



[新浪首页](#) [登录](#) [注册](#)

- 1 icp许可
- 2 西山壹号院
- 3 程序员外包公司
- 4 网在线客服系统
- 5 自己建个网站
- 6 翻译公司
- 7 网站开发 公司
- 8 顺义别墅出售
- 9 客服服务在线
- 10 开美国银行账户
- 11 水滴胸形
- 12 自己做亲子鉴定
- 13 天津市落户条件

Jay_见贤思齐的博客

<http://blog.sina.com.cn/openfoam> [\[订阅\]](#) [\[手机订阅\]](#)

[首页](#) [博文目录](#) [关于我](#)

正文

字体大小: [大](#) [中](#) [小](#)

OpenFOAM常用类的一些总结 (2009-05-31 02:50:29)

[转载](#) ▼

标签: [openfoam](#) [类](#) [cfd](#) [研究](#) [教育](#) 分类: [OpenFOAM](#)

OpenFOAM中有许多类，每个类的功能都很强大，这也使它面向对象设计得以实现。

对于程序，最常用到的，也是最底层的就是数据，在OpenFOAM中引入了三类基础数据类型：[标量scalar](#)，[向量vector](#)，[张量tensor](#)。这三个中数据类型，也是FOAM中最基础的三个类。(还有一个比较重要的就是bool和label，前者就是是非型，及对错型，只不过是更扩展一些，后者是标签型数据，相当于c中的整型。关于更多的其它数据类型可以参看目录..\src\OpenFOAM\primitives里面)

在上述数据类的基础上，[增加场\(field\)](#)的概念，就引入了[标量场scalarField](#)，[向量场vectorField](#)，[张量场tensorField](#)。实际上这三个类又是field类的typedef，如typedef field<scalar> saclarField。这些场类中都有对应的成员函数进行加减乘除运算，还有复杂的点积叉积等。说到这field class,其实他就像是一个数据存放的区域一样，存放上scalar，那它成了标量场scalarField。这些类中可以有接口实现数据的计算。从field类中又派生出了FieldField类，这个就是说场中场类，其实这个主要用于边界条件类的一个基类。因为边界条件算是网格类场中的一个特殊的场，后面会介绍。

比field类高一点的就是几何场类 [GeometricField class](#)，其相比field class多了[纪录场位置](#)的相关信息。说到这里请大家注意他和polyMesh class的区别，后者只是纪录网格的结构，如点的位置、面的组成、体的组成等等，[polyMesh class](#)中[对应](#)有[pointMesh](#)，[surfaceMesh](#)，[volMesh](#)等类，从字面上很容易理解其处理和记录网格点、网格面、网格体等信息。而GeometricField类，其则是记录了在什么样的网格上有量a的相关信息或数据。它包括了[内部区域](#)、[边界区域](#) (GeometricBoundaryField class)、[网格](#)、[尺度单位](#)、[计算的先后时间阶的值等](#)。在该类中有常用的三种（实际上还有其他的许多，可以参看OpenFOAM网上说明）：[volScalarField](#)体标量场，[volVectorField](#)体向量场，[volTensorField](#)体张量场。这里说的场与field有所不同，这里指的是网格区域上所对应的数据信息。上述的vol就是指ployMesh中的volMesh，如volscalarField类来说：见下例

```
volScalarField p
(
    IOobject
    (
        "p",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

看过老苏博客的朋友肯定都知道这是什么意思，这是[读入标量压力场文件，把压力值存储到网格体中心](#)。为加深对GeometricField类的理解，贴张PG中的图片：



Jay_见贤思齐

微博

[加好友](#) [发纸条](#)
[写留言](#) [加关注](#)



博客等级: **13**
博客积分: **210**
博客访问: **27,732**
关注人气: **40**
获赠金笔: **3**
赠出金笔: **1**
荣誉徽章:



定制冲锋衣

相关博文

- [万水千山总是情节后喜迎开门红 風雲在线66](#)
- [晒晒中秋节我在家自制的茶饮，家苹果小厨](#)
- [文物鉴定要立法：科学断年代真假 春水秋山居](#)
- [一个优秀气场的形成 苗靳婉青](#)

海南观澜湖的美食
伊夫

实拍中秋国庆中国日月超级圆
快乐考拉

楼市向左，有些开发商开始没底线
杨国英观察

当爱因斯坦选择流亡，那些有文化
凡人摸史

增强耐力健康饮食跑者提升速度的
夜夜野豹

情绪有时候有多可怕？《红楼梦》
我本真心托明月

更多>>



云服务器



孩子说话晚病因



域名购买多少钱



云免费



域名购买多少钱



电动推杆



特价云服务器



租服务器

推荐博文

查看更多>>

谁看过这篇博文

直笔杆	10月21日
Rabbitchen	9月27日
晓东	9月11日
驿路	8月30日
夏沐晴	8月18日
7点钟	6月29日
大嘴啊吧...	6月16日
songsutia...	6月11日
lili	5月31日
轻水云淡	5月25日
我是永强0	5月14日
罗富	5月10日

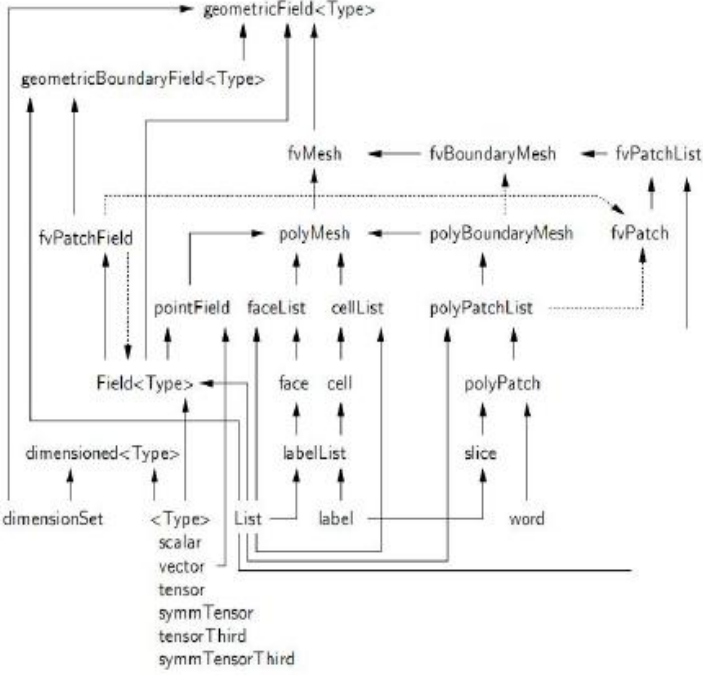


Figure 2.5: Basic class structure leading to geometricField<Type>

除了体的向量标量张量场外，还有面标量场surfaceScalarField、面向量场surfaceVectorField、面张量场surfaceTensorField。看下面的例子：

```
surfaceScalarField phi
(
    IObject
    (
        "phi",
        runTime.timeName(),
        mesh
    ),
    fvc::interpolate(alpha)*phia
    + fvc::interpolate(beta)*phib
);
```

这里的phi既是一个面向量场对象，他用来是纪录单元体面上流过的通量值。

除了常用到的标量向量张量的几何场外，还有一些特殊量的场：surfaceSymmTensor面对称张量几何场、体球面张量场等等。几何场里面还有一个比较重要的类就是GeometricBoundaryField，用来专门对边界进行处理的一个类。

如果说数据场类是处理数据的基础，那么时间类则是控制计算步进必不可少的一部分。Time class在进行瞬态计算，用它跟踪时间阶，并使时间按一定步长或者变步长累加，及输出计算参数，计算时间等。见下例：（相关说明见老苏博客：OpenFOAM>>solver>>incompressible>>icoFoam的说明）

```
Info<< "\nStarting time loop\n" << endl;
for (runTime++; !runTime.end(); runTime++)
{
    Info<< "Time = " << runTime.timeName() << nl << endl;
    //.....
    runTime.write();
    Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
        << " ClockTime = " << runTime.elapsedClockTime() << " s"
        << nl << endl;
```

谈到时间大家很容易想到的就是空间，time and space是cfd中非常重要的概念，在离散过程中时间和空间都需要进行离散（非稳态情况），与空间相关的类其实上面已经提到：poyMesh class，在这不在累述。需要附加说明的是边界条件类：边界条件在OpenFOAM被定义作为场的一个完整部分而不是场上额外附加的。在fvMatrix类中并入了patch用来定义区域的外部边界。每一个patch有一个边界条件，再由fv用合适的方式进行操作处理。patch有多种类型：calculated, fixed value, fixed gradient, zero gradient, symmetry, cyclic等。这些类型都继承于基类patchField。

在求解之前，需要对偏微分方程组进行离散，转化为线性方程组[A][x]=[b]的形式。其中[x]就是我们所要求的量，他也就是咱们前面介绍的GeometricField类，而[A]代数方程的系数，他就是我们下面所要提到的fvMatrix类。对于偏微分方程的每一项，OpenFOAM应用两个类来离散：finiteVolumeMethod和

finiteVolumeCalculus, 分别用typedef声明为fvM和fvc. fvM是计算隐式导数从而返回fvMatrix, 而fvc是计算显式导数或其它隐式计算返回geometricField, 该类不存储私有数据, 仅是执行操作从一个量map到另外一个量上。对于偏微分方程, 其中有很多种导数形式: 拉普拉斯、时间导数、二阶时间导数、对流项、散度、梯度、梯度平方、旋度、源等, 在OpenFOAM中的表示见下图:

Term description	Implicit / Explicit	Text expression	fvM::/fvc:: functions
Laplacian	Imp/Exp	$\nabla^2 \phi$ $\nabla \cdot \Gamma \nabla \phi$	laplacian(phi) laplacian(Gamma, phi)
Time derivative	Imp/Exp	$\frac{\partial \phi}{\partial t}$ $\frac{\partial \rho \phi}{\partial t}$	ddt(phi) ddt(rho, phi)
Second time derivative	Imp/Exp	$\frac{\partial}{\partial t} \left(\rho \frac{\partial \phi}{\partial t} \right)$	d2dt2(rho, phi)
Convection	Imp/Exp	$\nabla \cdot (\psi)$ $\nabla \cdot (\psi \phi)$	div(psi, scheme)* div(psi, phi, word)* div(psi, phi)
Divergence	Exp	$\nabla \cdot \chi$	div(chi)
Gradient	Exp	$\nabla \chi$ $\nabla \phi$	grad(chi) gGrad(phi) lsGrad(phi) snGrad(phi) snGradCorrection(phi)
Grad-grad squared	Exp	$ \nabla \nabla \phi ^2$	sqrGradGrad(phi)
Curl	Exp	$\nabla \times \phi$	curl(phi)
Source	Imp Imp/Exp†	$\rho \phi$	Sp(rho, phi) SuSp(rho, phi)

†fvM::SuSp source is discretised implicit or explicit depending on the sign of rho.
†An explicit source can be introduced simply as a vol<Type>Field, e.g.rho*phi.
Function arguments can be of the following classes:
phi: vol<Type>Field
Gamma: scalar volScalarField, surfaceScalarField, volTensorField, surfaceTensorField.
rho: scalar, volScalarField
psi: surfaceScalarField.
chi: surface<Type>Field, vol<Type>Field.

离散化时空之后, 就是解方程, 其实求解方程的过程主要分成如下几部: 1. 离散偏微分方程, 2. 线性化方程组, 3. 对应不同的倒数格式选择差分格式, 4. 求解系数矩阵, 5. 解方程。在这就涉及到了一个重要的部分: 矩阵。Matrix是OpenFOAM中的一个模板类, 他是一个用来存储及运算标量张量等类型数据的2维矩阵。这个矩阵类有点像数学上的矩阵一样。对于应用于数值求解矩阵, OpenFOAM引入了fvMatrix类, 这个就是有限体积 (finite volume) 矩阵类, 他是一个特殊的矩阵类型, 应用于求解有限体积标量方程组, 该类成员函数可以实现给定相应场的求解、通量的计算、残差的计算和控制、方程松弛因子的实现, 方程中心系数 (central coefficient, 公共成员函数A()) 和H操作源 (H operation source, 成员函数H()) 的计算、设定计算参考等。在此需要提到的是fvMesh类, 该类和GoeMesh类差不多, 不同之处在于fvMesh类它包含相应网格信息和拓扑结构的同时, 还对网格进行实时更新 (动网格的时候)。这些更新包括删除单元体面等, 并按要求重新定位并计算新的信息。下面举一些关于fvMatrix应用的例子: (选自icoFoam)

```
fvVectorMatrix UEqn
(
    fvM::ddt(U)
    + fvM::div(phi, U) //div, 散度是代表某量通过单元体的面积分, 此处phi为一个通
                        //量场, 该场的值被记录在单元体的面上, 而U就是由通量所输运的量, 而该速度值则被记录在单元体中心点上。
    - fvM::laplacian(nu, U)
); //源项可以使显式的, 在离散时进入方程的右端, 当源项为隐式的时候他进入方程的系数矩阵中。
```

solve(UEqn == -fvc::grad(p)); //关于==, 一直有所疑问, 听老苏分析挺有道理, 最近你看到一篇文章上说==的定义是用来表示数学意义上的方程左右两端的等于, 这个运算符为了使其有最低的的运算优先级所以采用了==, 而非=, 同时也强调了方程两端得等的概念, 而非赋值。在OpenFOAM中, 对==的操作实际上是形式上的, 而非实质上有什么运算, 它自动重排方程各项: 所有隐式项写进方程矩阵中, 而所有显式项则归于方程的b中。

```
volScalarField rUA = 1.0/UEqn.A();
U = rUA*UEqn.H();
pEqn.setReference(pRefCell, pRefValue);
pEqn.solve();
```

例子中fvVectorMatrix为一向量有限体积矩阵类, OpenFOAM中定义 typedef fvMatrix<Vector> fvVectorMatrix, 其他的还有标量、张量等。其他关于上述程序的说明参看老苏博客。

```
再看如下例子:
fvMatrixScalar rhoEq
(
    fvM::ddt(rho)+fvc::div(phi)
); //此处是一个关于质量守恒方程的求解, 对于phi为密度与速度的积, 而此时采用fvc即表示速度通量在方程中作为已知量, 出现在方程的b项中, 它是计算前一时间阶的值。
```

说了一些常用的类, 下面介绍一些比较基础底层的类: IOdictionary, argList, IOobject,

IODictionary类是继承于regIOobject类和dictionary类，其主要作于是读入和写入数据。如读取PISO控制参数，或读入transportProperties参数等等。

它派生出许多类：

- 1. basicThermo(用于基本热力学参数读取和计算)
- 2. LESModel（大涡模拟模型控制参数）
- 3. RASModel（RAS模型控制参数）
- 4. fvSchemes（离散格式参数）
- 5. motionSolver（动网格控制参数）
- 6. radiationModel（辐射模型控制参数）
- 7. solution（求解方程控制参数）
- 8. SRFModel（SRF模型控制参数）
- 9. tolerances（方程残差控制）
- 10. transportModel（运输模型参数）

见下例：

```
IODictionary transportProperties //在transportProperties字典中读入参数
(
    IOobject
    (
        "transportProperties",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    )
);
```

IOobject类：读入写入数据，他与IODictionary不同之处在于后者是读取一个文件中的一个字典“{}”之内的数据，而IOobject则是读入整个文件，如读入压力场，速度场等，并且有读入写出的控制参数，见上例中的“MUST_READ,NO_WRITE”等等。（老苏博客中有详细介绍，在此不多说了）

argList类：读入外部命令参数的一个类，如在命令窗口键入icoFoam -case <dir>,则对目录dir执行icoFoam计算，其参数有：-case 选择一个case目录替代当前工作目录；-parallel 指定并行计算参数；-doc显式该程序文档；-srDoc 在浏览器里显示该程序的源文件；-help 显示改程序的使用方法

dimensionSet类是对基本类型的单位设定，并检查其正确性。

tmp类是管理临时对象的一个类。

260

喜欢 赠金笔

分享：

阅读(6667) | 评论 (4) | 收藏(4) | 转载(70) | 喜欢▼ | 打印 | 举报/Report

前一篇：[\[转自linux联盟\]openfoam 网格类编程](#)
后一篇：[名言：三个俯卧撑](#)

cf研究

很好的一篇文章，可惜楼主很久不更新了。

2010-4-3 15:11 回复(0)

Fluid-Ming

好文章一枚，

2014-8-23 11:08 回复(0)

yu德利

如果想获取每个网格节点的物理量值 应该怎么办呢？







2017-2-23 11:12 回复(0)







春竹绿夏







好文

6月7日 11:56 回复(0)

发评论

[更多>>](#)





登录名： 密码： [找回密码](#) [注册](#) ☒ 记住登录状态

☐ 评论并转载此博文

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

[< 前一篇](#)
[\[转自linux联盟\]openfoam 网格类编程](#)

[后一篇 >](#)
[名言：三个俯卧撑](#)