

Focus Manuals: Programing Guide: Operations on structures

links from [Programing Guide](#)

Operations on structures

The functions and subroutines described below are used for manipulating structures without having to know how they are arranged. This paragraph will be limited to the functions related to the notion of structure itself. The traditional operations on matrices and vectors will be dealt with in Chapter A.III.

All the functions described hereafter will be naturally declared by a **USE BIEF** statement at the beginning of subroutines. Otherwise they would have to be declared as **EXTERNAL**.

General operations on structures

LOGICAL FUNCTION CMPOBJ(T1,T2)

Up to now **T1** is a vector or a block, as is **T2**.

CMPOBJ indicates if the two structures are identical. A check is made to see whether these two structures are of the same type and, if so, their characteristics are compared:

- for vectors: discretisation.
- for blocks: the number of structures that it contains.

Nothing has been done so far for the other structures.

This function is used by subroutine **OS**.

Operations on vectors

SUBROUTINE CHGDIS(VEC,OLDELT,NEWELT,MESH)

CHGDIS changes the discretisation of a vector.

- **VEC** is the vector
- **MESH** the structure containing the mesh integers
- **OLDELT** is the former vector discretisation and
- **NEWELT** is the new one.

A vector can thus go from a linear discretisation to a quasi-bubble discretisation, or the reverse. In the first case, the missing values are found by linear interpolation, while in the second case the superfluous values are forgotten. There are certain restrictions to the use of this subroutine:

- A vector cannot be extended if the required memory space is not provided for during allocation.
- Certain changes are impossible, for obvious reasons: changing a triangle to a quadrilateral, etc.

SUBROUTINE CPSTVC(T1,T2)

This subroutine copies a vector structure onto another. If **T1** is a vector, **T2** then becomes a vector with the same characteristics. Nevertheless, the memory allocated during allocation cannot be changed. The only data copied for the moment are:

- Discretisation type (component **ELM**)
- The first dimension (component **DIM1**)
- The second dimension (component **DIM2**)
- The component **DIMDISC** in case of discontinuous vectors.

This subroutine should be used when dealing with temporary all purpose **BIEF_OBJ** structures like **T1**, **T2**, etc. in TELEMAC-2D and SISYPHE and **T3_01**, **T3_02**, etc. in TELEMAC-3D. As a matter of fact, these structures may have been changed by a previous use, e.g. they may have been turned into boundary vectors with a smaller size than a full domain vector, hence an initialisation like **CALL OS('X=0', X=T1)** may have a random effect if not secured previously by specifying what must be **T1**. Copying the structure of a known object like e.g. the depth in structure H, will do it. **CALL CPSTVC(H, T1)** will give **T1** the same dimension and discretisation as the depth.

Operations on matrices

Note: all the operations on vectors may also be applied to the diagonal and the extradiagonal terms contained in the matrix structure (respectively M%D and M%X for a matrix M). The following subroutines only apply to matrices.

SUBROUTINE CPSTMT(M1,M2,TRANS)

Copies characteristics of the matrix **M1** on to the matrix **M2**, or of transposed of matrix **M1** to **M2** (if optional argument **TRANS** is set to true).

CPSTMT is similar to **CPSTVC**, it carries out the following operations:

1. copies types of elements.
2. copies types of diagonal and off-diagonal terms (calls CPSTVC for the diagonal and the off-diagonal terms).
3. copies characteristics of the matrix (components TYPDIA and **TYPEXT**).
4. checks that **M2** has enough memory for its new characteristics : sizes of diagonal and extra-diagonal terms.

INTEGER FUNCTION DIM1_EXT(IELM1,IELM2,STO,TYPEXT)

The extra-diagonal terms of matrices are stored in 2-dimensional arrays. **DIM1_EXT** returns the first dimension of this array, depending on:

- **IELM1** : Type of discretisation for rows (same convention as for the vectors).
- **IELM2** : Type of discretisation for columns.
- **STO** : Storage of the matrix (1: EBE, 3: edge based)
- **TYPEXT** : Type of the off-diagonal terms ('0': zero, 'Q': any, 'S': symmetrical)

INTEGER FUNCTION DIM2_EXT(IELM1,IELM2,STO,TYPEXT)

The extra-diagonal terms of matrices are stored in 2-dimensional arrays. **DIM2_EXT** returns the second dimension of this array, depending on:

- **IELM1** : Type of discretisation for rows (same convention as for the vectors).
- **IELM2** : Type of discretisation for columns.
- **STO** : Storage of the matrix (1: EBE, 3: edge based)
- **TYPEXT** : Type of the off-diagonal terms ('0': zero, 'Q': any, 'S': symmetrical)

Operations on blocks

SUBROUTINE ADDBLO(BLOCK,T):

Adds the structure **T** to the block.

- **BLOCK** is a block and
- **T** a structure.

Reaching objects in blocks

If **T1** is a vector stored as the second object in a block **B**, the address of **T1** is **B%ADR(2)%P**. As a matter of fact, **ADR** is an array of **POINTER_TO_BIEF_OBJ** structures, we take the second one, and its unique component **P** (**P** would not be present if Fortran 90 were accepting the arrays of pointers).

B%ADR(2)%P can then be treated as a **BIEF_OBJ** structure, for example the third real value of **T1** is **B%ADR(2)%P%R(3)**. It is not recommended to deal directly with objects in blocks, this can be done in a subroutine by calling it with the argument (e.g.) **B%ADR(2)%P**. It will be then received in the subroutine as a normal **BIEF_OBJ** structure.

Component selectors can be piled up if blocks themselves are stored in blocks as in the following example, where **T1** as been stored as the second object into a block **C** stored as the third object in the block **B**. **T1** is then:

```
B%ADR(3)%P%ADR(2)%P
```

The only difficulty and common error is to forget the component **P** which is due to Fortran obscure reasons.

From:

<http://wiki.opentelemac.org/> - open TELEMAC-MASCARET

Permanent link:

http://wiki.opentelemac.org/doku.php?id=programing_guide:operations_on_structures

Last update: 2014/10/10 16:01

