

SULUM 1.0

Sulum Optimization, www.sulumoptimization.com

Contents

1	Introduction	443
2	How to Run a Model with Sulum	444
3	Overview of GAMS/Sulum	444
4	GAMS Options	444
5	Summary of SULUM Options	445
5.1	Options related to the simplex optimizer	445
5.2	Options related to logging in Sulum	446
5.3	Options related to the presolve module	446
5.4	Other options	446
5.5	The GAMS/Sulum Options File	447
6	GAMS/Sulum Log File	447
7	Detailed Descriptions of SULUM Options	448

1 Introduction

Sulum Optimization Tools includes a high performance optimizer for Linear Programming Problems (LP). The LP optimizer is based on a very efficient simplex code, which uses state of the art implementation and algorithmic techniques. The key features can be summarized as :

- Highly optimized sparse vector and matrix implementation.
- Fast and stable LU factorization and sparsity exploiting solves.
- Numerical stabilizing techniques used, which include auto switching to using quad precision if deemed needed.
- Multiple pricing options i.e steepest edge, approximate steepest edge and partial pricing.
- Advanced crash of initial basis.
- Auto dualizer switch to solving the dual problem if it is smaller than the primal problem.
- Fast reoptimize from a previous found solution.

While numerous solving options are available, Sulum automatically calculates and sets most options at the best values for specific problems. All Sulum options available through GAMS/Sulum are summarized at the end of this chapter.

2 How to Run a Model with Sulum

The following statement can be used inside your GAMS program to specify using Sulum

```
Option LP = Sulum;
```

The above statement should appear before the `solve` statement. If Sulum was specified as the default solver during GAMS installation, the above statement is not necessary.

3 Overview of GAMS/Sulum

Sulum can solve LPs using a primal or a dual simplex algorithm.

The Sulum presolve can sometimes diagnose a problem as being infeasible or unbounded. When this happens, GAMS/Sulum can, in order to get better diagnostic information, rerun the problem with presolve turned off. The rerun without presolve is controlled by the option [rerun](#). In default mode only problems that are small (i.e. demo sized) will be rerun.

Sulum can either presolve a model or start from an advanced basis or primal/dual solution pair. Often the solve from scratch of a presolved model outperforms a solve from an unpresolved model started from an advanced basis/solution. It is impossible to determine a priori if presolve or starting from a given advanced basis/solution without presolve will be faster. By default, GAMS/Sulum will automatically use an advanced basis or solution from a previous solve statement. The GAMS BRatio option can be used to specify when not to use an advanced basis/solution. The GAMS/Sulum option [usebasis](#) can be used to ignore or force a basis/solution passed on by GAMS (it overrides BRatio). In case of multiple solves in a row and slow performance of the second and subsequent solves, the user is advised to set the GAMS BRatio option to 1.

4 GAMS Options

The following GAMS options are used by GAMS/Sulum:

Option BRatio = x;

Determines whether or not to use an advanced basis. A value of 1.0 causes GAMS to instruct Sulum not to use an advanced basis. A value of 0.0 causes GAMS to construct a basis from whatever information is available. The default value of 0.25 will nearly always cause GAMS to pass along an advanced basis if a solve statement has previously been executed.

Option IterLim = n;

Sets the simplex iteration limit. Simplex algorithms will terminate and pass on the current solution to GAMS. For MIP problems, if the number of the cumulative simplex iterations exceeds the limit, Sulum will terminate.

Option ResLim = x;

Sets the time limit in seconds. The algorithm will terminate and pass on the current solution to GAMS. Sulum measures time in wall time on all platforms. Some other GAMS solvers measure time in CPU time on some Unix systems. This GAMS option is overridden by the GAMS/Sulum option [opttimelimit](#).

ModelName.Cutoff = x;

Cutoff value. The algorithm terminates when it is proven that the optimal value is worse than the cutoff value. This GAMS option is overridden by the GAMS/Sulum options [simobjupcut](#) (for minimization problems) or [simobjlocut](#) (for maximization problems).

ModelName.OptFile = 1;

Instructs GAMS/Sulum to read the option file. The name of the option file is `sulum.opt`.

5 Summary of SULUM Options

5.1 Options related to the simplex optimizer

[simprimprice](#)

Controls which pricing strategy should be used by the primal simplex optimizer.

[simdualprice](#)

Controls which pricing strategy should be used by the dual simplex optimizer.

[simprimpricehotstart](#)

Controls which pricing strategy should be used by the primal simplex optimizer, when a hotstart is available.

[simdualpricehotstart](#)

Controls which pricing strategy should be used by the dual simplex optimizer, when a hotstart is available.

[simloglevel](#)

Controls the amounts of output from the simplex optimizer.

[simquadprecision](#)

Controls if quad precision is used in the simplex optimizer.

[simperturblevel](#)

Controls the level of perturbations in the simplex optimizer.

[simshifting](#)

Controls the if shifting is used in the simplex optimizer.

[simwarmstart](#)

If this key is switched off then the optimizer disregard any solution stored in the model.

[simprob](#)

Some time it might be faster to solve the respective dual formulation instead of the primal.

[simscale](#)

Controls if the simplex optimizer should scale data to be more numerical stable.

[simscalehotstart](#)

Controls if the simplex optimizer should scale data to be more numerical stable, when a hotstart is present.

[simmaxiter](#)

Maximum iterations allowed in simplex optimizers.

[simobjcutnosol](#)

Controls if a solution is needed when the optimizer stops premature due to objective cut .

[simusequadinf](#)

Controls if Sulum Simplex Optimizer should use switch to quad precision, when a problem is determined infeasible.

[simsolveunscaled](#)

Controls if the simplex optimizer should reoptimize on a unscaled problem if tolerances are not met.

[simtolprim](#)

Absolut tolerance used by the simplex optimizer to determine if a solution is primal feasible or not.

[simtoldual](#)

Absolut tolerance used by the simplex optimizer to determine if a solution is dual feasible or not.

[simtolpivot](#)

Absolut tolerance used by the simplex optimizer to control the minimum size of a pivot element.

[simtolmarko](#)

Absolut tolerance used by the simplex optimizer to control the stability of pivot size in LU factorization module.

[simtimelimit](#)

Maximum time allowed in the simplex optimizer.

[simobjpcut](#)

If the optimal objective value can be proved to be larger than this value, then the optimizer terminates.

[simobjlocut](#)

If the optimal objective value can be proved to be less than this value, then the optimizer terminates.

[usebasis](#)

Use basis from GAMS

5.2 Options related to logging in Sulum

[loglevel](#)

Controls the amount of output from Sulum in general.

[logprefix](#)

Skip doing prefix in logging i.

[lognomodulemessage](#)

Skip writing stop and start for each optimizer module in logging.

5.3 Options related to the presolve module

[Presolve](#)

Controls which type of presolve strategy should be used by the presolve module.

[presolvehotstart](#)

Controls which type of presolve strategy should be used by the presolve module, when a hotstart is present.

[presolvecompress](#)

Controls if the problem should be compressed after a call to the presolve module.

[presolvecompresshotstart](#)

Controls if the problem should be compressed after a call to the presolve module were a hotstart is present.

5.4 Other options

[optimizer](#)

Controls which optimizer will be used.

[primcrash](#)

Controls if the primal simplex optimizer should crash a advanced start basis.

[dualcrash](#)

Controls if the dual simplex optimizer should crash a advanced start basis.

[mpswritenames](#)

Controls if Sulum MPS writer should replace constraint and variable names with generic ones .

[debug](#)

This option can be switch on in debug mode and development phase to find bugs easier.

[updatesolquality](#)

Decides if the optimizer should update information solution quality items at the end of a call to the optimizer.

[lpwritenames](#)

Controls if Sulum LP writer should replace constraint and variable names with generic ones .

[lpwritenumonline](#)

Controls how many items Sulum LP writer should write on each line.

[lpwriteprecision](#)

Controls how many items Sulum LP writer should write on each line.

[lpwritefreecons](#)

Controls if Sulum LP writer should write free constraints.

[mpswritefreecons](#)

Controls if Sulum MPS writer should write free constraints.

[lpreadfreecons](#)

Controls if Sulum LP reader should read free constraints.

[mpsreadfreecons](#)

Controls if Sulum MPS reader should read free constraints.

[lpwritevarorder](#)

Controls if Sulum LP writer should write variables in same order i.

wrnlargea

If a absolute value in the constraint matrix is larger than this value a warning will be displayed, but only if debug is on.

wrnsmalla

If a absolute value in the constraint matrix is smaller than this value a warning will be displayed, but only if debug is on.

wrnlargec

If a absolute value in the objective is larger than this value a warning will be displayed, but only if debug is on.

wrnlargelo

If the absolute value of a lower bound is larger than this value a warning will be displayed, but only if debug is on.

wrnlargeup

If the absolute value of a upper bound is larger than this value a warning will be displayed, but only if debug is on.

opttimelimit

Maximum time allowed in the optimizer.

optsolvezero

Tolerance on what is considered zero in solves with basis call by the user.

dumpsolution

Controls export of alternate MIP solutions

names

Indicator for loading names

prntoptions

List values of all options to GAMS listing file

rerun

Resolve without presolve in case of unbounded or infeasible

writetprob

Save the problem instance

5.5 The GAMS/Sulum Options File

The GAMS/Sulum options file consists of one option or comment per line. An asterisk (*) at the beginning of a line causes the entire line to be ignored. Otherwise, the line will be interpreted as an option name and value separated by any amount of white space (blanks or tabs).

Following is an example options file *sulum.opt*.

```
simprimprice 2
optimizer 1
```

It will cause Sulum to use approximate steepest edge pricing strategy and will use the primal simplex algorithm.

6 GAMS/Sulum Log File

Sulum reports its progress by writing to the GAMS log file as the problem solves. Normally the GAMS log file is directed to the computer screen.

The log file shows statistics about the presolve and continues with an iteration log.

For the simplex algorithms, each log line prints the iteration number, followed by infeasibility or objective value, and the elapsed wall clock time in every 5th line. The simplex screen log has the following appearance:

```
Starting presolve
Problem to be optimized has 2725 constraints 6569 variables and 36535 non zeroes in A matrix.
Presolved problem has 2079 constraints 5453 variables and 32948 non zeroes in A matrix.
Completed presolve
Presolve time : 0.083
Starting Simplex Optimizer
```

```

No hotstart used
Starting Dual Simplex Optimizer
Itrs : 0          DInf : 4.9532861e+001 (289)
Itrs : 117        DInf : 4.1208861e+001 (35)
Itrs : 231        DInf : 2.5172856e+001 (35)
Itrs : 347        DObj : 1.1263950e+006 (1284)
Itrs : 463        DObj : 6.0641149e+005 (1242) Time : 0.028
Itrs : 587        DObj : 3.7404770e+005 (1145)
Itrs : 709        DObj : 2.7442594e+005 (1064)
Itrs : 823        DObj : 2.3131625e+005 (1012)
Itrs : 941        DObj : 2.0104705e+005 (883)
Itrs : 1065       DObj : 1.7118902e+005 (790) Time : 0.074
Itrs : 1190       DObj : 1.5203522e+005 (726)
Itrs : 1313       DObj : 1.4288398e+005 (710)
Itrs : 1441       DObj : 1.3519751e+005 (610)
Itrs : 1569       DObj : 1.3044210e+005 (572)
Itrs : 1688       DObj : 1.2853757e+005 (479) Time : 0.135
Itrs : 1810       DObj : 1.2252508e+005 (396)
Itrs : 1938       DObj : 1.2139219e+005 (287)
Itrs : 2051       DObj : 1.2109905e+005 (318)
Itrs : 2163       DObj : 1.1940995e+005 (279)
Itrs : 2275       DObj : 1.1888603e+005 (238) Time : 0.184
Itrs : 2387       DObj : 1.1862120e+005 (271)
Itrs : 2499       DObj : 1.1752886e+005 (278)
Itrs : 2611       DObj : 1.1708406e+005 (331)
Itrs : 2723       DObj : 1.1674983e+005 (251)
Itrs : 2835       DObj : 1.1647282e+005 (288) Time : 0.319
Itrs : 2947       DObj : 1.1617754e+005 (210)
Itrs : 3059       DObj : 1.1592797e+005 (132)
Itrs : 3171       DObj : 1.1566295e+005 (181)
Itrs : 3283       DObj : 1.1538715e+005 (95)
Itrs : 3395       DObj : 1.1495725e+005 (55) Time : 0.487
Itrs : 3467       DObj : 1.1487366e+005 (19)
Solution is optimal
Completed Dual Simplex Optimizer
Completed Simplex Optimizer
Starting postsolve
Completed postsolve
Completed optimizer
Optimizer Time : 0.666
Setup solution quality items
Model was solved to optimality (subject to tolerances).

```

7 Detailed Descriptions of SULUM Options

Presolve (*integer*) Controls which type of presolve strategy should be used by the presolve module.

(default = 1)

- 0 Do not apply any presolve strategies.
- 1 The optimizer automatically decides if presolve and which type of presolve should be applied.
- 2 Presolve will only use very simple methods to reduce problem size.
- 3 The presolve will only use strategies based on primal information.
- 4 The presolve will only use strategies based on dual information.

- 5 Presolve will reduce problem size using all methods.

optimizer (*integer*) Controls which optimizer will be used.

(*default = 0*)

- 0 The optimizer decides which optimizer to call based on the model structure.
- 1 The primal simplex optimizer should be applied.
- 2 The dual simplex optimizer should be applied.

presolvehotstart (*integer*) Controls which type of presolve strategy should be used by the presolve module, when a hot-start is present.

(*default = 1*)

- 0 Do not apply any presolve strategies.
- 1 The optimizer automatically decides if presolve and which type of presolve should be applied.
- 2 Presolve will only use very simple methods to reduce problem size.
- 3 The presolve will only use strategies based on primal information.
- 4 The presolve will only use strategies based on dual information.
- 5 Presolve will reduce problem size using all methods.

presolvecompress (*integer*) Controls if the problem should be compressed after a call to the presolve module.

(*default = 0*)

- 0 The presolve automatically decides if final problem should be compressed.
- 1 Presolve use will use compression.
- 2 Do not apply any presolve compression.

presolvecompresshotstart (*integer*) Controls if the problem should be compressed after a call to the presolve module were a hotstart is present.

(*default = 0*)

- 0 The presolve automatically decides if final problem should be compressed.
- 1 Presolve use will use compression.
- 2 Do not apply any presolve compression.

simprimprice (*integer*) Controls which pricing strategy should be used by the primal simplex optimizer.

(*default = 0*)

- 0 The simplex optimizer analyzes the model and decides the best choice in the given situation.
- 1 The simplex optimizer will use steepest edge strategy, which is the most expensive pricing strategy, but also often the one with fewest iterations.
- 2 The simplex optimizer will use approximate steepest edge strategy, which relaxes the steepest edge strategy by using only approximate norms.
- 3 The simplex optimizer will scan only a subset of candidates and choose between promising candidates by a very cheap scheme.

simdualprice (*integer*) Controls which pricing strategy should be used by the dual simplex optimizer.

(*default = 0*)

- 0 The simplex optimizer analyzes the model and decides the best choice in the given situation.
- 1 The simplex optimizer will use steepest edge strategy, which is the most expensive pricing strategy, but also often the one with fewest iterations.
- 2 The simplex optimizer will use approximate steepest edge strategy, which relaxes the steepest edge strategy by using only approximate norms.

- 3 The simplex optimizer will scan only a subset of candidates and choose between promising candidates by a very cheap scheme.

simprimpricestart (integer) Controls which pricing strategy should be used by the primal simplex optimizer, when a hotstart is available.

(default = 0)

- 0 The simplex optimizer analyzes the model and decides the best choice in the given situation.
- 1 The simplex optimizer will use steepest edge strategy, which is the most expensive pricing strategy, but also often the one with fewest iterations.
- 2 The simplex optimizer will use approximate steepest edge strategy, which relaxes the steepest edge strategy by using only approximate norms.
- 3 The simplex optimizer will scan only a subset of candidates and choose between promising candidates by a very cheap scheme.

simdualpricestart (integer) Controls which pricing strategy should be used by the dual simplex optimizer, when a hotstart is available.

(default = 0)

- 0 The simplex optimizer analyzes the model and decides the best choice in the given situation.
- 1 The simplex optimizer will use steepest edge strategy, which is the most expensive pricing strategy, but also often the one with fewest iterations.
- 2 The simplex optimizer will use approximate steepest edge strategy, which relaxes the steepest edge strategy by using only approximate norms.
- 3 The simplex optimizer will scan only a subset of candidates and choose between promising candidates by a very cheap scheme.

loglevel (integer) Controls the amount of output from Sulum in general.

(default = 5)

simloglevel (integer) Controls the amounts of output from the simplex optimizer.

(default = 5)

simquadprecision (integer) Controls if quad precision is used in the simplex optimizer.

(default = 0)

- 0 The given option is off.
- 1 The given option is on.

simperturblevel (integer) Controls the level of perturbations in the simplex optimizer.

(default = 50)

simshifting (integer) Controls the if shifting is used in the simplex optimizer.

(default = 1)

- 0 The given option is off.
- 1 The given option is on.

simwarmstart (integer) If this key is switched off then the optimizer disregard any solution stored in the model.

(default = 1)

- 0 The given option is off.
- 1 The given option is on.

simprob (integer) Some time it might be faster to solve the respective dual formulation instead of the primal.

(default = 0)

- 0 The optimizer decides if the primal or dual formulation should be solved.
- 1 The primal formulation should be solved.
- 2 The dual formulation should be solved.

primcrash (integer) Controls if the primal simplex optimizer should crash a advanced start basis.

(default = 1)

- 0 The given option is off.
- 1 The given option is on.

dualcrash (integer) Controls if the dual simplex optimizer should crash a advanced start basis.

(default = 1)

- 0 The given option is off.
- 1 The given option is on.

simscale (integer) Controls if the simplex optimizer should scale data to be more numerical stable.

(default = 1)

- 0 The given option is off.
- 1 The given option is on.

simscalehotstart (integer) Controls if the simplex optimizer should scale data to be more numerical stable, when a hotstart is present.

(default = 1)

- 0 The given option is off.
- 1 The given option is on.

mpswritenames (integer) Controls if Sulum MPS writer should replace constraint and variable names with generic ones .

(default = 0)

- 0 The given option is off.
- 1 The given option is on.

debug (integer) This option can be switch on in debug mode and development phase to find bugs easier.

(default = 0)

- 0 The given option is off.
- 1 The given option is on.

logprefix (integer) Skip doing prefix in logging i.

(default = 0)

- 0 The given option is off.
- 1 The given option is on.

lognomodulemessage (integer) Skip writting stop and start for each optimizer module in logging.

(default = 0)

- 0 The given option is off.
- 1 The given option is on.

simmaxiter (integer) Maximum iterations allowed in simplex optimizers.

(default = infinity)

updatesolquality (integer) Decides if the optimizer should update information solution quality items at the end of a call to the optimizer.

(default = 1)

0 The given option is off.

1 The given option is on.

simobjcutnosol (integer) Controls if a solution is needed when the optimizer stops premature due to objective cut .

(default = 0)

0 The given option is off.

1 The given option is on.

lpwritenames (integer) Controls if Sulum LP writer should replace constraint and variable names with generic ones .

(default = 0)

0 The given option is off.

1 The given option is on.

lpwritenumonline (integer) Controls how many items Sulum LP writer should write on each line.

(default = 5)

lpwriteprecision (integer) Controls how many items Sulum LP writer should write on each line.

(default = 4)

lpwritefreecons (integer) Controls if Sulum LP writer should write free constraints.

(default = 0)

0 The given option is off.

1 The given option is on.

mpswritefreecons (integer) Controls if Sulum MPS writer should write free constraints.

(default = 0)

0 The given option is off.

1 The given option is on.

lpreadfreecons (integer) Controls if Sulum LP reader should read free constraints.

(default = 0)

0 The given option is off.

1 The given option is on.

mpspreadfreecons (integer) Controls if Sulum MPS reader should read free constraints.

(default = 0)

0 The given option is off.

1 The given option is on.

lpwritevarorder (integer) Controls if Sulum LP writer should write variables in same order i.

(default = 1)

0 The given option is off.

1 The given option is on.

simusequadinf (integer) Controls if Sulum Simplex Optimizer should use switch to quad precision, when a problem is determined infeasible.

(default = 1)

0 The given option is off.

1 The given option is on.

simsolveunscaled (integer) Controls if the simplex optimizer should reoptimize on a unscaled problem if tolerances are not met.

(default = 1)

0 The given option is off.

1 The given option is on.

sintolprim (real) Absolut tolerance used by the simplex optimizer to determine if a solution is primal feasible or not.

Range: [1.0e-10,1.0e-4]

(default = 1.0e-6)

sintoldual (real) Absolut tolerance used by the simplex optimizer to determine if a solution is dual feasible or not.

Range: [1.0e-10,1.0e-4]

(default = 1.0e-6)

sintolpivot (real) Absolut tolerance used by the simplex optimizer to control the minimum size of a pivot element.

Range: [1.0e-12,1.0e-5]

(default = 1.0e-9)

sintolmarko (real) Absolut tolerance used by the simplex optimizer to control the stability of pivot size in LU factorization module.

Range: [1.0e-4,9.0e-1]

(default = 8.0e-3)

simtimelimit (real) Maximum time allowed in the simplex optimizer.

(default = GAMS reslim)

simobjupcut (real) If the optimal objective value can be proved to be larger than this value, then the optimizer terminates.

(default = maxdouble)

simobjlocut (real) If the optimal objective value can be proved to be less than this value, then the optimizer terminates.

(default = mindouble)

wrnlargea (real) If a absolute value in the constraint matrix is larger than this value a warning will be displayed, but only if debug is on.

(default = 1.0e+8)

wrnsmalla (real) If a absolute value in the constraint matrix is smaller than this value a warning will be displayed, but only if debug is on.

(default = 1.0e+8)

wrnlargec (real) If a absolute value in the objective is larger than this value a warning will be displayed, but only if debug is on.

(default = 1.0e+8)

wrnlargelo (real) If the absolute value of a lower bound is larger than this value a warning will be displayed, but only if debug is on.

(default = 1.0e+8)

wrnlargeup (real) If the absolute value of a upper bound is larger than this value a warning will be displayed, but only if debug is on.

(default = 1.0e+8)

opttimelimit (real) Maximum time allowed in the optimizer.

(default = GAMS reslim)

optsolvezero (real) Tolerance on what is considered zero in solves with basis call by the user.

Range: [0,1.0e-8]

(default = 1.0e-12)

dumpsolution (string) Controls export of alternate MIP solutions

names (integer) Indicator for loading names

(default = 1)

printoptions (integer) List values of all options to GAMS listing file

(default = 0)

rerun (integer) Resolve without presolve in case of unbounded or infeasible

(default = 0)

usebasis (integer) Use basis from GAMS

(default = GAMS bratio)

writeprob (string) Save the problem instance