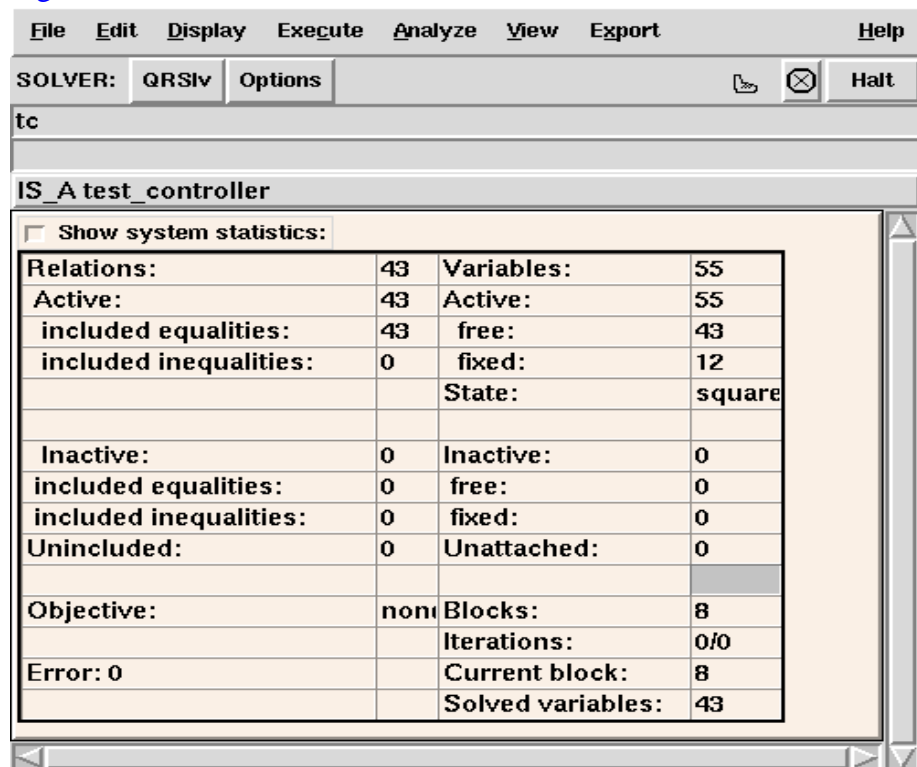


# CHAPTER 6 SOLVER

The purpose of the [Solver Utility](#) shown in [Figure 6-1](#) is to provide support for the numerical solving and debugging of an imported instance. To this end the Solver allows the user to access numerical solvers and analysis functions and displays statistical and status information for the problem being solved. The upper section of the solver window contains a menu bar; buttons for selecting numerical solvers, solver options, and halting the solver; a label containing the name of the current instance (or problem being solved); and a label containing the type of the current instance. The remainder of the Solver window is devoted to providing statistics about the problems relations and variables along with a description of the problem's state.

Figure 6-1 Solver Window



## 6.1 THE SOLVER MENU BAR

The menu bar on the Solver window has eight entries: File, Edit, Display, Execute, Analyze, View, Export, and Help.

### 6.1.1 SOLVER FILE MENU

- Close Window** Close the Solver window. To open it again, go to the Script window and select the Solvers tool under the Tools menu or select Solver on the Toolbox.
- Exit ASCEND** Exit the ASCEND system completely. You will be asked to verify that you really do want to exit.

### 6.1.2 SOLVER EDIT MENU

- Remove instance** Removes the current instance from the solver.
- Select objective** Provides a list of objectives from which one may select. The selected objective will be used in any subsequent optimizations until another objective is selected.

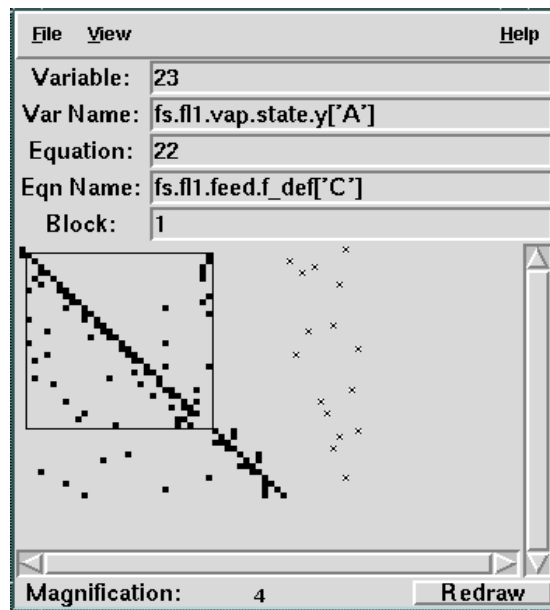
### 6.1.3 SOLVER DISPLAY MENU

- Status** Shows the internal status of the Solver along with the largest block scaled residual vector two-norm.
- OK**  
**converged**  
**max. block error (1)**  
**1.626725280789307e-13.**  
**Dismiss**
- Unattached variables** Shows variables not incident in any of the relations in the current system being solved. (These variables are not a part of the current problem. Changing their attributes will have no effect on the problem.)
- Unincluded relations** Shows relations not in the current system being solved.
- Incidence matrix** Incidence matrix shows the incidence of variables in relations (See Figure 6-2). Clicking mouse-1 (left button) on the matrix displays the names and numbers of the relation/variable at that coordinate, whether that coordinate is occupied or not. A box is drawn around the partitioned block containing the selected coordinate and the block number is displayed. The selected block or the entire incidence matrix

may be printed by selecting the PrintBlock or the Print button respectively. The scale of the incidence matrix can be changed by sliding the magnification bar and depressing the Redraw button. Depressing the OK button will close the INCIDENCE window.

Drawing large dense matrices may take a while. Drawing matrices on problems bigger than about 1000x1000 may be prohibitively expensive on slow machines. The row/column ordering is that done by the selected solver, except that fixed vars and unincluded relations are moved to the edges.

Figure 6-2 The Incidence Matrix



## 6.1.4 SOLVER EXECUTE MENU

### **Solve**

Solve the current problem as an algebraic or optimization problem depending on what solver is selected

### **Single step**

Perform a single iteration of the system with the solver in question. In some solvers (e.g. MINOS) there is no iteration mode. For these selecting single step will result in a full solve attempt.

For QRSIV, an iteration will be a Newton like step if there are many variables in the current block or if the current block is a blackbox singleton. Singletons not from blackboxes are numerically inverted when being solved.

**Integrate** Invoke the selected integrator (LSODE currently available) on the problem.

### 6.1.5 SOLVER ANALYZE MENU

**Reanalyze** Reanalyzes the current problem to determine if a proper set of variables has been fixed to make the problem well-posed (i.e., square).

**Debugger** Opens a tool which deals with the system as a numbered list of variables and relations. See Section 6.4, "Debugger," on page 69 for more information about the Debugger.

**Overspecified** Finds and displays the variables that can be freed to reduce the degrees of freedom in an overspecified system.

**Find dependent eqns.** Finds structural or numeric dependencies of a system.

**Numeric Dependency.** Doesn't mean much on an unsolved system. This command inverts one block at a time and checks the blocks for numeric dependency using the QRSlv solver. Any non-zero dependency is reported, but those relations with coefficients down around machine epsilon (1e-16) are probably not dependent. Poorly scaled problems can appear more singular than they really are.

**Structural Dependency.** Find the equations or variables involved in a structural dependency. For systems that should be square, this is similar to overspecified, but for DAE's this detects the equations which need to be differentiated to overcome an index problem in the model. The user interface for reporting the data returned is not complete.

**Find unassigned eqns.** Shows the equations which cannot be assigned by the structural analysis.

**Evaluate unincluded eqns.** Evaluates the residuals of unincluded relations and checks them for convergence. This may or may not be a wise idea, depending on why the relations have been excluded.

**Find vars near bounds** This will write variable names passing the test

$$\text{abs}(\text{value-bound})/\text{nominal} < \text{epsilon} \quad (6.1)$$

to the console. The test is performed first for lower bounds and then for upper bounds and the results are clearly marked. This tool can be used for locating variables which may yield a more tractable problem when moved to the bound and fixed while freeing other variables. The value

of Epsilon can be set on the Solvers General parameter page accessible through the Options button in the Solver window.

**Find vars far  
from nominal**

This tool will write variable names passing test

$$\text{abs}(\text{value-nominal})/\text{nominal} > \text{bignum} \quad (6.2)$$

to the console. This test can be used for locating variables which are poorly scaled and for evaluating where model initialization methods need improvement. The value of bignum can be set using the Solvers General parameter page accessible through the Options button in the Solver window.

### 6.1.6 SOLVER VIEW MENU

**Font**

Opens the window that lets you reset the fonts for this window. You can select the type of font, the style (bold, etc.) and the size for the font.

**Open  
automatically**

Toggles a switch which, if set, will cause this window to open whenever anything is placed into it.

**Save Solver  
appearance**

Saves the current settings for this window for font settings and window size and placement on your computer screen. These become the default settings for opening this window in the future. These settings are saved in a *.a4o* text file for this window which the system stores in the subdirectory *ascdata* in your “home” directory.

### 6.1.7 SOLVER EXPORT MENU

**to Browser**

This button sends the instance currently in the Solver to the Browser.

**to Probe**

This button sends the instance currently in the Solver to the Probe piecewise; that is, all the variables and relations get shipped, not the instance name itself.

## 6.2 SOLVER BUTTON BAR

The solver button bar, which is located just below the solver menu, contains three buttons, the solver select button, the solver options button, and the halt button.

**Solver Select  
Button**

This button contains the name of the currently selected solver. Depressing this button reveals a menu of available solvers which can be selected.

### Solver Options Button

The Options menu on the solver allows the user to view and to change the settings for the parameters associated with ASCEND's solvers. A solver's parameters may be changed even when the solver is empty or another solver is selected. Depressing the options button reveals a list of parameter pages which can be selected for viewing (and editing).

Below, we discuss using the parameter pages and the general solver parameters; solver specific parameters are discussed below in Section 6.3, "Available Solvers," on page 67.

### Halt Button

Halts the solver and returns control to the interface as soon as possible. Not all solvers connected to ASCEND will respond to the halt signal.

## 6.2.1 GENERAL PARAMETERS PAGE

Selecting General under Options will display the General Parameter Page (See Figure 6-3). This is where we keep items relevant to the interface and to the way mathematical specialty functions and utilities are handled in ASCEND. Following, we will discuss the parameters that appear on this page.

Figure 6-3 General Parameter Page

iterations before screen update	10
cpu sec before screen update	3
modified log epsilon	1e-08
bound check epsilon	1e-3
far from nom bignum	10e3
integrator state log	y.dat
integrator observation log	obs.dat
integrator log SI units	display
integrator log columns	variable
<input checked="" type="checkbox"/> overwrite integrator logs	
<input type="checkbox"/> check numeric rank after solving	
<input type="checkbox"/> show block summary	
<input type="button" value="OK"/> <input type="button" value="Help"/>	

### Iterations before screen update

Because the interface update is sometimes rather time consuming (or more accurately when the window manager is slow, the interface holds up the solver) this specifies how many iterations to stay down in the solver algorithm before returning to the user interface to update statistics. In the case of floating point errors or solution completion

before the limit is reached, the return and update will happen immediately rather than waiting for the limit to be reached. For solvers that don't truly iterate in an accessible fashion (e.g. MINOS) this parameter is ignored.

CPU seconds before screen update

For solvers which do offer access to status information between iterations, this is the maximum number of cpu seconds before an interface update. If, while still not done with the number of iterations given in "iterations before screen update," the solver algorithm detects that the cpu seconds limit has expired, then it will return early to update the interface. At least one iteration will be completed before the clock is checked.

Modified log epsilon

This parameter controls the value for epsilon in the "lnm" function. Lnm can be used instead of natural log (ln) when the argument is likely to be very small or to go negative in the solution process. This avoids a host of floating point errors in initialization and solving of many kinds of models.

The modified natural log function  $f$  is defined as

$$f(x) = \left\{ \begin{array}{l} \ln(x) \forall (x > \epsilon) \\ \frac{x}{\epsilon} - 1 + \ln(\epsilon) \forall (x \leq \epsilon) \end{array} \right\}$$

The first derivative of this function is continuous. The second derivative has a jump from 0 to  $-1/\epsilon^2$  at  $x = \epsilon$ .

Bound check epsilon

This is the epsilon parameter used in the [link: to obvious location/Find vars near bounds] utility under the Solvers Analyze Menu.

Far from nom bignum

This is the bignum parameter used in the [link: to obvious location/Find vars far from nom] utility under the Solvers Analyze Menu.

Integrator state log

This is the name of the file for integrator state variable output. It defaults to y.dat in the current directory.

Integrator observation log

This is the name of the file for user defined observation output during integration. It defaults to obs.dat in the current directory.

Integrator log SI units

This switch causes the output to be written in SI units or in the user's selected interface units.

<u>Integrator log columns</u>	This option selects how the state and observation logs should be formatted. We can produce fixed or variable width formats suitable for import into nearly any other software package.
<u>Overwrite integer logs</u>	This switch lets the user control whether integration log files should be appended or replaced with each run.
<u>Check numeric rank after solving</u>	When selected the numeric rank will be checked at the solution and a message will be displayed if the system is rank deficient.
<u>Show block summary</u>	When selected the cost statistics (cpu, iterations, evaluations) for all blocks of significant size will be listed to the screen after each solve.

## 6.3 AVAILABLE SOLVERS

Here is the list of solvers that at one time or another have been connected to ASCEND:

- Slv
- QRSlv
- LSODE
- MINOS
- LSSlv
- Opt (SQP)
- CONOPT
- Make MPS

All of these solvers may not be available in your installation of ASCEND. A brief description of ASCEND's primary solver, QRSlv, follows.

### 6.3.1 QRSLV

QRSlv is a nonlinear algebraic equation solver based on the paper "A Modified Least Squares Algorithm for Solving Sparse NxN Sets of Nonlinear Equations" by A. Westerberg and S. Director (EDRC TECH REPORT 06-5-79).



### 6.3.1.1 PARAMETERS

Following is an incomplete list of control parameters for the QRSlv algorithm. Most users will only change the time limit, iteration limit, and maximum residual as the default parameter values work quite well.

<u>Time limit</u>	The total number of seconds allowed in 1 push of the Solve button.
<u>Iteration limit</u>	Total number of iterations in for any single partition in the problem.
<u>Minimum pivot (epsilon)</u>	the smallest pivot value allowed in the linear solution of a subproblem.
<u>Pivot tolerance</u>	pivot selection criterion.
<u>Maximum residual</u>	This is the maximum absolute error that QRSlv is allowed to consider an equation as solved. Self scaling equations will more easily satisfy this than those that aren't. E.g. an energy balance (with terms the size of $10^8$ ) will have a far harder time meeting this convergence criterion if you do not divide them through by an appropriate constant.
<u>Partitioning</u>	If off, entire problem will be solved as a block. Divergence is usually the result on nonlinear problems of any size above 25 or so.
<u>Detailed info</u>	QRSlv spews all sorts of info if you turn this switch on. The utility of such info is often as much for the authors of slv as for the user. The volume of info is large. Most of the spew (that to do with singletons (1x1 blocks) is suppressed if the switch 'show singletons details' is off.
<u>Auto-resolve</u>	When complete, the solver is supposed to rerun itself for changes of significance made in the interface if this switch is on.
<u>write to file SlvLinsol.dat</u>	If this switch is on, a whole set of files named SlvLinsol.dat.X where X is integer are produced during the solution of the problem. The X increments for each successive linear system inversion or solution. The files contain Jacobian and rhs data in machine readable forms for import to stand-alone solver tools. There are generally quite a lot of them. X always starts at 0 for a given ascend session and goes up from there.
<u>show singletons details</u>	When the 'detailed solving info required' switch is ON this switch controls whether or not full singleton solving information is shown. In particular, if this is off all direct solve spew is cancelled, leaving that which usually of interest, the NxN block solution iterations, to be displayed.

bipartial pivoting

An experimental option for stabilizing the RANKI algorithm on hard problems. It enables searching of both current row and column during linear factorization. It is somewhat more expensive in terms on fill and CPU time, but can lead to solution of otherwise unsolvable problems. The modification is due to Joe Zaher. This option is likely to be replaced by a choice of several linear routines eventually. The original motivation came from distillation models which become illconditioned as tray number grows.

## 6.4 DEBUGGER

The Debugger shown in Figure 6-4 is an aid for examining the variables and relations in the Solver. The debugger is often used in tandem with the incidence matrix because the debugger is queried using the solvers's internal relation/variable indexing (which starts at 0). When a variable (relation) number is typed in the variable (relation) entry box the variable (relation) Name and Attribute buttons may be clicked to obtain information about the variable (relation). The information is printed to the console window. The variable (relation) may also be exported to either the Browser or Probe by making the appropriate selection under the export pull down menu.

Figure 6-4 The Debugger Window

Variable:
<input type="text"/>
Name
Attributes
Export
Equation:
<input type="text"/>
Name
Attributes
Export
Block:
<input type="text"/>
Variables
Equations
Export to probe
System:
Variables
Export to probe
<input type="button" value="OK"/>
<input type="button" value="Help"/>

When a variable or relation number is entered in the debugger, the corresponding partitioned block number appears in the 'block' entry box. Statistics on the number of rows and columns in the block are displayed just below the block entry box. Note that a block number can also be entered directly into the block entry box. The Variables (Equations) pull down menu below the block entry box contains the selections Values, Attributes, and Probe (and Find Dependent). Selecting Values or Attributes will write the requested information to the console for each variable (equation) in the block. Selecting Probe will export the block's variables (equations) to the probe. Selecting Find Dependent under the Equations pull down menu will write the name of any dependent equations within the block to the console. Selecting the Export to Probe button will export both the block's variables and equations to the probe.

The debugger also responds to requests relating to the overall system, or current solver instance. A Variables pull down menu and an Export to Probe button are located beneath the 'System' label on the debugger. The Variables menu contains the selections Values, Attributes, Reset Values, and Reset Nominals. Selecting Values or Attributes will write the requested information to the console for each variable in the system. Selecting Reset Values will reset the system's variables to their nominal values. Selecting Reset Values will reset the system's nominals to their current variable values.