General Equation Set Solver for Compressible and Incompressible Turbomachinery Flows

Extended abstract submitted for 2003 Joint Propulsion Conference

Douglas L. Sondak
Boston University

Daniel J. Dorney
NASA Marshall Space Flight Center

Abstract

Turbomachines for propulsion applications operate with many different working fluids and flow conditions. The flow may be incompressible, such as in the liquid hydrogen pump in a rocket engine, or supersonic, such as in the turbine which may drive the hydrogen pump. Separate codes have traditionally been used for incompressible and compressible flow solvers. The General Equation Set (GES) method can be used to solve both incompressible and compressible flows, and it is not restricted to perfect gases, as are many compressible-flow turbomachinery solvers. An unsteady GES turbomachinery flow solver has been developed and applied to both air and water flows through turbines. It has been shown to be an excellent alternative to maintaining two separate codes.

Nomenclature

Greek Symbols:
- $\xi, \eta, \zeta$: transformed coordinate directions
- $\delta$: Kroneker delta
- $\mu$: viscosity
- $\mu_t$: turbulent viscosity
- $\tau$: transformed time or shear stress (depending on context)
- $\tau^*$: transformed pseudo-time

Roman Symbols:
- $e$: Total internal energy per unit mass
- $E, F, G$: Flux vectors
- $J$: Jacobian of coordinate transformation
- $k$: Heat conduction coefficient
- $P$: Pressure
- $Q$: Conserved variables
- $Q_p$: Primitive variables
- $Re$: Reynolds number
- $T$: temperature
- $u, v, w$: Physical velocity components
- $U, V, W$: Contravariant velocity components

Superscripts:
- $k$: Pseudo-time step index
- $n$: Physical time step index
- $\nu$: viscous

Subscripts:
- $t$: turbulent
- $x, y, z$: Partial differentiation in physical coordinates
- $\xi, \eta, \zeta$: Partial differentiation in transformed coordinates

Other symbols:
- $\hat{\cdot}$: Transformed coordinates
Introduction

Turbomachines for propulsion applications operate on many different fluids and under a wide range of flow conditions. The working fluid may be air, liquid or gaseous hydrogen, or liquid or gaseous oxygen. The flow may be incompressible, such as in the fuel pump in a liquid-fuel rocket engine, or supersonic, such as in the turbine which may drive the fuel pump.

Both compressible and incompressible flows are governed by the Navier-Stokes equations. However, in a nearly-incompressible flow there is a great disparity in wave speeds, since the speed of sound approaches infinity for a truly incompressible fluid. A compressible flow solver will encounter numerical stiffness if applied to a nearly-incompressible flow, and the algorithm will fail. Because of this, it is common practice to use one algorithm for incompressible flows and a different algorithm for compressible flows.

The General Equation Set (GES) method has been developed to handle both compressible and incompressible flows. It can be used to solve the full, unsteady, 3-D Navier-Stokes equations, and with the introduction of a single input flag can reduce to the pseudo-compressibility method commonly used to solve incompressible flows. When used in conjunction with a dual time step, it can be used for time accurate simulations.

When the Navier-Stokes equations are cast in their general form, without applying the perfect gas relation, they can be solved for any working fluid for which properties are available. The GES technique in conjunction with the equations cast in this form results in a general flow solver applicable to most conditions encountered in propulsion turbomachinery applications.

A code has been written to solve 3-D unsteady turbomachinery flows using the GES method. Test cases have been run on turbines operating in low-speed air (inlet Mach number of 0.07), supersonic-inlet air, and water. Results have been compared with test data where available, and with the results of a well-developed ideal-gas, compressible turbomachinery flow solver. [Abstract note: The cases run so far are 2-D. The final paper will contain 3-D examples.]

Equations

The 3-D, unsteady, Navier-Stokes equations may be expressed in generalized, curvilinear coordinates in dual-time-step form as

\[
\dot{\mathbf{Q}}_t + \dot{\mathbf{Q}}_\tau + \dot{\mathbf{E}}_\tau + \dot{\mathbf{F}}_\eta + \dot{\mathbf{G}}_\zeta - \text{Re}^{-1}\left(\dot{\mathbf{E}}_\tau + \dot{\mathbf{F}}_\eta + \dot{\mathbf{G}}_\zeta\right) = 0
\]

(1)

where

\[\dot{\mathbf{Q}} = \begin{bmatrix} \rho & \rho u & \rho v & \rho w & E \end{bmatrix}^T,\]

the standard set of conserved variables. The fluxes also have their standard definitions. The variable \(\tau\) represents the transformed physical time step, and \(\tau^*\) represents the transformed pseudo-time step for subiterations.

Defining the vector of primitive variables as

\[\dot{\mathbf{Q}}_p = \begin{bmatrix} \rho & u & v & w & T \end{bmatrix}^T,\]

the Navier-Stokes equations can be rewritten as

\[
\Gamma_e \left(\dot{\mathbf{Q}}_p \right)_\tau + \Gamma_p \left(\dot{\mathbf{Q}}_p \right)_\tau + \dot{\mathbf{E}}_\tau + \dot{\mathbf{F}}_\eta + \dot{\mathbf{G}}_\zeta - \text{Re}^{-1}\left(\dot{\mathbf{E}}_\tau + \dot{\mathbf{F}}_\eta + \dot{\mathbf{G}}_\zeta\right) = 0
\]

(2)

where

\[
\Gamma_e \equiv \frac{\partial \mathbf{Q}_e}{\partial \mathbf{Q}_p} =
\begin{bmatrix}
\rho_p & 0 & 0 & 0 & \rho_r \\
\rho_p u & \rho & 0 & 0 & \rho_r u \\
\rho_p v & 0 & \rho & 0 & \rho_r v \\
\rho_p w & 0 & 0 & \rho & \rho_r w \\
\rho_p h + \rho p_\tau - 1 & \rho u & \rho v & \rho w & \rho_r h + \rho p_\tau
\end{bmatrix}
\]

(3)

In order to pre-condition the equations, \(\rho_p\) and \(\rho_r\) in Eqn. (3) are re-defined as the pseudo-properties \(\rho_r^p\) and \(\rho_r^r\), where the \(p\) subscript is used to indicate preconditioned matrices. The
values of the pseudo-properties can be chosen to optimize convergence for different types of flows, such as incompressible, subsonic compressible, supersonic, etc. The preconditioned matrix is given by

$$
\Gamma_e = \frac{\partial \hat{Q}}{\partial \hat{Q}_p} = \begin{bmatrix}
\rho_p & 0 & 0 & \rho_p \\
\rho_p u & \rho & 0 & \rho_p u \\
\rho_p v & 0 & \rho & \rho_p v \\
\rho_p w & 0 & 0 & \rho_p w \\
\rho_p h^* + \rho p_c - 1 & \rho u & \rho v & \rho w \\
\end{bmatrix}
$$

(4)

**Numerical Method**

Eqn (2) will be solved using approximate factorization. In the dual-time-step method, the solution is iterated to convergence in pseudo-time at each physical time step. The physical time differencing will have O(2) accuracy, and the pseudo-time differencing will be O(1). (Second-order accuracy is not required for pseudo-time since it is driven to convergence each step.)

Since two different time coordinates are being used, physical and pseudo, the following notation will be used to differentiate between numerical time differences:

$$\Delta \phi \equiv \phi^{n+1,k+1} - \phi^n$$

$$\Delta \pi \equiv \phi^{n+1,k} - \phi^n$$

and

$$\Delta \phi = \phi^{n+1,k+1} - \phi^{n+1,k}$$

with \(n\) indicating the physical time and \(k\) indicating the pseudo-time.

The O(1) pseudo-time difference term in Eqn. (2) is approximated by

$$
(\hat{Q}_p)_{r} \approx \frac{1}{\Delta \pi} \hat{Q}_{p}^{n+1,k+1} - \hat{Q}_{p}^{n+1,k}
$$

(5)

and the O(2) physical time difference is approximated by

$$
(\hat{Q}_p)_{r} \approx \frac{1}{\Delta \pi} \hat{Q}_{p}^{n+1,k+1} - 2\hat{Q}_{p}^{n+1,k+1} + \frac{1}{2} \hat{Q}_{p}^{n+1,k}
$$

(6)

Spatial differencing will be notated by the symbol \(\hat{\tau}\), with a subscript indicating the direction of the differencing. The \(\xi\)-direction inviscid flux derivative is approximated by

$$\hat{E}_{\xi}^{n+1} = \partial_{\xi} \hat{E}^n + \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p$$

and other inviscid and viscous flux differences are approximated in a similar fashion.

Applying Eqns. (5)-(6) to Eqn. (2) yields

$$\Gamma_e \left( \frac{1}{\Delta \pi} \right) \left( \hat{Q}_{p}^{n+1,k+1} - \hat{Q}_{p}^{n+1,k} \right) $$

$$+ \Gamma_e \left( \frac{1}{\Delta \pi} \right) \left( \frac{3}{2} \hat{Q}_{p}^{n+1,k+1} - 2\hat{Q}_{p}^{n+1,k} + \frac{1}{2} \hat{Q}_{p}^{n+1,k} \right) $$

$$+ \partial_{\xi} \hat{E}^n + \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p$$

$$+ \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p + \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p + \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p$$

$$- \Delta \xi \hat{A}_{\xi} \hat{Q}_p + \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p + \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p + \partial_{\xi} \hat{A}_{\xi} \Delta \xi \hat{Q}_p$$

$$= 0$$

(7)

We want to write this equation in terms of \(\Delta \hat{Q}_p\), since that is the variable for which we will be solving on the left hand side. Adding and subtracting \(\frac{3}{2} \hat{Q}_{p}^{n+1,k}\) within the parenthesis in Eqn.(7) and applying Eqn.(6) yields

$$\left( \hat{Q}_p \right)_r \approx \Gamma_e \left( \frac{1}{\Delta \pi} \right) \frac{3}{2} \Delta \hat{Q}_p $$

(8)

$$+ \Gamma_e \left( \frac{1}{\Delta \pi} \right) \left( \frac{3}{2} \hat{Q}_{p}^{n+1,k+1} - 2\hat{Q}_{p}^{n+1,k} + \frac{1}{2} \hat{Q}_{p}^{n+1,k} \right) $$

Applying Eqns. (5), (6), and (8) to Eqn.(7) yields
All flux Jacobians are at time level \( n \). The last two terms on the right hand side can be combined if desired:

\[
\begin{align*}
\left[ \Gamma_p \left( \frac{1}{\Delta \tau} \right) + \frac{3}{2} \Gamma_\xi \left( \frac{1}{\Delta \tau} \right) \right. \\
+ \partial_\xi \hat{A}_p + \partial_\eta \hat{B}_p + \partial_\zeta \hat{C}_p \\
- \text{Re}^{-1} \left( \partial_\xi \hat{A}_p^\tau + \partial_\eta \hat{B}_p^\tau + \partial_\zeta \hat{C}_p^\tau \right) \Delta_\xi \hat{Q}_p \\
\left. = -\Gamma_\xi \left( \frac{1}{\Delta \tau} \right) \left( \frac{3}{2} \hat{Q}_p^{n+1,k} - 2\hat{Q}_p^n + \frac{1}{2} \hat{Q}_p^{n+1} \right) \right]
\end{align*}
\]

\[
\begin{align*}
&- \left[ \partial_\xi \hat{A}_p + \partial_\eta \hat{B}_p + \partial_\zeta \hat{C}_p \right. \\
&- \text{Re}^{-1} \left( \partial_\xi \hat{A}_p^\tau + \partial_\eta \hat{B}_p^\tau + \partial_\zeta \hat{C}_p^\tau \right) \Delta_\xi \hat{Q}_p \\
&\left. = -\Gamma_\xi \left( \frac{1}{\Delta \tau} \right) \left( \frac{3}{2} \hat{Q}_p^{n+1,k} - 2\hat{Q}_p^n + \frac{1}{2} \hat{Q}_p^{n+1} \right) \right]
\end{align*}
\]

(9)

Before performing approximate factorization, it is convenient to define

\[
S = \Gamma_p + \frac{3}{2} \frac{\Delta \tau}{\Delta \tau'} \Gamma_\xi
\]

The factorized equation is given by

\[
\begin{align*}
&\left[ S + \Delta \tau \partial_\xi \left( \hat{A}_p - \text{Re}^{-1} \hat{A}_p^\tau \right) \right] \\
&\text{S}^{-1} \left[ S + \Delta \tau \left( \partial_\xi \hat{B}_p - \text{Re}^{-1} \partial_\xi \hat{B}_p^\tau \right) \right] \\
&\text{S}^{-1} \left[ S + \Delta \tau \left( \partial_\xi \hat{C}_p - \text{Re}^{-1} \partial_\xi \hat{C}_p^\tau \right) \right] \Delta_\xi \hat{Q}_p \\
&\left( \frac{\Delta \tau}{\Delta \tau'} \right) \hat{Q}_p^{n+1,k} - 2\hat{Q}_p^n + \frac{1}{2} \hat{Q}_p^{n+1} \right]
\end{align*}
\]

(11)

**Right Hand Side**

The time term on the right hand side is straightforward to implement, since it contains no differences. Solutions are required at two previous time steps in order to get second order temporal accuracy. For the viscous terms, standard second-order central differences are used, as is common practice. Inviscid terms are computed using the approximate Riemann solver of Roe\(^3\). The eigenvalues of the system are required for Roe’s scheme. Here, the relevant eigenvalues are those of the matrix \( \Gamma_p^{-1} A_p \), which are given by

\[
\hat{\lambda}_{1,2,3} = U
\]

and

\[
\hat{\lambda}_{4,5} = \frac{1}{2} \left\{ U \left( 1 + \frac{\Delta}{\Delta'} \right) \right. \\
\left. \pm \sqrt{U^2 \left( 1 - \frac{\Delta}{\Delta'} \right)^2 + 4 \frac{\rho h}{\Delta'} \left( \xi_x^2 + \xi_y^2 + \xi_z^2 \right) } \right\}
\]

(13)

where

\[
\Delta \equiv \rho \theta + \rho \rho_p h_f
\]

\[
\Delta' \equiv \rho \theta + \rho \rho_p h_f
\]

\[
\theta \equiv 1 - \rho h_p
\]

(14) \hspace{1cm} \text{(15)} \hspace{1cm} \text{(16)}

Also, for later use, we define

\[
\Delta_p \equiv \rho \theta + \rho \rho_p h_f
\]

(17)
\[ \Delta^T \equiv \rho_T \dot{\theta} + \rho \rho_p h_T \]  
(18)

Similar expressions can be written for the other two coordinate directions by substituting \( V \) or \( W \) for \( U \) and \( \eta \) or \( \zeta \) for \( \xi \) in Eqns.(12) and (13).

**Left Hand Side**

Ideally, fastest convergence is obtained when the same method of differencing is used on both the left and right hand sides of Eqn. (11). This is possible for low-order schemes since the block tridiagonal structure of the equations can be maintained. Higher order schemes require larger difference stencils, and using them on the left hand side precludes the use of block tridiagonal solvers. Here, Steger-Warming\(^4\) flux-vector splitting is used on the left hand side. This does not affect the accuracy of the solution, since the left hand side is driven to zero during the pseudo-time iterations, and it has been shown to be a robust technique.

**Preconditioning Parameters**

Equations (4), (15), (17) and (18) contain two yet-to-be-defined parameters, \( \rho_T \) and \( \rho_T \). The parameter \( \rho_T \) has limited effect on convergence rate for most fluids, and is simply set equal to \( \rho_T \) here.

The square of the speed of sound can be expressed as

\[ a^2 = \frac{\rho h_T}{\Delta} \]  
(19)

where \( \Delta \) is defined in Eqn. (14). If \( \Delta \) in Eqn. (19) is replaced by \( \Delta^x \) as defined in Eqn. (15), an artificial sound speed can be defined,

\[ a_p^2 = \frac{\rho h_T}{\Delta} \]  
(20)

Substituting Eqn. (15) into Eqn. (20) and solving for \( \rho_p \) yields

\[ \rho_p = \frac{1}{a_p^2} \frac{\rho_T (1 - \rho h_T)}{\rho h_T} \]  
(21)

Since \( \rho_T \) has been set equal to \( \rho_T \), we have

\[ \rho_p = \frac{1}{a_p^2} \frac{\rho_T (1 - \rho h_T)}{\rho h_T} \]  
(21)

Once the artificial sound speed is defined, the preconditioning parameter is known. The artificial sound speed, and hence the preconditioning parameter, would be expected to be a function of the characteristics of flow field. Venkateswaran and Merkle\(^1\) have developed a method for estimating \( a_p \) based on the calculation of inviscid, viscous, pressure-gradient, and unsteady velocity scales.

The inviscid velocity scale is simply the physical velocity,

\[ V_{\text{inv}} = \sqrt{u^2 + v^2 + w^2} \]

The pressure-gradient velocity scale is given by

\[ V_{\text{pg}} = \sqrt{\frac{\Delta \rho}{\rho}} \]  
(22)

The unsteady velocity scale is given by

\[ V_{\text{uns}} = \max \left( \frac{l_x}{\pi \Delta t}, \frac{l_y}{\pi \Delta t}, \frac{l_z}{\pi \Delta t} \right) \]

where \( l_x \) and \( l_y \) are characteristic unsteady length scales in the \( x \) and \( y \) directions.

The viscous velocity scale is a bit more complex. It is shown later in 2-D for simplicity. Variables are first defined based on the CFL (Courant Friedrichs Lewy) number and VNN (Von Neumann Number):

\[ \alpha = \frac{\text{CFL}}{\text{VNN}} \left( \xi_x^2 + \xi_y^2 + \xi_z^2 \right) \]
and

\[ \beta = \frac{CFL}{V_{NN}} \sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2} \]  \hspace{1cm} (23)

Defining \( \vec{U} \equiv \xi_x u + \xi_y v + \xi_z w \) (contravariant velocity without the time metric) and \( \vec{V} \equiv \eta_x u + \eta_y v + \eta_z w \), the viscous velocity scale is given by

\[ V_{vis}^2 = \max \left[ \frac{\beta (\beta - \vec{U})}{\vec{U} (\beta - \vec{U}) + \rho \frac{\partial}{\partial t} (\xi_x^2 + \xi_y^2 + \xi_z^2)} \right] \frac{\alpha (\alpha - \vec{V})}{\vec{V} (\alpha - \vec{V}) + \rho \frac{\partial}{\partial t} (\eta_x^2 + \eta_y^2 + \eta_z^2)} \]  \hspace{1cm} (24)

The final velocity scale is then given by

\[ V_p = \min \left[ \max \left( V_{vis}, V_{per}, V_{ata}, V_{visc} \right), a \right] \]

where \( a \) is the speed of sound.

**Code Structure**

The general structure code is based on a well-established compressible, 3-D, unsteady turbomachinery flow solver. It employs a system of overset O-grids and H-grids, with the values on the O-H boundaries being updated each time step by trilinear interpolation from the adjacent grid. The Baldwin-Lomax turbulence model is used for turbulence closure.

**Results**

2-D 1-1/2 Stage Turbine

The first test case is the simulation of the flow based on experiments on a 1-1/2 stage Low Speed Research Rig (LSRR) turbine at United Technologies Research Center.

The grid dimensions in each of the three blade rows are 109x45 in the H-grid and 251x41 in the O-grid. The grid is shown in Fig. 6.
The match is good for the stators, with some discrepancy on the suction surface of the rotor. The suction-surface mismatch has been observed in a number of previous simulations of this configuration by other researchers.

Figure 11 shows entropy contours. The boundary layer growth and wake shedding can be seen in each blade row. The wake from the first stator convects into the rotor passage and takes on the characteristic hairpin shape as it becomes entrained in the rotor blade surface boundary layers. Both the stator-1 wake and the rotor wake then interact with the downstream stator, creating a complex pattern of high and low entropy fluid.

**TPO Supersonic-Inflow Turbine**

The next test case is a simulation of the flow through a preliminary geometry of the supersonic-inflow Turbine Performance Optimization (TPO) turbine. This case was chosen to test the shock capturing ability of the scheme in order to validate the coding of the flux-difference splitting. Two vanes and five rotor blades were contained in the simulation. In the vanes, each H-grid contained 65x51 points, and each O-grid contained 171x31 points. The rotor blade H-grids each contained 75x51 points, and the O-grids each contained 171x21 points. The grid is shown in Fig. 12.
The General Equation Set method has been implemented along with general fluid properties for turbomachinery applications. A series of subsonic and supersonic test cases have been run, and the code has been demonstrated to perform well.

The dual time step algorithm requires specification of a parameter (the ratio of pseudo-time-step to physical time step), and the choice will affect the convergence rate. It is recommended that a series of simulations be performed on typical configurations using a range of values for this parameter. This will help to deduce a range of the parameter's value to optimize convergence rates. This will be particularly for computing incompressible and nearly-incompressible flows.

The code has been constructed in a modular fashion to make future enhancements as easy as possible. One such enhancement would be to develop a series of routines to compute the properties for a variety of fluids encountered in rocket propulsion turbomachinery (liquid and gaseous hydrogen, oxygen, etc.).

**Additional Results to be Shown for Final Paper**

The final paper will show results for water flow (presently underway), and also at least one 3-D
result. If necessary, the text and equations in this abstract will be cut down for brevity.

References


