NAME

CUTEST_cdimse − CUTEst tool to determine the number of nonzeros required to store the Hessian of the Lagrangian.

SYNOPSIS

CALL CUTEST_cdimse(status, ne, he_val_ne, he_row_ne)

DESCRIPTION

The CUTEST_cdimse subroutine determines the number of nonzero elements required to store the Hessian matrix of the Lagrangian function for the problem decoded from a SIF file by the script *sifdecode*. The matrix is stored in sparse "finite element" format

$$H = \sum_{e=1}^{ne} H_{e,}$$

where each square symmetric element H_e involves a small subset of the rows of the Hessian matrix.

The problem under consideration is to minimize or maximize an objective function $f(x)$ over all $x \in R^n$ subject to general equations $c_i(x) = 0$, $(i \in 1, \ldots, m_E)$, general inequalities $c_i^l(x) \leq c_i(x) \leq c_i^u(x)$, $(i \in m_E + 1, \ldots, m)$, 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_cdimse are as follows

status [out] - integer
: the outputr status: 0 for a succesful call, 1 for an array allocation/deallocation error, 2 for an array bound error, 3 for an evaluation error,

ne [out] - integer
: the number of "finite-elements" used,

he_val_ne [out] - integer
: the dimension of the array needed to store the real values of the Hessian, taking all the elements into account (i.e. the dimension of the array HE_val).

he_row_ne [out] - integer
: the dimension of the array needed to store the integer values of the Hessian (i.e. the dimension of the array HE_row).

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.

cutest_ceh(3M), cutest_csgreh(3M), sifdecode(1).

NAME

CUTEST_cdimse − CUTEst tool to determine the number of nonzeros required to store the Hessian of the Lagrangian.

SYNOPSIS

CALL CUTEST_cdimse(status, ne, he_val_ne, he_row_ne)

DESCRIPTION

The CUTEST_cdimse subroutine determines the number of nonzero elements required to store the Hessian matrix of the Lagrangian function for the problem decoded from a SIF file by the script *sifdecode*. The matrix is stored in sparse "finite element" format

$$H = \sum_{e=1}^{ne} H_{e,}$$

where each square symmetric element H_e involves a small subset of the rows of the Hessian matrix.

The problem under consideration is to minimize or maximize an objective function $f(x)$ over all $x \in R^n$ subject to general equations $c_i(x) = 0$, $(i \in 1, \ldots, m_E)$, general inequalities $c_i^l(x) \leq c_i(x) \leq c_i^u(x)$, $(i \in m_E + 1, \ldots, m)$, 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_cdimse are as follows

status [out] - integer
: the outputr status: 0 for a succesful call, 1 for an array allocation/deallocation error, 2 for an array bound error, 3 for an evaluation error,

ne [out] - integer
: the number of "finite-elements" used,

he_val_ne [out] - integer
: the dimension of the array needed to store the real values of the Hessian, taking all the elements into account (i.e. the dimension of the array HE_val).

he_row_ne [out] - integer
: the dimension of the array needed to store the integer values of the Hessian (i.e. the dimension of the array HE_row).

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.

cutest_ceh(3M), cutest_csgreh(3M), sifdecode(1).