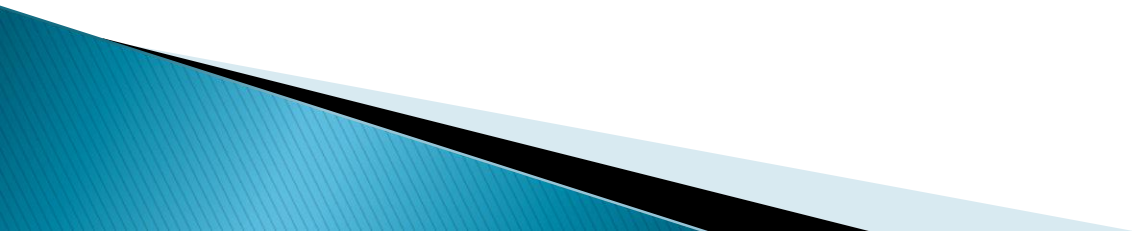


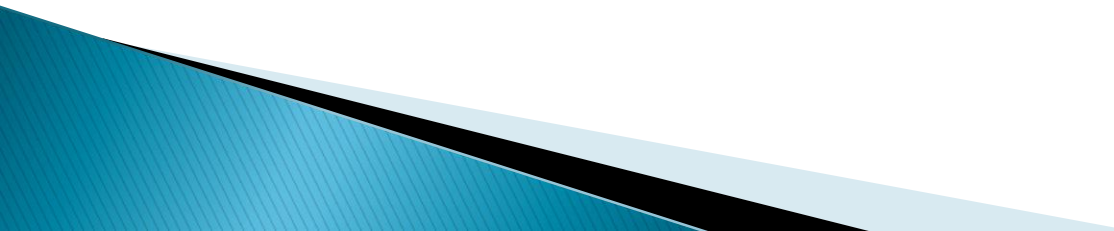
# Easymesh

Gerador de malhas bidimensionais

# Easymesh / Características

- Open Source
  - É possível definir mais de um material
  - Gera malha não estruturada
  - Possibilidade de fazer malhas com buracos
  - Refinamento ou engrossamento da malha localmente
- 

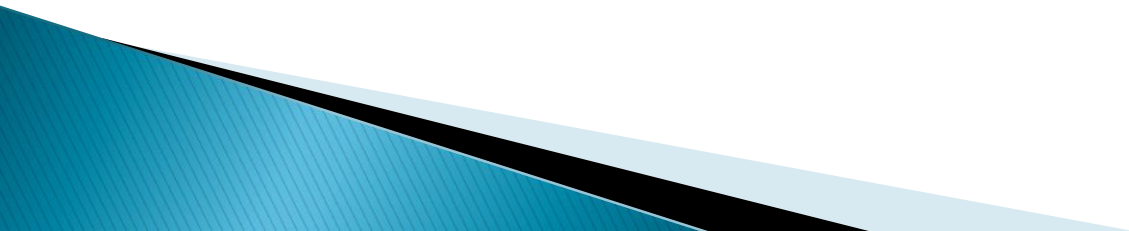
# Easymesh / Características

- Utiliza triangulação de Delaunay
  - Relaxamento da malha – evita criação de nós cercados de mais de 7 e menos de 5 elementos.
  - Utiliza suavização de Laplace
- 

# Onde encontrar

<http://www-dinma.univ.trieste.it/nirftc/research/easymesh/>

OBS: Mais fácil procurar “easymesh” no google!



# Compilação / Execução

- ▶ Comando para compilar no linux usando o GCC:  
`gcc easymesh.c -o easymesh -lm`
- ▶ É possível aumentar o tamanho máximo do número de nós:  
`#define MAX_NODES 3000`
- ▶ Para executar:  
`./easymesh arquivo_entrada`

# Arquivo de entrada

```
#=====
```

```
| PONTOS |
```

```
=====#
```

```
4 # Número de pontos #
```

```
# Nós que definem o contorno #
```

```
0:  0.0  >  0.0  >  0.1  1
1:  2.0  >  0.0  >  0.1  2
2:  2.0  >  2.0  >  0.1  3
3:  0.0  >  2.0  >  0.1  4
```

```
#=====
```

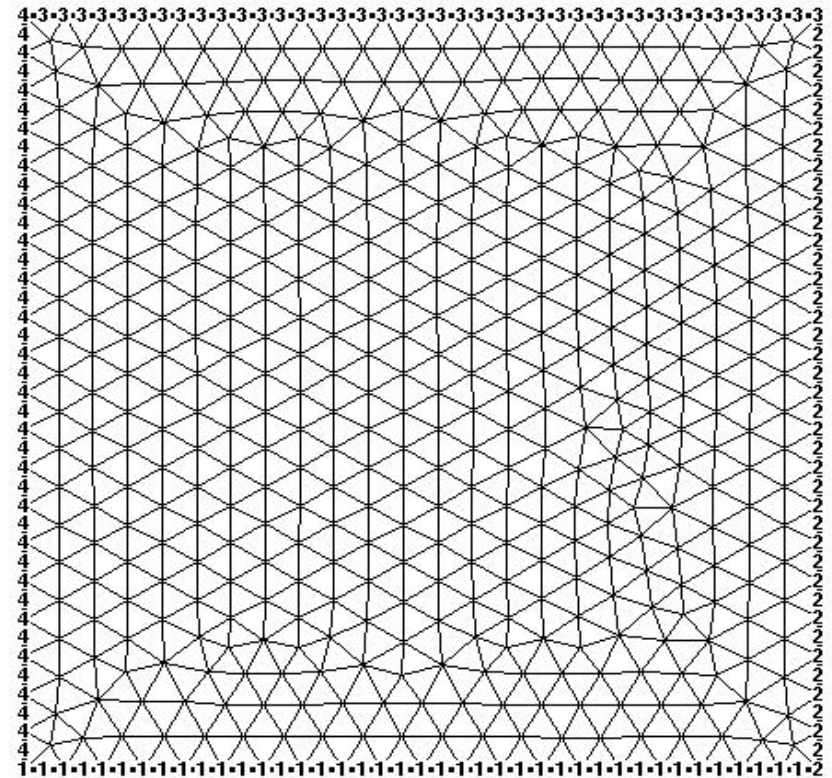
```
| SEGMENTOS |
```

```
=====#
```

```
4 # Número de segmentos #
```

```
# Segmentos do contorno externo #
```

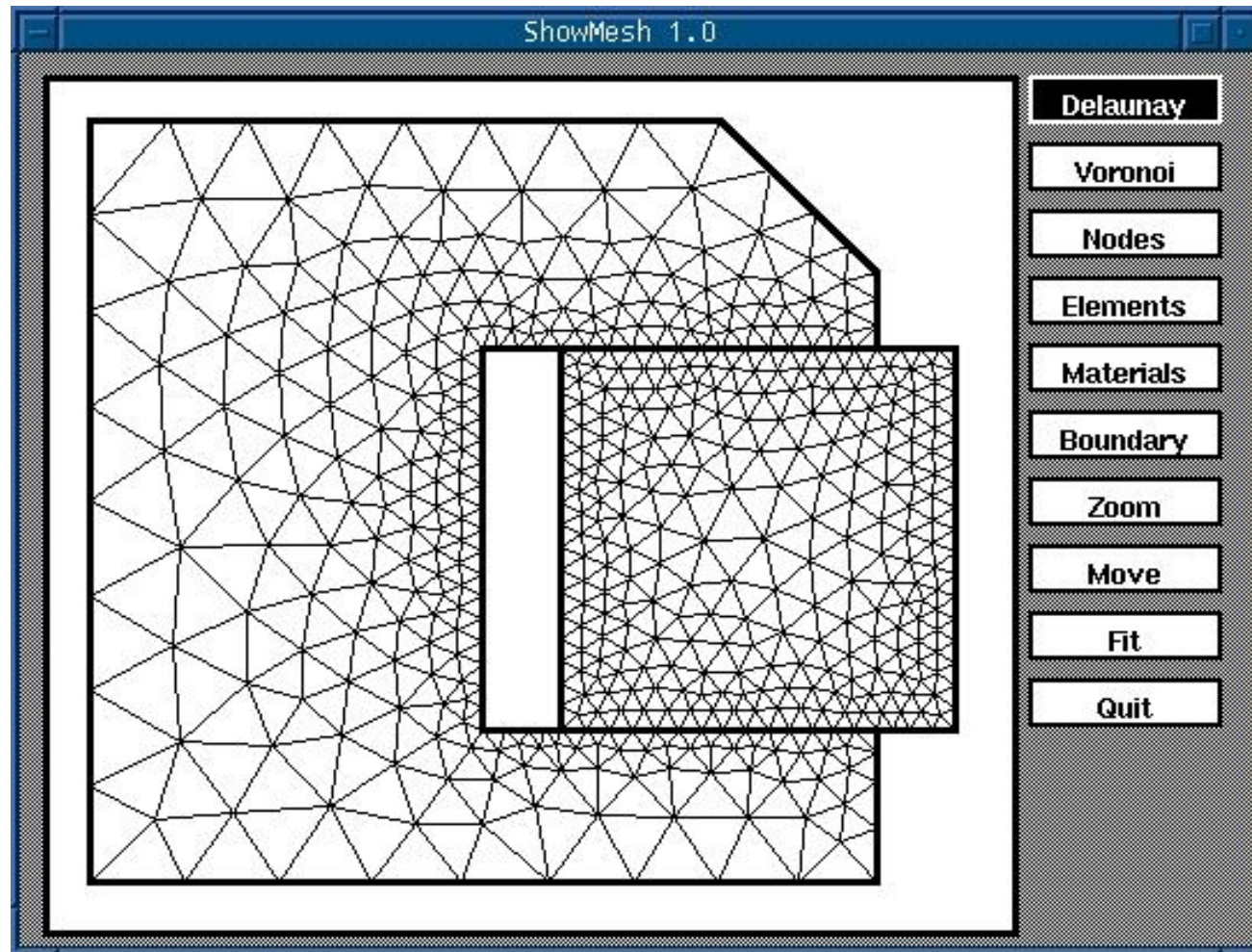
```
0:  0  1  1
1:  1  2  2
2:  2  3  3
3:  3  0  4
```



- Obs> Segmentos em sentido anti-horário



# Showmesh



# Showmesh

- Compilar:

```
gcc showmesh_1_0.c -o showmesh -lX11
```

- ▶ Para executar:

```
./showmesh arquivo_entrada
```

Obs: pode ser necessário instalar a biblioteca.

O que se pode aproveitar:

- Estruturas de elemento, lado e nó
- Função `load_mesh()`



# Arquivos de saída

- \* .n – arquivo dos nós
- \* .e – arquivo dos elementos
- \* .s – arquivo dos lados

# Arquivos de saída

- Arquivo de nós:

Primeira linha: <número de nós>

Linhas seguintes: <número do nó:> <x> <y> <marca do contorno>

```
507
0: 6.526388056742458e-02 6.542307428642820e-02 0
1: 0.000000000000000e+00 0.000000000000000e+00 1
2: 1.000000000000000e-01 0.000000000000000e+00 1
3: 0.000000000000000e+00 9.999999999999942e-02 4
4: 1.459049446463548e-01 8.098127622246956e-02 0
5: 8.039765041379812e-02 1.462464223161623e-01 0
6: 2.000000000000000e-01 0.000000000000000e+00 1
7: 0.000000000000000e+00 1.999999999999995e-01 4
8: 1.836383149521981e-01 1.852679585699063e-01 0
9: 2.461286259659382e-01 8.921823070242176e-02 0
10: 8.756293534236288e-02 2.460568706799738e-01 0
11: 3.000000000000000e-01 0.000000000000000e+00 1
12: 0.000000000000000e+00 2.999999999999996e-01 4
13: 2.977717310911008e-01 1.802031830847478e-01 0
```

# Arquivo de saída

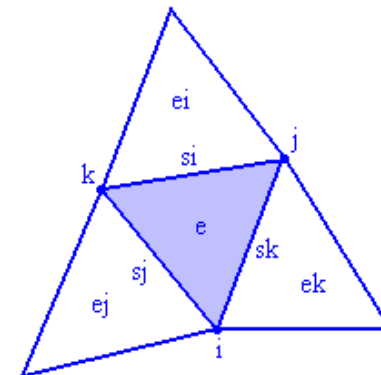
- Arquivo de elementos:

Primeira linha: <número de elementos>

Linhas seguintes: <número do elemento:> <i> <j> <k>  
<ei> <ej> <ek> <si> <sj> <sk> <xV> <yV> <marca  
do material>

p32

|    |   |   |   |    |    |    |    |    |    |                       |                       |    |
|----|---|---|---|----|----|----|----|----|----|-----------------------|-----------------------|----|
| 0: | 0 | 1 | 2 | -1 | 2  | 1  | 2  | 1  | 0  | 5.000000000000000e-02 | 1.538573906088764e-02 | -1 |
| 1: | 0 | 3 | 1 | -1 | 0  | 3  | 5  | 0  | 3  | 1.530130686222689e-02 | 4.999999999999971e-02 | -1 |
| 2: | 0 | 2 | 4 | 5  | 4  | 0  | 7  | 4  | 1  | 1.105377654083595e-01 | 4.752802329839616e-02 | -1 |
| 3: | 5 | 3 | 0 | 1  | 4  | 6  | 3  | 6  | 9  | 4.740496990037214e-02 | 1.105956020397727e-01 | -1 |
| 4: | 5 | 0 | 4 | 2  | 7  | 3  | 4  | 10 | 6  | 1.002801841504602e-01 | 1.006949812099583e-01 | -1 |
| 5: | 2 | 6 | 4 | 8  | 2  | -1 | 12 | 7  | 8  | 1.500000000000000e-01 | 2.515851050417709e-02 | -1 |
| 6: | 3 | 5 | 7 | 10 | -1 | 3  | 13 | 11 | 9  | 2.473873003137659e-02 | 1.499999999999995e-01 | -1 |
| 7: | 8 | 5 | 4 | 4  | 9  | 11 | 10 | 14 | 16 | 1.411170784881609e-01 | 1.416833894651589e-01 | -1 |



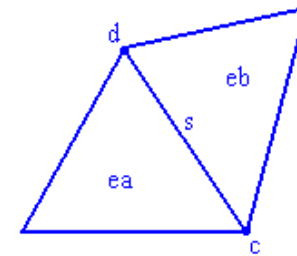
# Arquivo de saída

- Arquivo de lados:

Primeira linha: <número de lados>

Linhas seguintes> <número do lado: <no final> <no inicial> <ea> <eb> <marca do contorno>

```
1438
0:  1  0  1  0  0
1:  2  0  0  2  0
2:  1  2  0 -1  1
3:  3  0  3  1  0
4:  4  0  2  4  0
5:  1  3 -1  1  4
6:  5  0  4  3  0
7:  2  4  2  5  0
8:  2  6  5 -1  1
9:  3  5  6  3  0
10:  5  4  7  4  0
11:  7  3  6 -1  4
12:  6  4  5  8  0
13:  7  5 10  6  0
```



# Exemplos – Sela

- Solução conhecida (equação da sela)

$$u(x, y) = x^2 - y^2$$

$$u(x, y) = x^2 - 1, \quad -1 \leq x \leq 1 \quad y = -1 \text{ e } y = 1$$

$$u(x, y) = 1 - y^2, \quad -1 \leq y \leq 1 \quad x = -1 \text{ e } x = 1$$

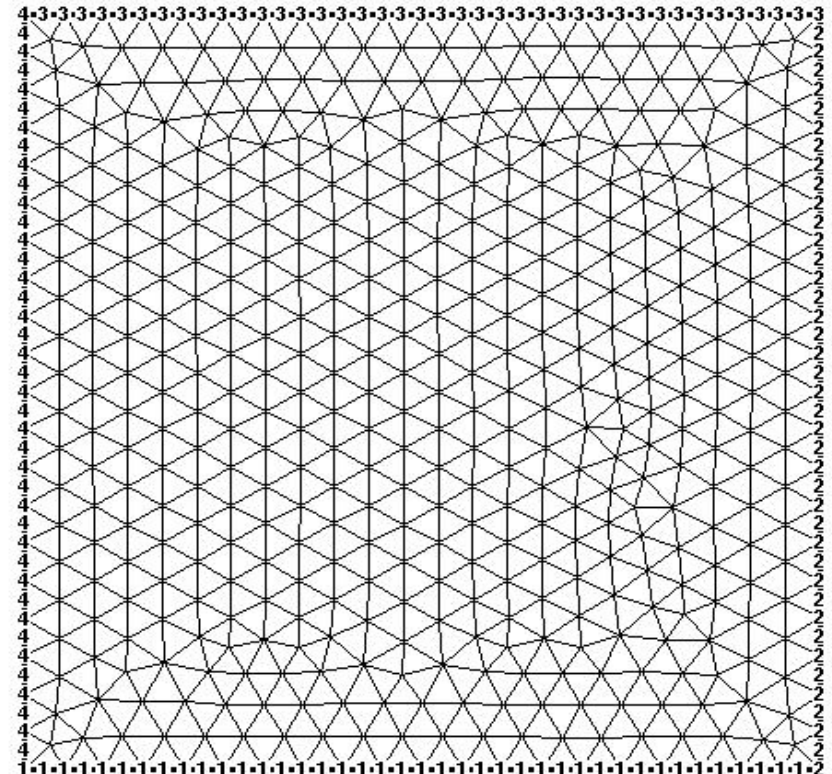
- Condição de valor prescrito

```
#=====
| PONTOS |
=====#
4 # Número de pontos #

# Nós que definem o contorno #
0:  0.0  0.0  0.1  1
1:  2.0  0.0  0.1  2
2:  2.0  2.0  0.1  3
3:  0.0  2.0  0.1  4

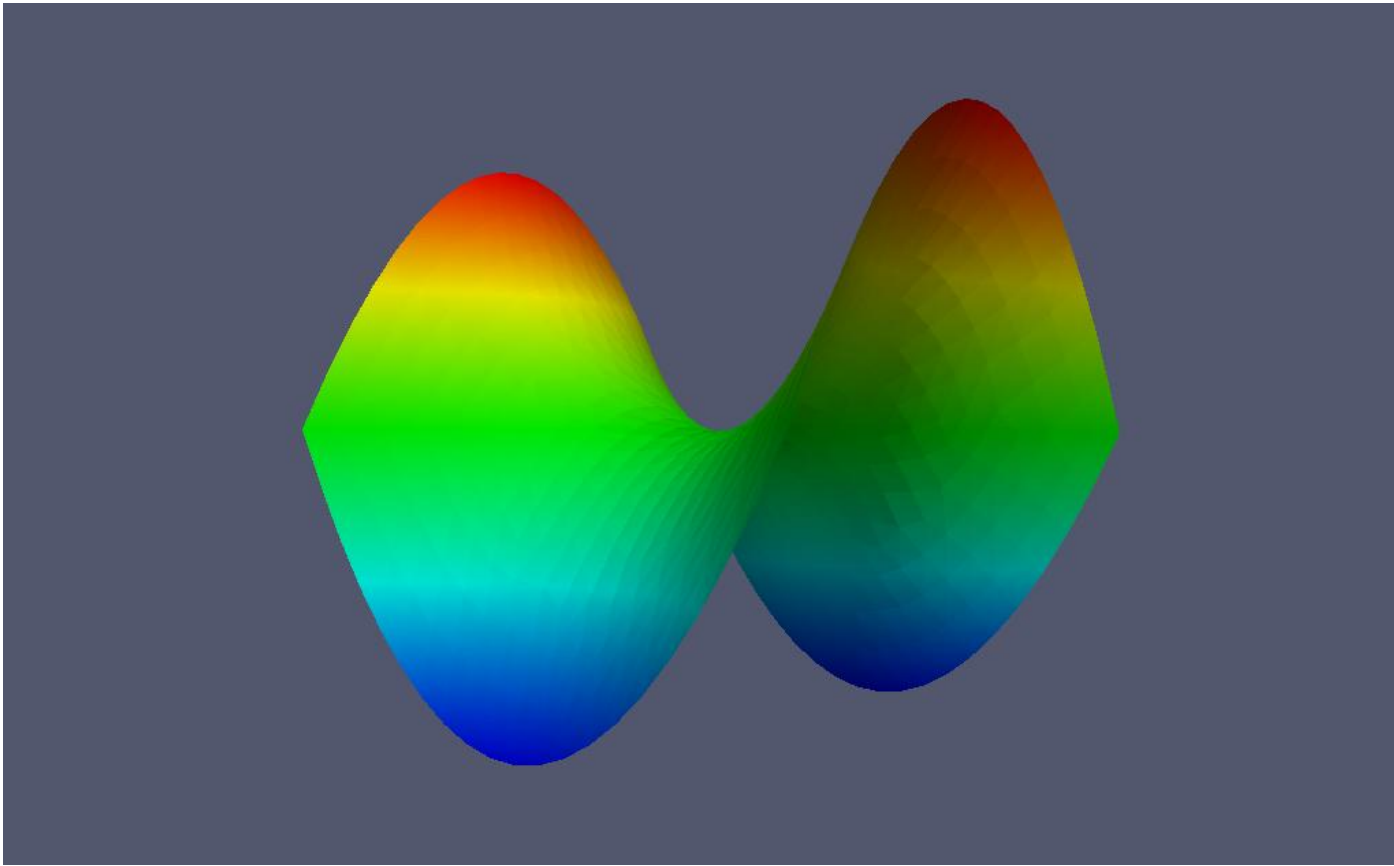
#=====
| SEGMENTOS |
=====#
4 # Número de segmentos #

# Segmentos do contorno externo #
0:  0  1  1
1:  1  2  2
2:  2  3  3
3:  3  0  4
```





# Exemplos – Sela (Paraview)





# Exemplo – Pudim

## ■ Problema Pudim

$$-\nabla \cdot (\mathbf{k} \nabla u) + \beta \nabla u + \sigma u = f$$

```
#=====
| PONTOS |
=====#
```

```
6 # Número de pontos #
```

```
# Nós que definem o contorno #
```

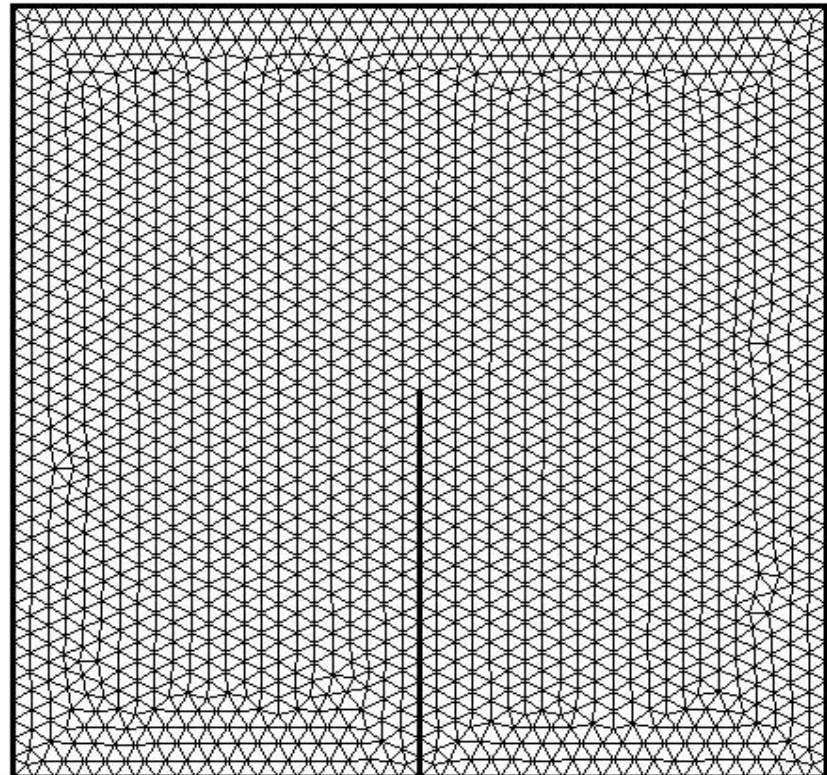
|    |     |   |     |   |      |   |
|----|-----|---|-----|---|------|---|
| 0: | 0.0 | > | 0.0 | > | 0.05 | 1 |
| 1: | 1.0 | > | 0.0 | > | 0.05 | 2 |
| 2: | 2.0 | > | 0.0 | > | 0.05 | 1 |
| 3: | 2.0 | > | 2.0 | > | 0.05 | 1 |
| 4: | 0.0 | > | 2.0 | > | 0.05 | 1 |
| 5: | 1.0 | > | 1.0 | > | 0.05 | 2 |

```
#=====
| SEGMENTOS |
=====#
```

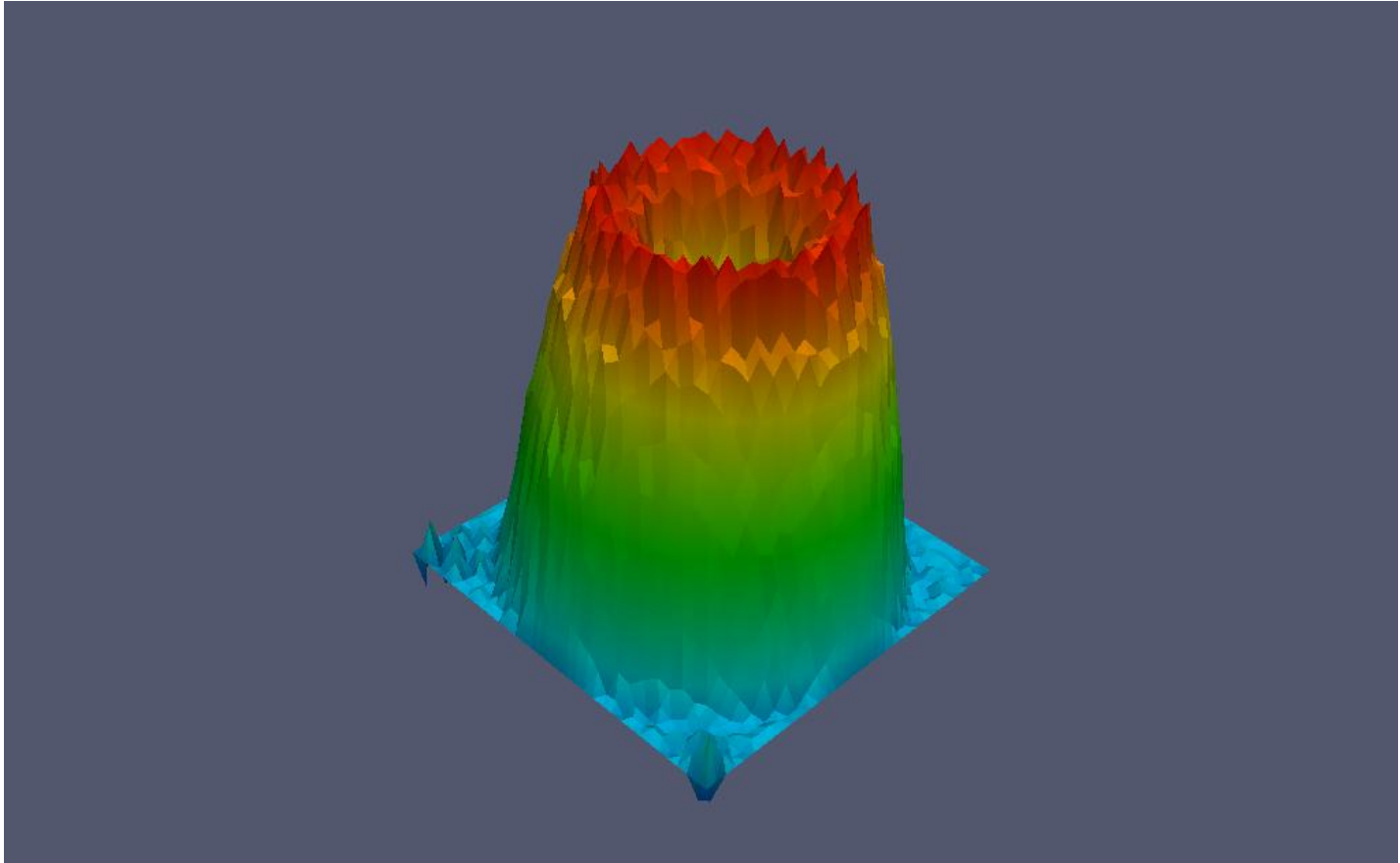
```
6 # Número de segmentos #
```

```
# Segmentos do contorno externo #
```

|    |   |   |   |
|----|---|---|---|
| 0: | 0 | 1 | 1 |
| 1: | 1 | 2 | 1 |
| 2: | 2 | 3 | 1 |
| 3: | 3 | 4 | 1 |
| 4: | 4 | 0 | 1 |
| 5: | 1 | 5 | 2 |



# Exemplo – Pudim (Paraview)



# Exemplo do site – Materiais

```
#-----#  
# Example 3 #  
#-----#
```

```
#####  
17 # Points #  
#####
```

```
0: 0 0 0.5 1  
1: 5 0 0.5 1  
2: 0 1 0.1 1  
3: 5 1 0.1 1  
4: 1.5 1.5 0.1 4  
5: 3.5 1.5 0.1 4  
6: 0 2.5 0.25 1  
7: 1.5 2.5 0.1 2  
8: 3.5 2.5 0.1 2  
9: 5 2.5 0.25 1  
10: 2.5 4 0.1 2  
11: 3.5 4 0.1 2
```

```
# line for coarsening #  
12: 2.5 2 0.4 0  
13: 3 3.5 0.4 0
```

```
# material markers #  
14: 3 3 0 1 # material 1 (red) #  
15: 3 1.25 0 2 # material 2 (blue) #  
16: 3 0.5 0 3 # material 3 (green) #
```

```
#####  
15 # Segments #  
#####
```

```
# domain contour #
```

```
0: 0 1 1  
1: 1 3 1  
2: 3 9 1  
3: 9 8 1  
4: 8 11 2  
5: 11 10 2  
6: 10 7 2  
7: 7 6 1  
8: 6 2 1  
9: 2 0 1
```

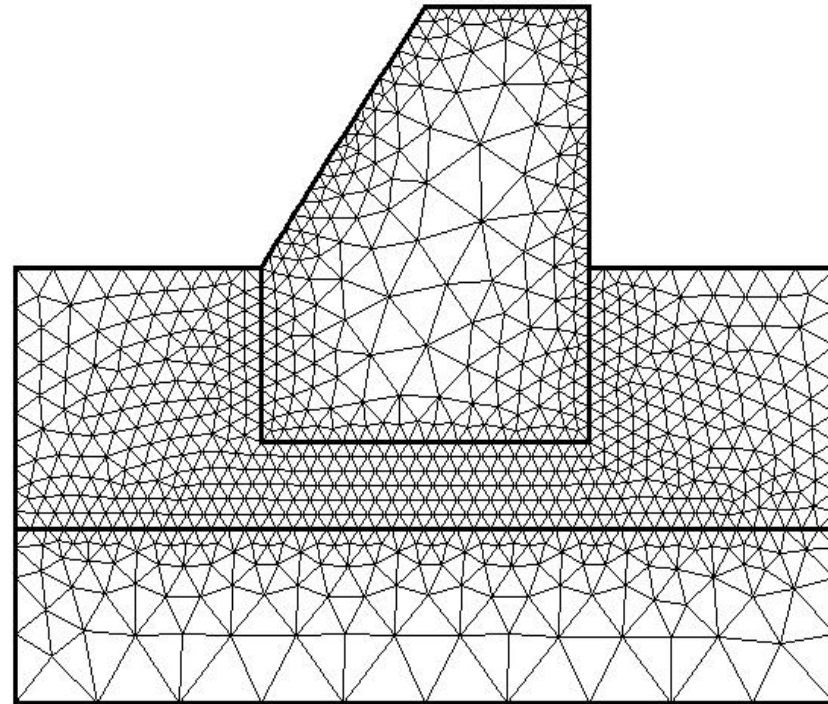
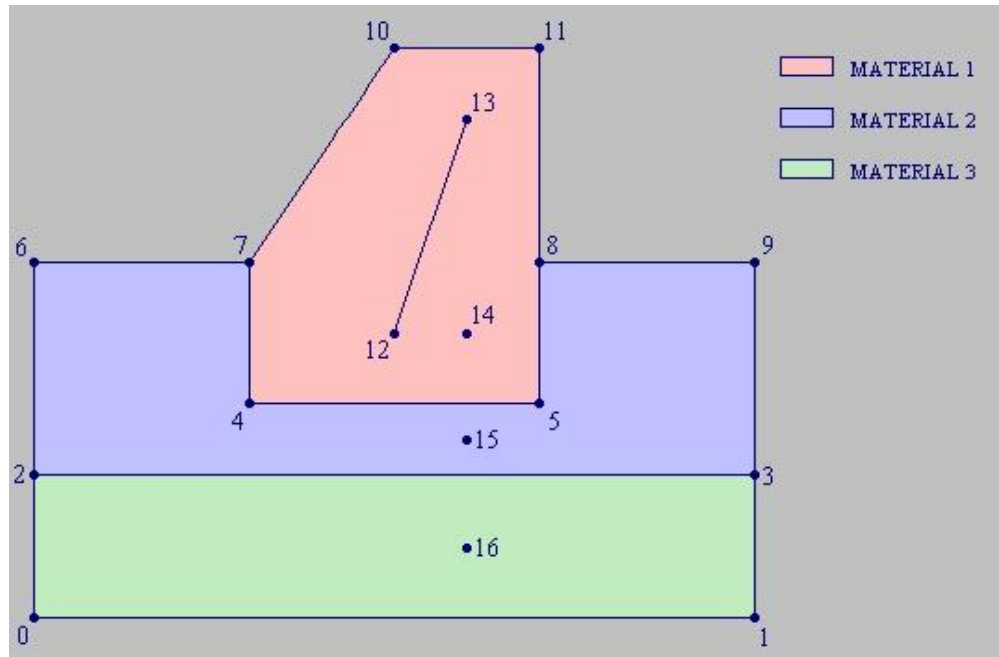
```
# frontier between green and blue #
```

```
10: 3 2 3
```

```
# frontier between blue and red#
```

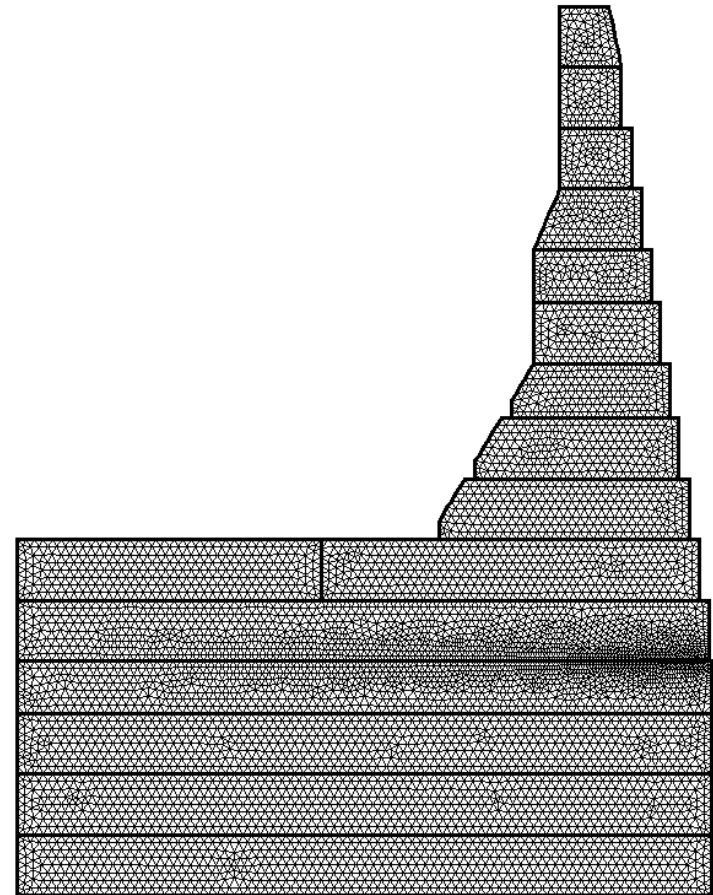
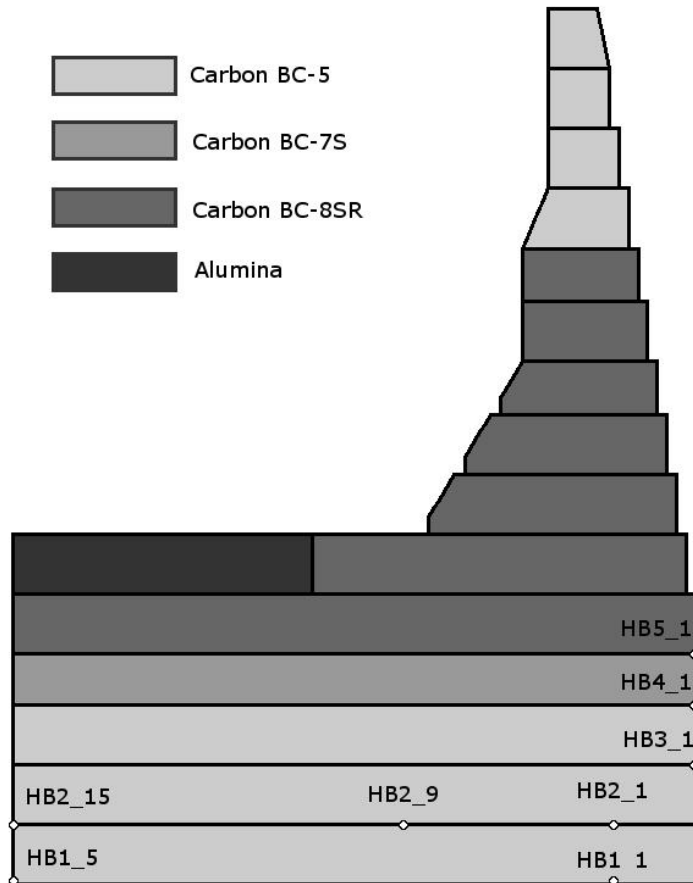
```
11: 8 5 4  
12: 5 4 4  
13: 4 7 4  
14: 12 13 0 |
```

# Exemplo do site – Materiais

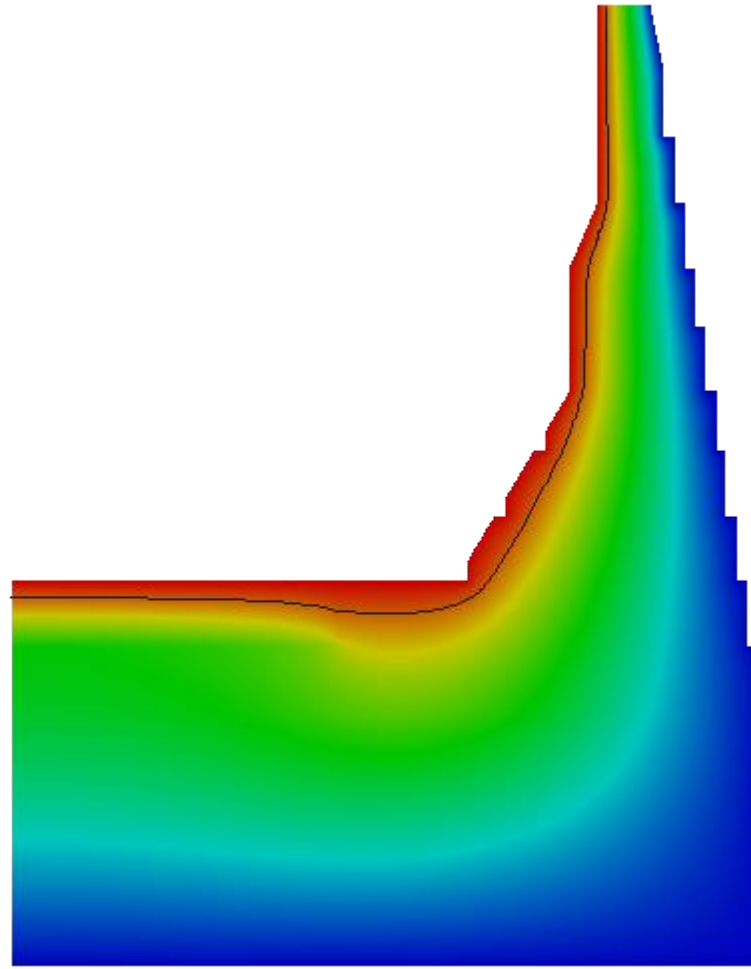




# Exemplo – Cadinho

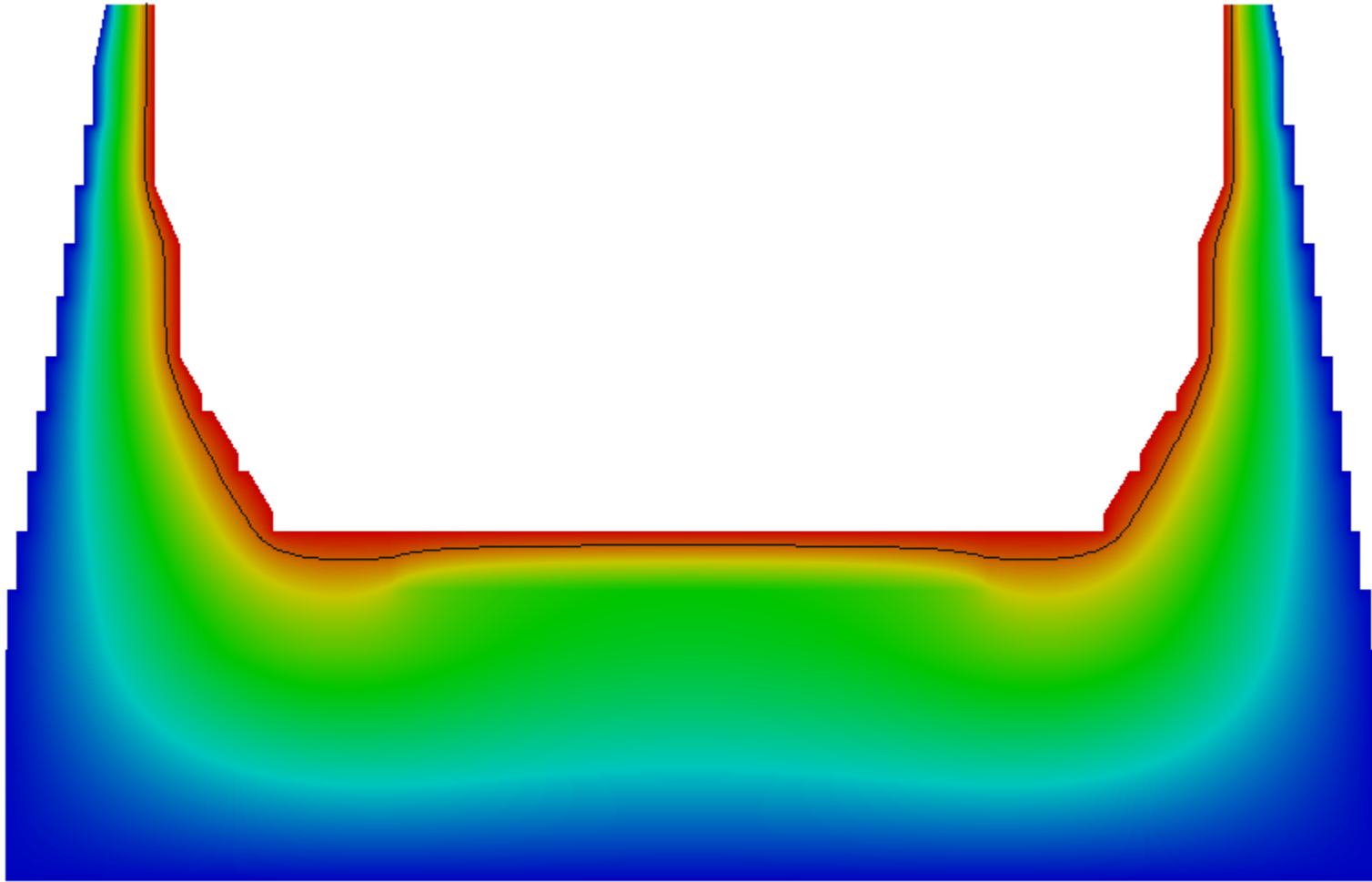


# Exemplo – Cadinho





# Exemplo – Cadinho



# Paraview – Arquivo de entrada

- ▶ Vtk
- ▶ Unstructured Grid

```
<VTKFile type="UnstructuredGrid" version="0.1" byte_order="BigEndian">
  <UnstructuredGrid>
    <Piece NumberOfPoints="8" NumberOfCells="6">
      <PointData Scalars="scalars">
        <DataArray type="Float32" Name="pressure" Format="ascii">
          0.1
          0.2
          ...
        </DataArray>
      </PointData>
      <Points>
        <DataArray type="Float32" NumberOfComponents="3" Format="ascii">
          0.0 0.0 0.0
          1.0 0.0 0.0
          ...
        </DataArray>
      </Points>
      <Cells>
        <DataArray type="Int32" Name="connectivity" Format="ascii">
          0 1 3
          1 4 3
          ...
        </DataArray>
        <DataArray type="Int32" Name="offsets" Format="ascii">
          3
          6
          ...
        </DataArray>
        <DataArray type="Int32" Name="types" Format="ascii">
          5
          5
          ...
        </DataArray>
      </Cells>
    </Piece>
  </UnstructuredGrid>
</VTKFile>
```