

Focus Manuals: Programing Guide: Building matrices and vectors: Available construction of Matrices

links from [Programing Guide](#) and from [Building matrices and vectors](#)

FORMULs of SUBROUTINE MATRIX  
MATMAS

FORMUL = 'MATMAS '

(in library BIEF subroutines with names which start with MT01)

Mass-matrix.

$$N(i, j) = XMUL \int_{\Omega} \psi_i \psi_j d\Omega$$

MATDIF

FORMUL = 'MATDIF '

(in library BIEF subroutines with names which start with MT02)

Diffusion matrix with different coefficients according to the directions x, y and z.

**In 2 dimensions - isotropic:**

$$N(i, j) = XMUL \int_{\Omega} \left( U \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + U \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right) d\Omega$$

The case of an isotropic viscosity is given above. But the viscosity may also be tensorial. In this case **U** (a **BIEF\_OBJ** structure) must have a second dimension, for example 3- in 2-dimensional applications. **U** will have the general form:

$$U = \begin{pmatrix} U_{xx} & U_{xy} \\ U_{yx} & U_{yy} \end{pmatrix}$$

but the tensor is symmetric and  $U_{xy} = U_{yx}$ .

Elements of the tensor must be stored in **U** as follows:

- $U_{xx}$  in **U(\*, 1)**
- $U_{yy}$  in **U(\*, 2)**
- $U_{xy}$  in **U(\*, 3)**

**In a 2 dimensions - non isotropic:**

$$N(i, j) = XMUL \int_{\Omega} \left( U_{xx} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + U_{yy} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} + U_{xy} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} + U_{xy} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \right) d\Omega$$

When a transversal **K\_t** and longitudinal **K\_l** dispersion are used (case of Elder's turbulence model), the formula giving the tensor **U** is:

- $U_{xx} = K_l \cos(\theta)^2 + K_t \sin(\theta)^2$
- $U_{yy} = K_l \sin(\theta)^2 + K_t \cos(\theta)^2$
- $U_{xy} = (K_l K_t)(\sin(\theta) \cos(\theta))$

**In 3 dimensions** (beware, **F**, **G** and **H** are used in this case, unlike in 2D where **U**, **V** and **W** are used):

$$N(i, j) = XMUL \int_{\Omega} (F \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + G \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} + H \frac{\partial \psi_i}{\partial z} \frac{\partial \psi_j}{\partial z}) d\Omega$$

**MATDIF2**

**FORMUL = 'MATDIF2 '**

In 3D only, formula MATDIF2 is like MATDIF, but the hydrostatic inconsistencies are dealt with.

**MATDIF3**

**FORMUL = 'MATDIF3 '**

In 2D only so far, diffusion matrix with diffusion coefficients which are piece-wise linear or constant, but may be discontinuous between elements (this is used in groundwater flows).

$$N(i, j) = XMUL \int_{\Omega} (F \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + G \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} + H \frac{\partial \psi_i}{\partial z} \frac{\partial \psi_j}{\partial z}) d\Omega$$

Here one must have :

- U%ELM=10, U%DIM2=3, U%DIMDISC=11
- V%ELM=10, V%DIM2=3, V%DIMDISC=11

**MASUPG**

**FORMUL = 'MASUPG '**

(subroutines with names which start with MT03)

Matrix used for the convection term with method SUPG option 1.

$$N(i, j) = XMUL \int_{\Omega} (\vec{F} \cdot \overrightarrow{grad}(\psi_i) \vec{U} \cdot \overrightarrow{grad}(\psi_j)) d\Omega$$

- $\vec{F}$  here is a vector with the components **F**, **G** and **H**.
- $\vec{U}$  is a vector with the components **U**, **V** and **W**.

**MAUGUG**

**FORMUL = 'MAUGUG '**

(subroutines with names which start with MT04)

Matrix used for the convection term with method SUPG option 2.

$$N(i, j) = XMUL \int_{\Omega} (\vec{U} \cdot \overrightarrow{\text{grad}}(\psi_i) \vec{U} \cdot \overrightarrow{\text{grad}}(\psi_j)) d\Omega$$

- $\vec{U}$  is a vector with the components **U**, **V** and **W**.

MATVGR

FORMUL = 'MATVGR '

(subroutines with names which start with MT05)

Matrix used for the convection term with centred discretisation.

$$N(i, j) = XMUL \int_{\Omega} (\psi_i \vec{U} \cdot \overrightarrow{\text{grad}}(\psi_j)) d\Omega$$

- $\vec{U}$  is a vector with the components **U**, **V** and **W**.

FMATMA

FORMUL = 'FMATMA '

(subroutines with names which start with MT06)

Matrix used for conservative smoothing.

$$N(i, j) = XMUL \int_{\Omega} (F \psi_i \psi_j) d\Omega$$

MSLUMP

FORMUL = 'MSLUMP '

(subroutines with names which start with MT07)

Mass matrix with local mass-lumping.

$$N(i, j) = XMUL \int_{\Omega} ((1 - F) \psi_i + F \psi_i \psi_j) d\Omega$$

Here, **F** must be a **P0** function, that is, constant for each element. If the value of **F** is locally 0, the mass-matrix will be locally lumped into a diagonal.

MATFGR

FORMUL = 'MATFGR X'

(subroutines with names which start with MT08)

$$N(i, j) = -XMUL \int_{\Omega} (\psi_j F \frac{\partial \psi_i}{\partial x}) d\Omega$$

Beware the minus sign !!!!

If **FORMUL(16:16)** is equal to '**Y**' or '**Z**' instead of '**X**' , the derivatives will be obtained

according to y or z.

**MATQGR**

**FORMUL = 'MATQGR '** (not programmed yet)

(subroutines with names which start with MT09)

$$N(i, j) = XMUL \int_{\Omega} (\psi_i F \vec{U} \cdot \overrightarrow{grad}(\psi_j)) d\Omega$$

Subroutines with names which start with MT10 are not yet programmed.

**MATGRF**

**FORMUL = 'MATGRF X'**

(subroutines with names which start with MT11)

$$N(i, j) = -XMUL \int_{\Omega} (\psi_j \frac{\partial F}{\partial x}) d\Omega$$

Beware the minus sign !!!!

If **FORMUL(16:16)** is equal to **'Y'** or **'Z'** instead of **'X'** , the derivatives will be obtained according to y or z.

**MATUGH**

**FORMUL = 'MATUGH X'**

(subroutines with names which start with MT12)

Matrix used for the method SUPG, options 1 and 2.

$$N(i, j) = XMUL \int_{\Omega} (\psi_j \frac{\partial F}{\partial x} \vec{U} \cdot \overrightarrow{grad}(\psi_i)) d\Omega$$

If **FORMUL(16:16)** is equal to **'Y'** or **'Z'** instead of **'X'** , the derivatives will be obtained according to y or z.

- $\vec{U}$  is a vector with the components **U**, **V** and **W**.

**MATGRA**

**FORMUL = 'MATGRA X'**

(subroutines with names which start with MT13)

Gradient matrix.

$$N(i, j) = XMUL \int_{\Omega} (\frac{\partial \psi_j}{\partial x} \psi_i) d\Omega$$

If **FORMUL(16:16)** is equal to 'Y' or 'Z' instead of 'X' , the derivatives will be obtained according to y or z.

MAMURD PSI and MAMURD2 PSI

FORMUL = 'MAMURD PSI' or FORMUL = 'MAMURD2 PSI'

(subroutines with names which start with MT14)

Distribution matrix in case of use of the Multidimensional Upwind Residual Distribution scheme in 3D. See reference [4] for more details.

If **FORMUL(14:16)** is equal to ' N' instead of 'PSI' the matrix will be assembled.

FFBT

FORMUL = 'FFBT'

(subroutines with names which start with MT99)

This is in fact a series of different matrices and the string **FORMUL(8:16)** is also used for defining the formula. For example if **FORMUL(8:16)=' 0XX0'** , the matrix will be:

$$N(i,j) = XMUL \int_{\Omega} (F \frac{\partial F}{\partial x} \frac{\partial \psi_j}{\partial x} \psi_i) d\Omega$$

**Explanation:** the term in the integral is a product of 4 terms based, for the first 2, on the vector **F**, and then on the Basis function called here **B** and the test function called **T**.

- If the first character is **0**, the first term will be **F**.
- If the first character is **X**, the first term will be  $\frac{\partial F}{\partial x}$
- \* If the first character is **Y**, the first term will be  $\frac{\partial F}{\partial y}$

Then we proceed to second character and again function F, to third character and function  $\psi_j$ , to fourth character and function  $\psi_i$

The combinations **0XX0, 0YY0, XX00, 0X0Y, XY00, YY00, 0Y0X** have so far been implemented. The formula **FORMUL(8:16)='00XX+00YY'** is also available. Note that missing combinations can be obtained because the first two characters can be exchanged. Moreover exchanging the last two characters gives the transposed matrix of the previous formula.

SUBROUTINE by names

The existing subroutines building matrices in version 6.+ are the following, their function can be deduced from the explanations above:

```
mt01aa.f mt01bb.f mt01cc.f mt01oo.f mt01pp.f
mt01tt.f mt02aa.f mt02aa_2.f mt02bb.f mt02cc.f
mt02pp.f mt02pt.f mt02tt.f mt03aa.f mt03bb.f
mt03cc.f mt04aa.f mt04bb.f mt04cc.f mt04pp.f
mt04tt.f mt05aa.f mt05bb.f mt05cc.f mt05pp.f
mt05tt.f mt06aa.f mt06bb.f mt06cc.f mt06ff.f
mt06ft.f mt06ft2.f mt06oo.f mt06oc.f mt06pp.f
```

mt06tt.f mt07aa.f mt07bb.f mt07cc.f mt08aa.f  
mt08ab.f mt08ac.f mt08ba.f mt08bb.f mt08pp.f  
mt08tt.f mt11aa.f mt11ab.f mt11ac.f mt11ba.f  
mt11bb.f mt12aa.f mt12ab.f mt12ac.f mt12ba.f  
mt12bb.f mt13aa.f mt13ab.f mt13ba.f mt13bb.f  
mt13cc.f mt13ca.f mt14pp.f mt99aa.f

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

Permanent link:  
[http://wiki.opentelemac.org/doku.php?id=programing\\_guide:available\\_constructions\\_of\\_matrices](http://wiki.opentelemac.org/doku.php?id=programing_guide:available_constructions_of_matrices)

Last update: **2014/10/10 16:01**

