NAME

CUTEST_cchprods − CUTEst tool to form the matrix-vector products of a vector with each of the Hessian matrices of the constraint functions.

SYNOPSIS

CALL CUTEST_cchprods(status, n, m, goth, X, Y, VECTOR, lchp, CHP_val, CHP_ind, CHP_ptr)

For real rather than double precision arguments, instead

CALL CUTEST_cchprods_s(...)

and for quadruple precision arguments, when available,

CALL CUTEST_cchprods_q(...)

DESCRIPTION

The CUTEST_cchprods subroutine forms the product of a vector with each of the Hessian matrices of the constraint functions $c(x)$ 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 $c_i(x) = 0$, $(i \in 1, \ldots, m_E)$, general inequalities $c_i^l \leq c_i(x) \leq c_i^{u,}$ $(i \in m_E + 1, \ldots, m)$, 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_cchprods 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,

n [in] - integer
the number of variables for the problem,

m [in] - integer
the total number of general constraints,

goth [in] - logical
a logical variable which specifies whether the second derivatives of the groups and elements, and the indexing information held in
CHP_ind and CHP_ptr (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,

lchp [in] - integer
a variable that specifies the declared lengths of CHP_val and CHP_ind. The precise length required may be found by calling *CUTEST_cdimchp* prior to *CUTEST_cchprods*,

CHP_val [out] - real/double precision
an array that gives the values of the nonzeros in the result obtained by multiplying the constraint Hessians by VECTOR. The values for the i-th constraint are stored in CHP_val(CHP_ptr(i):CHP_val(i+1)-1),

CHP_ind [inout] - integer
an array that gives the indices of the nonzeros in the result obtained by multiplying the constraint Hessians by VECTOR. The indices for the i-th constraint are stored in CHP_ind(CHP_ptr(i):CHP_ptr(i+1)-1), and match the values stored in CHP_val,

CHP_ptr [inout] - integer
an array of length m+1 that gives pointers to the starting positions in CHP_ind and CHP_val for the nonzeros for the product with each Hessian. CHP_ptr(m+1)-1 gives the total space required by CHP_ind and CHP_val.

NOTE

goth should be set to .TRUE. only when a previous call to CUTEST_cchprods, 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).