

**NAME**

CUTEST\_creport – CUTEst tool to obtain statistics concerning function evaluation and CPU time used.

**SYNOPSIS**

CALL CUTEST\_creport( status, CALLS, TIME )

**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  $c_i(x) = 0$ , ( $i \in 1, \dots, m_E$ ), general inequalities  $c_i^l(x) \leq c_i(x) \leq c_i^u(x)$ , ( $i \in m_E + 1, \dots, m$ ), 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\_creport are as follows

**status** [out] - integer

the output status: 0 for a successful 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 2:

TIME( 1 ): CPU time (in seconds) for CUTEST\_csetup

TIME( 2 ): CPU 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**

*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, TOMS, **21**:1, pp.123-160, 1995.

**NAME**

CUTEST\_creport – CUTEst tool to obtain statistics concerning function evaluation and CPU time used.

**SYNOPSIS**

CALL CUTEST\_creport( status, CALLS, TIME )

**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  $c_i(x) = 0$ , ( $i \in 1, \dots, m_E$ ), general inequalities  $c_i^l(x) \leq c_i(x) \leq c_i^u(x)$ , ( $i \in m_E + 1, \dots, m$ ), 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\_creport are as follows

**status** [out] - integer

the output status: 0 for a successful 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 2:

TIME( 1 ): CPU time (in seconds) for CUTEST\_csetup

TIME( 2 ): CPU 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**

*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, TOMS, **21**:1, pp.123-160, 1995.