Package: PumpingDiffusionFEMSolver

问题：

给定初态时，要考虑空间归一化。

checklist: MM(ok), MG(ok), vecR(should be ok), CB(ok), vecQ(ok), SS(ok)

# FEMsolver3D – FEM求解器

**继承自：**handle

**功能描述：**

* 用Mesh3D类生成网格（需要meshPars）；
* 用CoeffMatrixGenerator类生成系数矩阵，并处理边界条件和初值（需要simuPars、problemPars）；
* 生成ODE方程组；
* 调用ode45进行时间演化（需要simuPars、problemPars）；
* 把结果保存到文件，供ResultVisualizer类做后续处理

测试一下直接将MM求逆乘在右边做时间演化会不会更快。看看矩阵的性质会不会变坏。（逆矩阵全满。。。存不下）

## 内部属性 (Access = private)

[-coeffMatrix]

CoeffMatrixGenerator对象，记录系数矩阵。simuPars或problemPars被修改时需要重新生成。

## 外部可见属性 (Access = public)

[+meshPars](Dependent) @MeshPars

跟网格相关的参数。

[+simuPars](Dependent) @SimuPars

模拟相关的参数。

[+problemPars] Dependent) @ProblemPars

与问题相关的参数，例如PDE中的系数，边界条件等。

[+mesh] (Dependent)

Mesh3D对象，记录网格数据。meshPars被修改时需要重新生成。

[+M, +S, +MM, +SS, +MP, +MG, +vecQ, +CB, +vecR, +vecF, +MAB](Dependent)

coeffMatrix属性中的系数矩阵

[+H, +F, +u0, +sol\_u, +sol\_t, +FEMResult]

整合好的ODE方程组的系数矩阵以及求解结果

[+baseFunction] (Dependent)

simuPars属性中的baseFunction外链

[+tyRecord](SetAccess=private)

记录了时间演化的中间结果。当时间演化中途意外退出时，可在这里找到已计算好的解。

[-lastOutputTime, -lastReportedT, -lastSaveTime, -lastSavedT]

timeEvolutionOutputFunction()方法的辅助变量。

## 私有方法

[-timeEvolutionOutputFunction]

ode45的output function，用于输出求解进度以及保存中间状态。

## 公共方法

[+getInitialState]

计算初态在基函数下的展开系数。

[+timeEvolution]

进行时间演化。该方法会自动检测网格和系数矩阵的状态，并自动调用makeMesh()及genCoeffs()方法更新信息。

[+saveResultToFile]

创建一个FEMResult结构，记录时间演化的结果以及所有的参数。然后将FEMResult保存到文件。

[+plotMesh, +makeMesh]

Mesh3D对象的plotMesh(), makeMesh()方法接口

[+genCoeffs, + cleanCoeffMatrices]

CoeffMatrixGenerator对象的genCoeffs()和cleanCoeffMatrices()方法接口

# Mesh3D – 三维网格类 (Finished)

继承：handle

提供原来的mesh结构所提供的的所有信息。详情见“3D网格数据结构.docx”

从@MeshPars类读取网格参数。

提供网格可视化方法：plotMesh();

## 属性

[+meshPars\_current] (SetAccess=private)

记录已生成的网格的参数。@MeshPars类

[+meshPars]

记录下一次调用makeMesh()生成网格时将使用的参数。@MeshPars类

[+use\_mex=1]

指示调用makeMesh()方法时是否使用预先编译好的mex函数，mex函数速度较快。

[+outdated] (Dependent, SetAccess=private)

指示已生成的网格是否与meshPars所给的参数一致。实际上输出meshPars\_current与meshPars的比较结果。

[+nodes, +edges, +surfaces, +domains,......]

网格的具体内容。描述见“3D网格数据结构.docx”。

## 方法

[+Mesh3D()]

Constructor。可选参数：meshPars，给定网格参数。若不指定，则使用默认网格参数。

[+makeMesh()]

根据meshPars的参数生成网格。如果网格参数未改变，则不重新生成。makeMesh(1)可强制重新生成网格。

[+plotMesh()]

把网格显示出来。plotMesh(1)可以同时标出网格中各元素的编号。

[-makeMesh3D\_cubic()]

makeMesh()方法在内部实际调用的函数。

# CoeffMatrixGenerator

继承：handle

需要生成的矩阵/向量：MM, SS, MG, vecQ(第二类边界), CB(第一类边界), vecR(第一类边界), MAB(第三类边界), vecF(第三类边界)

## [属性]（默认SetAccess=private）

[+MM, +SS, +MG, +vecQ, +CB, +vecR, +MAB, +vecF]

生成的系数矩阵和向量

[+M, +S, +MP]

中间变量，可能有用。

[+simuPars, +problemPars, +mesh] (SetAccess=public)

输入的参数，由setter方法进行类型检查。

[+simuPars\_current, +problemPars\_current, meshPars\_current]

当前已生成的系数矩阵和向量对应的参数

[+outdated] (Dependent)

判断simuPars, problemPars,meshPars,mesh属性是否过期。若过期，则说明全部系数矩阵都需要重新生成。但未过期并不代表系数矩阵不需要重新生成，因为baseFunction中的基函数编号可能过期了。

[+baseFunHandle, +dimRho] (Dependent)

单纯地将simuPars和problemPars中的参数链接出来，方便访问。

## [方法]

[+CoeffMatrixGenerator()]

Constructor，无参数：用默认参数初始化。

1~3个参数：CoeffMatrixGenerator(varargin)

以输入的参数初始化对象（但不实际生成系数矩阵）。可输入@ProblemPars, @SimuPars, @Mesh3D, @MeshPars

[+genCoeffs()]

调用genCoeffs\_boundary\_independent()生成MM, SS, MG三个矩阵。然后根据边界条件的类型，调用相应的方法生成剩下的边界项。

[+cleanCoeffMatrices()]

清除所有已生成的系数矩阵，以节省内存空间。

[+baseProjectionOnBoundary()]

计算基函数与给定函数在边界上的二维积分：

[-genCoeffs\_boundary\_independent()]

生成M, S, MP矩阵

[-genVecQ()]

生成vecQ

[-genCBvecR()]

生成CB和vecR

[-genMABvecF()]

生成MAB和vecF

# BaseFunction – {抽象类}基函数类 (Finished)

继承：handle, matlab.mixin.Copyable

所有基函数类的父类。基在每个Domain内必须具有直积形式。基函数可以跨多个Domain，但必须连续。一维基函数的定义域为。

* innerProduct(weightFun)：基函数之间的带权重内积
* projection(fun)：基函数与给定函数的内积
* phiphi(), dphidphi(), phidphi()：基于基之间的内积。
* numbering(Mesh3D)：根据网格给基编号，永远包含边界基（不管边界条件类型）
* funVal(), funFirstDerivative(): 输出一维基在给定点的值、一阶导数。

## 属性

[+meshPars] (SetAccess=protected)

只能由baseNumbering()方法设定，指示getNoByIxyz属性对应的网格参数。

[+maxOrder] (SetAccess=protected)

基函数的最大阶数

[+basisName] (SetAccess=protected)

基函数的名字，用于唯一标识基函数的类型。

[+getNoByIxyz] (SetAccess=protected)

由baseNumbering()方法设置。通过基在xyz方向的阶数以及Domain ID来查询基的全局编号。

[+No2fun] (SetAccess=protected)

由baseNumbering()方法设置。通过基的全局编号查询基对应的网格元素。

[+fun2No] (SetAccess=protected)

由baseNumbering()方法设置。通过网格元素查询基的全局编号。

[+Nbasis] (SetAccess=protected)

基的总数量

[-gp\_x, -gw, -gHisLen] (Access=private)

用于辅助getGaussPts()方法的缓存。

## 方法

[+numbering()]

根据网格生成基的全局编号，设定属性getNoByIxyz, No2fun, fun2No以及meshPars。

输入参数：一个已生成网格的Mesh3D对象。

[-getGaussPts()] (Access=protected, Sealed, Hidden)

获取高斯点，用于数值积分。会在对象内部记录之前生成过的高斯点，重复调用无需重新计算。用法：[ gp\_x, gw ] = getGaussPts( ngp );

[+innerProduct()]

[+projection()]

[+funVal()] (Static, Abstract)

[+funFirstDerivative()] (Static, Abstract)

[+phiphi(), +phidphi(), +dphidphi()] (Abstract)

# ResultVisualizer – 画图类

继承：handle

功能：

* 把解映射回实空间；
* 画图。

## 属性(默认SetAccess=private)

[+sampleLines, +sampleLineData]

采样线相关的信息

[+mesh, +simuPars, +problemPars]

网格参数、模拟参数

[+sol\_u, +sol\_t, +solSampleRate]

FEM的求解结果

## 方法

[+ResultVisualizer]

初始化对象，并设定几条默认的采样线。

[+plotLines]

画出指定采样线上的模拟结果

[+addSampleLine, +showSampleLines, +delSampleLines]

添加、列出、删除采样线

[+setFEMResult]

设置FEM的模拟结果。

[-combineSolution]

由plotLines方法在内部调用，功能是将解映射到实空间。

# MeshPars – 网格参数 （Finished）

提供xList, yList和zList给Mesh3D类。

# SimuPars – 模拟参数 (Finished)

提供maxOrder、ngp、basisName、baseFunHandle给CoeffMatrixGenerator类。

# ProblemPars – PDE方程的参数 (Finished)

提供PDE的系数、初始条件、边界条件、求解时长、求解域大小以及解析解。

# LobattoBase – Lobatto基函数 (Finished)

继承：BaseFunction

其中是阶的勒让德多项式。