NAME
    CUTEST_creport – CUTEst tool to obtain statistics concerning function evaluation and CPU time used.

SYNOPSIS

    CALL CUTEST_creport(status, CALLS, TIME)

    For real rather than double precision arguments, instead

    CALL CUTEST_creport_s(...)

    and for quadruple precision arguments, when available,

    CALL CUTEST_creport_q(...)

DESCRIPTION
    The CUTEST_creport subroutine obtains statistics concerning function evaluation and CPU time used for constrained optimization in a standardized 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 $c_i^l \le c_i(x) \le c_i^{u,}$ $(i \in m_E + 1, \ldots, m)$, and simple bounds $x^l \le x \le x^{u.}$ The objective function is group-partially separable and all constraint functions are partially separable.

ARGUMENTS
    The arguments of CUTEST_creport 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,

    CALLS [out] - real array of length 7
        gives the number of calls to the problem functions:

    CALLS(1): number of calls to the objective function

    CALLS(2): number of calls to the objective gradient

    CALLS(3): number of calls to the objective Hessian

    CALLS(4): number of Hessian times vector products

    CALLS(5): number of calls to the constraint functions

    CALLS(6): number of calls to the constraint gradients

    CALLS(7): number of calls to the constraint Hessians

    TIME [out] - real array of length 4:

    TIME(1): CPU time (in seconds) for CUTEST_csetup

TIME(2): CPU time (in seconds) since the end of CUTEST_csetup

TIME(3): elapsed system clock time (in seconds) for CUTEST_csetup

TIME(4): elapsed system clock time (in seconds) since the end of CUTEST_csetup.

NOTE

Note that CALLS(4), CALLS(5) and CALLS(6) may account for codes which allow the evaluation of a selection of constraints only and may thus be much smaller than the number of constraints times the number of iterations.

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.