

# Создание блочно-структурированных сеток - blockMesh

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

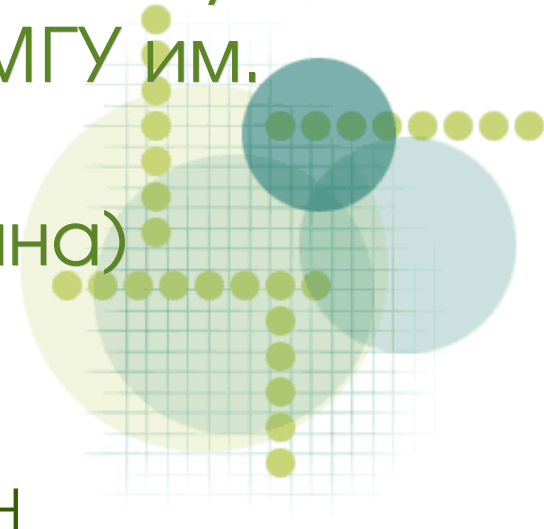
Калиш С.А. (НИЦ «Курчатовский институт»)

Крапошин М.В. (НИЦ «Курчатовский институт»)

Тагиров А.М. (НИЦ «Курчатовский институт»)

Сибгатуллин И.Н. (НИИ механики МГУ им.  
Ломоносова)

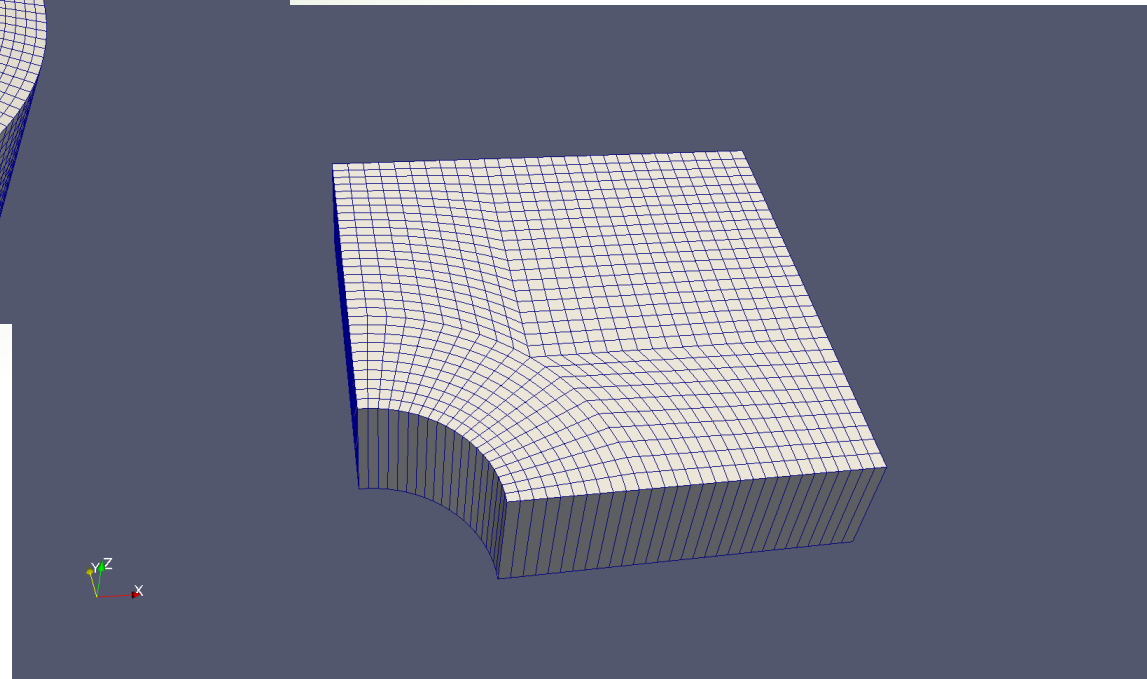
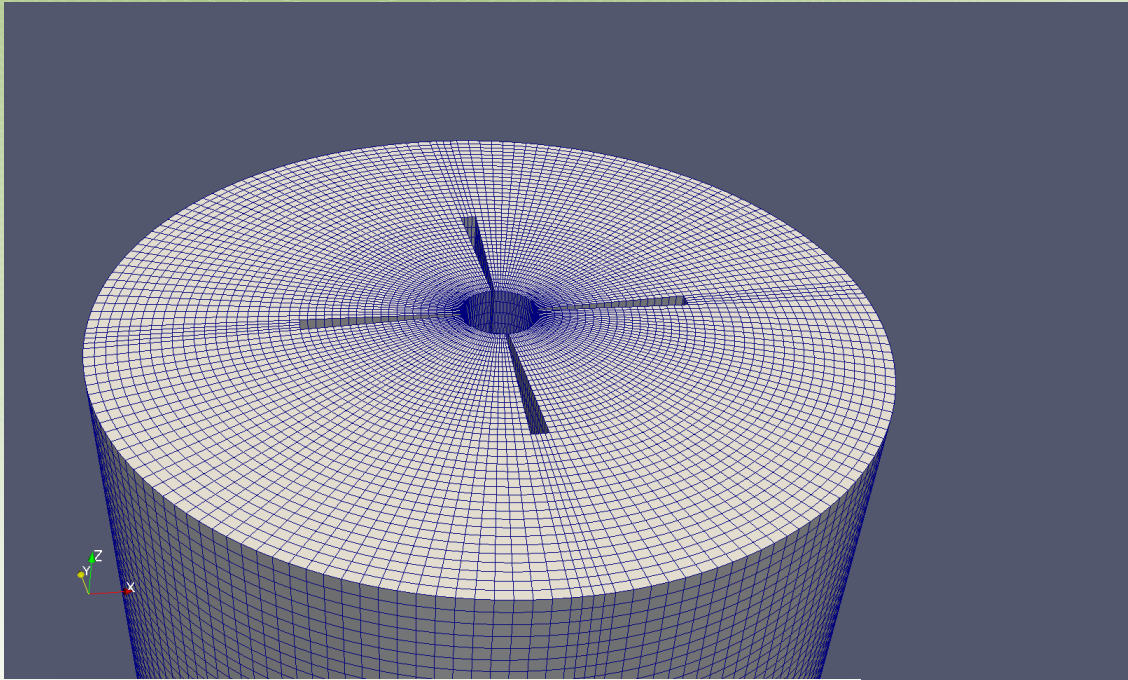
Стрижак С.В. (МГТУ им. Баумана)



# Создание блочно-структурированных сеток - blockMesh

## BlockMesh

Утилита blockMesh поставляемая вместе с пакетом openFOAM предназначена для генерации гексаэдральных структурированных сеток.



# Создание блочно-структурированных сеток - blockMesh

## Файл blockMeshDict

Для построения сетки необходим файл blockMeshDict, который имеет следующий вид:

```
FoamFile
{
  version    2.0;
  format     ascii;
  class      dictionary;
  object     blockMeshDict;
}
convertToMeters 1;
vertices
(
);
blocks
(
);
edges
(
);
boundary
(
);
mergePatchPairs
(
);
```

ConvertToMeters – параметр показывающий масштаб

В разделе vertices задаются координаты точек для построения геометрии

В разделе blocks указывается тип блока, его вершины и параметры разбиения блока

В разделе edges криволинейные ребра ( дуги окружностей или сплайны)

В разделе boundary определяются поверхности для задания граничных условий.

В разделе mergePatchPairs задаются пары поверхностей для слияния





# Создание блочно-структурированных сеток - blockMesh

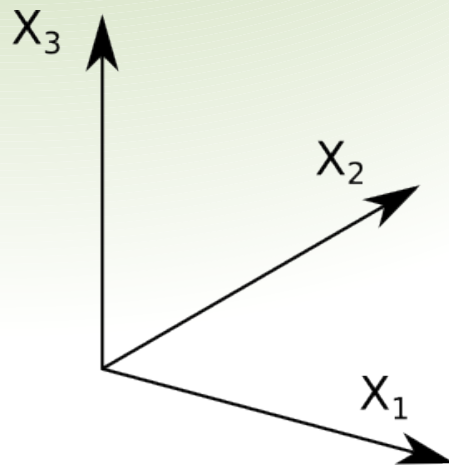
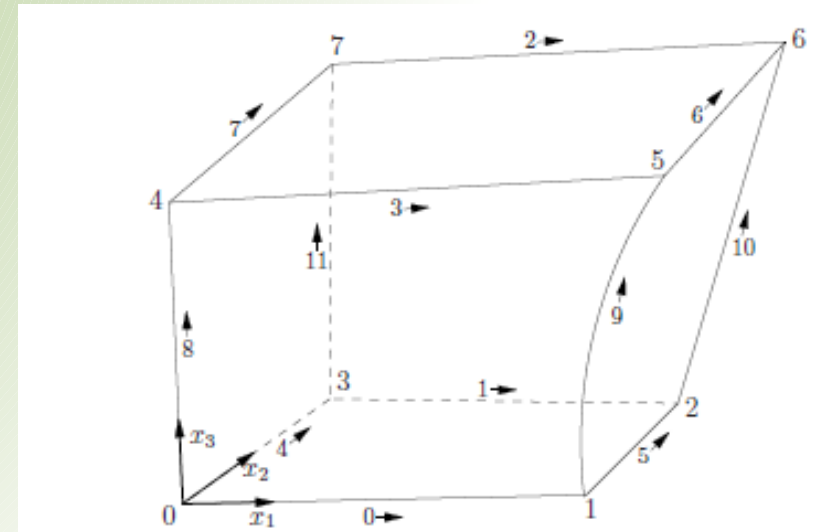
## Раздел Blocks

Блок задается следующей записью

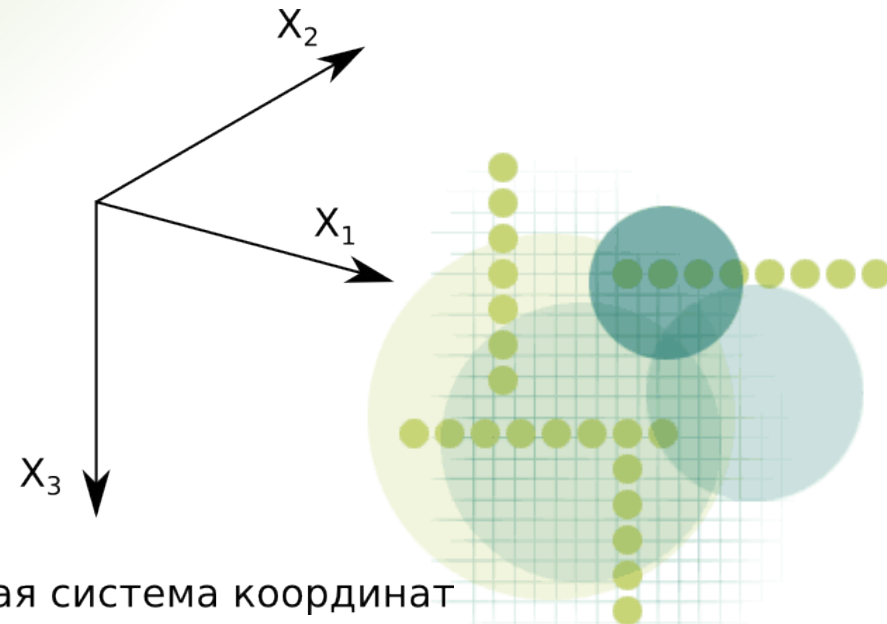
**hex** (0 1 2 3 4 5 6 7) (20 20 20) **simpleGrading** (1 2 3)

Первая группа чисел номера точек в вершинах блока.

Порядок, в котором указываются точки при задании блока, определяют какая будет система координат (левая или правая). У всех блоков порядок должен быть одинаковый, иначе появятся блоки с отрицательным объемом.



Правая система координат

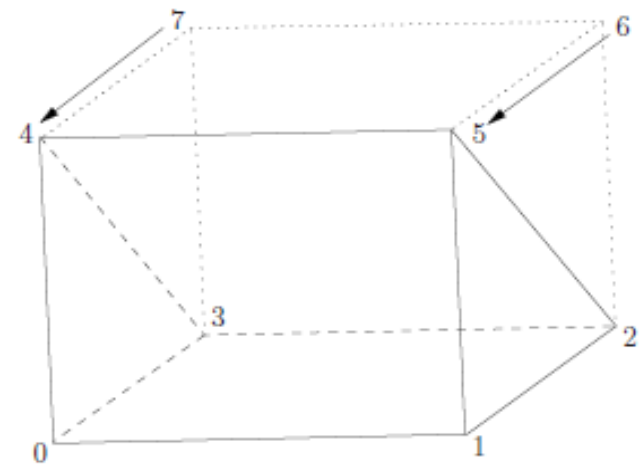


Левая система координат

## Раздел Blocks

Можно строить блоки с количеством вершин меньше чем 8, для этого надо часть вершин указать несколько раз например *hex (0 1 2 3 4 5 5 4)*.

В данном случае мы получаем треугольную призму.



## Раздел Blocks

*hex* (0 1 2 3 4 5 6 7) (20 20 20) *simpleGrading* (1 2 3)

Вторая группа показывают число разбиений вдоль соответствующих осей

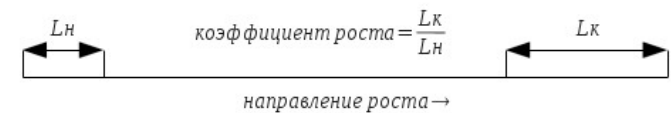
*hex* (0 1 2 3 4 5 6 7) (20 20 20) *simpleGrading* (1 2 3)

Третья группа определяет коэффициент роста ячеек.

Коэффициент равен отношению длин последней первой ячейки.

Если стоит параметр *simpleGrading*, то задаются коэффициенты вдоль соответствующих осей.

В случае параметра *edgeGrading*, задаются коэффициенты для каждого ребра, и запись принимает вид *edgeGrading* (1 1 1 1 2 2 2 2 3 3 3 3)



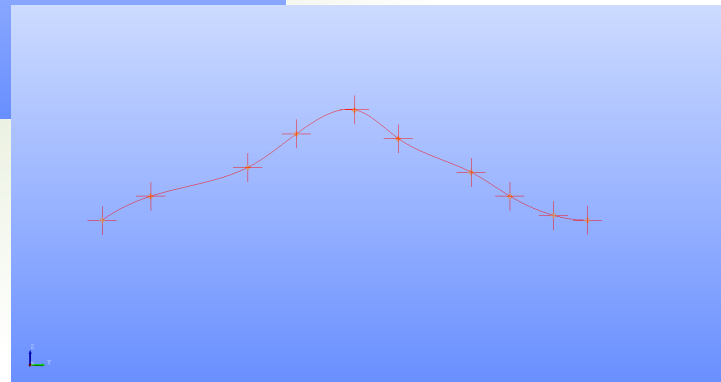
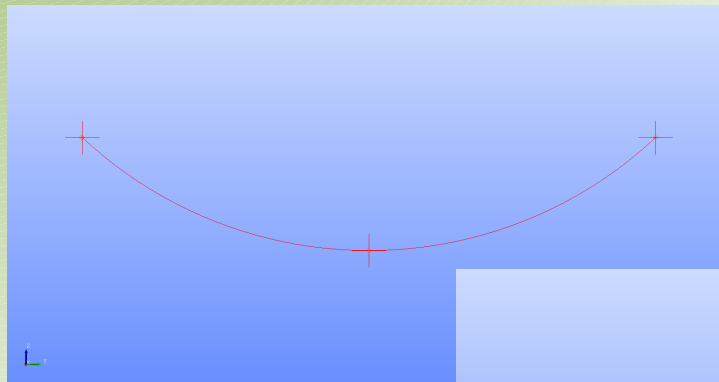


## Раздел edges

Ребра задаются двумя точками, типом ребра и дополнительными параметрами. Для окружности это точка на окружности, для сплайна список точек для экстраполяции.

```
arc 0 1 (0 50 -20)
```

```
spline 1 2 (  
(0 10 5)  
(0 30 11)  
(0 40 18)  
(0 52 23)  
(0 61 17)  
(0 76 10)  
(0 84 5)  
(0 93 1)  
)
```

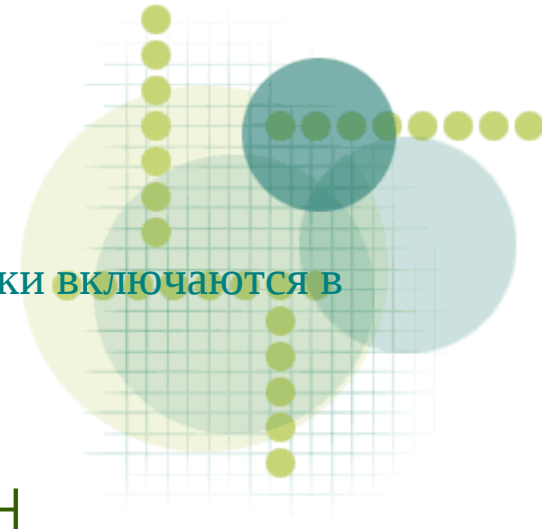


## Раздел boundary

Группы поверхностей задаются списком из граней, каждая из которых определяется четырьмя точками. Также необходимо указать тип граничного условия.

```
boundary
(
  boundary
  {
    type patch;
    faces
    (
      (0 1 2 3)
      (4 5 6 7)
    );
  }
);
```

Граничные поверхности неуказанные в данном разделе автоматически включаются в группу **defaultFaces** с типом **empty**.





## Раздел mergePatchPairs

Сетка может состоять из нескольких блоков.  
Соединение блоков может осуществляться двумя способами:

- ***Face matching*** (совпадение поверхностей) – в случае точного совпадения поверхностей двух граничащих блоков

В этом случае поверхности не указываются в разделе ***patches*** и они автоматически комбинируются в одну внутреннюю поверхность.



# Создание блочно-структурированных сеток - blockMesh

## Раздел mergePatchPairs

- **Face merging** (слияние поверхностей) – если поверхности двух блоков не совпадают по геометрии или разбиению.

В этом случае поверхности необходимо указать в разделе `patches`, а затем указать в разделе **`mergePatchPairs`** в виде ( `<masterPatch>` `<slavePatch>` ). Поверхность указанная как ***master*** не изменяется, точки поверхности ***slave*** перемещаются, так чтобы они совпадали с точками поверхности ***master***. Если поверхности перекрываются частично, то часть поверхностей, которая не сливается, остается внешней поверхностью, на которой необходимо задать граничные условия. Если поверхность сливается целиком, то после слияния она удаляется из списка поверхностей.

