

**NAME**

CUTEST\_cohprods\_threaded – CUTEst tool to form the matrix-vector product of a vector with the Hessian matrix of the objective function.

**SYNOPSIS**

CALL CUTEST\_cohprods\_threaded( status, goth, X, VECTOR, nnzohp, loh, RESULT, IND, thread )

For real rather than double precision arguments, instead

CALL CUTEST\_cohprods\_threaded\_s( ... )

**DESCRIPTION**

The CUTEST\_cohprods\_threaded subroutine forms the product of a vector with the Hessian matrix of the objective function  $f(x)$  corresponding to the problem decoded from a SIF file by the script *sifdecoder* at the point  $x = X$ .

The problem under consideration is to minimize or maximize an objective function  $f(x)$  over all  $x \in R^n$  subject to general equations  $c_i(x) = 0$ , ( $i \in 1, \dots, m_E$ ), general inequalities  $c_i^l \leq c_i(x) \leq c_i^u$  ( $i \in m_E + 1, \dots, m$ ), and simple bounds  $x^l \leq x \leq x^u$ . The objective function is group-partially separable and all constraint functions are partially separable.

**ARGUMENTS**

The arguments of CUTEST\_cohprods\_threaded are as follows:

**status** [out] - integer

the output status: 0 for a successful call, 1 for an array allocation/deallocation error, 2 for an array bound error, 3 for an evaluation error,

**n** [in] - integer

the number of variables for the problem,

**goth** [in] - logical

a logical variable which specifies whether the second derivatives of the groups and elements, and the indexing information held in IND (see below) have already been set (goth = .TRUE.) or if this information should be computed (goth = .FALSE.),

**X** [in] - real/double precision

when goth = .FALSE., the Hessians will be evaluated at X. Otherwise X is not used.

**VECTOR** [in] - real/double precision

an array which gives the vector whose product with the constraint Hessians is required,

**nnzohp** [out] - integer

the total number of nonzero entries required to store the product of the objective Hessian with a vector.

**loh** [in] - integer

a variable that specifies the declared lengths of IND and RESULT. The precise length required may be found by calling *CUTEST\_cdimohp* prior to *CUTEST\_cohprodsp*,

**RESULT** [out] - real/double precision

an array that gives the values of the nonzeros in the result obtained by multiplying the objective Hessians by VECTOR. The values are stored in RESULT(1:nnzohp).

**IND** [inout] - integer

an array that gives the indices of the nonzeros in the result obtained by multiplying the objective Hessian by VECTOR. The indices are stored in IND(1:nnzohp), and will match the values stored in RESULT.

**thread** [out] - integer

thread chosen for the evaluation; threads are numbered from 1 to the value threads set when calling CUTEst\_csetup\_threaded.

## NOTE

goth should be set to .TRUE. only when a previous call to CUTEst\_cohprods\_threaded, with goth = .FALSE., at the current point has been made. Otherwise, it should be set .FALSE.

## AUTHORS

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

## SEE ALSO

*CUTEst: a Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization,*

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

Computational Optimization and Applications **60**:3, pp.545-557, 2014.

*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,

ACM TOMS, **21**:1, pp.123-160, 1995.

cutest\_cdimchp(3M), sifdecoder(1).