NAME

CUTEST_csjprod − CUTEst tool to form the matrix-vector product of a sparse vector with the Jacobian of the constraints, or its transpose.

SYNOPSIS

CALL CUTEST_csjprod(status, n, m, gotj, jtrans, X, nnz_vector, INDEX_nz_vector, VECTOR, lvector, nnz_result, INDEX_nz_result, RESULT, lresult)

For real rather than double precision arguments, instead

CALL CUTEST_csjprod_s(...)

and for quadruple precision arguments, when available,

CALL CUTEST_csjprod_q(...)

DESCRIPTION

The CUTEST_csjprod subroutine forms the product of a sparse vector with the Jacobian matrix, or with its transpose, of the constraint functions of the problem decoded from a SIF file by the script *sifdecoder* evaluated at the point X.

The problem under consideration is to minimize or maximize an objective function $f(x)$ o ver 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_csjprod 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,

gotj [in] - logical
a logical variable which specifies whether the first derivatives of the groups and elements have already been set (gotj = .TRUE.) or if they should be computed (gotj = .FALSE.),

jtrans [in] - logical
a logical variable which specifies whether the product should involve the Jacobian (jtrans = .FALSE.) or its transpose (jtrans = .TRUE.),

X [in] - real/double precision
when gotj = .FALSE., the derivatives will be evaluated at X. Otherwise X is not used,

nnz_vector [in] - integer
the number of nonzeros in the vector whose product with the Hessian is required,

INDEX_nz_vector [in] - integer
an array that gives the indiices of the nonzeros of the vector whose product with the Hessian is required,

VECTOR [in] - real/double precision
an array that gives the vector whose product with the Hessian is required; only the nonzeros need be specified,

lvector [in] - integer
the actual declared dimension of VECTOR, that should be at least n when jtrans is ’FALSE. and at least m when jtrans is .TRUE.,

nnz_result [out] - integer
the number of nonzeros in the result obtained by multiplying the Hessian by VECTOR,

INDEX_nz_result [out] - integer
an array that gives the indiices of the nonzeros in the result obtained by multiplying the Hessian by VECTOR,

RESULT [out] - real/double precision
an array that gives the result of multiplying the Hessian by VECTOR; only the nonzeros will be set,

lresult [in] - integer
the actual declared dimension of RESULT, that should be at least m when jtrans is ’FALSE. and at least n when jtrans is .TRUE..

NOTE

gotj should be set to .TRUE. whenever

(1)
a call has been made to CUTEST_cjprod at the current point, or

(2)
a previous call to CUTEST_csjprod, with gotj = .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,
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_cjprod(3M), sifdecoder(1).