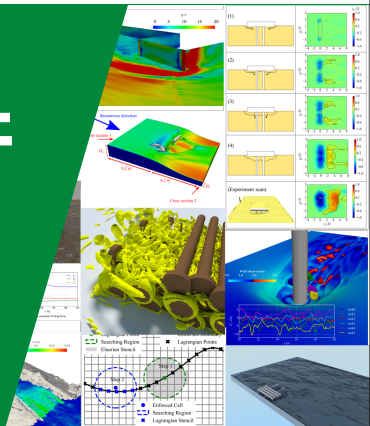




## 2023 年春季《计算流体力学编程实践》

徐云成

✉ycxu@cau.edu.cn



2023 年 2 月 28 日

## 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  $\phi$   
 $\Rightarrow$  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



constant/polyMesh/boundary

O/U

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

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

wallFvPatch 和 fvPatch 在  
src/finiteVolume/fvMesh/fvPatches

fixedValueFvPatchField 和  
zeroGradientFvPatchField 在  
src/finiteVolume/fields/fvPatchFields



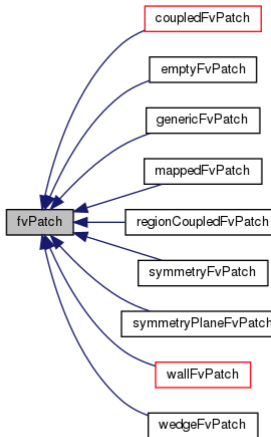
# fvPatch 相关

4/36

src/finiteVolume/fvMesh/fvPatches

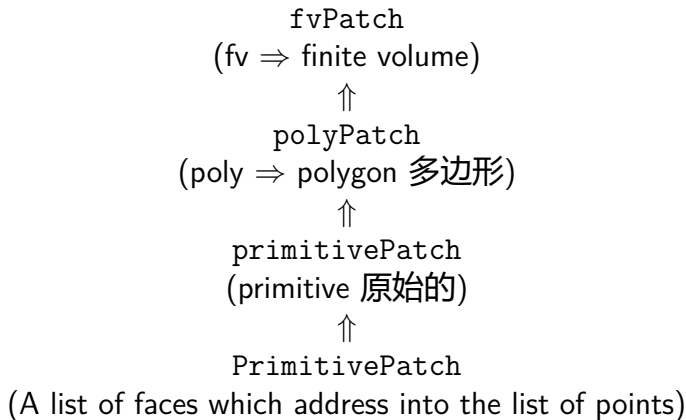
```
yux116@XuLinux:fvPatches$ tree -L 2
```

```
.
├── 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
```



[legend]

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  
↑  
polyPatch  
↑  
primitivePatch  
↑  
PrimitivePatch

fvMesh  
↑  
polyMesh  
↑  
primitiveMesh



```
$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



```
$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





```
$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，而不是零

## 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，而不是零

## 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，而不是零

## 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，而不是零

## 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



## 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



## 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





## 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 (zeroGradient) 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



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



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**



## 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



## 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



## 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



## 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

## 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





## 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

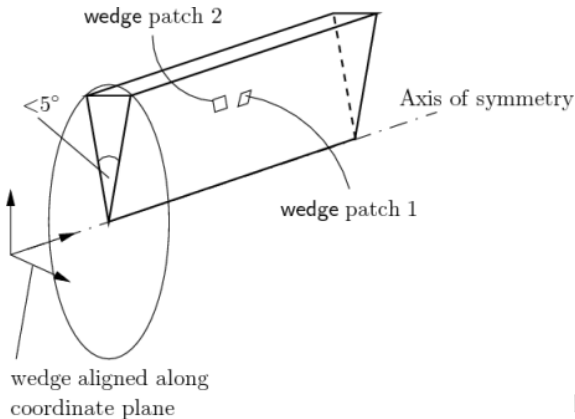
## jumpCyclicAMI

同上



## 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  $\mathbf{n} \times \mathbf{n}_c$

## 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



src/finiteVolume/fields/fvPatchFields/derived/

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



## 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);
    #};
}
```



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



## codedFixedValue

随时间变化

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



## 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;
    #};
}
```



## 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

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

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

## 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

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

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

## 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

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

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

## fixedMean

设定一个平均值，用近边界值的空间分布来调整

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

1. 计算边界网格中心的平均值

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

2. 对于指定的平均值  $\bar{\phi}_0$ ,

$$\phi_p = \begin{cases} \phi_i \frac{\bar{\phi}_0}{\bar{\phi}} & \frac{\bar{\phi}}{\bar{\phi}_0} > 0.5 \\ \phi_i + (\bar{\phi}_0 - \bar{\phi}) & \frac{\bar{\phi}}{\bar{\phi}_0} \leq 0.5 \end{cases} \quad (1)$$



## 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

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

2. 对于指定的平均值  $\bar{\phi}_0$ ,

$$\phi_p = \begin{cases} \phi_i \frac{\bar{\phi}_0}{\bar{\phi}} & \frac{\bar{\phi}}{\bar{\phi}_0} > 0.5 \\ \phi_i + (\bar{\phi}_0 - \bar{\phi}) & \frac{\bar{\phi}}{\bar{\phi}_0} \leq 0.5 \end{cases}$$

(1)



## 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  $Q_0$ ,

$$u_{n,p} = \begin{cases} u_n \frac{Q_0}{Q_{esti}} & \frac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \frac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \frac{Q_{esti}}{Q_0} \leq 0.5 \end{cases} \quad (2)$$

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

4. 如果extrapolateProfile 是no, 边界速度均匀分布

## 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  $Q_0$ ,

$$u_{n,p} = \begin{cases} u_n \frac{Q_0}{Q_{esti}} & \frac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \frac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \frac{Q_{esti}}{Q_0} \leq 0.5 \end{cases} \quad (2)$$

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

4. 如果extrapolateProfile 是no, 边界速度均匀分布

## 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  $Q_0$ ,

$$u_{n,p} = \begin{cases} u_n \frac{Q_0}{Q_{esti}} & \frac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \frac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \frac{Q_{esti}}{Q_0} \leq 0.5 \end{cases} \quad (2)$$

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

4. 如果extrapolateProfile 是no, 边界速度均匀分布



## 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  $Q_0$ ,

$$u_{n,p} = \begin{cases} u_n \frac{Q_0}{Q_{esti}} & \frac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \frac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \frac{Q_{esti}}{Q_0} \leq 0.5 \end{cases} \quad (2)$$

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

4. 如果extrapolateProfile 是no, 边界速度均匀分布

## 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  $Q_0$ ,

$$u_{n,p} = \begin{cases} u_n \frac{Q_0}{Q_{esti}} & \frac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \frac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \frac{Q_{esti}}{Q_0} \leq 0.5 \end{cases} \quad (2)$$

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

4. 如果extrapolateProfile 是no, 边界速度均匀分布

## 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  $Q_0$ ,

$$u_{n,p} = \begin{cases} u_n \frac{Q_0}{Q_{esti}} & \frac{Q_{esti}}{Q_0} > 0.5 \\ u_n - \frac{Q_0 - Q_{esti}}{\rho |\mathbf{S}_f|} & \frac{Q_{esti}}{Q_0} \leq 0.5 \end{cases} \quad (2)$$

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

4. 如果extrapolateProfile 是no，边界速度均匀分布

## 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;
}
```



## 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;

```
}
```



## 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);
}
```



## freestreamVelocity

<patchName>

{

type freestreamVelocity;  
freestreamValue uniform (300 0 0);

}

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

## freestreamPressure

<patchName>

{

type freestreamPressure;  
freestreamValue uniform (300 0 0);  
supersonic false;

}

if (supersonic==true):

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

else:

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



## freestreamVelocity

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

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

## freestreamPressure

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

```
if (supersonic==true):
    valueFraction_i =  $\begin{cases} 0.5 + 0.5 \frac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$ 
else:
    valueFraction_i =  $\begin{cases} 0.5 - 0.5 \frac{\mathbf{u}_i \cdot \mathbf{n}_f}{|\mathbf{u}_i|} & |\mathbf{u}_i| > 0 \\ 0.5 & |\mathbf{u}_i| = 0 \end{cases}$ 
```





## 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/



## 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  $\psi$  是压缩率
- ▶ compressible supersonic(可压、超音速)  $p_p = p_0/(1 + 0.5\psi G|\mathbf{u}|^{1/G})$



## 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;
}
```



## 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  $U$ ,  $\epsilon$ ,  $k$ ,  $\omega$ ,  $\nu_t$  等
- ▶ 还有指定数据存放位置，默认是在 `constant/boundaryData/inlet`



## noSlip

壁面上的速度为零，只能用于速度

## slip

等同于basicSymmetry



-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
```



- ▶ foamGet
- ▶ foamToVTK
- ▶ foamSearch
- ▶ foamCleanPath
- ▶ foamCleanPolyMesh



Thank you.

欢迎私下交流，请勿上传网络，谢谢！

