

**NAME**

CUTEST\_ccifg\_threaded – CUTEst tool to evaluate a single constraint function value and possibly its gradient.

**SYNOPSIS**

CALL CUTEST\_ccifg\_threaded( status, n, icon, X, ci, GCI\_val, grad, thread )

**DESCRIPTION**

The CUTEST\_ccifg\_threaded subroutine evaluates the value of a particular constraint function of the problem decoded from a SIF file by the script *sifdecoder* at the point X, and possibly its gradient in the constrained minimization case. 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  $c_i^l \leq c_i(x) \leq c_i^u$  ( $i \in m_E + 1, \dots, m$ ), 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\_ccifg\_threaded 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, 4 for an out-of-range thread,

**n** [in] - integer

the number of variables for the problem,

**icon** [in] - integer

the index of the constraint function to be evaluated,

**X** [in] - real/double precision

an array which gives the current estimate of the solution of the problem,

**ci** [out] - real/double precision

the value of constraint function icon at X,

**GCI\_val** [out] - real/double precision

an array which gives the gradient of constraint function icon evaluated at X,

**grad** [in] - logical

a logical variable which should be set .TRUE. if the gradient of the constraint functions are required and .FALSE. otherwise,

**thread** [in] - integer

thread chosen for the evaluation; threads are numbered from 1 to the value threads set when calling CUTEST\_csetup\_threaded.

**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)