NAME

mnsma – CUTEr L-BFGS-B test driver

SYNOPSIS

mnsma

DESCRIPTION

The *mnsma* main program test drives MINOS on SIF problems from the CUTEr distribution.

The MINOS package is designed to minimize smooth functions subject to constraints, which may include simple bounds, linear constraints, and smooth nonlinear constraints. The nonlinear contributions to the constraints are linearized so that linear programming technology can be exploited. MINOS allows matrices to be stored in either dense or sparse format, and is therefore suitable for large sparse problems.

MINOS is distributed by the Office of Technology Licensing (OTL) at Stanford University and is subject to certain license agreements. MINOS is copyrighted by Stanford University. Contact: Michael Saunders (e-mail address: mike@sol-michael.stanford.edu, postal address: Department of Operations Research, Stanford, CA 94305-4022, USA).

The object module *mnsma.o* is stored in $MYCUTER/*precision*/bin, where *precision* is either "single" or "double", according to your local installation.

CREATING THE OBJECT MODULE FOR MINOS 5.4 AND MINOS 5.5

Copy mi15blas.f into subsblas.f (or another name of your choice). Remove the subroutines daxpy, dcopy, ddot, dnrm2 and dscal from subsblas.f to create a new file with a subset of the MINOS BLAS subroutines. In our versions of MINOS 5.4 and MINOS 5.5, this corresponds to deleting lines 70 to 391 in subsblas.f (CUTEr also uses the BLAS subroutines daxpy, dcopy, ddot, dnrm2 and dscal. They are contained in the CUTEr Fortran file named linpac.f. To avoid multiply defined subroutines when MINOS is linked with the CUTEr tools, the duplicate BLAS subroutines must be removed from MINOS.)

You should also edit mi10unix.f or mi10vms.f, depending on which operating system you are using, so that it runs properly on your machine. See the MINOS documentation for details.

Concatenate

(1) *mi10unix.f* if you are using CUTEr on a system other than DEC VAX/VMS, or

(2) *mi10vms.f* if you are using CUTEr on DEC VAX/VMS,

with the fortran files given in *Table 1* into a single file named minos.f.

mi20amat.f	mi60srch.f
mi25bfac.f	mi65rmod.f
mi30spec.f	mi70nobj.f
mi35inpt.f	mi80ncon.f
mi40bfil.f	subsblas.f
mi50lp.f	

Table 1.

Compile (but do not link) minos.f. The resulting object file minos.o (minoss.o) should be placed in $MYCUTER/*precision*/bin/.

CREATING THE OBJECT MODULE FOR MINOS 5.3

Copy MI00MAIN.f into MI00MAIN_f (or another name of your choice, assuming you want to keep the original Fortran source file). Remove the subroutines FUNOBJ, FUNCON and MATMOD from MI00MAIN.f. In our version of MINOS 5.3, this corresponds to deleting lines 62 to 238 in MI00MAIN.f.

Copy MI15BLAS.f into SUBSBLAS.f. Remove the subroutines DAXPY, DCOPY, DDOT, DNRM2 and DSCAL from SUBSBLAS.f to create a new file with a subset of the MINOS BLAS subroutines. In our version of MINOS 5.3, this corresponds to deleting lines 34 to 184 in SUBSBLAS.f (CUTEr also uses the BLAS subroutines DAXPY, DCOPY, DDOT, DNRM2 and DSCAL, but it requires the general versions of these subroutines, rather than the specialized versions contained in MI15BLAS.f. The general versions are contained in the CUTEr Fortran file named linpac.f. To avoid multiply defined subroutines when MINOS is linked with the CUTEr tools, the specialized versions must be removed from MINOS.)

Copy MI10MACH.f into MI10MACH_f (or another name of your choice). Find the following three lines in MI10MACH.f:

```
LINCON = NNCON .EQ. 0
NLNCON = NNCON .GT. 0
NLNOBJ = NNOBJ .GT. 0
```

(They appear in lines 517 to 519 in our version of MINOS 5.3.) Move these three lines so they appear immediately after the call to M3INPT (which appears on line 536 in our version). This change is necessary because we calculate the values of NNCON and NNOBJ for you rather than insisting that you provide the exact values in MINOS.SPC

You should also edit MI10MACH.f so that it compiles and runs properly on your machine. See the MINOS User's Guide for details.

Now you must concatenate the following .f files into a single file named mi53.f:

MI00MAIN.f	MI50LP.f
MI10MACH.f	MI60SRCH.f
MI20AMAT.f	MI65RMOD.f
MI25BFAC.f	MI70NOBJ.f
MI30SPEC.f	MI80NCON.f
MI40BFIL.f	SUBSBLAS.f

Compile (but do not link) mi53.f. The resulting object file mi53.o should be placed in $MYCUTER/*precision*/bin/.

USAGE

If no MINOS.SPC file is present in the current directory, the default version is copied from $MYCUTER/*precision*/specs/.

MINOS.SPC is linked to fort.4. This link is established because MINOS reads the SPECS file from unit 4 (by default). If this unit number has been changed in the MINOS source code, then it must also be changed in the shell script mns.

An input file containing an old basis for the problem (probname.BASIS) is, when it exists, copied to MINOS.BASIS and then linked to fort.11. If this unit number has been changed in the MINOS source code, then it must also be changed in the shell script mns.

An output file containing the new basis for the problem is written by MINOS in fort.12 (by default) and then renamed to MINOS.NEWBASIS in mns. This file is subsequently renamed to probname.NEWBASIS

by sdmns. If this unit number has been changed in the MINOS source code, then it must also be changed in the shell script mns.

The main output from MINOS is written (by default) in a file named fort.9, and then renamed to MINOS.LIS by mns. This file is subsequently renamed to probname.mns by sdmns. If this unit number has been changed in the MINOS source code, then it must also be changed in the shell script mns.

In addition, MINOS sends a summary of the problem output to unit 6 (by default).

THE MINOS SPECIFICATIONS FILE

The file MINOS.SPC constitutes the MINOS SPECS file, which sets various MINOS run-time parameters. This SPECS file is used exactly as described in the MINOS User's Guide, with the following exception: MINOS (when used by itself, not with CUTEr) requires that the user specify exact numbers for the following keywords:

NONLINEAR CONSTRAINTS
NONLINEAR VARIABLES

and optionally also for the following keywords:

NONLINEAR OBJECTIVE VARIABLES
NONLINEAR JACOBIAN VARIABLES

Because the CUTEr tools determine the values of these quantities, the values given in MINOS.SPC need not be exact. Instead, the values given in MINOS.SPC should be at least as large as the true values, in order to guarantee that sufficient space is allotted to various arrays, but they can be much larger than the true values.

A word of caution is in order for the choice of value for NONLINEAR CONSTRAINTS when using MINOS 5.3: Suppose NONLINEAR CONSTRAINTS is set to M1 in MINOS.SPC. If no value is specified for PENALTY PARAMETER in MINOS.SPC, then the initial value of the penalty parameter defaults to 100/M1. Thus, if M1 is significantly larger than the true number of nonlinear constraints, the initial value of the penalty parameter may be unreasonably low. To avoid this situation, you should either be careful not to set NONLINEAR CONSTRAINTS too high above the true value, or explicitly set PENALTY PARAMETER to a reasonable initial value in MINOS.SPC.

The default version of MINOS.SPC sets only a few of the run-time parameters. Many more parameters are available. A full description of the available parameters is given in the SPECS file chapter in the MINOS User's Guide.

NOTE

There is a different default MINOS specifications for each size of CUTEr installation (small, medium, and large). In the CUTEr distribution, these different specifications files are named MINOS.sml, MINOS.med, and MINOS.lrg. At installation time, these three files are copied into $CUTER/common/src/pkg/minos/. The install procedure then casts and copies the specifications file of the appropriate size to MINOS.SPC.

ENVIRONMENT

CUTER

Parent directory for CUTEr

MYCUTER

Home directory of the installed CUTEr distribution.

AUTHORS

I. Bongartz, A.R. Conn, N.I.M. Gould, D. Orban and Ph.L. Toint

SEE ALSO

CUTEr (and SifDec): A Constrained and Unconstrained Testing Environment, revisited,
N.I.M. Gould, D. Orban and Ph.L. Toint,
ACM TOMS, **29**:4, pp.373-394, 2003.

CUTE: Constrained and Unconstrained Testing Environment, I. Bongartz, A.R. Conn, N.I.M. Gould and Ph.L. Toint, TOMS, **21**:1, pp.123-160, 1995.

MINOS 5.1 User's Guide, B.A. Murtagh and M.A. Saunders, Technical Report SOL 83-20R, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305, USA, December 1983, revised January 1987.