

LiveLink™ *for* MATLAB®

简介

LiveLink™ for MATLAB® 简介

© 2009–2018 COMSOL 版权所有

受 cn.comsol.com/patents 中列出的专利和美国专利 7,519,518、7,596,474、7,623,991、8,457,932、8,954,302、9,098,106、9,146,652、9,323,503、9,372,673 和 9,454,625 的保护。专利申请中。

本文档和本文所述的程序根据 COMSOL 软件许可协议 (cn.comsol.com/comsol-license-agreement) 提供，且仅能按照许可协议的条款进行使用和复制。

COMSOL、COMSOL 徽标、COMSOL Multiphysics、COMSOL Desktop、COMSOL Server 和 LiveLink 为 COMSOL AB 公司的注册商标或商标。Matlab 是 公司的注册商标。所有其他商标均为其各自所有者的财产，COMSOL AB 及其子公司和产品不与上述商标所有者相关联，亦不由其担保、赞助或支持。相关商标所有者的列表请参见 cn.comsol.com/trademarks。

版本：COMSOL 5.5

联系信息

请访问“联系 COMSOL”页面 cn.comsol.com/contact，以提交一般查询、联系技术支持或搜索我们的联系地址及号码。您还可以访问全球销售办事处页面 cn.comsol.com/contact/offices，以获取地址和联系信息。

如需联系技术支持，可访问 COMSOL Access 页面并填写在线申请表，位于：cn.comsol.com/support/case 页面。其他有用的链接还包括：

- 技术支持中心：cn.comsol.com/support
- 产品下载：cn.comsol.com/product-download
- 产品更新：cn.comsol.com/support/updates
- COMSOL 博客：cn.comsol.com/blogs
- 用户论坛：cn.comsol.com/community
- 活动：cn.comsol.com/events
- COMSOL 视频中心：cn.comsol.com/video
- 技术支持知识库：cn.comsol.com/support/knowledgebase

文档编号：CM020010

目录

简介	5
启动 COMSOL with MATLAB®	6
详细示例：母线板	8
与 COMSOL Desktop® 共享App	26
在 MATLAB® 命令行提取结果	31
通过 MATLAB® 脚本自动建模	37
使用外部 MATLAB® 函数	47

简介

LiveLink™ *for* MATLAB® 将 COMSOL Multiphysics® 与 MATLAB 脚本环境联系起来，您可以实现：

- 通过脚本设置模型。LiveLink™ *for* MATLAB® 包含 COMSOL® API，提供从零开始实现建模的所有函数和方法。对于在 COMSOL Desktop® 中的每一步操作，在 MATLAB 中都有对应的命令。它们一般是基于 Java® 简化的语法，但是并不要求您具有 Java 知识背景。
- 在模型设置中使用 MATLAB 函数。通过 LiveLink™ 可以使用 MATLAB 函数来设置模型的属性。例如，在求解过程中定义 MATLAB 脚本函数计算得到材料属性或边界条件等。
- 在 COMSOL Desktop 和 MATLAB 之间进行交互式建模。在 MATLAB 提示框中执行的每一步修改都会即时地在 COMSOL Desktop 得到更新。
- 通过 MATLAB 控制语句调节程序流程。通过 API 语法与 MATLAB 的控制语句相结合，可以控制模型的实际使用参数。例如，通过 `for` 或 `while` 语句执行嵌套的循环，通过 `if` 或 `switch` 语句进行条件式建模，或者使用 `try` 和 `catch` 来处理异常等。
- 在 MATLAB 中分析结果。API 包裹函数可以让您在命令行中更轻松地提取数据，可以在节点或任意位置访问结果，还可以获取延展网格的底层信息，例如有限元网格坐标，以及单元和节点之间的连接信息等。提取出的数据可以作为 MATLAB 函数使用的 MATLAB 变量。
- 创建定制模型接口。使用 MATLAB GUIDE 工具箱或 App 设计器功能可以创建用户定义的图形化接口连接 COMSOL 模型，使得模型可以通过开放特定设置和参数的自定义图形化用户接口，与其他使用者分享模型。
- LiveLink™ *for* MATLAB® 能与 COMSOL Server™ 以及 COMSOL Multiphysics Server 相连。这意味着利用 COMSOL 功能的 MATLAB 脚本和 GUI 可分发到对 COMSOL Server™ 具有访问权的任何用户并供他们使用。

本指南中的案例将引导大家了解设置 COMSOL 模型的流程，并解释如何在 MATLAB 脚本环境中使用 COMSOL Multiphysics。

启动 COMSOL with MATLAB®

在 Windows® 上启动

在桌面上双击 COMSOL with MATLAB 图标，启动 COMSOL Multiphysics with MATLAB。



MATLAB 的桌面环境将与 COMSOL Multiphysics Server 同时打开，后者以命令窗口的形式显示在背景中。

在 Mac OS X 上启动

前往 Applications>COMSOL 5.4>COMSOL 5.4 with MATLAB。

在 Linux® 上打开

打开一个终端提示窗口，在其中执行 `consol` 命令，它一般位于 COMSOL 安装目录的 `bin` 文件夹中：

```
consol mphserver matlab
```

COMSOL 客户端服务器连接

LiveLink™ for MATLAB® 基于 COMSOL 客户端 / 服务器架构，提供 COMSOL 和 MATLAB 之间的接口。在 MATLAB 中运行一个 COMSOL 瘦客户端，通过 MATLAB Java 接口来访问 COMSOL API，其中模型信息储存在 COMSOL Multiphysics Server 的模型对象中。这个瘦客户端与 COMSOL

Multiphysics Server 通讯，使得您可以在 MATLAB 提示窗口中生成、修改和求解 COMSOL 模型对象。

当启动 COMSOL with MATLAB 时，同时打开了 COMSOL Multiphysics Server 和 MATLAB 桌面。当第一次启动 COMSOL Multiphysics Server 时，需要键入一个用户名和密码。当这些信息输入以后，就建立了客户端 / 服务器通讯，而且这些信息将被保存到用户首选项中，在以后的使用中就不再需要用户重新输入。

COMSOL Multiphysics Server 和 MATLAB 桌面在同一台计算机中运行，如果需要更多内存的计算资源，您可以连接到远程 COMSOL Multiphysics Server，但是这种解决方案需要网络浮动许可证。

注意，当运行 COMSOL Multiphysics with MATLAB 时，COMSOL Desktop 不必启动。不过，您可以将 COMSOL Desktop 连到 COMSOL Multiphysics Server 并导入模型供服务器使用。这种操作方法可以同时更新 MATLAB 提示窗口和 COMSOL Desktop 中的模型。详情请参阅[与 COMSOL Desktop® 共享 App](#)。

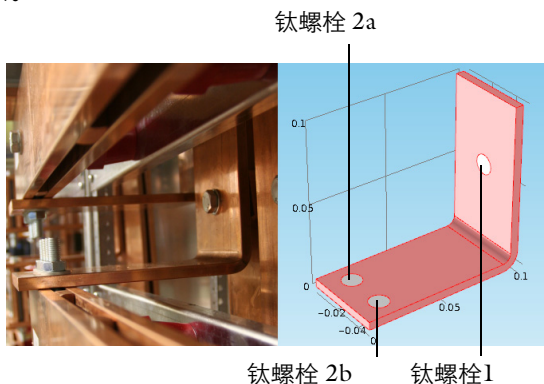
详细示例：母线板

本例可帮助您熟悉 COMSOL 模型对象，以及 COMSOL API 语法。本节中将学习如何：

- 创建几何
- 设置网格和应用物理属性
- 求解问题
- 生成分析结果
- 在 MATLAB 脚本接口和 COMSOL Desktop 之间交换模型

这个将会在 MATLAB 命令行构建的模型，与 COMSOL Multiphysics 简介中的模型类似，差别在于，在本指南中，您将使用的是 COMSOL 模型对象，而不是 COMSOL Desktop。

这个多物理场案例描述母线板中的电热现象，母线板一般用于从变压器将直流电传导到电气设备，本例中采用铜制母线板，通过钛螺栓连接，如下图所示。



注：以下分步操作指南是顺序衔接的，如果跳过其中的某些段落，有可能会无法进行后续的步骤。请从[关于模型对象](#)开始，依次执行到最后一段[保存模型](#)。

关于缩写表示法

本书使用了缩写表示法来减少在 MATLAB 命令行窗口中输入的命令行的长度，这些缩写表示法使用 MATLAB 变量作为链接来直接访问 COMSOL 模型的对象特征。

例如，以下命令使用缩写表示法创建一个长方体：

```
blk = geom.feature.create('blk1', 'Block');  
blk.setIndex('size', '2', 0);  
blk.setIndex('size', '3', 1);
```

以上指令将创建一个长方体，然后改变其宽度为 2，深度为 3。

与上面的指令相比，完整表示法的形式应该如下所示：

```
geom.feature.create('blk1', 'Block');  
geom.feature('blk1').setIndex('size', '2', 0);  
geom.feature('blk1').setIndex('size', '3', 1);
```

当我们将一个模型保存成 M 文件格式时，程序会采用完整表示法形式。

注：'blk1' 是 COMSOL 模型对象中定义的长方体几何的标记。而变量 blk 则是 MATLAB 定义的对象，表示与长方体特征的链接。通过将 MATLAB 变量与特征链接，您可以高效地直接使用和修改 COMSOL 模型对象的特征。

关于模型对象

模型对象包含所有关于模型的信息，从几何到结果。模型对象在 COMSOL 服务器上定义，可以使用 MATLAB 链接在 MATLAB 命令行中直接访问。

1 启动 COMSOL with MATLAB，请参见[启动 COMSOL with MATLAB®](#)。

2 通过在 COMSOL Multiphysics Server 上创建模型对象开始建模。即在 MATLAB 命令行中输入以下命令：

```
model = ModelUtil.create('Model');
```

COMSOL Multiphysics Server 上的模型对象标签为 Model，而变量 model 是它在 MATLAB 中的标签。

要访问和修改模型对象，请使用 COMSOL API 语法。使用函数 mphdoc 可获得有关特定 API 命令的文档。

- 3 要从 *COMSOL Programming Reference Manual* 获取节点模型的文档，请输入以下命令：

```
mphdoc(model)
```

- 4 要获取几何特征 WorkPlane 的文档，输入以下命令：

```
mphdoc(model.geom, 'WorkPlane')
```

连接 COMSOL Desktop® 中的模型

可以将您正在运行的模型从 MATLAB 提示窗口连接到 COMSOL Desktop，这样做可以更加方便地跟踪在提示窗口中键入命令时对模型的修改。详情参见与 [COMSOL Desktop® 共享 App](#)。

- 1 在 MATLAB 提示符下输入：

```
mphlaunch
```

现在这个模型既可以在 MATLAB 提示窗口，也可以在 COMSOL Desktop 访问，在提示窗口中键入的每条命令将会更新 COMSOL Multiphysics Server 和 COMSOL Desktop 的模型。

如果服务器上仅加载了一个模型，则不含变元时 `mphlaunch` 也会起作用。如果服务器上加载了多个模型，则在使用 `mphlaunch` 时需要指定模型标记，此命令将写入一个可供选择的已加载模型列表。可以使用命令 `mphtags -show` 显示使用 `mphlaunch` 前已加载模型的列表。

关于全局参数

如果您打算求解在几个不同参数值下模型的结果，较为方便的做法是在参数节点定义它们，并充分利用 COMSOL 的参数化扫描功能。全局参数可以用于设置模型的所有步骤的表达式，例如，创建几何、应用物理场条件，或者定义网格等。

- 1 定义母线板模型的参数：

```
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 coefficient');
```

```
model.param.set('Vtot', '20[mV]', 'Applied electric potential');
```

几何

1 创建一个三维几何节点：

```
geom1 = model.geom.create('geom1', 3);
```

几何节点的创建需要输入一个几何名称（'geom1'）标签和空间维度（3 表示三维）：

2 通过拉伸一个二维绘图得到初始的几何。首先创建一个标签为 'wp1' 的工作平面，然后设置为三维几何的 xz 平面：

```
wp1 = geom1.feature.create('wp1', 'WorkPlane');  
wp1.set('quickplane', 'xz');
```

3 在这个工作平面中，创建一个矩形，设置宽度为 $L+2*tbb$ ，高度为 0.1：

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

注：当矩形的尺寸属性被单引号 ' ' 引起来时，表示 L 和 tbb 是在模型对象中定义。在本例中，它们是在参数节点中定义。

4 创建第二个矩形，设置宽度为 $L+tbb$ ，高度为 $0.1-tbb$ ，然后将矩形的位置调整到 (0;tbb)：

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

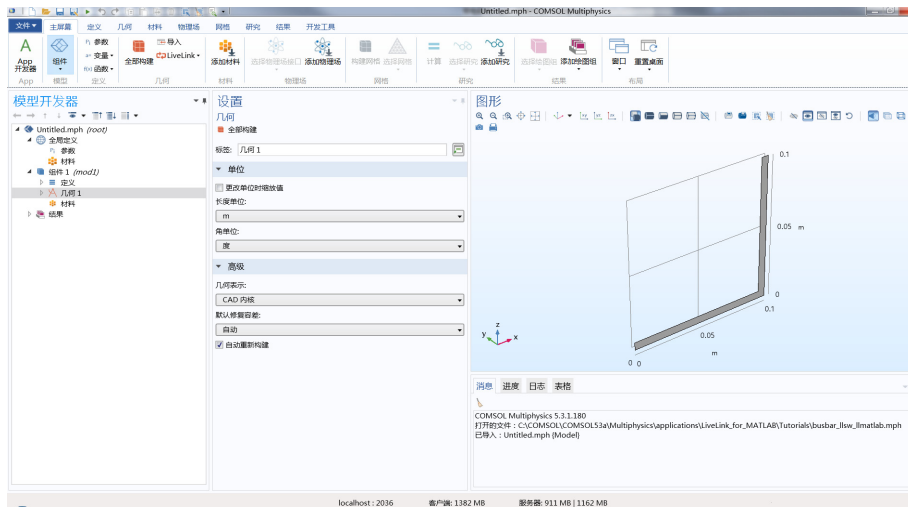
5 如果要在 COMSOL Desktop 中显示当前的几何，需要在 MATLAB 提示窗口中键入：

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

如果要在 COMSOL Desktop 中显示当前的几何，需要在 MATLAB 提示窗口中键入：

```
geom1.run;
```

在 COMSOL Desktop，可以图形化显示当前在 MATLAB 提示窗口中构建的几何。



- 6 创建一个圆角特征将内角进行圆角处理，在选择属性中设置为 3，然后将半径设置为 `tbb`：

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

- 7 创建一个新圆角特征将外角进行圆角处理，选择点 6，然后设置半径为 `2*tbb`：

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

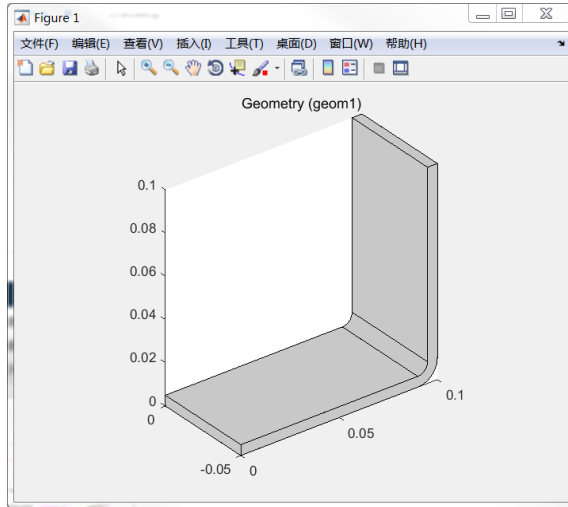
对于几何运算，得到的几何对象的名称是在几何运算标签后面加上括号括起来的一个数字。上面的 '`fil1(1)`' 就是 '`fil1`' 的运算结果。

- 8 拉伸工作平面上的几何对象。创建一个拉伸特征，将平面 `wp1` 设置为输入，距离设置为 `wbb`：

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

您可以在 MATLAB 提示窗口中键入以下命令在 MATLAB 图像窗口中显示当前的几何：

mphgeom(model)



注意 mphgeom 自动地构建几何节点，并更新 COMSOL Desktop。

现在生成了母线板的几何形状。接下来，创建用来表示将母线板联接到外部框架（本案例中不考虑）的螺栓的圆柱。

9 创建一个新的工作平面，将平面类型属性设置为**面平行**，然后选择边界 8：

```
wp2 = geom1.feature.create('wp2', 'WorkPlane');  
wp2.set('planetype', 'faceparallel');  
wp2.selection('face').set('ext1(1)', 8);
```

10 创建一个圆，将半径设置为 rad_1：

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

11 创建一个**拉伸**节点，选择第二个工作平面 wp2 作为输入，然后设置拉伸距离为 $-2 \cdot t_{bb}$ ：

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

12 创建一个新工作平面，设置**平面类型**为**面平行**，然后选择边界 4：

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

13 创建一个圆，将半径设置为 `rad_1`，把圆心的位置设置为 $(-L/2 + 1.5e-2; -wbb/4)$ ：

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

14 通过复制前一个圆来创建第二个圆，并在 y 方向上偏离 `wbb/2`：

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

15 从工作平面 `wp3` 拉伸圆 `c2` 和 `copy1` 到距离 `-2*tbb`：

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

16 构建整个几何序列，使用 `run` 方法来包含定型节点：

```
geom1.run;
```

注：`run` 方法只构建需要重新构建或还没有被构建的特征，包括定型节点。

至此完成模型的几何构建步骤。在 COMSOL Desktop 中，您现在可以看到完整的模型几何。

选择

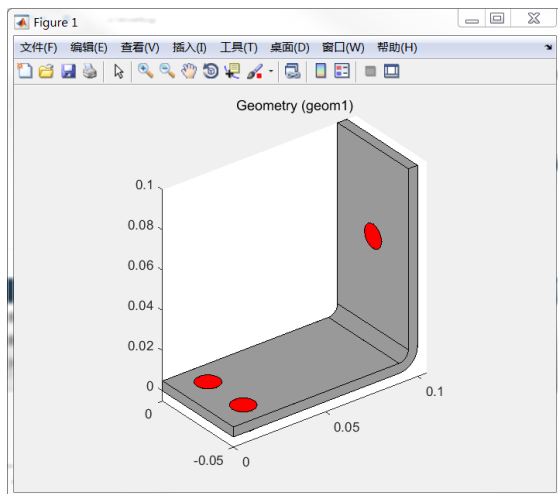
您可以创建几何实体的选择，例如域、边界、边或点等。这些选择可以在建模过程中使用，而不需要反复选择相同实体的操作。

1 创建命名为 `Ti bolts`，对应于钛螺栓的域选择，键入：

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

2 键入以下命令在 MATLAB 图像窗口中显示选择：

```
mphviewselection(model,'sel1');
```



材料属性

模型对象的材料节点储存材料属性。在本例中，使用焦耳热接口，其中包含电流和热平衡物理场。因此，需要定义电导率、热容、相对介电常数、密度，以及热导率等材料属性。

母线板由铜制成，螺栓由钛制成。这两种材料所需的属性列表如下：

属性	COPPER	TITANIUM
电导率	5.998e7[S/m]	7.407e5[S/m]
热容	385[J/(kg*K)]	710[J/(kg*K)]
相对介电常数	1	1
密度	8700[kg/m^3]	4940[kg/m^3]
热导率	400[W/(m*K)]	7.5[W/(m*K)]

1 创建第一种材料，铜：

```
mat1 = model.material.create('mat1');
```

- 2 根据上表设置属性, 'electricconductivity'、'heatcapacity'、'relpermittivity'、'density' 和 'thermalconductivity' :

```
mat1.materialModel('def').set('electricconductivity', {'5.998e7[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)]'});
```

- 3 设置材料名称为 'Copper', 默认情况下, 第一种材料赋给所有域:

```
mat1.label('Copper');
```

- 4 使用上表中钛的材料属性创建第二种材料, 名称为 'Titanium':

```
mat2 = model.material.create('mat2');  
mat2.materialModel('def').set('electricconductivity', {'7.407e5[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');
```

- 5 将材料赋给前面创建的选择 'sel1' (对应于螺栓求解域):

```
mat2.selection.named('sel1');
```

每个求解域只能赋值一种材料, 这意味着最后这一步操作从铜材料的选择中移除了螺栓。

物理场接口

物理场节点包含物理场接口的设置, 其中包括边界设置, 所有的设置按照所属的物理场接口分组。要模拟本例中的电热耦合, 需要在模型中添加导电介质和传热接口。在上螺栓边界上施加一个固定的电势, 下面的两个螺栓边界接地。此外, 假设整个设备对流散热, 除了几个连接的螺栓面, 几乎是在所有外表面上根据定义的传热系数产生热通量。

- 1 在 geom1 几何上创建传热物理场接口:

```
ht = model.physics.create('ht', 'HeatTransfer', 'geom1');
```

- 2 在物理场接口中添加热通量边界条件, 设置类型为 InwardHeatFlux:

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


注：创建方法的第三个变元，空间维度 **sdim**，表示该特征应用到哪个几何层次（域、边界、边或点）。在上面的命令中，'HeatFluxBoundary' 特征应用于边界，空间维度为 2。

- 3 现在应用冷却到所有的外部边界 1 到 43，除了螺栓连接边界 8、15 和 43。InwardHeatFlux 边界条件需要传热系数，将它设置为前面定义的参数 **htc**：

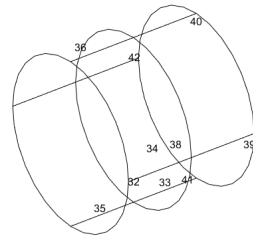
```
hf1.set('HeatFluxType', 'InwardHeatFlux');  
hf1.selection.set([1:7 9:14 16:42]);  
hf1.set('h', 'htc');
```

- 4 您也许注意到了，定义选择需要您知道实体的编号。如果要显示边界索引，可以显示带面标记的几何：

```
mphgeom(model, 'geom1', 'facemode', 'off', 'facelabels', 'on')
```

您可以在窗口中使用缩放和平移的控制，读取索引。

注：在 *LiveLink™ for MATLAB® User's Guide* 中有其他获取实体索引的方法描述，包括使用点坐标、选择框、或邻接信息等。



- 5 在 **geom1** 几何上创建导电介质物理场接口：

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

- 6 现在创建一个电势边界条件，选择边界 43；然后设置电势为 **Vtot**：

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

- 7 在边界 8 和 15 上施加接地边界条件：

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

模型对象包含默认属性，一般不需要在脚本中编写，因此您不需要对所有边界设置属性。例如，导电介质物理场接口使用电绝缘作为电流平衡的默认边界条件。

多物理场接口

多物理场耦合节点包含模型中使用的物理场之间的可能存在的耦合。本例中要模拟电热耦合，需要在模型中添加**电磁热源**接口。还有一些关于电热耦合的其他多物理场耦合，例如**边界电磁热源**，或**温度耦合**接口，它们通常在 COMSOL Desktop 中创建电磁热模型时属于默认项。在本例中，不需要计算这些耦合。

- 1 在 **geom1** 几何上创建 **ElectromagneticHeatSource** 多物理场接口，选择所有求解域：

```
emh = model.multiphysics.create('emh','ElectromagneticHeatSource',...  
    'geom1',3);  
emh.selection.all;
```

- 2 现在设置在耦合中包含**传热和导电介质**物理场接口：

```
emh.set('EMHeat_physics', 'ec');  
emh.set('Heat_physics', 'ht');
```

网格

网格序列储存在**网格**节点。可以在一个模型对象中创建多个网格序列，又称为网格实例。

- 1 创建 **geom1** 的新网格实例：

```
mesh = model.mesh.create('mesh', 'geom1');
```

- 2 默认情况下，网格序列至少包含一个**尺寸**特征，应用于所有的网格运算。首先创建一个链接到已有的尺寸特征，然后设置最大网格尺寸 **hmax** 为 **mh**，最小网格尺寸 **hmin** 为 **mh-mh/3**，曲率因子 **hcurve** 为 **0.2**：

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

- 3 创建一个自由的四面体网格：

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

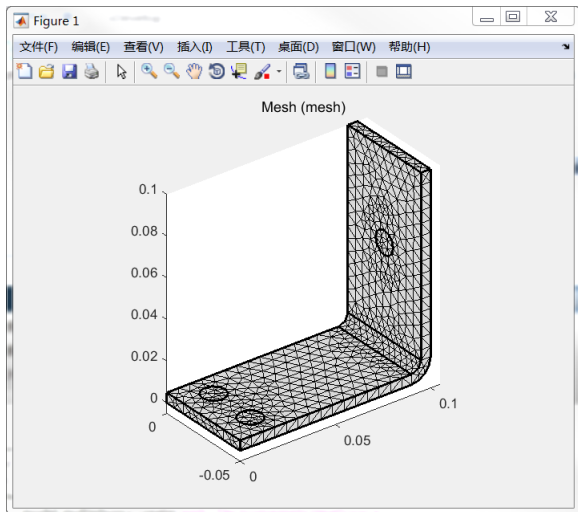
- 4 构建网格：

```
mesh.run;
```

现在可以在 COMSOL Desktop 的图形窗口中看到生成的网格。

5 可以使用 `mphmesh` 命令在 MATLAB 图像窗口中显示网格：

```
mphmesh(model)
```



研究

要求解模型，您需要创建一个研究节点，在其中设置求解器的分析类型。

1 创建研究节点，并添加稳态研究步骤：

```
std = model.study.create('std');  
stat = std.feature.create('stat', 'Stationary');
```

2 默认情况下，并不会显示求解和网格运算的进度条。键入以下命令来激活进度条：

```
ModelUtil.showProgress(true);
```

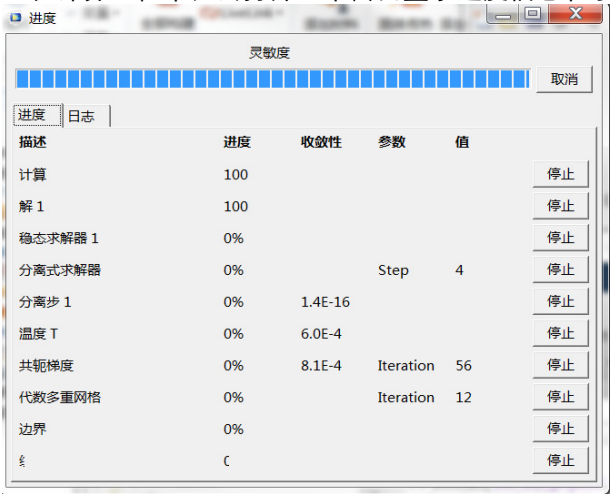
激活进度条后，它将会在网格和求解器运算时显示一个独立的窗口，其中还包含日志信息。

注：在 Mac OS X 上不能使用进度条。

3 使用以下命令来求解模型：

std.run;

在计算过程中，会打开一个窗口显示进度信息和求解器日志。



注：本例中不需要对求解器进行设置，因为研究节点会根据求解类型、物理场接口，以及模型的空间维度等，自动地构建求解器序列。

绘制结果

在结果节点中定义绘图组来分析结果。可以使用 `mphplot` 命令，在 MATLAB 图形窗口中生成绘图。

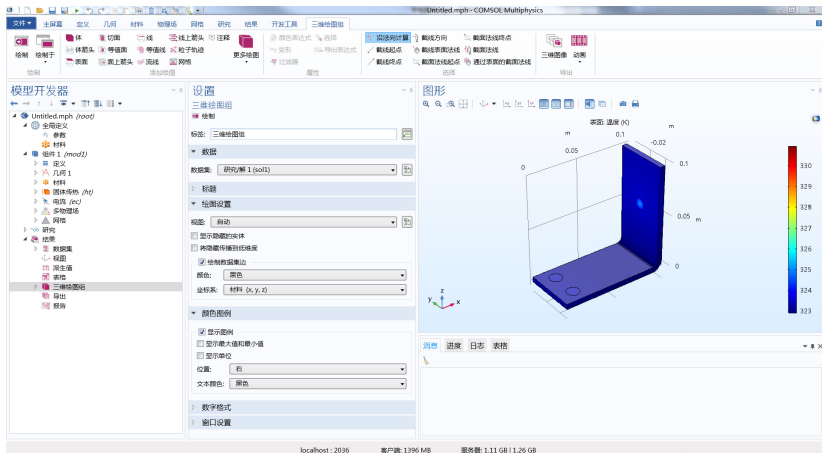
1 创建一个三维绘图组：

```
pg = model.result.create('pg', 'PlotGroup3D');
```

2 在绘图组中，创建一个面图来显示母线板上的温度：

```
surf = pg.feature.create('surf', 'Surface');  
surf.set('expr', 'T');
```

您可以切换到 COMSOL Desktop。在模型构建器中，选择三维绘图组 1 节点来显示结果。



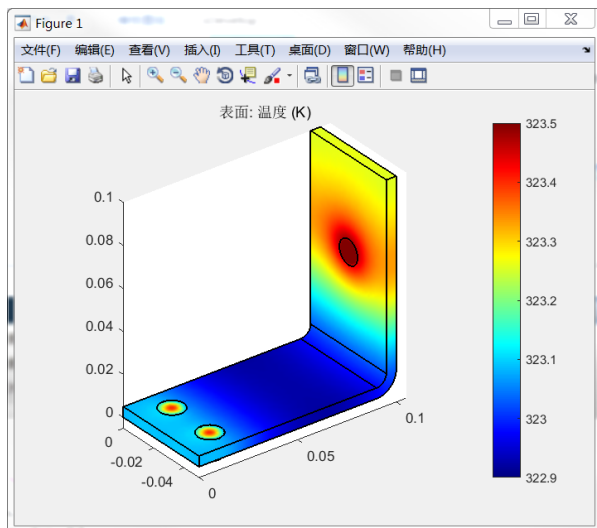
您可能会注意到最大温度区位于螺栓处，要获得母线板中温度分布的更好的渲染效果，您需要改变颜色范围。

3 激活手动控制颜色范围；设置最小颜色值为 322.9，最大颜色值为 323.5：

```
surf.set('rangecoloractive', 'on');
surf.set('rangecolormin', '322.9');
surf.set('rangecolormax', '323.5');
```

4 在 MATLAB 图形中显示包含颜色条的绘图组：

```
mphplot(model, 'pg', 'rangenum', 1)
```



注：如果在同一个绘图组中有多个绘图类型，您可以定义显示哪一个颜色条。rangenum 属性值对应于绘图组序列中的绘图类型序号。

导出结果

使用 LiveLink for MATLAB，您可以使用模型中的导出特征将数据导出成文本文件，也可以使用 MATLAB 支持的 COMSOL 函数将数据直接导出到 MATLAB 工作空间。参见在 [MATLAB® 命令行提取结果](#)，了解更多关于在 MATLAB 中导出数据的信息。

使用**导出节点**，可以轻松地将结果导出成文本文件。

- 1 键入以下的命令，创建一个数据导出节点，导出温度变量 T。设置文件名为 `<filepath>\Temperature.txt`，其中 `<filepath>` 应当替换成您希望保存文件的目录。

```
data = model.result.export.create('data', 'Data');
data.setIndex('expr', 'T', 0);
data.set('filename', 'filepath\Temperature.txt');
```

```
data.run;
```

上面的步骤中提取几何中每个计算节点上的温度，按照以下格式保存到文本文件：

```
% Model:
% Version:          COMSOL 5.4
% Date:             Sep 1 2018, 11:53
% Dimension:        3
% Nodes:            1347
% Expressions:       1
% Description:       Temperature
% Length unit:      m
% x          y          z          T (K)
0.095         0          0          323.25594747
0.0974999999 0          0.09696321 323.25823821
0.0999999999 0          0.1         323.25594764
0.0974844290 -0.0040859 0.1         323.25798259
```

保存模型

1 使用 **save** 方法来保存模型：

```
mphsave(model, '<path>/busbar');
```

在上面的命令中，将 **<path>** 替换成您想保存模型的目录。如果没有定义路径，模型将会保存到 MATLAB 的当前目录。您可以使用命令 `pwd` 获取 MATLAB 的当前目录。

默认的保存格式为 COMSOL 二进制格式，扩展名为 `mph`。使用以下命令可以将模型保存成 M 文件：

```
mphsave(model, '<path>/busbar.m');
```

用于 MATLAB® 的代码

```
model = ModelUtil.create('Model');
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 coefficient');
model.param.set('Vtot', '20[mV]', 'Applied electric potential');
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'});
geom1.run;
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'});
mphgeom(model)
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;
sel1 = model.selection.create('sel1');
sel1.set([2 3 4 5 6 7]);
sel1.label('Ti bolts');
mphviewselection(model, 'sel1');
mat1 = model.material.create('mat1');
mat1Def = mat1.materialModel('def');
mat1.materialModel('def').set('electricconductivity', {'5.998e7[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', {'7.407e5[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');
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');
mphgeom(model,'geom1','facemode','off','facelabels','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','ElectromagneticHeatSource','geom1',3);
emh.selection.all;
emh.set('EMHeat_physics', 'ec');
emh.set('Heat_physics', 'ht');
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;
mphmesh(model)
std = model.study.create('std');
stat = std.feature.create('stat', 'Stationary');
ModelUtil.showProgress(true);
std.run;
pg = model.result.create('pg', 'PlotGroup3D');
surf = pg.feature.create('surf', 'Surface');
surf.set('expr', 'T');
surf.set('rangecoloractive', 'on');
surf.set('rangecolormin', '322.9');
surf.set('rangecolormax', '323.5');
mphplot(model,'pg','rangenum',1)
data = model.result.export.create('data', 'Data');
data.setIndex('expr', 'T', 0);
data.set('filename','filepath\Temperature.txt');
data.run;
mphsave(model,'<path>/busbar.m');

```



与 COMSOL Desktop® 共享 App

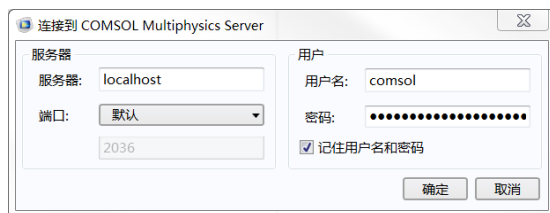
在 MATLAB 中构建母线板模型时，您学习了如何将模型保存成一个 MPH 文件，它可以在 COMSOL Desktop 中打开。在 COMSOL Desktop 中，您还可以在 MATLAB 中调用模型，或者在 COMSOL 中调用 MATLAB 程序，通过这种方式，您可以在 MATLAB 提示窗口或 COMSOL Desktop 中访问 App。例如，您可以一边在 MATLAB 提示窗口中键入命令，一边在 COMSOL Desktop 中看到模型的修改结果。

注：在 Linux 中，您需要在同一个终端上运行 COMSOL with MATLAB 和 COMSOL Desktop。您可以使用 & 标记将进程置入后台，例如，`comsol &` 或 `comsol mphserver matlab &`。

在 MATLAB® 中调用模型

正如在 [COMSOL 客户端服务器连接](#) 节中所述，MATLAB 中创建的模型存在于本地或远程的 COMSOL Multiphysics Server。如果要在 MATLAB 中调用模型，可以将它从 COMSOL Desktop 导出到 COMSOL Multiphysics 服务器，然后在 MATLAB 工作空间中创建一个模型的链接。

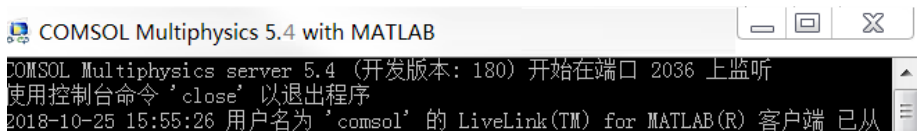
- 1 如果还没有打开模型，请先启动一个新的 COMSOL Desktop。从主屏幕工具栏上选择 **案例库** ()，在案例库窗口中，选择 **COMSOL Multiphysics > 多物理场 > busbar**，单击打开。
- 2 启动 COMSOL with MATLAB（参见 [启动 COMSOL with MATLAB®](#)）。
- 3 在 COMSOL Desktop 中，从文件菜单（Windows 用户）或选项菜单（Mac 和 Linux 用户），选择 **客户端服务器 > 连接到服务器** ()。



4 确认使用了正确的导出信息：

- 在**服务器**栏填写**服务器名称**和**端口号**。默认的服务器名称是 localhost，表示 COMSOL 服务器在本机。您可以在 COMSOL 服务器窗口的第一行文字中察看端口号，一般默认值为 2036：

COMSOL Multiphysics Server 5.4 在端口 2036 启动：



- **用户**栏包含用户的登录信息，**用户名**和**密码**编辑框默认自动填写基于第一次启动 COMSOL 服务器时定义的用户首选项。

5 单击**确定**。

现在模型对象储存在 COMSOL 服务器上了，接下来在 MATLAB 中创建一个链接来获得访问许可。

- #### 6 当导出模型到服务器窗口关闭后，您需要在 MATLAB 中创建一个链接指向导出到 COMSOL 服务器的模型。在 MATLAB 提示窗口中输入：

```
model = ModelUtil.model('Model12');
```

其中 **Model12** 是 COMSOL 服务器上的模型对象标签。另一种方法是在 COMSOL 服务器上键入以下指令来获取模型列表：

```
mphtags -show
```

- #### 7 现在可以访问模型的所有特征。例如，在 MATLAB 图像窗口中显示网格：

```
mphmesh(model)
```

- #### 8 模型还可以被修改，例如，从结果节点中删除第二个绘图组：

```
model.result.remove('pg2');
```

您可以看到，当在 MATLAB 提示窗口中键入命令时，在 COMSOL Desktop 中自动更新了模型。

在 COMSOL Desktop® 中调用模型

除了在 MATLAB 中调用模型外，还可以在 COMSOL Desktop 中访问 MATLAB 命令行中创建的模型。这只需要您从 COMSOL Multiphysics 服务器将模型加载到 COMSOL Desktop。

1 启动 COMSOL with MATLAB。

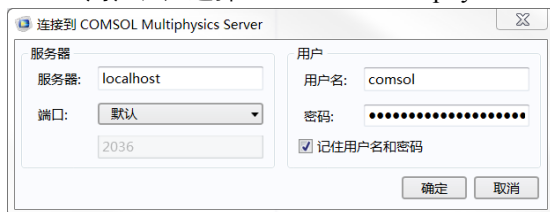
2 创建一个模型对象，然后创建一个三维几何并绘制一个长方体：

```
model = ModelUtil.create('ImportExample');  
model.geom.create('geom1', 3);  
model.geom('geom1').feature.create('blk1', 'Block');  
model.geom('geom1').run;
```

3 要启动 COMSOL Desktop，将其与 COMSOL Multiphysics Server 相连并自动加载模型，输入以下命令：

```
mphlaunch
```

如果不使用命令 `mphlaunch`，您也可以手动执行同一操作。要连接到 COMSOL Desktop，从**文件**菜单（Windows 用户）或**选项**菜单（Mac 和 Linux 用户），选择 COMSOL Multiphysics Server> **连接到服务器** (📡)。



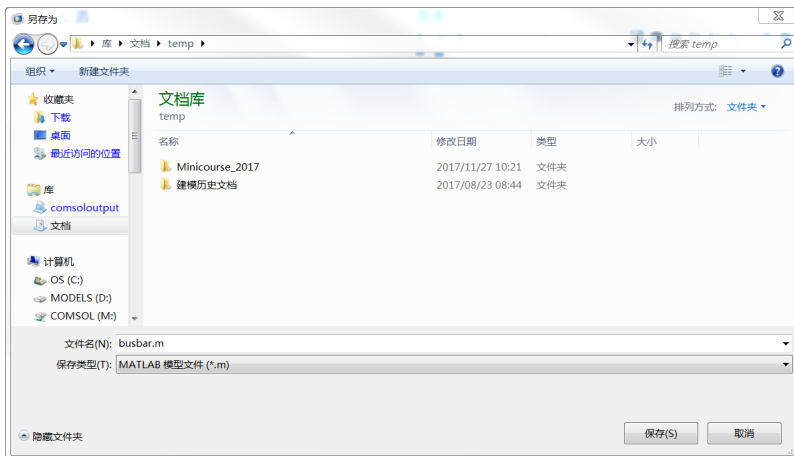
现在 COMSOL Desktop 连接到了 COMSOL Multiphysics Server，您可以导入模型。从**文件**菜单（Windows 用户）或**选项**菜单（Mac 和 Linux 用户），选择 COMSOL Multiphysics Server> **从服务器导入 App** (📡)。在**从服务器导入 App**窗口，选择 `Untitled.mph {ImportExample}`，它对应于前面在 MATLAB 提示窗口中创建的模型。

保存和运行模型 M 文件

最方便的学习 COMSOL API 的方法是从 COMSOL Desktop 将模型保存成 M 文件，在 M 文件中包含创建模型的命令序列。您可以在文本编辑器中打开 M 文件，并修改它，如下例所示。

注：默认情况下，M 文件包含模型的完整命令历史，其中包括部分不再是构建模型所需内容的命令。如果希望只包括当前模型中有效的命令，应该在保存 M 文件之前压缩模型历史。

- 1 启动 COMSOL with MATLAB 和 COMSOL Desktop。
- 2 在 COMSOL Desktop，找到**案例库**窗口。
- 3 在**案例库**窗口中，展开 COMSOL Multiphysics，然后是**多物理场**，选中 busbar，单击**打开**。
- 4 一个好的做法是包含压缩模型历史记录，以获取仅与当前模型状态对应的命令文件。在 COMSOL Desktop 中选择**文件 > 压缩历史记录** (🗑️)。
- 5 单击**文件 > 另存为**，在**保存窗口**，找到**另存为类型**列表，选择适用于 **MATLAB** 的模型文件类型 (*.m)。输入文件名为 busbar，然后选择保存文件的目录，单击**确定**。



- 6 使用 MATLAB 编辑器或其他文本编辑器打开保存的 M 文件，找到下列行：

```
model.result('pg4').feature('surf1').set('expr', 'ec.normJ');  
model.result('pg4').feature('surf1').set('descr', 'Current density norm');  
model.result('pg4').feature('surf1').set('rangecoloractive', true);
```

```
model.result('pg4').feature('surf1').set('rangecolormax', '1e6');
```

上面的命令定义绘图组 pg4，其中包括表达式 `ec.normJ` 的面图，最大颜色范围设置为 `1e6`。

- 7 修改绘图组 pg4，用来显示总热 `jh.Qtot`，最大颜色范围为 `1e4`，在 M 文件中添加以下代码：

```
model.result('pg4').feature('surf1').set('expr','ht.Qtot');  
model.result('pg4').feature('surf1').set('descr','Total heat source');  
model.result('pg4').feature('surf1').set('rangecoloractive', true);  
model.result('pg4').feature('surf1').set('rangecolormax','1e4');
```

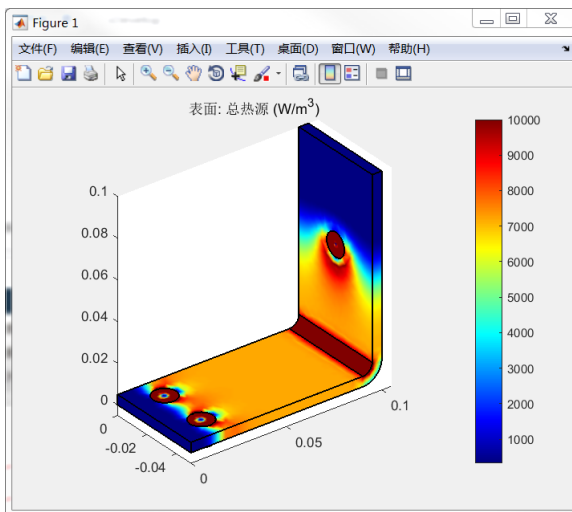
这些命令修改了绘图组 pg4，使得它按照设置的最大颜色范围 `1e4` 来显示总热 `jh.Qtot`。您也可以直接在原来的命令中修改相应的设置，但现在这种做法可以很容易改回到原来的命令。

- 8 保存修改过的 M 文件。
- 9 在 MATLAB 提示窗口按照以下命令来运行模型：

```
model = busbar;
```

- 10 显示绘图组 pg4：

```
mphplot(model,'pg4','rangenum',1)
```



另一种修改 MATLAB 中的模型对象的方法是在 MATLAB 中载入模型，然后在 MATLAB 命令行直接键入步骤 7 中的命令。这种方法的好处是它不需要运行整个模型。参见[使用嵌套式循环示例](#)，其中就使用了这种方法。

在 MATLAB® 命令行提取结果

LiveLink™ for MATLAB® 提供一些函数，可以轻松地从 MATLAB 命令行直接访问结果，最常用的函数包括：

- `mphinterp`，计算任意位置的表达式。
- `mpheval`，在给定域选择上计算所有节点的表达式。
- `mphglobal`，计算全局表达式。
- `mphint2`，在选定域上对表达式的值进行积分。
- `mphmax`、`mphmin` 和 `mphmean`，分别计算表达式的最大、最小和平均值。
- `mphgetexpressions`，获取模型变量表达式和模型参数表达式。
- `mphevaluate`，计算模型中的参数表达式（常数）。

注：要获取可用函数的完整列表，请参阅 *LiveLink for MATLAB User's Guide* 中的章节 [Summary of Commands](#)。

在 MATLAB 命令行从案例库直接加载 `busbar` 模型。

1 启动 COMSOL with MATLAB。

2 从案例库加载 `busbar` 案例模型，键入命令：

```
model = mphopen('busbar');
```

`mphopen` 命令会自动搜索 MATLAB 路径以及案例库的路径，以查找所请求的文件，您可以在文件名中指定路径。

计算任意点数据

如果要得到某个指定位置的数据，不一定是节点定义的位置，可以使用命令 `mphinterp`，通过插值得到结果。使用单元形函数来作为插值方法。

1 键入以下命令来提取点 `[5e-2;-2e-2;1e-3]` 的总热源：

```
Qtot = mphinterp(model,'ht.Qtot','coord',[5e-2;-2e-2;1e-3])
```

此命令返回：

```
Qtot =  
7.2272e+003
```

2 可以指定计算的单位，例如使用以下命令提取华氏温度：

```
[Temp, unit]= mphinterp(model,'T','coord',[5e-2;-2e-2;1e-3],'unit','degF')
```

此命令返回：

```
Temp =  
    121.7608
```

```
unit =  
    'degF'
```

3 还可以计算一系列点，键入下面的命令：

```
x0=0:5e-2/4:5e-2;  
y0=0:-2.5e-2/4:-2.5e-2;  
z0=[0 5e-3];  
[x,y,z]=meshgrid(x0,y0,z0);  
xx=[x(:),y(:),z(:)]';  
Qtot = mphinterp(model,'ht.Qtot','coord',xx);
```

4 将变量 Qtot 的结果重排得到更好的显示效果：

```
Qtot = reshape(Qtot,length(x0),length(y0),length(z0))
```

得到：

```
Qtot(:,:,1) =  
1.0e+05 *  
    0.0000    0.0323    0.0764    0.0725    0.0723  
    0.0008    0.3482    0.0768    0.0722    0.0723  
    0.0000    2.7021    0.0725    0.0720    0.0723  
    0.0006    0.3134    0.0778    0.0723    0.0723  
    0.0000    0.0291    0.0766    0.0725    0.0723
```

```
Qtot(:,:,2) =  
1.0e+03 *  
    0.0006    3.0838    7.6792    7.2481    7.2264  
    0.0787    5.2628    7.1174    7.2193    7.2253  
    0.0004    8.9329    5.7534    7.1990    7.2249  
    0.0616    5.2332    7.3128    7.2300    7.2267  
    0.0003    2.9790    7.6433    7.2540    7.2284
```

计算节点数据

使用 mpheval 函数计算电势，它是 busbar 案例模型的一个因变量。

mpheval 返回单纯网络节点上计算的表达式结果。默认情况下，单纯网络对应于模型对象中激活的网络。

1 键入命令行：

```
data = mpheval(model,'V')
```

返回一个如下所示的 MATLAB 结构体：

```
data =
```



```

expr: {'V'}
d1: [1x1313 double]
p: [3x1313 double]
t: [4x4134 int32]
ve: [1313x1 int32]
unit: {'V'}

```

`mpheval` 返回包含以下字段的 MATLAB 结构体：

- `expr` 包含计算表达式列表，
- `d1` 包含计算表达式的数据，
- `p` 包含计算点的坐标，
- `t` 包含 `p` 字段的列索引，它的每一列对应于计算用到的一个网格单元，
- `ve` 包含每个计算点的网格单元索引，
- `unit` 包含计算表达式的单位

2 可以同时计算多个变量，例如，同时得到温度和电势，键入：

```
data2 = mpheval(model,{'T' 'V'})
```

这条命令会返回下面这个结构体：

```

data2 =
  expr: {'T' 'V'}
  d1: [1x1313 double]
  d2: [1x1313 double]
  p: [3x1313 double]
  t: [4x4134 int32]
  ve: [1313x1 int32]
  unit: {'K' 'V'}

```

字段 `data2.d1` 和 `data2.d2` 包含分别包含温度和电势结果。

3 现在计算温度 `T`，并引入二阶形函数定义的单元中心点，使用 `refine` 属性，并将它设置为对应于离散阶次的 2：

```
data3 = mpheval(model,'T','refine',2)
```

这条命令会返回下面这个结构体：

```

data3 =
  expr: {'T'}
  d1: [1x7900 double]
  p: [3x7900 double]
  t: [4x33072 int32]
  ve: [7900x1 int32]
  unit: {'K'}

```

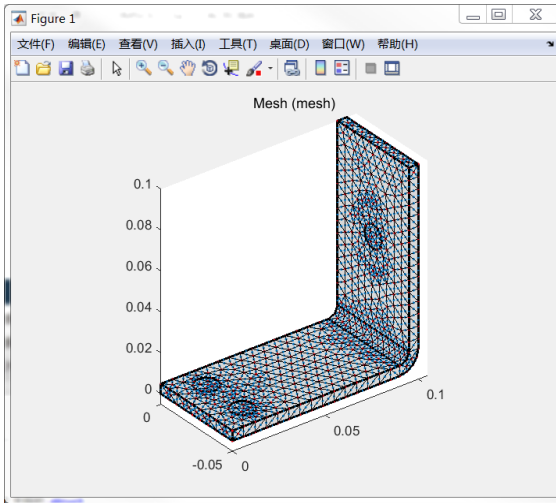
键入以下命令，图形化显示 `data` 和 `data3` 的位置和网格：

```

mphmesh(model)
hold on;

```

```
plot3(data3.p(1,:),data3.p(2,:),data3.p(3,:),'.');
plot3(data2.p(1,:),data2.p(2,:),data2.p(3,:),'.r');
```



- 4 这个步骤测试如何使用 **edim** 属性和 **selection** 属性。**edim** 用来确定空间维度，**selection** 用来设置在哪些对象上计算表达式。

在一个指定域上计算数据，例如，边界 43 上的法向电流密度：

```
data4 = mpheval(model,'ec.normJ',...
    'edim','boundary','selection',43)
```

这条命令会返回下面这个结构体：

```
data4 =
    expr: {'ec.normJ'}
    d1: [1x18 double]
    p: [3x18 double]
    t: [3x22 int32]
    ve: [18x1 int32]
    unit: {'A/m^2'}
```

全局计算和积分

使用函数 **mphglobal** 来得到全局范围内定义的表达式结果。

- 1 计算由 **Vtot** 定义的施加在母线板上的总电势：

```
Vtot = mphglobal(model,'Vtot')
```

```
Vtot =
    0.0200
```

- 2 计算在母线板上产生的总热能，可以使用命令 `mphint2` 对总热源进行积分，键入：

```
Q = mphint2(model,'ec.Qh','volume','selection',1)
```

总热能为：

```
Q =  
    0.2524
```

- 3 计算流过边界 43 的总电流，并得到积分的单位：

```
[I,unit] = mphint2(model,'ec.normJ','surface','selection',43)
```

```
I =  
    164.4663
```

```
unit =  
    'A'
```

- 4 计算母线板上的最大温度：

```
maxT = mphmax(model,'T','volume','selection',1)
```

```
maxT =  
    323.6905
```

- 5 计算边界 43 上的最大温度：

```
maxT = mphmax(model,'T','surface','selection',43)
```

```
maxT =  
    331.0802
```

计算表达式

使用函数 `mphgetexpressions` 可计算模型中的参数、变量或表达式。

- 1 计算模型中定义的参数表达式：

```
expr = mphgetexpressions(model.param)
```

```
expr =
```

'L'	'9[cm]'	'Length'	[0.0900]	'm'
'rad_1'	'6[mm]'	'Bolt radius'	[0.0060]	'm'
'tbb'	'5[mm]'	'Thickness'	[0.0050]	'm'
'wbb'	'5[cm]'	'Width'	[0.0500]	'm'
'mh'	'6[mm]'	'Maximum element s...	[0.0060]	'm'
'htc'	'5[W/m^2/K]'	'Heat transfer coe...	[5]	'W/(m^2*K)'
'Vtot'	'20[mV]'	'Applied voltage'	[0.0200]	'V'

- 2 使用函数 `mphevaluate` 可计算特定参数表达式。如下计算母线板 L 的长度，以英寸计：

```
L = mphevaluate(model,'L','in')
L =
    3.5433
```

注：此计算不要求模型中存在解数据集。

用于 MATLAB[®] 的代码

```
model = mphopen('busbar');

% Evaluating Data at Arbitrary Points
Qtot = mphinterp(model,'ht.Qtot','coord',[5e-2;-2e-2;1e-3])
[Temp, unit]= mphinterp(model,'T','coord',[5e-2;-2e-2;1e-3],'unit','degF')
x0=0:5e-2/4:5e-2;
y0=0:-2.5e-2/4:-2.5e-2;
z0=[0 5e-3];
[x,y,z]=meshgrid(x0,y0,z0);
xx=[x(:),y(:),z(:)]';
Qtot = mphinterp(model,'ht.Qtot','coord',xx);
Qtot = reshape(Qtot,length(x0),length(y0),length(z0))

% Evaluating Data at Node Points
data = mpheval(model,'V')
data2 = mpheval(model,{ 'T' 'V' })
data3 = mpheval(model,'T','refine',2)
hold on;
plot3(data3.p(1,:),data3.p(2,:),data3.p(3,:),'.');
plot3(data2.p(1,:),data2.p(2,:),data2.p(3,:),'.r');
mphmesh(model)

data4 = mpheval(model,{ 'ec.normJ' },...
    'edim',2,'selection',43)

% Global Evaluation and Integration
Vtot = mphglobal(model,'Vtot')
Q = mphint2(model,'ec.Qh','volume','selection',1)
[I unit] = mphint2(model,'ec.normJ','surface','selection',43)
maxT = mphmax(model,'T','volume','selection',1)
maxT = mphmax(model,'T','surface','selection',43)
```

通过 MATLAB® 脚本自动建模

LiveLink™ for MATLAB® 支持将 MATLAB 编程工具与 COMSOL 模型对象相集成，这样做的好处之一是可以从 MATLAB 工作空间访问结果。另一个好处是可以把 COMSOL 建模命令集成到 MATLAB 脚本中，从而可以利用所有可供使用的工具来控制代码流。本节说明如何有效地做到这一点，也就是将 MATLAB 变量引入到 COMSOL 模型，并且只更新受到影响的部分模型对象。

获取模型信息

要修改现有的模型，您需要知道模型是如何设置的。有一些函数可以用来得到这些信息，然后您才能添加或删除模型特征，或者修改已有的属性值。函数 `mphnavigator` 用于获取模型对象信息，在 MATLAB 提示窗口调用 `mphnavigator`，会在 MATLAB 工作空间中显示模型对象的特征及其属性。

1 启动 COMSOL with MATLAB。

2 加载 busbar 案例模型：

```
model = mphopen('busbar');
```

或者使用以下更简单的形式：

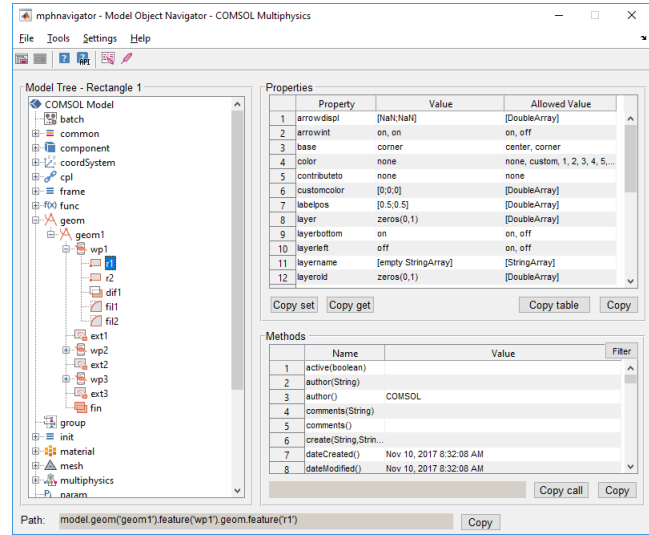
```
mphopen busbar
```

3 在 MATLAB 提示窗口键入以下命令来得到模型信息：

```
mphnavigator
```

在 `mphnavigator` 图形化用户界面，模型树列出了模型的特征，这些节点可以展开看到下级特征。当从模型树中选择一个特征时，在属性区列出了它的属性。Methods 栏中列出选定特征的方法。在 Methods 栏上方，您可以看到相应的 COMSOL API 语法。您可以将命令复制到剪贴板，并在 MATLAB 提示窗口中方便地使用 — 只需要单击复制按钮就可以复制相应的语法。

4 在模型树中，展开模型特征节点 `geom>geom1>wp1>r1`，然后选择节点 `r1`。



5 在属性区，您可以看到几何特征 `r1` 的属性 `lx` 设置为 `L+2*tbb`，表示矩形的长度。而矩形的宽度，由属性 `ly` 表征，设置为 `0.1`。

更新模型设置

使用 `set` 方法，可以修改 COMSOL 模型对象的参数、特征，或者特征属性。当一个属性值更新后，根据在模型中引入变化的位置——例如，在几何或网格序列中，可以运行恰当的序列。如果您运行求解器序列，则运行所有被检测到修改的序列。

启动[详细示例](#)：[母线板](#)中描述的 `busbar` 模型，本节修改其中的参数值，并求解模型。为此，我们使用了长度参数 `L` 和厚度参数 `tbb` 来绘制几何。

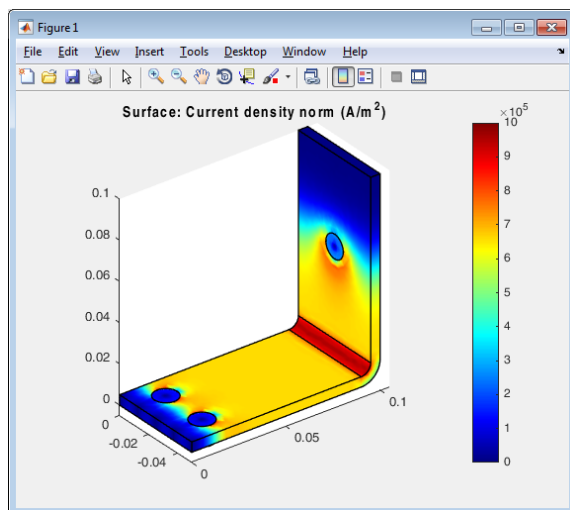
1 启动 COMSOL with MATLAB（如果还没有启动）。

2 从 COMSOL 案例库加载模型：

```
mphopen busbar
```

3 绘制当前解：

```
mphplot(model, 'pg4', 'rangenum', 1);
```



4 练习的第一步是修改长度参数 L （母线板长度），键入：

```
model.param.set('L', '18[cm]');
```

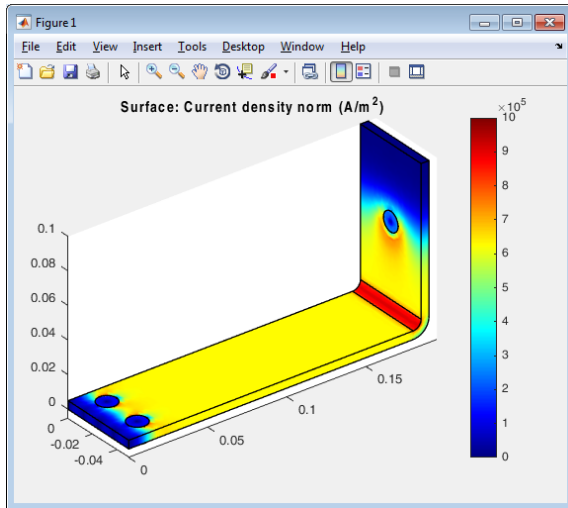
5 运行求解器序列求解结果，键入：

```
model.sol('sol1').run;
```

注：求解器节点自动检测模型中所有修改项目，如果有必要就运行几何和网格序列。本练习中，新的长度参数 L 改变了几何结构，因此就需要重新剖分网格。还需要确保相关性，保证所有的物理设置与原始模型保持一致。

6 绘制更新解，可以看到明显的几何变化：

```
mphplot(model,'pg4','rangenum',1);
```



调用 MATLAB® 代码

```
model = mphload('busbar');
mphplot(model,'pg4','rangenum',1);
model.param.set('L','18[cm]');
model.sol('sol1').run;
mphplot(model,'pg4','rangenum',1);
```

在 COMSOL 模型中使用 MATLAB® 变量

使用上一节中描述的 `set` 方法来使用 MATLAB 变量定义特征属性，重点是确保 MATLAB 变量的单位与模型中使用的单位一致。在开始前，请确认已经在 MATLAB 中载入了上一节使用的 busbar 模型。

首先，改变参数 `L` 描述的母线板长度。

1 创建一个等于 `9e-2` 的 MATLAB 变量 `L0`：

```
L0 = 9e-2;
```

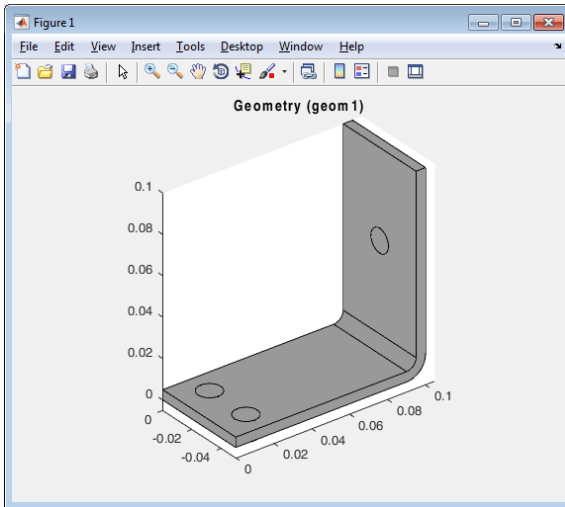
2 更新参数 `L` 为变量 `L0`：

```
model.param.set('L',L0);
```

注：与上述命令相比，在模型对象中使用表达式来设置一个值，第二个变元必须是字符串表达式，如上节[更新模型设置](#)所示。

3 显示新几何，键入：

```
mphgeom(model)
```



使参数生效很简单，因为只有一个地方需要检查或修改。下一步使用另一种方法，编辑特征节点来修改其属性。例如，修改母线板几何，编辑工作平面 wp1 上的矩形 r1 和 r2。

4 通过键入以下命令创建连接到几何特征 r1 和 r2 的链接：

```
r1 = model.geom('geom1').feature('wp1').geom.feature('r1');  
r2 = model.geom('geom1').feature('wp1').geom.feature('r2');
```

5 键入以下命令定义 MATLAB 变量：

```
H1 = 0.2;  
H2 = 0.195;
```

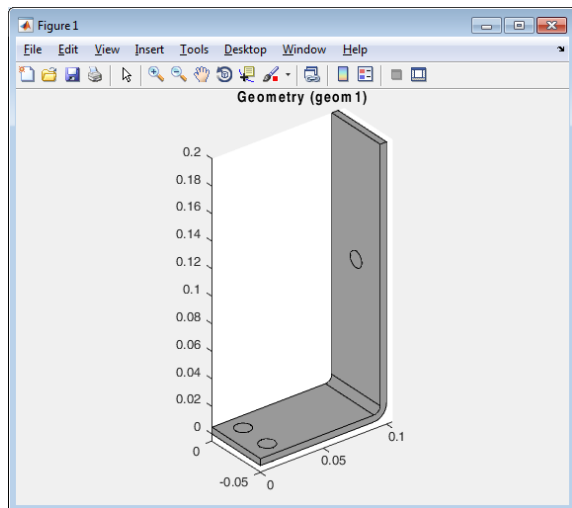
6 使用变量 H1 和 H2 分别设置矩形 r1 和 r2 的高度：

```
r1.set('ly', H1);  
r2.set('ly', H2);
```

注：函数 `mphgetproperties` 可以用来获得属性列表和特征节点的值，详情参见[获取模型信息](#)。

7 键入以下命令绘制几何：

```
mphgeom(model);
```



还有第三种选项来定义几何特征的属性值，即将 MATLAB 变量与 COMSOL 模型对象中定义的参数或变量结合。为此需要将 MATLAB 变量转换成字符串。

最后练习如何将矩形 `r1` 和 `r2` 的高度分别修改为 `L0` 和 `L0-tbb`，其中 `L0` 是一个 MATLAB 变量。

8 定义变量 `L0`:

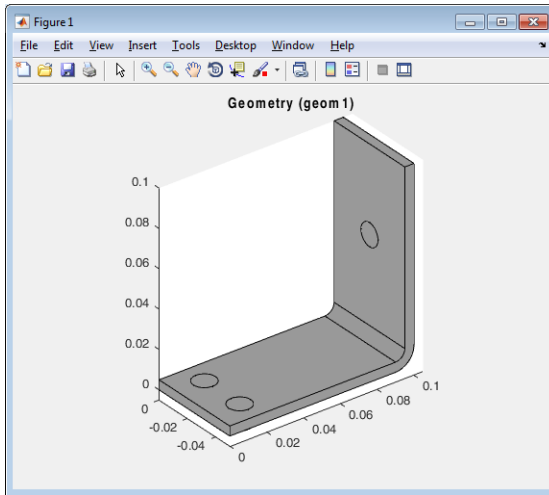
```
L0 = 0.1;
```

9 修改矩形 `r1` 和 `r2` 的 `ly` 属性:

```
r1.set('ly',L0);  
r2.set('ly',[num2str(L0) '-tbb']);
```

10 更新并绘制几何:

```
mphgeom(model);
```



用于 MATLAB® 的代码

```
model = mphload('busbar');  
L0 = 9e-2;  
model.param.set('L',L0);  
mphgeom(model)  
  
r1 = model.geom('geom1').feature('wp1').geom.feature('r1');  
r2 = model.geom('geom1').feature('wp1').geom.feature('r2');  
H1 = 0.2;  
H2 = 0.195;  
r1.set('ly', H1);  
r2.set('ly', H2);  
mphgeom(model);  
  
L0 = 0.1;  
r1.set('ly',L0);  
r2.set('ly',[num2str(L0) '-tbb']);  
mphgeom(model);
```

使用嵌套式循环示例

本例中，创建一个 MATLAB 函数求解不同长度和厚度的母线板在不同输入电势条件下的结果。该函数求解并返回母线板中的最大温度、生成的热量，以及电流等。模型每变化一次，函数会将相应的结果保存到一个文件，并将

模型对象保存成 MPH 文件。为了方便地识别 MPH 文件名称，文件名中包含了设置模型所用的不同参数值。

M 函数的输入变元是 busbar 模型对象，以及保存数据和模型的路径名称。

1 打开一个文本编辑器，例如，MATLAB 文本编辑器。

2 在新建的 M 文件的第一行，键入以下函数题头：

```
function modelParam(model,filepath)
```

函数首先创建一个文件，然后设置输出文件格式的题头。

3 在文本编辑器中，继续键入以下命令行：

```
filename = fullfile(filepath,'results.txt');  
fid=fopen(filename,'wt');  
fprintf(fid,'*** run parametric study ***\n');  
fprintf(fid,'L[m] | tbb[m] | Vtot[V] | ');  
fprintf(fid,'MaxT[K] | TotQ[W] | Current[A]\n');
```

4 当做很大规模的循环运算时，最好是禁用模型历史，防止循环运算时模型保存的历史信息太大，导致内存溢出。

键入以下命令行：

```
model.hist.disable;
```

5 现在让我们开始创建一个 for 循环，参数化设置参数 L 和相应的参数，键入命令：

```
for L = [9e-2 15e-2]  
model.param.set('L',L);
```

6 接下来继续创建第二个 for 循环，这一次参数化 tbb，修改它在模型对象中的值：

```
for tbb = [5e-3 10e-3]  
model.param.set('tbb',tbb);
```

7 定义最后一个 for 循环，参数化设置电势 Vtot：

```
for Vtot = [20e-3 40e-3]  
model.param.set('Vtot',Vtot);
```

8 每个参数对应的模型，将用到的参数值输出到指定的文件：

```
fprintf(fid,[num2str(L),' | ',num2str(tbb),...  
' | ',num2str(Vtot),' | ']);
```

9 添加以下命令行来求解模型：

```
model.sol('sol1').run;
```

10 以下几行计算结果：

```

MaxT = mphmax(model,'T',3,'selection',1);
TotQ = mphint2(model,'ht.Qtot',3,'selection',1);
Current = mphint2(model,'ec.normJ','surface','selection',43);

```

11 保存结果到输出文件:

```

fprintf(fid,[num2str(MaxT),' | ',num2str(TotQ),...
' | ',num2str(Current),' \n']);

```

12 命名和保存模型:

```

modelName = fullfile(filepath,['busbar_L=',num2str(L),...
'_tbb=',num2str(tbb),...
'_Vtot=',num2str(Vtot),'.mph']);
mphsave(model,modelName);

```

13 结束循环:

```

end
end
end

```

14 关闭输出文件:

```

fclose(fid);

```

现在文本编辑器中的脚本应该如下所示:

```

function modelParam(model,filepath)
filename = fullfile(filepath,'results.txt');
fid=fopen(filename,'wt');
fprintf(fid,'*** run parametric study ***\n');
fprintf(fid,'L[m] | tbb[m] | Vtot[V] | ');
fprintf(fid,'MaxT[K] | TotQ[W] | Current[A]\n');
model.hist.disable;
for L = [9e-2 15e-2]
    model.param.set('L',L);
    for tbb = [5e-3 10e-3]
        model.param.set('tbb',tbb);
        for Vtot = [20e-3 40e-3]
            model.param.set('Vtot',Vtot);
            fprintf(fid,[num2str(L),' | ',...
                num2str(tbb),' | ',...
                num2str(Vtot),' | ']);
            model.sol('sol1').run;
            MaxT = mphmax(model,'T',3,'selection',1);
            TotQ = mphint2(model,'ht.Qtot',3,'selection',1);
            Current = mphint2(model,'ec.normJ','surface',...
                'selection',43);
            fprintf(fid,[num2str(MaxT),' | ',...
                num2str(TotQ),' | ',...
                num2str(Current),' \n']);
            modelName = fullfile(filepath,...
                ['busbar_L=',num2str(L),...
                '_tbb=',num2str(tbb),...
                '_Vtot=',num2str(Vtot),'.mph']);

```

```

        mphsave(model,modelName);
    end
end
end
fclose(fid);

```

15 保存 M 文件到一个 MATLAB 可以调用到脚本的目录，文件名为 **modelParam.m**。

16 如果 COMSOL MATLAB 尚未运行，将其打开，并从案例库加载母线板模型：

```
model = mphload('busbar');
```

17 运行编好的函数，注意将下面命令行的 'filepath' 修改为您选中的路径名称：

```
modelParam(model,'filepath')
```

18 用文本编辑器打开 **results.txt**，检查结果：

```

*** run parametric study ***
L[m] | tbb[m] | Vtot[V] | MaxT[K] | TotQ[W] | Current[A]
0.09 | 0.005 | 0.02 | 323.6905 | 0.25241 | 164.4663
0.09 | 0.005 | 0.04 | 415.4685 | 1.0096 | 328.9325
0.09 | 0.01 | 0.02 | 308.7039 | 0.04845 | 96.8226
0.09 | 0.01 | 0.04 | 355.4455 | 0.1938 | 193.6452
0.15 | 0.005 | 0.02 | 316.1346 | 0.33729 | 159.0172
0.15 | 0.005 | 0.04 | 385.2061 | 1.3491 | 318.0344
0.15 | 0.01 | 0.02 | 305.2178 | 0.065732 | 95.8105
0.15 | 0.01 | 0.04 | 341.4212 | 0.26293 | 191.6209

```

尽管本例是复制 COMSOL Desktop® 中可供使用的函数，但是您还可以采用类似的技术将一个 COMSOL 模型耦合到定制的优化脚本。

使用外部 MATLAB® 函数

使用 LiveLink™ for MATLAB® 的另一个优势是可以在 COMSOL Desktop® 中直接调用 MATLAB 函数。您可以用函数在 COMSOL® 模型对象中定义属性，例如材料定义或物理场等。COMSOL 自动检测外部函数，并启动 MATLAB 引擎来计算它，然后将得到的结果应用到相应的属性。

如果要使用这个功能，只需要启动 COMSOL Desktop，如果有需要，MATLAB 会自动在后台启动。

注：运行模型对象的 MATLAB 函数，先要在 COMSOL 的安全设置中启用外部函数调用。在**首选项 > 安全性**中进行设置，选择**允许外部程序和库**。

注：在 Linux 操作系统，需要在启动 COMSOL Desktop 时指定 MATLAB 的根目录 MLROOT，并载入 gcc 库：`comsol -mlroot MLROOT -forcegcc`。

本节中的案例描述如何在 COMSOL 模型中使用外部 MATLAB 函数。模型计算了由外部热流产生的温度分布，其中的材料热导率和外部热通量都由 MATLAB 函数来定义。

创建 MATLAB® 函数

假设案例中的材料具有随机热导率，为了定义这种热导率属性，创建一个 MATLAB 函数，以 x 轴坐标作为输入变元，使得热导率围绕定值 $400 \text{ W}/(\text{m}\cdot\text{K})$ 在范围 $200 \text{ W}/(\text{m}\cdot\text{K})$ 内随机变化。

定义一个 MATLAB 函数作为外热流，其中包含以下变元：

- x 和 y ，坐标。
- $x0$ 和 $y0$ ，热源原点。
- $Q0$ ，最大热源。
- $scale$ ，周期性变化的外热流比例因子。

x 和 y 是模型的因变量， $x0$ 、 $y0$ 、 $Q0$ ，以及 $scale$ 在本例中定义为常数值。

这些函数使用时必须采用特定的结构。为使 COMSOL 到 MATLAB 的调用尽可能高效，变元以块的形式传递到 MATLAB。这意味着对于这两个函数，所有输

入变元都是矢量，且函数返回的矢量长度必须与输入变元相同。因此强烈建议在 MATLAB 命令行中测试这些函数，对运用的矢量变元使用合理值。

1 打开文本编辑器，例如，MATLAB 文本编辑器。

2 新建文件，输入：

```
function out=conductivity(x)
out = 400+5*randn(size(x));
```

3 保存为 conductivity.m。

4 新建文件，输入：

```
function out = heatflux(x,y,x0,y0,Q0,scale)
radius = sqrt((x-x0).^2+(y-y0).^2);
out = Q0/5+Q0/2.*sin(scale*pi.*radius)./(scale*pi.*radius);
```


5 在 MATLAB 的用户文档文件夹中保存文件为 heatflux.m。将文件保存到目录 /Documents/MATLAB，确保 COMSOL 调用 MATLAB 时能够找到函数。



模型向导

注：本操作指南用于 Windows 操作平台上的用户界面，稍做修改也可以用于 Linux 和 Mac。


1 双击桌面上的 COMSOL Multiphysics 快捷图标。


2 通过**模型向导**  创建一个模型。

3 在**选择空间维度**单击**三维** .

4 在**选择模型树**的**传热** ，下面选择**固体传热 (ht)** .


5 单击**添加**，然后单击**研究** .

6 在**选择研究窗口**选择**预设研究 > 稳态** .

7 单击**完成** .

在模型中定义外部 MATLAB® 函数



在模型中定义一个 MATLAB 函数，使得 COMSOL 可以将它识别为外部函数，并调用 MATLAB 引擎计算。

- 1 在**主屏幕**工具栏，单击**函数 f(x)**，在**全局**中选择 **MATLAB** 。如果是 Linux 和 Mac，**主屏幕**工具栏相当于是 Desktop 顶上的一组快捷工具。
- 2 在**设置**页面的**函数**下面，在**函数名称**编辑框中输入 conductivity，**变元**编辑框中输入 x。




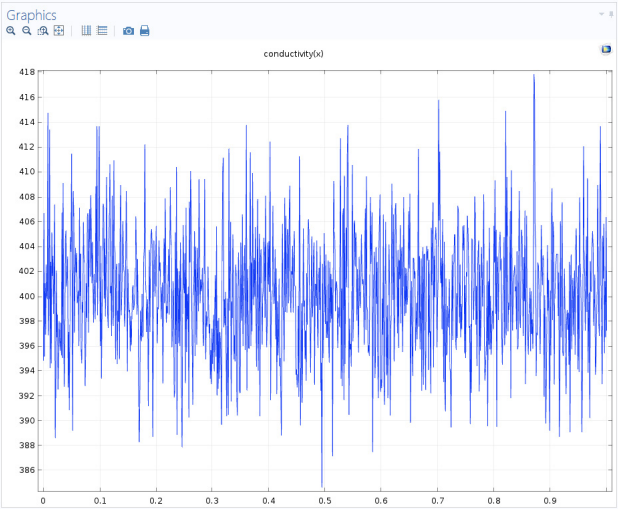
通过在这里的函数定义，COMSOL 知道 conductivity 是一个 MATLAB 函数，COMSOL 会在必要时自动启动 MATLAB 引擎计算函数。

注：将 COMSOL 模型与 MATLAB 外部函数保存在同一个文件夹中。或者采用其他处理方法，在 MATLAB 配置中添加路径，或者将相关路径名称添加到环境变量 COMSOL_MATLAB_PATH。

- 3 在 COMSOL Desktop，从**文件**菜单（Windows 用户）或**选项**菜单（Mac 和 Linux 用户），单击**另存为** ，选择**另存为 MPH 文件** ，并浏览到保存 conductivity.m 和 heatflux.m 的文件夹。

您可以通过定义输入变元的绘图限制来显示函数结果。

- 4 展开**绘图参数**栏，在表单中输入 0 作为**下限**，1 作为**上限**。
- 5 单击**绘制**按钮 。



接下来您可以定义第二个 MATLAB 函数，用来计算外部热流条件。

6 重复上面的步骤，在函数名称编辑框中输入 `heatflux`，设置变元为 `x,y,x0,y0,Q0,scale`。

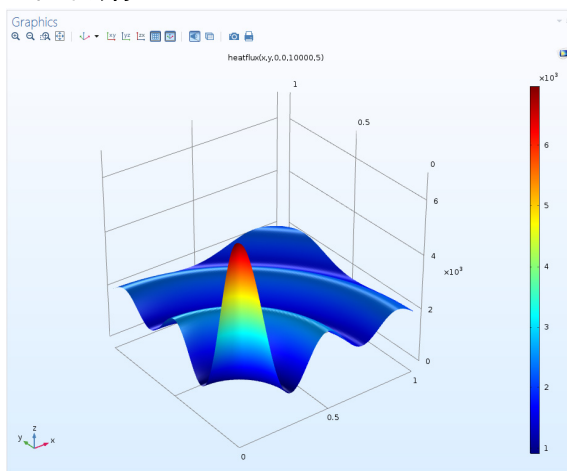


7 按照下表设置绘图参数，显示函数结果：

下限	上限
0	1
0	1
0	0

下限	上限
0	0
1e4	1e4
5	5

8 单击绘制按钮 。






在继续进行模型设置前，您需要手动指定函数对所有函数变元的导数，否则求解器会返回一个警告信息。对于本案例，函数变元与因变量温度无关，所以这个数值问题属于线性。因此，可以将所有变元的导数设为 0。

9 展开派生值栏，然后如下表所示填写：

函数名称	变元	偏导数
conductivity	x	0
heatflux	x	0
heatflux	y	0
heatflux	x0	0
heatflux	y0	0
heatflux	Q0	0
heatflux	scale	0

注：对于非线性问题，有必要指定各个偏导。它可以是表达式，或者另一个 MATLAB 函数。






几何和材料定义

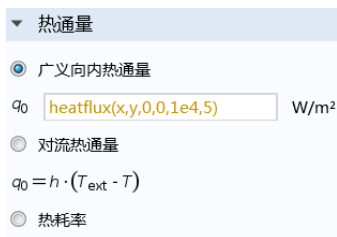
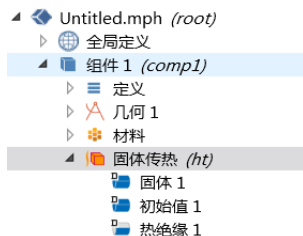
- 1 在几何工具栏，单击长方体 。
- 2 在长方体设置窗口，单击构建所有对象 。
- 3 在屏幕工具栏，单击新材料 。
- 4 右键单击材料节点，选择新材料。在材料目录下的设置窗口的值列中，设置热导率的表达式为 `conductivity(x)`，密度为 `8e3`，常压热容的表达式为 `2e3`。

材料属性明细					
属性	变量	值	单位	属性组	
<input checked="" type="checkbox"/> 密度	rho	8e3	kg/m³	基本	
<input checked="" type="checkbox"/> 恒压热容	Cp	2e3	J/(kg...	基本	
<input checked="" type="checkbox"/> 导热系数	k_iso...	conductivit...	W/(m...	基本	

注：表达式 `conductivity(x)` 显示为橙色，因为 MATLAB 函数没有量纲。


固体传热

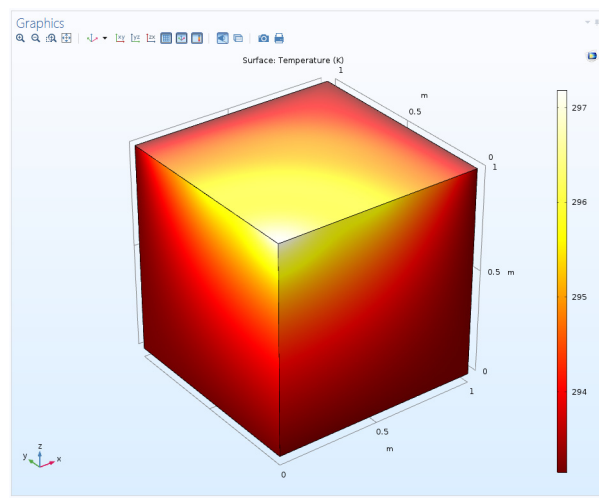
- 1 在模型开发器的组件 1 下面，单击固体传热 。
- 2 在物理场工具栏，单击边界 ，选择温度 。
- 3 在温度设置页面，选择边界 3、5 和 6（您也可以使用粘贴选择 ，直接输入选择）。
- 4 接下来添加边界热流，在物理场工具栏，单击边界，选择热通量 。
- 5 在热通量设置页面，选择边界 4，设置广义向内热通量表达式为 `heatflux(x,y,0,0,1e4,5)`。





计算并绘制结果


设置好模型后，便可计算解，在此过程中会自动生成默认网格。

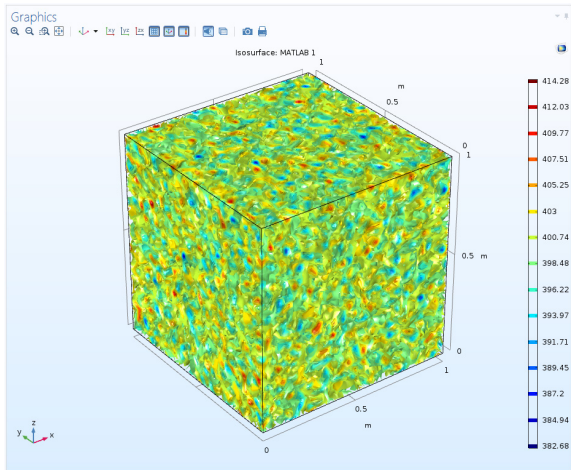
- 1 在研究工具栏，单击计算 。
- 2 单击温度 (ht) 节点，显示几何表面的温度场。



- 1 添加一个新三维绘图组，在几何上显示材料的热导率。在结果标签，选择三维绘图组 ，会显示一个包含三维绘图组绘图工具的工具栏。
- 2 在三维绘图组 3 工具栏上，单击等值面 。



- 3 在模型开发器的三维绘图组 3 节点下，选择等值面 1 节点。
- 4 在相关的设置页面，表达式编辑框中键入 conductivity(x)。
- 5 在总水平数编辑框中输入 15。
- 6 单击绘制按钮 。



注：需要在 MATLAB 中再次运行 conductivity 函数，生成上面这个图。因为函数中用到了随机分布，所以上面的图并不一定会与求解时用到的热导率完全相同。