用MATLAB和COMSOL API进行模拟计算

MATLAB连接到COMSOL Server

MATLAB和COMSOL混合编程的时候，先要运行COMSOL Server，MATLAB中调用COMSOL提供的API和COMSOL Server进行通讯，传递相应的计算模型和参数给COMSOL Server，由COMSOL Server进行计算后返回结果给MATLAB，再由MATLAB进行相应的运算。

MATLAB和COMSOL混合编程的步骤如下

打开COMOSL Server，可以直接打开COMSOL Multiphysics Server程序，或者在cmd（Windows）或者Terminal（Linux，MacOS）中直接输入 `mphstart` ，有些情况下该程序不在当前操作的目录下，应该找到该程序的目录，输入完整路径，比如 `/Applications/COMSOL51/Multiphysics/bin/comsol mphserver` 。第一次打开的时候会要求设置用户名和密码，输入相应的用户名和密码并记下来，之后会用到。在这里我们设置用户名和密码分别为 `zhangchuheng` 和 `zhangchuheng` 。成功运行之后可以再相应的命令行中看到如下信息

```
COMSOL 5.1 (Build: 234) Multiphysics Server started listeni
ng on port 2036
Use the console command 'close' to exit the program
```

下面开始编程的部分，先 将COMSOL所提供的接口文件的路径 `COMSOLXX/Multiphysics/mli` 加入到MATLAB的搜索目录里面，这样之后要用到的一些function和object才能被MATLAB找到，接下来用 `mphstart` 建立和COMSOL Server的连接。如果COMSOL Server运行在本地，只需要使用 `mphstart(2036)` 就可以了。

```
addpath('/Applications/COMSOL51/Multiphysics/mli');
mphstart(2036);
```

如果COMSOL Server运行在远程计算机或者服务器，需要输入相应的IP、用户名和密码，例如

```
addpath('/Applications/COMSOL51/Multiphysics/mli');
mphstart('210.28.142.59',2036,'zhangchuheng','zhangchuheng'
);
```

接下来需要导入相应的java库 `import com.comsol.model.*; import com.comsol.model.util.*;`

总结来说，将MATLAB连接到COMSOL的代码如下

```
addpath('/Applications/COMSOL51/Multiphysics/mli');
mphstart(2036);
import com.comsol.model.*;
import com.comsol.model.util.*;
```

在MATLAB中建立COMSOL的模型

下面给出了一个在MATLAB中建立COMSOL模型的例子，一个建模的过程包括设置参数、建立几何模型、设置材料、设置物理模型、划分网格、设置求解器并求解和获取结果几个步骤。具体的可以参看以下代码。

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Zhang Chuheng (zhangchuheng123@live.com)
% Date: Jan. 15, 2016
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

import com.comsol.model.*;
import com.comsol.model.util.*;

%% Setup the Model
ModelUtil.clear;
list = ModelUtil.tags;
if (numel(list) == 0)
    model = ModelUtil.create('Model');
else
    model = ModelUtil.model(string(list(1)));
end
```

```
model.label('example1.mph');

%% Set up parameters
model.param.set('L', '9[cm]', 'Length of the busbar');
model.param.set('rad_1', '6[mm]', 'Radius of the fillet');
model.param.set('tbb', '5[mm]', 'Thickness');
model.param.set('wbb', '5[cm]', 'Width');
model.param.set('mh', '6[mm]', 'Maximum element size');
model.param.set('htc', '5[W/m^2/K]', 'Heat transfer coeffic
ient');
model.param.set('Vtot', '20[mV]', 'Applied electric potenti
al');

%% Set up Geometry
geom1 = model.geom.create('geom1', 3);
wp1 = geom1.feature.create('wp1', 'WorkPlane');
wp1.set('quickplane', 'xz');

r1 = wp1.geom.feature.create('r1', 'Rectangle');
r1.set('size', {'L+2*tbb' '0.1'});

r2 = wp1.geom.feature.create('r2', 'Rectangle');
r2.set('size', {'L+tbb' '0.1-tbb'});
r2.set('pos', {'0' 'tbb'});

dif = wp1.geom.feature.create('dif', 'Difference');
dif.selection('input').set({'r1'});
dif.selection('input2').set({'r2'});

fil1 = wp1.geom.feature.create('fil1', 'Fillet');
fil1.selection('point').set('dif(1)', 3);
fil1.set('radius', 'tbb');

fil2 = wp1.geom.feature.create('fil2', 'Fillet');
fil2.selection('point').set('fil1(1)', 6);
fil2.set('radius', '2*tbb');

ext1 = geom1.feature.create('ext1', 'Extrude');
ext1.selection('input').set({'wp1'});
ext1.set('distance', {'wbb'});

wp2 = geom1.feature.create('wp2', 'WorkPlane');
wp2.set('planetype', 'faceparallel');
```

```
wp2.selection('face').set('ext1(1)', 8);

c1 = wp2.geom.feature.create('c1', 'Circle');
c1.set('r', 'rad_1');

ext2 = geom1.feature.create('ext2', 'Extrude');
ext2.selection('input').set({'wp2'});
ext2.set('distance', {'-2*tbb'});

wp3 = geom1.feature.create('wp3', 'WorkPlane');
wp3.set('planetype', 'faceparallel');
wp3.selection('face').set('ext1(1)', 4);

c2 = wp3.geom.feature.create('c2', 'Circle');
c2.set('r', 'rad_1');
c2.set('pos', {'-L/2+1.5e-2' '-wbb/4'});

copy = wp3.geom.feature.create('copy', 'Copy');
copy.selection('input').set({'c2'});
copy.set('disply', 'wbb/2');

ext3 = geom1.feature.create('ext3', 'Extrude');
ext3.selection('input').set({'wp3.c2' 'wp3.copy'});
ext3.set('distance', {'-2*tbb'});

geom1.run;
figure, mphgeom(model);

%% Set up Material
sel1 = model.selection.create('sel1');
sel1.set([2 3 4 5 6 7]);
sel1.label('Ti bolts');

figure, mphviewselection(model,'sel1');

mat1 = model.material.create('mat1');
mat1.materialModel('def').set('electricconductivity', {'5.9
98e7[S/m]'});
mat1.materialModel('def').set('heatcapacity', '385[J/(kg*K)
]');
mat1.materialModel('def').set('relpermittivity', {'1'});
mat1.materialModel('def').set('density', '8700[kg/m^3]');
mat1.materialModel('def').set('thermalconductivity', {'400[
W/(m*K)]'});

mat1.label('Copper');

mat2 = model.material.create('mat2');
mat2.materialModel('def').set('electricconductivity', {'7.4
07e5[S/m]'});
mat2.materialModel('def').set('heatcapacity', '710[J/(kg*K)
]');
mat2.materialModel('def').set('relpermittivity', {'1'});
mat2.materialModel('def').set('density', '4940[kg/m^3]');
mat2.materialModel('def').set('thermalconductivity', {'7.5[
W/(m*K)]'});
mat2.label('Titanium');

mat2.selection.named('sel1');

%% Set up Physics
ht = model.physics.create('ht', 'HeatTransfer', 'geom1');

hf1 = ht.feature.create('hf1', 'HeatFluxBoundary', 2);

hf1.set('HeatFluxType', 'InwardHeatFlux');
hf1.selection.set([1:7 9:14 16:42]);
hf1.set('h', 'htc');

% figrue, mphgeom(model,'geom1','facemode','off','facelabel
s','on')

ec = model.physics.create('ec', 'ConductiveMedia', 'geom1')
;

pot1 = ec.feature.create('pot1', 'ElectricPotential', 2);
pot1.selection.set(43);
pot1.set('V0', 'Vtot');

gnd1 = ec.feature.create('gnd1', 'Ground', 2);
gnd1.selection.set([8 15]);

emh = model.multiphysics.create('emh','ElectromagneticHeatS
ource',...
    'geom1',3);
emh.selection.all;
emh.set('EMHeat_physics', 'ec');
emh.set('Heat_physics', 'ht');

%% Set up Mesh
mesh = model.mesh.create('mesh', 'geom1');

size = mesh.feature('size');
size.set('hmax', 'mh');
size.set('hmin', 'mh-mh/3');
size.set('hcurve', '0.2');

ftet = mesh.feature.create('ftet', 'FreeTet');

mesh.run;

figure, mphmesh(model);

%% Set up Study
std = model.study.create('std');
stat = std.feature.create('stat', 'Stationary');

std.run;

%% Extract Result
pg = model.result.create('pg', 'PlotGroup3D');

surf = pg.feature.create('surf', 'Surface');
surf.set('expr', 'T');
surf.set('rangecoloractive', 'on');
surf.set('rangecolormin', '322.6');
surf.set('rangecolormax', '323.3');

figure, mphplot(model,'pg','rangenum',1);
% ModelUtil.disconnect;
```

如何获知COMSOL API的用法

获得COMSOL API用法的方法主要有以下两个方法

> 查看COMSOL提供的官方文档：Introduction To LiveLink For

> MATLAB.pdf，LiveLink For MATLAB Users Guide.pdf
>
> 使用COMSOL GUI正常地编辑一个mph文件，点击`File(文件) > Compact History（压缩历史）`，然后点击`File（文件） > Save As（另存为）`然后再保存类型中选择`Model File For MATLAB(.m)`这样你就可以查看搭建这样一个模型需要写怎样的MATLAB代码了。