

实验室做网格处理方面的算法，写界面很麻烦，所以有了利用 **maya** 和 **matlab** 进行数据连通的念头，于是有了这个插件。

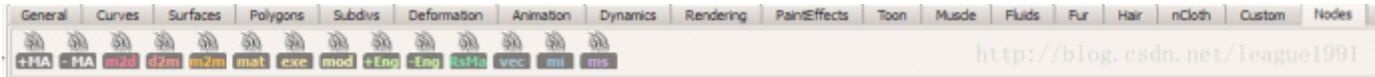
这个插件可以把 **maya** 的网格数据导入 **matlab** 之中，完成计算之后重新返回 **maya**。适合平时快速做一些网格数据的可视化工作，以及一些小试验。

选择 **maya** 而不是 **max** 的原因是，**maya** 的数据是用节点组织的，可以很方便地组合不同的节点构造出新的功能，而在 **max** 里面一切都是隐藏的，要实现新的功能十分不方便。

如果需要代码，可以在这里下载 <https://github.com/league1991/Maya2Matlab>

下面以一个简单的例子演示插件的用法。在这个例子中，我们给模型加噪声，就像 **max** 的噪波修改器一样。

1.配置好插件之后，打开 **maya**，可以看到插件工具栏：



下面介绍一些按钮的功能：

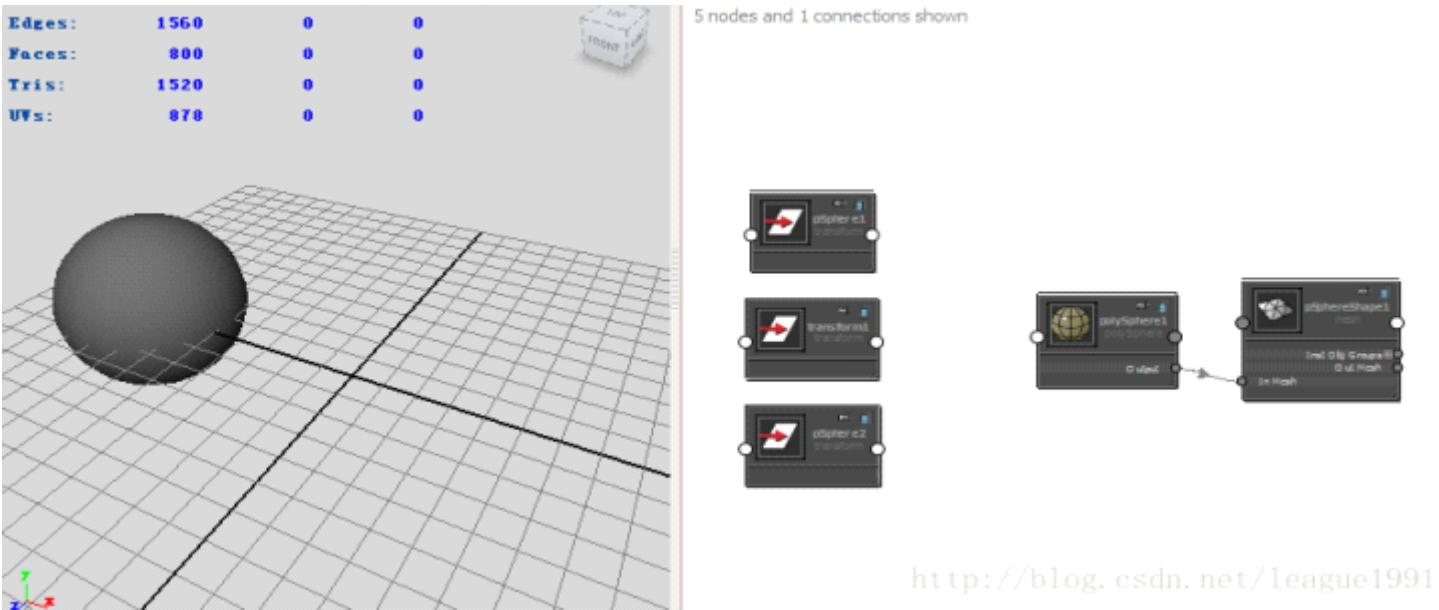
+/-MA	加载、卸载插件
+/-Eng	打开/关闭 matlab 引擎
RsMa	彻底刷新 matlab 引擎的按钮，当你修改了某个 m 文件，想 matlab 重新读取你新的代码，就要先刷新 matlab 引擎
m2d/d2m/m2m	数据转换节点按钮，为了统一处理各种数据，实现了一种自定义的矩阵数据类型 matlabData ， maya 的各种数据都统一转成 matlabData 然后才能在插件节点里面传输。这 3 个节点的意思分别是 maya 数据-> matlabData matlabData -> maya 数据 maya 网格-> matlabData 。
mat	matlab 节点，允许用户输入 matlab 脚本
mi	动画序列插值节点
ms	点边面选择节点，提取选中的顶点、边、面序号
mod	修改网格节点
exe	与外部 exe 程序通过 socket 通信

2. 为了方便编辑，建议把 **maya** 界面视图配置成三块：透视图+节点视图+属性编辑器，如图：



这样可以同时看到节点结构和计算结果

3. 好，现在开始教程，首先我们创建或导入一个模型，这里简单起见，创建了一个球。之后的场景大概是这样：



左边的透视图多了一个球，右边的节点视图多了三个节点。

节点的方框里面有两行字，第一行表示节点的名字，第二行表示节点的类型。

熟悉面向对象编程的人可以发现，一个节点就好像一个类的实例，名字就相当于这个实例的变量名，而节点类型自然就相当于类名了。

对于这三个节点，pSphere1是一个变换节点（类型为 transform），记录了球的位置、旋转这些信息。polySphere1是一个球生成器（类型为 polySphere），根据各种参数，例如半径，分段数等等生成一个球网格，而 pSphereShape1（类型为 mesh）就是一个放着网格数据的容器。

图中的箭头表示什么呢？对程序员来说，可以理解成表示“赋值”，相当于一句话：

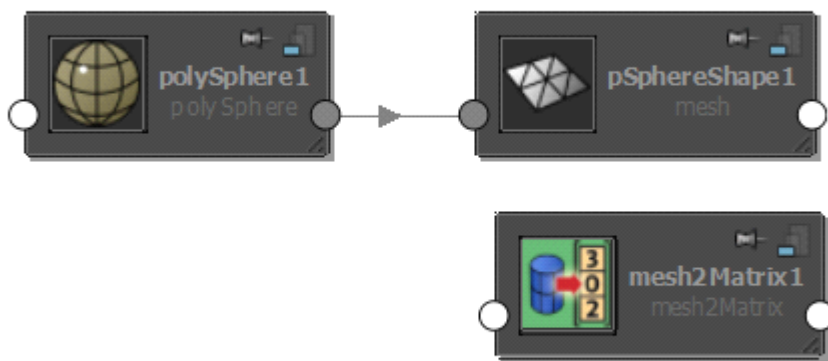
`pSphereShape1.InMesh = polySphere1.Output;`

`inMesh` 和 `output` 都是各自节点类型的一个数据成员。通常列在右边的是作为输出，在左边的作为输入。

也可以理解成数据的流向。也就是说，把球生成器的 `output` 变量的数据（就是一个球网格）放到 `pSphereShape1` 的 `inMesh` 变量里面。

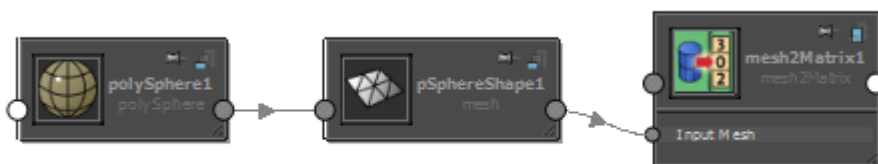
3. 现在我们已经有了原数据，下一步要做的是：把它转换成 `matlabMatrix` 格式。

在工具架上点一下 `m2m`, 创建一个 `mesh2Matrix` 类型的节点：



<http://blog.csdn.net/league1991>

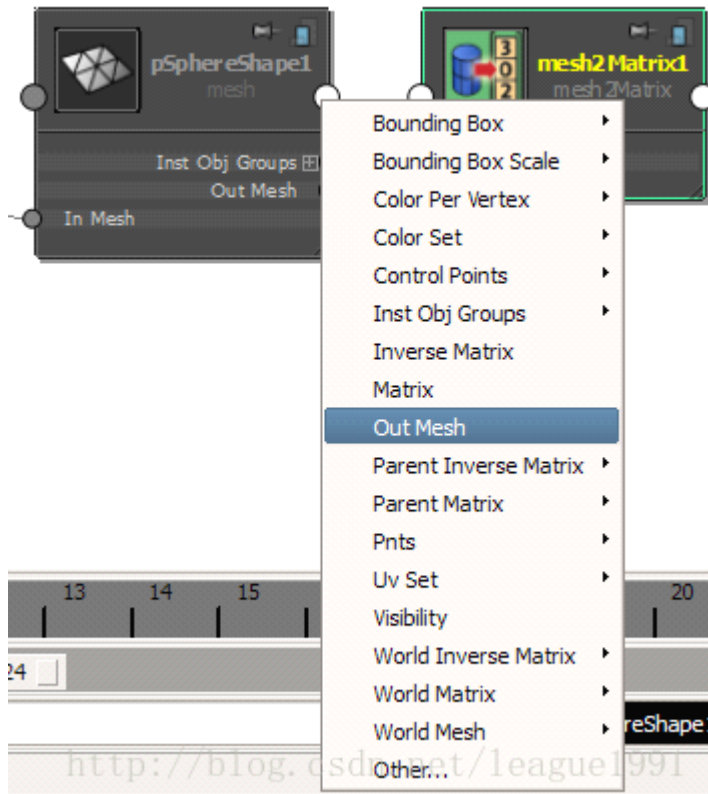
然后把 `pSphereShape1` 的 `out mesh` 连到 `mesh2Matrix` 的 `inputMesh` 里面：



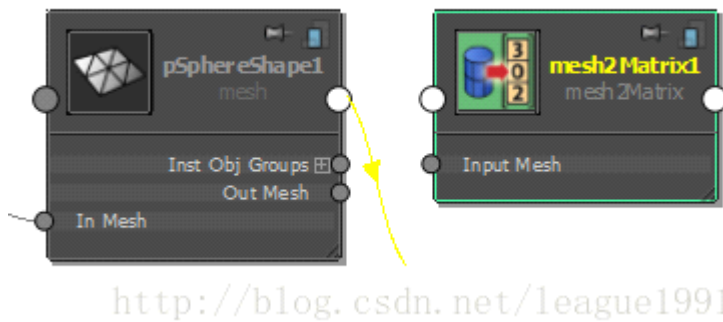
<http://blog.csdn.net/league1991>

怎么连？请看下面几幅图：

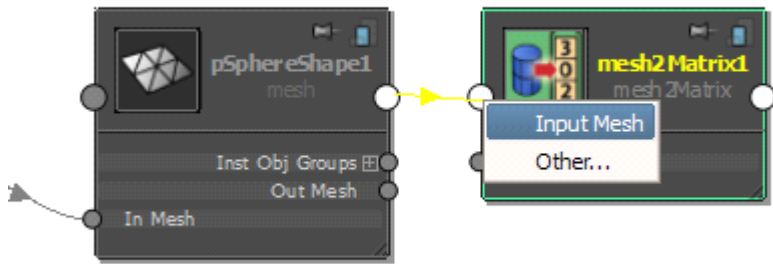
(1) 在 `pSphereShape1` 的右边白点上点一下，出来一个菜单



(2) 选择 OutMesh 之后，拉出来一个黄色箭头的线。

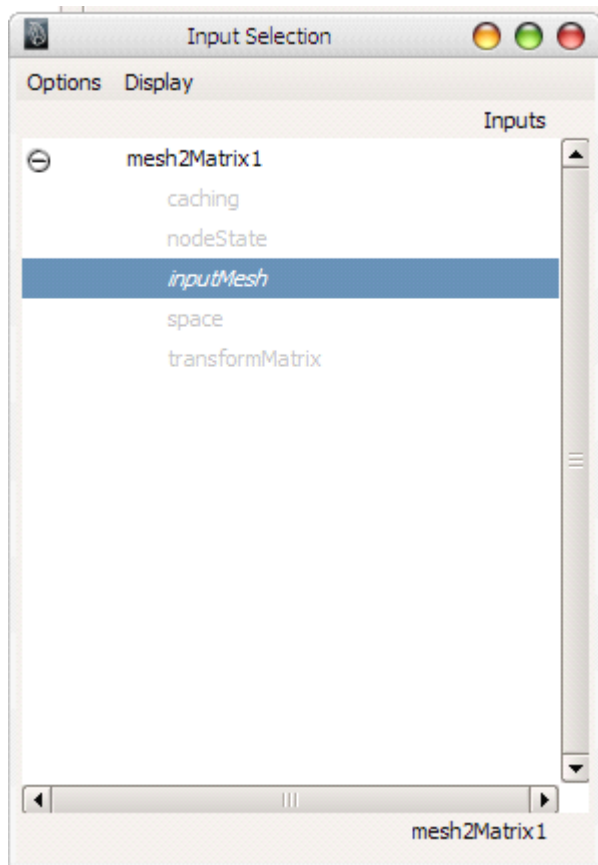


(3) 在 mesh2Matrix 的左边白点上点一下，出来一个菜单，这时候可以直接选 inputmesh。有时候你要的变量没有出现，就要点 other..



<http://blog.csdn.net/league1991>

(4) 点 **other** 之后弹出来一个菜单，下面是成员变量（或者说是节点的各种数据）的列表。点一下 **inputMesh**，连接成功。



4. **mesh2Matrix** 负责数据转化，现在可以获得转化之后的 **matlabMatrix** 矩阵了。但是它在哪里呢？点开 **mesh2Matrix1** 看看：

出来很多个 **Matrix** 结尾的名字，下面逐个解释：

Adjacency Matrix 顶点邻接矩阵，是一个稀疏矩阵，矩阵大小为点数 \times 点数，若顶点 i 与顶点 j 相邻 对应元素非0

Avg Normal Matrix , **Bitangent Matrix** , **Tangent Matrix** 法线，切线，副切线矩阵，大小为点数 $\times 3$

Min/Max Curvature Direction 最小/最大曲率方向，大小为点数 $\times 3$

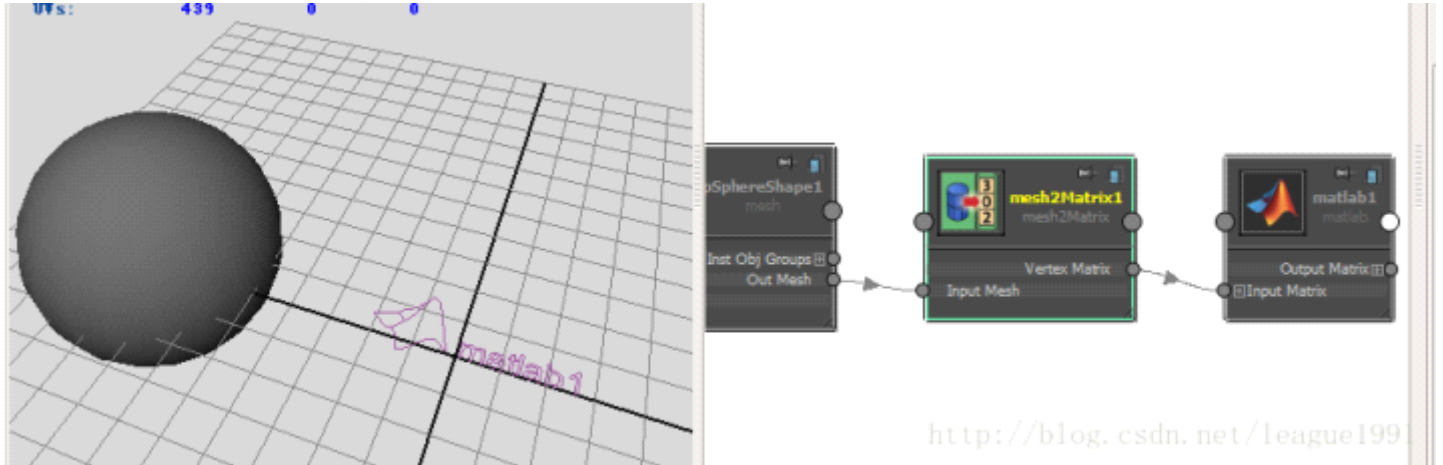
Min/Max Curvature 最小/最大曲率大小，大小为点数 $\times 3$

Vertex Matrix 顶点坐标矩阵，大小为点数 $\times 3$

Vertex Edge/Face Incidence Matrix 顶点和边/面的关联矩阵，若顶点与某条边/某个面邻接，对应元素非0

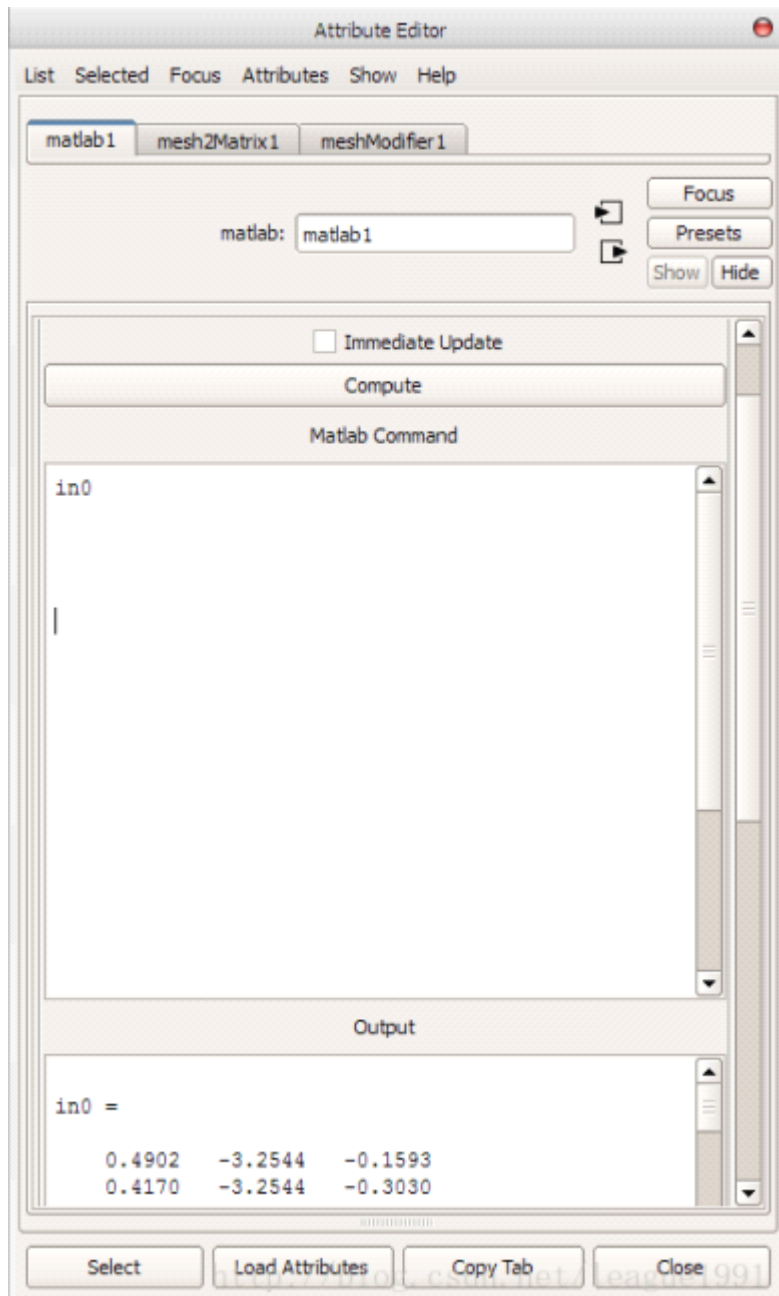
现在我们需要的是顶点矩阵。

5. 把顶点数据导入 **matlab** ，点一下 **mat** 按钮，创建一个 **matlab** 节点，然后把 **vertex matrix** 连到 **matlab** 节点的 **inputMatrix[0]**，如图：



可以发现场景多了一个 **matlab** 图标，可以通过点击这个图标直接选定 **matlab** 对象。

为了测试数据是否连通，在属性编辑器输入 **in0**，点一下 **compute**，可以看到有输出



在这个窗口里面，**Matlab Command**表示你要输入的 **matlab** 命令，**in0**就是对应 **inputMatrix[0]**的 **matlab** 变量，点 **compute** 之后，这个命令会被发送到 **matlab** 引擎执行，结果显示在 **output** 窗口。如果勾选了 **immediate update**，则每次节点被求值的时候都会调用 **matlab** 引擎，这样可以在调参数的时候实时预览结果。

也可以在 **matlab** 引擎输入 **in0**，结果是一样的

MATLAB Command Window

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

>> in0

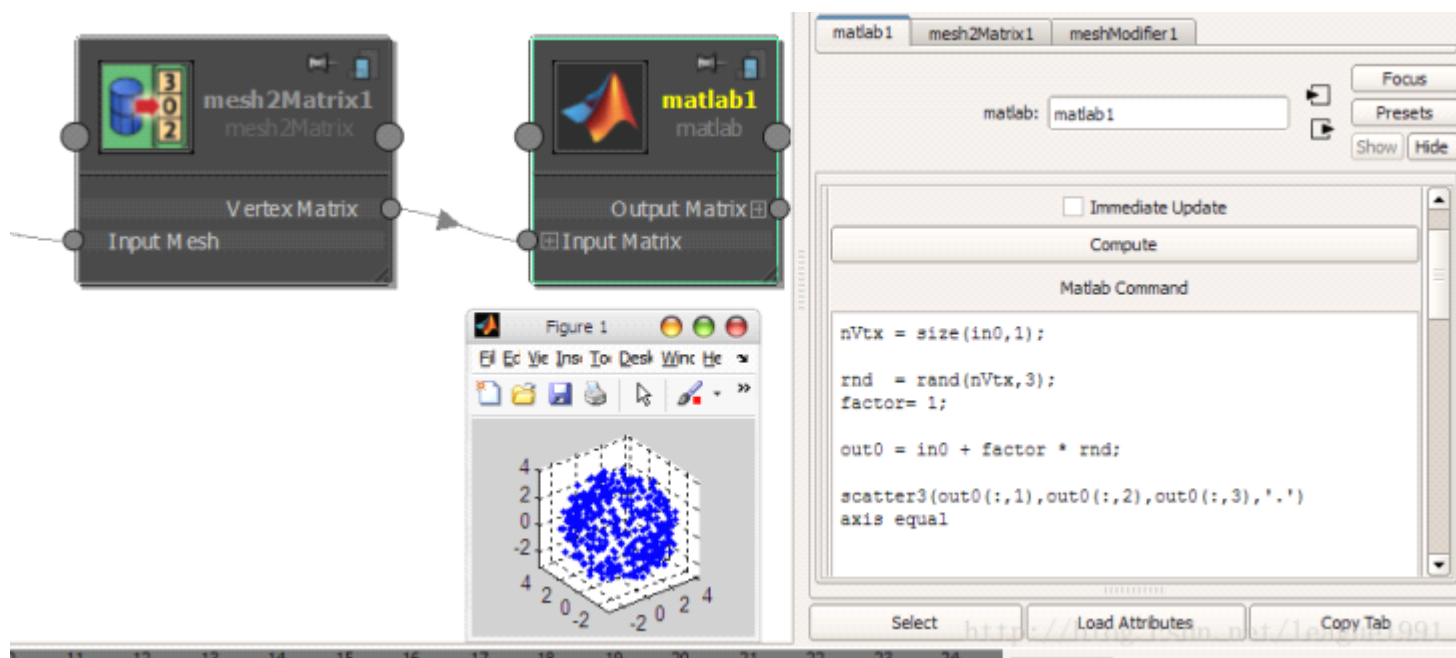
in0 =

0.4902	-3.2544	-0.1593
0.4170	-3.2544	-0.3030
0.3030	-3.2544	-0.4170
0.1593	-3.2544	-0.4902
0	-3.2544	-0.5155
-0.1593	-3.2544	-0.4902
-0.3030	-3.2544	-0.4170
-0.4170	-3.2544	-0.3030
-0.4902	-3.2544	-0.1593
-0.5155	-3.2544	0
-0.4902	-3.2544	0.1593
-0.4170	-3.2544	0.3030
-0.3030	-3.2544	0.4170
-0.1593	-3.2544	0.4902
-0.0000	-3.2544	0.5155
0.1593	-3.2544	0.4902
0.3030	-3.2544	0.4170

<http://blog.csdn.net/league1991>

6. 现在顶点数据已经被读出，我们利用 matlab 加入噪声。

输入 matlab 命令，按 compute 之后的结果是这样的：



首先根据矩阵大小获得点数，存在 nVtx 变量里面

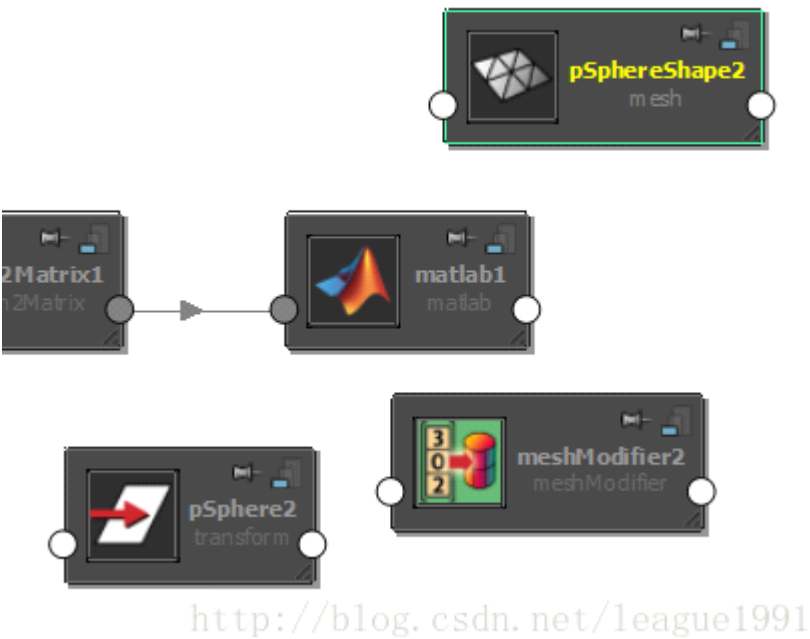
然后新建一个随机数矩阵 `rnd`

最后两者相加，结果存在 `out0`变量，这个变量对于 `outputMatrix[0]`

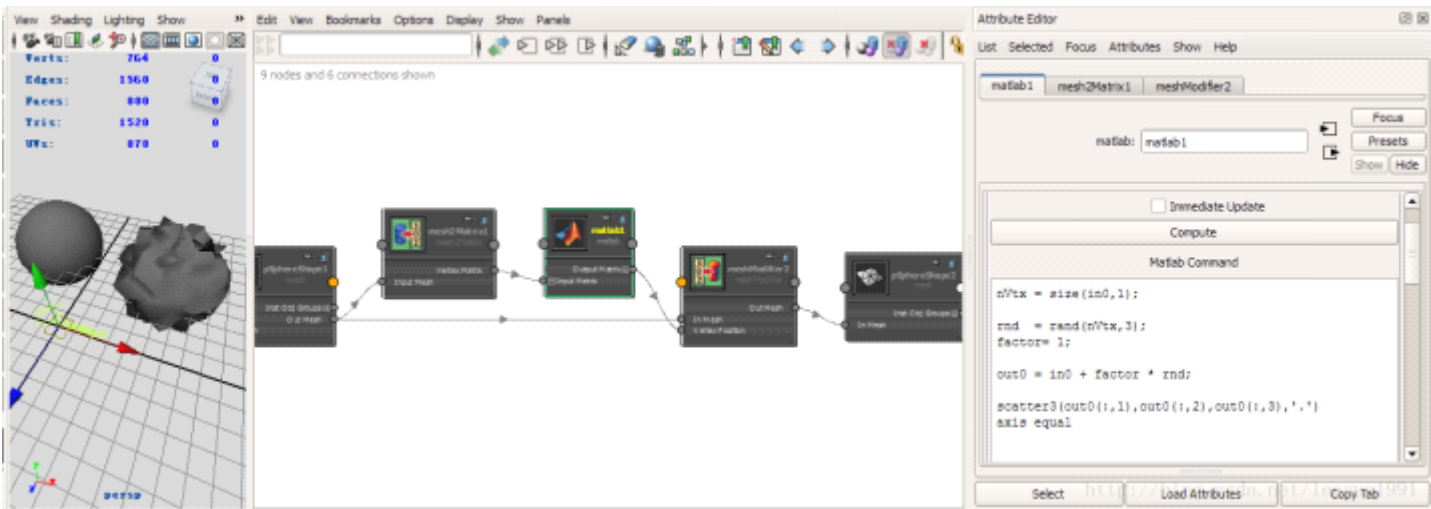
`scatter3`是一个画散点图指令，所以命令执行完就会显示一个 `matlab` 窗口。

7.现在已经有了扰动的结果，下一步是重建网格。

创建一个 `meshmodifier` 节点，同时把原网格复制一份，可以看出多了三个节点。



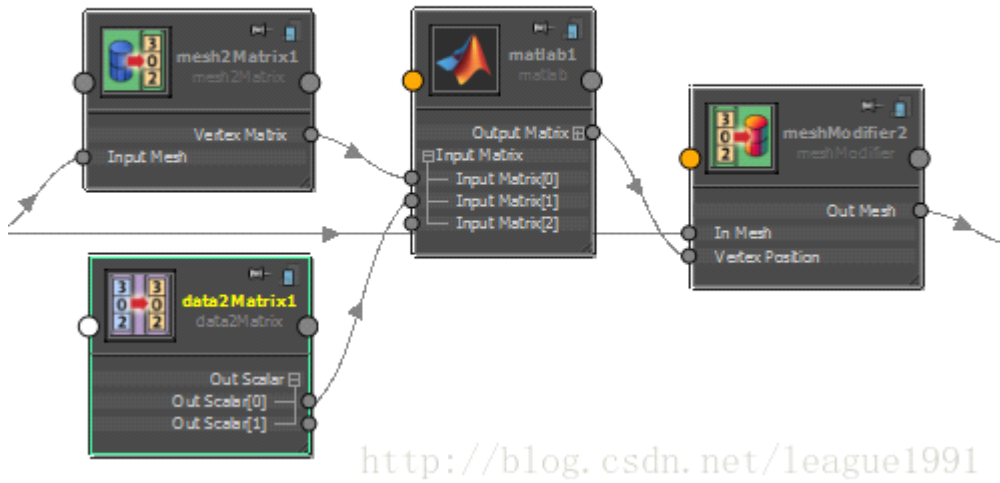
8. 按下图连接节点，连了之后重新按 `Compute` 计算一次，可以发现有了一个扰动的网格



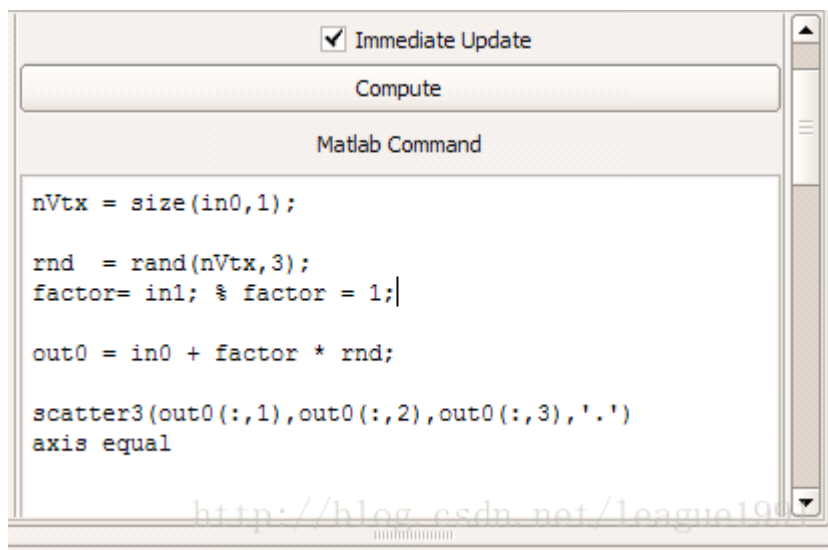
值得注意的是原网格的 `outMesh` 变量要连在 `meshModifier` 的 `inMesh` 之中，因为 `meshModifier` 需要原网格的拓扑结构，结合新的顶

点位置来构建新网格。

9. 若要实时控制扰动程度，可以添加一个 **data2Matrix** 节点，连到 **inputMatrix[1]**的地方



同时 **matlab** 代码改为如下，**factor** 接受 **in1**的输入，勾选 **immediate update**



10. 拖动 **data2Matrix** 的 **current scalar value** 的时候，噪声大小会实时变化，如图



教程完毕。