



Школа-семинар
**«ОСНОВЫ ИСПОЛЬЗОВАНИЯ
OpenFOAM, SALOME и ParaView»**

**ДЕМОНСТРАЦИЯ: ТЕЧЕНИЕ В
КАВЕРНЕ**

*М.В. Крапошин (Институт системного программирования РАН)
О.И. Самоваров (Институт системного программирования РАН)
С.В. Стрижак (ГОУ ВПО МГТУ им. Баумана)*



Демонстрация работы с пакетами OPENFOAM и PARAVIEW на основе готовых примеров

Освещаемые разделы:

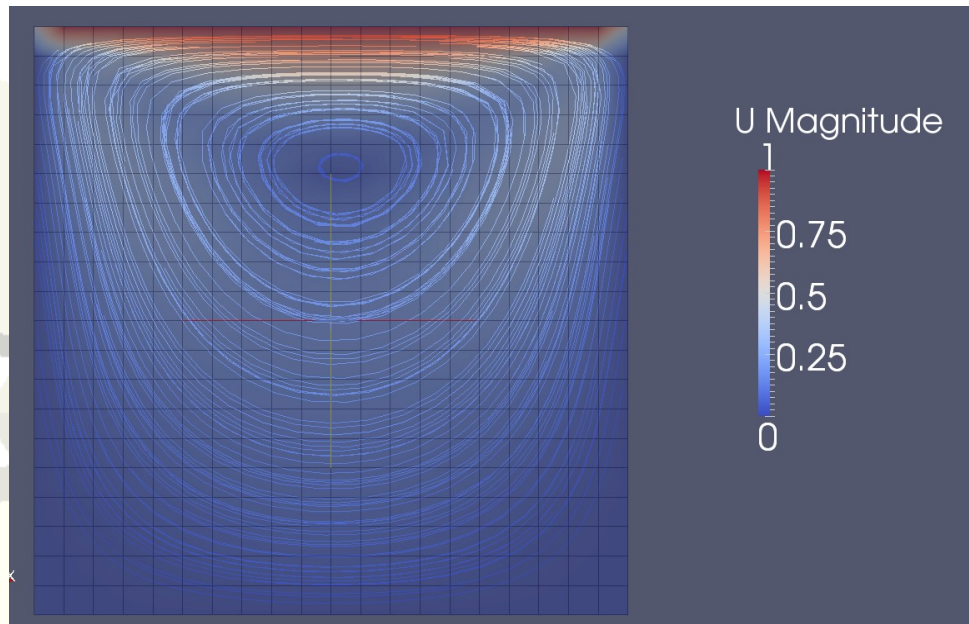
- I. Несжимаемое течение в каверне**
- II. Турбулентное течение за обратным уступом**
- III. Свободная конвекция в комнате с подогревом**
- IV. Построение гексаэдральных
структурированных сеток в SALOME**



I. ТЕЧЕНИЕ В КАВЕРНЕ

Здесь и далее предполагается следующий порядок действий:

- а) анализ физической постановки задачи
- б) выбор математических моделей
- в) подготовка сетки
- г) настройка константного окружения (расчетной модели)
- д) анализ результатов, визуализация



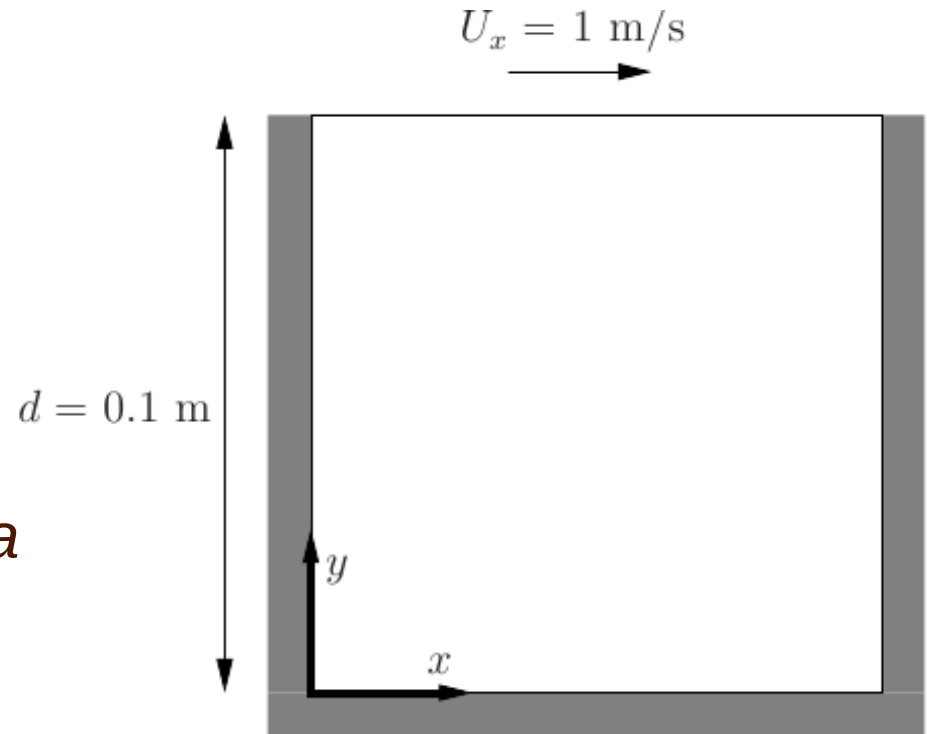


КАВЕРНА: КРАТКОЕ ОПИСАНИЕ

В задаче исследуется плоское течение несжимаемой Ньютоновской жидкости при числах Re , соответствующих ламинарному и переходному режимам: 100 и 1000, соответственно.

Схема с размерами представлена на рис. слева.

Движение жидкости в каверне происходит из-за равномерного перемещения верхней крышке — что равносильно постоянной скорости потока в направлении X



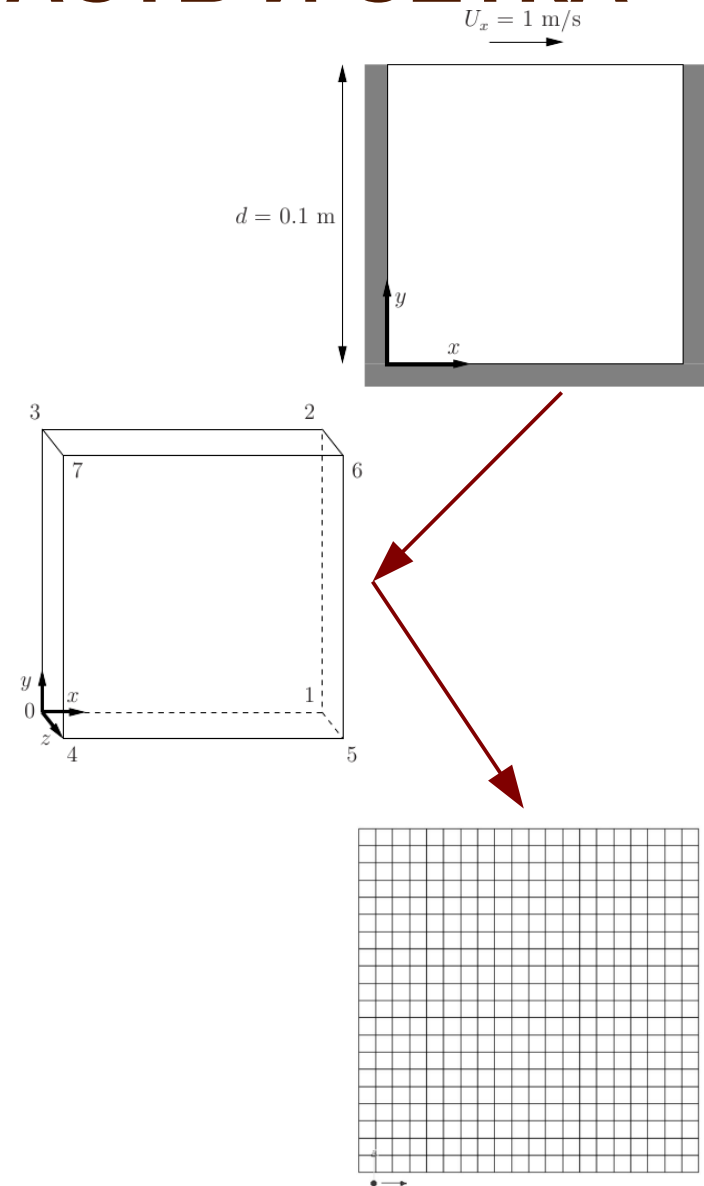


КАВЕРНА: РАСЧЕТНЫЕ ОБЛАСТЬ И СЕТКА

Рисунок с предыдущего слайда вполне отвечает нашему представлению о расчетной области. Наложим на неё расчетную сетку.

Для этого воспользуемся блочным генератором сеток, встроенным в OpenFOAM — `blockMesh`.

Для определения сетки `blockMesh` требуется следующая информация: описание узлов, описание блоков (объёмов) и поверхностей для задания граничных условий





КАВЕРНА: ВЗГЛЯД В СТОРОНУ: blockMesh (1)

Также, как и все приложения OpenFOAM, *blockMesh* запускается из командной строки и как большинство из них, не требует аргументов, а информацию считывает из управляющего файла

Располагается:

constant/polyMesh/blockMeshDict

Содержит:

- Параметр масштабирования *convertToMeters*
- Список узлов *vertices*
- Список блоков *blocks*
- Список криволинейных рёбер *edges*
- Список поверхностей *patches*
- Список объединяемых граней *mergePatchPairs*

```
16
17   convertToMeters 0.1;
18
19   vertices
20   (
21       (0 0 0)
22       (1 0 0)
23       (1 1 0)
24       (0 1 0)
25       (0 0 0.1)
26       (1 0 0.1)
27       (1 1 0.1)
28       (0 1 0.1)
29   );
30
```

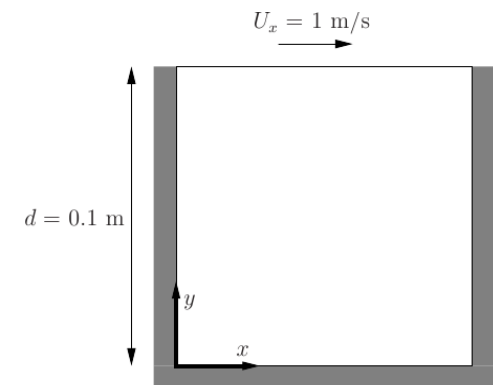
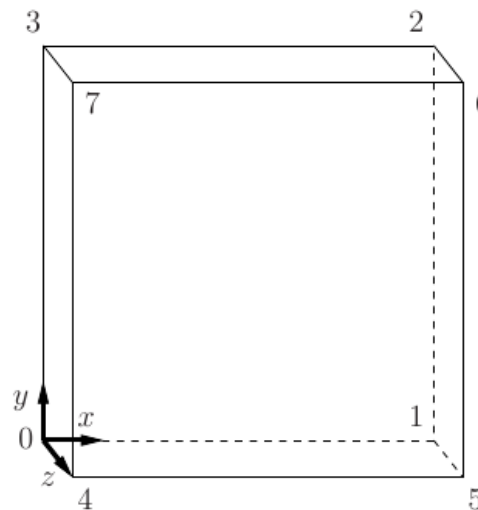


КАВЕРНА: ВЗГЛЯД В СТОРОНУ: blockMesh (2)

После определения точек следует сформировать гексаэдральные блоки — шестигранные объёмы с 8 узлами, 12 ребрами. Каждая грань имеет строго 4 ребра и 4 узла. Блоки задаются через перечисление узлов определенном порядке (например, против часовой стрелки).

Также задаются и внешние поверхности, определяющие ГУ

```
31 blocks
32 (
33     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34 );
35
36 edges
37 (
38 );
39
40 patches
41 (
42     wall movingWall
43     (
44         (3 7 6 2)
45     )
46     wall fixedWalls
47     (
48         (0 4 7 3)
49         (2 6 5 1)
50         (1 5 4 0)
51     )
52     empty frontAndBack
53     (
54         (0 3 2 1)
55         (4 5 6 7)
56     )
57 );
```





КАВЕРНА: ПРОВЕРКА СЕТКИ (1)

Перед запуском на расчет следует проверить качество сетки: ***checkMesh***

Time = 0

Mesh stats

points:	882
internal points:	0
faces:	1640
internal faces:	760
cells:	400
boundary patches:	3
point zones:	0
face zones:	0
cell zones:	0

Overall number of cells of each type:

hexahedra:	400
prisms:	0
wedges:	0
pyramids:	0
tet wedges:	0
tetrahedra:	0
polyhedra:	0

Checking topology...

Boundary definition OK.
Point usage OK.
Upper triangular ordering OK.
Face vertices OK.
Number of regions: 1 (OK).

Вывод программы ***checkMesh*** включает в себя следующие разделы (выделены жирным курсивом с подчеркиванием):

- ***общая статистика;***
- ***статистика типов ячеек;***
- ***общая топология сетки;***
- ***топология внешних границ;***
- ***геометрические характеристики (неортогональность, скособоченность, масштабность и пр.)***



КАВЕРНА: ПРОВЕРКА СЕТКИ (2)

Checking patch topology for multiply connected surfaces ...

Patch	Faces	Points	Surface topology
movingWall	20	42	ok (non-closed singly connected)
fixedWalls	60	122	ok (non-closed singly connected)
frontAndBack	800	882	ok (non-closed singly connected)

Checking geometry...

Overall domain bounding box (0 0 0) (0.1 0.1 0.01)

Mesh (non-empty, non-wedge) directions (1 1 0)

Mesh (non-empty) directions (1 1 0)

All edges aligned with or perpendicular to non-empty directions.

Boundary openness (8.47033e-18 -8.47033e-18 -4.51751e-17) OK.

Max cell openness = 1.35525e-16 OK.

Max aspect ratio = 1 OK.

Minumum face area = 2.5e-05. Maximum face area = 5e-05. Face area magnitudes OK.

Min volume = 2.5e-07. Max volume = 2.5e-07. Total volume = 0.0001. Cell volumes OK.

Mesh non-orthogonality Max: 0 average: 0

Non-orthogonality check OK.

Face pyramids OK.

Max skewness = 1e-08 OK.

Mesh OK.

End



КАВЕРНА: КРАЕВЫЕ УСЛОВИЯ (1)

Поля величин — файлы с соответствующим именем

Срезы времени (моменты) — папки с файлами, хранящими в себе рассчитываемые поля

Обычно папка с именем 0 (ноль) соответствует начальному условию

*На примере давления (p)
Каждый файл в себе хранит:*

- Заголовок (не показан);*
- размерность (dimensions);*
- значения в центрах ячеек (internalField)*
- значения на границе (boundaryField)*

```
dimensions      [0 2 -2 0 0 0 0];

internalField    uniform 0;

boundaryField
{
    movingWall
    {
        type      zeroGradient;
    }

    fixedWalls
    {
        type      zeroGradient;
    }

    frontAndBack
    {
        type      empty;
    }
}
```



КАВЕРНА: КРАЕВЫЕ УСЛОВИЯ (2)

Рассмотрим поле скорости (поле давления было выше).

Размерность — м/с, поэтому
dimensions [0 1 -1 0 0 0 0]

Поле скорости в нулевой момент времени невозмущенное, поэтому
internalField *uniform* (0 0 0)

На границах *movingWall* и *fixedWalls* действует условие «прилипания» - *uniform* (0 0 0) на *fixedWalls* (стоячая стенка) и *uniform* (1 0 0) на *movingWalls* (равномерно движущаяся сетка)

Наконец, специальный *min* ГУ — *empty* для *frontAndBack* — поверхностей, перпендикулярным к направлению, исключенному из расчета

```
dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    movingWall
    {
        type      fixedValue;
        value      uniform (1 0 0);
    }

    fixedWalls
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }

    frontAndBack
    {
        type      empty;
    }
}
```



КАВЕРНА: НАСТРОЙКА КОНСТАНТНОГО ОБЕСПЕЧЕНИЯ

Данная задача является несжимаемой и ламинарной, поэтому в ней есть только одна величина, определяющая физические свойства среды — кинематическая вязкость ν $\text{м}^2/\text{с}$

Эта величина задается в файле `constant/transportProperties`:

```
/*-----*-- C++ -*-----*\
|=====|
|  \ \   /   F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
|  \ \   /   O p e r a t i o n | Version: 1.7.1 |
|   \ \   /   A n d | Web: www.OpenFOAM.com |
|    \ \ /   M a n i p u l a t i o n |
\*-----*--/

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       transportProperties;
}
// * * * * *

nu [ 0 2 -1 0 0 0 0 ] 0.01;
```



КАВЕРНА: НАСТРОЙКА ЧИСЛЕННЫХ СХЕМ

Настройка численных схем интегрирования производится в файле **system/fvSchemes**

ГруппаОператора

{

ИмяОператора Схема;

}

ddtSchemes //первая производная по t

```
{
  default       $\frac{\partial \psi}{\partial t}$  Euler;
}
```

gradSchemes //градиент

```
{
  default       $\nabla \psi$  Gauss linear;
  grad(p)      Gauss linear;
}
```

divSchemes //дивергенция

```
{
  default       $\nabla \cdot \psi$  none;
  div(phi,U)   Gauss linear;
}
```

laplacianSchemes //диффузия

```
{
  default      none;  $\nabla \cdot D_\psi \nabla \psi$ 
  laplacian(nu,U) Gauss linear corrected;
  laplacian((1|A(U)),p) Gauss linear corrected;
}
```

interpolationSchemes //интерполяция на грани

```
{
  default      linear;
  interpolate(HbyA) linear;
}
```

snGradSchemes //производная по нормали

```
{
  default      corrected;  $\frac{\partial \psi}{\partial n}$ 
}
```

fluxRequired //поля для которых рассч. поток

```
{
  default      no;
  p            i;
}
```



КАВЕРНА: НАСТРОЙКА МЕТОДОВ РЕШЕНИЯ СЛАУ

В OpenFOAM используется метод расщепления переменных → для каждой искомой переменной (скаляра или тензора) своя система уравнений, свой метод решения.

Настройка методов решения систем линейных алгебраических уравнений производится в файле **system/fvSolution**

Каждому полю — свой метод решения СЛАУ

Для компонент тензора метод решения один, но процедура - последовательная

В отдельном разделе задаются параметры алгоритма связывания полей давления и скорости (PISO)

```
PISO
{
    nCorrectors          2;
    nNonOrthogonalCorrectors 0;
    pRefCell             0;
    pRefValue            0;
}
```

```
solvers
{
    p
    {
        solver          PCG;
        preconditioner   DIC;
        tolerance       1e-06;
        relTol          0;
    }
    U
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance       1e-05;
        relTol          0;
    }
}
```



КАВЕРНА: КОНТРОЛЬ ЗА ХОДОМ РЕШЕНИЯ

Контроль за ходом решения задачи осуществляется в файле ***system/controlDict***

В этом файле задаются:

- начальный момент физического времени;
- конечный момент физического времени;
- шаг по времени;
- шаг вывода результатов;
- формат вывода результатов;
- точность вывода результатов (для ASCII);
- степень сжатия результатов;
- прочие параметры.

```
application      icoFoam;
startFrom        startTime;
startTime        0;
stopAt           endTime;
endTime          0.5;
deltaT           0.005;
writeControl      timeStep;
writeInterval     20;
purgeWrite        0;
writeFormat       ascii;
writePrecision    6;
writeCompression  uncompressed;
timeFormat        general;
timePrecision     6;
runTimeModifiable yes;
```




КАВЕРНА: ЗАПУСК И МОНИТОРИНГ РЕШЕНИЯ

Запуск осуществляется по команде:

```
rm -rf run.log; icoFoam | tee -a run.log
```

После окончания работы программы в файле run.log появится всё содержимое стандартного вывода:

```
tail -n 14 run.log
```

```
Time = 0.5
```

```
Courant Number mean: 0.116925 max: 0.852134
```

```
DILUPBiCG: Solving for Ux, Initial residual = 1.89493e-07, Final residual = 1.89493e-07, No Iterations 0
```

```
DILUPBiCG: Solving for Uy, Initial residual = 4.14522e-07, Final residual = 4.14522e-07, No Iterations 0
```

```
DICPCG: Solving for p, Initial residual = 1.06665e-06, Final residual = 3.39604e-07, No Iterations 1  
time step continuity errors : sum local = 5.25344e-09, global = -9.50761e-19, cumulative = 8.05678e-18
```

```
DICPCG: Solving for p, Initial residual = 5.36118e-07, Final residual = 5.36118e-07, No Iterations 0  
time step continuity errors : sum local = 6.86432e-09, global = 4.62063e-19, cumulative = 8.51884e-18
```

```
ExecutionTime = 0.19 s ClockTime = 0 s
```

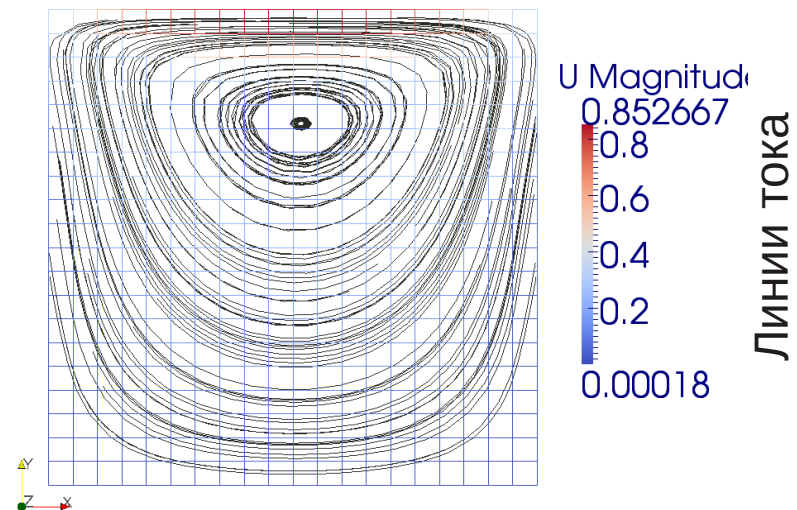
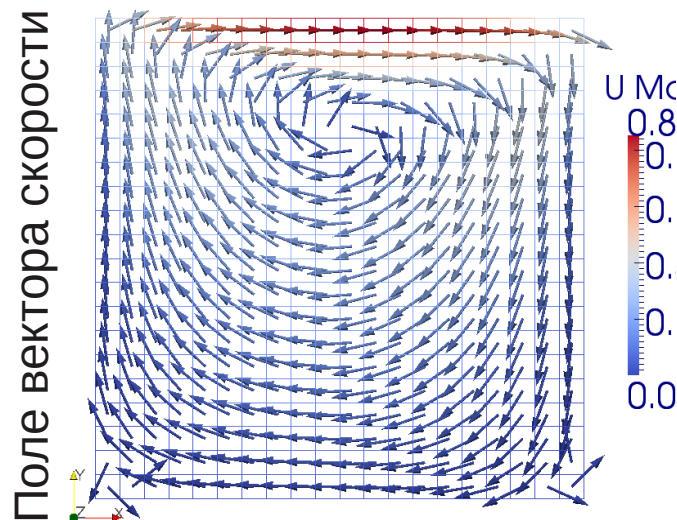
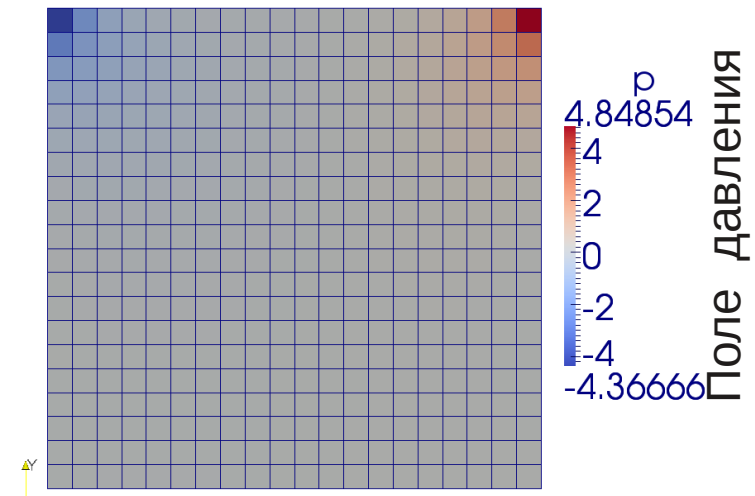
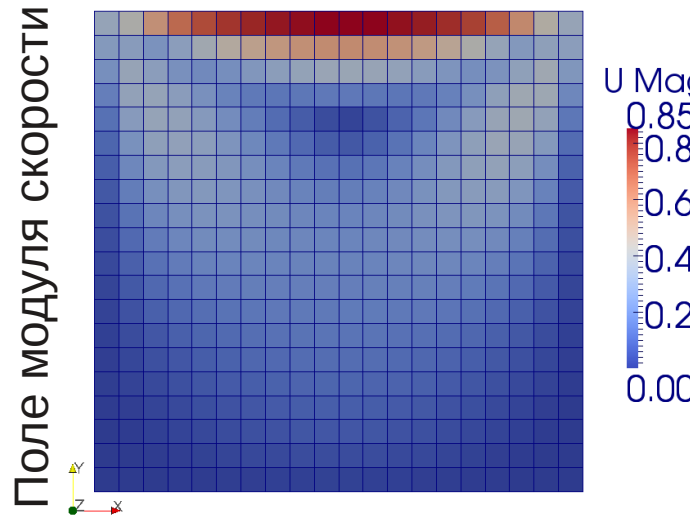
```
End
```

Созданный файл с журналом вывода программы может быть в дальнейшем использована для построения графиков сходимости решения:

```
foamLog -n run.log
```



КАВЕРНА: ВИЗУАЛИЗАЦИЯ В PARAVIEW





КАВЕРНА: ВИЗУАЛИЗАЦИЯ В PARAVIEW: КАК ПОЛУЧЕНО?

После запуска paraFoam переходим к последнему моменту времени

Для построения полей модуля скорости и давления: во вкладке Display выбираем способ расцветки (Color By) — скорость (U) или давление (p), соответственно

Для построения векторного поля: извлекаем центры ячеек (Filters → Alphabetical → Cell Centers); строим векторное поле в центрах ячеек, (Filters → Alphabetical → Glyphs), отключив масштабирование вектора скорости (во вкладке Display для Glyph1 в раскрывающемся меню Scale Mode выбираем off)

Для построения линий тока: выбираем объект, соответствующий каверне и вызываем фильм Stream Tracer (Filters → Alphabetical → Stream Tracer)

