NAME

CUTEST_classification − CUTEst tool to obtain the name of the problem directly from OUTSDIF.d.

SYNOPSIS

CALL CUTEST_classification(status, input, classification)

For real rather than double precision arguments, instead

CALL CUTEST_classification_s(...)

and for quadruple precision arguments, when available,

CALL CUTEST_classification_q(...)

DESCRIPTION

The CUTEST_classification subroutine obtains the classification string by interogating the datafile OUTSDIF.d that was created by the script *sifdecoder* when decoding a SIF file. Problems are classified according to the scheme described in

https://ralna.github.io/SIFDecode/html/classification/ .

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_classification are as follows

status [out] - integer
the outputr status: 0 for a succesful call, -1 for a read error.

input [in] - integer
the unit number for the decoded data; the unit from which OUTSDIF.d is read,

classification [out] - character
a 30-character string containing the SIF classification of the problem.

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_probname(3M), sifdecoder(1).