NAME

ve12/sdve12 − CUTEr interface with the VE12 solver.

SYNOPSIS

[sd]ve12 [−decode] [−s] [−n] [−h] [−k] [−r] [−o *0|1*] [−l *secs*] [−f] [−b] [−a *1|2*] [−show] [[−param *name=value*[,*name=value...*]] [−force]] [−debug] [−L*path/to/lib*] [−−blas *keyword*] [−−lapack *keyword*] [*problem[.SIF]*]

DESCRIPTION

sdve12 interfaces the VE12 solver with the CUTEr tools and SIF-encoded problems. *sdve12* applies the SIF decoder to the problem *problem.SIF* to produce the OUTSDIF.d file and the problem-dependant Fortan subroutines. If *problem.SIF* has already been decoded, the script *ve12* may be run by itself. VE12 minimizes a quadratic objective subject to linear equality and bound constraints using an active-set method.

sdve12 is only included for backward compatibility with earlier versions of CUTE, and simply calls *ve12* with the −*decode* flag.

ve12 sets the environment and calls *runpackage* to link all the needed libraries and modules for the VE12 solver to solve the problem decoded from the SIF file and generates the executable.

[sd]ve12 Options

You can start [sd]ve12 with the following options:

−decode
Applies the SIF decoder to the problem *problem.SIF* to produce the OUTSDIF.d file and the problem-dependant Fortan subroutines. It is redundant to call *sdve12* with the −decode flag.

−s Run [sd]ve12 in single-precision mode if available. Double precision is the default.

−n Use the load module if it already exists. Compiling a new load module is the default.

−h Print a short help message.

−k Keep the generated executable after use. May be useful when solving a particular problem with the same solver with different parameters. Deleting the executable after use is the default.

−r Discourage recompilation of the test problem. Default is to recompile object files.

−o *0|1*
Regulates the output level of *[sd]ve12*. Verbose mode is **-o** 1, silent mode is **-o** 0. Silent mode is the default.

−l *secs*
sets a limit of *secs* second on VE12 runtime. Unlimited cputime is the default.

−f Use automatic differentiation in Forward mode

−b Use automatic differentiation in Backward mode

−a *1|2*
−a 1 uses the older HSL automatic differentiation package AD01 and −a 2 uses the newer HSL automatic differentiation package AD02. −a 2 is the default.

−show
displays possible parameter settings for problem[.SIF]. Other options are ignored.

−param
cast problem[.SIF] against explicit parameter settings. Several parameter settings may be given as a comma-separated list following −param or using several −param flags. Use *sifdec -show problem* to view possible settings. If a setting is not allowed in the SIF file, no action is taken unless −*force* is present.

−force

Forces the setting of the parameters named using *−param* to the given values, even if those values are not predefined in the SIF file.

−L*path/to/lib*

This option is passed directly to the linker and causes the path *path/to/lib* to be searched for libraries.

−−blas *keyword*

Overrides usage of the default *linpack* library packaged with CUTEr. Instead, use the BLAS library specified by *keyword*. The keyword *keyword* has one of two forms. The first, *-lmyblas* causes the linker to search for BLAS subprograms in the libmyblas.a library. The second, *none*, causes the linker to skip inclusion of any external BLAS. Use the first option if an optimized BLAS library is available on the host system, e.g., the ATLAS BLAS. The second option is useful for packages which already include the necessary BLAS subprograms.

−−lapack *keyword*

Overrides usage of the default *linpack* library packaged with CUTEr. Instead, use the LAPACK library specified by *keyword*. The keyword *keyword* has one of two forms. The first, *-lmylapack* causes the linker to search for LAPACK subroutines in the libmylapack.a library. The second, *none*, causes the linker to skip inclusion of any external LAPACK. Use the first option if an optimized LAPACK library is available on the host system. The second option is useful for packages which already include the necessary LAPACK subprograms.

problem

problem.SIF is the name of the file containing the SIF information on the problem to be solved.

ENVIRONMENT

CUTER

Home directory for CUTEr

MYCUTER

Home directory of the installed CUTEr distribution.

MASTSIF

A pointer to the directory containing the CUTEr problems collection. If this variable is not set, the current directory is searched for *problem.SIF*. If it is set, the current directory is searched first, and if *problem.SIF* is not found there, $MASTSIF is searched.

AUTHORS

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

NOTES

VE12 is part of the Harwell Subroutine Library. It is distributed by the United Kingdom Atomic Energy Authority, Harwell, subject to certain license agreements. It is copyrighted jointly by the UKAEA and the Science and Engineering Research Council. VE12 is authored by N.I.M. Gould.

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.

A primal-dual trust-region algorithm for non-convex nonlinear optimization,
A.R.Conn, N.I.M. Gould, D. Orban and Ph. L. Toint
Mathematical Programming, **87**, pp.215-249,
2000.

sifdecode(1), gen(1).