

**NAME**

CUTEST\_cofsg – CUTEst tool to evaluate function value and possibly gradient.

**SYNOPSIS**

CALL CUTEST\_cofsg( status, n, X, f, nnzg, lg, G\_val, G\_var, grad )

**DESCRIPTION**

The CUTEST\_cofsg subroutine evaluates the value of the objective function of the problem decoded from a SIF file by the script *sifdecoder* at the point X, and possibly its gradient 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  $c_i(x) = 0$ , ( $i \in 1, \dots, m_E$ ), 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\_cofsg 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,

**n** [in] - integer

the number of variables for the problem,

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

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

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

the value of the objective function evaluated at X,

**nnzg** [out] - integer

the number of nonzeros in G\_val,

**lg** [in] - integer

the declared length of G\_val and G\_var,

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

an array which gives the nonzeros of the gradient of the objective function evaluated at X. The i-th entry of G\_val gives the value of the derivative with respect to variable G\_var(i) of the objective function,

**G\_var** [out] - integer

an array whose i-th component is the index of the variable with respect to which G\_val(i) is the derivative, and

**grad** [in] - logical

a logical variable which should be set to .TRUE. if the gradient of the objective function is required and .FALSE. otherwise.

**NOTE**

A call to CUTEST\_cofsg is more efficient than two separate calls to CUTEST\_cfn and CUTEST\_csgr.

**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,

Technical Report, Rutherford Appleton Laboratory, 2013.

*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\_uofg(3M), cutest\_cofg(3M), sifdecoder(1).