NAME

CUTEST_cdimsj − CUTEst tool to determine the number of nonzeros required to store the matrix of gradients of the objective function and constraints, in sparse format.

SYNOPSIS

```
CALL CUTEST_cdimsj( status, nnzj )
```

For real rather than double precision arguments, instead

```
CALL CUTEST_cdimsj_s( ... )
```

and for quadruple precision arguments, when available,

```
CALL CUTEST_cdimsj_q( ... )
```

DESCRIPTION

The CUTEST_cdimsj subroutine determines the number of nonzero elements required to store the matrix of gradients of the objective function and constraint functions for the problem decoded into OUTSDIF.d in the constrained minimization case. 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 $c_i(x) = 0$, $(i \in 1, \ldots, m_E)$, 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_cdimsj 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,

nnzj [out] - integer

the number of nonzero elements in the Jacobian matrix.

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.

sifdecoder(1).