TGRID – 2 preprocessor for structured non–orthogonal 3D finite element grids

A. KOLCUN

Institute of Geonics ASCR, Studentská 1768, 708 00 Ostrava – Poruba, Czech Republic

1 Introduction

New pre-processor for FEM–system GEM32 is developed. In comparison with the TGRID pre-processor [1], more general geometry (structured meshes — see below) and boundary conditions are aviable. It consists of

mesh3 : mesh generation based on structured meshes,

warp3 : global mesh refinement,

mapro : material properties,

madis : material distribution,

bocon : boundary conditions,

mkfel : the generation of file `fel.g32`.

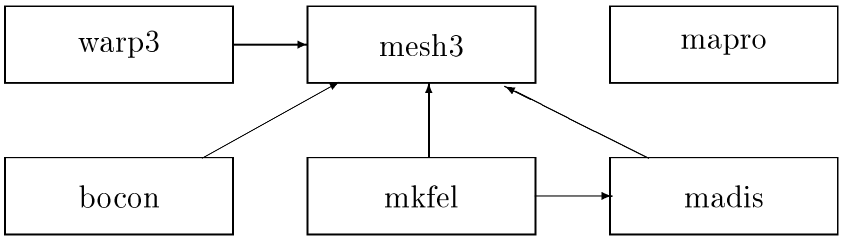


Figure 1: Hierarchical structure of pre-processor. Array points to the process, which must be done before actual one.

2 Structure of pre–processor

Each part of pre–processor is called independently:

```
name_of_code <directory_of_task>
```

The input information is read from the input data file. Each part of pre–processor (except `mkfel`, where no input information is need) has its own input datafile, i.e. we have
`mesh3.inp, warp3.inp, mapro.inp, madis.inp` and `bocon.inp` files.
Input datafiles have the common structure:

```
code of action
input information
code of action
input information
    :
    :
code of action
input information
```

where `code of action` indicates the action, we want to do (for example *change the coordinates of nodes*) and `input information` contains all information, necessery to this action (*indexes and coordinates of nodes*).
`code of action` consists of `input type (it)` and `number of lines (nl)` (both integers in one line separated with one or more spaces).
`it < 0` means that the `input information` lines are ignored.
`it = 0` means the comment line (i.e. no `input information` lines continues).
Notes:

- Empty lines in the input data files are forbidden!
- We can write comments in the end of each line too.
- Some differences are in the `mapro.inp` file.

3 Mesh generator

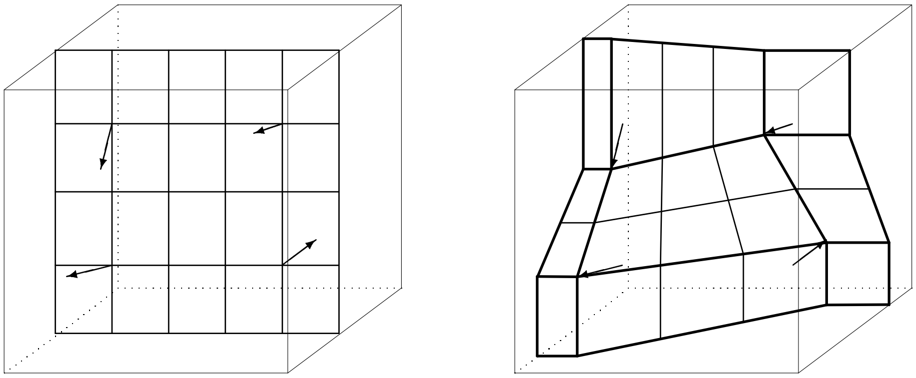
Mesh generator creaters *structured mesh* i.e. mesh which is obtained by deformation of regular rectangular one. It is based on

1. deformation of index plane in the rectangular regular mesh,
   - deformation of maximal inner index rectangle,
   - projection to the boundary of index plane,
2. interpolation between two deformed index planes,
3. global refinement (adding new index planes).

For more complex situation first step is realized recurently in the inner area of index plane. Deformation and projection of index plane is illustrated in the Fig.2.
Code `mesh3` creates the basic information `fv.g32` and the structured mesh `fx.g32`.

Input files : `mesh3.inp`.

Output files: `fv.g32, fx.g32`



a) b)

Figure 2: Schema of deformation of an "index" plane: a) index plane in the regular rectangular mesh with indicating the deformatin of quadrangle, b) index plane after deformation: thick lines – deformated quadrangle with orthogonal projections to the faces, medium lines – interpolated index lines.

Description : of the file `mesh.inp`:

input type = 1 – the input of the basic information about problem. In this case

number of lines `nl` = 8 and

input information consists of

```
nx, ny, nz
xmin, xmax
ymin, ymax
zmin, zmax
name
date
number of materials
homogen. kinematic. bc.c. 1=yes,0=no
```

input type = 2 – input of the indexes and coordinates of the nodes.

number of lines `nl` = number of input nodes.

input information consists of `nl` lines
i_x, i_y, i_z, x, y, z — indexes and coordinates of a node;

input type = 11 – deformation of x-index plane.

number of lines = 1

input information = i_x – number of index plane.

input type = 12 – projection to the boundary face.

number of lines = 1

input information =
: `1` – orthogonal projection,
: `0` – regular interpolation.

input type = `13` – projection of x-index planes to the boundary x-face.

number of lines = `1`

input information = i_{x1}, i_{x2} – indexes of x-index planes which are projected to the boundary faces $i_x = 1$, $i_x = nx$ respectively.

input type = `14` – shift of x-index plane

number of lines = `nl` number of shifted planes

input information `nl` lines
: i_{x1}, i_{x2}, dx, where x-index plane i_{x2} is copy of plane i_{x1} – shifted to the dx in x-direction.

note – input types 21–24, 31–34 case the y and z-index planes respectively.

example 1 file `mesh3.inp` for the stope stability problem ([1])

```
1 8
40 40 40
0 100
0 100
0 100
'stope stability problem'
'24.07.97'
1
1
2 xx
1 1 1  xx yy zz
:
11 1

tu pride stope!
na niekolko riadkov
```

example 2 file `mesh3.inp` for the 'Dolni Rozinka'- problem.

```
1 8
 5 11 6
 0 1430
 0 550
```

```
 0 600
 'Dolni Rozinka'
 '02.05.97'
 2
 1
2 12     nodes input
 2 2 2  200 100 100
 3 2 2  515 100 100
 4 2 2  830 100 100
 2 2 3  200 193 200
 3 2 3  515 192 200
 4 2 3  830 191 200
 2 2 4  200 230 300
 3 2 4  515 234 300
 4 2 4  830 230 300
 2 2 5  200 260 400
 3 2 5  515 270 400
 4 2 5  830 290 400
22 1     projection in the y-index plane
 2
23 1     projection of the y-index planes
 2 2
24 8     copies of the y-index planes, using the shift
 2  3 10
 3  4 17
 4  5 10
 5  6 23
 6  7 14
 7  8 11
 8  9 30
 9 10 30
```

4 Global mesh refinement

Code `warp3` adds the new index planes into the mesh.

Input files : `fv.g32, fx.g32, warp3.inp`

Output files: `fv.g32, fx.g32, old-new.warp`

Description : of the file `warp3.inp`:

note ! in the whole this section `number of lines` for each input type `nl = 1`

input type $= 11$ – the regular injection of index planes $i_x = const$

input information $= i_{x1}, i_{x2}, k$.

k new index lines are injected between each pair of indexes $(i_{x1}, i_{x1}+1), (i_{x1}+1, i_{x1}+2), \dots i_{x2}-1, i_{x2}$. Coordinates of nodes of index planes are regularly interpolated.

example 4: Let $nx = 6$. `input information = 2 5 3` changes the mesh in direction x as in the Fig. 4.

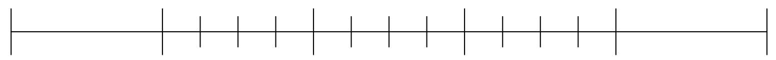


Figure 3: Example of regular injection of index planes.

input type $= 12$ – the exponential injection of index planes $i_x = const$

input information $= i_x, k, q, s$.

if $s = 1$ then k planes are injected regularly between $i_x - 1$ and i_x lines and this operation is repeated q times (q steps of injection).
if $s = 2$ then k planes are injected regularly between i_x and $i_x + 1$ lines and this operation is repeated q times (q steps of injection);

example 5: Let $nx = 3$. `input information = 3 1 4 1` changes the mesh in direction x as in the Fig. 4 a) and `input information = 2 2 3 2` changes the mesh in direction x as in the Fig. 4 b)

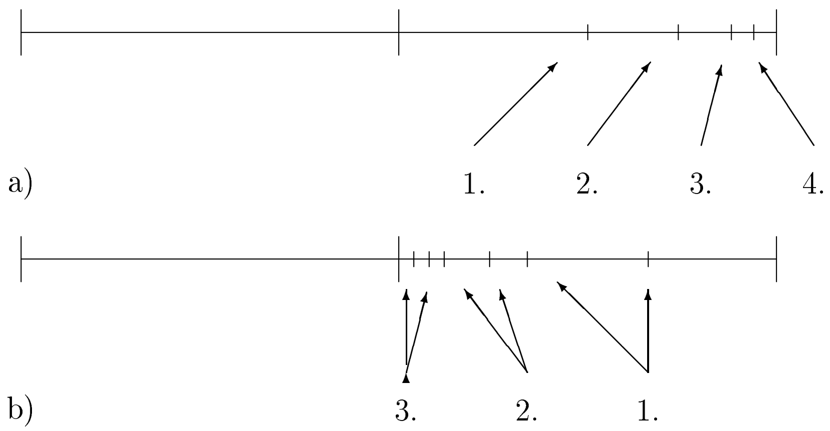


Figure 4: Examples of exponential injection of index planes. Number indicate the step of injection.

input type $= 13$ – the linear injection of index planes $i_x = const$

input information $= i_{x1}, i_{x2}, k$.

In the case $i_{x1} < i_{x2}$ there are $k, k-1, k-2, \ldots 2, 1, 0$ new index planes are regularly injected in each pair of the lines $(i_{x1}, i_{x1}+1), (i_{x1}+1, i_{x1}+2), \ldots i_{x2}-1, i_{x2}$ respectively. In the case $i_{x1} > i_{x2}$ there are $0, 1, 2, \ldots k-1, k$ new index planes are regularly injected in each pair of the lines $(i_{x2}, i_{x2}+1), (i_{x2}+1, i_{x2}+2), \ldots i_{x1}-1, i_{x1}$ respectively.

example 6: Let $nx = 6$. There are three different situations described here.
a) `input information = 2 5 4`
b) `input information = 2 5 2`
c) `input information = 5 2 2`
We can see, that the injection is made only within the boundary indexes i_{x1}, i_{x2}.

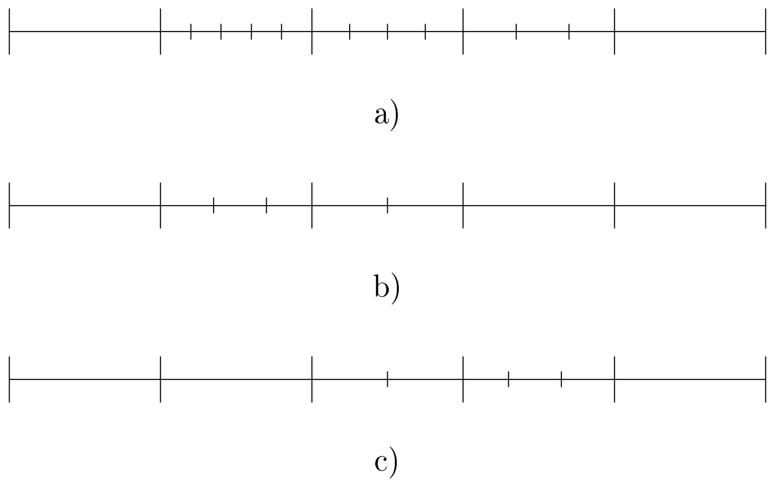


Figure 5: Examples of linear injections of index planes.

File `old-new.warp` contains the information of the changes of the mesh for the actualization of the material distribution and boundary conditions:
1. line : n_x, n_y, n_z from the original mesh
in the rest of lines there are the actual index numbers of original index spaces.

example 7 :

```
3 4 3
1
6
8
1
2
4
5
1
2
3
```

This `old-new.warp` means, that in the original mesh was $n_x = 3, n_y = 4, n_z = 3$. Four lines are injected between the first and the second index planes and one plane is injected between the second and the third planes in the x-direction.
One line is injected between the second and the third index planes in the y-direction.
z-direction is without changes.

5 Material properties

Code `mapro` (MAterial PROperties) creates the `fmat.g32` file. We can use the isotropic or the orthotropic materials.

In the case of orthotropic material the E – Young moduli and ν – Poisson constants must be input in the plane and in the orthogonal direction. Shoot. moduli and the angles defining the orthogonal direction must be input too.

When the Young moduli and ν – Poisson constants are the same in the plane and orthogonal directions, the material is supposed to be isotropic. In this case the shoot. moduli and the angles of the main direction are zero (you must fill this values in the `mapro.inp` file.

Input files : `mapro.inp`

Output files: `fmat.g32, fv.g32`

Description : When we finish the input of material properties, the *number of materials* is actualized in the file `fv.g32`.
The structure of file `mapro.inp` differs a little from the other input data files:
the first two lines (begining with number zero) create the header of the input table.
The next lines (begining with the order of material) contain information about the materials – see example. We needn't keep the ordering of the materials. Negative order number ignores the material properties line.

example : materials No 1., 3. are isotropic, material No. 2 is orthopropic.

```
0        E1        E2       ni1     ni2  gama   g   fi theta
0 ---------------------------------------------------------
1   20000.00   30000.00   .200   .150  2.1   .5  30.  25.
3   25000.00   25000.00   .250   .200  2.4   .4  40.  30.
2   10000.00   10000.00   .150   .150  2.0   .0   0.   0.
```

6 Distribution of materials

Code `madis` (MAterial DIStribution) realises the decomposition of the bricks to tetrahedra and the material distribution of the whole set of tetrahedra. We can input

- *homogeneous index block*
  defined by indexes $i_{x1} \leq i_x \leq i_{x2}, \quad i_{y1} \leq i_y \leq i_{y2}, \quad i_{x1} \leq i_x \leq i_{x2}$, where decomposition No. 72 (see Appendix) in each brick is used and all tetrahedra are filled with the same material type,
- *nonhomogeneous index block* where we can choose the decomposition type of brick and material type of each tetrahedra in the index brick,
- *rename*, i.e. change of one material type with another.

There are two possibilities how to input the new material type:

- *injection* – we rewrite original material type with new material type in the described area,
- *intersection* – we make intersection of the original material distribution and new material distribution. The rest of area is fill with material type 1.

Input files : `fv.g32, madis.inp`.

Output files : `mel.g32` (materials in elements), `part.g32` (decomposition of hexahedral bricks).

Description : of the file `madis.inp`:

input type $= 1$ – homogeneous index block

number of lines $=$ `nl` – amount of input blocks

input information $= i_{x1}, i_{y1}, i_{z1}, i_{x2}, i_{y2}, i_{z2}, mat, type$
where
$i_{x1}, i_{y1}, i_{z1} \quad i_{x2}, i_{y2}, i_{z2}$ — 3-indexes of nodes which define the index block,
mat — material type index and
$type \in \{1, 2\}$, where $1 =$ injection, $2 =$ intersection,

input type $= 2$ – nonhomogeneous index block

number of lines $=$ `nl` – amount of input blocks

input information $= i_{x1}, i_{y1}, i_{z1}, i_{x2}, i_{y2}, i_{z2}, dec, m_1, m_2, m_3, m_4, m_5, m_6$
where
$i_{x1}, i_{y1}, i_{z1} \quad i_{x2}, i_{y2}, i_{z2}$ are 3-indexes of nodes which define the index block,
$dec \in \{1, \ldots 72\}$ is code of decomposition, (see Appendix 1),
$m_1, m_2, m_3, m_4, m_5, m_6$ are the materials in the tetrahedra due to decomposition dec;
in this case only *injection* of input material type is possible.

Input type $= 3$ – rename,

number of lines $=$ `nl` – amount of renamed material types,

input information $= mat_1, mat_2$
where mat_1 and mat_2 are original and new material type respectively.

7 Boundary conditions

Code `bocon` (BOundary CONditions) creates both displacement and force boundary conditions.

Input files : `fv.g32, fx.f32, bocon.inp`

Output files : `fbc.g32` (displacements), `fr.g32` (forces).

Description : of the `bocon.inp` file:

input type $= 11$ – node force,

number of lines `nl` = number of input node forces

input information $= i_x, i_y, i_z, \quad f_x, f_y, f_z$
3-index of node and force vector in this node

input type $=12$ – pressure faces

number of lines `nl =` number of input pressure faces

input information $= i_{x1}, i_{y1}, i_{z1}, \quad i_{x2}, i_{y2}, i_{z2}, \qquad p$
3-indexes of index face and orthogonal preasure on it,

input type $=13$ – standard force conditions (pressure on the top face of the whole space)

number of lines `nl = 1`

input information $= h, \gamma,$
depth of the top face of domain and specific weight of the material above the domain,

input type $=21$ – displacement of the node

number of lines `nl =` number of input node displacements

input information $= i_x, i_y, i_z, \quad l_x, l_y, l_z$
3-index of node and displacement vector in this node

input type $=22$ – displacement face

number of lines `nl =` number of input displacement faces

input information $= i_{x1}, i_{y1}, i_{z1}, \quad i_{x2}, i_{y2}, i_{z2}, \qquad l_x, l_y, l_z$
3-indexes of index face and vector of displacememt on it,

input type $=23$ – standard displacement conditions (zero orthogonal displacement on the bottom and side faces)

number of lines `nl = 0`

input information – no input information in this case.

8 Finish of the input

Code `mkfel` (MaKe the file FEL.g32) creates the file `fel.g32` where the information about nodes coordinates, tetrahedra elements and material distribution is included.

Input files : `fv.g32, fx.f32, part.g32, mel.g32`.

Output files : `fel.g32, fel.rep`.

Description : when the `fel.g32` is created the irregularity of the mesh is tested (negative volume of tetrahedral element) and in the case of irregularity in the mesh the message in the file `fel.rep` appears:
`wrong brick` i_x, i_y, i_z

References

[1] R. Blaheta and R. Kohut, Benchmarks and programs for testing iterative solvers for 3D problems of geomechanics, Report 9502, Dept. Appl. Math., Institute of Geonics, Ostrava, August 1995.