AeroSoft, Inc.
200 Kraft Drive, Suite 1400
Blacksburg, Virginia 24060-6363
(540) 557-1900
(540) 557-1919 (FAX)
http://www.aerosoftinc.com

Contents

Chapter 1

Multi-Zone Rocket Problem

In a previous tutorial (Duct-1), a single-zone, two-dimensional, transonic duct problem was discussed. We recommend that you complete the Duct-1 tutorial before beginning this tutorial. This tutorial will cover another test case, with the focus being on three-dimensional, multiple-zone problems. The tutorial will give the user a good feel for how to run large, multi-zone problems in parallel.

Before beginning this tutorial, the user will benefit from reading the chapters on the graphics window and tree view in the $\mathcal{GASP}$ Version 5 reference manual. With these chapters read and with a basic knowledge of CFD, a user should be able to start with this tutorial and learn the basics of $\mathcal{GASP}$ Version 5.

This tutorial is intended to take a user with nothing more than a grid file and the $\mathcal{GASP}$ software, and successfully run a fairly complex problem using CFD. The process will involve setting up the problem using $\mathcal{GASP}$'s graphical user interface (GUI). Once the problem is set up, the CFD solution can be computed using the flow solver. This article is written as a tutorial, so it is best if the user follows along using $\mathcal{GASP}$.

1.1 The Supersonic Rocket Problem

The sample case for this tutorial is a rocket. The rocket grid is three dimensional and consists of eight zones. The eight zones make up a total of 213,120 cells (234,732 grid points). The flow conditions are consistent with a small angle of attack and zero yaw. With the constraint of zero yaw, only half of the computational grid is required since the solution will be symmetric in one of the three coordinate directions. A picture of the symmetry plane, exit plane and rocket surface grid is shown in Fig. 1 on pg. 5.

A viscous solution about the rocket is obtained that assumes a calorically and thermally perfect gas. The rocket contains a nozzle, which is also modeled assuming a calorically and thermally perfect gas. This assumption may not be valid for real life applications, but is a sufficient approximation for this tutorial. The freestream conditions for the rocket are given below.

$$\begin{aligned} MachNumber &= 2.0 \\ Temperature &= 220.0K \\ Density &= 0.10kg/m^3 \\ FlowAngleofAttack &= 2^\circ \end{aligned}$$

Conditions for the nozzle are also needed. These are given next.

$$\begin{aligned} MachNumber &= 1.05 \\ Temperature &= 3000.0K \\ Density &= 2.0kg/m^3 \end{aligned}$$

Starting with the grid and flow conditions described above, the process of computing a CFD solution will be covered.

The case described in this tutorial is located in the `aerosoft/share/tutorial/Rocket-2` directory. With the standard download and install, this directory contains the grid and documentation for the rocket tutorial. You should find the following files there: `rocket.p3d`, `rocket.pdf` and `README`. The file `rocket.p3d` is the PLOT3D grid file that will be used for this problem. The file `rocket.pdf` is a `PDF` file containing this tutorial.

A completed version of the rocket problem, as well as a ready-to-run input deck, can be download from AeroSoft's $\mathcal{GASP}$ Version 5 ftp site. This is provided for users who desire to run the flow solver without necessarily going through the process of setting up the input deck.

Note that files with the `.gsp` extension represent the input deck. The `.gsp` file contains all the GUI parameters, including post processing settings. Files with the `.grd` extension contain the grid and files with the `.sln` extension contain the solution. One last file generated from setting up the case is the `.aux.gsp` file which contains auxiliary solution information created at run-time.

1.2 Setting Up The Input Deck

While inside of the `Rocket-2` directory, run the $\mathcal{GASP}$ graphical user interface (GUI). This is done by typing gaspfrom the command line, which assumes that the users path directory has been properly set up to find the gasp executable.

1.2.1 Importing the Grid

The first step in setting up a new input deck is to import the grid. At the top of the $\mathcal{GASP}$ GUI there is a menu bar with the following options: File, Commands, Wizards,

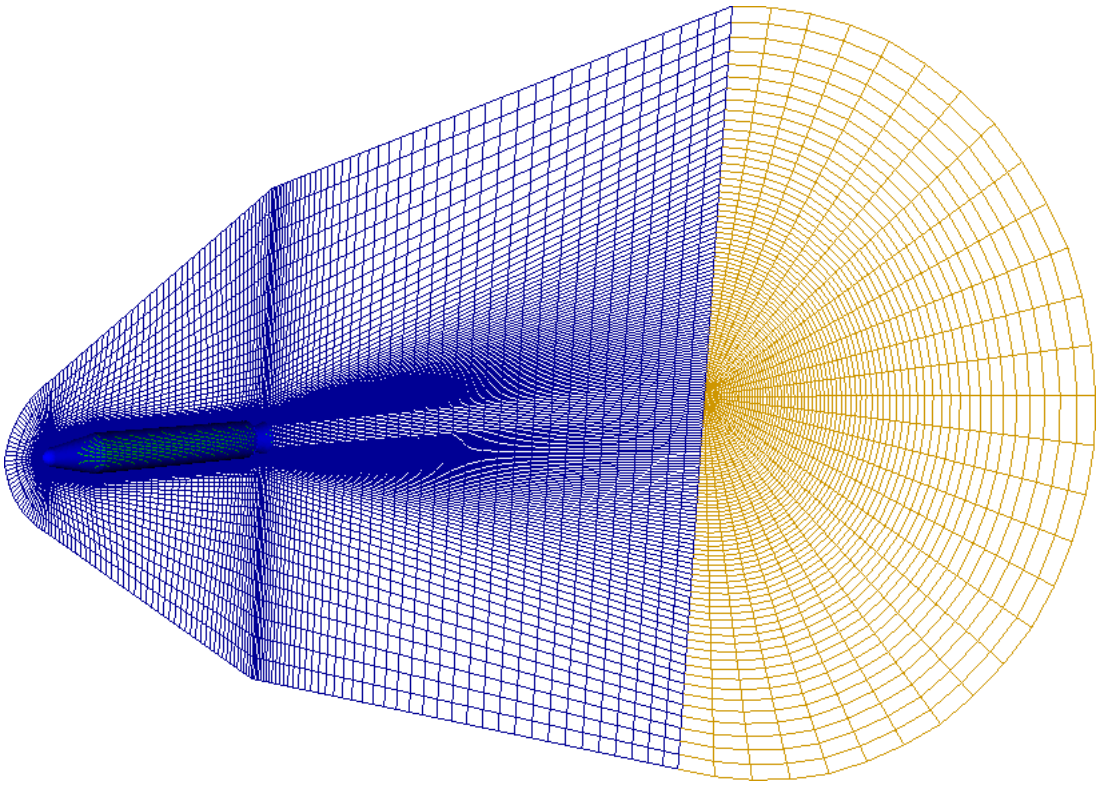

Figure 1: A view of the problem symmetry plane, exit plane, and rocket surface grid for the eight zone, supersonic rocket problem. Total number of grid points is 234,732 (213,120 cells).

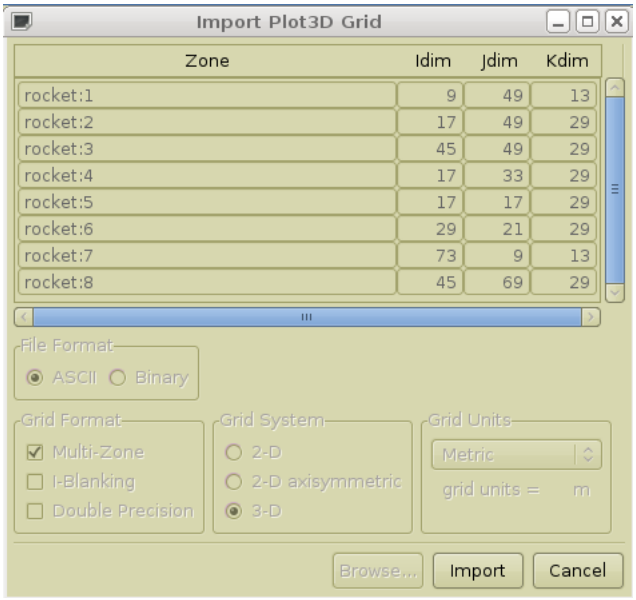

Figure 2: The import grid pop-up menu after selecting the rocket grid file.

View, Options, and Tabs. The File→Import Grid→Plot3D Grid... option from the main menu is used to import PLOT3D grids. Upon selecting this option, a pop-up window will appear. Before selecting the grid file, the user should make sure that all of the appropriate formatting options are correctly set. The default values are correct for the current tutorial. For example, the grid supplied in the plot3d file for the current problem is stored in meters (SI units). The Grid Units should therefore display "`Metric`". Press the Browse... button, after which time a file selection pop-up window will appear displaying all the files in the current directory. Select `rocket.p3d` from the displayed list of files. With the grid file selected, press Open to finalize the selection. The grid information should now appear in the pop-up window, listing the *i,j,k* dimensions for each zone. The pop-up window should now appear as shown in Fig. 2 on pg. 6. At this point, the grid file has been opened and the basic grid information has been read. If the user does not wish to use this grid system, a new grid file may be opened and the current grid would be discarded. To use the currently selected grid, import it into $\mathcal{GASP}$ by pressing the Import button. This will add or append the selected grid to $\mathcal{GASP}$. The grid for the rocket has now been imported and is ready for use.

1.2.2 Viewing the Grid

When $\mathcal{GASP}$ is first executed, the drawing mode is set to "`Sparse`". Upon importing the PLOT3D grid, an outline of all the zones in the grid will be visible. In this example, the body of the rocket should be visible in the graphics window. The remainder of the grid domain is not in view at this time. In order to view the entire grid domain, select all of the objects by clicking the "`Flow Solver`" entry within the tree view located in the upper left portion

of the GUI. For more information on manipulating the tree view, refer to the Reference Guide. Select the Zoom Selected menu button found under the Commands menu, or press Ctrl - Z in order to center the selected objects within the viewing window. The entire grid outline should now be in view. The grid contains eight zones, each of which became highlighted when the "`Flow Solver`" text was selected.

To view individual zones, you first need to see a list of the zones. The zones are located in the tree view under the "`Zones`" folder. Open this folder by clicking on the icon and a list of eight zone nodes should appear. Using the mouse, select the "`Zones`" entry such that it is highlighted. Notice that the grid lines in the graphics window are no longer highlighted. This is because the "`Zones`" folder has a render mode of "`None`" upon startup. Change the draw mode for this folder to "`Sparse`" in order to see each zone highlighted when individually selected. This is done by changing the Render Still selection located beneath tree view or by using the right mouse button and selecting Render→Sparse. With display of the zones enabled, now select a zone inside of the "`Zones`" folder and it should appear highlighted in the graphics window. The zone can be zoomed in on by using the Zoom Selected menu button as before.

Notice that the render mode for each zone is Full, but the zones are being drawn as sparse. This is because the level of rendering is limited to the "`Zones`" setting, which in turn is limited to the "`Flow Solver`" render setting. In other words, a node's rendering is limited by it parents mode.

1.2.3 Setting the Reference Quantities

Now that the grid has been imported, we will step through the process of setting up the input variables for the problem. We will start with the reference quantities.

Directly beneath the graphics window is a set of eight tabs labeled: File, Zone, Zonal Boundaries, Physical Models, Run Definitions, Solution Visualization, Graphics Window, and File Output. These buttons control the display area below the graphics window. When *GASP* is first executed, the File tab is selected. Here, the user can set the reference quantities used in the non-dimensionalization of the flow quantities. The reference quantities aid in the reduction of round-off error that can be significant for certain problems. A good rule of thumb is to use the freestream values for the reference quantities. For this problem, this would yield the following:

$$\begin{aligned} \rho_{\text{ref}} &= 0.10\, kg/m^3 \\ T_{\text{ref}} &= 220.0\, K \\ U_{\text{ref}} &= 595.0\, m/s \\ L_{\text{ref}} &= 1.0\, m \end{aligned}$$

After entering these values for the reference quantities, the display should look like Fig. 3 on pg. 8 Note that we have chosen to set up this problem using metric units. We could also

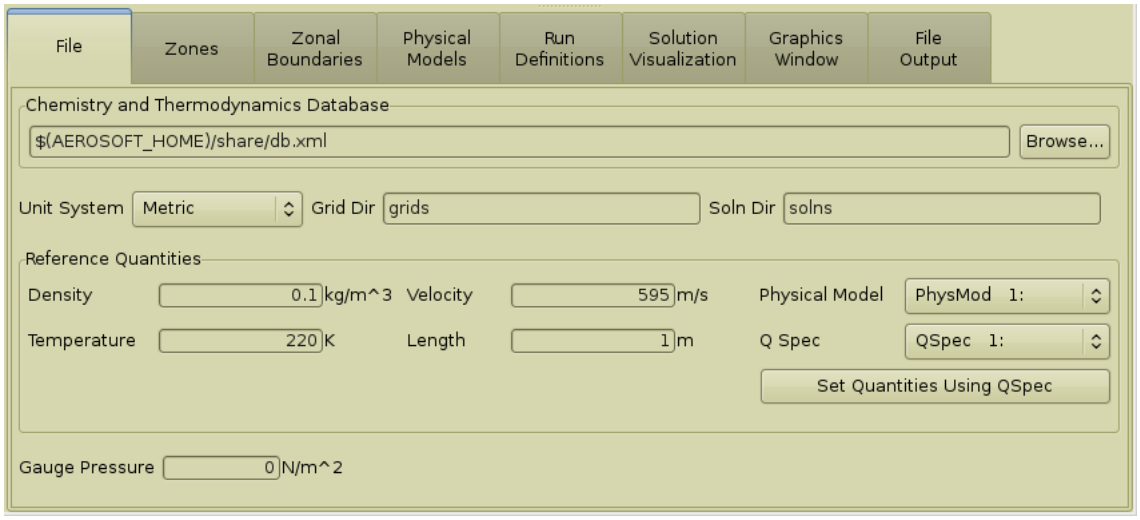

Figure 3: The area of the GUI under the graphics window (Main frame) showing the reference quantities after set up.

use English units even though the grid is in meters. $\mathcal{GASP}$ will take care of any differences regarding the grid units and the input deck units.

Also available in the File frame is the chemistry and thermodynamics database and directory names for the grid and solution files. The chemistry database is required to set up and run $\mathcal{GASP}$. A default database file named `thermochemdb.tcd` comes with $\mathcal{GASP}$ and will be read in when the GUI is launched.

1.2.4 Saving the Input Deck

Now is a good time to save what we have done so far. Under the File menu, select Save. The GUI will prompt the user for a filename if one has not already been given. By default, the file name "untitled" is given. Type in a filename with or without the ".gsp" extension, and press the OK button. If the file extension is left off, $\mathcal{GASP}$ will automatically add it.

If, for example, you choose the name `rocket.gsp` for the file, additional files are created named `rocket.grd`, `rocket.sln`, and `rocket.aux.gsp`. The `rocket.grd` file is placed in the `grids` directory, the `rocket.sln` file is placed in the `solns` directory, and the `rocket.aux.gsp` is place in the current directory with `rocket.gsp`. These four files (.gsp, .aux.gsp, .grd, and .sln) are the basic files for a given problem.

1.2.5 Computing the Zonal Boundaries

With the information inside the File frame completed, we will move on to the Zonal Bouns frame. Select the Zonal Boundaries tab located beneath the graphics window. A new display will appear with with the following tabs: Pt-2-Pt ZB, Cut Holes, Chimera Coefs, and Zipper Grids, Sequence, and Grid Range. Only the Pt-2-Pt ZB tab will be used here.

The remaining tabs are used for Chimera overlapping grids, which we do not have in this problem. These tabs are described and used in the Chimera-5 tutorial.

GASP has an algorithm for detecting point-to-point zonal boundaries. If zonal boundary information is not provided upon import of a CGNS file, most zonal boundary detection will be performed automatically using an algorithm within the GUI. While inside the Pt-2-Pt ZB tab, press the Compute Pt2PtZB button and all the zonal boundaries that have point-to-point zonal boundaries will be found. *GASP* may take a few seconds to perform the calculation.

Upon completion of the automatic zonal boundary detection, it may not appear as if anything has changed. The Zonal Boun display appears the same. However, the computed zonal boundaries have been added to the "`Pt2Pt ZB`" folder in the tree view, which is a sub-folder of "`Zonal Boundaries`". Open this folder to view all the zonal boundaries. The folder should appear as in Fig. 4 on pg. 10. After viewing the contents of the folder, you may wish to close the folder to simplify the tree view.

By default, the display of point-to-point zonal-boundary surfaces is disabled. For complex three dimensional geometries with many zonal boundaries, the display of surfaces in the viewing window should become noticeably less cluttered upon completion of the zonal boundary detection.

The zonal boundaries are now setup and ready. The user does not need to set any other parameters for the zonal boundaries. If the user wishes to see the zonal boundary information for a given ZB, this can be done by selecting a zonal boundary folder in the tree view. The information for that zonal boundary will appear in the Pt-2-Pt ZB tab as direction 1 and direction 2 data. In this way a user may change the zonal boundary information if necessary. The user also has the freedom to manually create the zonal boundary, which is explained in the *GASP* Version 5 reference manual.

1.2.6 Creating the Boundary Conditions

GASP allows the user to group surfaces into families (or surface groups), which may then have a single boundary condition applied to the group. This is all done in the tree view where a boundary condition family is represented by a folder inside of the "`Surfaces`" folder.

Upon importing a new grid, by default, all of the surfaces are placed in the "`Undefined`" folder under the "`Surfaces`" folder. Clicking on the "`Undefined`" folder icon will now highlight all the surfaces for the eight zones. The surfaces that correspond to the zonal boundaries have already been removed and placed in the "`Pt2PtZB`" folder as described in the previous section.

The user may add as children to the "`Surfaces`" folder any number of folders, but keep in mind that each folder will have only one boundary condition applied to it. In this problem, there will be five boundary conditions needed, therefore five new folders will be created; one for each boundary condition.

To create a folder, select the "`Surfaces`" node and then click and hold down the right mouse button (use the left mouse button to select the folder). Select the Edit→New Folder

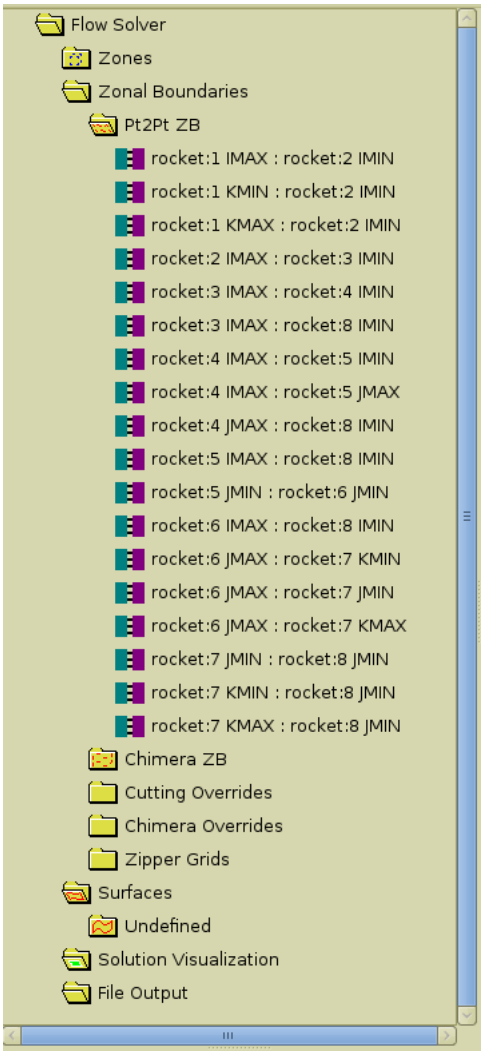

Figure 4: The point-to-point zonal boundary folder displaying additional folders containing the zonal boundaries.

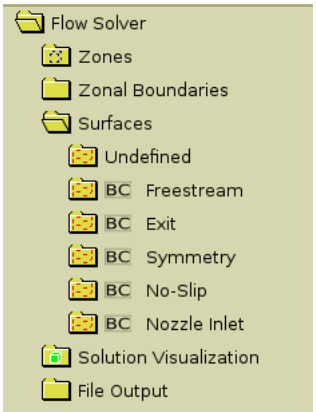

Figure 5: The Surfaces folder displaying additional created folders for the boundary conditions.

pop-up menu item to create a new folder as a child of the "`Surfaces`" folder. Perform this operation five times to create five new folders. The new folders will appear inside the "`Surfaces`" folder and will have the name "`Untitled`" by default. Tree item names may be edited by double clicking on the text for that item. Give the new folders the following names: Freestream, Exit, Symmetry, No-Slip, and Nozzle Inlet. The tree view should appear as in Fig. 5 on pg. 11.

With the new folders created, each surface needs to be moved from the "`Undefined`" folder to its new surface family. To do this, the user must first select the surface to be moved. This can be done by either left clicking on the text next to the appropriate icon, or graphically selecting the item from the graphics window by moving the mouse over the desired item, and pressing the left mouse button while simultaneously pressing the Alt key. The render mode for the "`Zones`" folder should be set to None and the "`Surfaces`" folder should be set to Sparse in order to efficiently select surfaces from the graphics window. For either method, the Undefined folder needs to be opened such that all the surfaces are displayed in the tree.

Once a surface has been selected it can be moved to its new location by one of two ways. The first method is by pressing and holding the middle mouse button over the new folder in order to bring up a Drop Selected Nodes menu item. The second method of moving selected items to a new location is by pressing and holding the middle mouse button over a selected tree item, followed by moving the cursor to the desired location and releasing the middle mouse button. Notice that visual feedback is provided to identify acceptable drop locations. With either of these two operations, move the surfaces into their appropriate folders. A list of the boundary conditions with their respective surfaces is given below.

Folder	Surface		
Freestream	rocket:1 JMAX	rocket:2 JMAX	rocket:3 JMAX
	rocket:8 JMAX		
Exit	rocket:7 IMAX	rocket:8 IMAX	
Symmetry	rocket:1 IMIN	rocket:2 KMIN	rocket:2 KMAX
	rocket:3 KMIN	rocket:3 KMAX	rocket:4 KMIN
	rocket:4 KMAX	rocket:5 KMIN	rocket:5 KMAX
	rocket:6 KMIN	rocket:6 KMAX	rocket:7 JMAX
	rocket:8 KMIN	rocket:8 KMAX	
No-Slip	rocket:1 JMIN	rocket:2 JMIN	rocket:3 JMIN
	rocket:4 JMIN	rocket:6 JMIN	
Nozzle Inlet	rocket:6 IMIN	rocket:7 IMIN	

Before we move on, we must make one correction to the surface families. With the rocket grid used here, there are two surfaces that should be no-slip, but were computed as zonal boundaries because they were represented as an infinitely thin surface. These two surfaces define the aft portion of the nozzle and are labeled as rocket:5 JMIN and rocket:6 JMIN. These two surfaces are located in the "`Pt2PtZB`" folder and must be moved to the "`No-Slip`" folder. We will perform this operation next.

Open the "`Pt2Pt ZB`" folder, which will display additional folders. Each one of these folders contains a pair of surfaces, which represents the zonal boundary surface pair. The folder we need to delete is called "`rocket:5 JMIN:rocket:6 JMIN`". Click on the folder text (to highlight it) and select Edit→Cut using the right mouse button. This moves the two surfaces to the "`Undefined`" folder. Open the "`Undefined`" folder and verify that the two surfaces are there. Now move the two surfaces to the "`No-Slip`" folder.

We are now at the final step in setting up the surface families. From the tree view, the user must identify surface folders to which a boundary condition will be applied. To make a surface folder an active boundary condition, select the Edit→Toggle Folder BC On/Off pop-up menu choice using the right mouse button. In the current example, the BC button should be enabled for each of the five new surface folders.

1.2.7 Sequencing the Grid

Mesh sequencing is a powerful tool that can help reduce the required CPU time needed to get the final solution of a problem. When possible, this feature should be exploited. For the current rocket problem, we will sequence the grid twice, giving a total of three grids on which to compute. The grid that we imported earlier will be the fine grid. Grid points can never be added to a grid using $\mathcal{GASP}$, only removed. The two new grids will be denoted the medium and coarse grids. Sequencing will take place from the Zones frame.

If the Zones frame is not already displayed, select the Zone tab located directly under the graphics view. Sequencing operations in the Zones frame are always performed on all zones in order to maintain consistency between zones across sequence levels.

In the Zones frame, select the Sequencing tab in order to display the sequencing infor-

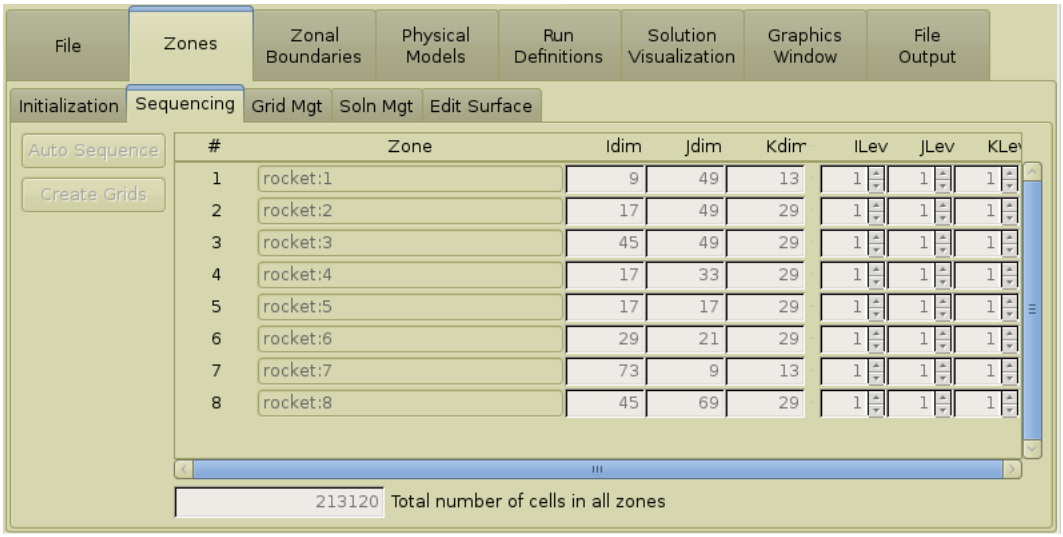

Figure 6: The sequence information for the fine grid.

mation. The lower half of the GUI should look like Fig. 6 on pg. 13. In the Sequencing tab, all the zones are always displayed in table format with their sequencing information. This differs from the Initialization and Grid/Soln Mgt tabs which only display those zones which are currently selected in the tree view. The dimensions of each zone are given along with the level of sequencing for each of the i, j and k grid directions.

Located in the lower left corner of the GUI is the list of sequences. At this point, only one sequence exists, which corresponds to the imported rocket grid: Sequence 1. This sequence will be our fine grid, and so we will rename it to 'Fine Grid'. You can rename the sequence by double clicking on the name and entering new text.

To create the medium grid (the second sequence), first press the New button just above the sequences list. Give the new sequence the name "`Medium Grid`". Pressing the New button creates an empty sequence. The next step will be to define the new sequence, which can be done using either the Auto Sequence button or by adjusting the ILev, JLev, and KLev inputs manually. Initially, these values are set to 1, meaning that every grid line is kept. A value of 2 in a specific direction indicates that every other grid line will be deleted from the grid, thus reducing the dimension in that direction by a factor of two. For this problem, the i, j, and k grid directions will all be reduced by a factor of 2. Press the Auto Sequence button at this time, and notice that the number 2 now appears in all of the ILev, JLev, and KLev input boxes. Note that this could have been done manually as well. Now press the Create Grids button to finish making the new sequence. When the input deck is saved, the new sequence will be saved to the `.grd` file.

The coarse grid can be created following the same steps as the medium grid. Create an empty sequence using the New sequence button. Press Auto Sequence followed by Create Grids. Give the final sequence the name 'Coarse Grid'. The GUI should now appear as shown in Fig. 7 on pg. 15. Save the input deck at this time and the new grids will be saved

to file.

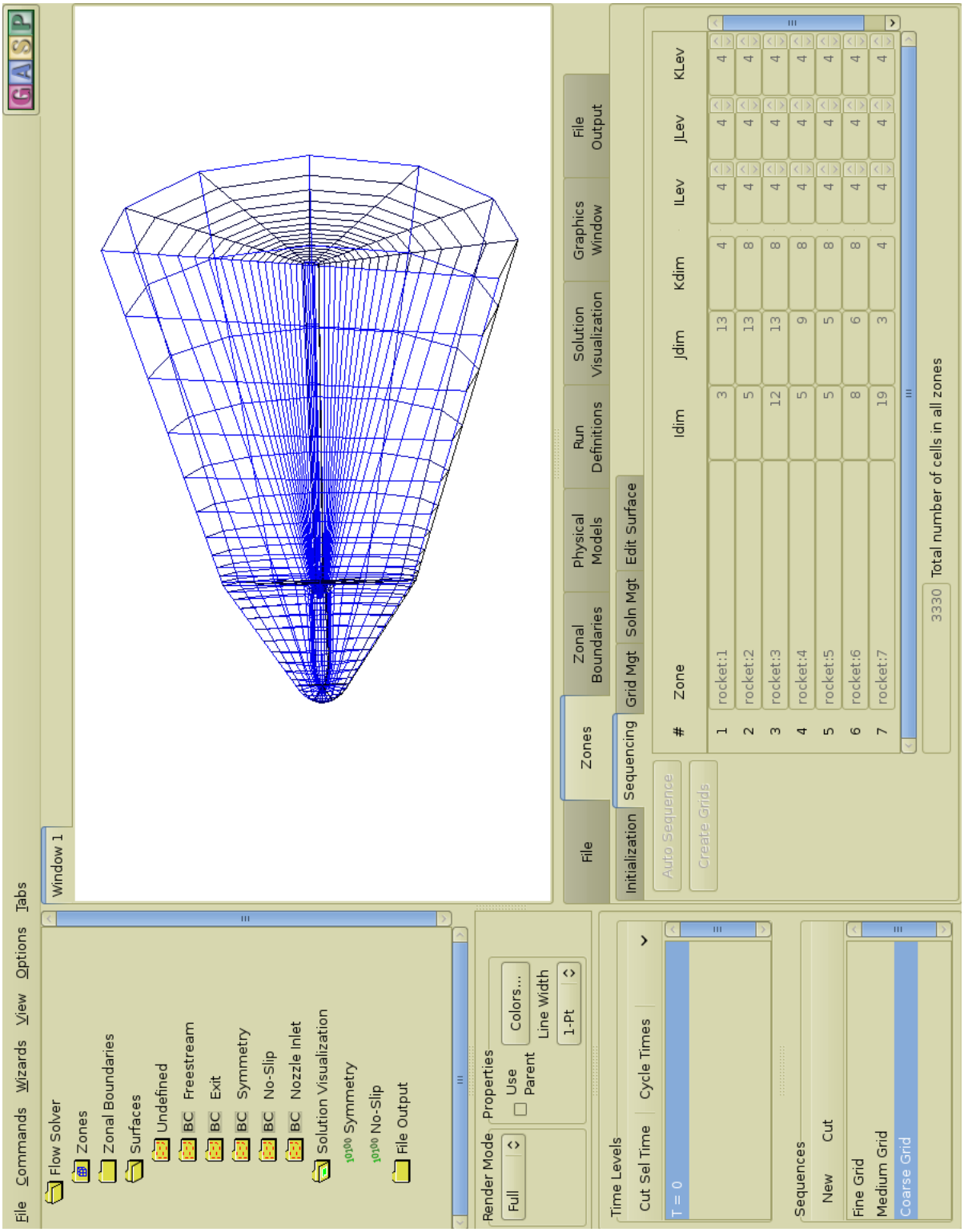

Figure 7: *GASP* after creating the medium and coarse sequence levels with zone wire frame rendering turned on.

1.2.8 Creating the Physical Model

We must now create the physical model. A physical model specifies the freestream and initial conditions, the inviscid and viscous fluxes, the chemistry model, and the boundary conditions. For the current problem, we will need to create just one physical model. This physical model will be used to specify the initial flow conditions, the freestream values, and the nozzle inlet conditions. For the inviscid fluxes, the Roe flux difference splitting scheme will be specified with third order spatial accuracy. For turbulence, we will use Wilcox's k-omega turbulence model. And for the chemistry model, the perfect gas assumption is made.

At this point, select the Physical Models tab to display the Physical Models frame. A physical model will always exist by default. Initially, the physical model is given the name 'PhysMod 1:', which is displayed to the left of the Physical Models frame (bottom left corner of the GUI window). We will rename the default physical model to be the primary model, so change the name of the model to "`Primary Phys Mod`".

1.2.9 Thermodynamics and Chemistry

The default setting for the chemistry model is perfect gas, and the default for the thermodynamics model is equilibrium translation and rotation. These settings are appropriate for the current sample problem, and no changes need to be made to the Thermo-Chem tab.

1.2.10 Inviscid Fluxes

The next set of parameters to set for the physical model will be the inviscid fluxes. This is done by selecting the Inviscid tab.

The Roe flux will be used for all three coordinate directions. The 'Accuracy will be set to "`3rd Order Up. Bias,`" which stands for third order upwind bias based on the MUSCL scheme. Note that the Kappa parameter reflects the appropriate value used in the MUSCL scheme. When higher-order spatial accuracy is applied to a compressible problem, the use of a limiter is always recommended. Select "`Min-Mod`" for the Limiter in the I-Direction, J-Direction, and K-Direction. The Inviscid tab should now appear as shown in Fig. 8 on pg. 17.

1.2.11 Viscous Fluxes

While the inviscid fluxes are always necessary in setting up a problem, the viscous effects may sometimes be neglected. If the viscous effects are included (as for this problem), the user may choose to model the flow as either laminar or turbulent. For this particular problem, the flow will be simulated as turbulent. Note that a turbulent calculation will require greater grid resolution near no-slip walls and free-shear regions.

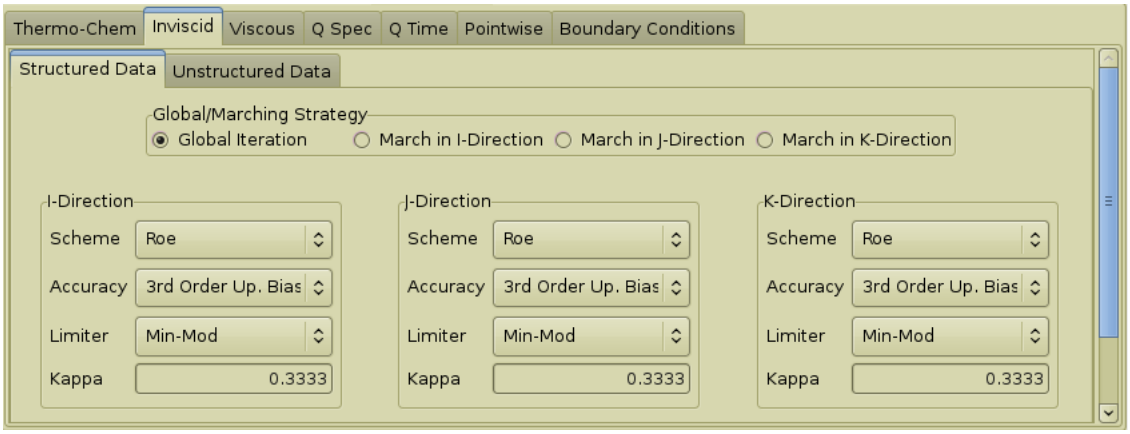

Figure 8: The complete setup for the inviscid fluxes.

Select the Viscous tab to show the various options for running a viscous problem. As stated above, this problem will be run turbulent, so select the Turbulent radio button inside the Viscous Flux Mode group.

The viscous flux contributions are grouped into two types: thin-layer (primary) terms and cross-derivative (secondary) terms. The user can select thin-layer fluxes to be computed in any of the three coordinate directions. Any combination of the thin-layer terms can be selected. As an example, if only the J-Direction flux is selected, the viscous flux will only be composed of numerical derivatives calculated in the j direction. In most cases, the cross derivatives are insignificant compared to the thin-layer terms (or cannot be adequately resolved), and may be safely neglected without affecting the solution. For the current problem, we will use the thin-layer terms in each of the three coordinate directions, and will neglect the cross-derivative terms. The thin-layer terms will need to be turned on, so make sure the toggle button for each variable in the Viscous Flux Terms box is selected. Note that these were turned on when the Turbulent radio button was pressed.

The cross-derivative contributions are symmetric. For example, cross derivatives on a constant i face in the j direction are included with cross derivatives on a constant j face in the i direction. The thin-layer contributions are included whenever cross derivatives are included in any given coordinate direction. This is because the magnitude of the thin-layer component will always be greater than the magnitude of the cross-derivative component.

Proper resolution of viscous terms in any given direction requires adequate grid refinement in that direction. For example, if two walls are to be modeled, but the grid is only clustered towards one wall, then the boundary layer at the second wall may not be adequately resolved even if the viscous terms are included in the simulation. Adequate grid resolution is one reason why the complete Navier-Stokes equations are usually not modeled.

The Wall-Gradient parameter controls the spatial accuracy of the viscous-gradient calculations at the boundaries. Viscous calculations use second-order, central-difference gradients for interior faces. For consistency at a wall, the Wall-Gradient Calculation should be "`2nd Order Accurate`", which is the default setting.

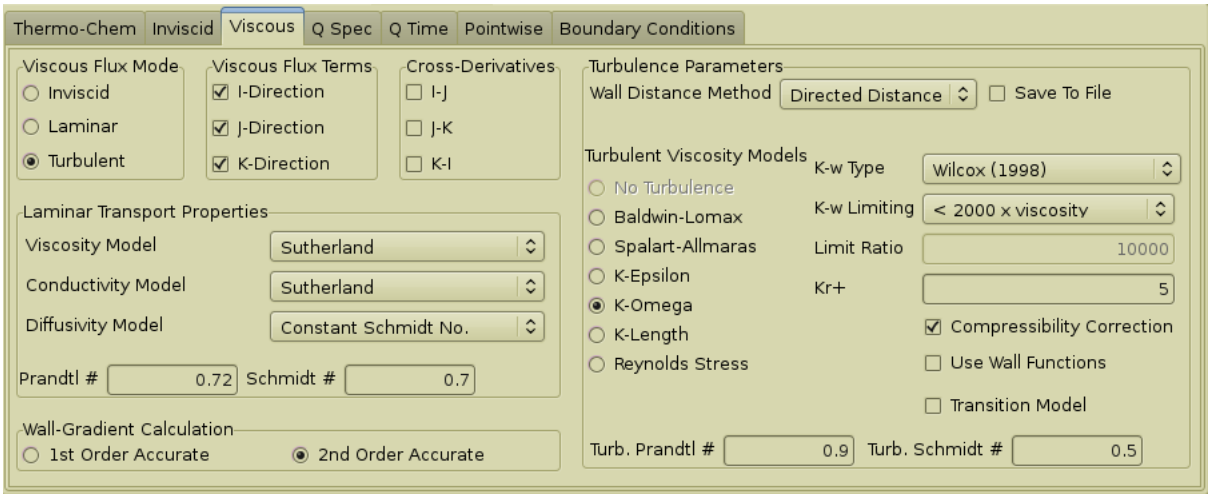

Figure 9: The viscous flux settings for the rocket problem.

Moving on, the basic viscous-transport properties are located at the bottom left of the display. The viscosity, thermal conductivity and diffusivity are called a fluid's transport properties. Several different methods are currently available to approximate these quantities. However, not all models are supported by all chemistry models. The terms in the Laminar Transport Properties are used when running either a laminar or turbulent problem. The default settings are used for this problem.

The right hand portion of the frame contains the Turbulence Parameters group, where the turbulence model needs to be set. Choose the "`K-Omega`" turbulence model and notice that several more options will appear for this model. For the K-w Type, choose the "`Wilcox (1998)`" model. For K-w Limiting, select the "`< 2000 x viscosity`" limit option. This option will limit the eddy viscosity such that it will never exceed a value that is 2000 times the laminar viscosity. This limiting is done on a cell level, since the laminar viscosity differs from cell to cell. The limiting is for numerical reasons, and may be relaxed as the solution nears convergence. Turn on Compressibility Correction because we are concerned with the spreading rate of the plume. Note that the compressibility correction only affects mixing layers and shear layers (in theory), and not wall bounded flows. In reality, the compressibility correction can introduce errors in the solution in hypersonic wall bounded flows. The viscous flux settings should look like Fig. 9 on pg. 18.

1.2.12 Q Specification

Select the Q Spec tab to show and edit the information regarding flow conditions. The user can create any number of q specifications, and have them be used for various purposes such as solution initialization or references to boundary conditions. There will be three q specifications for this problem, which will be used to initialize the solution, specify the freestream conditions, and specify the nozzle inlet conditions respectively. The default q

specification is named "`QSpec 1:`". We will start with the q specification used to initialize the solution, so change the q specification name to "`Initial Q`". We will initialize the solution to a subsonic Mach number, which will help to properly set up the flow physics. Starting with a large Mach number may cause numerical problems in the vicinity of the blunt nose or around the nozzle.

For the initial conditions, set the Mach number to 0.2, while the temperature and density are set according to the freestream conditions. The temperature is set to $220K$ and the density to $0.1kg/m^2$. The problem is going to be run at a two degree angle of attack. This should be reflected in the initial conditions, so change the direction cosines. The $u/|V|$ value should read 0.99939082, the $v/|V|$ value should read 0.03489949, and $w/|V|$ 0.0. The user could also use the Input angle option and enter the angle in degrees. This completes setting up the q specification for the initial conditions, which should appear as in Fig. 10 on pg. 20.

The freestream q spec will be set up next. Press Copy from the Q Spec list. This is the entry that contains "`Initial Q`". Then press Paste and rename the q spec to "`Freestream Q`". We are using Copy and Paste because the freestream values are the same as the initial values, with the one exception that the Mach number is now 2. Change that value now.

Make the final q spec by pressing New again and give the third q specification the name "`Nozzle Q`". The nozzle inlet conditions were given at the beginning of the chapter. The Mach number is 1.05, the temperature is $3000K$, and the density is $2.0kg/m^3$. The direction of the flow is aligned with the x-axis, which is the default setting for the direction cosines. We are now finished setting up the three q specifications.

1.2.13 Boundary Conditions

The Boundary Conditions tab in the Physical Models frame is for setting the boundary conditions. Select this tab now. The display will show a table listing each of the surface folders that were created in Sec. 1.2.6 on pg. 9. If there are no surfaces listed here, make sure the BC button has been enabled for each surface folder in the tree view.

We have five boundary conditions to set for the rocket problem. The boundary condition is set by selecting one of the options from the option menu in the Flow B.C. column. By default, each surface group (or surface family), has the "`1st Order Extrapolation`" boundary condition assigned to it.

For the Freestream surface family, assign the "`Fixed at Q`" boundary condition with a split flux. It is appropriate for this problem to leave the Turbulence B.C. set to "`Default Setting`". Assign "`1st Order Extrapolation`" to the "`Exit`" family, "`Symmetry Plane`" to the "`Symmetry`" family, "`No Slip Adiabatic`" to the "`No-Slip`" family, and "`Fixed at Q`" to the "`Nozzle Inlet`" family. For the "`No-Slip`" family, the Split toggle button should be turned off to indicate a full flux (see Fig. 11 on pg. 21). The split value will automatically be set when the user selects a boundary condition. The user may then override the setting if desired, but in most cases the default setting is appropriate.

The Q Spec column of option menus allows the boundary condition to be associated with any of the q specifications. Not all, but some of the boundary conditions require a

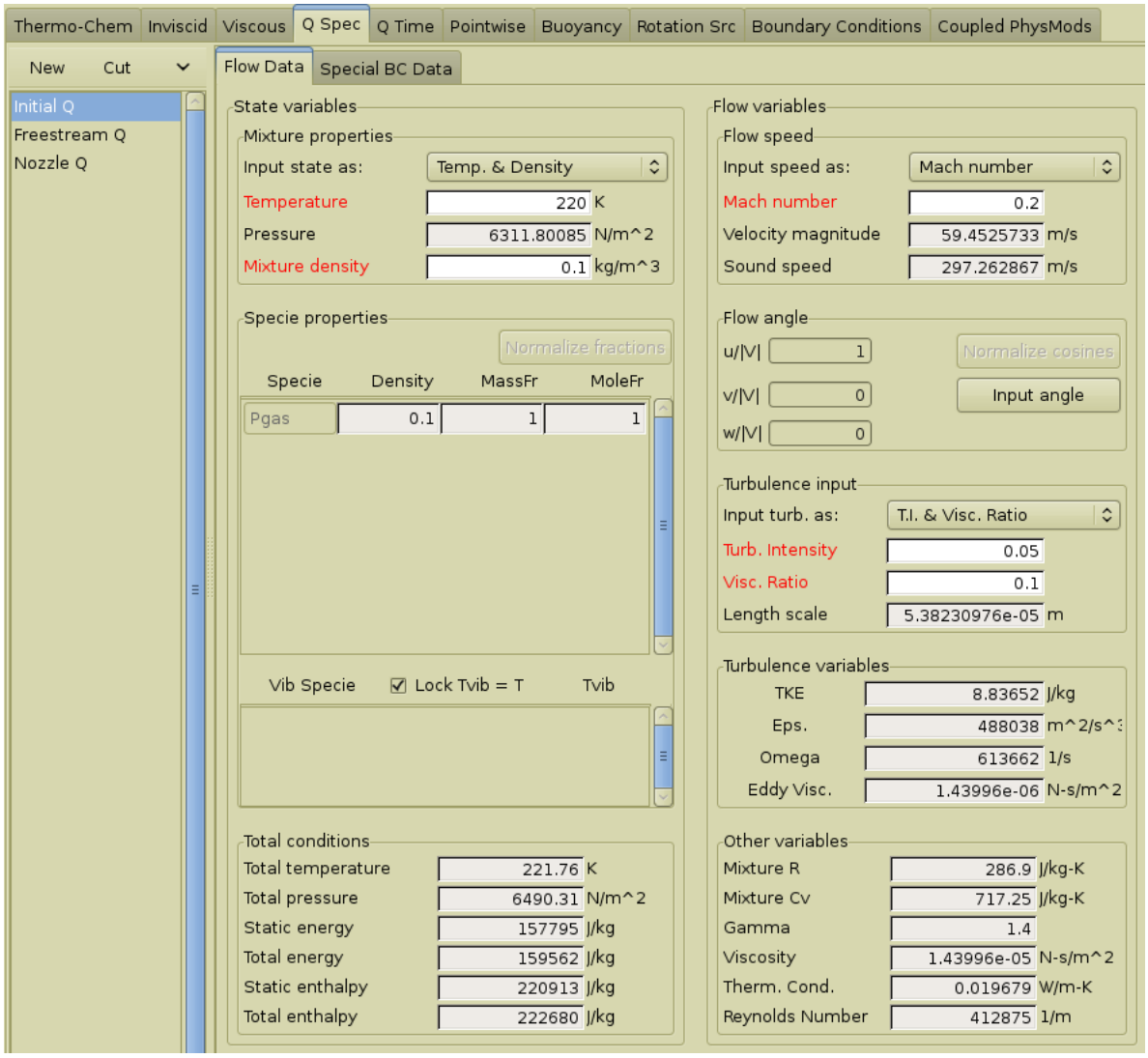

Figure 10: The q specification for initializing the flow solution.

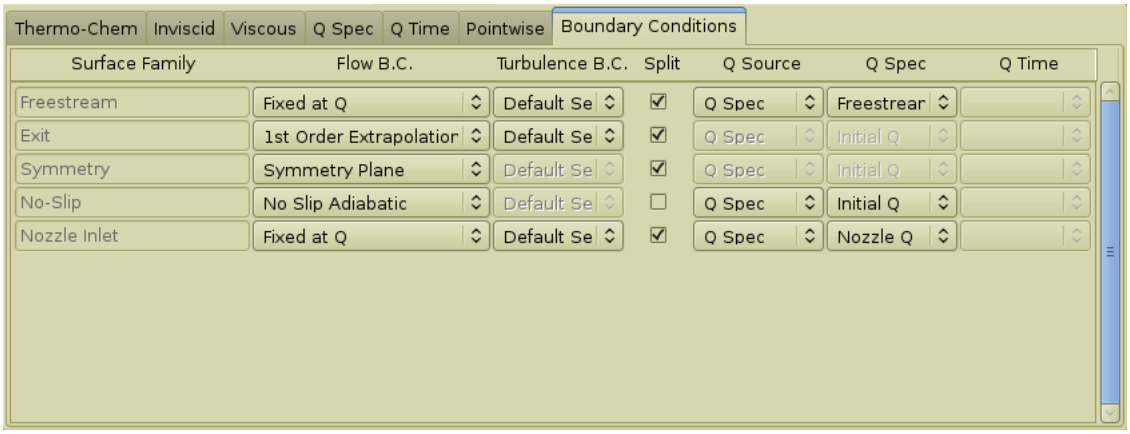

Figure 11: The boundary conditions display for the rocket problem.

reference state to compute values at the boundary.

For example, the "`Fixed at Q`" boundary condition will set all the boundary values based on a reference state. For the rocket problem, there are two surface families that use the "`Fixed at Q`" boundary condition; the "`Freestream`" surface and the "`Nozzle Inlet`" surface.

The Freestream surfaces are set to have the "`Freestream Q`" specification. The Nozzle Inlet surfaces will use the "`Nozzle Q`" specification. The boundary conditions should look like Fig. 11 on pg. 21.

1.2.14 Zone Initialization

We have just completed setting up the physical model. Recall that there were three q specifications associated with the physical model: Initial Q, Freestream Q, and Nozzle Q. The Freestream Q and the Nozzle Q were used in the boundary conditions, but we have yet to mention where the Initial Q is used.

Each zone must have a q specification associated with it for initialization purposes. This association is made inside the Zones frame. Go there now and select the Initialization tab. Remember that the Initialization tab in the Zones frame operates on a zone folder. If a folder or zone is not selected in the tree view, the initialization table will list all the zones.

You should now see a table listing the eight zones (if not then select the Zones folder in the tree view). Here the user can specify the physical model and q specification with which to initialize the solution. Since the "`Initial Q`" specification was created first, it should already be selected in the Q Spec column. Verify that this is correct.

1.2.15 Setting up the Run Definition

At this point, we have completed most of the tasks required to run the problem. In summary, the grid has been imported, coarse and medium grids were created, the surfaces of the grid

were placed into families, and the physical model was set up. The remaining task to be completed before running the rocket is specifying how to run the simulation. This involves telling $\mathcal{GASP}$ what type of iteration scheme to use, the CFL number to use, how many cycles to perform, *etc.*. These parameters are set inside the Run Definitions frame, which can be viewed by selecting the Run Definitions tab located under the graphics window.

The user is free to create as many run definitions as desired. For the current case, there are three grid sequences and a run definition will be created for each sequence. We will start with setting up the run definition for the coarse grid. After executing the run definition, we will view the solution and add a run definition for the medium grid sequence. This will allow us to run the problem in stages, giving us a chance to view the solution before continuing to the next sequence level.

Within the Run Definitions frame the information is divided into a number of sections (or tabs). These will be discussed next.

1.2.16 Main Parameters for the Run Definition

The Main tab in the Run Definitions frame allows the user to select the sequence level, solution backup information, and the convergence information. At this point, the user would also select whether the problem is to be run as time accurate or as steady state.

The first option is the current sequence to compute. Three sequences were created for this problem with the following names: "`Fine Grid`", "`Medium Grid`", and "`Coarse Grid`". We will start the computation on the coarse grid, so select "`Coarse Grid`" from the Sequence option menu.

The remaining information under the "Run Settings" heading allow the user to control the execution of the current run definition, solution initialization, and solution interpolation. The Execute This Run Definition should be on by default. The Re-Initialize Solution option should only be used to reset the solution back to the initial values, and would normally be used to restart the problem. The Interp Solution Up is used when the current sequence is not the fine grid. For this run definition, having the Interp Solution Up option turned on will cause the solution on the coarse grid to be interpolated to the medium grid at the end of executing the run definition. This is the desired behavior for this case.

The problem will be run to a steady state, so the default setting for the temporal accuracy is appropriate. Located under the Temporal Info selection are parameters describing time dependent calculations. Since we are only concerned with the steady-state solution and have selected this for the temporal accuracy, the time-dependent variables will not be used.

For the convergence information, we will have to make a few changes. Set up the problem so that the maximum number of cycles to run is 1000, the relative stopping convergence is 1.0e-10, and the absolute stopping convergence is 1.0e-12. The convergence settings will allow 1000 cycles to be performed unless a relative tolerance of 1.0e-10 or an absolute tolerance of 1.0e-12 is first met.

It is always a good idea to save the solution to file during a run. This is specified using the Write Restart Every input. Set this value to 250, which means the solution is written to the

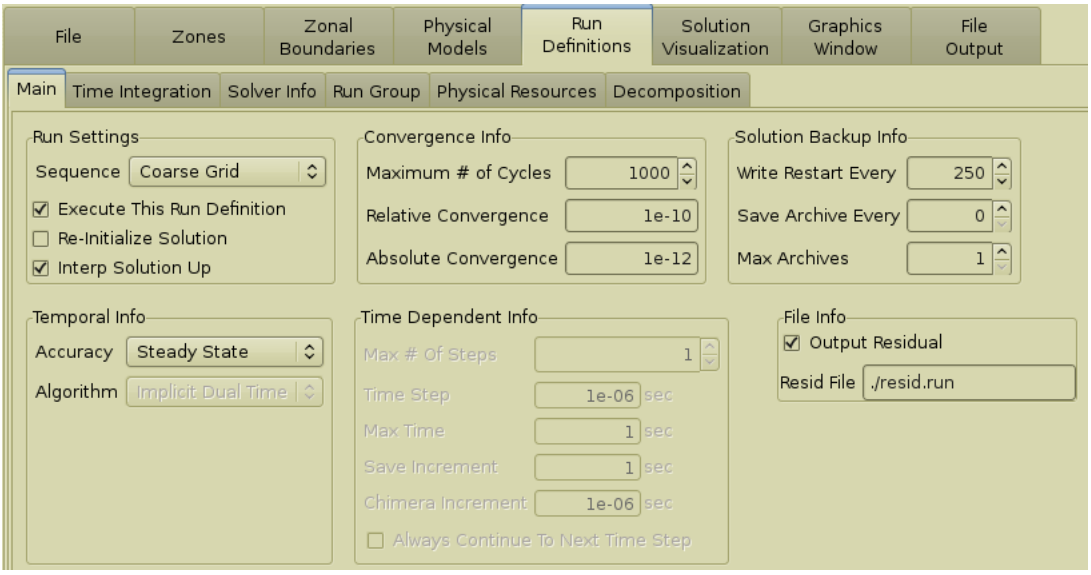

Figure 12: The Main tab under the Run Definitions frame. The run is set up to run the coarse grid sequence.

`.sln` file every 250 cycles. A separate solution file can be saved as an archive if desired. This is done using the `Save Archive Every` input which is not used for the rocket simulation.

The last option for this tab is the residual output. While the flow solver is running, residual information may be printed to a single file. The user controls this with the Output Residual option. The default setting is to print residual information to the `resid.run` file. This will print the global residual value computed using a root-mean-square method. The Main tab will appear as shown in Fig. 12 on pg. 23.

1.2.17 Time Integration for the Run Definition

The Time Integration tab allows the user to specify which time integration scheme to use, the CFL or physical time step, and the time-step type.

For the rocket, the "`Gauss Seidel`" time-integration scheme will be used. Select this option from the Solution Scheme option menu. The Gauss Seidel scheme approximates the solution to the linearized system using a symmetric block Gauss Seidel algorithm. In most cases performing a single Gauss Seidel iteration does not provide an adequate solution to the linearized system. In order to improve the solution as well as numerical stability, additional iterations are specified under the Inner Iteration Scheme label. For this example, the default value of five inner iterations is used.

For most CFD simulations, it is desirable to use a non-dimensional time step (CFL) value of 1 or higher. For the rocket case, starting with a CFL value of 1 will cause instabilities in the solution since the solution is changing dramatically at first. The user therefore has two options for this case.

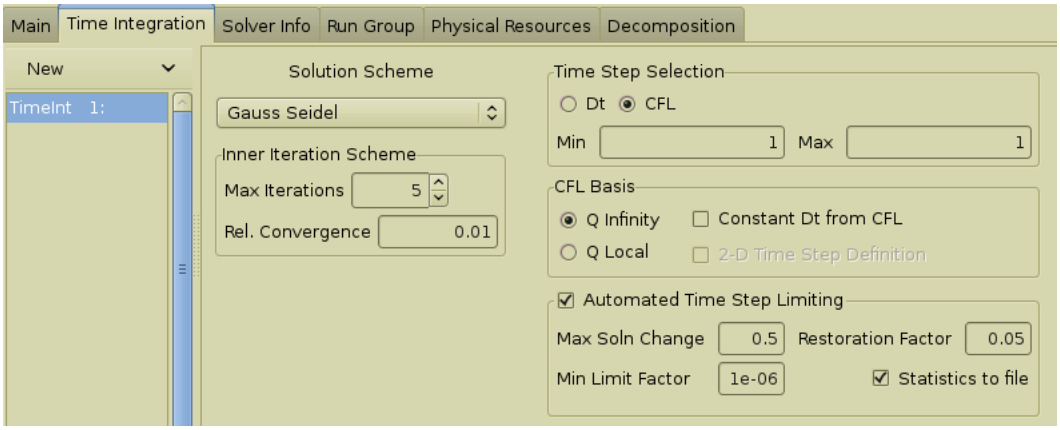

Figure 13: The time integration parameters for the rocket after setting up to run the coarse grid.

Option 1: A freestream based CFL number can be used with a starting value of 0.1 and an ending value of 1. This is specified by selecting the CFL radio button, the Q Infinity radio button, and entering values of "`0.1`" for Min and "`1.0`" for Max. With these settings, the CFL will have a linear ramp starting at 0.1 and ending at 1.0 over the number of cycles specified in the Main tab. In this way a smaller time step will be used at first, allow the solution to develop more before raising the time step.

Option 2: A freestream based CFL number can be used with a constant value of 1, along with automated time step limiting to assist in convergence stability. This is specified by selecting the CFL radio button, the Q Infinity radio button, and entering values of "`1`" for Min and "`1`" for Max. The Automated Time Step Limiting flag should be turned on, which lowers the CFL value at a cell if the solution tries to change more than the Max Soln Change value, which is given as a decimal percentile (0.5 means 50% change). The value Min Limit Factor is the maximum amount the time step will be lowered and the Restoration Factor is how fast the time step is returned to normal once the limiting is no longer required. The user may track the limiting activity by selecting the Statistics to file option which prints information to the file `dtlimit.dat`.

The second option is used for the current tutorial and the time integration should now look like Fig. 13 on pg. 24.

1.2.18 Solver Info for the Run Definition

The Solver Info tab allows the user to specify specific options for the solver being used. For the current rocket case, the solver info corresponds to solving the Navier-Stokes equations, which consist of conservation of mass, momentum and energy. Additional equations such as those used for turbulence may also be present.

When turbulence equations are used, $\mathcal{GASP}$ has the ability to solve the equation set as fully or loosely coupled. Running uncoupled turbulence reduces the overall size of the linear system that is solved, thus reducing the required memory and computational time per cycle. The use of uncoupled turbulence is recommended when possible. The default settings specify loosely coupled turbulence and fully coupled chemistry, which are suitable for the current tutorial.

Additional inputs are required when using space marching (not being used in this tutorial) and global limiting. The global limiters allow the user to put absolute min and max limits on the pressure and temperature. This is sometimes useful to keep the solution stable due to poor grid quality or during transient runs.

1.2.19 Run Group for the Run Definition

A Run Group is characterized by a set of zones, a physical model, and a solver info. By default, a single run group exists with the "`Zones`" folder selected. The physical model and solver info are selected using option menus, which will allow the user to select a single entry from all of the possible choices. For the current tutorial, a single run group will be used which contains all the zones and the previously defined physical model and solver info.

Recall that in the time integration section a CFL was used which was based on "`Q Infinity`". This means that a global velocity is used to compute the time step from the CFL value. The velocity is taken from the "`Current Q Spec`" option menu. The freestream value is suitable for this case so set the Q Spec to "`Freestream Q`".

If multiple run groups were created, the user has the ability to set the cycle frequency (how often a run group is peformed). When only a single run group exists, the frequency should be set to 1.

Similar to the global residual file, the run group residual can be printed to file. For the run group, the residual information is printed for each group of equations being peformed (*e.g.,*mass, momentum, energy, turbulence, *etc.*).

The implicit zonal boundary flag allows the user to run implicit or explicit zonal boundaries across point-to-point zonal boundaries. For the rocket, we will run implicit point-to-point zonal boundaries. Implicit zonal boundaries are recommended unless memory is an issue. Choosing between running implicit and explicit across partition boundaries is handled in the Decomposition tab.

The run group should now be set up to run on all eight zones. The Run Group interface should appear as in Fig. 14 on pg. 26.

1.2.20 Physical Resources for the Run Definition

The Physical Resources tab for the Run Definitions frame is used to specify computational resources used in the current run. It is here that the user will specify how many processors will be used to run the problem, relative speeds of the different processors, as well as physical

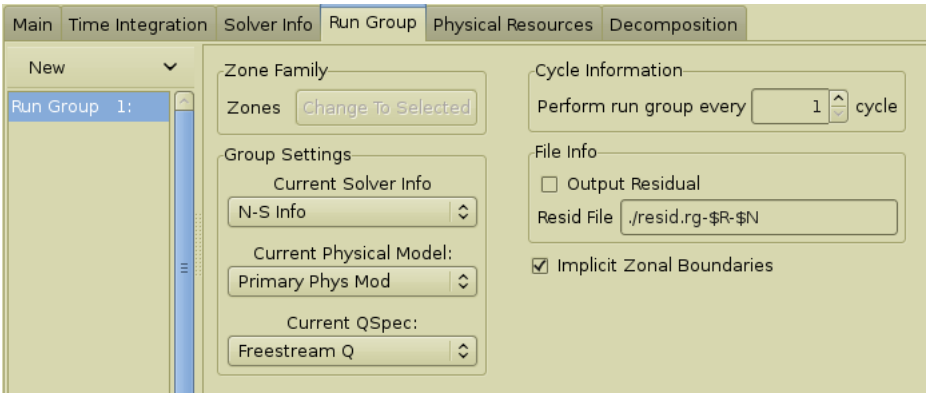

Figure 14: The run group parameters for the rocket after setting up to run the coarse grid.

memory available for each processor. Set the number of processors by adjusting nCPU in the last column.

Another important piece of information specified here is the memory available for each processor. This is given in megabytes and is used in the auto decomposition feature (discussed in the next section). If you are using processors with distributed memory, then you would give the available memory for each processor. For a shared memory platform, divide the total memory available by the number of processors. Fill in the appropriate information for nCPU and Mem/CPU (MB) based upon your system.

For the coarse sequence, we will perform a single processor run.

1.2.21 Zone Decomposition

A zone folder was selected in the Run Group tab. The zones in this folder (all eight in our case) will be computed on each run cycle. Under the Decomposition tab, the eight zones for the rocket will appear in table format. This tab allows the user to perform domain decomposition, which is the process of dividing a zone up into two or more partitions.

One of the reasons to do zone decomposition in $\mathcal{GASP}$ is to run a problem efficiently in parallel. The ideal situation is to have each zone be the same size such that each processor takes the same amount of CPU time to compute it's share of the problem. The Auto Decomp button is designed to set up the decomposition to give proper load balancing for parallel runs. Auto decomp will set the partitions (in I, J, and K) necessary to meet the target efficiency specified by the 'Target' input. The actual efficiency is based on how many processors will be used and the decomposition set up. The default number of processors is one, so the actual efficiency will be 1 (or 100%) initially. Because the actual efficiency exceeds the target, performing Auto Decomp will suggest that no new partitions be made.

Another reason to perform zone decomposition is for memory efficiency. By breaking up larger zones into smaller ones and running explicit partition boundaries, the memory required to run a problem can be drastically reduced. The auto decomp feature will also take memory requirements into consideration when suggesting a decomposition.

To actually make new partitions (to decompose a zone), you must press the Compute Decomp button. Pressing the Auto Decomp button will only set up (or prepare) the zones for decomposition.

If you are running the problem on a single processor, zone decomposition may not be necessary. If you are running on multiple processors, press the Auto Decomp button followed by the Compute Decomp button.

Even if you are not going to run with multiple processors, you may set up the problem as if you were and then select "Recomp" to undo the zone decomposition. This will allow you to get a feel for zone decomposition.

1.2.22 Save And Exit

The problem has now been set up to run 1000 cycles on the coarse grid. The input deck can now be saved and the *GASP* GUI can be exited. After running on the coarse grid, we will run the GUI again and view the solution.

1.3 Running the Coarse Grid

Assuming that the `AEROSOFT_HOME` environment variable is defined correctly, and the *GASP* binaries have been added to the `PATH` environment, the following command will execute the flow solver for the rocket problem.

```
gasp --solve -i rocket.gsp
```

This assumes that the name of the input deck is `rocket.gsp` and that the processor information is set up correctly inside the input deck. The flow solver will run 1000 cycles on the coarse grid, and converge five orders of magnitude on the residual. The convergence information is printed to the file `resid.run`. The coarse grid will take less than 30 MB of memory to run using a single processor.

The solution on the coarse grid can be viewed using the *GASP* GUI. Additional output can be performed within the GUI and is referred to as post processing. Run the GUI again and open the input deck using the File→Open... menu option. You can also type

```
gasp -i rocket.gsp
```

in order to run the GUI, and immediately open the solution file referenced by `rocket.gsp`.

1.3.1 Creating On Screen Visualizations

To visualize the current solution, flooded contours representing the magnitude of the Mach number will be created for the symmetry plane.

With the introduction of Version 5, *GASP* began using the open source Visualization ToolKit (VTK) library as the foundation for all three-dimensional visualization within the

GUI. With the switch to this library, came a need to re-organize the $\mathcal{GASP}$ input specification of post processing to adapt to the VTK pipeline oriented model. While this pipeline model may be the source of some initial confusion, once the user is familiar with the techniques, complicated visualizations can be easily achieved. A more comprehensive tutorial focusing upon the visualization techniques is presented in the visualization tutorial included as a separate exercise. For the purpose of the current tutorial, the surface visualization found in previous versions of $\mathcal{GASP}$ will be re-created using a very simple, one-step VTK pipeline.

The Visualization Data Group

The "`Solution Visualization`" folder is empty by default. There are two ways to create entries in the Solution Visualization folder: by drag-n-drop creation linking existing zones and surfaces to visualization data, or by using the pop-up menu to create pipeline elements such as isosurfaces, cutting planes, or stream traces. In order to begin the visualization process, the user will need to select the physical domain to be viewed. To do this, find and select the "`Symmetry`" folder, located within the "`Surfaces`" folder of the tree view. Next, drop the selected "`Symmetry`" folder into the "`Solution Visualization`" folder, found beneath the "`Flow Solver`" folder. With this action, the user will create a VTK data source which is based upon the surfaces found in the "`Symmetry`" folder.

The Visualization Data Folder is a container of Visualization Data Nodes which can be taken as data associated with a zones or surface. The user controls which variables are computed for the Visualization Data Node, and the grid range of the zone as well as to where the data is interpolated.

Open the "`Symmetry`" Visualization Data Folder, and you will find two children, "`Input`" and "`Output`". If you open these folders you will see that "`Input`" contains fourteen children, and "`Output`" is empty at this time, as seen in Fig. 15 on pg. 29. The "`Input`" folder contains all elements in the VTK pipeline which feed the current visualization element. In the case of the Visualization Data Folder, the filter is fed by Visualization Data Nodes which are based upon the surface nodes found in the "`Symmetry`" surface folder . The "`Output`" folder contains links to all elements in the pipeline which are fed by the current element. Because the current pipeline ends with this Visualization Data Folder, the "`Output`" folder is empty.

The data group inherits many of the properties of the creating folder, which in this case is the surface folder, "`Symmetry`". This includes render modes, colors, and line thicknesses. Because the "`Surface`" folder may not be displayed graphically when you first create the visualization, you may need to change the render mode for the top level folder that you just created in the "`Solution Visualization`" folder. To do this, select the text for the folder you just created and change the "`Render Still`" mode from "`None`" to "`Full`".

Basic Visualization Properties

Now select the Solution Visualization tab located beneath the graphics window. Within this frame you will find a variable set of tabs. The tabs that are presented in this frame depend upon the particular visualization element that you have selected. Make sure that the

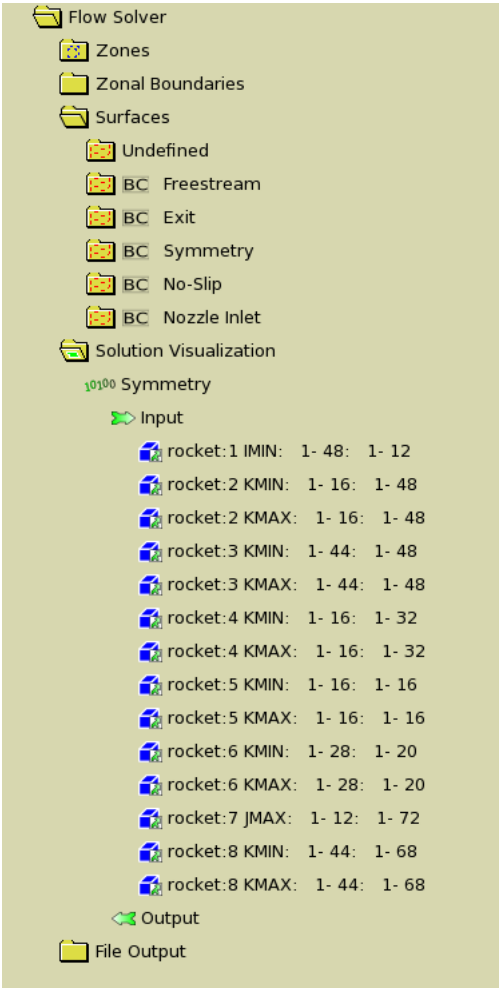

Figure 15: The "`Symmetry`" Visualization Data Folder and its children.

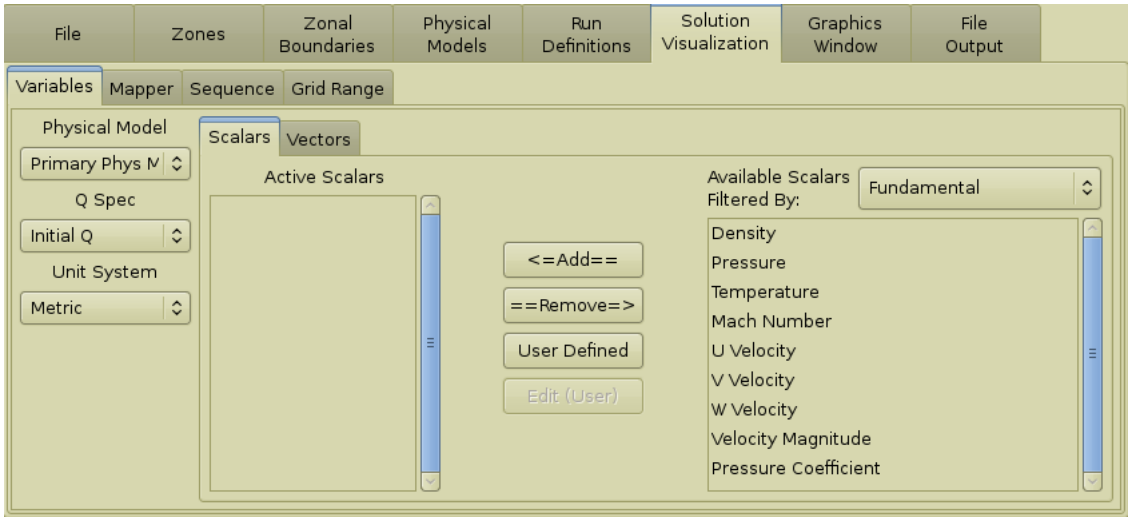

Figure 16: The **Solution Visualization** frame.

Symmetry Visualization Data Folder is selected, and you will see the **Solution Visualization** frame as depicted in Fig. 16 on pg. 30.

When you are editing a Visualization Data Folder you will be able to

1. control which variables are computed and available for display (**Variables** tab),
2. control the mapping of a selected scalar to a color ramp (**Mapper** tab),
3. set the sequence to which the current data refers (**Sequence** tab, and
4. adjust the limits on the grid indices, as well as the interpolation type (**Grid Range** tab).

With the **Variables** tab selected, the user is presented with a list of available variables which can be computed at each grid point or control volume. Because of the large quantity of available variables, this list is filtered by variable type. In order to actually compute a variable, it must be selected from the **Available Variables** list, and added to the **Active Variables** list by pressing the **Add** button. For the current visualization, find and select the variable **Mach Number**, and press the **Add** button. The user will notice the the boundary of the domain is immediately shaded with a color map scaled to the Mach number.

Select the **Mapper** tab in order to edit properties associated with mapping the selected scalar to the displayed geometry. Because there is only one computed scalar quantity for the current problem, **Mach Number**, this value is set by default to the **Current Scalar** option menu. The **Display Scalar** check box can be toggled to turn off and on the mapping of the color map of the current scalar to the current geometry. By default, this value is set to true. Check the **Display** box under the **Color Bar Legend** frame in order to position a legend mapping the color ramp to scalar values for the current scalar. The user can move and resize

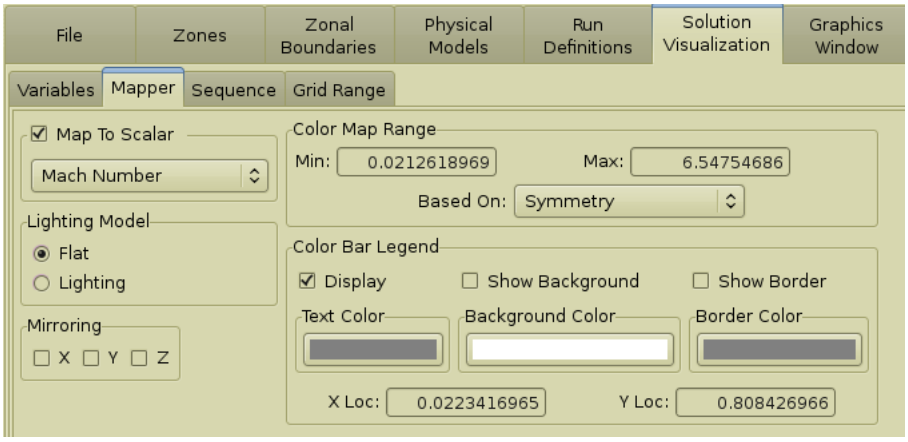

Figure 17: The Mapper frame.

the scalar bar interactively using the left mouse button. The limits on the scalar bar can be manually set, or automatically scaled to the limits of the current data. By default, the limits are set based upon the minimum and maximum values found in the current data set.

For three-dimensional objects, a lighting model is applied to all surfaces regardless of whether the color map is being applied. Because the lighting model will vary the color on a surface based upon the surfaces orientation to lights, the colors shown on a surface may not directly correspond to the color ramp. For the current problem, the symmetry plane colors may appear dull because the symmetry plane is not perpendicular to the light source. In order to render the color map of flat surface without the lighting model, select the "`Flat`" option under the Lighting Model frame. The Mapper property frame will be similar to that depicted in Fig. 17 on pg. 31.

Select the Sequence tab. You will see the frame as depicted in Fig. 18 on pg. 32. Output can either be uniquely tied to a single sequence level or it can simply be associated with the current sequence level. With the latter representation, it is possible to examine data for a given location in physical space for all sequences using a single output element. The default behavior is to associate the element to the currently selected sequence. This behavior is controlled in the sequence frame through the Set To Current check box. When the Set To Current flag is enabled, it will not be possible to control the sequence level associated with the current visualization element. When un-checked, the Fixed To option menu allows the user to select the sequence level with which the current element is associated.

Select the Grid Range tab. The grid range properties are meant to limit the display of a given zone's computational domain. Selecting the parent folder will display grid range properties for all nodes within that folder, while selecting a single node will display only grid range values for that node.

The Range Type will limit ranges of the computational domain to a given volume, surface, line, or point. For viewing the duct solution inside of *GASP*, looking at a k surface will be appropriate. Select the K Surface option for the range type. All surfaces except for the k=*1* surface will disappear from the graphics window.

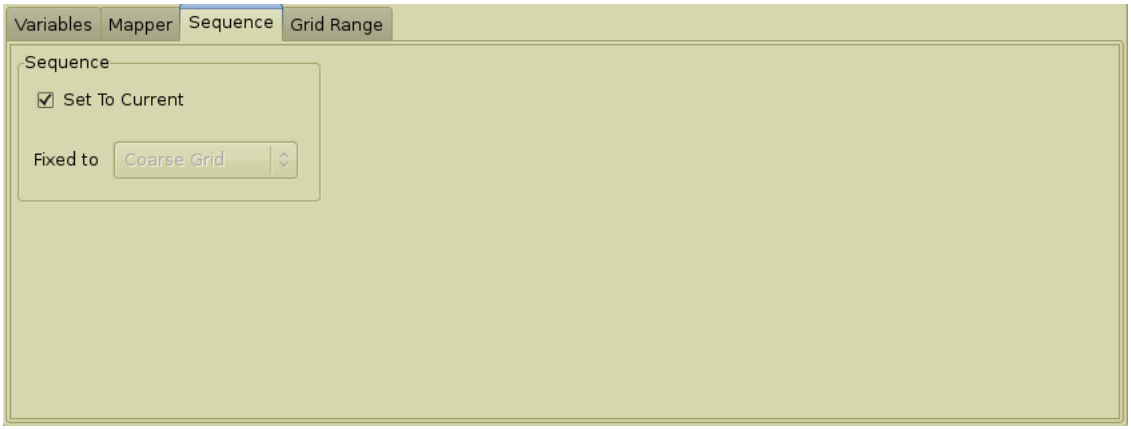

Figure 18: The Sequence frame.

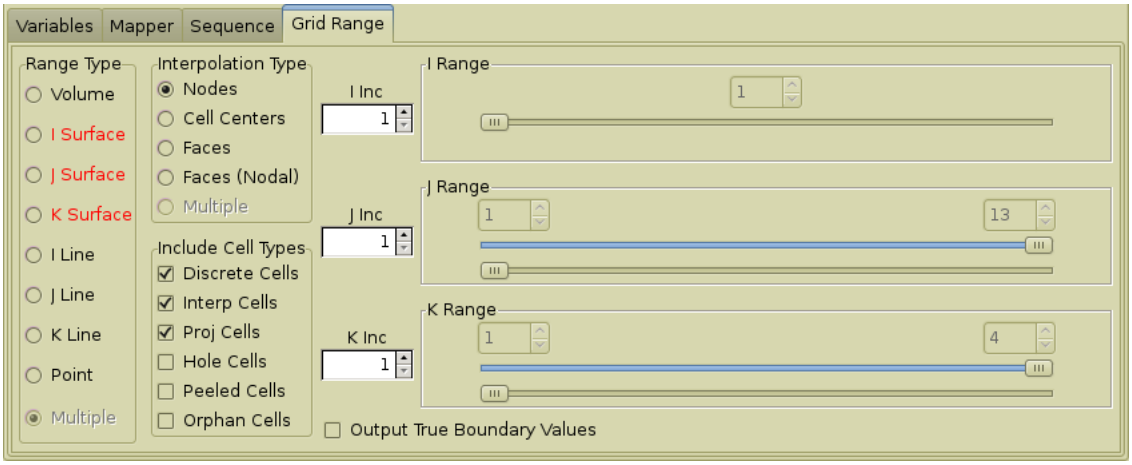

Figure 19: The Grid Range tab with the proper settings for displaying the "`Symmetry`" boundary surfaces interpolated to nodes on the coarse sequence.

The interpolation type is set to nodes. $\mathcal{GASP}$ stores the solution at cell centers, so viewing cell center values will reflect the true solution. Viewing the solution at nodes or faces will require interpolation from the cell centers. This will introduce some error into the output due to the interpolation. However, interpolation to nodes will provide more accurate geometric representation.

For cell centers, the grid ranges for i and j start at a 0 index value. A cell center index of 0 indicates a boundary value. Boundary values may or may not be interpolated depending on the boundary condition. Some boundary conditions store the boundary value at the boundary face, such as the no-slip condition, while others store the boundary values at ghost cell locations. For the case where ghost cells are being used to store values, the boundary values are interpolated to the boundary face. The grid range information should appear as shown in Fig. 19 on pg. 32.

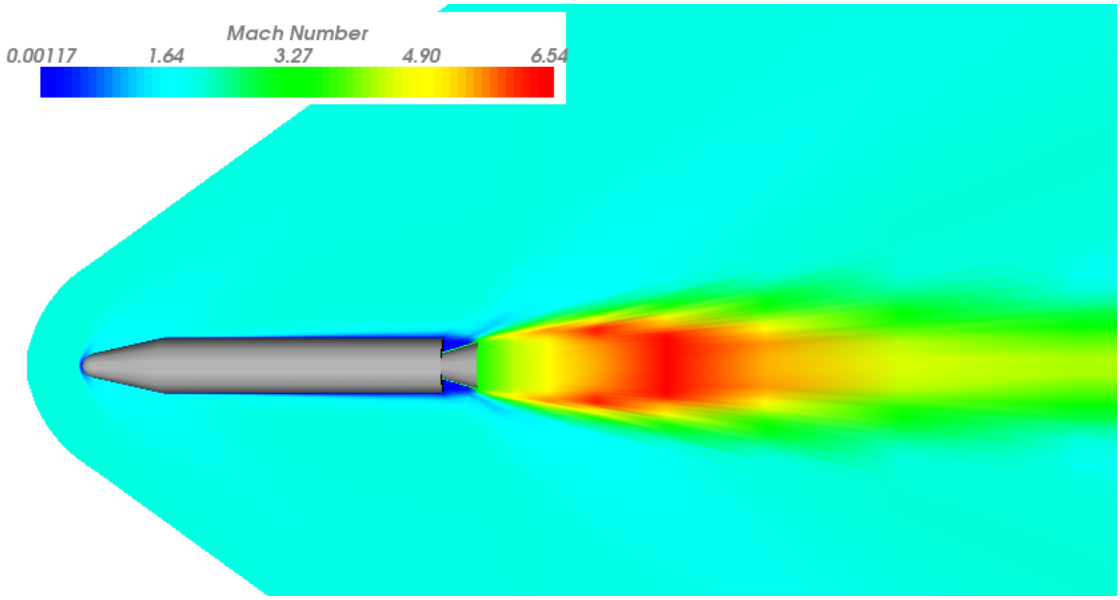

Figure 20: Flooded contours of Mach Number on the coarse sequence of the Rocket Tutorial.

Rendering the Rocket Surface

In order complete the current visualization, we wish to render the surface of the Rocket as a steel-gray solid. To do this, select the "`No-Slip`" surface folder from the "`Surfaces`" folder, and drop the selected surface into the "`Solution Visualization`" folder. Next, select the newly created Visualization Data Folder and change the Render Still mode to "`Full`". Also find the Colors... button in the Properties frame, and open the color selection dialog for the selected folder. Change the "`Line`" color to light gray.

The solution on the coarse grid should now appear in the graphics window similar to Fig. 20 on pg. 33.

1.4 Exporting the Solution for External Visualization

Now that the Mach number contours have been viewed, directions for creating PLOT3D grid and solution files are presented. Remember that with the introduction of Version 5, there is a distinction between exporting data for external use, and internal visualization. Exporting data will require the creation of Output Data Folder entries within the "`File Output`" folder in the tree view.

1.4.1 The Output Data Folder

The "`File Output`" folder is empty by default. The only way to create entries in the output section is by drag-n-drop creation linking existing zones and surfaces to output data. In order to create an output entity, find and select the "`Zones`" folder, located within the

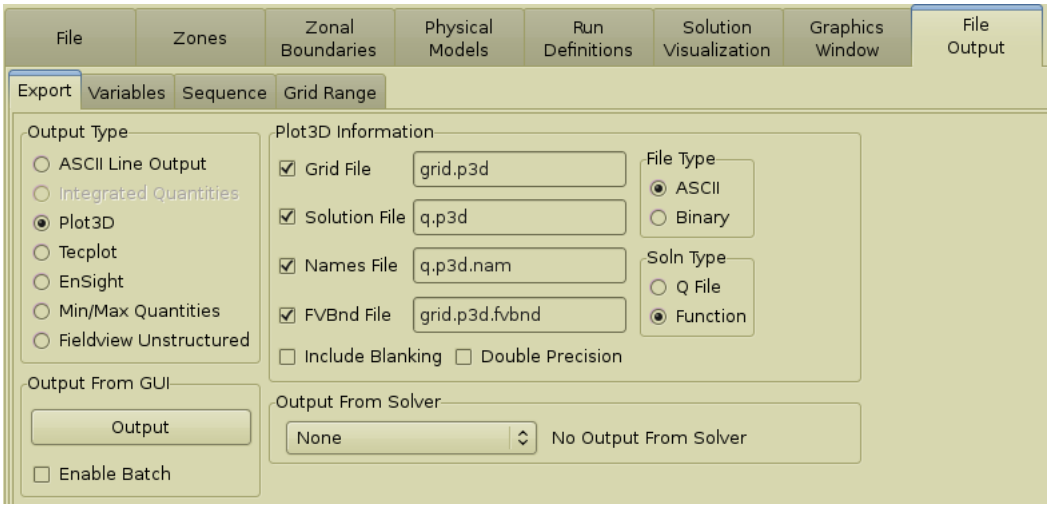

Figure 21: The Export frame with the proper settings for output Plot3D formatted files.

"`Flow Solver`" folder of the tree view. Next, drop the selected "`Zones`" folder into the "`File Output`" folder, also found within the "`Flow Solver`" folder. With this action, the user will create Output Data Folder which is based upon the zones found in the "`Zones`" folder. In this example, this consists of the eight structured zones describing the Rocket flow domain.

Open the "`Zones`" Output Data Folder, and you will find eight elements, which are outputDataLinkNodes, one for each zone of the problem. To better organize the output data, you can re-name the "`Zones`" Output Data Folder to "`Plot3D`".

With the Plot3D output folder selected, select the File Output tab found beneath the graphics window. Within this frame you will find a notebook with four entries: Export, Variables, Sequence, and Grid Range.

1.4.2 The Export Frame

Make sure that the Export tab is current. From this frame, select the Plot3D option from the Output Type. The default information is set up for an ASCII file, with a function file for the solution (versus a q file). Since the conservative variables are not being printed, the function file option is appropriate. The names of the files are shown as well, allowing the user to change the location of the output as desired. Blanking should only be included when an overset grid has been created. Pressing the Output button will cause the GUI to create the desired files. However, do not perform this operation until the Output Data Folder is completely defined. The Export frame will look similar to Fig. 21 on pg. 34.

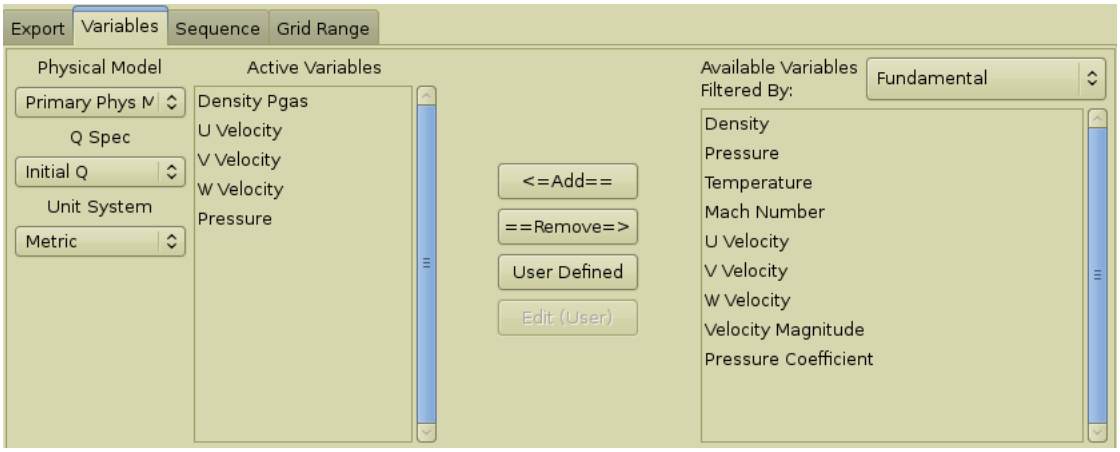

Figure 22: The Variables frame under the File Output tab with the proper settings for outputting primitive variables.

1.4.3 The Variables Frame

The layout and function of the Variables frame found under the File Output tab is similar to that found under the Solution Visualization tab. Within the context of the Output Data Folder, all solution quantities found under the Active Variables list will be output to the files specified under the Export section. For the purpose of this example we will add the density, the three components of velocity, u, v, and w, and the pressure. These primitive variables are found under the filter heading Primitive found in the Available Variables Filtered By: option menu. Once the primitive variables have been added to the Active Variables list, the Variables frame will look similar to Fig. 22 on pg. 35.

1.4.4 The Sequence Frame

The layout and function of the Sequence frame found under the File Output tab is identical to that found under the Solution Visualization tab. The default behavior for file output is the same as that for solution visualization in that the currently selected sequence will be used.

1.4.5 The Grid Range Frame

Select the Grid Range tab. The layout and function of the Grid Range frame found under the File Output tab is identical to that found under the Solution Visualization tab. For the current example, change the Interpolation Type to Nodes. The I Range, J Range, and K Range limits will automatically adjust to reflect the node based limits.

1.4.6 Create the External Output Files

As discussed in a previous section, now that the output elements have been completely defined, the user can perform the actual export operation. Return to the Export frame and press the Output button. At this time, the GUI will create the necessary plot3d files for the currently selected coarse grid.

1.5 Running the Medium Grid

At this stage, a solution has been obtained on the coarse grid, and the solution has been viewed and verified. At the end of each run definition, $\mathcal{GASP}$ will automatically interpolate the solution up one sequence level. So now that the coarse grid has been completed, the solution has been interpolated to the medium grid level. Note that the fine grid solution has not been changed since the solution is only interpolated up one sequence level. The interpolation process was done after the last cycle was performed for the coarse grid run.

Now we must set up the input deck to run the next sequence level. A new run definition for the medium sequence needs to be created from the coarse grid run definition. To do this, first copy the coarse grid run definition, and then paste it. Now we can easily make a few minor modifications to create the new run definition. Go to the Main tab under the Run Definitions frame (select the Run Definitions button). and select the "`Medium Grid`" from the "`Sequence`" list. The run will now perform the computation on the medium grid.

Since the medium grid needs to be converged further than the coarse grid, the number of cycles should be changed. Set the number of cycles to be 2000 for the medium grid run. Also, from the Time Integration tab set the time step to "`Q Local`".

If you wish to run the problem in parallel, you will need to set up the zone decomposition information again. Recall that this is done under the Decomposition tab.

Since we have already run the coarse grid, we want to make sure the run definition for the coarse grid is turned off. Select "`Run 1:`" (the coarse grid run) from the Run list, and go to the Main tab. Turn off the Execute This Run Definition button. Now we are ready to run the medium sequence only, so we can save the input deck and exit.

The flow solver is executed the same for the medium grid run as was done for the coarse grid. Type

```
gasp --solve -i rocket.gsp
```

from the command line to run using a single processor. $\mathcal{GASP}$ will require less than 100 MB of memory for the medium grid.

1.6 Running the Fine Grid

If you have a solution on the medium grid, then you are just one step away from completing this tutorial. The remaining task is to set up the run for the fine grid and run the flow solver one last time.

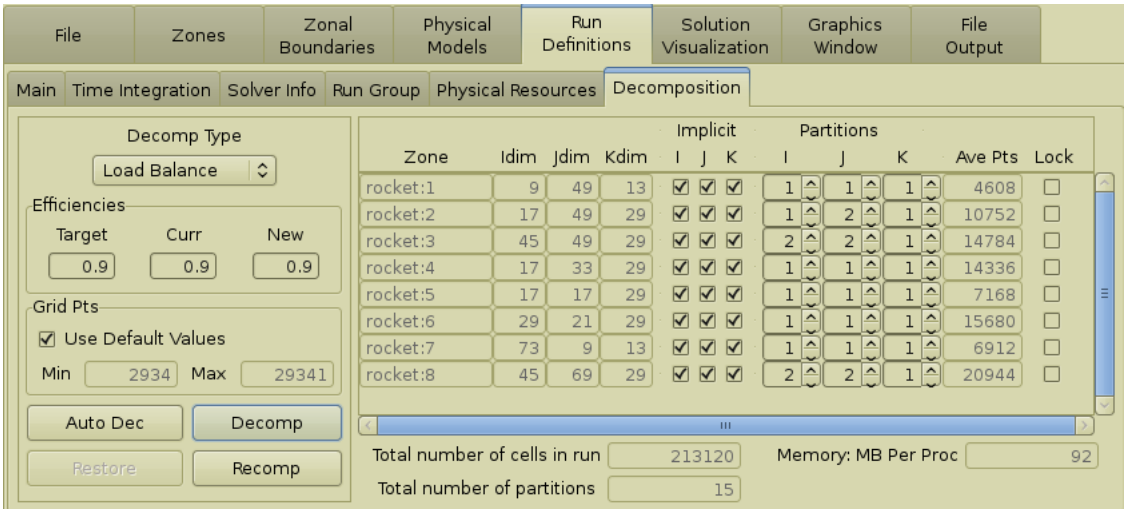

Figure 23: The zone decomposition settings for the fine grid on 8 processors. The memory is limited to 500 MB per processor.

As a default setting, *GASP* will interpolate the solution up one sequence level at the end of executing a run definition. So if you have completed running the medium grid sequence, *GASP* will have interpolated the medium grid solution up one sequence level to the fine grid.

We will now set up the input deck for the fine grid run. Start up the *GASP* GUI and open the input deck as before. Copy and paste the medium sequence run definition. Go to the Run Definitions frame and change the sequence selection to the fine grid. The solution will converge in about 2000 cycles, so leave this as the cycle number.

The fine grid is eight times larger than the medium grid, and will require additional memory if zone decomposition is not exercised. Therefore, we will set up the fine grid to run on eight processors and use no more than 500 MB of memory per processor. Go to the Physical Resources tab and set the memory to 500 and the number of processors to 8. Now go to the Decomposition tab. Select the Auto Decomp and then Compute Decomp to complete the set up. The decomposition settings should be similar to Fig. 23 on pg. 37.

Since we have already run the medium grid, we want to make sure the run definition for the medium grid is turned off. Select "`Run 2:`" (the medium grid run) from the Run list, and go to the Main tab. Turn off the Execute This Run Definition button. Now we are ready to run the fine sequence only, so we can save the input deck and exit. Run the flow solver once again following the same procedure as before. When the run is complete, try viewing the solution like we did for the coarse grid.

For your convenience, a completed input deck and solution can be found in the `Finished` directory. This directory may not come with the standard download, but may be download from AeroSoft's ftp site (gasp5@ftp.aerosoftinc.com).