

## ПРИМЕР: СМЕСИТЕЛЬ

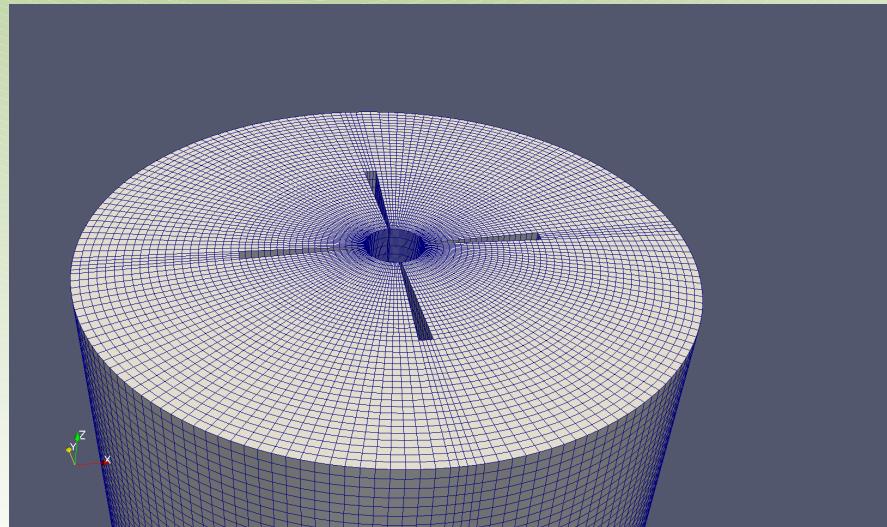
Эффективное использование открытых  
пакетов SALOME, CalculiX, OpenFOAM  
для создания расчётных сеток в задачах  
МСС

Калиш С.А. (НИЦ «Курчатовский институт»)  
Крапошин М.В. (НИЦ «Курчатовский институт»)  
Тагиров А.М. (НИЦ «Курчатовский институт»)  
Сибгатуллин И.Н. (НИИ механики МГУ им.  
Ломоносова)  
Стрижак С.В. (МГТУ им. Баумана)



# ПРИМЕР: СМЕСИТЕЛЬ

## Пример 4. Смеситель



Внешний диаметр: 100 мм

Диаметр вала: 10 мм

Высота: 200 мм

Длина лопастей: 50 мм

Количество лопастей: 4 шт.



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки с помощью утилиты BlockMesh

Редактируем файл blockMeshDict:

Раздел vertices:

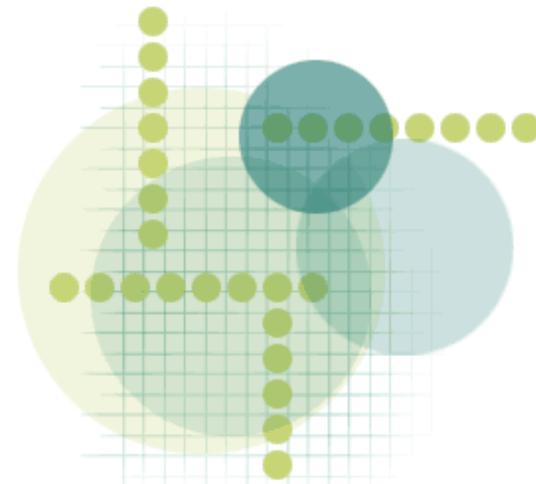
```
( 9.9905  0.4362 0.0000) // 0
( 0.4362  9.9905 0.0000) // 1
(-0.4362  9.9905 0.0000) // 2
(-9.9905  0.4362 0.0000) // 3
(-9.9905 -0.4362 0.0000) // 4
(-0.4362 -9.9905 0.0000) // 5
( 0.4362 -9.9905 0.0000) // 6
( 9.9905 -0.4362 0.0000) // 7
(50.0000  2.1800 0.0000) // 8
( 2.1800 50.0000 0.0000) // 9
(-2.1800 50.0000 0.0000) //10
(-50.0000 2.1800 0.0000) //11
(-50.0000 -2.1800 0.0000) //12
(-2.1800 -50.0000 0.0000) //13
( 2.1800 -50.0000 0.0000) //14
(50.0000 -2.1800 0.0000) //15
(74.9289  3.2669 0.0000) //16
( 3.2669 74.9289 0.0000) //17
(-3.2669 74.9289 0.0000) //18
(-74.9289 3.2669 0.0000) //19
(-74.9289 -3.2669 0.0000) //20
(-3.2669 -74.9289 0.0000) //21
( 3.2669 -74.9289 0.0000) //22
(74.9289 -3.2669 0.0000) //23
```

Также добавляем точки с теми же координатами по X  
и Y, но с Z = 200.

Точки для внешнего цилиндра:

```
( 99.9048  4.3619 0.0000) //48
( -4.3619 99.9048 0.0000) //49
(-99.9048 -4.3619 0.0000) //50
( 4.3619 -99.9048 0.0000) //51
```

Также добавляем точки с теми же координатами по X  
и Y, но с Z = 200.



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки с помощью утилиты BlockMesh

Раздел blocks:

Внутренняя часть смесителя

```
hex ( 0  8  9  1 24 32 33 25) (20 40 60) simpleGrading (1 1 1)    // 0
hex ( 2 10 11  3 26 34 35 27) (20 40 60) simpleGrading (1 1 1)    // 1
hex ( 4 12 13  5 28 36 37 29) (20 40 60) simpleGrading (1 1 1)    // 2
hex ( 6 14 15  7 30 38 39 31) (20 40 60) simpleGrading (1 1 1)    // 3
hex ( 8 16 17  9 32 40 41 33) (10 40 60) simpleGrading (1 1 1)    // 4
hex ( 9 17 18 10 33 41 42 34) (10 4 60) simpleGrading (1 1 1)    // 5
hex (10 18 19 11 34 42 43 35) (10 40 60) simpleGrading (1 1 1)    // 6
hex (11 19 20 12 35 43 44 36) (10 4 60) simpleGrading (1 1 1)    // 7
hex (12 20 21 13 36 44 45 37) (10 40 60) simpleGrading (1 1 1)    // 8
hex (13 21 22 14 37 45 46 38) (10 4 60) simpleGrading (1 1 1)    // 9
hex (14 22 23 15 38 46 47 39) (10 40 60) simpleGrading (1 1 1)    //10
hex (15 23 16  8 39 47 40 32) (10 4 60) simpleGrading (1 1 1)    //11
```

Внешняя часть смесителя

```
hex (16 48 49 18 40 52 53 42) (10 44 60) simpleGrading (1 1 1)    //12
hex (18 49 50 20 42 53 54 44) (10 44 60) simpleGrading (1 1 1)    //13
hex (20 50 51 22 44 54 55 46) (10 44 60) simpleGrading (1 1 1)    //15
hex (22 51 48 16 46 55 52 40) (10 44 60) simpleGrading (1 1 1)    //17
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки с помощью утилиты BlockMesh

Раздел edges:

```
arc 0 1 ( 7.0711 7.0711 0.0000)
arc 2 3 ( -7.0711 7.0711 0.0000)
arc 4 5 ( -7.0711 -7.0711 0.0000)
arc 6 7 ( 7.0711 -7.0711 0.0000)
arc 8 9 ( 35.3553 35.3553 0.0000)
arc 10 11 ( -35.3553 35.3553 0.0000)
arc 12 13 ( -35.3553 -35.3553 0.0000)
arc 14 15 ( 35.3553 -35.3553 0.0000)
arc 16 17 ( 53.0330 53.0330 0.0000)
arc 17 18 ( 0.0000 75.0000 0.0000)
arc 18 19 ( -53.0330 53.0330 0.0000)
arc 19 20 ( -75.0000 0.0000 0.0000)
arc 20 21 ( -53.0330 -53.0330 0.0000)
arc 21 22 ( 0.0000 -75.0000 0.0000)
arc 22 23 ( 53.0330 53.0330 0.0000)
arc 23 16 ( 75.0000 0.0000 0.0000)

arc 16 18 ( 53.0330 53.0330 0.0000)
arc 18 20 ( -53.0330 53.0330 0.0000)
arc 20 22 ( -53.0330 -53.0330 0.0000)
arc 22 16 ( 53.0330 53.0330 0.0000)
arc 48 49 ( 70.7107 70.7107 0.0000)
arc 49 50 ( -70.7107 70.7107 0.0000)
arc 50 51 ( -70.7107 -70.7107 0.0000)
arc 51 48 ( 70.7107 70.7107 0.0000)
```

Аналогично задаем дуги для верхнего торца



# ПРИМЕР: СМЕСИТЕЛЬ

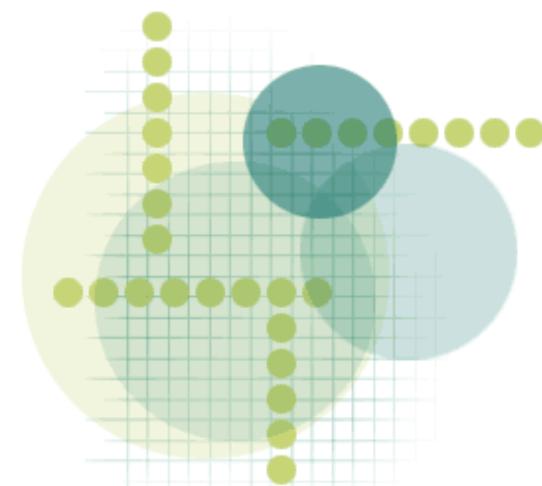
## Построение сетки с помощью утилиты

### BlockMesh

Раздел boundary:

```
inlet
{
    type patch;
    faces
    (
        (0 8 9 1)
        (2 10 11 3)
        (4 12 13 5)
        (6 14 15 7)
        (8 16 17 9)
        (9 17 18 10)
        (10 18 19 11)
        (11 19 20 12)
        (12 20 21 13)
        (13 21 22 14)
        (14 22 23 15)
        (15 23 16 8)
    );
}
```

```
outlet
{
    type patch;
    faces
    (
        (24 32 33 25)
        (26 34 35 27)
        (28 36 37 29)
        (30 38 39 31)
        (32 40 41 33)
        (33 41 42 34)
        (34 42 43 35)
        (35 43 44 36)
        (36 44 45 37)
        (37 45 46 38)
        (38 46 47 39)
        (39 47 40 32)
        (40 52 53 42)
        (42 53 54 44)
        (44 54 55 46)
        (46 55 52 40)
    );
}
```



# ПРИМЕР: СМЕСИТЕЛЬ

Построение сетки с помощью утилиты

BlockMesh

```
wall_inside
{
    type wall;
    faces
    (
        ( 0 1 25 24)
        ( 1 9 33 25)
        ( 9 10 34 33)
        (10  2 26 34)
        ( 2 3 27 26)
        ( 3 11 35 27)
        (11 12 36 35)
        (12  4 28 36)
        ( 4 5 29 28)
        ( 5 13 37 29)
        (13 14 38 37)
        (14  6 30 38)
        ( 6 7 31 30)
        ( 7 15 39 31)
        (15  8 32 39)
        ( 8 0 24 32)
    );
}

wall_outside
{
    type wall;
    faces
    (
        (48 49 53 52)
        (49 50 54 53)
        (50 51 55 54)
        (51 48 52 55)
    );
}

ami_outside
{
    type wall;
    faces
    (
        (16 18 42 40)
        (18 20 44 42)
        (20 22 46 44)
        (22 16 40 46)
    );
}
```



# ПРИМЕР: СМЕСИТЕЛЬ

Построение сетки с помощью утилиты

BlockMesh

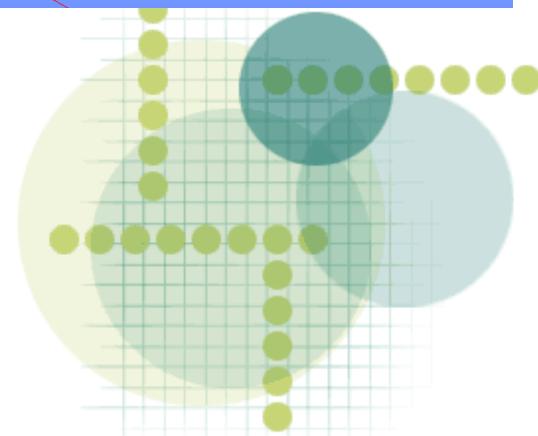
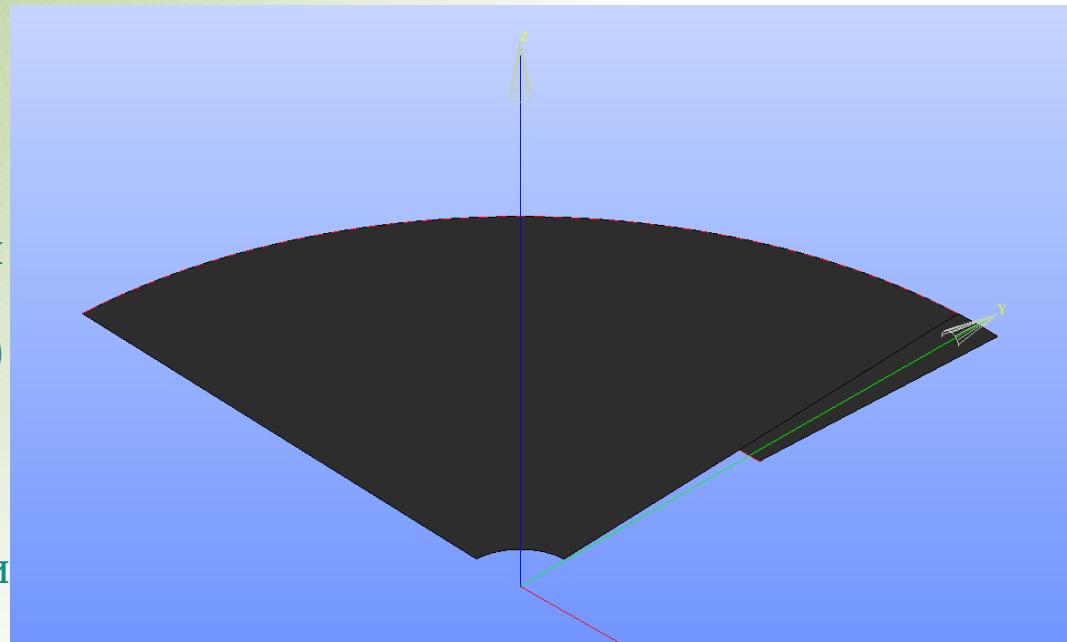
```
ami_inside
{
    type wall;
    faces
    (
        (16 17 41 40)
        (17 18 42 41)
        (18 19 43 42)
        (19 20 44 43)
        (20 21 45 44)
        (21 22 46 45)
        (22 23 47 46)
        (23 16 40 47)
    );
}
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки в SALOME

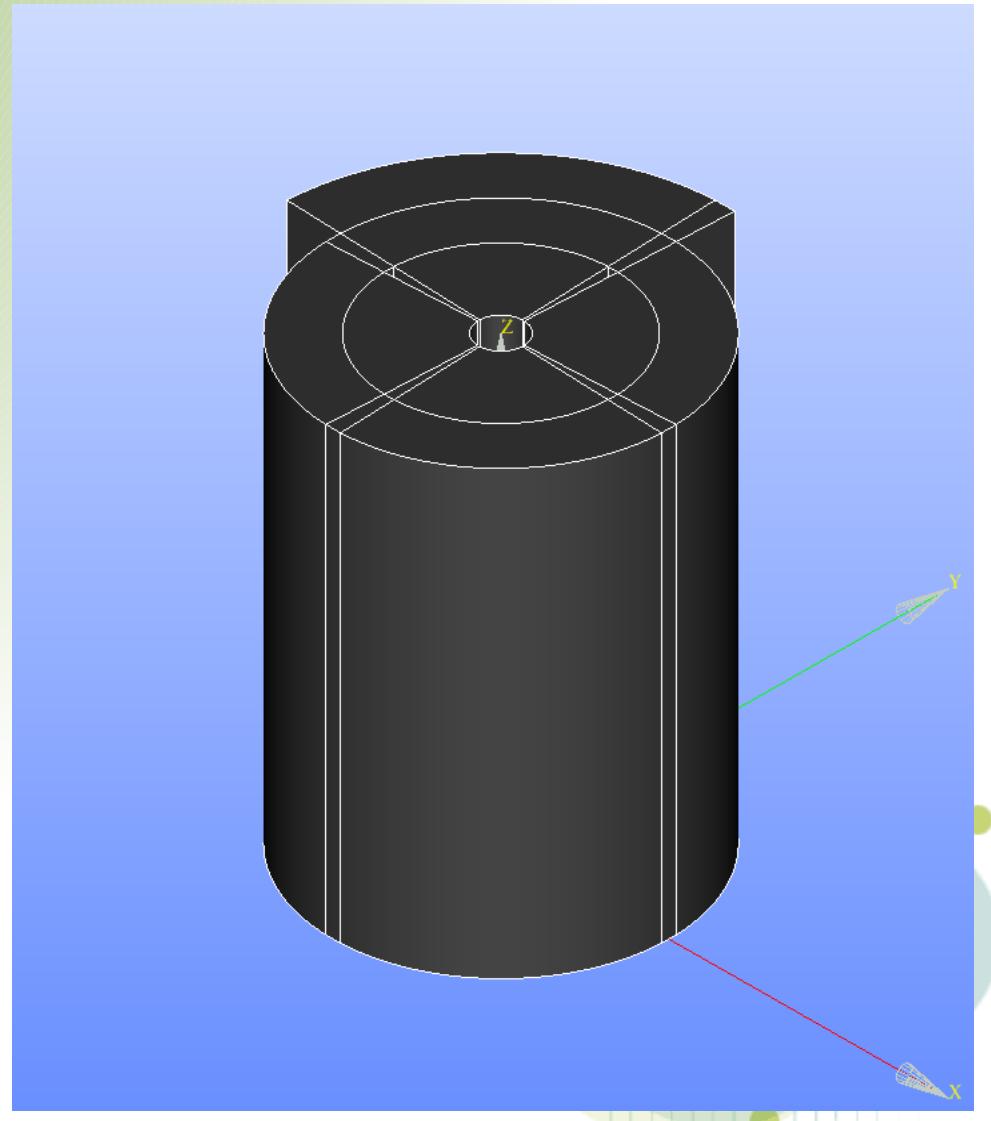
- Строим две окружности с центром в точке **O** и нормалью по оси **Z** с радиусами 10 и 100 (окружности 1 и 2).
- Строим две вершины лопасти (точки 1 (2,18 50 0) и 2 (-2,18 50 0)).
- Строим прямые проходящие через **O** и вершины лопастей (линии 1 и 2) и прямую соединяющие вершины лопасти (линия 3).
- Вытягиваем точку **O** вдоль линий 1 и 2 на 110 (вытягивания 1 и 2)
- Поворачиваем вытягивание 2 на 90 градусов вокруг оси **Z** (поворот 1)
- Разрезаем окружности 1 и 2 вытягиванием 1 и 2 и поворотом 1
- На дугах получившихся в результате разрезания и линии 3 строим две квадратичных грани.



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки в SALOME

- Строим окружность радиусом 75, центром в точке **O** и нормалью по оси **Z** (окружность 3) и с центром в точке **O** и по точкам 1 и 2 (окружность 4).
- Разрезаем квадратичные грани 1 и 2 окружностями 3 и 4.
- Выделяем две группы поверхностей: группа 1 – поверхности внутри окружности 3; группа 2 – поверхности снаружи окружности 3;
- Вытягиваем группы 1 и 2 на 200 вдоль оси **Z** (вытягивание 3 и 4).
- Применяем к вытягиваниям 3 и 4 операцию *multi Rotation* вокруг оси **Z**, количество 4.
- Применяем к объектам многократный поворот 1 и 2 операцию *Glue Faces*.



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки в SALOME

- На объекте склеивание 1 выделяем следующие группы:  
*R\_1* – ребра вдоль лопатки смесителя;  
*R\_2* – ребро продолжающее ребро лопатки смесителя;  
*Az* – большие дуги окружностей;  
*H* – ребро вдоль оси *Z*;  
*inlet* – нижний торец;  
*outlet* – верхний торец;  
*wall\_inside* – поверхность лопаток и вала;  
*ami\_inside* – внешняя цилиндрическая поверхность;



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки в SALOME

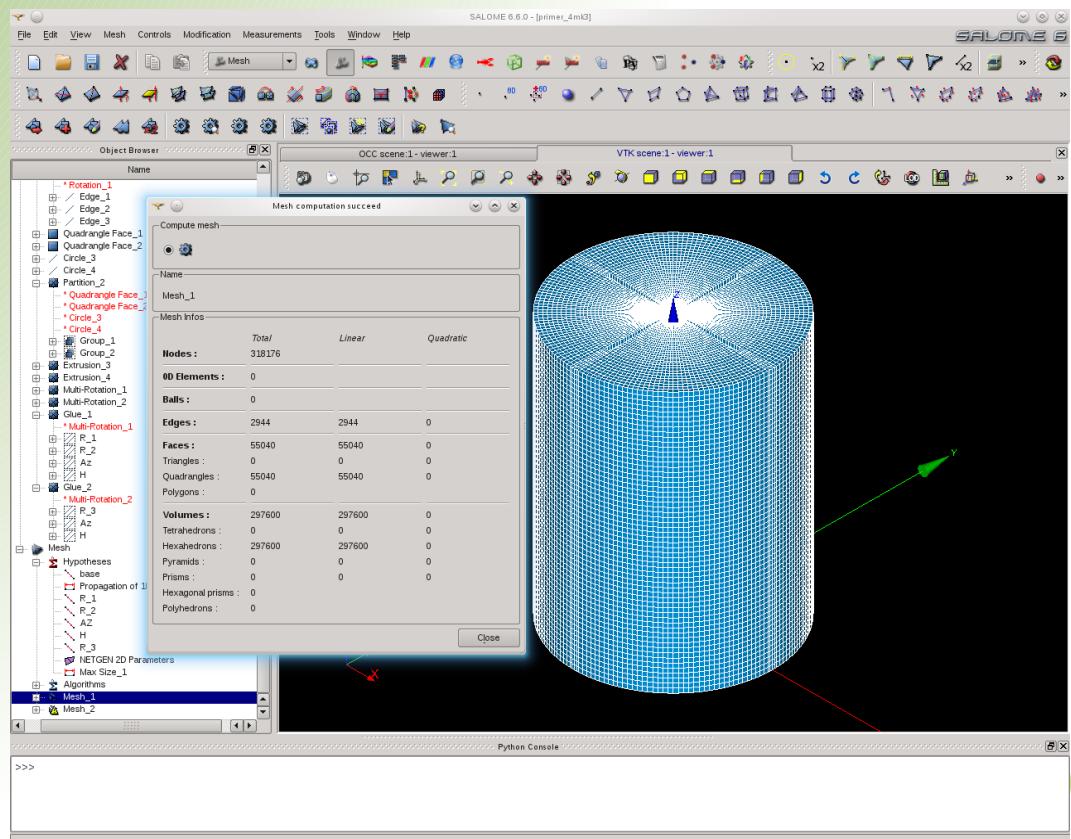
- На объекте склейивание 2 выделяем следующие группы:  
*R\_3* – радиус кольца;  
*Az* – большие дуги окружностей;  
*H* – ребро вдоль оси Z;  
*inlet* – нижний торец;  
*outlet* – верхний торец;  
*wall\_outside* – внешняя цилиндрическая поверхность;  
*ami\_outside* – внутренняя цилиндрическая поверхность;



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки в SALOME

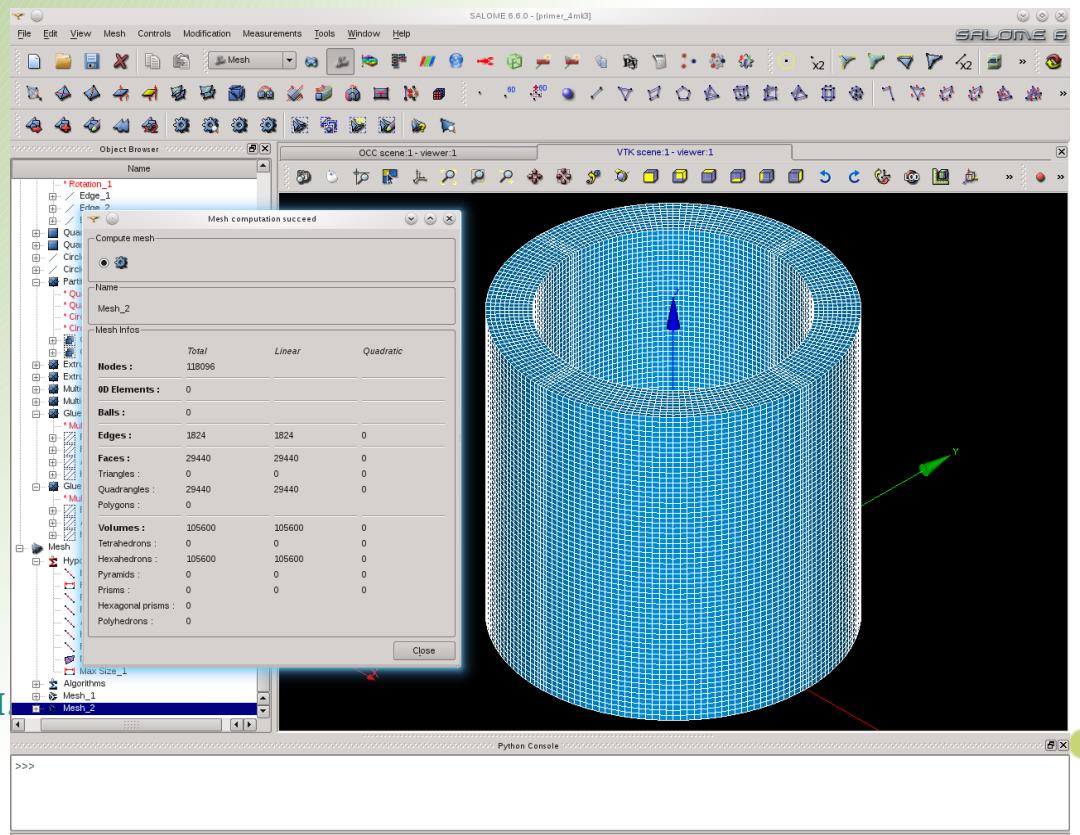
- Создаем сетку со следующими параметрами:  
*Geometry – Glue\_1*  
*3D algorithm – Hexaedral (i,j,k)*  
*2D algorithm – Quadrangle(mapping)*  
*1D algorithm – Wire Discretization*  
*1D Hypothesis – Nb.Segments = 4*
- Создаем под сетки для ребер:  
*Az – Nb.Segments = 40*  
*R\_1 – Nb.Segments = 20,*  
*R\_2 – Nb.Segments = 10*  
*H – Nb.Segments = 60,*
- Добавляем группы поверхностей



# ПРИМЕР: СМЕСИТЕЛЬ

## Построение сетки в SALOME

- Создаем сетку со следующими параметрами:  
*Geometry – Glue\_2*  
*3D algorithm – Hexaedral (i,j,k)*  
*2D algorithm – Quadrangle(mapping)*  
*1D algorithm – Wire Discretization*  
*1D Hypothesis – Nb.Segments = 4*
- Создаем под сетки для ребер:  
*Az – Nb.Segments = 40*  
*R\_3 – Nb.Segments = 10*  
*H – Nb.Segments = 60,*
- Добавляем группы поверхностей
- Объединяем две получившиеся сетки
- Экспортируем объединенную сетку в *.unv*-файл



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- Создаем фоновую сетку:

Точки: (-150 -150 -50), (150 -150 -50),  
(150 150 -50), (-150 150 -50),  
(-150 -150 250), (150 -150 250),  
(150 150 250), (-150 150 250)

Блок: **hex** (0 1 2 3 4 5 6 7) (30 30 30)

**simpleGrading** (1 1 1)

- В SALOME на основе группы **wall\_inside** строим сетку с параметрами:

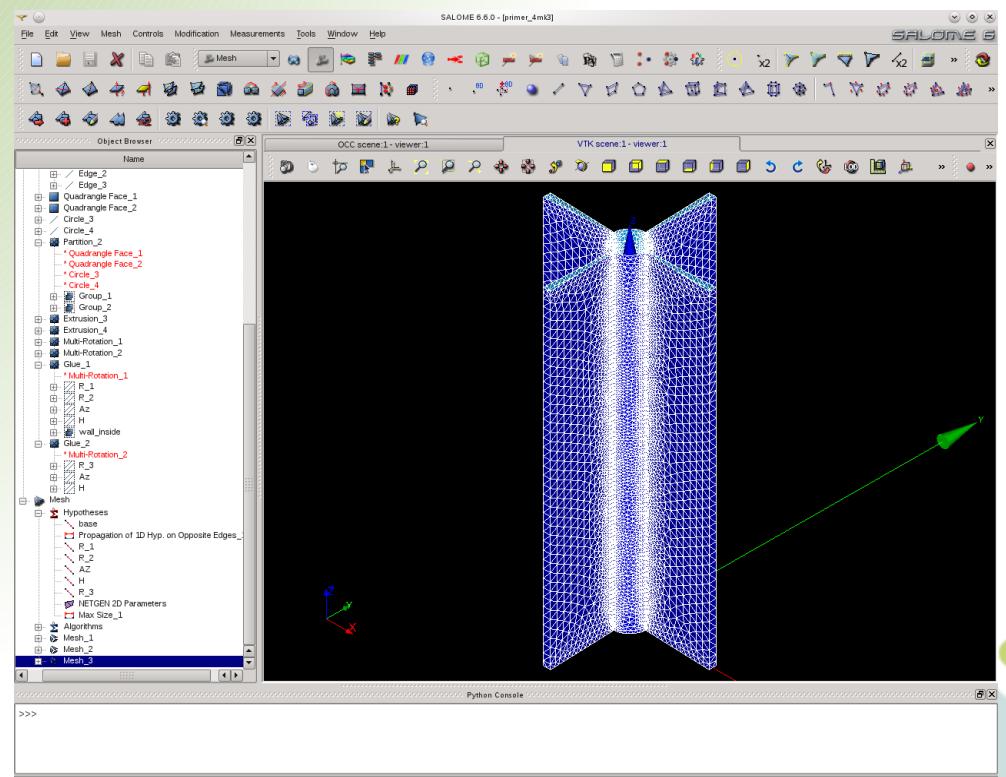
**2D algorithm – Netgen 1D-2D**

**2D Hypothesis – Netgen 2D parameters**

**Max. Size = 5**

**Min. Size = 0,5**

**Fineness – Fine**



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- В SALOME строим цилиндр в точке  $O$ , осью вдоль оси Z, высотой 200 и радиусом 100. На нем строим сетку с параметрами:  
*2D algorithm – Netgen 1D-2D*  
*2D Hypothesis – Netgen 2D parameters*  
*Max. Size = 0,5*  
*Min. Size = 0,1*  
*Fineness – Fine*

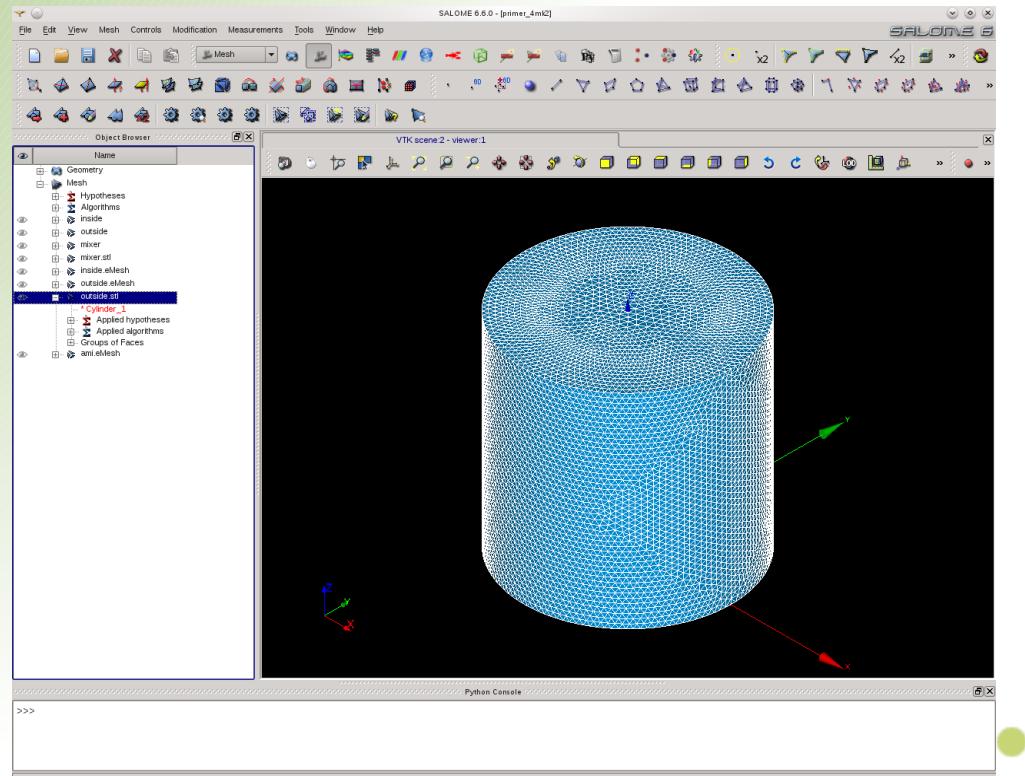
При этом выделяем следующие группы:

wall\_outside – боковая поверхность;

inlet – нижний торец;

outlet – верхний торец;

Создаем **STL**-файл с именнованными областями как показано в примере с балкой.



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- В *snappyHexMeshDict* в разделе *Geometry* создаем запись:

```
mixer.stl
{
    type triSurfaceMesh;
    name inside;
}
ami
{
    type searchableCylinder;
    point1 (0 0 0);
    point2 (0 0 200);
    radius 75;
}
```

```
outside.stl
{
    type triSurfaceMesh;
    name outside;
    regions
    {
        inlet
        {
            name inlet;
        }
        outlet
        {
            name outlet;
        }
    }
}
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- В разделе *features* создаем запись:

```
{  
    file "inside.eMesh";  
    level 3;  
}  
  
{  
    file "outside.eMesh";  
    level 2;  
}  
  
{  
    file "ami.eMesh";  
    level 2;  
}
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- В разделе *refinementSurfaces* создаем запись:  
*inside*

```
{  
    level (3 3);  
}  
outside  
{  
    level (2 2);  
}
```

```
ami  
{  
    level (3 3);  
    cellZone ami;  
    faceZone ami;  
    cellZoneInside inside;  
}
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- В разделе *refinementRegions* создаем запись:

```
ami
{
    mode inside;
    levels ((10. 3));
}
outside
{
    mode inside;
    levels ((10. 2));
}
```

- Параметр *locationInMesh* (19 21 101)



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- В разделе *layers* создаем запись:

```
"mixer_*"  
{  
    nSurfaceLayers 2;  
}  
expansionRatio 2.0;  
finalLayerThickness 0.5;  
minThickness 0.25;  
nGrow 0;
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- Создаем файл *removeZones.topoSetDict* в нем запись:

```
actions
(
    {
        name ami;
        type cellZoneSet;
        action remove;
    }
);

);
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- Создаем файл *ami.topoSetDict* в нем запись:

```
actions
(
{
    name ami;
    type cellSet;
    action new;
    source cylinderToCell;
    sourceInfo
    {
        p1 (0 0 0);
        p2 (0 0 200);
        radius 75;
    }
}
{
    name outside;
    type cellSet;
    action new;
    source cellToCell;
    sourceInfo
    {
        set ami;
    }
}
```

```
{
    name outside;
    type cellSet;
    action invert;
}

{
    name ami;
    type cellZoneSet;
    action new;
    source setToCellZone;
    sourceInfo
    {
        set ami;
    }
}
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

```
{  
    name amiFace;  
    type faceSet;  
    action new;  
    source cellToFace;  
    sourceInfo  
    {  
        set ami;  
        option all;  
    }  
}  
  
{  
    name amiFace;  
    type faceSet;  
    action subset;  
    source cellToFace;  
    sourceInfo  
    {  
        set outside;  
        option all;  
    }  
}  
  
{  
    name ami;  
    type faceZoneSet;  
    action new;  
    source setsToFaceZone;  
    sourceInfo  
    {  
        faceSet amiFace;  
        cellSet ami;  
    }  
};
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- Создаем файл *changeDictionaryDict* в нем запись:

```
dictionaryReplacement
```

```
{
```

```
    boundary
```

```
{
```

```
    ami_inside
```

```
{
```

```
        type      cyclicAMI;
```

```
        nFaces    0;
```

```
        startFace 6169944;
```

```
        neighbourPatch ami_outside;
```

```
        transform   noOrdering;
```

```
        surface
```

```
{
```

```
}
```

```
}
```

```
ami_outside
```

```
{
```

```
    type      cyclicAMI;
```

```
    nFaces    0;
```

```
    startFace 6169944;
```

```
    neighbourPatch ami_inside;
```

```
    transform   noOrdering;
```

```
    surface
```

```
{
```

```
}
```

```
}
```

```
}
```

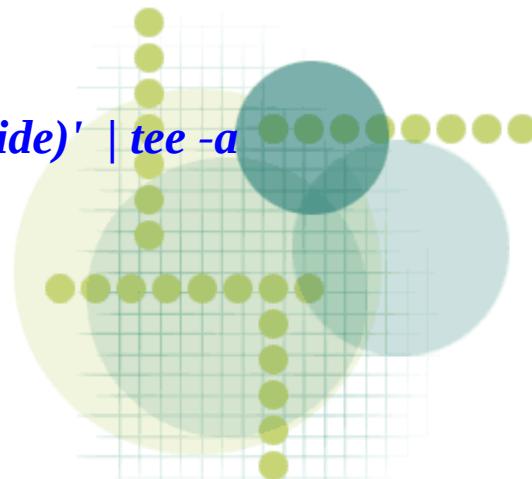


# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой snappyHexMesh

- Создаем скрипт *mkMesh* в нем запись:

```
#!/bin/sh
rm -rf *.log
# - meshing
blockMesh
snappyHexMesh -overwrite | tee -a snappy.log
# - generate face/cell sets and zones
topoSet -dict system/removeZones.topoSetDict | tee -a remove.log
topoSet -dict system/ami.topoSetDict | tee -a createAmi.log
# - create the AMI faces by creating baffles, and then splitting the mesh
changeDictionary | tee -a createAmi.log
# force removal of fields generated by snappy
\rm -rf 0
createBaffles -internalFacesOnly -overwrite ami '(ami_inside ami_outside)' | tee -a
createAmi.log
mergeOrSplitBaffles -split -overwrite | tee -a createAmi.log
# - apply the initial fields
cp -rf 0.org 0
```



# ПРИМЕР: СМЕСИТЕЛЬ

## Создание сетки утилитой `snappyHexMesh`

- Для генерации сетки запускаем скрипт `mkMesh`

