Training-seminar
«Basis of use of
OpenFOAM and ParaView»

DEMONSTRATION:
FLOW IN A CAVITY

Sergei Strijhak (Institute of System Programming of the RAS)

27.06.2016

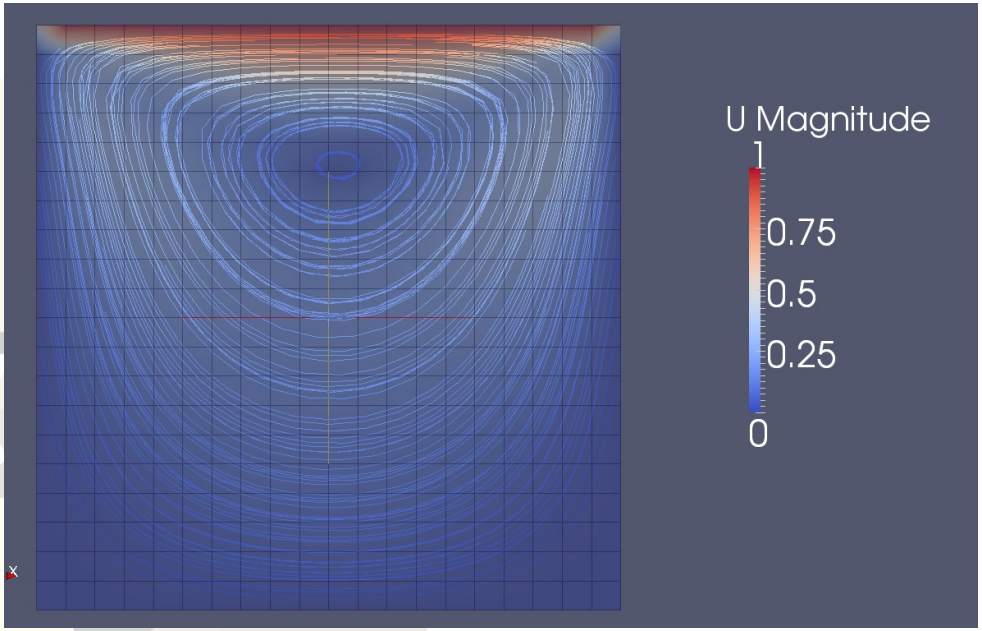
Demonstration of work with the packets OPENFOAM and PARAVIEW in terms of ready examples

Considered issues:

I. Incompressible flow in a cavity
II. Turbulent flow behind a backward step
III. Free convection in a heated chamber

I. Flow in a Cavity

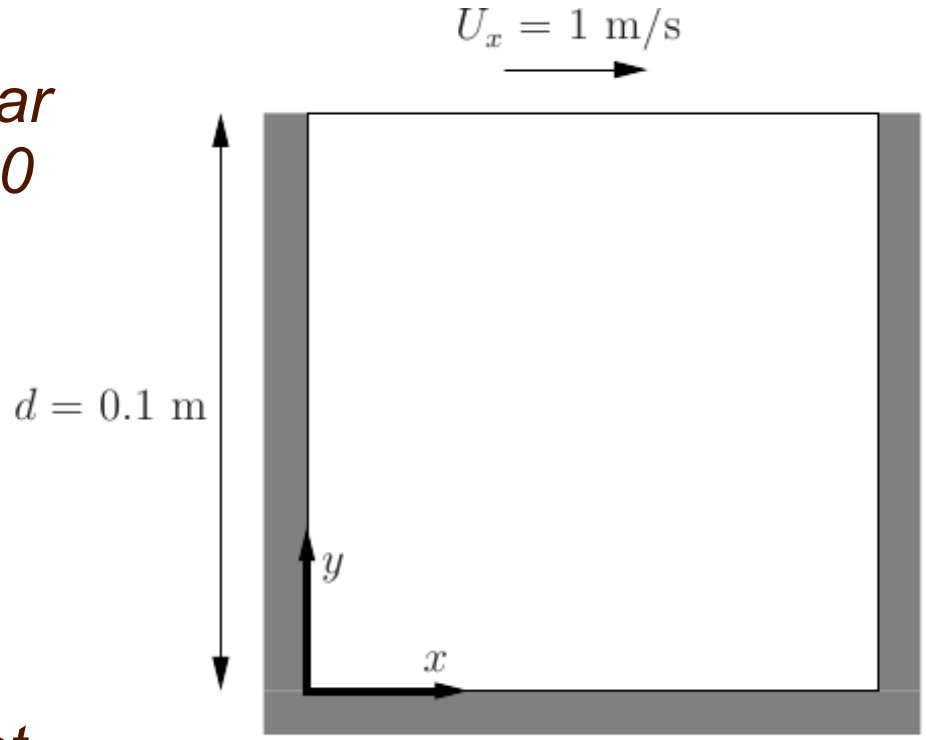
Here and later the next order of operations is supposed:
a) Analysis of physical statement of a problem
b) Choice of mathematical model
c) Mesh settings
d) Settings for the constant environment (of the computational model)
e) Result Analysis and Visualization



Cavity: Brief review

Plane flow of incompressible Newtonian liquid with Reynolds numbers corresponding to the laminar and transient regimes : 100 and 1000 correspondingly.

The dimensioning scheme is presented in the picture on the left.



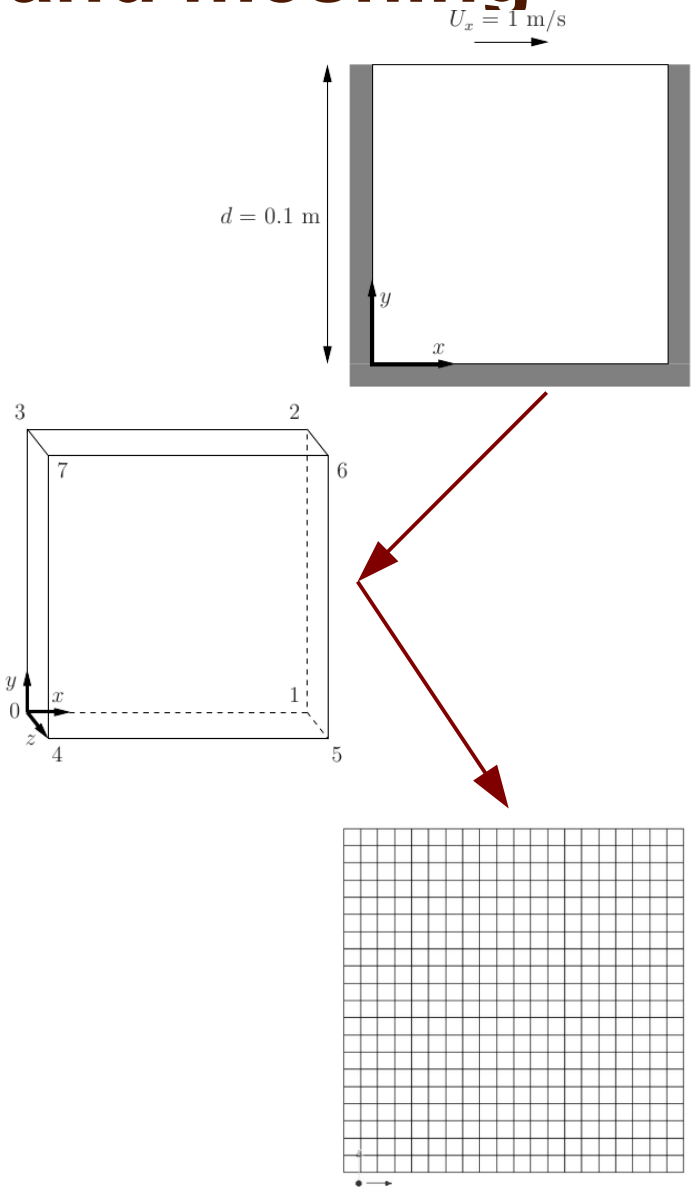
The motion of liquid in the cavity occurs besause of an even displacement of the upper cover that is equivalent to the x-directional constant flow velocity

Cavity: Computitional domain and meshing

The picture from the previous slide corresponds enough to our conception of a computitional domain. Let's trace the mesh on it.

For this we are going to applicate the block mesh generator embedded in OpenFOAM — blockMesh.

For definition of a blockMesh the next information is demanded: node description, block (volume) description and surface description for the boundary condition assignment



CAVITY: LOOK ASIDE: blockMesh (1)

As all the OpenFOAM applications, blockMesh can be run from the command line and as the majority of the applications it doesn't require the arguments, reading all the information from the controlling file

Location:

constant/polyMesh/blockMeshDict

Content:

- *Scale parameter: convertToMeters*
- *List of nodes: vertices*
- *List of blocks: blocks*
- *List of curved edges: edges*
- *List of surfaces: patches*
- *List of merged faces: mergePatchPairs*

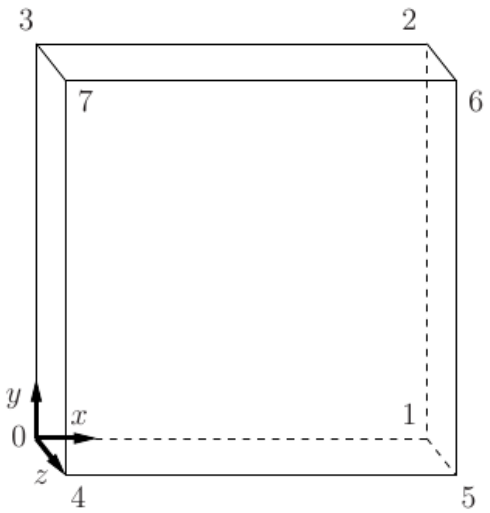
```
16
17  convertToMeters 0.1;
18
19  vertices
20  (
21      (0 0 0)
22      (1 0 0)
23      (1 1 0)
24      (0 1 0)
25      (0 0 0.1)
26      (1 0 0.1)
27      (1 1 0.1)
28      (0 1 0.1)
29  );
30
```

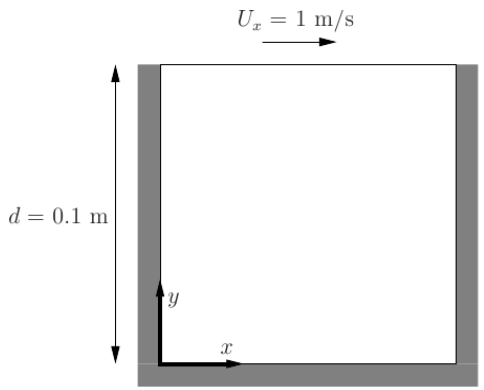
CAVITY: LOOK ASIDE: blockMesh (2)

After point definition we need to form hexahedral blocks — hexahedral volumes with 8 nodes, 12 edges. Each face has strictly 4 edges and 4 nodes. The blocks are defined through a list of nodes in a fixed order (for example, against hour-hand)

The outer surfaces defining the boundary conditions are set in the same manner.

```
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  );
```





CAVITY: MESH CONTROL

It is convenient to check the mesh quality before launching the computation:

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).
```

The output of checkMesh utility includes next issues (set in bold italic type with underlining):

– Overall statistics;

– Statistics of cell types;

– Overall mesh topology;

– Topology of outer boundaries;

– Geometric characteristics (non-orthogonality, skewness, scale range etc.)

CAVITY: MESH CONTROL (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
```

CAVITY: BOUNDARY CONDITIONS (1)

Fields of values — files with the appropriate name

Time cuts (moments) — folders with files, storing the calculated fields

Generally the folder with the title 0 (zero) corresponds to the initial condition

Example with pressure (p):
Each file contains:
– Title(doesn't shown);
– Dimensions;
– Values in the cell centers (internalField);
– Values on the boundary (boundaryField)

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

internalField   uniform 0;

boundaryField
{
    movingWall
    {
        type            zeroGradient;
    }

    fixedWalls
    {
        type            zeroGradient;
    }

    frontAndBack
    {
        type            empty;
    }
}
```

CAVITY: BOUNDARY CONDITIONS (2)

Let's take a look to the velocity field(the pressure field was examined above)

Dimensions — m/s: [0 1 -1 0 0 0 0]

Velocity field in the zero moment of time is unperturbed, that's why the settings are:

internalField uniform (0 0 0)

The slip condition operates on the boundaries movingWall and fixedWalls: uniform (0 0 0); on fixedWalls and movingWalls the settings are: uniform (1 0 0)

Finally, special type of BC — empty — for frontAndBack surfaces, perpendicular to the direction excluded from the computation

```
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;
    }
}
```

CAVITY: SETTINGS FOR THE CONSTANT ENVIRONMENT

The present problem is incompressible and laminar, thus it has only one value determining the physical properties of the medium — kinematic viscosity ν, m^2/s.

This value is chosen in constant/transportProperties:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                      |                                            |
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox      |
|  \\    /   O peration          | Version:  1.7.1                            |
|   \\  /    A nd                | Web:      www.OpenFOAM.com                 |
|    \\/     M anipulation       |                                            |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

CAVITY: SETTINGS FOR NUMERICAL SCHEMES

Settings for the numerical schemes is implemented in ***system/fvSchemes***

```
Operator's group
{
OperatorName Scheme;
}
```

```
ddtSchemes //первая производная по t
{
    default         Euler;
}
gradSchemes //градиент
{
    default         Gauss linear;
    grad(p)         Gauss linear;
}
divSchemes //дивергенция
{
    default         none;
    div(phi,U)      Gauss linear;
}
```

$\frac{\partial \psi}{\partial t}$ $\nabla \psi$ $\nabla \cdot \psi$

```
laplacianSchemes //диффузия
{
    default         none;
    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;
}
fluxRequired //поля для которых рассч. поток
{
    default         no;
    p               ;
}
```

$\nabla \cdot D_\psi \nabla \psi$ $\frac{\partial \psi}{\partial \boldsymbol{n}}$

CAVITY: SETTINGS FOR THE SOLUTION METHODS

In OpenFOAM the method of variable splitting is used → for each sought variable (scalar or tensor) there are its own equation system and its own solution method .

Settings for solution methods for systems of linear algebraic equations is realized in the file ***system/fvSolution***

For each field — its own method to solve the SLE

For the tensor's components the solution method is unique but the procedure is consecutive

Parameters of algorithme of binding together the pressure and velocity fields (PISO) are set in individual section

```
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;
    }
}
```

CAVITY: SOLUTION CONTROL

Control of solution turn is realized in ***system/controlDict***

In this file there are next definitions:

– *Initial moment of the physical time;*
– *Terminal moment of the physical time;*
– *Time step;*
– *Write interval;*
– *Output format;*
– *Output accuracy(for ASCII);*
– *Degree of result's compression ;*
– *Other parameters.*

```
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;
```

CAVITY: RUN & MONITOR THE SOLUTION

The launching can be implemented by the command :

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

After the calculus is accomplished all the standart contents of the output will appear in the file run.log :

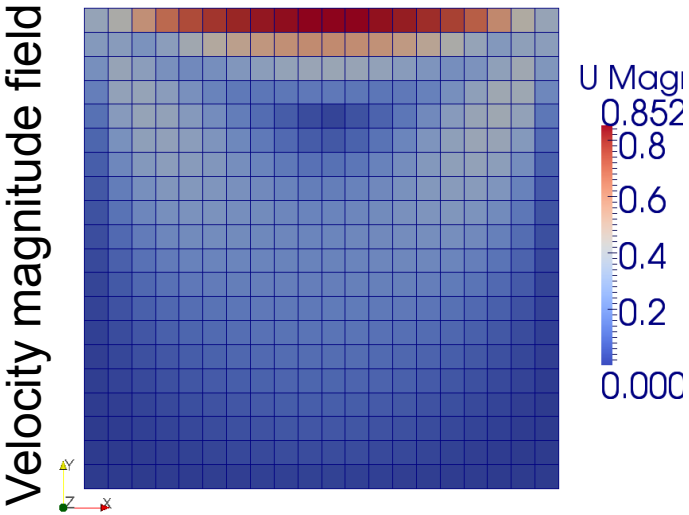
```
tail -n 14 run.log
```

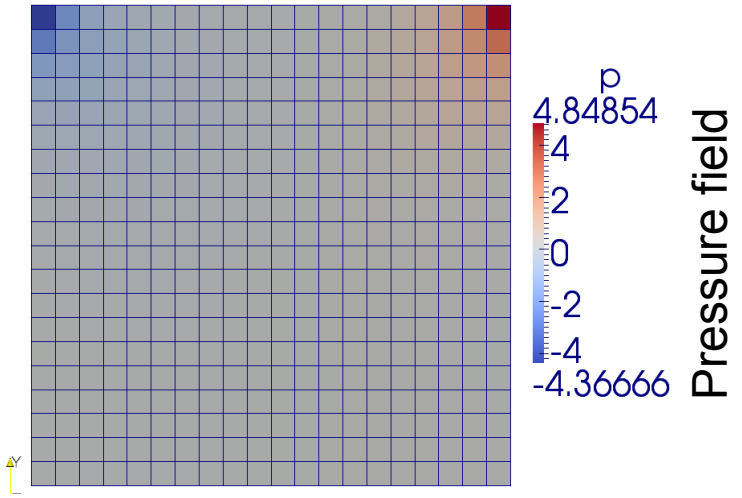
```
Time = 0.5

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
```

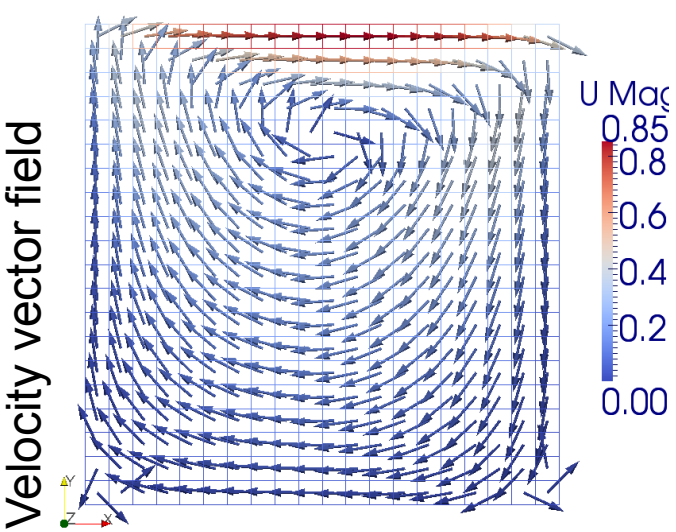
The produced file with the output register can be used later on for plotting of the solution's convergence diagramme:

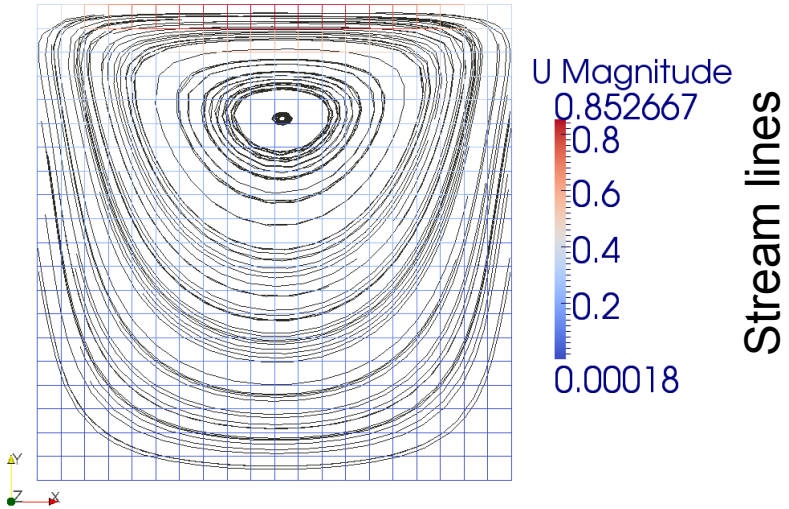
```
foamLog -n run.log
```

CAVITY: VISUALIZATION IN PARAVIEW









CAVITY: VISUALIZATION IN PARAVIEW: HOW TO GET IT?

After execution of paraFoam we choose the last moment of time

For plotting of pressure and velocity magnitude fields : in the inset ***Display*** *choose the manner of coloring (****Color By****) — velocity (U) and pressure (p) correspondingly*

*For construction of the vector field: extract cell centers (****Filters*** *→* ***Alphabetical*** *→* ***Cell Centers****); construct vector field in the cell centers, (****Filters*** *→* ***Alphabetical*** *→* ***Glyphs****) with the velocity vector scaling off (in the inset* ***Display*** *choose Glyph1, in its PopupMenu* ***Scale Mode*** *choose* ***off****)*

For streem lines construction: choose the object, corresponding to the cavity and choose the filter ***Stream Tracer*** *(****Filters*** *→* ***Alphabetical*** *→* ***Stream Tracer****)*