

SAS/OR[®] 14.3 User's Guide

Mathematical Programming

The OPTQP Procedure

This document is an individual chapter from *SAS/OR® 14.3 User's Guide: Mathematical Programming*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS/OR® 14.3 User's Guide: Mathematical Programming*. Cary, NC: SAS Institute Inc.

SAS/OR® 14.3 User's Guide: Mathematical Programming

Copyright © 2017, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2017

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Chapter 14

The OPTQP Procedure

| Contents | |
|---|------------|
| Overview: OPTQP Procedure | 673 |
| Getting Started: OPTQP Procedure | 675 |
| Syntax: OPTQP Procedure | 679 |
| Functional Summary | 680 |
| PROC OPTQP Statement | 680 |
| Details: OPTQP Procedure | 684 |
| Output Data Sets | 684 |
| Interior Point Algorithm: Overview | 686 |
| Iteration Log for the OPTQP Procedure | 688 |
| ODS Tables | 688 |
| Irreducible Infeasible Set | 692 |
| Macro Variable _OROPTQP_ | 693 |
| Examples: OPTQP Procedure | 694 |
| Example 14.1: Linear Least Squares Problem | 695 |
| Example 14.2: Portfolio Optimization | 697 |
| Example 14.3: Portfolio Selection with Transactions | 700 |
| References | 702 |

Overview: OPTQP Procedure

The OPTQP procedure solves quadratic programs—problems with quadratic objective function and a collection of linear constraints, including lower or upper bounds (or both) on the decision variables.

Mathematically, a quadratic programming (QP) problem can be stated as follows:

$$\begin{aligned}
 &\min \quad \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\
 &\text{subject to} \quad \mathbf{A} \mathbf{x} \{ \geq, =, \leq \} \mathbf{b} \\
 &\quad \quad \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}
 \end{aligned}$$

where

- $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is the quadratic (also known as Hessian) matrix
- $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the constraints matrix
- $\mathbf{x} \in \mathbb{R}^n$ is the vector of decision variables
- $\mathbf{c} \in \mathbb{R}^n$ is the vector of linear objective function coefficients
- $\mathbf{b} \in \mathbb{R}^m$ is the vector of constraints' right-hand sides (RHS)
- $\mathbf{l} \in \mathbb{R}^n$ is the vector of lower bounds on the decision variables
- $\mathbf{u} \in \mathbb{R}^n$ is the vector of upper bounds on the decision variables

The quadratic matrix \mathbf{Q} is assumed to be symmetric; that is,

$$q_{ij} = q_{ji}, \quad \forall i, j = 1, \dots, n$$

Indeed, it is easy to show that even if $\mathbf{Q} \neq \mathbf{Q}^T$, the simple modification

$$\tilde{\mathbf{Q}} = \frac{1}{2}(\mathbf{Q} + \mathbf{Q}^T)$$

produces an equivalent formulation $\mathbf{x}^T \mathbf{Q} \mathbf{x} \equiv \mathbf{x}^T \tilde{\mathbf{Q}} \mathbf{x}$; hence symmetry is assumed. When you specify a quadratic matrix, it suffices to list only lower triangular coefficients.

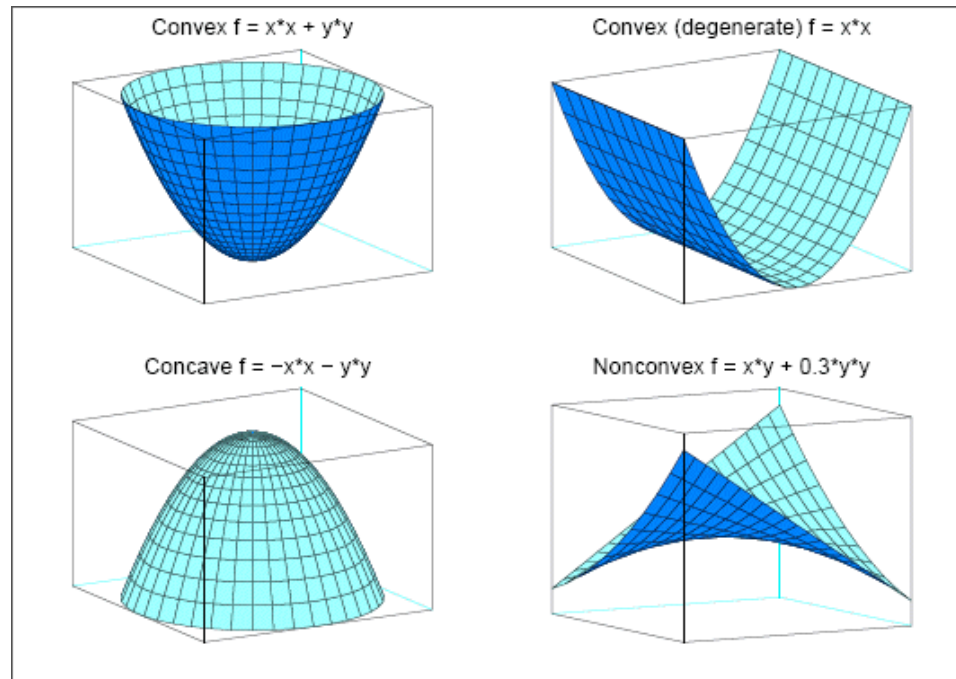
In addition to being symmetric, \mathbf{Q} is also required to be positive semidefinite,

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^n$$

for minimization type of models; it is required to be negative semidefinite for the maximization type of models. Convexity can come as a result of a matrix-matrix multiplication

$$\mathbf{Q} = \mathbf{L} \mathbf{L}^T$$

or as a consequence of physical laws, and so on. See [Figure 14.1](#) for examples of convex, concave, and nonconvex objective functions.

Figure 14.1 Examples of Convex, Concave, and Nonconvex Objective Functions

The order of constraints is insignificant. Some or all components of \mathbf{l} or \mathbf{u} (lower and upper bounds, respectively) can be omitted.

Getting Started: OPTQP Procedure

Consider a small illustrative example. Suppose you want to minimize a two-variable quadratic function $f(x_1, x_2)$ on the nonnegative quadrant, subject to two constraints:

$$\begin{array}{llllll} \min & 2x_1 & + & 3x_2 & + & x_1^2 & + & 10x_2^2 & + & 2.5x_1x_2 \\ \text{subject to} & x_1 & - & x_2 & \leq & 1 \\ & x_1 & + & 2x_2 & \geq & 100 \\ & x_1 & & & \geq & 0 \\ & & & x_2 & \geq & 0 \end{array}$$

The linear objective function coefficients, vector of right-hand sides, and lower and upper bounds are identified immediately as

$$\mathbf{c} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 100 \end{bmatrix}, \quad \mathbf{l} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} +\infty \\ +\infty \end{bmatrix}$$

Carefully construct the quadratic matrix \mathbf{Q} . Observe that you can use symmetry to separate the main-diagonal and off-diagonal elements:

$$\frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} \equiv \frac{1}{2} \sum_{i,j=1}^n x_i q_{ij} x_j = \frac{1}{2} \sum_{i=1}^n q_{ii} x_i^2 + \sum_{i>j} x_i q_{ij} x_j$$

The first expression

$$\frac{1}{2} \sum_{i=1}^n q_{ii} x_i^2$$

sums the main-diagonal elements. Thus, in this case you have

$$q_{11} = 2, \quad q_{22} = 20$$

Notice that the main-diagonal values are doubled in order to accommodate the 1/2 factor. Now the second term

$$\sum_{i>j} x_i q_{ij} x_j$$

sums the off-diagonal elements in the strict lower triangular part of the matrix. The only off-diagonal $(x_i x_j, i \neq j)$ term in the objective function is $2.5 x_1 x_2$, so you have

$$q_{21} = 2.5$$

Notice that you do not need to specify the upper triangular part of the quadratic matrix.

Finally, the matrix of constraints is as follows:

$$A = \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}$$

The SAS input data set with a quadratic programming system (QPS) format for the preceding problem can be expressed in the following manner:

```
data gsdata;
  input field1 $ field2 $ field3 $ field4 field5 $ field6 @;
  datalines;
NAME      .      EXAMPLE      .      .      .
ROWS      .      .      .      .      .
N          OBJ      .      .      .      .
L          R1      .      .      .      .
G          R2      .      .      .      .
COLUMNS  .      .      .      .      .
.          X1      R1          1.0      R2          1.0
.          X1      OBJ          2.0      .      .
.          X2      R1          -1.0     R2          2.0
.          X2      OBJ          3.0      .      .
RHS       .      .      .      .      .
.          RHS      R1          1.0      .      .
.          RHS      R2          100      .      .
RANGES    .      .      .      .      .
BOUNDS     .      .      .      .      .
QUADOBJ    .      .      .      .      .
.          X1      X1          2.0      .      .
.          X1      X2          2.5      .      .
.          X2      X2          20       .      .
ENDATA     .      .      .      .      .
;
```

For more details about the QPS-format data set, see Chapter 17, “[The MPS-Format SAS Data Set](#).”

Alternatively, if you have a QPS-format flat file named `gs.qps`, then the following call to the SAS macro `%MPS2SASD` translates that file into a SAS data set, named `gsdata`:

```
%mps2sasd(mpsfile =gs.qps, outdata = gsdata);
```

NOTE: The SAS macro `%MPS2SASD` is provided in SAS/OR software. For more information, see “[Converting an MPS/QPS-Format File: %MPS2SASD](#)” on page 824.

You can use the following call to PROC OPTQP:

```
proc optqp data=gsdata
  primalout = gspout
  dualout   = gsdout;
run;
```

The procedure output is displayed in [Figure 14.2](#).

Figure 14.2 Procedure Output
The OPTQP Procedure

| Problem Summary | |
|---------------------------------|--------------|
| Problem Name | EXAMPLE |
| Objective Sense | Minimization |
| Objective Function | OBJ |
| RHS | RHS |
| Number of Variables | 2 |
| Bounded Above | 0 |
| Bounded Below | 2 |
| Bounded Above and Below | 0 |
| Free | 0 |
| Fixed | 0 |
| Number of Constraints | 2 |
| LE (\leq) | 1 |
| EQ ($=$) | 0 |
| GE (\geq) | 1 |
| Range | 0 |
| Constraint Coefficients | 4 |
| Hessian Diagonal Elements | 2 |
| Hessian Elements Below Diagonal | 1 |

Figure 14.2 *continued*

| Solution Summary | |
|-----------------------------|----------------|
| Solver | QP |
| Algorithm | Interior Point |
| Objective Function | OBJ |
| Solution Status | Optimal |
| Objective Value | 15018.000051 |
| Primal Infeasibility | 0 |
| Dual Infeasibility | 0 |
| Bound Infeasibility | 0 |
| Duality Gap | 8.3025547E-9 |
| Complementarity | 0 |
| Iterations | 4 |
| Presolve Time | 0.00 |
| Solution Time | 0.02 |

The optimal primal solution is displayed in [Figure 14.3](#).

Figure 14.3 Optimal Solution

| Obs | Objective | | Variable Name | Variable Type | Linear | | Upper Bound | Variable Value | Variable Status |
|-----|-------------|--------|---------------|---------------|-----------------------|-------------|-------------|----------------|-----------------|
| | Function ID | RHS ID | | | Objective Coefficient | Lower Bound | | | |
| 1 | OBJ | RHS | X1 | N | 2 | 0 | 1.7977E308 | 34.0000 | O |
| 2 | OBJ | RHS | X2 | N | 3 | 0 | 1.7977E308 | 33.0000 | O |

The SAS log shown in [Figure 14.4](#) provides information about the problem, convergence information after each iteration, and the optimal objective value.

Figure 14.4 Iteration Log

NOTE: The problem EXAMPLE has 2 variables (0 free, 0 fixed).

NOTE: The problem has 2 constraints (1 LE, 0 EQ, 1 GE, 0 range).

NOTE: The problem has 4 constraint coefficients.

NOTE: The objective function has 2 Hessian diagonal elements and 1 Hessian elements above the diagonal.

NOTE: The QP presolver value AUTOMATIC is applied.

NOTE: The QP presolver removed 0 variables and 0 constraints.

NOTE: The QP presolver removed 0 constraint coefficients.

NOTE: The presolved problem has 2 variables, 2 constraints, and 4 constraint coefficients.

NOTE: The QP solver is called.

NOTE: The Interior Point algorithm is used.

NOTE: The deterministic parallel mode is enabled.

NOTE: The Interior Point algorithm is using up to 4 threads.

| | | | Primal | Bound | Dual | |
|------|------------|-------------|------------|------------|------------|------|
| Iter | Complement | Duality Gap | Infeas | Infeas | Infeas | Time |
| 0 | 4.4604E+03 | 2.6380E-01 | 1.7962E-02 | 1.8143E+00 | 2.7770E-14 | 0 |
| 1 | 1.3043E+02 | 8.2721E-03 | 1.7962E-04 | 1.8143E-02 | 8.7274E-15 | 0 |
| 2 | 1.3042E+00 | 8.3006E-05 | 1.7970E-06 | 1.8151E-04 | 9.8968E-15 | 0 |
| 3 | 1.3043E-02 | 8.3017E-07 | 1.7970E-08 | 1.8151E-06 | 1.9375E-14 | 0 |
| 4 | 0.0000E+00 | 8.3026E-09 | 1.9527E-09 | 1.8151E-08 | 3.5928E-07 | 0 |

NOTE: Optimal.

NOTE: Objective = 15018.000051.

NOTE: The Interior Point solve time is 0.00 seconds.

NOTE: There were 20 observations read from the data set WORK.GSDATA.

NOTE: The data set WORK.GSPOUT has 2 observations and 9 variables.

NOTE: The data set WORK.GSDOUT has 2 observations and 10 variables.

See the section “[Interior Point Algorithm: Overview](#)” on page 686 and the section “[Iteration Log for the OPTQP Procedure](#)” on page 688 for more details about convergence information given by the iteration log.

Syntax: OPTQP Procedure

The following statements are available in the OPTQP procedure:

PROC OPTQP <options> ;

Functional Summary

Table 14.1 outlines the options available for the OPTQP procedure classified by function.

Table 14.1 Options in the OPTQP Procedure

| Description | Option |
|--|---------------|
| Data Table Options | |
| Specifies a QPS-format input data set | DATA= |
| Specifies a dual solution output data set | DUALOUT= |
| Specifies whether the QP model is a maximization or minimization problem | OBJSENSE= |
| Specifies the primal solution output data set | PRIMALOUT= |
| Solver Options | |
| Enables or disables IIS detection | IIS= |
| Control Options | |
| Specifies the stopping criterion based on duality gap | DUALITYGAP= |
| Specifies the dual feasibility tolerance | DUALTOL= |
| Specifies how often to print the solution progress | LOGFREQ= |
| Specifies how much solution progress detail to print in log | LOGLEVEL= |
| Specifies the maximum number of iterations | MAXITER= |
| Specifies the time limit for the optimization process | MAXTIME= |
| Specifies the maximum number of threads | NTHREADS= |
| Specifies the parallel processing mode | PARALLELMODE= |
| Specifies the type of presolve | PRESOLVER= |
| Specifies the primal feasibility tolerance | PRIMALTOL= |
| Enables or disables printing summary | PRINTLEVEL= |
| Specifies units of CPU time or real time | TIMETYPE= |

PROC OPTQP Statement

The following options can be specified in the PROC OPTQP statement.

DATA=SAS-data-set

specifies the input SAS data set. This data set can also be created from a QPS-format flat file by using the SAS macro %MPS2SASD. If the DATA= option is not specified, PROC OPTQP uses the most recently created SAS data set. See Chapter 17, “[The MPS-Format SAS Data Set](#),” for more details.

DUALITYGAP= δ

specifies the desired relative duality gap, $\delta \in [1\text{E-}9, 1\text{E-}4]$. This is the relative difference between the primal and dual objective function values and is the primary solution quality parameter. For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686. The default value is $1\text{E-}6$.

DUALOUT=SAS-data-set

DOUT=SAS-data-set

specifies the output data set to contain the dual solution. For more information, see the section “[Output Data Sets](#)” on page 684.

DUALTOL= β

OPTTOL= β

specifies the maximum relative dual constraints violation, $\beta \in [1\text{E-}9, 1\text{E-}4]$. For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686. The default value is $1\text{E-}6$.

IIS=FALSE | TRUE

specifies whether to attempt to identify a set of constraints and variables that form an irreducible infeasible set (IIS). You can specify the following values:

FALSE disables IIS detection.

TRUE enables IIS detection.

If an IIS is found, information about infeasible constraints or variable bounds is written to the data sets that are specified in the DUALOUT= and PRIMALOUT= options. For more information, see the section “[Irreducible Infeasible Set](#)” on page 692. By default, IIS=FALSE.

LOGFREQ= k

PRINTFREQ= k

prints the solution progress to the iteration log after every k iterations, where k is an integer between 0 and the largest four-byte, signed integer, which is $2^{31} - 1$. The value $k = 0$ suppresses printing of the progress of the solution. By default, LOGFREQ=1.

LOGLEVEL=NONE | BASIC | MODERATE | AGGRESSIVE

PRINTLEVEL2=NONE | BASIC | MODERATE | AGGRESSIVE

controls the amount of information displayed in the SAS log. You can specify the following values:

NONE turns off all solver-related messages in the SAS log.

BASIC displays a solver summary after stopping.

MODERATE prints a solver summary and an iteration log by using the interval specified in the LOGFREQ= option.

AGGRESSIVE prints a detailed solver summary and an iteration log by using the interval specified in the LOGFREQ= option.

By default, LOGLEVEL=MODERATE.

MAXITER= k

specifies the maximum number of predictor-corrector iterations performed by the interior point algorithm (see the section “[Interior Point Algorithm: Overview](#)” on page 686). The value k is an integer between 1 and the largest four-byte, signed integer, which is $2^{31} - 1$. If you do not specify this option, the procedure does not stop based on the number of iterations performed.

MAXTIME=*t*

specifies an upper limit of *t* seconds of time for reading in the data and performing the optimization process. The value of the **TIMETYPE=** option determines the type of units used. If you do not specify this option, the procedure does not stop based on the amount of time elapsed. The value of *t* can be any positive number; the default value is the positive number that has the largest absolute value that can be represented in your operating environment.

NTHREADS=*k***NUMTHREADS=*k***

specifies the number of threads that PROC OPTQP can use, where *k* can be any integer between 1 and 256, inclusive. This option overrides the **THREADS** | **NOTHREADS** SAS system option. The default value of this option is the value of the **CPUCOUNT=** SAS system option.

Specifying *k* as a number greater than the actual number of available cores might result in reduced performance. Specifying a high value for *k* does not guarantee shorter solution time; the actual change in solution time depends on the computing hardware and the scalability of the underlying algorithms in PROC OPTQP. In some circumstances, PROC OPTQP might use fewer than *k* threads because the procedure's internal algorithms have determined that a smaller number is preferable.

OBJSENSE=MIN | MAX

specifies whether the QP model is a minimization or maximization problem. You can specify the following values:

MIN treat the QP model as a minimization problem.

MAX treat the QP model as a maximization problem.

Alternatively, you can specify the objective sense in the input data set; for more information, see the section “**ROWS Section**” on page 817. This option supersedes any objective sense that is specified in the input data set. If the objective sense is not specified anywhere, then PROC OPTQP interprets and solves the quadratic program as a minimization problem.

PARALLELMODE=DETERMINISTIC | NONDETERMINISTIC

specifies the parallel processing mode. This mode determines the solution results that are obtained from running the same model with the same option values on the same platform multiple times. You can specify the following values:

DETERMINISTIC requires algorithms to produce the same results every time.

NONDETERMINISTIC permits algorithms to produce different solution results. This mode requires less synchronization and might attain better performance than DETERMINISTIC mode.

By default, **PARALLELMODE=DETERMINISTIC**.

PRESOLVER=AUTOMATIC | NONE | BASIC | MODERATE | AGGRESSIVE**PRESOL=AUTOMATIC | NONE | BASIC | MODERATE | AGGRESSIVE**

specifies the presolve level. You can specify the following values:

| | |
|-------------------|---|
| AUTOMATIC | applies the presolver by using default setting. |
| NONE | disables the presolver. |
| BASIC | applies the basic presolver. |
| MODERATE | applies the moderate presolver. |
| AGGRESSIVE | applies the aggressive presolver. |

By default, PRESOLVER=AUTOMATIC.

PRIMALOUT=*SAS-data-set*

POUT=*SAS-data-set*

specifies the output data set to contain the primal solution. For more information, see the section “[Output Data Sets](#)” on page 684.

PRIMALTOL= α

FEASTOL= α

specifies the maximum relative bound and primal constraints violation, $\alpha \in [1\text{E-}9, 1\text{E-}4]$. For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686. The default value is $1\text{E-}6$.

PRINTLEVEL=0 | 1 | 2

specifies whether to print summary Output Delivery System (ODS) tables of the problem and solution. You can specify the following values:

- 0** does not produce or print any ODS tables.
- 1** produces and prints the following ODS tables: ProblemSummary, SolutionSummary, and optional OutputCasTables.
- 2** produces and prints the following ODS tables: ProblemSummary, SolutionSummary, ProblemStatistics, Timing, and optional OutputCasTables.

For more information about the ODS tables that PROC OPTQP creates, see the section “[ODS Tables](#)” on page 688. By default, PRINTLEVEL=1.

TIMETYPE=CPU | REAL

specifies whether CPU time or real time is used for the **MAXTIME**= option and the `_OROPTQP_` macro variable in a PROC OPTQP call. You can specify the following values:

- CPU** specifies that units are in CPU time.
- REAL** specifies that units are in real time.

The default value of the TIMETYPE= option depends on the value of the NTHREADS= option.

If you specify a value greater than 1 for the NTHREADS= option, the default value of the TIMETYPE= option is REAL. If you specify a value of 1 for the NTHREADS= option, the default value of the TIMETYPE= option is CPU.

Details: OPTQP Procedure

Output Data Sets

This section describes the PRIMALOUT= and DUALOUT= output data sets.

Definitions of Variables in the PRIMALOUT= Data Set

The PRIMALOUT= data set contains the primal solution to the quadratic programming (QP) model. The variables in the data set have the following names and meanings.

_OBJ_ID_

specifies the name of the objective function. Naming objective functions is particularly useful when there are multiple objective functions, in which case each objective function has a unique name. See the section “[ROWS Section](#)” on page 817 for details.

NOTE: PROC OPTQP does not support simultaneous optimization of multiple objective functions in this release.

_RHS_ID_

specifies the name of the variable that contains the right-hand-side value of each constraint. See the section “[RHS Section \(Optional\)](#)” on page 819 for details.

VAR

specifies the name of the decision variable.

TYPE

specifies the type of the decision variable. **_TYPE_** can take one of the following values:

- N nonnegative variable
- D bounded variable (with both lower and upper bound)
- F free variable
- X fixed variable
- O other (with either lower or upper bound)

OBJCOEF

specifies the coefficient of the decision variable in the linear component of the objective function.

LBOUND

specifies the lower bound on the decision variable.

UBOUND

specifies the upper bound on the decision variable.

VALUE

specifies the value of the decision variable.

STATUS

specifies the status of the decision variable. **_STATUS_** can indicate one of the following two cases:

- O The QP problem is optimal.
- I The QP problem could be infeasible or unbounded, or PROC OPTQP was not able to solve the problem.

The following values can appear only if **IIS=ON**. See the section “[Irreducible Infeasible Set](#)” on page 692 for details.

- I_L The lower bound of the variable is needed for the IIS.
- I_U The upper bound of the variable is needed for the IIS.
- I_F Both bounds of the variable are needed for the IIS (the variable is fixed or has conflicting bounds).

Definitions of Variables in the DUALOUT= Data Set

The DUALOUT= data set contains the dual solution to the QP model. Information about the objective rows of the QP problems is not included. The variables in the data set have the following names and meanings.

_OBJ_ID_

specifies the name of the objective function. Naming objective functions is particularly useful when there are multiple objective functions, in which case each objective function has a unique name. See the section “[ROWS Section](#)” on page 817 for details.

NOTE: PROC OPTQP does not support simultaneous optimization of multiple objective functions in this release.

_RHS_ID_

specifies the name of the variable that contains the right-hand-side value of each constraint. See the section “[RHS Section \(Optional\)](#)” on page 819 for details.

ROW

specifies the name of the constraint. See the section “[ROWS Section](#)” on page 817 for details.

TYPE

specifies the type of the constraint. **_TYPE_** can take one of the following values:

- L “less than or equals” constraint
- E equality constraint
- G “greater than or equals” constraint
- R ranged constraint (both “less than or equals” and “greater than or equals”)

See the sections “[ROWS Section](#)” on page 817 and “[RANGES Section \(Optional\)](#)” on page 820 for details.

RHS

specifies the value of the right-hand side of the constraint. It takes a missing value for a ranged constraint.

_L_RHS_

specifies the lower bound of a ranged constraint. It takes a missing value for a non-ranged constraint.

_U_RHS_

specifies the upper bound of a ranged constraint. It takes a missing value for a non-ranged constraint.

VALUE

specifies the value of the dual variable associated with the constraint.

STATUS

specifies the status of the constraint. **_STATUS_** can indicate one of the following two cases:

- O The QP problem is optimal.
- I The QP problem could be infeasible or unbounded, or PROC OPTQP was not able to solve the problem.

The following values can appear only if option **IIS=ON**. See the section “[Irreducible Infeasible Set](#)” on page 692 for details.

I_L The “GE” (\geq) condition of the constraint is needed for the IIS.

I_U The “LE” (\leq) condition of the constraint is needed for the IIS.

I_F Both conditions of the constraint are needed for the IIS (the constraint is an equality or a range constraint with conflicting bounds).

ACTIVITY

specifies the value of a constraint. In other words, the value of **_ACTIVITY_** for the i th constraint is equal to $\mathbf{a}_i^T \mathbf{x}$, where \mathbf{a}_i refers to the i th row of the constraints matrix and \mathbf{x} denotes the vector of current decision variable values.

Interior Point Algorithm: Overview

The interior point solver in PROC OPTQP implements an infeasible primal-dual predictor-corrector interior point algorithm. To illustrate the algorithm and the concepts of duality and dual infeasibility, consider the following QP formulation (the primal):

$$\begin{array}{ll} \min & \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \mathbf{A} \mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

The corresponding dual is as follows:

$$\begin{array}{ll} \max & -\frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{b}^T \mathbf{y} \\ \text{subject to} & -\mathbf{Q} \mathbf{x} + \mathbf{A}^T \mathbf{y} + \mathbf{w} = \mathbf{c} \\ & \mathbf{y} \geq \mathbf{0} \\ & \mathbf{w} \geq \mathbf{0} \end{array}$$

where $\mathbf{y} \in \mathbb{R}^m$ refers to the vector of dual variables and $\mathbf{w} \in \mathbb{R}^n$ refers to the vector of slack variables in the dual problem.

The dual makes an important contribution to the certificate of optimality for the primal. The primal and dual constraints combined with complementarity conditions define the first-order optimality conditions, also known as KKT (Karush-Kuhn-Tucker) conditions, which can be stated as follows:

$$\begin{aligned} \mathbf{Ax} - \mathbf{s} &= \mathbf{b} && \text{(primal feasibility)} \\ -\mathbf{Qx} + \mathbf{A}^T \mathbf{y} + \mathbf{w} &= \mathbf{c} && \text{(dual feasibility)} \\ \mathbf{W}\mathbf{X}\mathbf{e} &= \mathbf{0} && \text{(complementarity)} \\ \mathbf{S}\mathbf{Y}\mathbf{e} &= \mathbf{0} && \text{(complementarity)} \\ \mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{s} &\geq \mathbf{0} \end{aligned}$$

where $\mathbf{e} \equiv (1, \dots, 1)^T$ is of appropriate dimension and $\mathbf{s} \in \mathbb{R}^m$ is the vector of primal slack variables.

NOTE: Slack variables (the \mathbf{s} vector) are automatically introduced by the solver when necessary; it is therefore recommended that you not introduce any slack variables explicitly. This enables the solver to handle slack variables much more efficiently.

The letters \mathbf{X} , \mathbf{Y} , \mathbf{W} , and \mathbf{S} denote matrices with corresponding x , y , w , and s on the main diagonal and zero elsewhere, as in the following example:

$$\mathbf{X} \equiv \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_n \end{bmatrix}$$

If $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{w}^*, \mathbf{s}^*)$ is a solution of the previously defined system of equations that represent the KKT conditions, then \mathbf{x}^* is also an optimal solution to the original QP model.

At each iteration the interior point algorithm solves a large, sparse system of linear equations as follows:

$$\begin{bmatrix} \mathbf{Y}^{-1}\mathbf{S} & \mathbf{A} \\ \mathbf{A}^T & -\mathbf{Q} - \mathbf{X}^{-1}\mathbf{W} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{y} \\ \Delta \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{\Xi} \\ \mathbf{\Theta} \end{bmatrix}$$

where $\Delta \mathbf{x}$ and $\Delta \mathbf{y}$ denote the vector of *search directions* in the primal and dual spaces, respectively, and $\mathbf{\Theta}$ and $\mathbf{\Xi}$ constitute the vector of the right-hand sides.

The preceding system is known as the reduced KKT system. PROC OPTQP uses a preconditioned quasi-minimum residual algorithm to solve this system of equations efficiently.

An important feature of the interior point solver is that it takes full advantage of the sparsity in the constraint and quadratic matrices, thereby enabling it to efficiently solve large-scale quadratic programs.

The interior point algorithm works simultaneously in the primal and dual spaces. It attains optimality when both primal and dual feasibility are achieved and when complementarity conditions hold. Therefore, it is of interest to observe the following four measures where $\|v\|_2$ is the Euclidean norm of the vector v :

- relative primal infeasibility measure α :

$$\alpha = \frac{\|\mathbf{Ax} - \mathbf{b} - \mathbf{s}\|_2}{\|\mathbf{b}\|_2 + 1}$$

- relative dual infeasibility measure β :

$$\beta = \frac{\|Q\mathbf{x} + \mathbf{c} - \mathbf{A}^T\mathbf{y} - \mathbf{w}\|_2}{\|\mathbf{c}\|_2 + 1}$$

- relative duality gap δ :

$$\delta = \frac{|\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{y}|}{|\frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}| + 1}$$

- absolute complementarity γ :

$$\gamma = \sum_{i=1}^n x_i w_i + \sum_{i=1}^m y_i s_i$$

These measures are displayed in the iteration log.

Iteration Log for the OPTQP Procedure

The interior point solver in PROC OPTQP implements an infeasible primal-dual predictor-corrector interior point algorithm. The following information is displayed in the iteration log:

| | |
|---------------|--|
| Iter | indicates the iteration number. |
| Complement | indicates the (absolute) complementarity. |
| Duality Gap | indicates the (relative) duality gap. |
| Primal Infeas | indicates the (relative) primal infeasibility measure. |
| Bound Infeas | indicates the (relative) bound infeasibility measure. |
| Dual Infeas | indicates the (relative) dual infeasibility measure. |
| Time | indicates the time elapsed (in seconds). |

If the sequence of solutions converges to an optimal solution of the problem, you should see all columns in the iteration log converge to zero or very close to zero. Nonconvergence can be the result of insufficient iterations being performed to reach optimality. In this case, you might need to increase the value that you specify in the **MAXITER=** or **MAXTIME=** option. If the complementarity or the duality gap does not converge, the problem might be infeasible or unbounded. If the infeasibility columns do not converge, the problem might be infeasible.

ODS Tables

PROC OPTQP creates two Output Delivery System (ODS) tables by default: the ProblemSummary table is a summary of the input QP problem, and the SolutionSummary table is a brief summary of the solution status.

You can use ODS table names to select tables and create output data sets. For more information about ODS, see the *SAS Output Delivery System: User's Guide*.

If you specify a value of 2 for the **PRINTLEVEL=** option, then the ProblemStatistics table is produced. This table contains information about the problem data. See the section “[Problem Statistics](#)” on page 692 for more information.

[Table 14.2](#) lists all the ODS tables that can be produced by the OPTQP procedure, along with the statement and option specifications required to produce each table.

Table 14.2 ODS Tables Produced by PROC OPTQP

| ODS Table Name | Description | Statement | Option |
|-------------------|-----------------------------------|------------|------------------------|
| ProblemSummary | Summary of the input QP problem | PROC OPTQP | PRINTLEVEL=1 (default) |
| SolutionSummary | Summary of the solution status | PROC OPTQP | PRINTLEVEL=1 (default) |
| ProblemStatistics | Description of input problem data | PROC OPTQP | PRINTLEVEL=2 |
| Timing | Summary of time consumption | PROC OPTQP | PRINTLEVEL=2 |

A typical output of PROC OPTQP is shown in [Output 14.5](#).

Figure 14.5 Typical OPTQP Output

The OPTQP Procedure

| Problem Summary | |
|---------------------------------|--------------|
| Problem Name | BANDM |
| Objective Sense | Minimization |
| Objective Function |1 |
| RHS | ZZZZ0001 |
| | |
| Number of Variables | 472 |
| Bounded Above | 0 |
| Bounded Below | 472 |
| Bounded Above and Below | 0 |
| Free | 0 |
| Fixed | 0 |
| | |
| Number of Constraints | 305 |
| LE (<=) | 0 |
| EQ (=) | 305 |
| GE (>=) | 0 |
| Range | 0 |
| | |
| Constraint Coefficients | 2494 |
| | |
| Hessian Diagonal Elements | 25 |
| Hessian Elements Below Diagonal | 16 |

Figure 14.5 *continued*

| Solution Summary | |
|-----------------------------|----------------|
| Solver | QP |
| Algorithm | Interior Point |
| Objective Function |1 |
| Solution Status | Optimal |
| Objective Value | 16352.342037 |
| Primal Infeasibility | 2.627791E-11 |
| Dual Infeasibility | 8.144725E-16 |
| Bound Infeasibility | 0 |
| Duality Gap | 1.592006E-10 |
| Complementarity | 1.376135E-7 |
| Iterations | 22 |
| Presolve Time | 0.00 |
| Solution Time | 0.04 |

You can create output data sets from these tables by using the ODS OUTPUT statement. This can be useful, for example, when you want to create a report to summarize multiple PROC OPTQP runs. The output data sets that correspond to the preceding output are shown in [Output 14.6](#), where you can also find (in the row following the heading of each data set in the display) the variable names that are used in the table definition (template) of each table.

Figure 14.6 ODS Output Data Sets**Problem Summary**

| Obs | Label1 | cValue1 | nValue1 |
|-----|---------------------------------|--------------|-------------|
| 1 | Problem Name | BANDM | . |
| 2 | Objective Sense | Minimization | . |
| 3 | Objective Function |1 | . |
| 4 | RHS | ZZZZ0001 | . |
| 5 | | | . |
| 6 | Number of Variables | 472 | 472.000000 |
| 7 | Bounded Above | 0 | 0 |
| 8 | Bounded Below | 472 | 472.000000 |
| 9 | Bounded Above and Below | 0 | 0 |
| 10 | Free | 0 | 0 |
| 11 | Fixed | 0 | 0 |
| 12 | | | . |
| 13 | Number of Constraints | 305 | 305.000000 |
| 14 | LE (<=) | 0 | 0 |
| 15 | EQ (=) | 305 | 305.000000 |
| 16 | GE (>=) | 0 | 0 |
| 17 | Range | 0 | 0 |
| 18 | | | . |
| 19 | Constraint Coefficients | 2494 | 2494.000000 |
| 20 | | | . |
| 21 | Hessian Diagonal Elements | 25 | 25.000000 |
| 22 | Hessian Elements Below Diagonal | 16 | 16.000000 |

Solution Summary

| Obs | Label1 | cValue1 | nValue1 |
|-----|----------------------|----------------|--------------|
| 1 | Solver | QP | . |
| 2 | Algorithm | Interior Point | . |
| 3 | Objective Function |1 | . |
| 4 | Solution Status | Optimal | . |
| 5 | Objective Value | 16352.342037 | 16352 |
| 6 | | | . |
| 7 | Primal Infeasibility | 2.627791E-11 | 2.627791E-11 |
| 8 | Dual Infeasibility | 8.144725E-16 | 8.144725E-16 |
| 9 | Bound Infeasibility | 0 | 0 |
| 10 | Duality Gap | 1.592006E-10 | 1.592006E-10 |
| 11 | Complementarity | 1.376135E-7 | 0.000000138 |
| 12 | | | . |
| 13 | Iterations | 22 | 22.000000 |
| 14 | Presolve Time | 0.00 | 0 |
| 15 | Solution Time | 0.04 | 0.040799 |

Problem Statistics

Optimizers can encounter difficulty when solving poorly formulated models. Information about data magnitude provides a simple gauge to determine how well a model is formulated. For example, a model whose constraint matrix contains one very large entry (on the order of 10^9) can cause difficulty when the remaining entries are single-digit numbers. The `PRINTLEVEL=2` option in the OPTQP procedure causes the ODS table ProblemStatistics to be generated. This table provides basic data magnitude information that enables you to improve the formulation of your models.

The example output in [Output 14.7](#) demonstrates the contents of the ODS table ProblemStatistics.

Figure 14.7 ODS Table ProblemStatistics

| The OPTQP Procedure | |
|--|------|
| Problem Statistics | |
| Number of Constraint Matrix Nonzeros | 4 |
| Maximum Constraint Matrix Coefficient | 2 |
| Minimum Constraint Matrix Coefficient | 1 |
| Average Constraint Matrix Coefficient | 1.25 |
| Number of Linear Objective Nonzeros | 2 |
| Maximum Linear Objective Coefficient | 3 |
| Minimum Linear Objective Coefficient | 2 |
| Average Linear Objective Coefficient | 2.5 |
| Number of Nonzeros Below Diagonal in the Hessian | 1 |
| Number of Diagonal Nonzeros in the Hessian | 2 |
| Maximum Hessian Coefficient | 20 |
| Minimum Hessian Coefficient | 2 |
| Average Hessian Coefficient | 6.75 |
| Number of RHS Nonzeros | 2 |
| Maximum RHS | 100 |
| Minimum RHS | 1 |
| Average RHS | 50.5 |
| Maximum Number of Nonzeros per Column | 2 |
| Minimum Number of Nonzeros per Column | 2 |
| Average Number of Nonzeros per Column | 2 |
| Maximum Number of Nonzeros per Row | 2 |
| Minimum Number of Nonzeros per Row | 2 |
| Average Number of Nonzeros per Row | 2 |

Irreducible Infeasible Set

For a quadratic programming problem, an irreducible infeasible set (IIS) is an infeasible subset of constraints and variable bounds that becomes feasible if any single constraint or variable bound is removed. It is possible

to have more than one IIS in an infeasible QP. Identifying an IIS can help isolate the structural infeasibility in a QP. The **IIS=ON** option directs the OPTQP procedure to search for an IIS in a specified QP.

Whether a quadratic programming problem is feasible or infeasible is determined by its constraints and variable bounds, which have nothing to do with its objective function. When you specify the **IIS=ON** option, the OPTQP procedure treats this problem as a linear programming problem by ignoring its objective function. Then finding IIS is the same as what PROC OPTLP does with the **IIS=ON** option. See the section “**Irreducible Infeasible Set**” on page 584 in Chapter 12, “**The OPTLP Procedure**,” for more information about the irreducible infeasible set.

Macro Variable _OROPTQP_

The OPTQP procedure defines a macro variable named `_OROPTQP_`. This variable contains a character string that indicates the status of the procedure. The various terms of the variable are interpreted as follows.

STATUS

indicates the solver status at termination. It can take one of the following values:

| | |
|---------------|---|
| OK | The procedure terminated normally. |
| SYNTAX_ERROR | Incorrect syntax was used. |
| DATA_ERROR | The input data were inconsistent. |
| OUT_OF_MEMORY | Insufficient memory was allocated to the procedure. |
| IO_ERROR | A problem occurred in reading or writing data. |
| ERROR | The status cannot be classified into any of the preceding categories. |

ALGORITHM

indicates the algorithm that produced the solution data in the macro variable. This term appears only when **STATUS=OK**. It can take the following value:

| | |
|----|--|
| IP | The interior point algorithm produced the solution data. |
|----|--|

SOLUTION_STATUS

indicates the solution status at termination. It can take one of the following values:

| | |
|-------------------------|--|
| OPTIMAL | The solution is optimal. |
| CONDITIONAL_OPTIMAL | The solution is optimal, but some infeasibilities (primal, dual or bound) exceed tolerances due to scaling or preprocessing. |
| INFEASIBLE | The problem is infeasible. |
| UNBOUNDED | The problem is unbounded. |
| INFEASIBLE_OR_UNBOUNDED | The problem is infeasible or unbounded. |
| ITERATION_LIMIT_REACHED | The maximum allowable number of iterations was reached. |
| TIME_LIMIT_REACHED | The maximum time limit was reached. |
| ABORTED | The solver was interrupted externally. |

| | |
|------------|--|
| FAILED | The solver failed to converge, possibly due to numerical issues. |
| NONCONVEX | The quadratic matrix is nonconvex (minimization). |
| NONCONCAVE | The quadratic matrix is nonconcave (maximization). |

OBJECTIVE

indicates the objective value obtained by the solver at termination.

PRIMAL_INFEASIBILITY

indicates the (relative) infeasibility of the primal constraints at the solution. For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686.

DUAL_INFEASIBILITY

indicates the (relative) infeasibility of the dual constraints at the solution. For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686.

BOUND_INFEASIBILITY

indicates the (relative) violation by the solution of the lower or upper bounds (or both). For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686.

DUALITY_GAP

indicates the (relative) duality gap. For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686.

COMPLEMENTARITY

indicates the (absolute) complementarity at the solution. For more information, see the section “[Interior Point Algorithm: Overview](#)” on page 686.

ITERATIONS

indicates the number of iterations taken to solve the problem.

PRESOLVE_TIME

indicates the time (in seconds) taken for preprocessing.

SOLUTION_TIME

indicates the time (in seconds) taken to solve the problem, including preprocessing time.

NOTE: The time that is reported in PRESOLVE_TIME and SOLUTION_TIME is either CPU time or real time. The type is determined by the [TIMETYPE=](#) option.

Examples: OPTQP Procedure

This section contains examples that illustrate the use of the OPTQP procedure. [Example 14.1](#) illustrates how to model a linear least squares problem and solve it by using PROC OPTQP. [Example 14.2](#) and [Example 14.3](#) explain in detail how to model the portfolio optimization and selection problems.

Example 14.1: Linear Least Squares Problem

The linear least squares problem arises in the context of determining a solution to an overdetermined set of linear equations. In practice, these equations could arise in data fitting and estimation problems. An overdetermined system of linear equations can be defined as

$$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $m > n$. Since this system usually does not have a solution, you need to be satisfied with some sort of approximate solution. The most widely used approximation is the least squares solution, which minimizes $\|\mathbf{Ax} - \mathbf{b}\|_2^2$.

This problem is called a least squares problem for the following reason. Let \mathbf{A} , \mathbf{x} , and \mathbf{b} be defined as previously. Let $k_i(x)$ be the i th component of the vector $\mathbf{Ax} - \mathbf{b}$:

$$k_i(x) = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - b_i, \quad i = 1, 2, \dots, m$$

By definition of the Euclidean norm, the objective function can be expressed as follows:

$$\|\mathbf{Ax} - \mathbf{b}\|_2^2 = \sum_{i=1}^m k_i(x)^2$$

Therefore, the function you minimize is the sum of squares of m terms $k_i(x)$; hence the term least squares. The following example is an illustration of the *linear* least squares problem; that is, each of the terms k_i is a linear function of x .

Consider the following least squares problem defined by

$$\mathbf{A} = \begin{bmatrix} 4 & 0 \\ -1 & 1 \\ 3 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

This translates to the following set of linear equations:

$$4x_1 = 1, \quad -x_1 + x_2 = 0, \quad 3x_1 + 2x_2 = 1$$

The corresponding least squares problem is

$$\text{minimize} \quad (4x_1 - 1)^2 + (-x_1 + x_2)^2 + (3x_1 + 2x_2 - 1)^2$$

The preceding objective function can be expanded to

$$\text{minimize} \quad 26x_1^2 + 5x_2^2 + 10x_1x_2 - 14x_1 - 4x_2 + 2$$

In addition, you impose the following constraint so that the equation $3x_1 + 2x_2 = 1$ is satisfied within a tolerance of 0.1:

$$0.9 \leq 3x_1 + 2x_2 \leq 1.1$$

You can create the QPS-format input data set by using the following SAS statements:

```

data lsdata;
  input field1 $ field2 $ field3 $ field4 field5 $ field6 @;
  datalines;
NAME      .      LEASTSQ      .      .      .
ROWS      .      .      .      .      .
N          OBJ      .      .      .      .
G          EQ3      .      .      .      .
COLUMNS  .      .      .      .      .
.          X1      OBJ      -14      EQ3      3
.          X2      OBJ      -4       EQ3      2
RHS        .      .      .      .      .
.          RHS      OBJ      -2       EQ3      0.9
RANGES    .      .      .      .      .
.          RNG      EQ3      0.2      .      .
BOUNDS     .      .      .      .      .
FR         BND1     X1      .      .      .
FR         BND1     X2      .      .      .
QUADOBJ    .      .      .      .      .
.          X1      X1      52      .      .
.          X1      X2      10      .      .
.          X2      X2      10      .      .
ENDATA     .      .      .      .      .
;

```

The decision variables x_1 and x_2 are free, so they have bound type FR in the BOUNDS section of the QPS-format data set.

You can use the following SAS statements to solve the least squares problem:

```

proc optqp data=lsdata
  printlevel = 0
  primalout = lspout;
run;

```

The optimal solution is displayed in [Output 14.1.1](#).

Output 14.1.1 Solution to the Least Squares Problem

Primal Solution

| Obs | Objective | | Variable Name | Variable Type | Linear | | Upper Bound | Variable Value | Variable Status |
|-----|-------------|--------|---------------|---------------|-----------------------|-------------|-------------|----------------|-----------------|
| | Function ID | RHS ID | | | Objective Coefficient | Lower Bound | | | |
| 1 | OBJ | RHS | X1 | F | -14 | -1.7977E308 | 1.7977E308 | 0.23810 | O |
| 2 | OBJ | RHS | X2 | F | -4 | -1.7977E308 | 1.7977E308 | 0.16190 | O |

The iteration log is shown in [Output 14.1.2](#).

Output 14.1.2 Iteration Log

```

NOTE: The problem LEASTSQ has 2 variables (2 free, 0 fixed).
NOTE: The problem has 1 constraints (0 LE, 0 EQ, 0 GE, 1 range).
NOTE: The problem has 2 constraint coefficients.
NOTE: The objective function has 2 Hessian diagonal elements and 1 Hessian
      elements above the diagonal.
NOTE: The QP presolver value AUTOMATIC is applied.
NOTE: The QP presolver removed 0 variables and 0 constraints.
NOTE: The QP presolver removed 0 constraint coefficients.
NOTE: The presolved problem has 2 variables, 1 constraints, and 2 constraint
      coefficients.
NOTE: The QP solver is called.
NOTE: The Interior Point algorithm is used.
NOTE: The deterministic parallel mode is enabled.
NOTE: The Interior Point algorithm is using up to 4 threads.

      Iter  Complement  Duality Gap      Primal      Bound      Dual
            0  4.4635E-02  7.3436E-03  1.2741E-12  1.1785E-01  4.8074E-14  0
            1  6.0962E-03  2.0162E-03  1.1813E-11  1.1785E-03  4.9674E-16  0
            2  2.0205E-04  6.7545E-05  1.1113E-10  2.4631E-05  6.2541E-17  0
            3  2.0975E-06  7.0139E-07  1.3791E-11  2.4631E-07  3.3605E-17  0
            4  1.8029E-06  5.5869E-07  1.3791E-11  2.0900E-07  1.3713E-07  0
            5  0.0000E+00  9.4105E-08  2.7318E-13  2.0900E-09  6.2816E-08  0

NOTE: Optimal.
NOTE: Objective = 0.0095238095.
NOTE: The Interior Point solve time is 0.00 seconds.
NOTE: There were 19 observations read from the data set WORK.LSDATA.
NOTE: The data set WORK.LSPOUT has 2 observations and 9 variables.

```

Example 14.2: Portfolio Optimization

Consider a portfolio optimization example. The two competing goals of investment are (1) long-term growth of capital and (2) low risk. A good portfolio grows steadily without wild fluctuations in value. The Markowitz model is an optimization model for balancing the return and risk of a portfolio. The decision variables are the amounts invested in each asset. The objective is to minimize the variance of the portfolio's total return, subject to the constraints that (1) the expected growth of the portfolio reaches at least some target level and (2) you do not invest more capital than you have.

Let x_1, \dots, x_n be the amount invested in each asset, \mathcal{B} be the amount of capital you have, \mathbf{R} be the random vector of asset returns over some period, and \mathbf{r} be the expected value of \mathbf{R} . Let G be the minimum growth you hope to obtain, and \mathcal{C} be the covariance matrix of \mathbf{R} . The objective function is $\text{Var} \left(\sum_{i=1}^n x_i R_i \right)$, which can be equivalently denoted as $\mathbf{x}^T \mathcal{C} \mathbf{x}$.

Assume, for example, $n = 4$. Let $B = 10,000$, $G = 1000$, $r = [0.05, -0.2, 0.15, 0.30]$, and

$$C = \begin{bmatrix} 0.08 & -0.05 & -0.05 & -0.05 \\ -0.05 & 0.16 & -0.02 & -0.02 \\ -0.05 & -0.02 & 0.35 & 0.06 \\ -0.05 & -0.02 & 0.06 & 0.35 \end{bmatrix}$$

The QP formulation can be written as follows:

$$\begin{aligned} \min \quad & 0.08x_1^2 - 0.1x_1x_2 - 0.1x_1x_3 - 0.1x_1x_4 + 0.16x_2^2 \\ & - 0.04x_2x_3 - 0.04x_2x_4 + 0.35x_3^2 + 0.12x_3x_4 + 0.35x_4^2 \\ \text{subject to} \quad & \\ \text{(budget)} \quad & x_1 + x_2 + x_3 + x_4 \leq 10000 \\ \text{(growth)} \quad & 0.05x_1 - 0.2x_2 + 0.15x_3 + 0.30x_4 \geq 1000 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

The corresponding QPS-format input data set is as follows:

```
data portdata;
  input field1 $ field2 $ field3 $ field4 field5 $ field6 @;
datalines;
NAME . PORT . . .
ROWS . . . .
N OBJ.FUNC . . . .
L BUDGET . . . .
G GROWTH . . . .
COLUMNS . . . .
. X1 BUDGET 1.0 GROWTH 0.05
. X2 BUDGET 1.0 GROWTH -.20
. X3 BUDGET 1.0 GROWTH 0.15
. X4 BUDGET 1.0 GROWTH 0.30
RHS . . . .
. RHS BUDGET 10000 . .
. RHS GROWTH 1000 . .
RANGES . . . .
BOUNDS . . . .
QUADOBJ . . . .
. X1 X1 0.16 . .
. X1 X2 -.10 . .
. X1 X3 -.10 . .
. X1 X4 -.10 . .
. X2 X2 0.32 . .
. X2 X3 -.04 . .
. X2 X4 -.04 . .
. X3 X3 0.70 . .
. X3 X4 0.12 . .
. X4 X4 0.70 . .
ENDATA . . . .
;
```

Use the following SAS statements to solve the problem:

```
proc optqp data=portdata
  primalout = portpout
  printlevel = 0
  dualout   = portdout;
run;
```

The optimal solution is shown in [Output 14.2.1](#).

Output 14.2.1 Portfolio Optimization

The OPTQP Procedure Primal Solution

| Obs | Objective Function | | RHS ID | Variable Name | Variable Type | Linear | | Upper Bound | Variable Value | Variable Status |
|-----|--------------------|-----|--------|---------------|---------------|-----------------------|-------------|-------------|----------------|-----------------|
| | ID | | | | | Objective Coefficient | Lower Bound | | | |
| 1 | OBJ.FUNC | RHS | X1 | N | | 0 | 0 | 1.7977E308 | 3452.86 | O |
| 2 | OBJ.FUNC | RHS | X2 | N | | 0 | 0 | 1.7977E308 | 0.00 | O |
| 3 | OBJ.FUNC | RHS | X3 | N | | 0 | 0 | 1.7977E308 | 1068.81 | O |
| 4 | OBJ.FUNC | RHS | X4 | N | | 0 | 0 | 1.7977E308 | 2223.45 | O |

Thus, the minimum variance portfolio that earns an expected return of at least 10% is $x_1 = 3452.86$, $x_2 = 0$, $x_3 = 1068.81$, $x_4 = 2223.45$. Asset 2 gets nothing, because its expected return is -20% and its covariance with the other assets is not sufficiently negative for it to bring any diversification benefits. What if you drop the nonnegativity assumption? You need to update the BOUNDS section in the existing QPS-format data set to indicate that the decision variables are free.

```
...
RANGES . . . .
BOUNDS . . . .
FR BND1 X1 . .
FR BND1 X2 . .
FR BND1 X3 . .
FR BND1 X4 . .
QUADOBJ . . . .
...
```

Financially, that means you are allowed to short-sell—that is, sell low-mean-return assets and use the proceeds to invest in high-mean-return assets. In other words, you put a negative portfolio weight in low-mean assets and “more than 100%” in high-mean assets. You can see in the optimal solution displayed in [Output 14.2.2](#) that the decision variable x_2 , denoting Asset 2, is equal to -1563.61 , which means short sale of that asset.

Output 14.2.2 Portfolio Optimization with Short-Sale Option**The OPTQP Procedure
Primal Solution**

| Obs | Objective Function | | RHS ID | Variable Name | Variable Type | Linear Objective | | Upper Bound | Variable Value | Variable Status |
|-----|--------------------|-----|--------|---------------|---------------|------------------|-------------|-------------|----------------|-----------------|
| | ID | | | | | Coefficient | Lower Bound | | | |
| 1 | OBJ.FUNC | RHS | X1 | F | | 0 | -1.7977E308 | 1.7977E308 | 1684.35 | O |
| 2 | OBJ.FUNC | RHS | X2 | F | | 0 | -1.7977E308 | 1.7977E308 | -1563.61 | O |
| 3 | OBJ.FUNC | RHS | X3 | F | | 0 | -1.7977E308 | 1.7977E308 | 682.51 | O |
| 4 | OBJ.FUNC | RHS | X4 | F | | 0 | -1.7977E308 | 1.7977E308 | 1668.95 | O |

Example 14.3: Portfolio Selection with Transactions

Consider a portfolio selection problem with a slight modification. You are now required to take into account the current position and transaction costs associated with buying and selling assets. The objective is to find the minimum variance portfolio. In order to understand the scenario better, consider the following data.

You are given three assets. The current holding of the three assets is denoted by the vector $\mathbf{c} = [200, 300, 500]$, the amount of asset bought and sold is denoted by b_i and s_i , respectively, and the net investment in each asset is denoted by x_i and is defined by the following relation:

$$x_i - b_i + s_i = c_i, \quad i = 1, 2, 3$$

Suppose you pay a transaction fee of 0.01 every time you buy or sell. Let the covariance matrix \mathcal{C} be defined as

$$\mathcal{C} = \begin{bmatrix} 0.027489 & -0.00874 & -0.00015 \\ -0.00874 & 0.109449 & -0.00012 \\ -0.00015 & -0.00012 & 0.000766 \end{bmatrix}$$

Assume that you hope to obtain at least 12% growth. Let $\mathbf{r} = [1.109048, 1.169048, 1.074286]$ be the vector of expected return on the three assets, and let $\mathcal{B}=1000$ be the available funds. Mathematically, this problem can be written in the following manner:

$$\begin{aligned} \min \quad & 0.027489x_1^2 - 0.01748x_1x_2 - 0.0003x_1x_3 + 0.109449x_2^2 \\ & - 0.00024x_2x_3 + 0.000766x_3^2 \end{aligned}$$

subject to

$$\text{(return)} \quad \sum_{i=1}^3 r_i x_i \geq 1.12\mathcal{B}$$

$$\text{(budget)} \quad \sum_{i=1}^3 x_i + \sum_{i=1}^3 0.01(b_i + s_i) = \mathcal{B}$$

$$\text{(balance)} \quad x_i - b_i + s_i = c_i, \quad i = 1, 2, 3$$

$$x_i, b_i, s_i \geq 0, \quad i = 1, 2, 3$$

The QPS-format input data set is as follows:

```

data potrdata;
  input field1 $ field2 $ field3 $ field4 field5 $ field6 @;
datalines;
NAME      .      POTRAN      .      .      .
ROWS      .      .      .      .      .
N      OBJ.FUNC      .      .      .      .
G      RETURN      .      .      .      .
E      BUDGET      .      .      .      .
E      BALANC1      .      .      .      .
E      BALANC2      .      .      .      .
E      BALANC3      .      .      .      .
COLUMNS  .      .      .      .      .
.      X1      RETURN      1.109048      BUDGET      1.0
.      X1      BALANC1      1.0      .      .
.      X2      RETURN      1.169048      BUDGET      1.0
.      X2      BALANC2      1.0      .      .
.      X3      RETURN      1.074286      BUDGET      1.0
.      X3      BALANC3      1.0      .      .
.      B1      BUDGET      .01      BALANC1      -1.0
.      B2      BUDGET      .01      BALANC2      -1.0
.      B3      BUDGET      .01      BALANC3      -1.0
.      S1      BUDGET      .01      BALANC1      1.0
.      S2      BUDGET      .01      BALANC2      1.0
.      S3      BUDGET      .01      BALANC3      1.0
RHS      .      .      .      .      .
.      RHS      RETURN      1120      .      .
.      RHS      BUDGET      1000      .      .
.      RHS      BALANC1      200      .      .
.      RHS      BALANC2      300      .      .
.      RHS      BALANC3      500      .      .
RANGES  .      .      .      .      .
BOUNDS  .      .      .      .      .
QUADOBJ  .      .      .      .      .
.      X1      X1      0.054978      .      .
.      X1      X2      -.01748      .      .
.      X1      X3      -.0003      .      .
.      X2      X2      0.218898      .      .
.      X2      X3      -.00024      .      .
.      X3      X3      0.001532      .      .
ENDATA  .      .      .      .      .
;

```

Use the following SAS statements to solve the problem:

```

proc optqp data=potrdata
  primalout = potrpout
  printlevel = 0
  dualout   = potrdout;
run;

```

The optimal solution is displayed in [Output 14.3.1](#).

Output 14.3.1 Portfolio Selection with Transactions

The OPTQP Procedure Primal Solution

| Obs | Objective | RHS | Variable | Variable | Linear | | Upper | Variable | Variable |
|-----|-----------|-----|----------|----------|-------------|-------|------------|----------|----------|
| | Function | | | | Objective | Lower | | | |
| ID | ID | ID | Name | Type | Coefficient | Bound | Bound | Value | Status |
| 1 | OBJ.FUNC | RHS | X1 | N | 0 | 0 | 1.7977E308 | 397.584 | O |
| 2 | OBJ.FUNC | RHS | X2 | N | 0 | 0 | 1.7977E308 | 406.115 | O |
| 3 | OBJ.FUNC | RHS | X3 | N | 0 | 0 | 1.7977E308 | 190.165 | O |
| 4 | OBJ.FUNC | RHS | B1 | N | 0 | 0 | 1.7977E308 | 197.584 | O |
| 5 | OBJ.FUNC | RHS | B2 | N | 0 | 0 | 1.7977E308 | 106.115 | O |
| 6 | OBJ.FUNC | RHS | B3 | N | 0 | 0 | 1.7977E308 | 0.000 | O |
| 7 | OBJ.FUNC | RHS | S1 | N | 0 | 0 | 1.7977E308 | 0.000 | O |
| 8 | OBJ.FUNC | RHS | S2 | N | 0 | 0 | 1.7977E308 | 0.000 | O |
| 9 | OBJ.FUNC | RHS | S3 | N | 0 | 0 | 1.7977E308 | 309.835 | O |

References

- Freund, R. W. (1991). "On Polynomial Preconditioning and Asymptotic Convergence Factors for Indefinite Hermitian Matrices." *Linear Algebra and Its Applications* 154–156:259–288.
- Freund, R. W., and Jarre, F. (1997). "A QMR-Based Interior Point Algorithm for Solving Linear Programs." *Mathematical Programming* 76:183–210.
- Freund, R. W., and Nachtigal, N. M. (1996). "QMRPACK: A Package of QMR Algorithms." *ACM Transactions on Mathematical Software* 22:46–77.
- Vanderbei, R. J. (1999). "LOQO: An Interior Point Code for Quadratic Programming." *Optimization Methods and Software* 11:451–484.
- Wright, S. J. (1997). *Primal-Dual Interior-Point Methods*. Philadelphia: SIAM.

Subject Index

- `_ACTIVITY_` variable
 - DUALOUT= data set, 686
- DUALOUT= data set
 - OPTQP procedure, 685, 686
 - variables, 685, 686
- IIS
 - PROC OPTQP statement, 681
- IIS option
 - OPTQP procedure, 692
- interior point algorithm
 - complementarity, 688
 - dual infeasibility, 681, 687
 - duality gap, 680, 688
 - overview, 686
 - primal infeasibility, 683, 687
 - stopping criteria, 687
- iteration log
 - LOGFREQ= option, 681
 - LOGLEVEL= option, 681
 - OPTQP procedure, 688
 - stopping criteria, 687
- `_LBOUND_` variable
 - PRIMALOUT= data set, 684
 - `_L_RHS_` variable
 - DUALOUT= data set, 686
- %MPS2SASD
 - MPS2SASD, 677, 680
- OROPTQP
 - `_OROPTQP_`, 693
- `_VAR_` variable
 - PRIMALOUT= data set, 684
 - `_OBJ_ID_` variable
 - DUALOUT= data set, 685
 - PRIMALOUT= data set, 684
- ODS
 - ODS table names, 689
 - PRINTLEVEL= option, 683
 - problem statistics, 692
- ODS table names
 - OPTQP procedure, 688
- OPTQP examples
 - covariance matrix, 697
 - data fitting, 695
 - estimation, 695
 - linear least squares, 695
 - Markowitz model, 697
 - portfolio optimization, 697
 - portfolio selection with transactions, 700
 - short-sell, 699
- OPTQP procedure
 - output data sets, 684
 - definitions of DUALOUT= data set variables, 685, 686
 - definitions of DUALOUT=data set variables, 685, 686
 - definitions of PRIMALOUT= data set variables, 684, 685
 - DUALOUT= data set, 685, 686
 - examples, 694
 - functional summary, 680
 - IIS option, 692
 - interior point algorithm, 686
 - iteration log, 688
 - %MPS2SASD macro, 677, 680
 - ODS table names, 688
 - options, 680
 - `_OROPTQP_` macro variable, 693
 - overview, 673
 - PRIMALOUT= data set, 684, 685
 - QPS format, 676
 - quadratic programming, 673
 - syntax, 679
- overview
 - OPTQP procedure, 673
- positive semidefinite matrix, 674
- PRIMALOUT= data set
 - OPTQP procedure, 684, 685
 - variables, 684, 685
- PROC OPTQP statement
 - dual infeasibility, 681
 - dual output data set, 681
 - duality gap, 680
 - IIS, 681
 - input data table, 680
 - log frequency, 681
 - log level, 681
 - maximum iteration, 681
 - maximum time, 682
 - number of threads, 682
 - objective sense, 682
 - ODS print level, 683

- parallel mode, [682](#)
- presolver level, [682](#)
- primal infeasibility, [683](#)
- time type, [683](#)

- quadratic programming
 - overview, [673](#)
 - quadratic matrix, [674](#)

- _RHS_ variable
 - DUALOUT= data set, [686](#)
 - _RHS_ID_ variable
 - DUALOUT= data set, [685](#)
 - PRIMALOUT= data set, [684](#)
 - _ROW_ variable
 - DUALOUT= data set, [685](#)

- _STATUS_ variable
 - DUALOUT= data set, [686](#)
 - PRIMALOUT= data set, [685](#)

- _TYPE_ variable
 - DUALOUT= data set, [685](#)
 - PRIMALOUT= data set, [684](#)

- _UBOUND_ variable
 - PRIMALOUT= data set, [684](#)

- _U_RHS_ variable
 - DUALOUT= data set, [686](#)

- _VALUE_ variable
 - DUALOUT= data set, [686](#)
 - PRIMALOUT= data set, [685](#)

- _VAR_ variable
 - PRIMALOUT= data set, [684](#)

Syntax Index

DATA= option
 PROC OPTQP statement, 680
DUALITYGAP= option
 PROC OPTQP statement, 680
DUALOUT= option
 PROC OPTQP statement, 681
DUALTOL= option
 PROC OPTQP statement, 681

FEASTOL= option
 PROC OPTQP statement, 683

IIS= option
 PROC OPTQP statement, 681

LOGFREQ= option
 PROC OPTQP statement, 681
LOGLEVEL= option
 PROC OPTQP statement, 681

MAXITER= option
 PROC OPTQP statement, 681
MAXTIME= option
 PROC OPTQP statement, 682

NTHREADS= option
 PROC OPTQP statement, 682
NUMTHREADS= option
 PROC OPTQP statement, 682

OBJSENSE= option
 PROC OPTQP statement, 682
OPTQP procedure, 679
OPTTOL= option
 PROC OPTQP statement, 681

PARALLELMODE= option
 PROC OPTQP statement, 682
PRESOLVER= option
 PROC OPTQP statement, 682
PRIMALOUT= option
 PROC OPTQP statement, 683
PRIMALTOL= option
 PROC OPTQP statement, 683
PRINTFREQ= option
 PROC OPTQP statement, 681
PRINTLEVEL2= option
 PROC OPTQP statement, 681
PRINTLEVEL= option

 PROC OPTQP statement, 683
PROC OPTQP statement
 DATA= option, 680
 DUALITYGAP= option, 680
 DUALOUT= option, 681
 DUALTOL= option, 681
 FEASTOL= option, 683
 IIS= option, 681
 LOGFREQ= option, 681
 LOGLEVEL= option, 681
 MAXITER= option, 681
 MAXTIME= option, 682
 NTHREADS= option, 682
 NUMTHREADS= option, 682
 OBJSENSE= option, 682
 OPTTOL= option, 681
 PARALLELMODE= option, 682
 PRESOLVER= option, 682
 PRIMALOUT= option, 683
 PRIMALTOL= option, 683
 PRINTFREQ= option, 681
 PRINTLEVEL2= option, 681
 PRINTLEVEL= option, 683
 TIMETYPE= option, 683

TIMETYPE= option
 PROC OPTQP statement, 683