

**NAME**

CUTEst\_csgrsh – CUTEst tool to evaluate constraints gradients, sparse Lagrangian Hessian and the gradient of either the objective/Lagrangian in sparse format.

**SYNOPSIS**

CALL CUTEst\_csgrsh( status, n, m, X, Y, grlagf, nnzj, lj, J\_val, J\_var, J\_fun, nnzh, lh, H\_val, H\_row, H\_col )

**DESCRIPTION**

The CUTEst\_csgrsh subroutine evaluates the gradients of the general constraints, the Hessian matrix of the Lagrangian function  $l(x, y) = f(x) + y^T c(x)$  and the gradient of either the objective function or the Lagrangian corresponding to the problem decoded from a SIF file by the script *sifdecoder* at the point  $(x, y) = (X, Y)$ . The data 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\_csgrsh 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,

**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,

**grlagf** [in] - logical

a logical variable which should be set .TRUE. if the gradient of the Lagrangian function is required and .FALSE. if the gradient of the objective function is sought,

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

an array which should give the Lagrange multipliers whenever grlagf is set .TRUE. but need not otherwise be set,

**nnzj** [out] - integer

the number of nonzeros in J\_val,

**lj** [in] - integer

the actual declared dimensions of J\_val, J\_fun and J\_var,

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

an array which gives the values of the nonzeros of the gradients of the objective, or Lagrangian, and general constraint functions evaluated at X and Y. The i-th entry of J\_val gives the value of the derivative with respect to variable J\_var(i) of function J\_fun(i),

**J\_var** [out] - integer

an array whose i-th component is the index of the variable with respect to which J\_val(i) is the derivative,

**J\_fun** [out] - integer

an array whose i-th component is the index of the problem function whose value J\_val(i) is the derivative. J\_fun(i) = 0 indicates the objective function whenever grlagf is .FALSE. or the Lagrangian function when grlagf is .TRUE., while J\_fun(i) = j > 0 indicates the j-th general constraint function,

**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 value of the Hessian matrix of the Lagrangian function evaluated at X 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 Lagrangian function evaluated at X and Y,

**H\_col** [out] - integer  
an array which gives the column indices of the nonzeros of the Hessian matrix of the Lagrangian function evaluated at X and Y.

**NOTE**

Calling this routine is more efficient than separate calls to CUTEST\_csgr and CUTEST\_csh.

**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\_cgr(3M), sifdecoder(1).