

**NAME**

CUTEST\_ccifsg – CUTEst tool to evaluate a single constraint function value and possibly gradient in sparse format.

**SYNOPSIS**

CALL CUTEST\_ccifsg( status, n, icon, X, ci, nnzgci, lgci, GCI\_val, GCI\_var, grad )

**DESCRIPTION**

The CUTEST\_ccifsg subroutine evaluates the value of a particular constraint function of the problem decoded from a SIF file by the script *sifdecode* at the point X, and possibly its gradient in the constrained minimization case. The gradient is stored 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\_ccifsg 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,

**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,

**nnzgci** [out] - integer

the number of nonzeros in GCI\_val,

**lgci** [in] - integer

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

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

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

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

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

**grad** [in] - logical

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

**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.

sifdecode(1)

**NAME**

CUTEST\_ccifsg – CUTEst tool to evaluate a single constraint function value and possibly gradient in sparse format.

**SYNOPSIS**

CALL CUTEST\_ccifsg( status, n, icon, X, ci, nnzgci, lgci, GCI\_val, GCI\_var, grad )

**DESCRIPTION**

The CUTEST\_ccifsg subroutine evaluates the value of a particular constraint function of the problem decoded from a SIF file by the script *sifdecode* at the point X, and possibly its gradient in the constrained minimization case. The gradient is stored 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\_ccifsg 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,

**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,

**nnzgci** [out] - integer

the number of nonzeros in GCI\_val,

**lgci** [in] - integer

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

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

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

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

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

**grad** [in] - logical

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

**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.

sifdecode(1)