Solving Constrained Nonlinear System (CNS) with GAMS

Introduction

GAMS introduces a new model class called CNS - Constrained Nonlinear System. A CNS model is defined as a square system of equations (i.e. a model in which the number of non-fixed variables is equal to the number of constraints) in which bounds on the variables are allowed, for example to prevent function evaluation errors for functions that only are defined for some arguments or to force the solution to lie in a meaningful region. The bounds are assumed to be non-binding. Inequalities are allowed for the same reason, but the slacks in the inequalities are counted as non-fixed variables, so that a square system can be augmented by an inequality and remain square.

The CNS model class is solved with a solve statement of the form:

```
SOLVE <MODEL> USING CNS;
```

without the usual objective term. The CNS solver can be selected during installation or with the usual "OPTION CNS = Solver;" statement. Currently, PATHCNS and CONOPT2 can be used as CNS solvers.

GAMS will check that the number of non-fixed variables including slacks is equal to the number of constraints and it will stop with a model generation error if it is not. Otherwise, the CNS solver will be called and it will attempt to solve the constraints with respect to the non-fixed variables.

Status values

A CNS solver will return one of the following solver status values:

Solverstat = 1	Normal completion
Solverstat = 2	Iteration Interrupt
Solverstat = 3	Resource Interrupt
Solverstat = 5	Evaluation Error Limit
Solverstat = 8	User Interrupt

where the definitions are the same as for other model classes. The CNS solver will return one of the following model status values:

Modelstat = 15	Solved Unique	The model was solved and the solution is globally unique.
Modelstat = 16	Solved	The model was solved.
Modelstat = 17	Solved Singular	The model was solved, but the Jacobian is singular.
Modelstat = 4	Infeasible	The model is guaranteed to have no solution, for example because the model or a critical subset of the model is linear and inconsistent.
Modelstat = 5	Locally infeasible	The model could not be solved. The word locally indicates that the solver has not determined that no solution could exist, as compared to the global infeasibility implied by Modelstat=4.
Modelstat = 6	Intermediate infeasible	The solver was stopped by an iteration, resource, or evaluation error limit or by a user interrupt.

Note that there are three new model status values: 15, 16, and 17. The solution itself is returned to the GAMS database and printed in the listing file in the usual way.

Additional comments

Some special comments relating to CNS models apply:

- There is no objective and therefore no marginal values, either for variables or for equations. The solution listing will therefore not have the MARGINAL column. Any marginal values already stored in the GAMS database will remain untouched.
- A singular model flags a set of linearly dependent rows and columns with "DEPND" in the solution listing. The number of dependencies reported is made available by GAMS via the new <model>.numdepnd model attribute. This can be tested in the usual way. Note that a row/column pair for a linear dependence contributes one to numdepnd. Also note that there may be more linear dependencies than the ones reported.
- An infeasible or locally infeasible model, or a singular model with infeasibilities, flags the infeasible constraints and variables with the usual "INFES" flag. The number of infeasibilities is available via the usual <model>.numinfes model attribute.