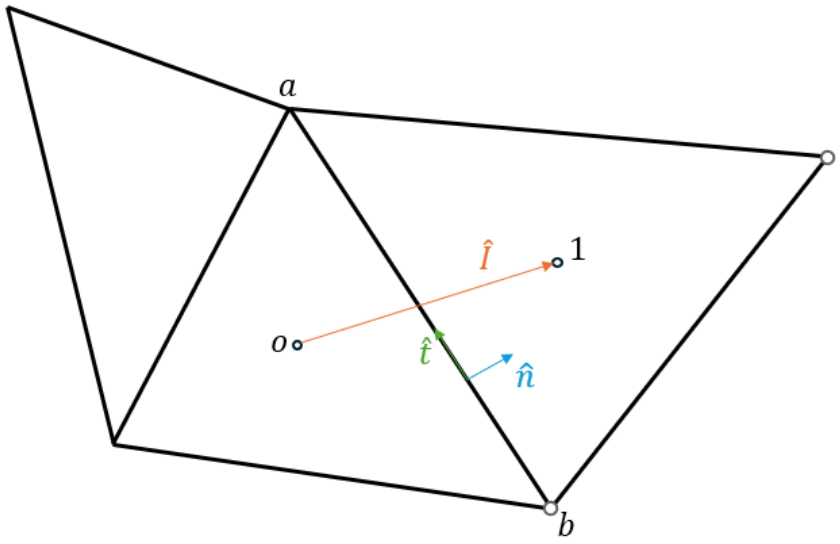

Fig. 1 SIMPLE Algorithm Mesh Relation Terminology

2. Application

Some Common Expressions

$$\left.\frac{\partial p}{\partial x_i}\right|_f n_{i,f} = \frac{p_{\mathcal{N}(f)} - p_o}{\delta_f} + \text{tangential component} \tag{7}$$

$$\left.\frac{\partial p}{\partial x_i}\right|_o = \frac{1}{V_o}\sum_f p_f A_f n_{i,f} \tag{8}$$

$$p_f = w_o p_o + (1 - w_o) p_1 \tag{9}$$

Momentum Equation The linear system to be solved for the momentum equation is given in eq. 10

$$A_o u_o + \sum_f A_{\mathcal{N}(f)} u_{\mathcal{N}(f)} = Q_o \tag{10}$$

where the diagonal elements are given by A_o

$$A_o = \sum_f \left(\frac{|\dot{m}_f| + \dot{m}_f}{2} + \frac{\mu_f A_f}{\delta_f} \right) \tag{11}$$

and the off-diagonal terms are a result of neighboring cells and are defined by the equation below

$$A_{\mathcal{N}(f)} = -\frac{|\dot{m}_f| - \dot{m}_f}{2} - \frac{\mu_f A_f}{\delta_f} \tag{12}$$

The right hand side of the momentum equation is a combination of source terms due to pressure (S_o), skew terms due to the unstructured nature of the solver ($S_{\text{skew},o}$), and contributions from boundary faces (S_{bc}).

$$Q_o = S_o + S_{\text{skew},o} + S_{bc} \tag{13}$$

The skew term is given below. Note that if a pure Cartesian grid is provided to the solver, $\hat{t}_f \cdot \vec{I}_f = 0$, so the skew term goes away.

$$S_{\text{skew},o} = -\sum_f \left(\left[\frac{u^*_{a(f)} - u^*_{b(f)}}{\delta_f} \right] \hat{t}_f \cdot \vec{I}_f \right) \mu_f \tag{14}$$

In the $S_{\text{skew},o}$ equation above, the u^* terms represent the nodal velocities, which are calculated explicitly using a distance weighted scheme on the cell face velocities:

$$u^* = w_{i,f} u_{i,o} \tag{15}$$

Equation 16 shows the contribution due to the pressure source.

$$S_o = -\sum_f p_f n_{i,f} A_f \tag{16}$$

The contribution from the boundary faces must also be accounted for in the source terms. Effectively, the boundary face source is due a result of a neighbor cell contribution ($A_{N(f)}$) being moved to the right hand side of the linear system:

$$S_{bc} = u_{bc} \left(\frac{|\dot{m}_f| - \dot{m}_f}{2} + \frac{\mu_f A_f}{\delta_f} \right) \tag{17}$$

Pressure Correction Equation The pressure correction is calculated by correcting any mass imbalance into the cells, as given by the equation below.

$$\dot{m}_{imb} = \sum_f \rho_f \underbrace{\left[w_f \frac{V_o}{A_o|_o} + (1 - w_f) \frac{V_1}{A_o|_1} \right] \left(\left. \frac{\partial p'}{\partial x_i} \right|_f n_{i,f} \right)}_{u'_{i,f} n'_{i,f}} A_f \tag{18}$$

where $u'_{i,f}$ is the velocity correction and we recall that the pressure gradient can be written as a difference in the cell pressure and neighboring cell pressures:

$$\left. \frac{\partial p}{\partial x_i} \right|_f n_{i,f} = \frac{p_{N(f)} - p_o}{\delta_f} + \text{tangential component} \tag{19}$$

The mass imbalance $\dot{m}_{imb}$, is determined by adding all the mass flux sources coming into the cell:

$$\dot{m}_{imb} = \sum_f \dot{m}_f \tag{20}$$

Therefore, by plugging equations 19 and 20 into eq 18, we can write the mass imbalance equation in terms of cell pressures, their neighbors, and the mass imbalance

$$A^p_o p_o + \sum_f A^p_{N(f)} p_{N(f)} = -\dot{m}_{imb} \tag{21}$$

where the coefficients are given as

$$A^p_o = \sum_{f(o)} \left[w_f \frac{V_o}{A_o|_o} + (1 - w_f) \frac{V_1}{A_o|_1} \right] \left(\frac{\rho_f A_f}{\delta_f} \right) \tag{22}$$

$$A^p_{N(f)} = -\left[w_f \frac{V_o}{A_o|_o} + (1 - w_f) \frac{V_1}{A_o|_1} \right] \frac{\rho_f A_f}{\delta_f} \tag{23}$$

Note that when solving the pressure imbalance equation, the mass imbalance should ultimately reduce to 0, as we are solving for an incompressible flow. Additionally, the pressure correction should ultimately reduce to 0 upon convergence, so it is important to use an initial guess of 0 when solving the pressure correction equation to help with convergence. If a non-zero initial guess is used, the final iteration may have trouble converging.

After the pressure has been corrected, a relaxation factor, α^p_{relax}, is applied to help the solver stop from diverging.

$$p' = \alpha^{p}_{\text{relax}} p' \tag{24}$$

The lower the value of $\alpha^{p}_{\text{relax}}$, the more stable the solution will be. However, the solution will also converge slower with lower α_{relax}.

Velocity and Mass Corrections Corrections for the cell center velocities:

$$u'_o = -\frac{1}{A_o|_o} \left.\frac{\partial p'}{\partial x_i}\right|_o V_o \tag{25}$$

$$= -\frac{1}{A_o|_o} \sum_f p'_f A_f n_{i,f} \tag{26}$$

Corrections for the cell face velocities:

$$u'_f = -\left[w_f \frac{V_o}{A_o|_o} + (1 - w_f) \frac{V_1}{A_o|_1} \right] \left.\frac{\partial p'}{\partial x_i}\right|_f \tag{27}$$

Correction to the mass face flux

$$\dot{m}'_f = \rho_f A_f \left(u'_i n_{i,f} \right) \tag{28}$$

$$= -\rho_f A_f \left[w_f \frac{V_o}{A_o|_o} + (1 - w_f) \frac{V_1}{A_o|_1} \right] \left(\frac{\partial p'}{\partial x_i} n_{i,f} \right) \tag{29}$$

$$= -\rho_f A_f \left[w_f \frac{V_o}{A_o|_o} + (1 - w_f) \frac{V_1}{A_o|_1} \right] \left(\frac{p'_{\mathcal{N}(f)} - p'_o}{\delta_f} + S_{\text{tangent}} \right) \tag{30}$$

Note that the source term due to the tangential component component of the mass flux (S_{tangent}) is generally neglected as the mass imbalance will ultimately go to zero. Therefore the final solution will not be impacted by the tangential terms, though the convergence may be negatively affected.

After all of the correction terms have been collected, they can be added to the original values (after being relaxed) in order to get the cell value for the next iteration:

$$p = p + p' \tag{31}$$

$$u = u + \alpha^{u}_{\text{relax}} u' \tag{32}$$

$$\dot{m}_f = \dot{m}_f + \alpha^{u}_{\text{relax}} \dot{m}'_f \tag{33}$$

Note that because the velocity and mass corrections are linearly related, the same relaxation factor is applied to both systems.

VII. Math

A. Linear Algebra

1. Sparse Matrices

In an implicit CFD solver, a linear system is formed and solved for all transport equations over every cell. This means a matrix must be created for each each of these systems, which has a size of $n \times n$, where n is the number of cells in the mesh. Obviously, this matrix could quickly become very large and take up a huge amount of memory. However, in these matrices, there are a large number of 0 elements. This fact allows us to cleverly store matrix data to save on memory. The LUNA solver uses a Compressed Sparse Row (CSR) matrix format to save on memory, which is detailed here.

The CSR matrix stores data in 3 different vectors, which are detailed below. NNZ refers to the number of nonzero elements in a general $m \times n$ matrix, where m is the number of rows, and n is the number of columns.

- $\vec{v}$: A vector which stores all the non-zero vectors in the matrix (size: NNZ)
- $\vec{c}$: A vector which stores the column associated each value in the $\vec{v}$ vector (size: NNZ)
- $\vec{r}$: A vector which always starts with 0, and stores the difference in NNZ elements before and after the row (size: $m + 1$)

An example is given below for clarity:

$$A = \begin{bmatrix} 0 & 0 & 4 & 0 \\ 1 & 0 & 0 & 2 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \end{bmatrix} \tag{34}$$

$$\vec{v} = \begin{bmatrix} 4 & 1 & 2 & 8 & 7 \end{bmatrix} \tag{35}$$

$$\vec{c} = \begin{bmatrix} 3 & 1 & 4 & 1 & 2 \end{bmatrix} \tag{36}$$

$$\vec{r} = \begin{bmatrix} 0 & 1 & 3 & 4 & 4 & 5 \end{bmatrix} \tag{37}$$

The $\vec{v}$ vector is relatively self explanatory in the above example. The $\vec{c}$ vector starts with 3, indicating the that the value of 4 is in the 3$^{\text{rd}}$ column, followed by 1 indicating that the value of 1 is in the 1$^{\text{st}}$ column, and so on.

The $\vec{r}$ vector is the least obvious. It starts with 0, as it always does, and then 1 indicates there was 1 NNZ in the first row. Since there were 2 NNZ in the second row, the third value is $1 + 2 = 3$, where the 1 is the NNZ in the previous rows. The fourth value is 4 since there was 1 NNZ in the third row ($1 + 3 = 4$), and in the fourth row there are no NNZ, so the vector stays at 4, finally in the last row there is one last NNZ, so the final value is $4 + 1 = 5$.

While this method does not appear to save much memory for small matrices, it can have a huge impact on the amount of storage used for bigger matrices.

VIII. Conclusion

Appendix

An Appendix, if needed, should appear before the acknowledgments.

Acknowledgments