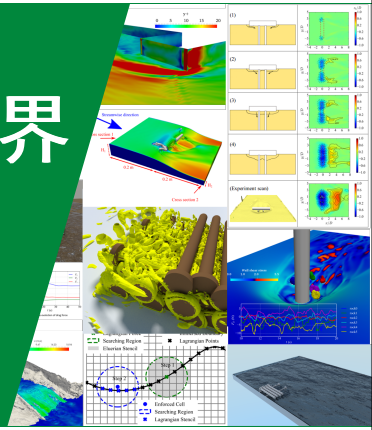
中国农业大学　流体机械与流体工程系

2024年春季《计算流体动力学编程实践》

2.4 OpenFOAM®边界条件



徐云成

✉ycxu@cau.edu.cn

2024 年 3 月 21 日

回顾FVM中的边界条件

数值问题中常见的边界条件

- Dirichlet B.C.(第一类边界条件): fixed boundary value of ϕ
  $\Rightarrow$ `fixedValue`
- Neumann B.C.(第二类边界条件): zero gradient or no flux condition:$\mathbf{n} \cdot \overrightarrow{q_s} = 0$
  $\Rightarrow$ `zeroGradient`

OpenFOAM®中的边界条件

constant/polyMesh/boundary

```
    lowerWall
    {
        type            wall;
        inGroups        List<word> 1(wall);
        nFaces          62;
        startFace       4532;
    }
    atmosphere
    {
        type            patch;
        nFaces          46;
        startFace       4594;
    }
```

wallFvPatch和fvPatch在
src/finiteVolume/fvMesh/fvPatches

0/U

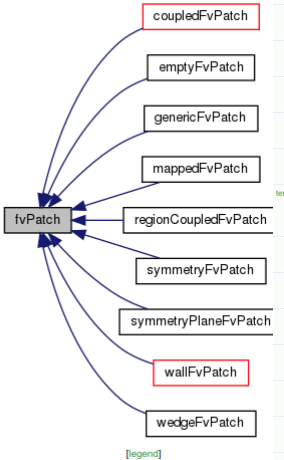
```
boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (1.3394 0 0);
    }
    outlet
    {
        type            zeroGradient;
    }
```

fixedValueFvPatchField和
zeroGradientFvPatchField在
src/finiteVolume/fields/fvPatchFields

fvPatch 相关

src/finiteVolume/fvMesh/fvPatches

```
yux116@XuLinux:fvPatches$ tree -L 2
.
├── basic
│   ├── coupled
│   └── generic
├── constraint
│   ├── cyclic
│   ├── cyclicACMI
│   ├── cyclicAMI
│   ├── cyclicRepeatAMI
│   ├── cyclicSlip
│   ├── empty
│   ├── processor
│   ├── processorCyclic
│   ├── symmetry
│   ├── symmetryPlane
│   └── wedge
├── derived
│   ├── mapped
│   ├── regionCoupled
│   └── wall
└── fvPatch
    ├── fvPatch.C
    ├── fvPatchFvMeshTemplates.C
    ├── fvPatch.H
    ├── fvPatchList.H
    ├── fvPatchNew.C
    └── fvPatchTemplates.C
```

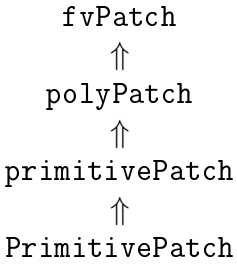


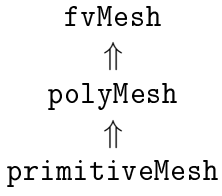
virtual const **word** &	**name** () const Return name. More...
virtual **label**	**start** () const Return start label of this patch in the **polyMesh** face list. More...
virtual **label**	**size** () const Return size. More...
virtual bool	**coupled** () const Return true if this patch is coupled. More...
label	**index** () const Return the index of this patch in the **fvBoundaryMesh**. More...
const **fvBoundaryMesh** &	**boundaryMesh** () const Return **boundaryMesh** reference. More...
template<class T >	
const **List**< T >::subList	**patchSlice** (const **List**< T > &l) const Slice list to patch. More...
virtual const **labelUList** &	**faceCells** () const Return faceCells. More...
const **vectorField** &	**Cf** () const Return face centres. More...
tmp< **vectorField** >	**Cn** () const Return neighbour cell centres. More...
const **vectorField** &	**Sf** () const Return face area vectors. More...
const **scalarField** &	**magSf** () const Return face area magnitudes. More...
tmp< **vectorField** >	**nf** () const Return face normals. More...
virtual **tmp**< **vectorField** >	**delta** () const Return cell-centre to face-centre vector. More...

fvPatch 源头

`fvPatch`
(fv ⇒ finite volume)
⇑
`polyPatch`
(poly ⇒ polygon 多边形)
⇑
`primitivePatch`
(primitive 原始的)
⇑
`PrimitivePatch`
(A list of faces which address into the list of points)

fvPatch vs fvMesh





fvPatchFields

$FOAM_SRC/finiteVolume/fields/fvPatchFields/basic/

```
yux116@XuLinux:basic$ ls
basicSymmetry  directionMixed          fixedValue  transform
calculated     extrapolatedCalculated  mixed       zeroGradient
coupled        fixedGradient           sliced
yux116@XuLinux:basic$ pwd
/home/yux116/OpenFOAM/OpenFOAM-8/src/finiteVolume/fields/fvPatchFields/basic
```

衍生出来的新的BC：

$FOAM_SRC/finiteVolume/fields/fvPatchFields/basic/

总共有83个，包括noSlip、inletOutlet等

形状约束性条件的BC：

$FOAM_SRC/finiteVolume/fields/fvPatchFields/constraint/

包括cyclic、symmetry、processor等

fvPatchFields

$FOAM_SRC/finiteVolume/fields/fvPatchFields/basic/

```
yux116@XuLinux:basic$ ls
basicSymmetry  directionMixed          fixedValue  transform
calculated     extrapolatedCalculated  mixed       zeroGradient
coupled        fixedGradient           sliced
yux116@XuLinux:basic$ pwd
/home/yux116/OpenFOAM/OpenFOAM-8/src/finiteVolume/fields/fvPatchFields/basic
```

衍生出来的新的BC：

$FOAM_SRC/finiteVolume/fields/fvPatchFields/basic/

总共有83个，包括noSlip、inletOutlet等

形状约束性条件的BC：

$FOAM_SRC/finiteVolume/fields/fvPatchFields/constraint/

包括cyclic、symmetry、processor等

fvPatchFields

$FOAM_SRC/finiteVolume/fields/fvPatchFields/basic/

```
yux116@XuLinux:basic$ ls
basicSymmetry  directionMixed          fixedValue  transform
calculated     extrapolatedCalculated  mixed       zeroGradient
coupled        fixedGradient           sliced
yux116@XuLinux:basic$ pwd
/home/yux116/OpenFOAM/OpenFOAM-8/src/finiteVolume/fields/fvPatchFields/basic
```

衍生出来的新的BC：

$FOAM_SRC/finiteVolume/fields/fvPatchFields/basic/

总共有83个，包括noSlip、inletOutlet等

形状约束性条件的BC：

$FOAM_SRC/finiteVolume/fields/fvPatchFields/constraint/

包括cyclic、symmetry、processor等

basicSymmetry

symmetry和symmetryPlane的母类，不能直接用

coupled

耦合（coupled）边界的母类，例如cyclic和并行中生成的processor，不能直接用

calculated

只是被动的进行计算，不需要在系数矩阵中进行额外的边界条件设置，经常用于nut

extrapolatedCalculated

基于calculated，如果需要在系数矩阵中进行边界条件定值(evaluation)，那么相当于zeroGradient，不过snGrad输出的是边界网格中心值和边界值形成的梯度，

OpenFOAM®中的基础BC

basicSymmetry

symmetry和symmetryPlane的母类，不能直接用

coupled

耦合（coupled）边界的母类，例如cyclic和并行中生成的processor，不能直接用

calculated

只是被动的进行计算，不需要在系数矩阵中进行额外的边界条件设置，经常用于nut

extrapolatedCalculated

基于calculated，如果需要在系数矩阵中进行边界条件定值(evaluation)，那么相当于zeroGradient，不过snGrad输出的是边界网格中心值和边界值形成的梯度，

OpenFOAM®中的基础BC

basicSymmetry

symmetry和symmetryPlane的母类，不能直接用

coupled

耦合（coupled）边界的母类，例如cyclic和并行中生成的processor，不能直接用

calculated

只是被动的进行计算，不需要在系数矩阵中进行额外的边界条件设置，经常用于nut

extrapolatedCalculated

基于calculated，如果需要在系数矩阵中进行边界条件定值(evaluation)，那么相当于zeroGradient，不过snGrad输出的是边界网格中心值和边界值形成的梯度，

OpenFOAM®中的基础BC

basicSymmetry

symmetry和symmetryPlane的母类，不能直接用

coupled

耦合（coupled）边界的母类，例如cyclic和并行中生成的processor，不能直接用

calculated

只是被动的进行计算，不需要在系数矩阵中进行额外的边界条件设置，经常用于nut

extrapolatedCalculated

基于calculated，如果需要在系数矩阵中进行边界条件定值(evaluation)，那么相当于zeroGradient，不过snGrad输出的是边界网格中心值和边界值形成的梯度，

OpenFOAM®中的基础BC

fixedValue

对边界值进行固定约束，也是需要其他边界条件的母类，通常也叫第一类边界条件，Dirichlet BC

fixedGradient

对边界法向梯度进行固定约束，也叫第二类边界条件，Neumann BC

zeroGradient

法向梯度固定为零的约束

OpenFOAM®中的基础BC

fixedValue

对边界值进行固定约束，也是需要其他边界条件的母类，通常也叫第一类边界条件，Dirichlet BC

fixedGradient

对边界法向梯度进行固定约束，也叫第二类边界条件，Neumann BC

zeroGradient

法向梯度固定为零的约束

OpenFOAM®中的基础BC

fixedValue

对边界值进行固定约束，也是需要其他边界条件的母类，通常也叫第一类边界条件，Dirichlet BC

fixedGradient

对边界法向梯度进行固定约束，也叫第二类边界条件，Neumann BC

zeroGradient

法向梯度固定为零的约束

OpenFOAM®中的基础BC

directionMixed

混合类边界条件的母类，一般不直接用。定义的时候需要一个valueFraction来混合法向(zeroGraident)和切向(fixedValue)的边界条件

mixed

混合类边界条件，现在也不常用，一般用衍生出来的`inletOutlet`

sliced

就是一个模板，不常用，不需要了解

transform

就是一个模板，不常用，不需要了解

形状约束的BC

src/finiteVolume/fields/fvPatchFields/constraint/

cyclic	cyclicRepeatAMI	jumpCyclic	processorCyclic	wedge
cyclicACMI	cyclicSlip	jumpCyclicAMI	symmetry	
cyclicAMI	empty	processor	symmetryPlane	

都有对应的fvPatch:
src/finiteVolume/fvMesh/fvPatches/constraint/

和polyPatch:
src/OpenFOAM/meshes/polyMesh/polyPatches/constraint/

注意：这一类边界条件通常在constant/boundary中也要求进行设置

形状约束的BC

src/finiteVolume/fields/fvPatchFields/constraint/

cyclic	cyclicRepeatAMI	jumpCyclic	processorCyclic	wedge
cyclicACMI	cyclicSlip	jumpCyclicAMI	symmetry	
cyclicAMI	empty	processor	symmetryPlane	

都有对应的fvPatch:
src/finiteVolume/fvMesh/fvPatches/constraint/

和polyPatch:
src/OpenFOAM/meshes/polyMesh/polyPatches/constraint/

注意：这一类边界条件通常在constant/boundary中也要求进行设置

形状约束的BC

empty

用于降维，法向方向不进行计算

symmetry

基本等同于`slip`，边界法向梯度为零，切向值等于边界网格中心的值

symmetryPlane

与`symmetry`基本一样，要求边界是平面

形状约束的BC

empty

用于降维，法向方向不进行计算

symmetry

基本等同于`slip`，边界法向梯度为零，切向值等于边界网格中心的值

symmetryPlane

与`symmetry`基本一样，要求边界是平面

形状约束的BC

empty

用于降维，法向方向不进行计算

symmetry

基本等同于`slip`，边界法向梯度为零，切向值等于边界网格中心的值

symmetryPlane

与`symmetry`基本一样，要求边界是平面

形状约束的BC

cyclic

循环边界，需要在constant/boundary中进行设置

```
sides1
{
    type            cyclic;
    neighbourPatch  sides2
    faces           ((2 4 5 3));
}
sides2
{
    type            cyclic;
    neighbourPatch  sides1
    faces           ((8 9 11 10));
}
```

注意：要求两个边界几何上一致

形状约束的BC

cyclicAMI

AMI=arbitrary mesh interface
基于cyclic的循环边界，但不要求两个边界的网格形状完全一致，可进行插值，计算速度稍慢

cyclicACMI

ACMI=arbitrary coupled mesh interface
除了neighbourPatch外，还需要设定nonOverlapPatch,具体可见
tutorials/incompressible/pimpleFoam/RAS/oscillatingInletACMI2D

cyclicRepeatAMI

除了neighbourPatch外，还需要设定transformPatch,具体可见
tutorials/incompressible/pimpleFoam/RAS/impeller

形状约束的BC

cyclicSlip

基本等同于`cyclic`

jumpCyclic

是jump系列边界条件的母类，指的是两个边界的值具有指定的差

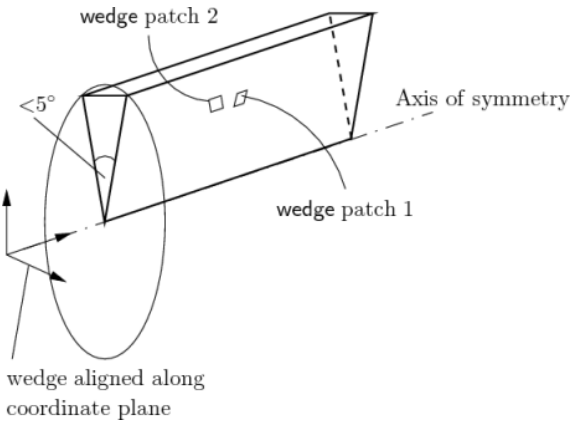
jumpCyclicAMI

同上

形状约束的BC

wedge

与cyclic相似，但是用于二维计算，轴对称问题



1. 计算边界平面的平均法向$\mathbf{n} = (n_1, n_2, n_3)$
2. 检查是否是平面：检查每一个boundary face是否与平均方向相同
3. 得到一个新的法向$\mathbf{n_c} = (\max(n_1, 0.5) - 0.5, \max(n_2, 0.5) - 0.5, \max(n_3, 0.5) - 0.5)$，再作标准化处理即$\mathbf{n_c} = \mathbf{n_c}/|\mathbf{n_c}|$
4. 检查$\mathbf{n_c}$三个分量之和是否大于1
5. 轴方向 $\mathbf{n} \times \mathbf{n_c}$

形状约束的BC

processor

基于coupled边界，一般是由decomposePar自动生成，用于并行计算中不同processor之间的信息交换

processorCyclic

用于并行计算中涉及不同processor之间循环边界的信息交换，一般是由decomposePar自动生成

衍生的BC

src/finiteVolume/fields/fvPatchFields/derived/

activeBaffleVelocity
activePressureForceBaffleVelocity
advective
codedFixedValue
codedMixed
cylindricalInletVelocity
dynamicPressure
entrainmentPressure
externalCoupledMixed
fanPressure
fanPressureJump
fixedFluxExtrapolatedPressure
fixedFluxPressure
fixedInternalValue
fixedJump
fixedJumpAMI
fixedMean
fixedMeanOutletInlet
fixedNormalInletOutletVelocity
fixedNormalSlip
fixedPressureCompressibleDensity
fixedProfile
flowRateInletVelocity
flowRateOutletVelocity
fluxCorrectedVelocity
freestream
freestreamPressure
freestreamVelocity
inletOutlet
inletOutletTotalTemperature
interfaceCompression
interstitialInletVelocity
mappedField
mappedFixedInternalValue
mappedFixedValue
mappedFlowRate
mappedVelocityFluxFixedValue
matchedFlowRateOutletVelocity
movingWallVelocity
noSlip
outletInlet
outletMappedUniformInlet
outletPhaseMeanVelocity
partialSlip
phaseHydrostaticPressure
plenumPressure
pressure
pressureDirectedInletOutletVelocity
pressureDirectedInletVelocity
pressureInletOutletParSlipVelocity
pressureInletOutletVelocity
pressureInletUniformVelocity
pressureInletVelocity
pressureNormalInletOutletVelocity
PrghPressure
prghTotalHydrostaticPressure
rotatingPressureInletOutletVelocity
rotatingTotalPressure
rotatingWallVelocity
slip
supersonicFreestream
surfaceNormalFixedValue
surfaceNormalUniformFixedValue
swirlFlowRateInletVelocity
swirlInletVelocity
syringePressure
timeVaryingMappedFixedValue
totalPressure
totalTemperature
translatingWallVelocity
turbulentInlet
turbulentIntensityKineticEnergyInlet
uniformDensityHydrostaticPressure
uniformFixedGradient
uniformFixedValue
uniformInletOutlet
uniformJump
uniformJumpAMI
uniformTotalPressure
variableHeightFlowRate
variableHeightFlowRateInletVelocity
waveSurfacePressure
waveTransmissive

衍生的BC

codedFixedValue

```
inlet
{
    type            codedFixedValue;
    value           uniform (1 0 0);
    name            swirl;
    code            #{
        const vector axis(1, 0, 0);

        vectorField v(2.0*this->patch().Cf() ^ axis);
        v.replace(vector::X, 1.0);
        operator==(v);
    #};
}
```

https://cpp.openfoam.org/v9/

OpenFOAM运算

Operation	Comment	Mathematical Description	Description in OpenFOAM
Addition		$\mathbf{a} + \mathbf{b}$	`a + b`
Subtraction		$\mathbf{a} - \mathbf{b}$	`a - b`
Scalar multiplication		$s\mathbf{a}$	`s * a`
Scalar division		$\mathbf{a}/s$	`a / s`
Outer product	rank $\mathbf{a}, \mathbf{b} >= 1$	$\mathbf{ab}$	`a * b`
Inner product	rank $\mathbf{a}, \mathbf{b} >= 1$	$\mathbf{a} \cdot \mathbf{b}$	`a & b`
Double inner product	rank $\mathbf{a}, \mathbf{b} >= 2$	$\mathbf{a} : \mathbf{b}$	`a && b`
Cross product	rank $\mathbf{a}, \mathbf{b} = 1$	$\mathbf{a} \times \mathbf{b}$	`a ^ b`
Square		$\mathbf{a}^2$	`sqr(a)`
Magnitude squared		$\lvert\mathbf{a}\rvert^2$	`magSqr(a)`
Magnitude		$\lvert\mathbf{a}\rvert$	`mag(a)`
Power	$n = 0, 1, ..., 4$	$\mathbf{a}^n$	`pow(a,n)`
Component average	$i = 1, ..., N$	$\overline{a_i}$	`cmptAv(a)`
Component maximum	$i = 1, ..., N$	$\max(a_i)$	`max(a)`
Component minimum	$i = 1, ..., N$	$\min(a_i)$	`min(a)`
Scale		$\text{scale}(\mathbf{a},\mathbf{b})$	`scale(a,b)`
Geometric transformation	transforms $\mathbf{a}$ using tensor $\mathbf{T}$		`transform(T,a)`

衍生的BC

codedFixedValue

随时间变化

```
code
#{
    operator==(min(10, 0.1*this->db().time().value()));
#};
```

衍生的BC

codedMixed

```
<patchName>
{
    type            codedMixed;
    refValue        uniform (0 0 0);
    refGradient     uniform (0 0 0);
    valueFraction   uniform 1;
    name    rampedMixed;   // name of generated BC
    code
    #{
        this->refValue() =
            vector(1, 0, 0)
           *min(10, 0.1*this->db().time().value());
        this->refGrad() = Zero;
        this->valueFraction() = 1.0;
    #};
}
```

衍生的BC

fixedFluxPressure

这是一个压力的边界条件，根据速度的边界条件来确定，主要通过pEqn中的

```
// Update the pressure BCs to ensure flux consistency
constrainPressure(p_rgh, U, phiHbyA, rAUf, MRF);
```

其中具体的计算可见
src/finiteVolume/cfdTools/general/constrainPressure/constrainPressure.C

$$\nabla p = \left(\mathbf{H}/a_P \cdot \mathbf{S}_f - \mathbf{u} \cdot \mathbf{S}_f\right) \frac{a_P}{|\mathbf{S}_f|}$$

tutorials 里主要是用于多相流计算，但理论上只要有constrainPressure，就可以使用fixedFluxPressure。

fixedFluxPressure使泊松方程更好收敛

衍生的BC

fixedFluxPressure

这是一个压力的边界条件，根据速度的边界条件来确定，主要通过pEqn中的

```
// Update the pressure BCs to ensure flux consistency
constrainPressure(p_rgh, U, phiHbyA, rAUf, MRF);
```

其中具体的计算可见
src/finiteVolume/cfdTools/general/constrainPressure/constrainPressure.C

$$\nabla p = \left(\mathbf{H}/a_P \cdot \mathbf{S}_f - \mathbf{u} \cdot \mathbf{S}_f\right) \frac{a_P}{|\mathbf{S}_f|}$$

tutorials 里主要是用于多相流计算，但理论上只要有constrainPressure，就可以使用fixedFluxPressure。

fixedFluxPressure使泊松方程更好收敛

衍生的BC

fixedFluxPressure

这是一个压力的边界条件，根据速度的边界条件来确定，主要通过pEqn中的

```
// Update the pressure BCs to ensure flux consistency
constrainPressure(p_rgh, U, phiHbyA, rAUf, MRF);
```

其中具体的计算可见
src/finiteVolume/cfdTools/general/constrainPressure/constrainPressure.C

$$\nabla p = \left(\mathbf{H}/a_P \cdot \mathbf{S}_f - \mathbf{u} \cdot \mathbf{S}_f\right) \frac{a_P}{|\mathbf{S}_f|}$$

tutorials 里主要是用于多相流计算，但理论上只要有constrainPressure，就可以使用fixedFluxPressure。

fixedFluxPressure使泊松方程更好收敛

衍生的BC

fixedMean

设定一个平均值，用近边界值的空间分布来进行调整

```
<patchName>
{
    type            fixedMean;
    meanValue       1.0;
}
```

1. 计算边界网格中心的平均值

$$\overline{\phi} = \frac{\sum \phi_i |\mathbf{S}_f|_i}{\sum |\mathbf{S}_f|_i}$$

2. 对于指定的平均值$\overline{\phi}_0$，

$$\phi_p = \begin{cases} \phi_i \dfrac{\overline{\phi}_0}{\overline{\phi}} & \dfrac{\overline{\phi}}{\overline{\phi}_0} > 0.5 \\ \phi_i + (\overline{\phi}_0 - \overline{\phi}) & \dfrac{\overline{\phi}}{\overline{\phi}_0} \leqslant 0.5 \end{cases} \tag{1}$$

衍生的BC

fixedMean

设定一个平均值，用近边界值的空间分布来进行调整

```
<patchName>
{
    type            fixedMean;
    meanValue       1.0;
}
```

1. 计算边界网格中心的平均值

$$\overline{\phi} = \frac{\sum \phi_i |\mathbf{S}_f|_i}{\sum |\mathbf{S}_f|_i}$$

2. 对于指定的平均值$\overline{\phi}_0$，

$$\phi_p = \begin{cases} \phi_i \dfrac{\overline{\phi}_0}{\overline{\phi}} & \dfrac{\overline{\phi}}{\overline{\phi}_0} > 0.5 \\ \phi_i + (\overline{\phi}_0 - \overline{\phi}) & \dfrac{\overline{\phi}}{\overline{\phi}_0} \leqslant 0.5 \end{cases} \tag{1}$$

衍生的BC

flowRateInletVelocity

与fixedMean相似，不过设定的是质量流量/体积流量

```
<patchName>
{
    type                flowRateInletVelocity;
    volumetricFlowRate  0.2;// m3/s
    extrapolateProfile  yes;
    value               uniform (0 0 0);
}
{
    type                flowRateInletVelocity;
    massFlowRate        0.2;// kg/s
    extrapolateProfile  yes;
    rho                 rho;
    rhoInlet            1.0;// kg/m3
    value               uniform (0 0 0);
}
```

衍生的BC

flowRateInletVelocity

与fixedMean相似，不过设定的是质量流量/体积流量

```
<patchName>
{
    type                flowRateInletVelocity;
    volumetricFlowRate  0.2;// m3/s
    extrapolateProfile  yes;
    value               uniform (0 0 0);
}
{
    type                flowRateInletVelocity;
    massFlowRate        0.2;// kg/s
    extrapolateProfile  yes;
    rho                 rho;
    rhoInlet            1.0;// kg/m3
    value               uniform (0 0 0);
}
```

1. 利用边界网格中心速度计算法向速度$u_n = \mathbf{u} \cdot \mathbf{n}_f$
2. 估计流量$Q_{esti} = \sum u_{n,i} |\mathbf{S}_f|_i$
3. 对于指定的流量Q_0，

$$u_{n,p} = \begin{cases} u_n \dfrac{Q_0}{Q_{esti}} & \dfrac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \dfrac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \dfrac{Q_{esti}}{Q_0} \leqslant 0.5 \end{cases} \tag{2}$$

$$\mathbf{u}_p = \mathbf{u} - u_n \mathbf{n}_f - u_{n,\text{更新}} \mathbf{n}_f$$

4. 如果extrapolateProfile是no，边界速度均匀分布

衍生的BC

flowRateInletVelocity

与fixedMean相似，不过设定的是质量流量/体积流量

```
<patchName>
{
    type                flowRateInletVelocity;
    volumetricFlowRate  0.2;// m3/s
    extrapolateProfile  yes;
    value               uniform (0 0 0);
}
{
    type                flowRateInletVelocity;
    massFlowRate        0.2;// kg/s
    extrapolateProfile  yes;
    rho                 rho;
    rhoInlet            1.0;// kg/m3
    value               uniform (0 0 0);
}
```

1. 利用边界网格中心速度计算法向速度$u_n = \mathbf{u} \cdot \mathbf{n}_f$
2. 估计流量$Q_{esti} = \sum u_{n,i} |\mathbf{S}_f|_i$
3. 对于指定的流量Q_0，

$$u_{n,p} = \begin{cases} u_n \dfrac{Q_0}{Q_{esti}} & \dfrac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \dfrac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \dfrac{Q_{esti}}{Q_0} \leqslant 0.5 \end{cases} \tag{2}$$

$$\mathbf{u}_p = \mathbf{u} - u_n \mathbf{n}_f - u_{n,\text{更新}} \mathbf{n}_f$$

4. 如果extrapolateProfile是no，边界速度均匀分布

衍生的BC

flowRateInletVelocity

与fixedMean相似，不过设定的是质量流量/体积流量

```
<patchName>
{
    type                flowRateInletVelocity;
    volumetricFlowRate  0.2;// m3/s
    extrapolateProfile  yes;
    value               uniform (0 0 0);
}
{
    type                flowRateInletVelocity;
    massFlowRate        0.2;// kg/s
    extrapolateProfile  yes;
    rho                 rho;
    rhoInlet            1.0;// kg/m3
    value               uniform (0 0 0);
}
```

1. 利用边界网格中心速度计算法向速度$u_n = \mathbf{u} \cdot \mathbf{n}_f$
2. 估计流量$Q_{esti} = \sum u_{n,i} |\mathbf{S}_f|_i$
3. 对于指定的流量Q_0，

$$u_{n,p} = \begin{cases} u_n \dfrac{Q_0}{Q_{esti}} & \dfrac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \dfrac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \dfrac{Q_{esti}}{Q_0} \leqslant 0.5 \end{cases} \tag{2}$$

$$\mathbf{u}_p = \mathbf{u} - u_n \mathbf{n}_f - u_{n,\text{更新}} \mathbf{n}_f$$

4. 如果extrapolateProfile是no，边界速度均匀分布

衍生的BC

flowRateInletVelocity

与fixedMean相似，不过设定的是质量流量/体积流量

```
<patchName>
{
    type                flowRateInletVelocity;
    volumetricFlowRate  0.2;// m3/s
    extrapolateProfile  yes;
    value               uniform (0 0 0);
}
{
    type                flowRateInletVelocity;
    massFlowRate        0.2;// kg/s
    extrapolateProfile  yes;
    rho                 rho;
    rhoInlet            1.0;// kg/m3
    value               uniform (0 0 0);
}
```

1. 利用边界网格中心速度计算法向速度$u_n = \mathbf{u} \cdot \mathbf{n}_f$
2. 估计流量$Q_{esti} = \sum u_{n,i} |\mathbf{S}_f|_i$
3. 对于指定的流量Q_0，

$$u_{n,p} = \begin{cases} u_n \dfrac{Q_0}{Q_{esti}} & \dfrac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \dfrac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \dfrac{Q_{esti}}{Q_0} \leqslant 0.5 \end{cases} \tag{2}$$

$$\mathbf{u}_p = \mathbf{u} - u_n \mathbf{n}_f - u_{n,\text{更新}} \mathbf{n}_f$$

4. 如果extrapolateProfile是no，边界速度均匀分布

衍生的BC

flowRateInletVelocity

与fixedMean相似，不过设定的是质量流量/体积流量

```
<patchName>
{
    type                flowRateInletVelocity;
    volumetricFlowRate  0.2;// m3/s
    extrapolateProfile  yes;
    value               uniform (0 0 0);
}
{
    type                flowRateInletVelocity;
    massFlowRate        0.2;// kg/s
    extrapolateProfile  yes;
    rho                 rho;
    rhoInlet            1.0;// kg/m3
    value               uniform (0 0 0);
}
```

1. 利用边界网格中心速度计算法向速度$u_n = \mathbf{u} \cdot \mathbf{n}_f$
2. 估计流量$Q_{esti} = \sum u_{n,i} |\mathbf{S}_f|_i$
3. 对于指定的流量Q_0，

$$u_{n,p} = \begin{cases} u_n \dfrac{Q_0}{Q_{esti}} & \dfrac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \dfrac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \dfrac{Q_{esti}}{Q_0} \leqslant 0.5 \end{cases} \tag{2}$$

$$\mathbf{u}_p = \mathbf{u} - u_n \mathbf{n}_f - u_{n,\text{更新}} \mathbf{n}_f$$

4. 如果extrapolateProfile是no，边界速度均匀分布

衍生的BC

inletOutlet

是一个出流边界，对回流有特定的设置

- 正通量（出流）：zeroGradient
- 负通量（入流）：fixedValue（inletValue）

```
<patchName>
{
    type            inletOutlet;
    phi             phi;
    inletValue      uniform 0;
    value           uniform 0;
}
```

outletInlet

是一个入流边界，对出流有特定的设置

- 正通量（出流）：fixedValue（outletValue）
- 负通量（入流）：zeroGradient

```
<patchName>
{
    type            inletOutlet;
    phi             phi;
    outletValue     uniform 0;
    value           uniform 0;
}
```

衍生的BC

inletOutlet

是一个出流边界，对回流有特定的设置

- 正通量（出流）：zeroGradient
- 负通量（入流）：fixedValue（inletValue）

```
<patchName>
{
    type            inletOutlet;
    phi             phi;
    inletValue      uniform 0;
    value           uniform 0;
}
```

outletInlet

是一个入流边界，对出流有特定的设置

- 正通量（出流）：fixedValue（outletValue）
- 负通量（入流）：zeroGradient

```
<patchName>
{
    type            inletOutlet;
    phi             phi;
    outletValue     uniform 0;
    value           uniform 0;
}
```

衍生的BC

freestream

与inletOutlet相似，根据通量正负来确定是zeroGradient还是fixedValue

```
<patchName>
{
    type            freestream;
    freestreamValue uniform (300 0 0);
}
```


衍生的BC

freestreamVelocity

```
<patchName>
{
    type            freestreamVelocity;
    freestreamValue uniform (300 0 0);
}
```

$$\text{valueFraction}_i = \begin{cases} 0.5 - 0.5\dfrac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$$

freestreamPressure

```
<patchName>
{
    type            freestreamPressure;
    freestreamValue uniform (300 0 0);
    supersonic      false;
}
```

if (supersonic_=true):

$$\text{valueFraction}_i = \begin{cases} 0.5 + 0.5\dfrac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$$

else:

$$\text{valueFraction}_i = \begin{cases} 0.5 - 0.5\dfrac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$$

衍生的BC

freestreamVelocity

```
<patchName>
{
    type            freestreamVelocity;
    freestreamValue uniform (300 0 0);
}
```

$$\text{valueFraction}_i = \begin{cases} 0.5 - 0.5\dfrac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$$

freestreamPressure

```
<patchName>
{
    type            freestreamPressure;
    freestreamValue uniform (300 0 0);
    supersonic      false;
}
```

if (supersonic_=true):

$$\text{valueFraction}_i = \begin{cases} 0.5 + 0.5\dfrac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$$

else:

$$\text{valueFraction}_i = \begin{cases} 0.5 - 0.5\dfrac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$$

衍生的BC

supersonicFreestream

用于超音速问题，用于U的BC

```
<patchName>
{
    type      supersonicFreestream;
    UInf      500;//velocity
    pInf      1e4;//pressure
    TInf      265;//Temperature
    gamma     1.4;//Heat capacity ratio
}
```

src/finiteVolume/fields/fvPatchFields/derived/supersonicFreestream/

衍生的BC

totalPressure

```
<patchName>
{
    type      totalPressure;
    p0        uniform 1e5;
}
```

这个边界条件有四种变化：

- incompressible subsonic(不可压、亚音速) $p_p = p_0 - 0.5|\mathbf{u}|^2$
- compressible subsonic(可压、亚音速) $p_p = p_0 - 0.5\rho|\mathbf{u}|^2$
- compressible transonic(可压、跨音速) $p_p = p_0/(1 + 0.5\psi|\mathbf{u}|^2)$ 其中ψ是压缩率
- compressible supersonic(可压、超音速) $p_p = p_0/(1 + 0.5\psi G|\mathbf{u}|^{1/G})$

衍生的BC

turbulentInlet

主要用于LES进口速度边界条件

$$u_{p,n} = (1-\alpha)u_{p,n-1} + \alpha(u_{ref} + sC_{RMS}u_{ref})$$

```
<patchName>
{
    type            turbulentInlet;
    fluctuationScale 0.1;
    referenceField  uniform 10;
    alpha           0.1;
}
```

衍生的BC

timeVaryingMappedFixedValue

通过对一组时空上的点的值进行插值，作为边界条件，可参考 `tutorials/incompressible/simpleFoam/pitzDailyExptInlet`

```
inlet
{
    type            timeVaryingMappedFixedValue;
    offset          (0 0 0);
    setAverage      off;
}
```

- 可以用于U, ϵ, k, ω, ν_t等
- 还有指定数据存放位置，默认是在`constant/boundaryData/inlet`

衍生的BC

noSlip

壁面上的速度为零，只能用于速度

slip

等同于`basicSymmetry`

foamInfo


```
  -all      | -a        list all tutorials that use <name> (otherwise maximum 10)
  -browser  | -b <name> output C++ source guide web page with specified browser,
                        e.g. foamInfo -browser "firefox"
  -help     | -h        print the usage
  -keyword  | -k        uses <name> as a keyword, rather than an exact match
  -web      | -w        output C++ source guide web page with the browser
                        specified in the global controlDict file

Examples:
  foamInfo simpleFoam
  foamInfo turbulentIntensityKineticEnergyInlet
  foamInfo fixedTemperatureConstraint
  foamInfo surfaces
  foamInfo foamNewBC
  foamInfo wallFunction
  foamInfo kEpsilon
  foamInfo -k kEpsilon
  foamInfo fixedValue
  foamInfo -k fixedValue
```


还有一些

- foamGet
- foamToVTK
- foamSearch
- foamCleanPath
- foamCleanPolyMesh

Thank you.

欢迎私下交流，请勿上传网络，谢谢！