

**NAME**

CUTEST\_cshj\_threaded – CUTEst tool to evaluate the Hessian of the John function, in sparse format.

**SYNOPSIS**

CALL CUTEST\_cshj\_threaded( status, n, m, X, y0, Y, nnzh, lh, H\_val, H\_row, H\_col, thread )

For real rather than double precision arguments, instead

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

**DESCRIPTION**

The CUTEST\_cshj\_threaded subroutine evaluates the Hessian of the John function  $j(x, y_0, y) = y_0 f(x) + y^T c(x)$  for the problem decoded from a SIF file by the script *sifdecoder* at the point  $(x, y_0, y) = (X, y_0, Y)$ . The matrix is stored in sparse format.

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\_cshj\_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, 4 for an out-of-range thread,

**n** [in] - integer

the number of variables for the problem,

**m** [in] - integer

the total number of general constraints,

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

an array which gives the current estimate of the solution of the problem,

**y0** [in] - real/double precision

the John scalar associated with the objective,

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

an array which gives the John multipliers,

**nnzh** [out] - integer

the number of nonzeros in H\_val,

**lh** [in] - integer

the actual declared dimensions of H\_val, H\_row and H\_col,

**H\_val** [out] - real/double precision

an array which gives the values of the Hessian matrix of the John function evaluated at X, y0 and Y. The i-th entry of H\_val gives the value of the nonzero in row H\_row(i) and column H\_col(i). Only the upper triangular part of the Hessian is stored,

**H\_row** [out] - integer

an array which gives the row indices of the nonzeros of the Hessian matrix of the John function evaluated at X, y0 and Y,

**H\_col** [out] - integer

an array which gives the column indices of the nonzeros of the Hessian matrix of the John function evaluated at X, y0 and Y,

**thread** [in] - integer

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

## 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*,  
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\_ush\_threaded(3M), sifdecoder(1).