

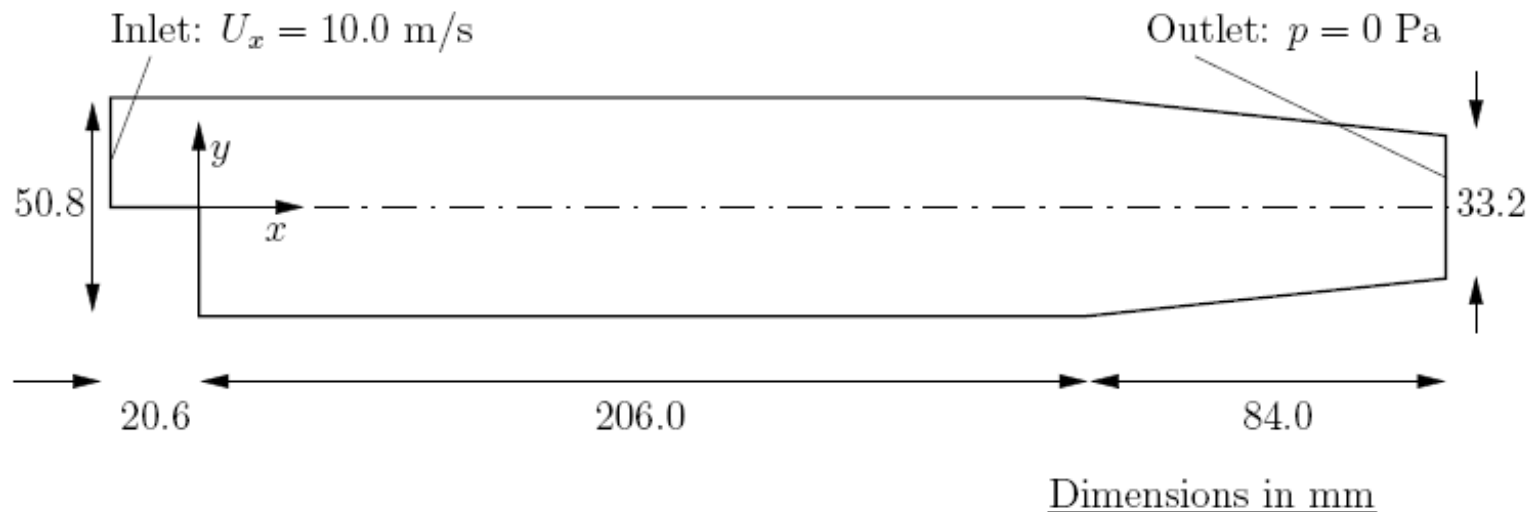


Школа-семинар
«ОСНОВЫ ИСПОЛЬЗОВАНИЯ
OpenFOAM, SALOME и ParaView»
ДЕМОНСТРАЦИЯ: ОБТЕКАНИЕ
ОБРАТНОГО УСТУПА

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

РАСЧЕТ ТЕЧЕНИЯ В КАНАЛЕ (pitzDaily)

РАСЧЕТНАЯ ОБЛАСТЬ И ИСХОДНЫЕ ДАННЫЕ



$U = 10 \text{ м/с}$ на входе, на остальных границах -стенка

URANS + K-е модель , K-omega SST модель

LES (метод крупны вихрей) + K one eddy equation model

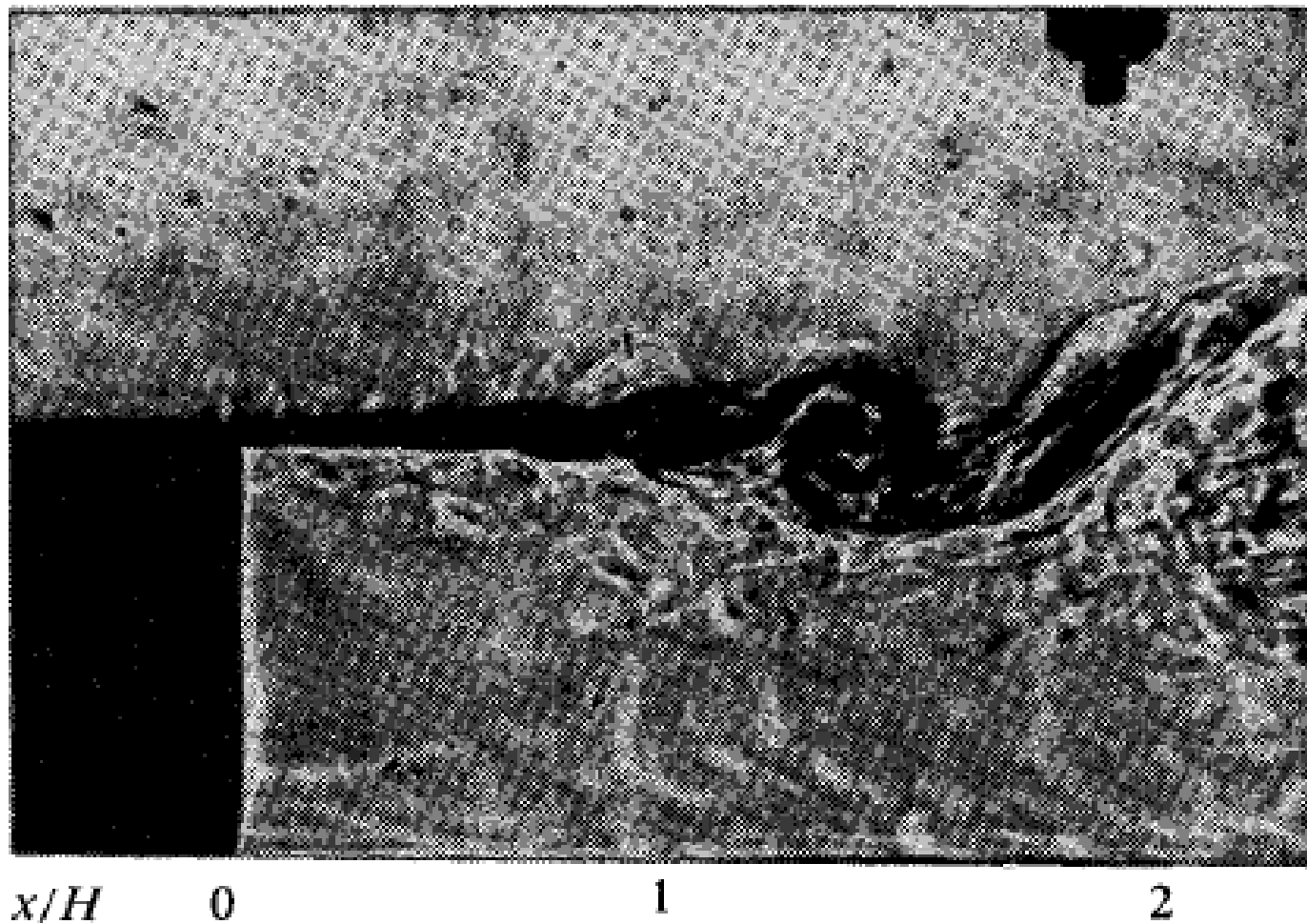
Решатели: simpleFoam , pisoFoam

Использование пристеночных функций в модели турбулентности

Цель: а) стационарный расчет течения в канале (simpleFoam)

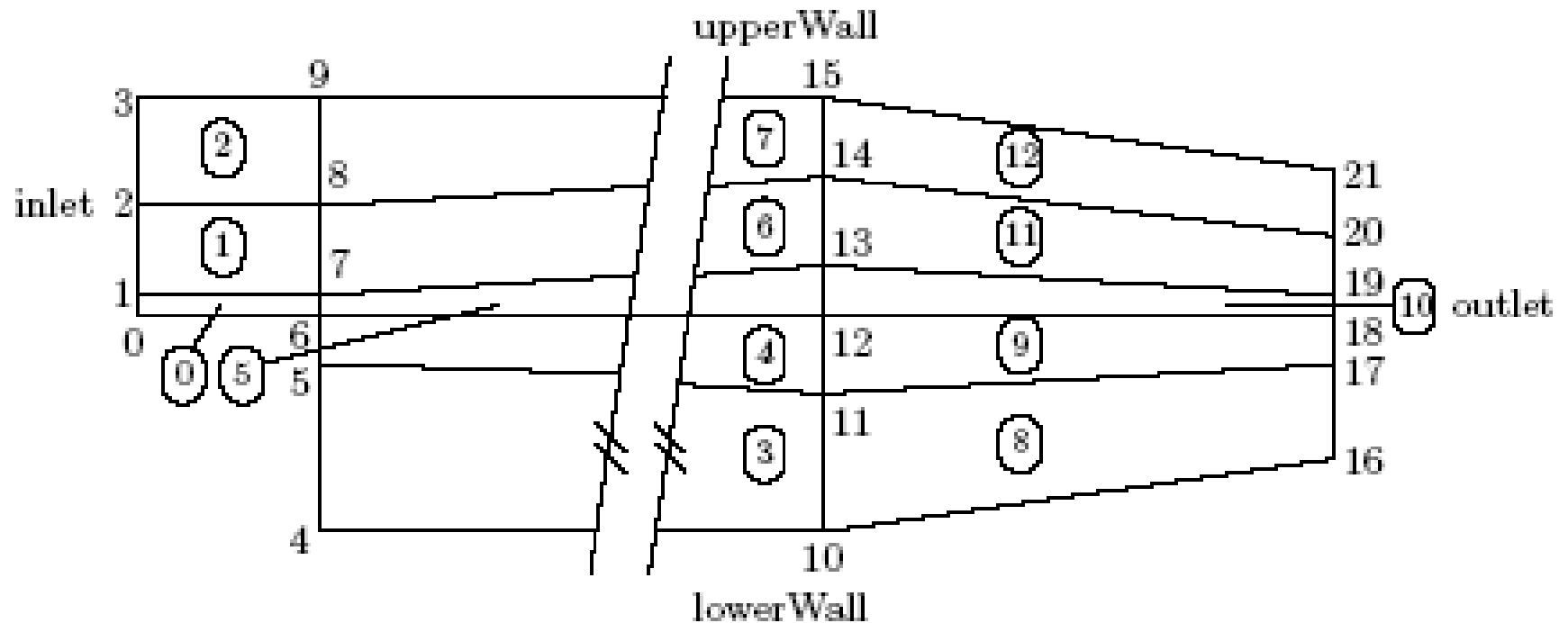
б) нестационарный расчет течения в канале (pisoFoam)

Шлирен фотография Re=22000



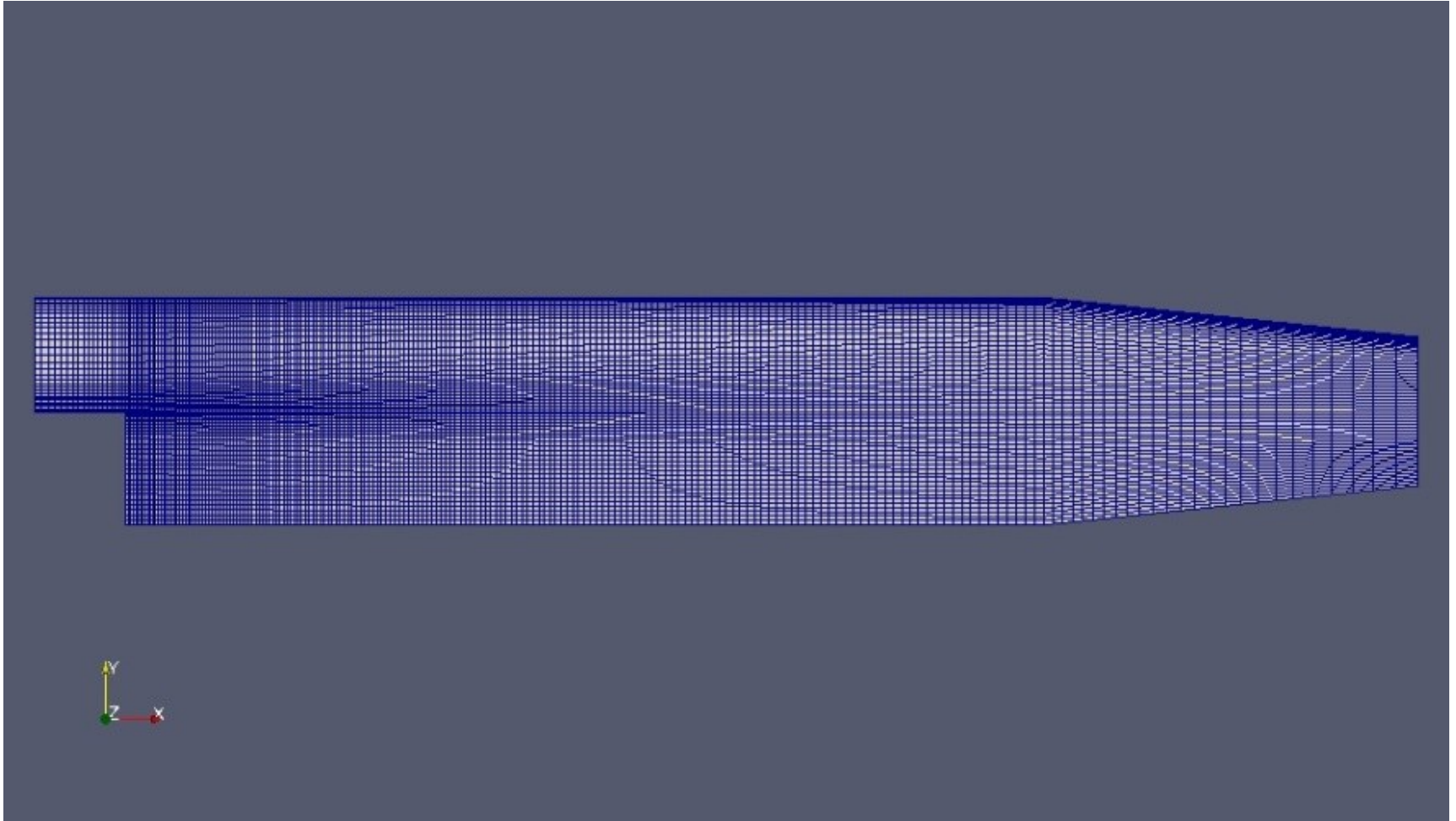
Р.В. Питц, Дж.У. Дейли. Горение в турбулентном слое смешения за уступом.
Аэрокосмическая техника. 1984. N7. с.74-82

Блочная сетка



*Блочная сетка — файл /constant/blockMeshDict
12 блоков*

Расчетная сетка с адаптацией



Двухмерная расчетная сетка содержит 12225 ячеек,
для трехмерного случая - 244500 ячеек (combustion/XiFoam/pitzDaily3D)

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Governing equations

- Mass continuity for incompressible flow

$$\nabla \cdot \mathbf{U} = 0 \quad (3.4)$$

- Steady flow momentum equation

$$\nabla \cdot (\mathbf{U}\mathbf{U}) + \nabla \cdot \mathbf{R} = -\nabla p \quad (3.5)$$

where p is kinematic pressure and (in slightly over-simplistic terms) $\mathbf{R} = \nu_{eff} \nabla \mathbf{U}$ is the viscous stress term with an effective kinematic viscosity ν_{eff} , calculated from selected transport and turbulence models.

Initial conditions $U = 0$ m/s, $p = 0$ Pa — required in OpenFOAM input files but not necessary for the solution since the problem is steady-state.

Boundary conditions

- Inlet (left) with fixed velocity $\mathbf{U} = (10, 0, 0)$ m/s;
- Outlet (right) with fixed pressure $p = 0$ Pa;
- No-slip walls on other boundaries.

Transport properties

- Kinematic viscosity of air $\nu = \mu/\rho = 18.1 \times 10^{-6}/1.293 = 14.0 \text{ } \mu\text{m}^2/\text{s}$

Turbulence model

- Standard $k - \epsilon$;
- Coefficients: $C_\mu = 0.09$; $C_1 = 1.44$; $C_2 = 1.92$; $\alpha_k = 1$; $\alpha_\epsilon = 0.76923$.

Solver name simpleFoam: an implementation for steady incompressible flow.

Математическая модель расчета параметров течения (URANS – Unsteady Reynolds Averaged Navier-Stokes)

Обобщенное уравнение, отражающее законы сохранения и модель турбулентности, в интегральной форме

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \phi \, d\Omega + \int_{\Omega} \rho \phi \mathbf{V} \cdot \mathbf{n} \, dS = \int_S \Gamma \operatorname{grad} \phi \cdot \mathbf{n} \, dS + \int_{\Omega} q_{\phi} \, d\Omega \quad (1)$$

Здесь ϕ - обобщенная переменная $\phi = \{1, u, v, w, k, \omega, h\}$ Ω - контрольный объем,

\mathbf{V} - вектор скорости, Γ - коэффициенты переноса, \mathbf{n} - вектор нормали

dS - Дифференциальный элемент площади

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_i k)}{\partial x_i} = \beta^* \rho k \omega + \frac{\partial}{\partial x_i} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i} \right] \quad \text{Модель k-omega SST Ментера}$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_i \omega)}{\partial x_i} = \alpha \rho S^2 - \beta \rho \omega^2 + \frac{\partial}{\partial x_i} \left[(\mu + \sigma_{\omega} \mu_t) \frac{\partial \omega}{\partial x_i} \right] + 2(1 - F_1) \rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad (2)$$

$$\frac{\partial(\rho \nu)}{\partial t} + \frac{\partial(\rho \nu_j)}{\partial x_j} = \frac{1}{\sigma_{\nu}} \left\{ \frac{\partial}{\partial x_j} \left[(\mu + \rho \nu) \frac{\partial \nu}{\partial x_j} \right] + C_{b2} \rho \left(\frac{\partial \nu}{\partial x_j} \right)^2 \right\} + G_{\nu} - Y_{\nu}; \quad \text{Модель Спаларта-Аллмараса} \quad (3)$$

$\rho, u, p, k, \omega, h, t, \mu$ - плотность, скорость, давление, кинетическая энергия турбулентности, скорость диссипации энергии, энтальпия, время, динамическая вязкость

Постановка задачи: Задание граничных и начальных условий, выбор расчетных схем

Математическая модель расчета параметров течения
(LES – Large Eddy Simulation)

$$u = \bar{u} + u' \quad \bar{u} = \int_{D_1} G(\xi, \zeta, \eta, \Delta) u(\xi, \zeta, \eta, t) d\xi d\zeta d\eta \quad (4)$$

$$u' \text{ - подсеточный масштаб скорости} \quad \Delta = V^{1/3} = (\Delta x \Delta y \Delta z)^{1/3} \quad (5)$$

$$\partial_t \bar{u} + \nabla \times (\bar{u} \otimes \bar{u}) = \nabla \times (\bar{S} - B) \quad S = -pI + 2\nu D \quad (6)$$

$$D = 0.5(\nabla u + \nabla u^T) \quad B = L + C + R$$

I - единичный тензор L - напряжения Леонарда

C - перекрестные члены R - подсеточные напряжения Рейнольдса

Дифференциальное уравнение для подсеточной кинетической энергии:

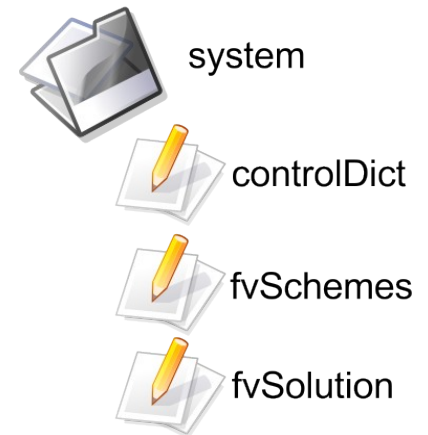
$$\frac{\partial k_{sgs}}{\partial t} + \nabla \times (k_{sgs} \bar{u}) = \nabla \left[(\nu + \nu_{SGS}) \times \nabla k_{sgs} \right] + 2\nu_{sgs} |\bar{S}|^2 - 0.916 \frac{k_{sgs}^{3/2}}{\Delta} \quad (7)$$

$$\nu_{sgs} = 0.067 k_{sgs}^{0.5} \Delta \quad \bar{S} \text{ - тензор подсеточной скорости деформации} \quad (8)$$

ИСХОДНЫЕ ДАННЫЕ для примера pitzDaily.

КАТАЛОГ ЗАДАЧИ

```
[cfd1@master simpleFoam]$ cd pitzDailyParallel/  
[cfd1@master pitzDailyParallel]$ ll  
total 12  
drwxr-xr-x 2 cfd1 sm3 4096 Dec 22 16:43 0  
drwxr-xr-x 3 cfd1 sm3 4096 Dec 22 16:43 constant  
drwxr-xr-x 2 cfd1 sm3 4096 Dec 22 16:48 system  
[cfd1@master pitzDailyParallel]$  
[cfd1@master pitzDailyParallel]$ cd system/  
[cfd1@master system]$ ll  
total 16  
-rw-r----- 1 cfd1 sm3 1222 Dec 22 16:43 controlDict  
-rw-r----- 1 cfd1 sm3 1206 Dec 22 16:48 decomposeParDict  
-rw-r----- 1 cfd1 sm3 1877 Dec 22 16:43 fvSchemes  
-rw-r----- 1 cfd1 sm3 1940 Dec 22 16:43 fvSolution
```



ЗНАЧЕНИЯ ДЛЯ СКОРОСТИ 0/U

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

```
internalField  uniform (0 0 0);
```

```
boundaryField
```

```
{  inlet      {      type      fixedValue;      value      uniform (10 0 0);  }
```

```
outlet        {      type      zeroGradient;  }
```

```
upperWall     {      type      fixedValue;      value      uniform (0 0 0);  }
```

```
lowerWall     {      type      fixedValue;      value      uniform (0 0 0);  }
```

```
frontAndBack  {      type      empty;  }}//
```

```
***** //
```

ЗНАЧЕНИЯ ДЛЯ ДАВЛЕНИЯ

O/P

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

internalField uniform 0;

boundaryField

{

inlet

{ type zeroGradient; }

outlet

{ type fixedValue;
value uniform 0; }

upperWall

{ type zeroGradient; }

lowerWall

{ type zeroGradient; }

frontAndBack

{ type empty;}

}

ЗНАЧЕНИЯ ДЛЯ КИНЕТИЧЕСКОЙ ЭНЕРГИИ ТУРБУЛЕНТНОСТИ $0/k$

Использование пристеночной функции kqRWallFunction

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

```
internalField  uniform 0.375;
```

```
boundaryField
```

```
{  
inlet  {      type      fixedValue;      value      uniform 0.375;  }  
outlet {      type      zeroGradient;    }  
upperWall {    type      kqRWallFunction;  value      uniform 0.375;  }  
lowerWall {    type      kqRWallFunction;  value      uniform 0.375;  }  
frontAndBack {    type      empty;  
}  
}  
// ***** //
```

ЗНАЧЕНИЯ ДЛЯ ЭНЕРГИИ ДИССИПАЦИИ $0/\epsilon$

Использование пристеночной функции epsilonWallFunction

```
dimensions    [0 2 -3 0 0 0 0];
internalField uniform 14.855;
boundaryField
{
    inlet
    {
        type      fixedValue;
        value      uniform 14.855; }
    outlet
    {
        type      zeroGradient;}
    upperWall
    {
        type      epsilonWallFunction;
        value      uniform 14.855; }
    lowerWall
    {
        type      epsilonWallFunction;
        value      uniform 14.855; }
    frontAndBack
    {
        type      empty;}
}
```

ЗНАЧЕНИЯ для тензора Рейнольдсовых напряжений 0/R

Использование пристеночной функции kqRWallFunction

```
dimensions    [0 2 -2 0 0 0 0];
internalField uniform (0 0 0 0 0 0);
boundaryField
{
    inlet
    {
        type      fixedValue;
        value      uniform (0 0 0 0 0 0); }

    outlet
    {
        type      zeroGradient; }

    upperWall
    {
        type      kqRWallFunction;
        value      uniform ( 0 0 0 0 0 0 ); }

    lowerWall
    {
        type      kqRWallFunction;
        value      uniform ( 0 0 0 0 0 0 ); }

    frontAndBack
    {
        type      empty; }
}
```

ЗНАЧЕНИЯ ДЛЯ ТУРБУЛЕНТНОЙ ВЯЗКОСТИ 0/nut

Использование пристеночной функции nutWallFunction

```
dimensions    [0 2 -1 0 0 0 0];
internalField  uniform 0;
boundaryField
{
    inlet
    {
        type      calculated;
        value      uniform 0;}
    outlet
    {
        type      calculated;
        value      uniform 0;}
    upperWall
    {
        type      nutWallFunction;
        value      uniform 0;}
    lowerWall
    {
        type      nutWallFunction;
        value      uniform 0;}
    frontAndBack
    {
        type      empty;}
}
```

УПРАВЛЕНИЕ РАСЧЕТОМ Файл /system/controlDict

```
application    simpleFoam;  
startFrom      startTime;  
startTime      0;  
stopAt         endTime;  
endTime        10;  
deltaT         1;  
writeControl    timeStep;  
writeInterval   1;  
purgeWrite      0;  
writeFormat     ascii;  
writePrecision  6;  
writeCompression uncompressed;  
timeFormat      general;  
timePrecision   6;  
runTimeModifiable yes;
```


СХЕМЫ ДИСКРЕТИЗАЦИИ /system/fvSchemes

```
gradSchemes{ default Gauss linear;
grad(p) Gauss linear;
grad(U) Gauss linear;}
divSchemes{ default none;
div(phi,U) Gauss GammaV 1.0;
div(phi,k) Gauss Gamma 1.0;
div(phi,epsilon) Gauss Gamma 1.0;
div(phi,omega) Gauss Gamma 1.0;
div(phi,R) Gauss Gamma 1.0;
div(R) Gauss linear;
div(phi,nuTilda) Gauss upwind;
div((nuEff*dev(grad(U).T())) Gauss linear;}
laplacianSchemes{ default none;
laplacian(nuEff,U) Gauss linear corrected;
laplacian((1/A(U)),p) Gauss linear corrected;
laplacian(DkEff,k) Gauss linear corrected;
laplacian(DepsilonEff,epsilon) Gauss linear corrected;
laplacian(DomegaEff,omega) Gauss linear corrected;
laplacian(DREff,R) Gauss linear corrected;
laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;}
interpolationSchemes{ default linear; interpolate(U) linear;}
snGradSchemes{ default corrected;}
fluxRequired{ default no; p;}
```

МЕТОДЫ РЕШЕНИЯ СЛАУ Файл

/system/fvSolutions

FoamFile

```
{ version 2.0; format ascii; class dictionary; object fvSolution;}  
// **** //
```

Solvers {

```
p PCG { preconditioner DIC; tolerance 1e-06; relTol 0.01; };  
U PBiCG { preconditioner DILU; tolerance 1e-05; relTol 0.1; };  
k PBiCG { preconditioner DILU; tolerance 1e-05; relTol 0.1; };  
epsilon PBiCG { preconditioner DILU; tolerance 1e-05; relTol 0.1; };  
omega PBiCG { preconditioner DILU; tolerance 1e-05; relTol 0.1; };  
R PBiCG { preconditioner DILU; tolerance 1e-05; relTol 0.1; };  
nuTilda PBiCG { preconditioner DILU; tolerance 1e-05; relTol 0.1; };
```

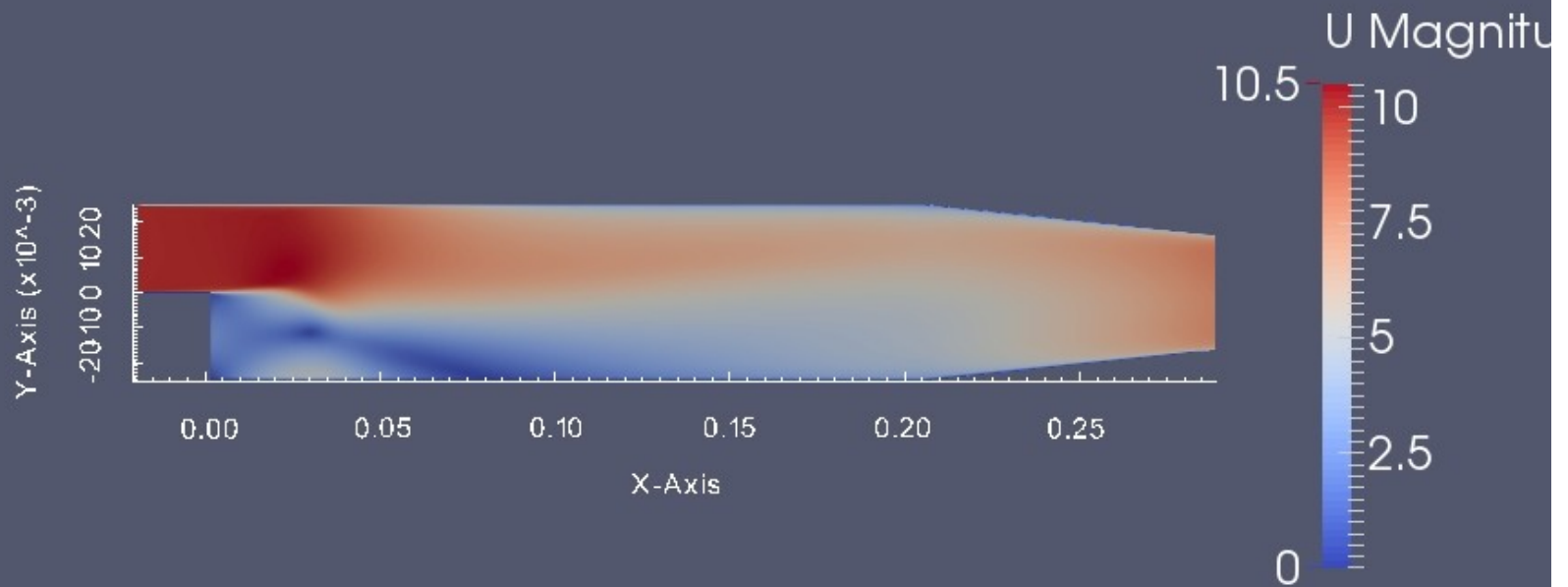
SIMPLE

```
{ nNonOrthogonalCorrectors 0;  
relaxationFactors  
{ p 0.3; U 0.7; k 0.7; epsilon 0.7; omega 0.7; R  
0.7; nuTilda 0.7;}
```

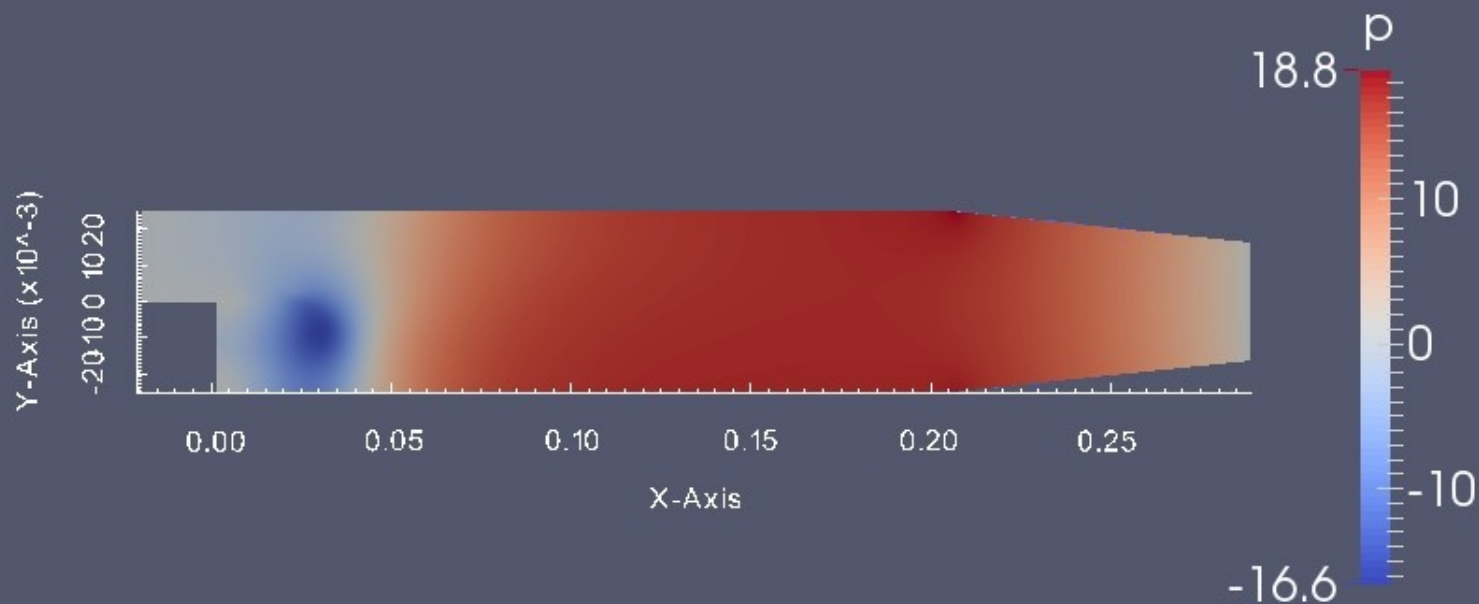
PISO

```
{ nCorrectors 4; nNonOrthogonalCorrectors 0; pRefCell 0; pRefValue 0;}  
// **** //
```

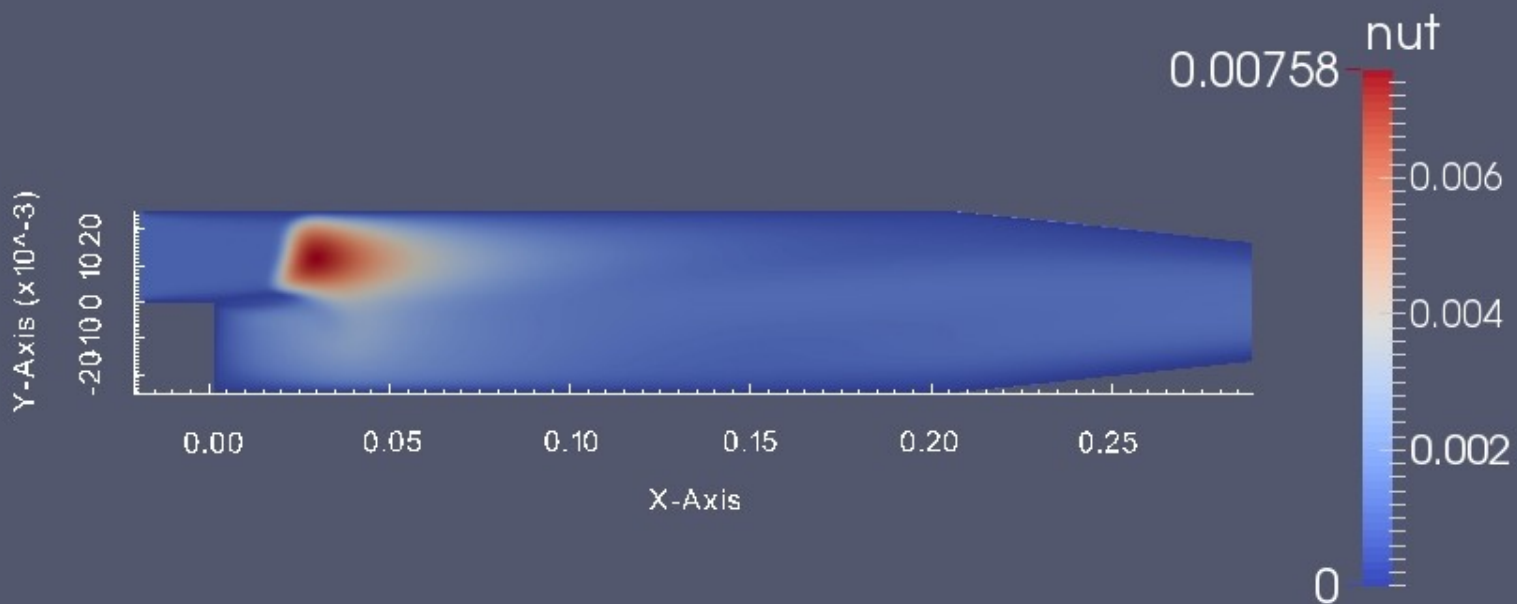
Результаты расчета UMagnitude в simpleFoam. T=100 с.



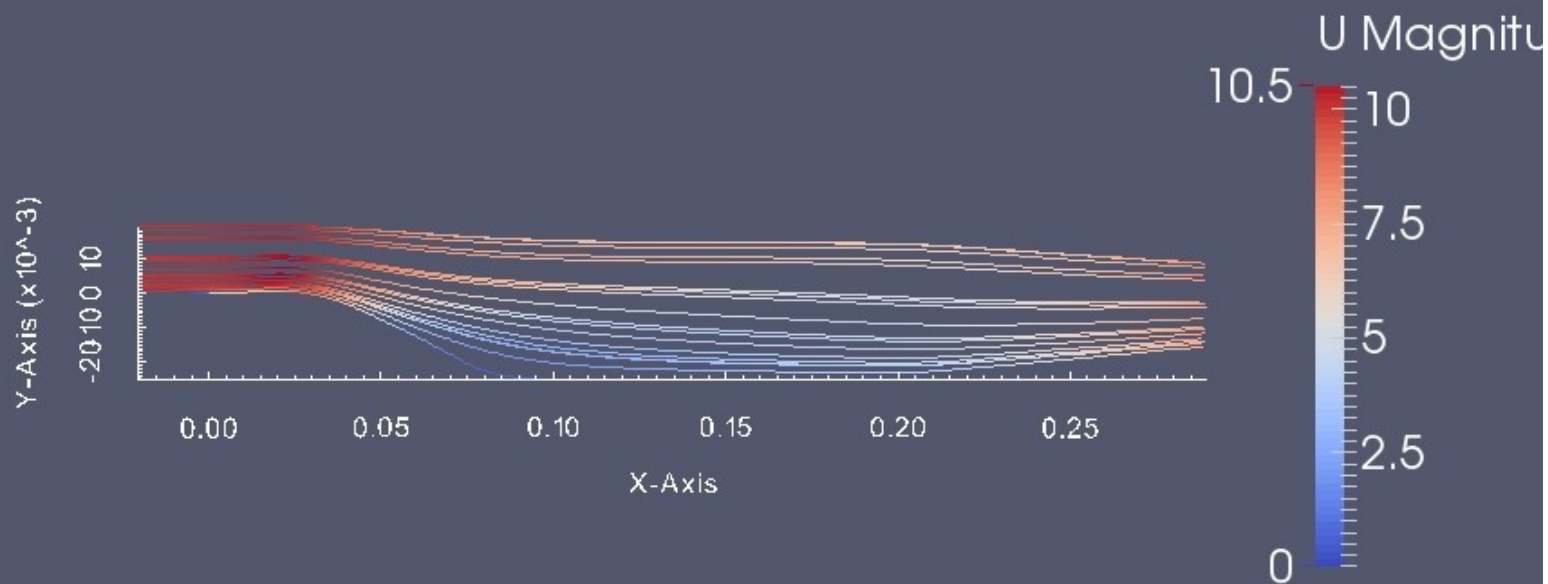
Результаты расчета p в simpleFoam. $T=100$ с.



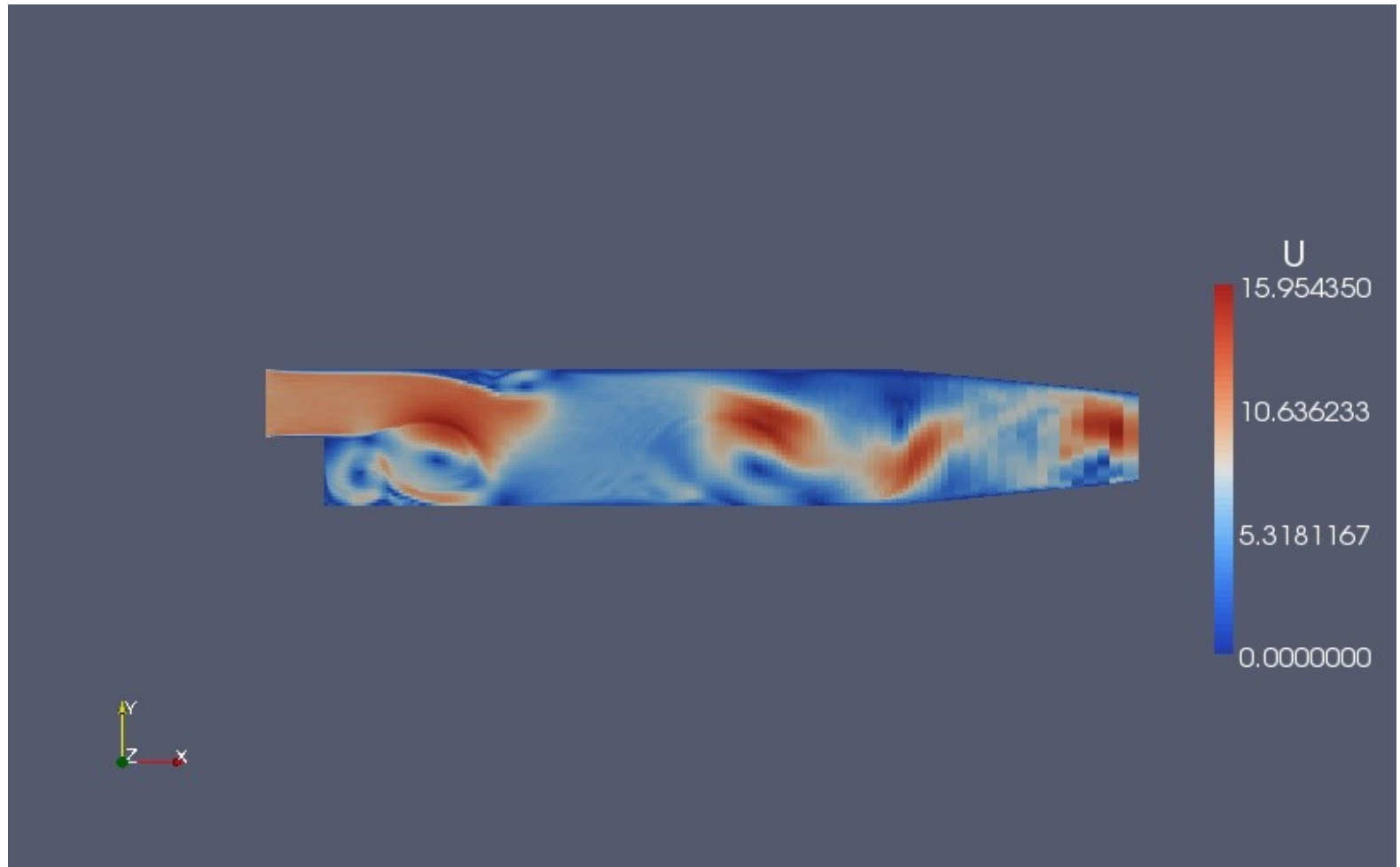
Результаты расчета ν_{t} в simpleFoam. $T=100$ с.



Результаты расчета линий тока в simpleFoam. T=100 с.



Результаты расчета скорости в pisoFoam



LES model. One eddy equation.

Общий алгоритм для любого расчетного кейса в OpenFOAM

\$ blockMesh – подготовка расчетной сетки

\$ checkMesh – проверка качества расчетной сетки

\$ simpleFoam (\$pisoFoam) – запуск решателя

\$ foamToVTK - трансляция результатов в формат VTK

\$ touch 1.foam – создание файла с расширением 'foam'

\$ simpleFoam > log & - создание log файла

\$ foamLog log – запуск скрипта, использующего команды Linux grep, awk, sed для извлечения данных из log файла.

\$ yPlusRAS – определение значения yPlus для URANS модели

\$ gnuplot – построение графиков

Запуск Paraview и загрузка файлов с расширением vtk для обработки результатов расчета.