



SimCube. Automation 入门手册

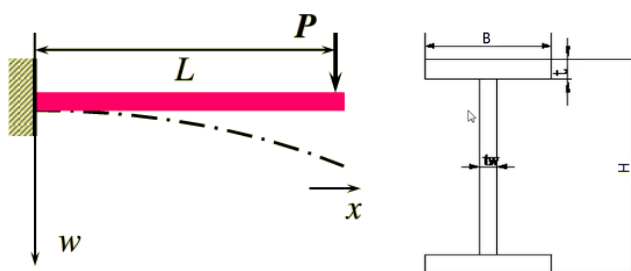
Version 1.0

目 录

1.	工程用例	1
1.1	自动化脚本创建模型	1
1.1.1	可视化创建模型	1
1.1.1.1	启动	1
1.1.1.2	添加仿函	2
1.1.1.3	添加参数	5
1.1.1.4	运行模型	8
1.1.2	编写 JavaScript 脚本创建模型	9
1.1.2.1	编写 JavaScript 代码	9
1.1.2.2	运行模型	11
1.2	应用创建模型	12
1.2.1	可视化创建模型	12
1.2.1.1	启动	12
1.2.1.2	添加仿函	13
1.2.1.3	添加文件参数	16
1.2.1.4	提取文件中的参数	18
1.2.1.5	运行模型	22
1.2.2	编写 JavaScript 脚本创建模型	24
1.2.2.1	编写 JavaScript 代码	24
1.2.2.2	提取文件中的参数	25
1.2.2.3	运行模型	27

1. 工程用例

工程用例背景：悬臂梁长度为 $L = 100\text{mm}$ ，一端受力 $P = 100\text{N}$ ，梁的截面为工字形，如下图所示，宽度 $B = 20\text{mm}$ ，高度 $H = 25\text{mm}$ ，腹板厚度 $t_w = 4\text{mm}$ ，翼缘厚度 $t = 4\text{mm}$ ，设弹性模量 $E = 210000\text{MPa}$ ，计算工字梁的最大位移和挠度。



根据《材料力学》中的公式，可知

$$\text{最大位移 } disp = \frac{12 * P * L^3}{3 * E * (B * H^3 - (B - t_w) * (H - 2 * t)^3)}$$

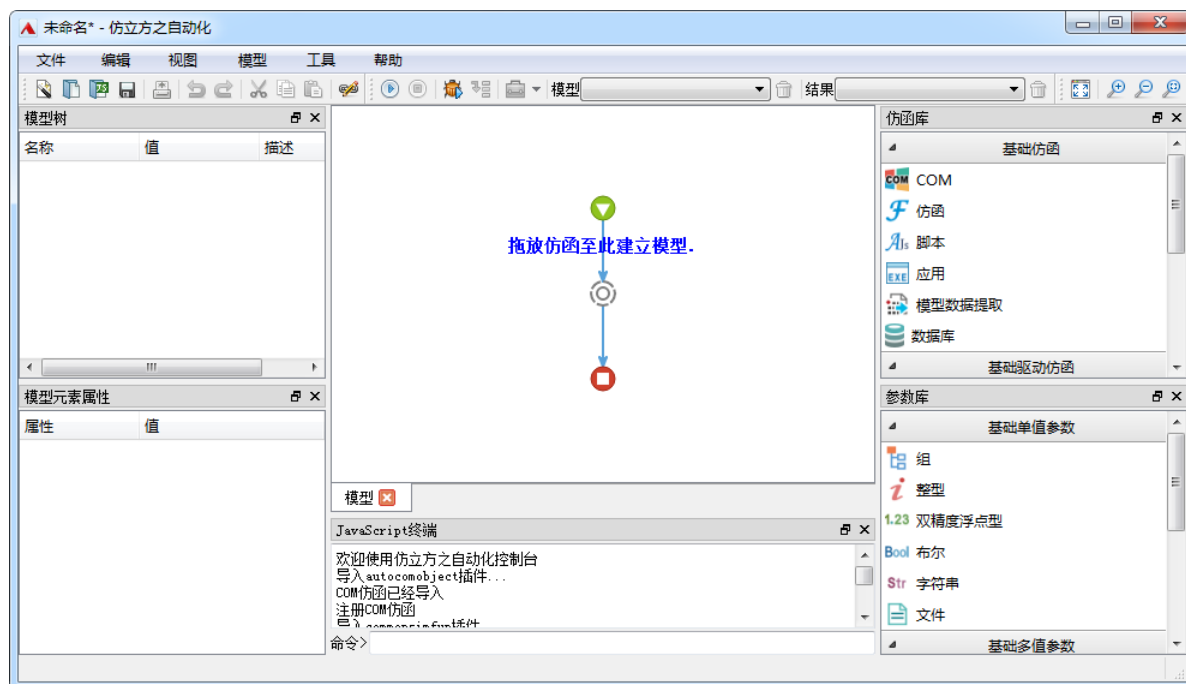
$$\text{挠度 } angle = \frac{12 * P * L^2}{2 * E * (B * H^3 - (B - t_w) * (H - 2 * t)^3)}$$

1.1 自动化脚本创建模型

1.1.1 可视化创建模型

1.1.1.1 启动

启动 Automation 程序，界面如下图所示。

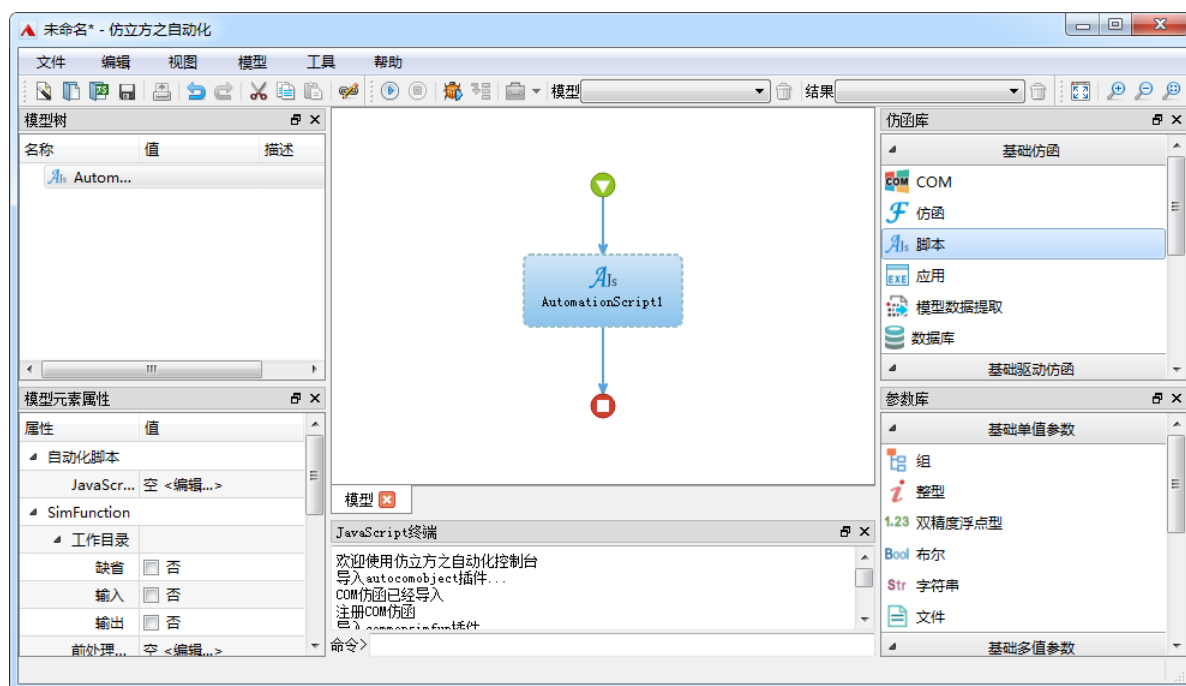


1.1.1.2 添加仿函

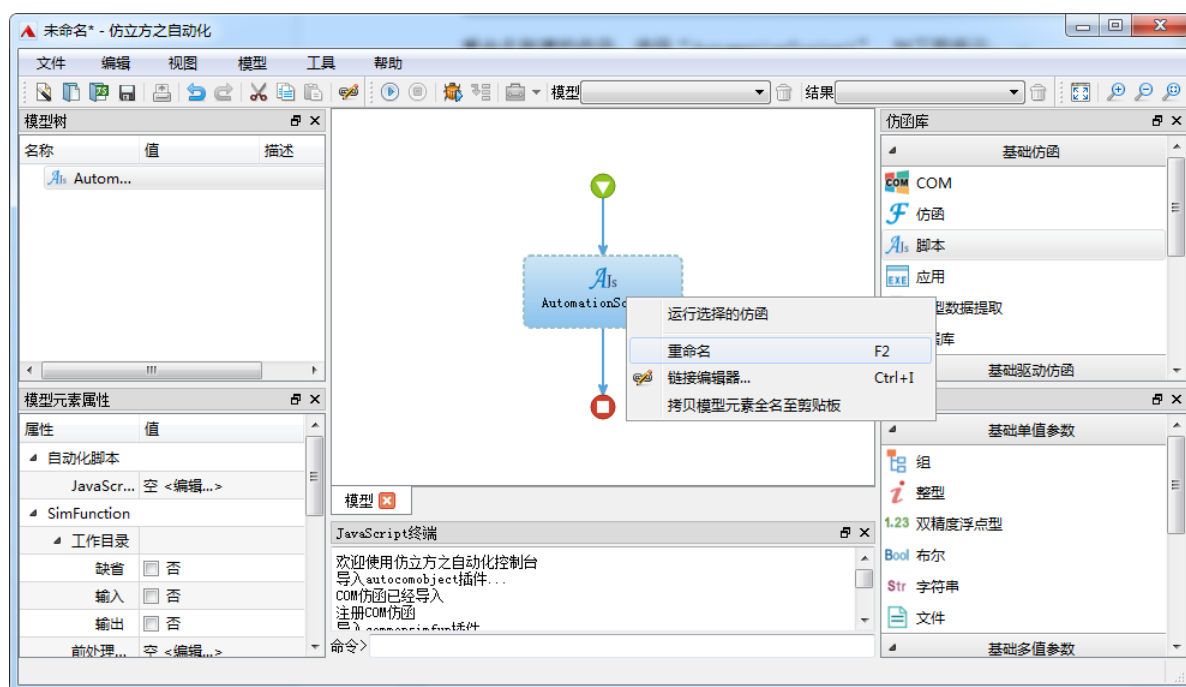
拖动“自动化脚本”仿函到工作区，拖动的过程如下，在工作区中心位置有一个中心圆环，中心圆环表示可以添加新的仿函。拖拉“基础驱动仿函”库中的“自动化脚本”驱动仿函到中心圆环，当拖拉至中心圆环附近时，中心圆环变为方形并高亮显示，如下图所示。



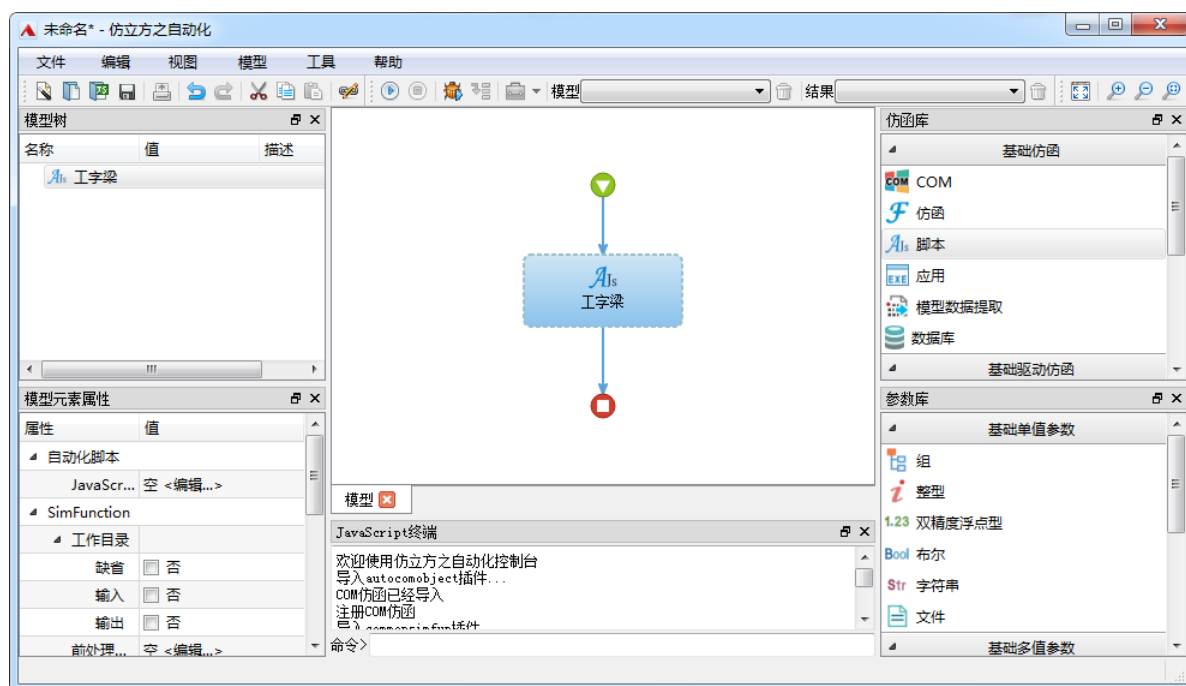
此时松开鼠标左键，“自动化脚本”驱动仿函被添加至对应的位置上，如下图所示。



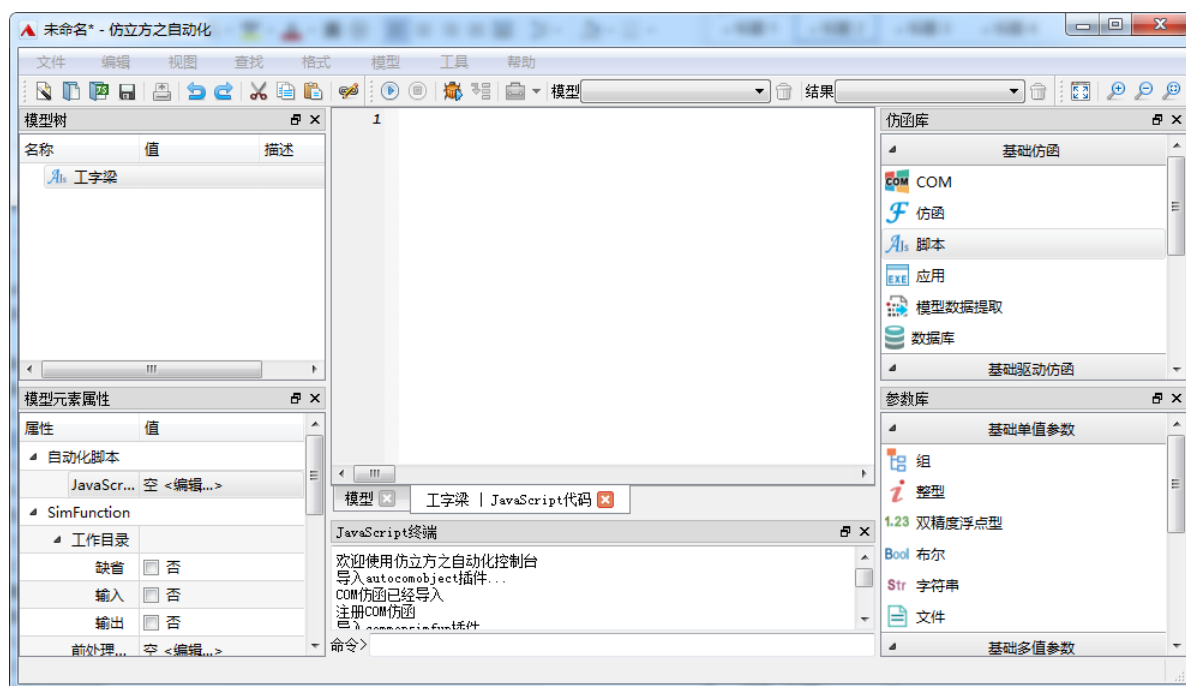
重命名新建的仿函数，选择“AutomationScript1”，如下图所示。



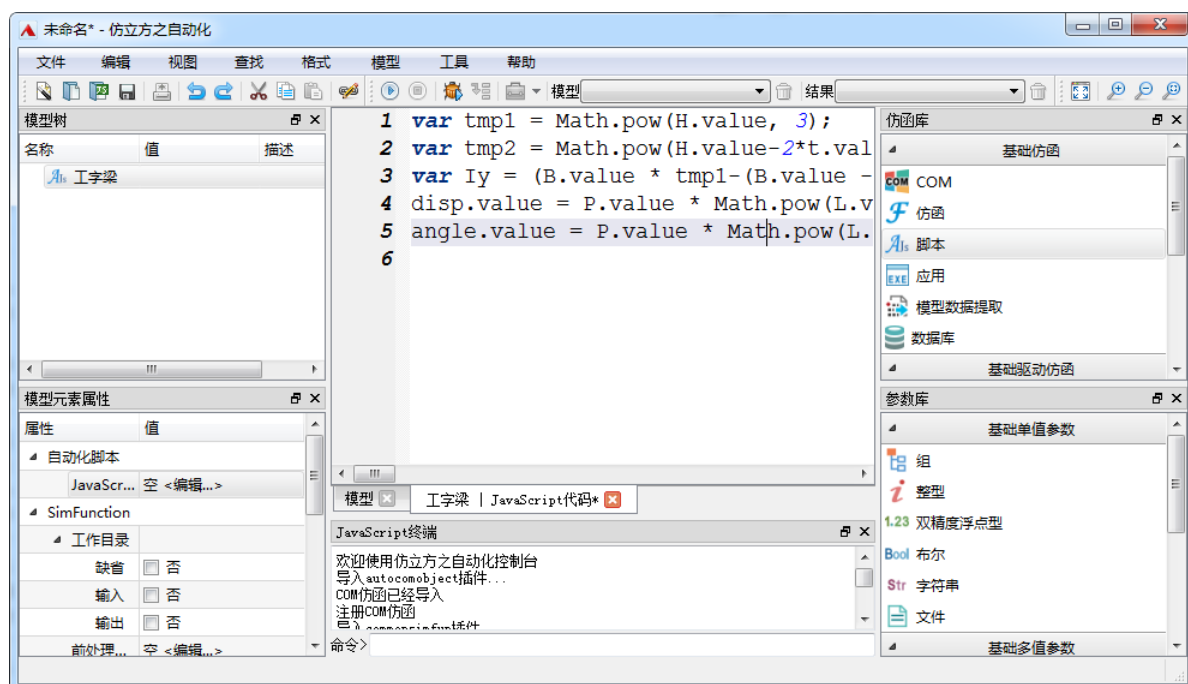
在右键菜单中点击“重命名”，在弹出的对话框中输入新的名称，这里输入“工字梁”，如下图所示。



在模型元素属性窗口中，点击“自动化脚本 | JavaScript 代码”，点击后会显示出 JavaScript 代码编辑窗口，如下图所示。



在 JavaScript 代码编辑窗口输入代码，内容如下图所示。



代码如下所示。

```

var tmp1 = Math.pow(self.H.value, 3);
var tmp2 = Math.pow(self.H.value-2*self.t.value, 3);
var Iy = (self.B.value * tmp1-(self.B.value - self.tw.value) * tmp2)/12;
self.disp.value = self.P.value * Math.pow(self.L.value,3)/(3 * self.E.value * Iy);
self.angle.value = self.P.value * Math.pow(self.L.value,2)/(2 * self.E.value * Iy);

```

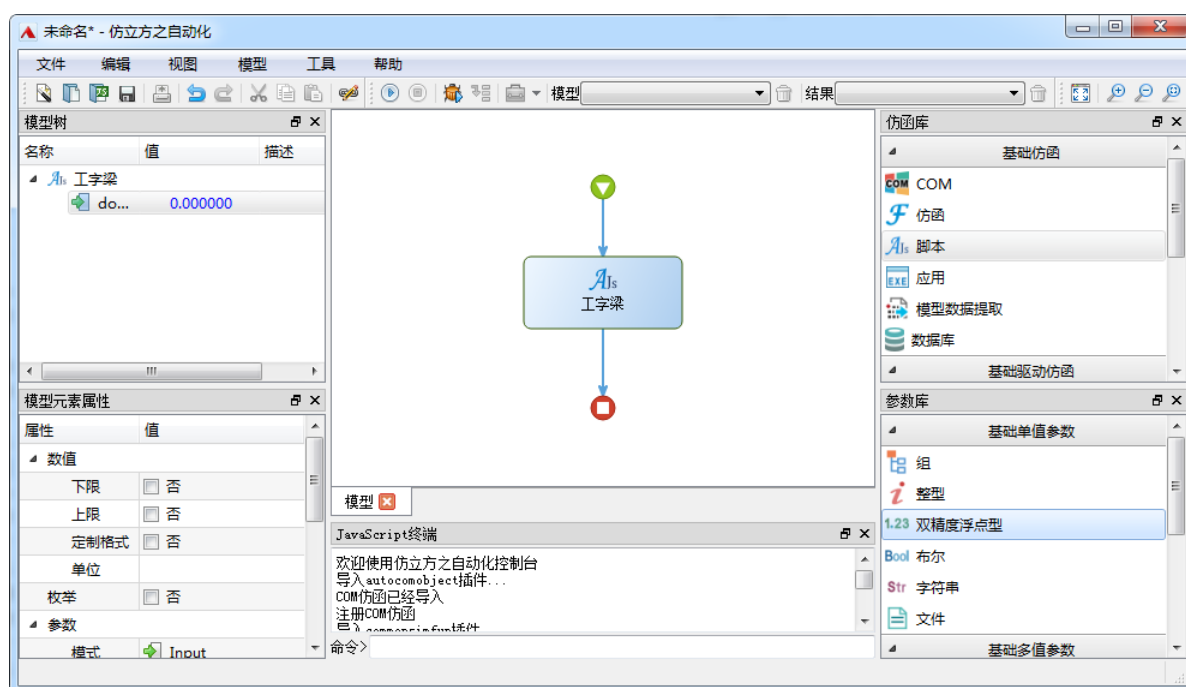
1.1.1.3 添加参数

添加变量 L、P、E、H、B、tw、t 六个输入变量和 disp、angle 两个输出变量，以添加输入变量 L 为例说明添加过程。

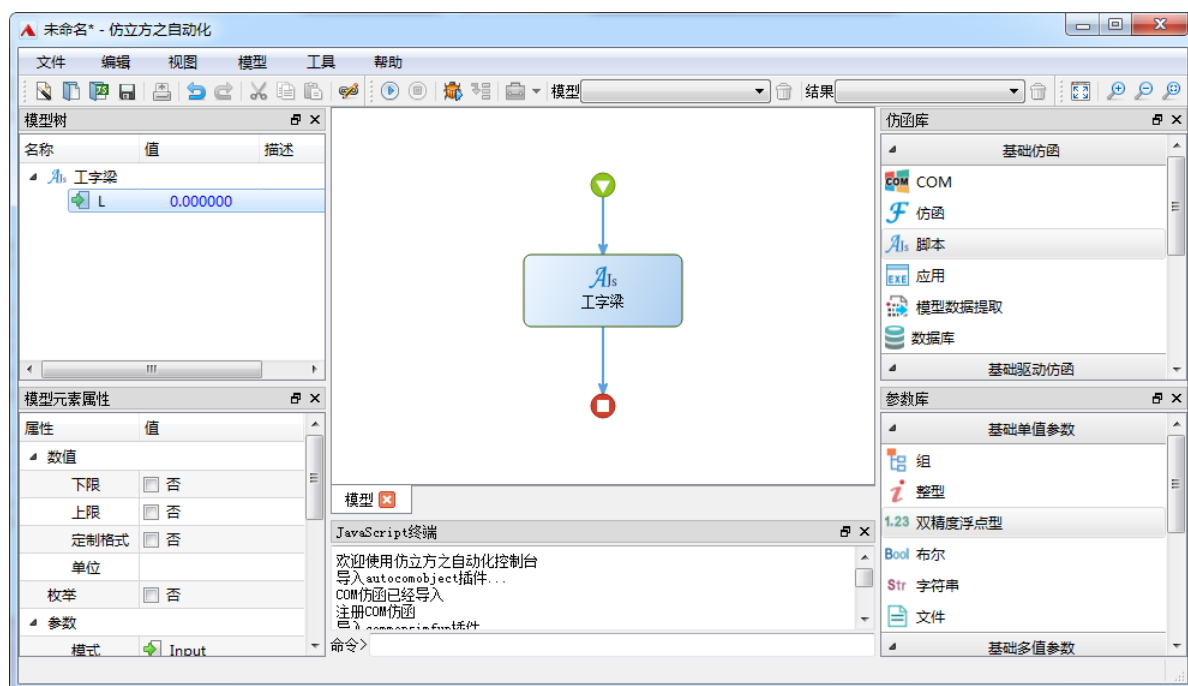
拖拉“基础单值参数”库中的“双精度”到“工字梁”仿函上，“工字梁”仿函高亮显示，并且分成两种颜色，左边是黄色，右边是绿色，如下图所示。



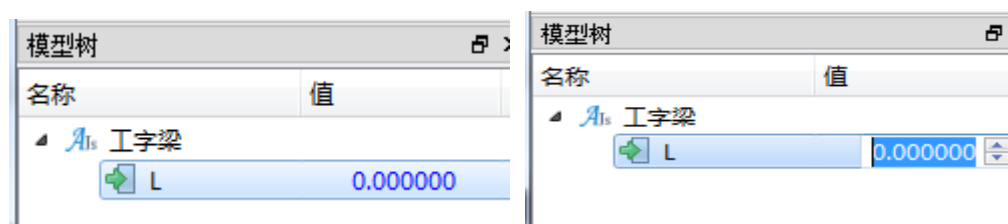
黄色代表输入项，绿色代表输出项。选择黄色区域松开鼠标左键，参数即可被添加至仿函中，在模型树中可以看到新添加的参数，如下图所示。



更改新添加的参数名称，在模型树上点击“double1”节点，在右键菜单中选择“重命名”，在弹出的对话框中输入新的名称，这里输入“L”，更改后的效果如下图所示。

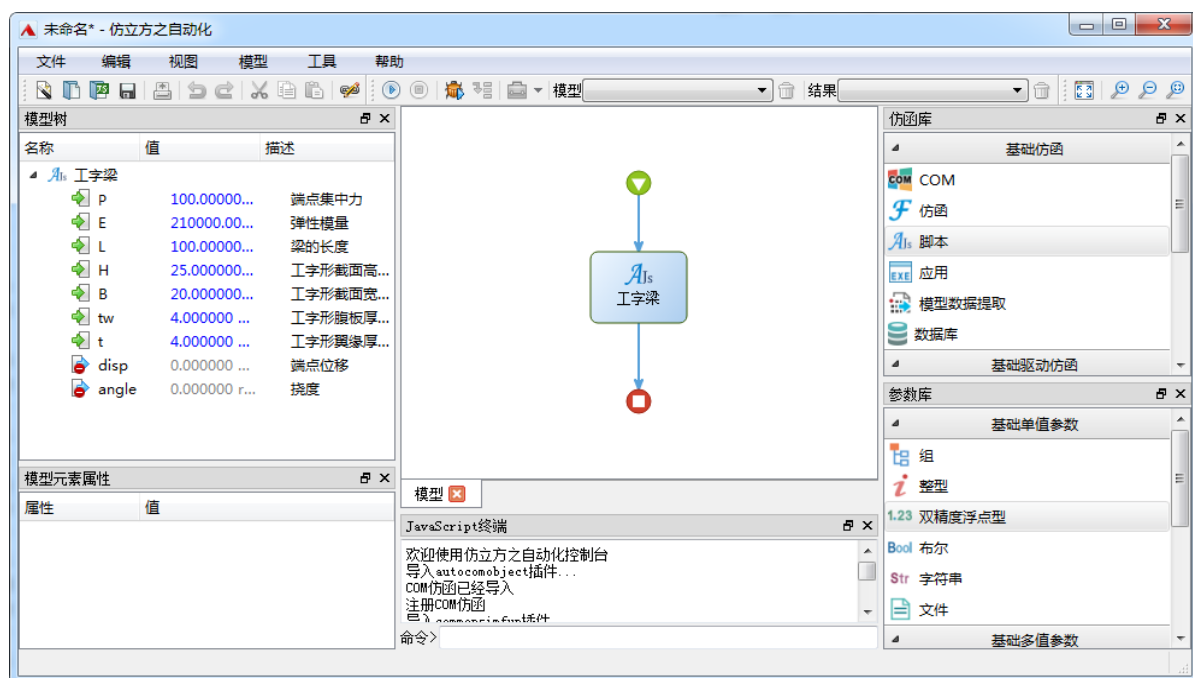


同时更改参数 L 的数值，修改数值的方法如下，在模型树上选择对应的参数 L，点击“值”栏，鼠标点入后，界面变为如下图所示。



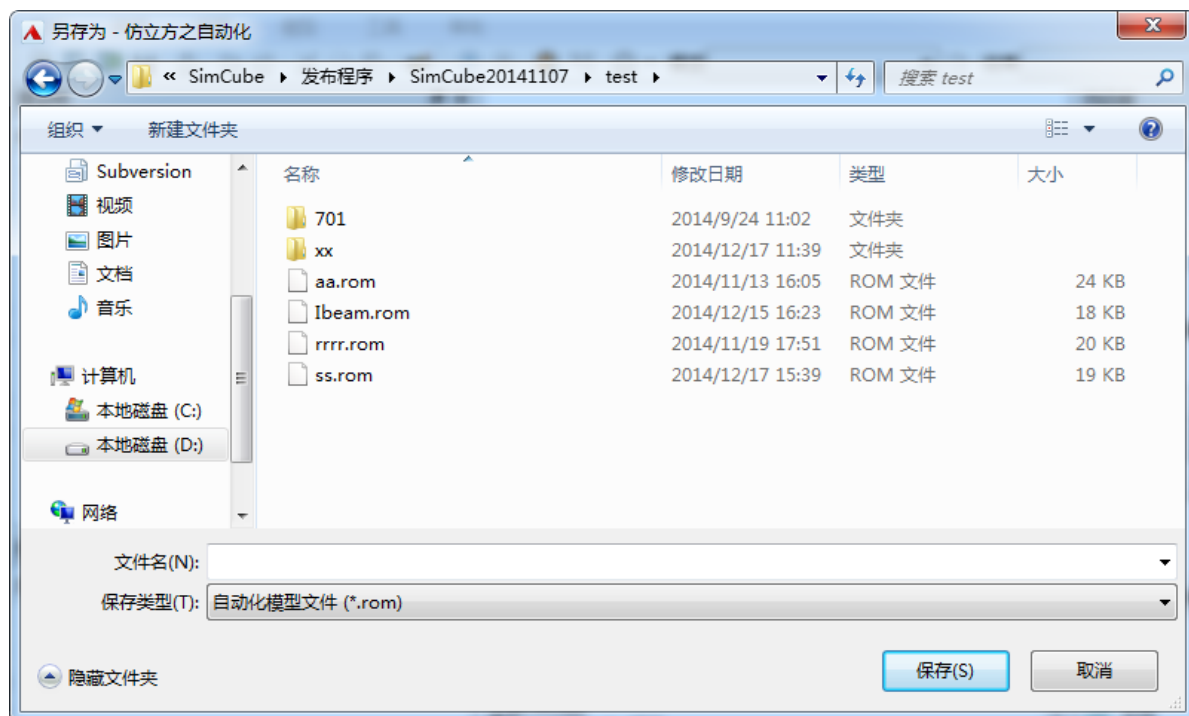
用户输入 100，即可更改参数 L 的数值为 100。

按照同样的方法，添加其它的输入变量和输出变量，添加完的界面如下图所示。



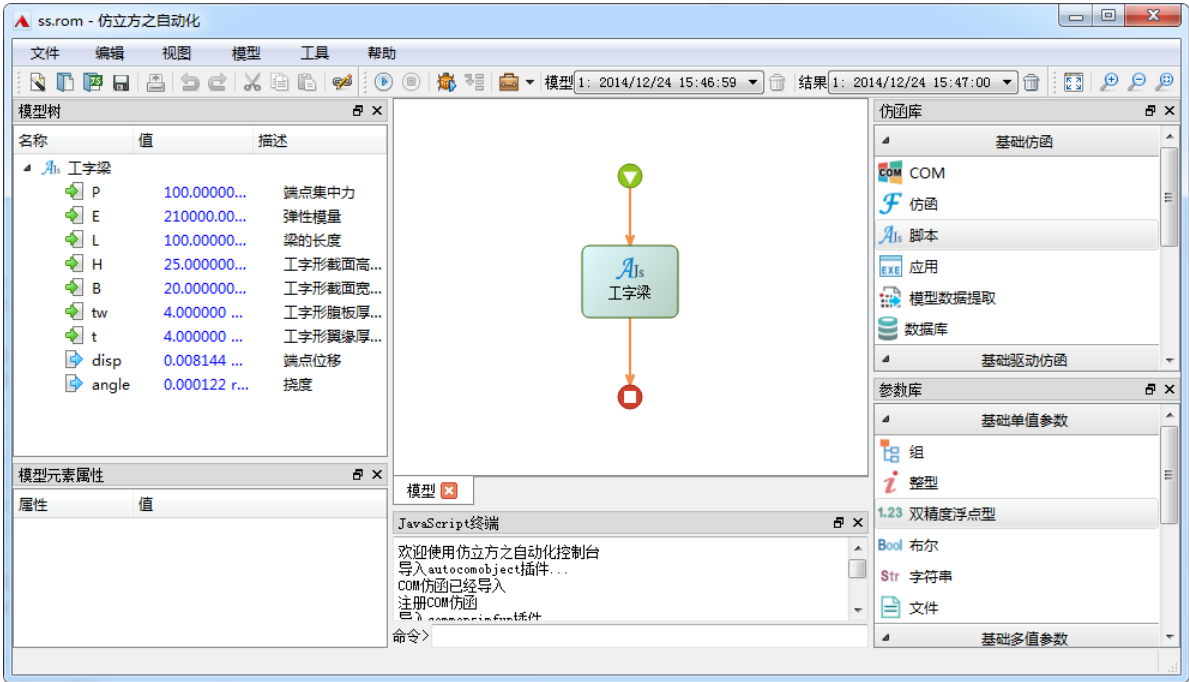
1.1.1.4 运行模型

点击“运行模型”，会要求首先保存一个自动化模型 rom 文件，在弹出的对话框中输入模型名称，如下图所示。



点击“保存”，程序开始计算，计算完成后，可以看到计算的输出值，如下图所示

示。



1.1.2 编写 JavaScript 脚本创建模型

1.1.2.1 编写 JavaScript 代码

在记事本或 SimEditor 中直接编写 JavaScript 代码，内容如下所示。

```
rom.myCreateParameter = function(factory, mode, parent, name, value, desc, unit) {  
    var tmpdata = new factory(name, mode, parent);  
    tmpdata.value = value;  
    tmpdata.description = desc;  
    tmpdata.units = unit;  
    return tmpdata;  
};  
  
rom.model = new rom.Assembly('model', null);  
var model = rom.model;  
var as = new rom.AutomationScript('工字梁', model);  
as.name = '工字梁';  
  
var data1 = rom.myCreateParameter(rom.DoubleParameter, rom.Input, as, 'P', 100, '端点集中力', 'N');  
var data2 = rom.myCreateParameter(rom.DoubleParameter, rom.Input, as, 'E', 210000, '弹性模量', 'MPa');  
var data3 = rom.myCreateParameter(rom.DoubleParameter, rom.Input, as, 'L', 100, '梁的长度', 'mm');
```

```

var data4 = rom.myCreateParameter(rom.DoubleParameter, rom.Input, as, 'H', 25, '工字形截面高度', 'mm');
var data5 = rom.myCreateParameter(rom.DoubleParameter, rom.Input, as, 'B', 20, '工字形截面宽度', 'mm');
var data6 = rom.myCreateParameter(rom.DoubleParameter, rom.Input, as, 'tw', 4, '工字形腹板厚度', 'mm');
var data7 = rom.myCreateParameter(rom.DoubleParameter, rom.Input, as, 't', 4, '工字形翼缘厚度', 'mm');
var data8 = rom.myCreateParameter(rom.DoubleParameter, rom.Output, as, 'disp', 0, '端点位移', 'mm');
var data9 = rom.myCreateParameter(rom.DoubleParameter, rom.Output, as, 'angle', 0, '挠度', 'rad');

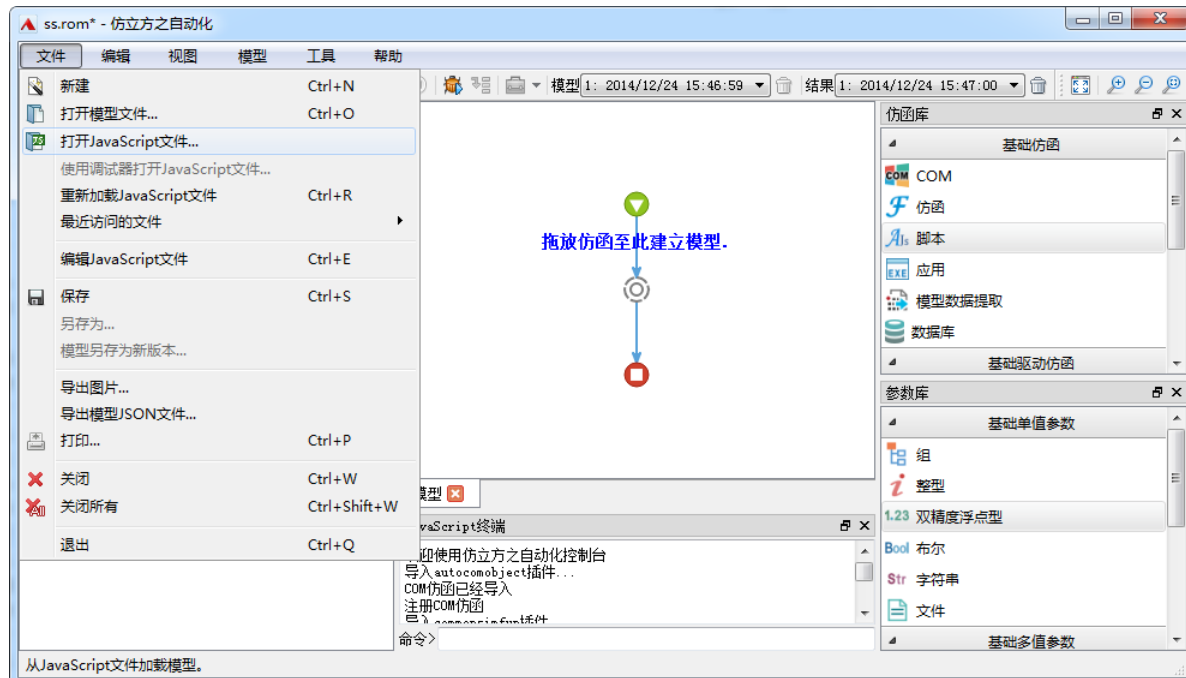
```

```

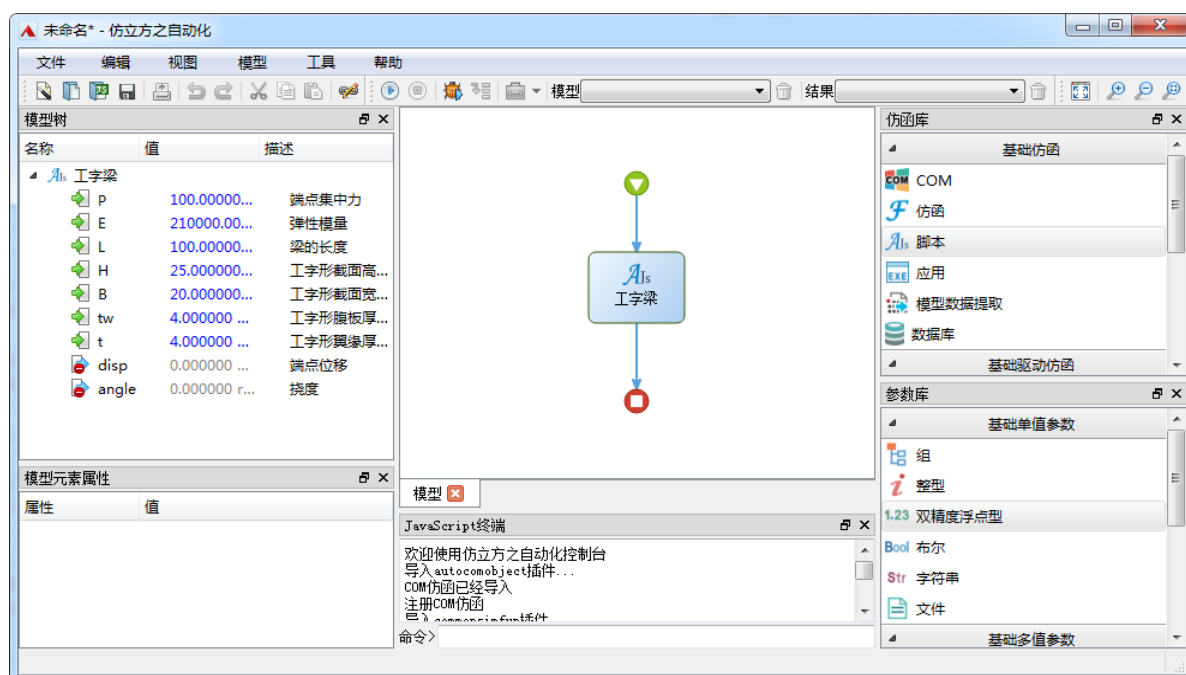
as.scriptFunction = function() {
    var tmp1 = Math.pow(self.H.value, 3);
    var tmp2 = Math.pow(self.H.value-2*self.t.value, 3);
    var Iy = (self.B.value * tmp1-(self.B.value - self.tw.value) * tmp2)/12;
    self.disp.value = self.P.value * Math.pow(self.L.value, 3)/(3 * self.E.value * Iy);
    self.angle.value = self.P.value * Math.pow(self.L.value, 2)/(2 * self.E.value * Iy);
};

```

编写完成后，保存为 js 文件，然后启动 Automation，在 Automation 中点击“文件|打开 JavaScript 文件”，如下图所示。

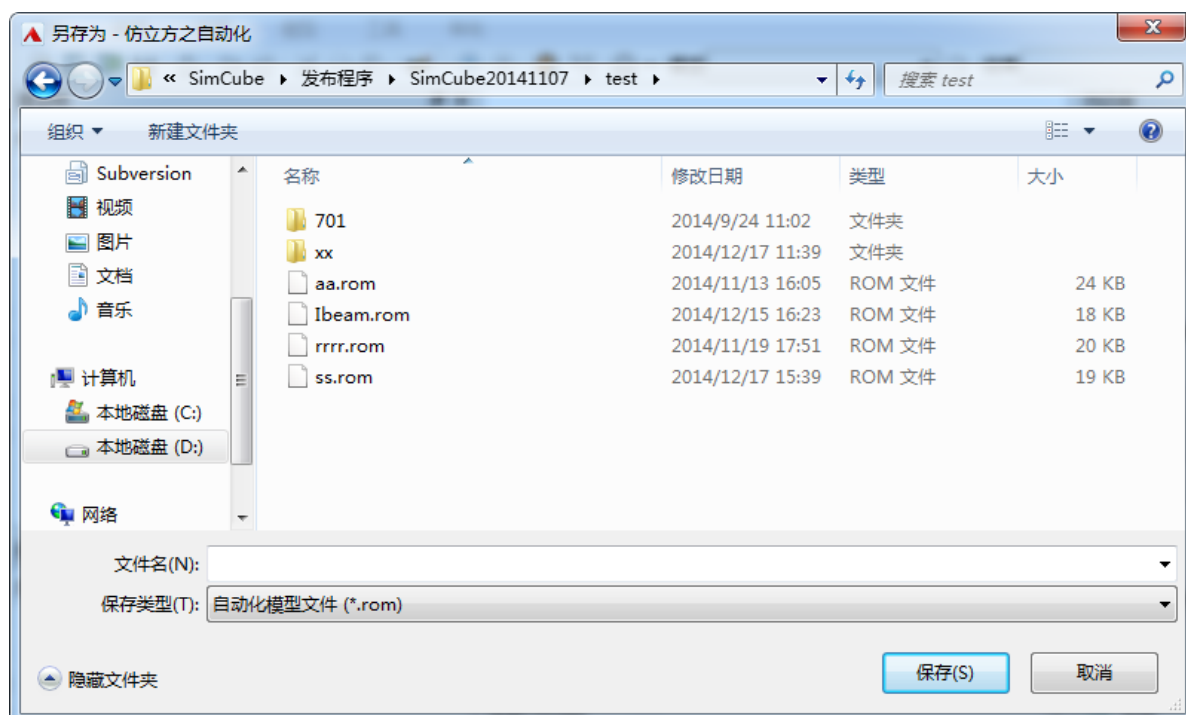


打开后界面如下图所示。



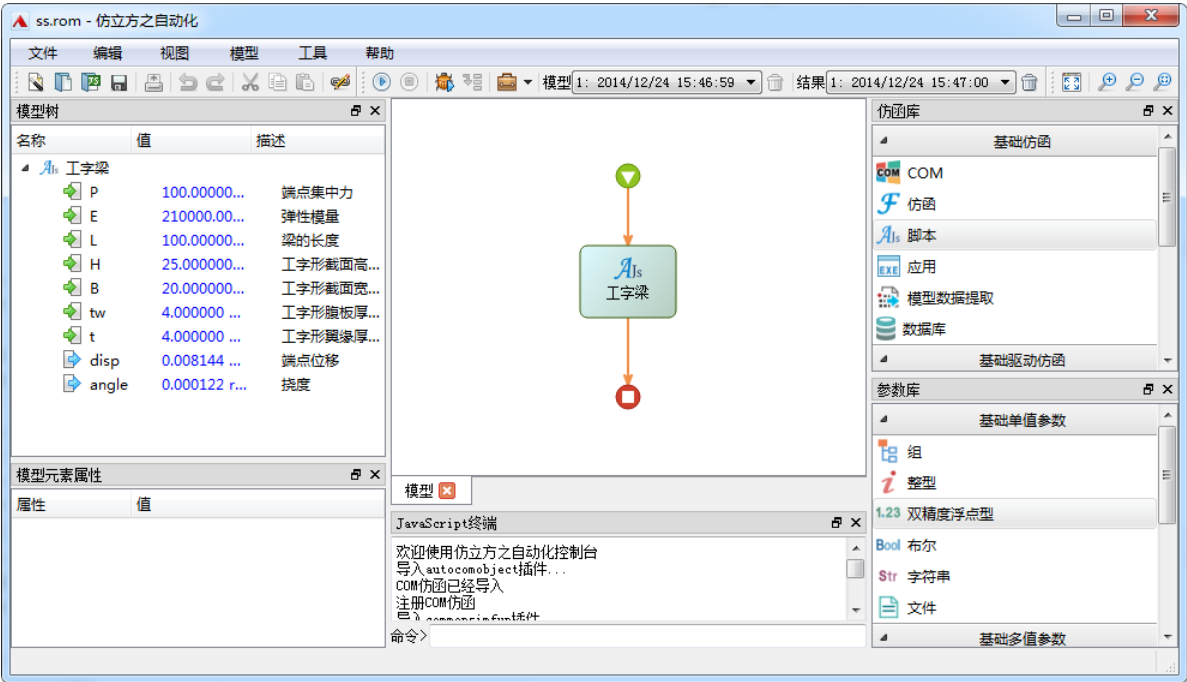
1.1.2.2 运行模型

点击“运行模型”，会要求首先保存一个自动化模型 rom 文件，在弹出的对话框中输入模型名称，如下图所示。



点击“保存”，程序开始计算，计算完成后，可以看到计算的输出值，如下图所示

示。

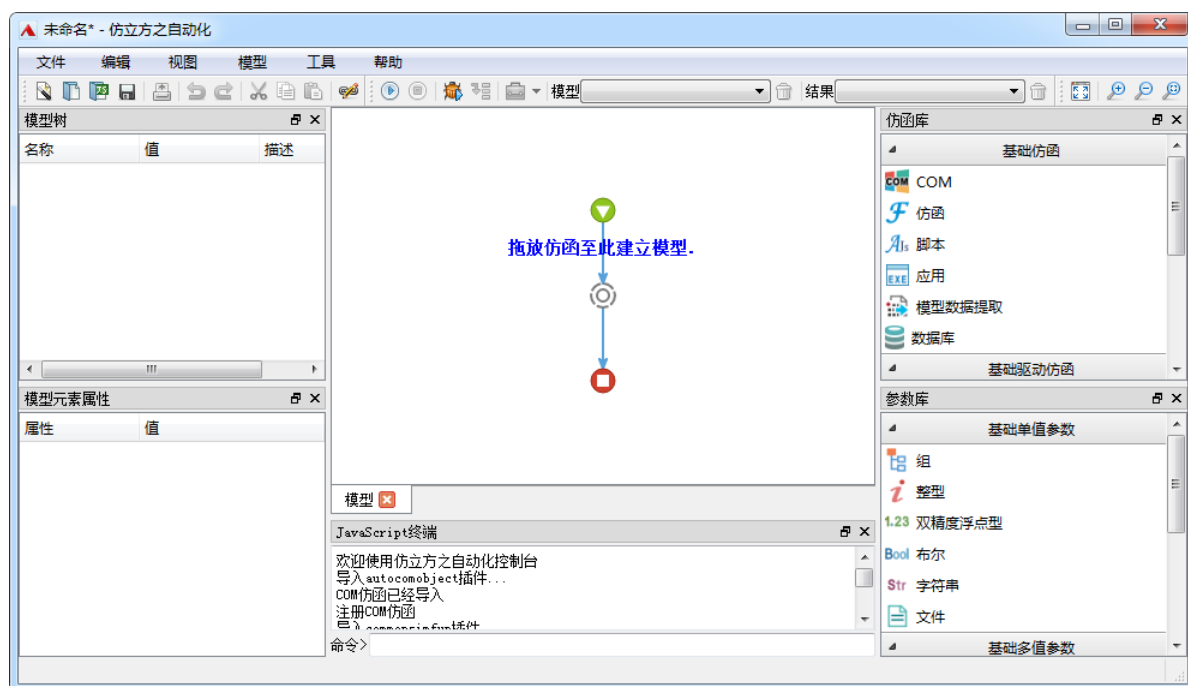


1.2 应用创建模型

1.2.1 可视化创建模型

1.2.1.1 启动

启动 Automation 程序，界面如下图所示。

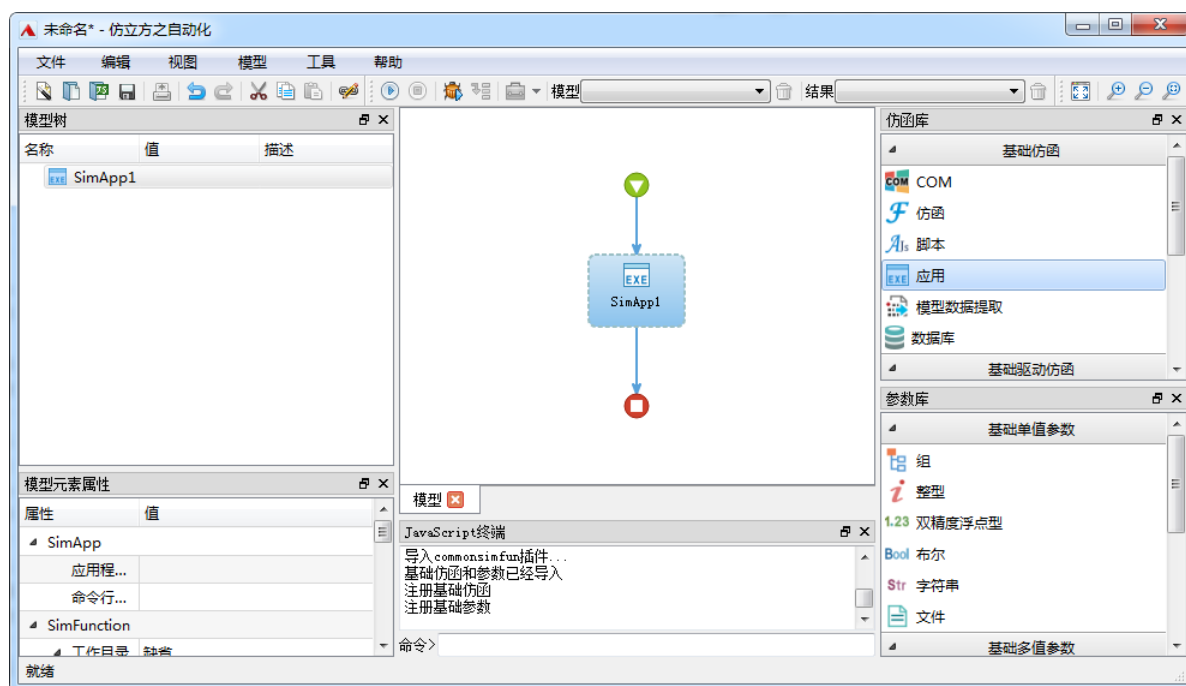


1.2.1.2 添加仿函

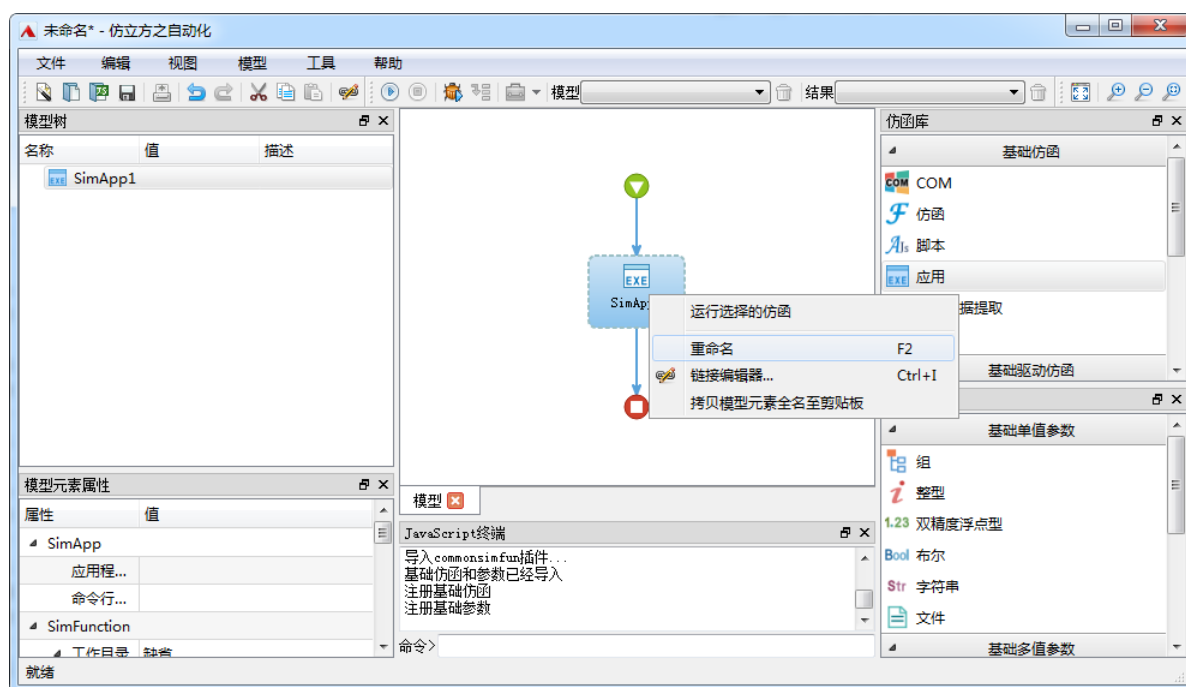
拖动“应用”仿函到工作区，拖动的过程如下，在工作区中心位置有一个中心圆环，中心圆环表示可以添加新的仿函。拖拉“基础驱动仿函”库中的“应用”驱动仿函到中心圆环，当拖拉至中心圆环附近时，中心圆环变为方形并高亮显示，如下图所示。



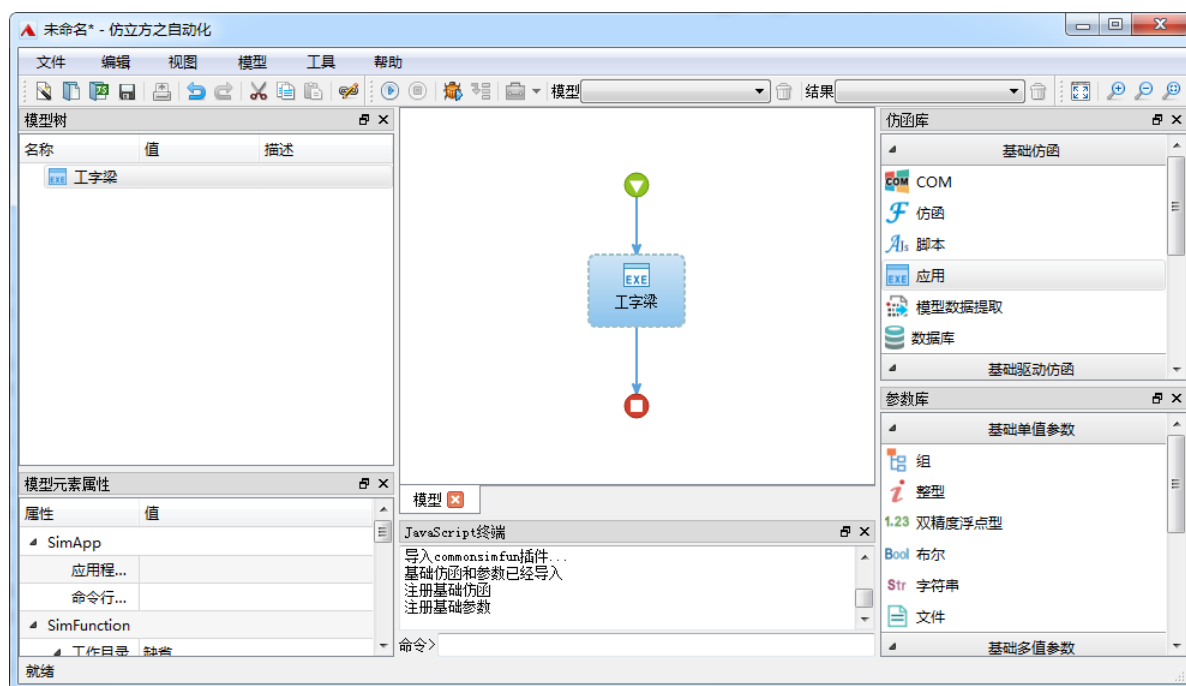
此时松开鼠标左键，“应用”驱动仿函被添加至对应的位置上，如下图所示。



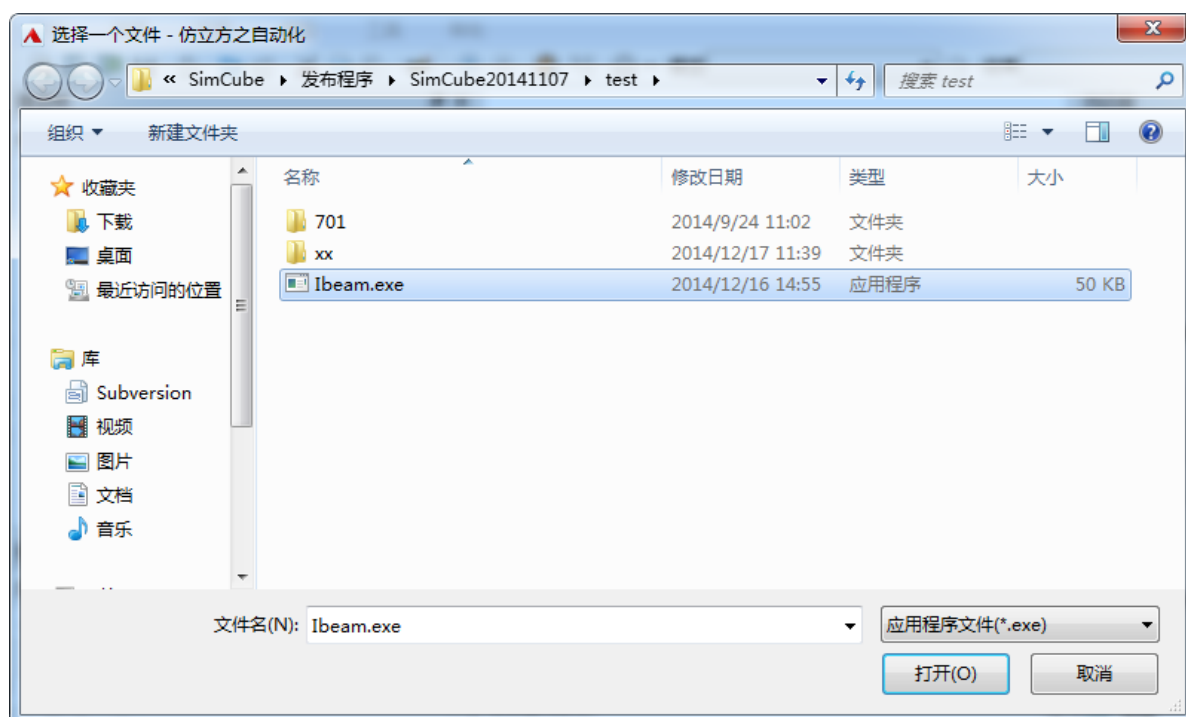
重命名新建的仿函，选择“SimApp1”，如下图所示。



在右键菜单中点击“重命名”，在弹出的对话框中输入新的名称，这里输入“工字梁”，如下图所示。



指定运行的程序，在“模型元素属性”窗口中点击“SimApp | 应用程序路径”，点击“编辑”会弹出浏览应用程序的窗口，如下图所示。

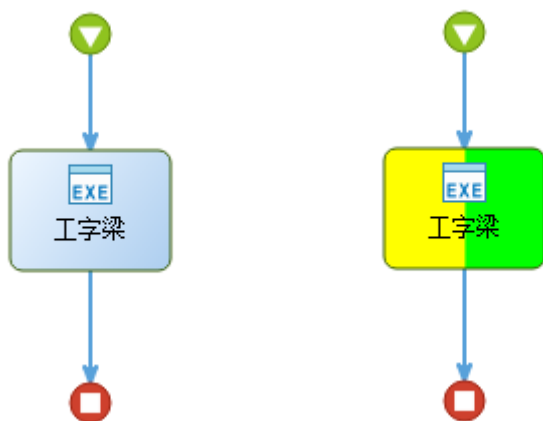


点击“打开”，“应用程序路径”设置为选择的路径。

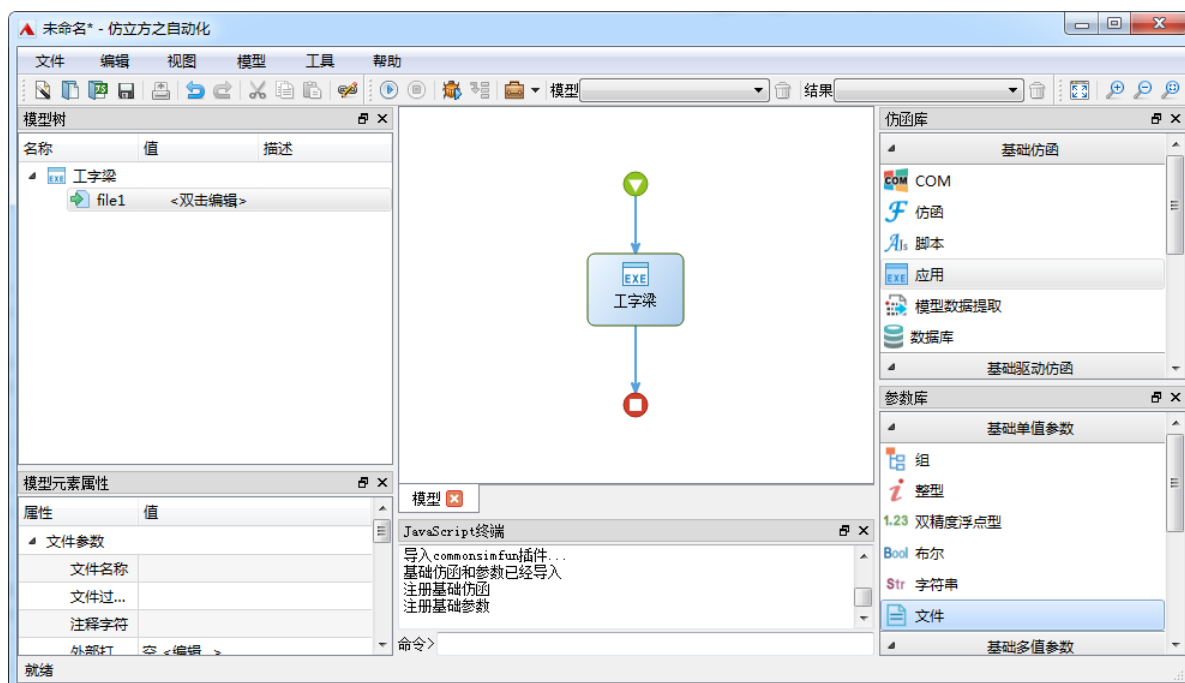
1.2.1.3 添加文件参数

添加运行应用程序所需要的输入文件和输出文件。

拖拉“基础单值参数”库中的“文件”到“工字梁”仿函上，“工字梁”仿函高亮显示，并且分成两种颜色，左边是黄色，右边是绿色，如下图所示。

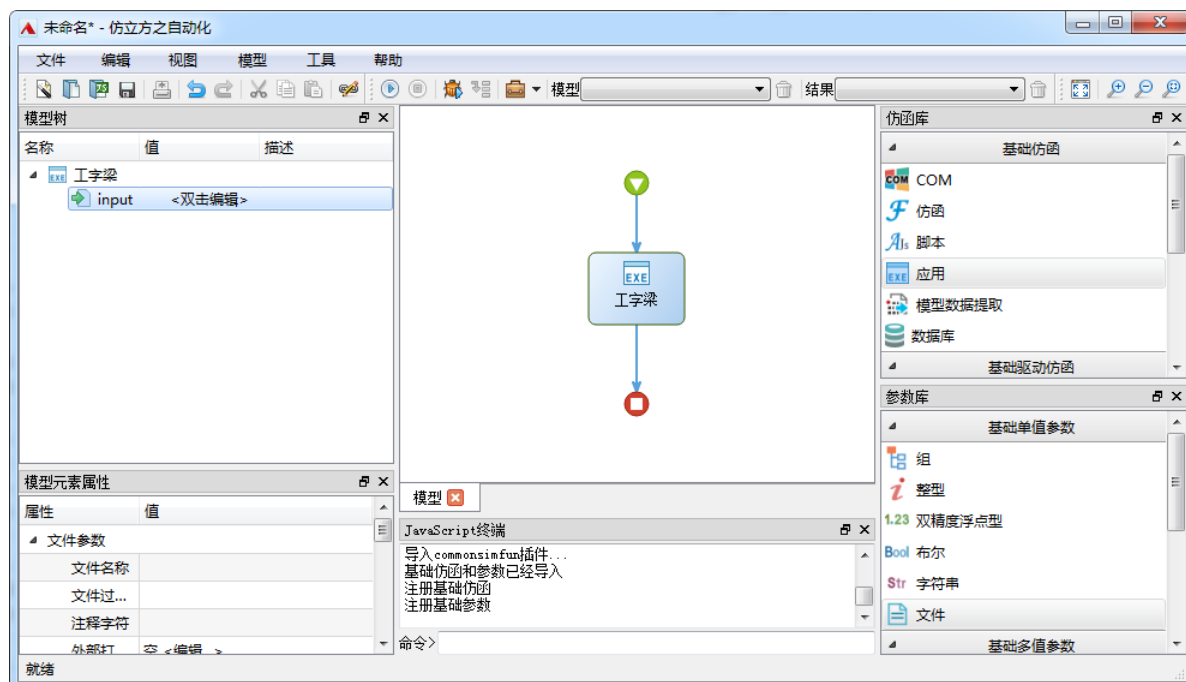


黄色代表输入项，绿色代表输出项。选择黄色区域松开鼠标左键，参数即可被添加至仿函中，在模型树中可以看到新添加的参数，如下图所示。

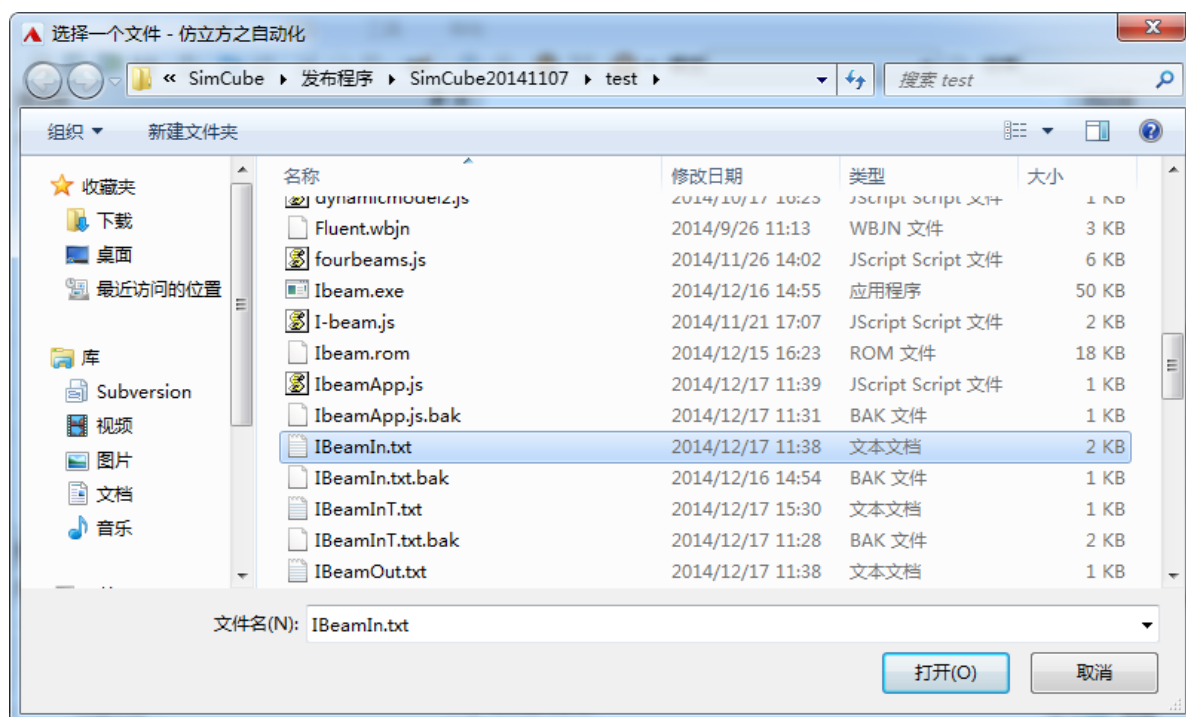


更改新添加的参数名称，在模型树上点击“file1”节点，在右键菜单中选择“重命名”，在弹出的对话框中输入新的名称，这里输入“input”，更改后的效果如下图

所示。

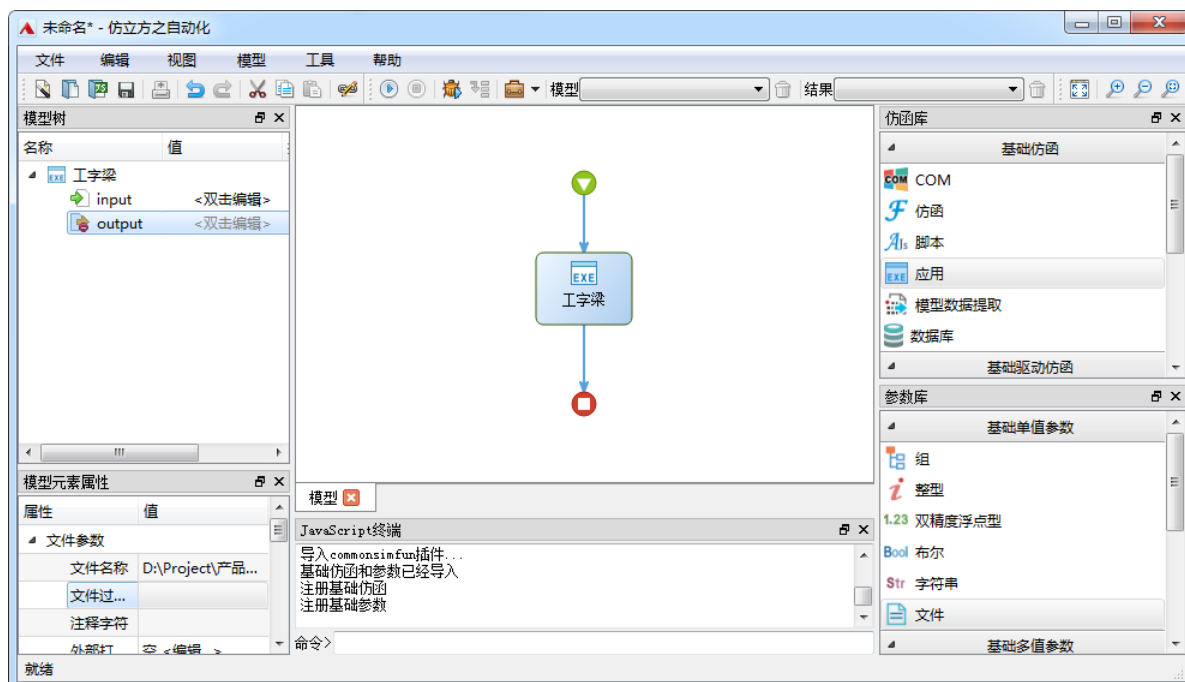


设置输入参数 input 的值，设置方法如下，在“模型元素属性”窗口中点击“文件参数 | 文件名称”，点击“编辑”会弹出浏览应用程序的窗口，如下图所示。



点击“打开”，“文件名称”设置为选择的路径。

按照同样的方法，添加输出文件变量 output，添加完的界面如下图所示。

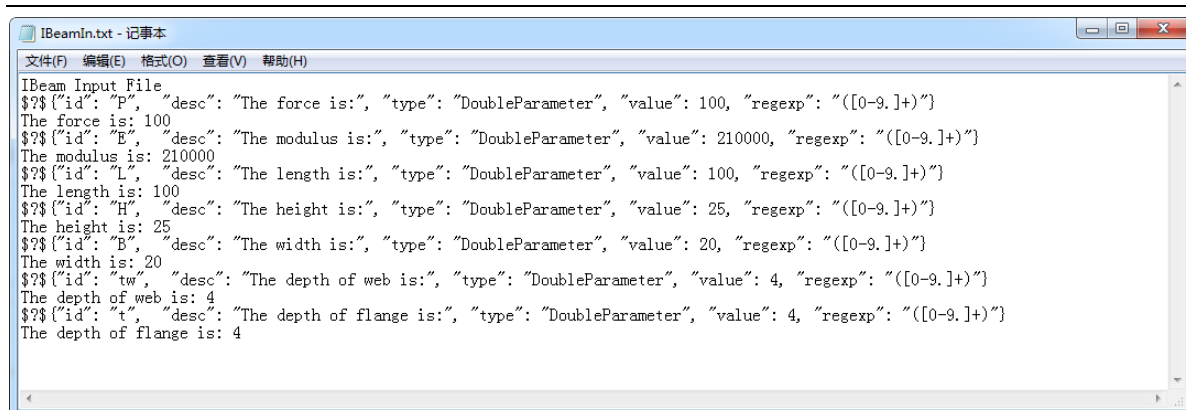


1.2.1.4 提取文件中的参数

通过在输入/输出文件中标识符的方法来提取输入/输出文件中的参数，以输入文件为例，共提取 7 个输入变量，在原来的输入文件中提取参数的行前增加标识，如下图所示。

```

1 IBeam Input File
2 The force is: 100
3 The modulus is: 210000
4 The length is: 100
5 The height is: 25
6 The width is: 20
7 The depth of web is: 4
8 The depth of flange is: 4
  
```



增加的标识符说明如下：

"id": "P"	表示提取参数的名称；
"desc": "The force is:"	表示提取参数的描述；
"type": "DoubleParameter"	表示提取参数的类型；
"value": 100	表示提取参数的数值；
"regexp": "([0-9.]+)"	表示提取参数的正则表达式。

输入输出文件内容更改完之后，在界面中进行如下设置：

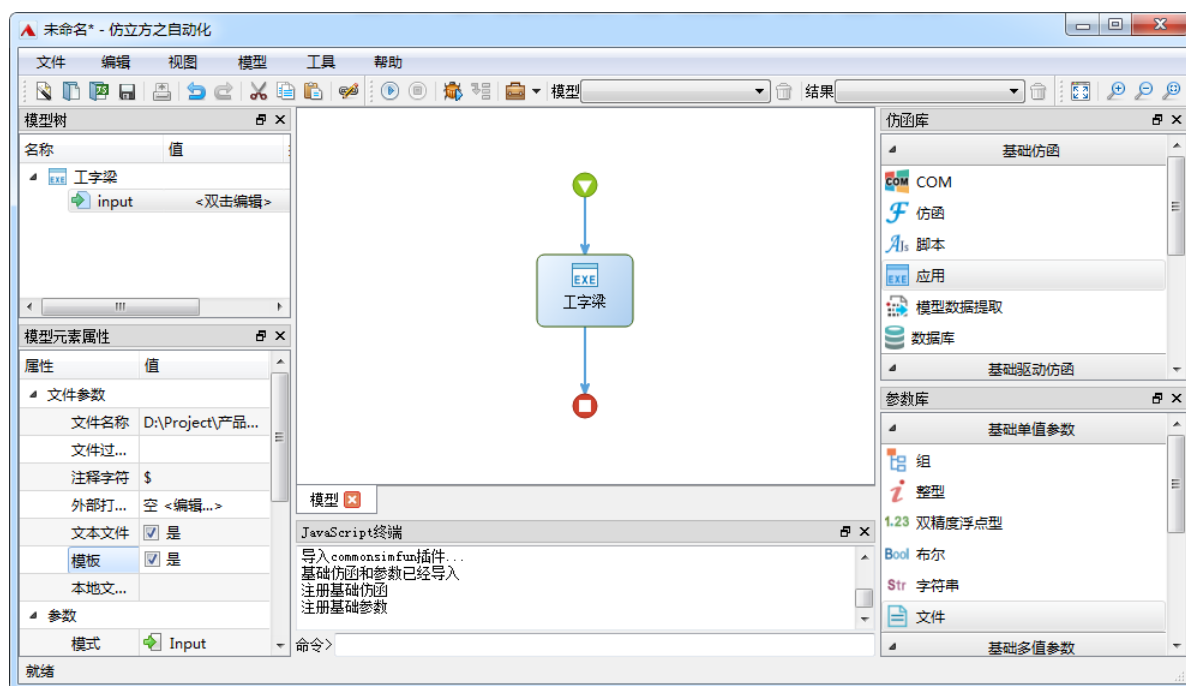
点击文件名称中“值”，点击“编辑”浏览编辑后的输入文件；

设置注释符号为\$；

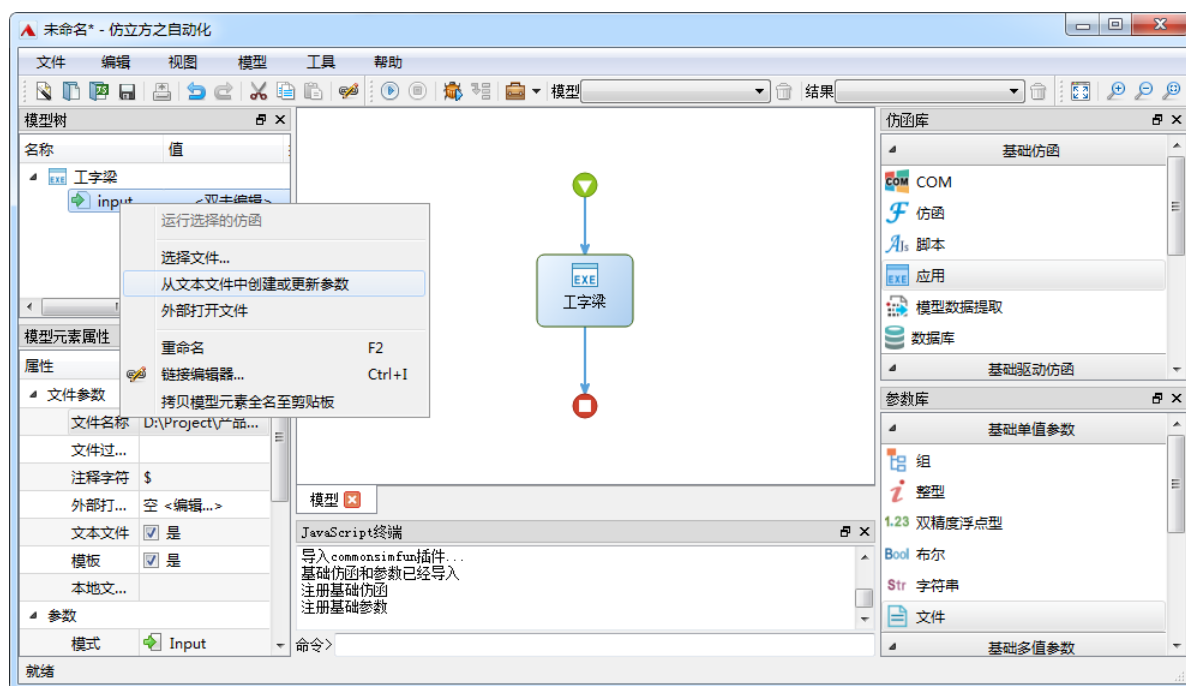
勾选文本文件的复选框；

勾选模板的复选框。

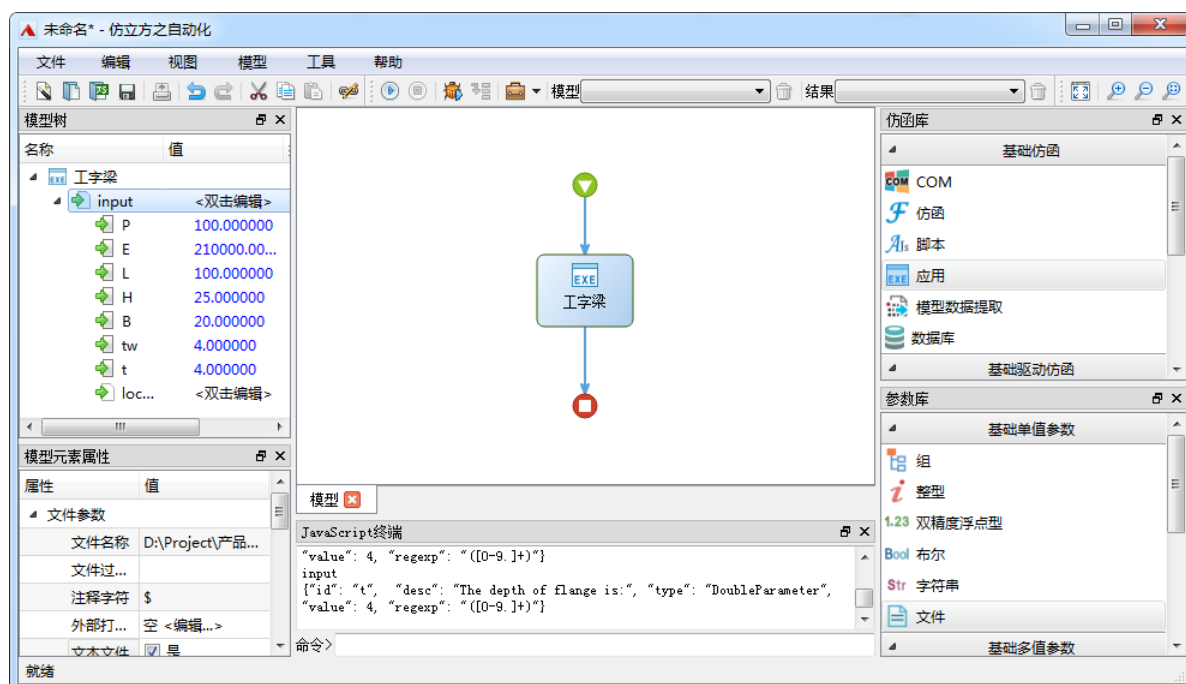
设置完成后的界面，如下图所示。



输入文件设置完成后，在模型树上选择“input”节点，在右键菜单中选择“从文本文件中创建或更新参数”，如下图所示。



程序会自动提取输入文件中的参数，如下图所示。

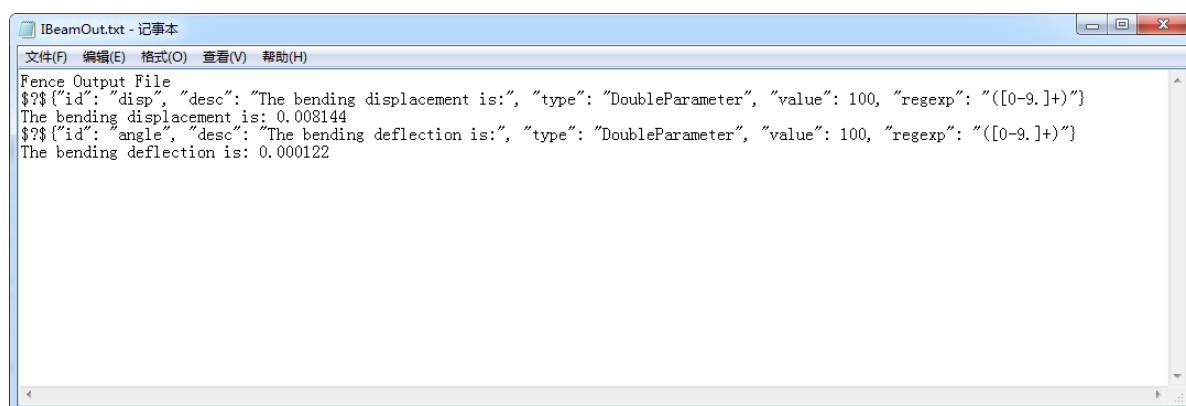


同样对输出文件增加标识，如下图所示。

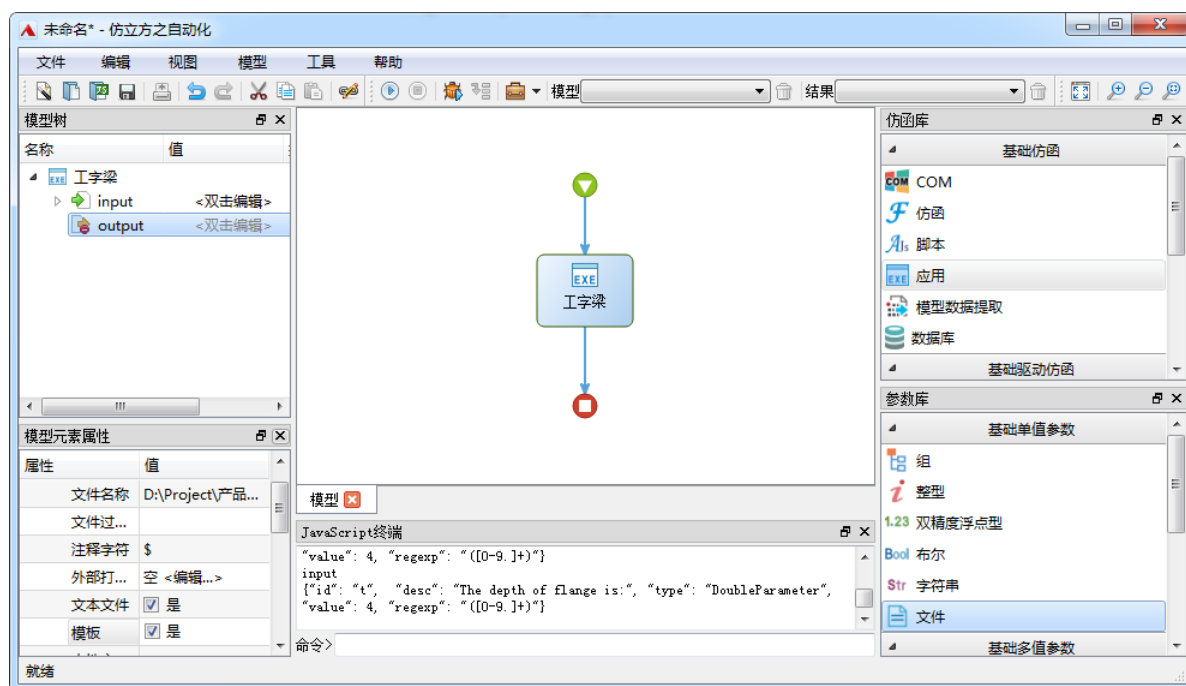
```

1 Fence Output File
2 The bending displacement is: 0.008144
3 The bending deflection is: 0.000122

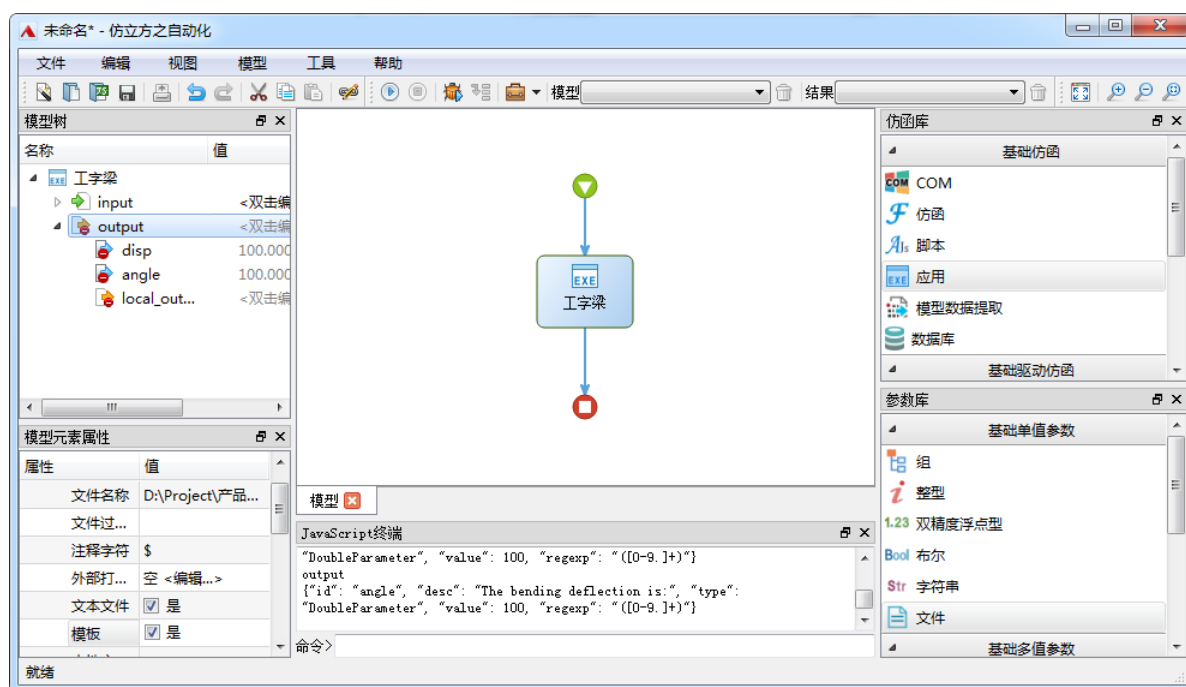
```



输入输出文件内容更改完之后，在界面中进行如下设置。

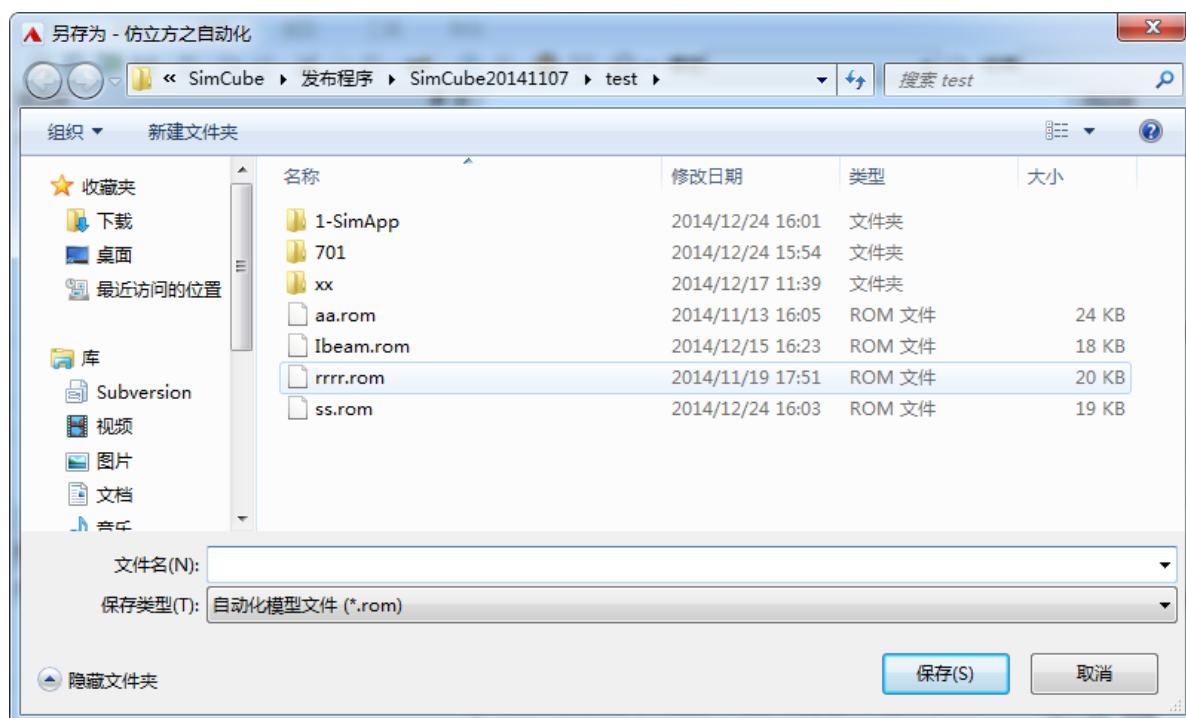


参数提取完成后，在界面中显示如下图所示。

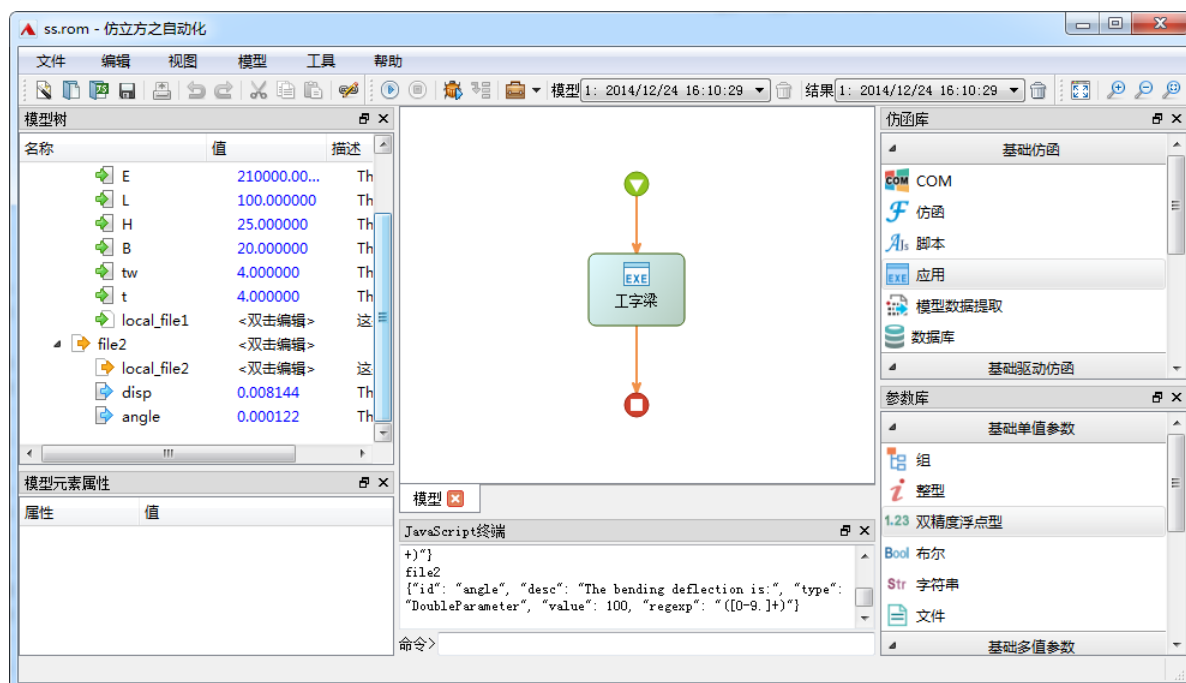


1.2.1.5 运行模型

点击“运行模型”，会要求首先保存一个自动化模型 rom 文件，在弹出的对话框中输入模型名称，如下图所示。



点击“保存”，程序开始计算，计算完成后，可以看到计算的输出值，如下图所示。



1.2.2 编写 JavaScript 脚本创建模型

1.2.2.1 编写 JavaScript 代码

在记事本或 SimEditor 中直接编写 JavaScript 代码，内容如下所示。

```
rom.model = new rom.Assembly('model', null);
var model = rom.model;

var sa = new rom.SimApp('工字梁', model);
sa.appPath = path.join(path.dirname(__FILE__), 'Ibeam.exe');
sa.differentDirectory = true;

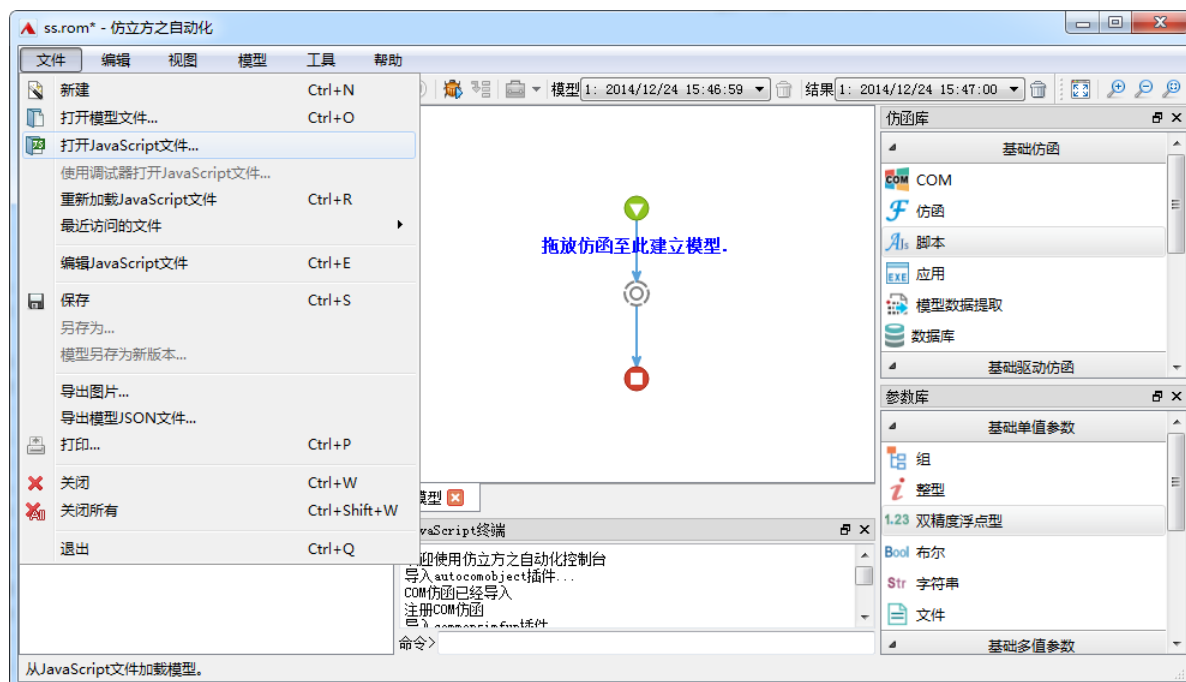
var file1 = new rom.FileParameter('file1', rom.InputFile, sa);
file1.fileName = path.join(path.dirname(__FILE__), 'IBeamIn.txt');
if (!fs.exists(file1.fileName))
    throw Error('输入文件不存在: ' + file1.fileName);
file1.isText = true;
file1.comment = '$';
rom.createParametersFromFileParameter(file1);
file1.localFileName = path.filename(file1.fileName);

var file2 = new rom.FileParameter('file2', rom.OutputFile, sa);
file2.fileName = path.join(path.dirname(__FILE__), 'IBeamOut.txt');
if (!fs.exists(file2.fileName))
    throw Error('输出文件模板不存在: ' + file2.fileName);
file2.isText = true;
file2.comment = '$';
file2.localFileName = path.filename(file2.fileName);
rom.createParametersFromFileParameter(file2);

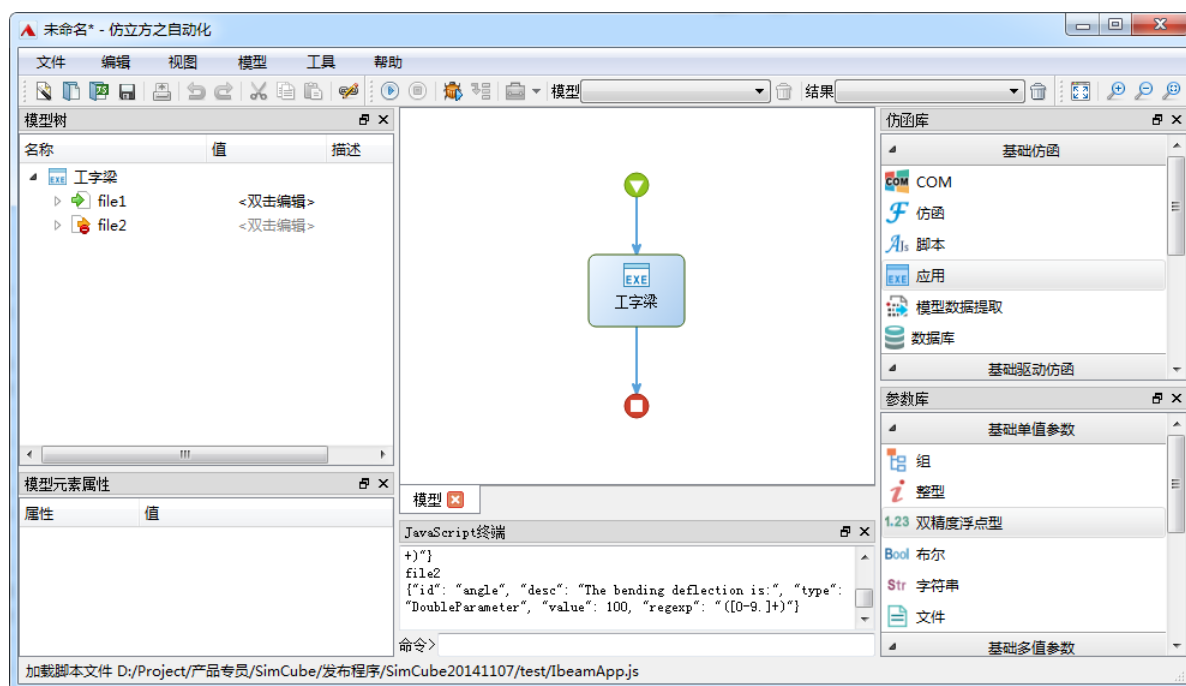
sa.args = [file1.fullName, file2.fullName];
```

编写完成后，保存为 js 文件，然后启动 Automation，在 Automation 中点击“文

件|打开 JavaScript 文件”，如下图所示。



打开后界面如下图所示。



1.2.2.2 提取文件中的参数

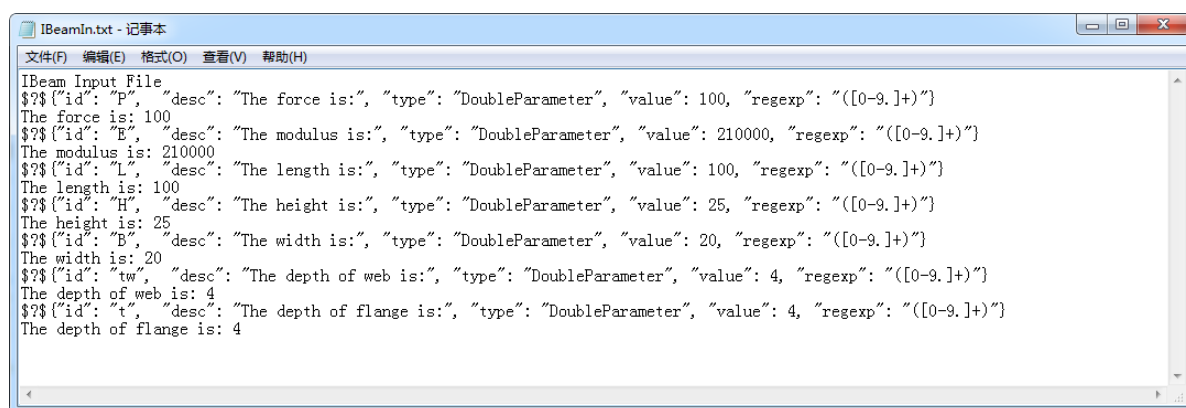
通过在输入/输出文件中标识符的方法来提取输入/输出文件中的参数，以输入文

件为例，共提取 7 个输入变量，在原来的输入文件中提取参数的行前增加标识，如下图所示。

```

1 IBeam Input File
2 The force is: 100
3 The modulus is: 210000
4 The length is: 100
5 The height is: 25
6 The width is: 20
7 The depth of web is: 4
8 The depth of flange is: 4

```



增加的标识符说明如下：

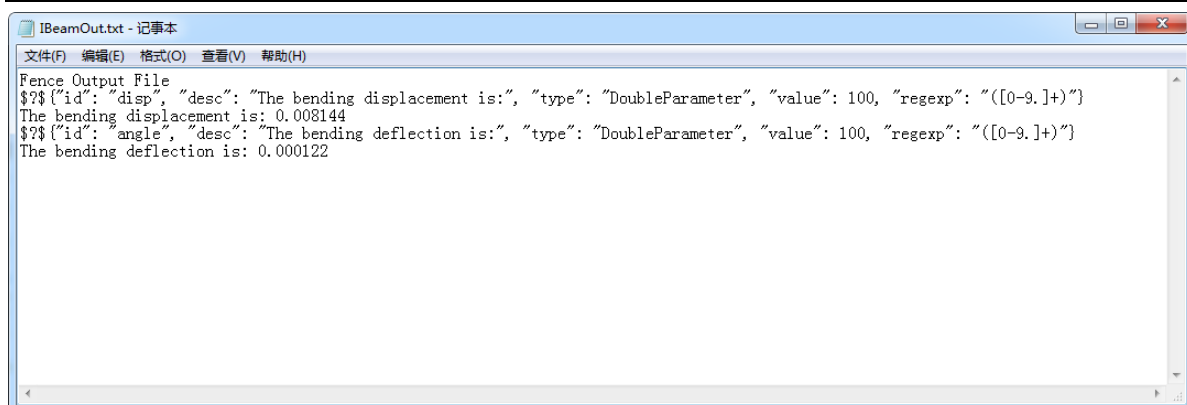
"id": "P"	表示提取参数的名称；
"desc": "The force is:"	表示提取参数的描述；
"type": "DoubleParameter"	表示提取参数的类型；
"value": 100	表示提取参数的数值；
"regexp": "([0-9.]+)"	表示提取参数的正则表达式。

同样对输出文件增加标识，如下图所示。

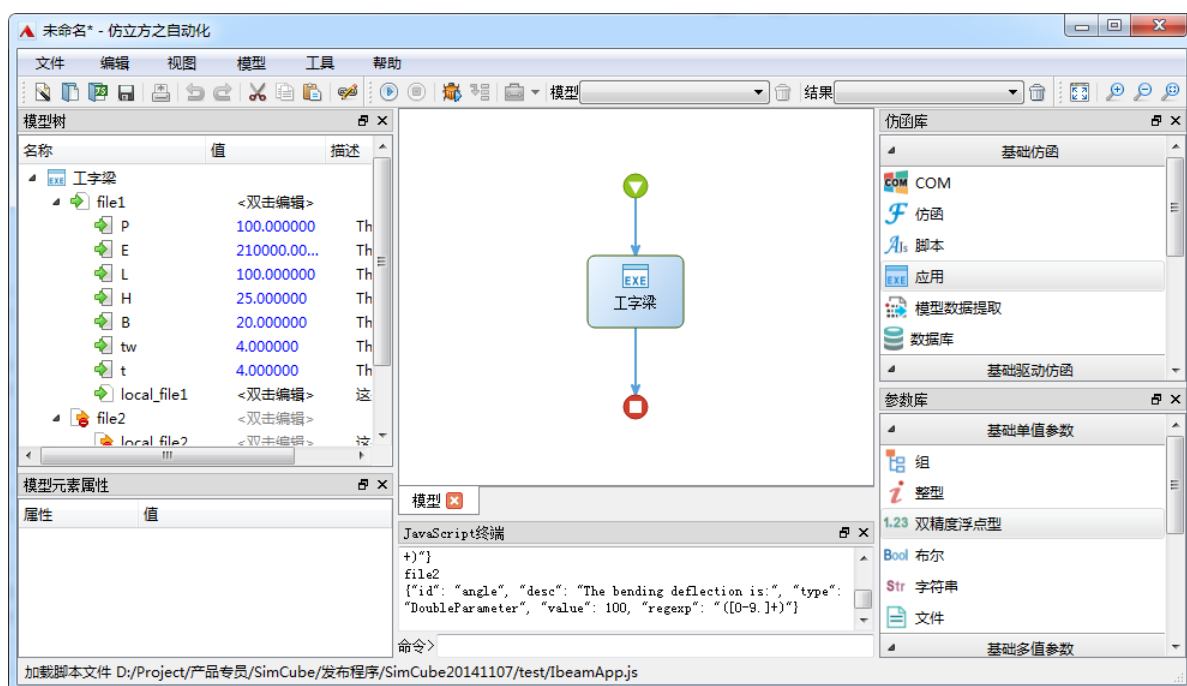
```

1 Fence Output File
2 The bending displacement is: 0.008144
3 The bending deflection is: 0.000122

```

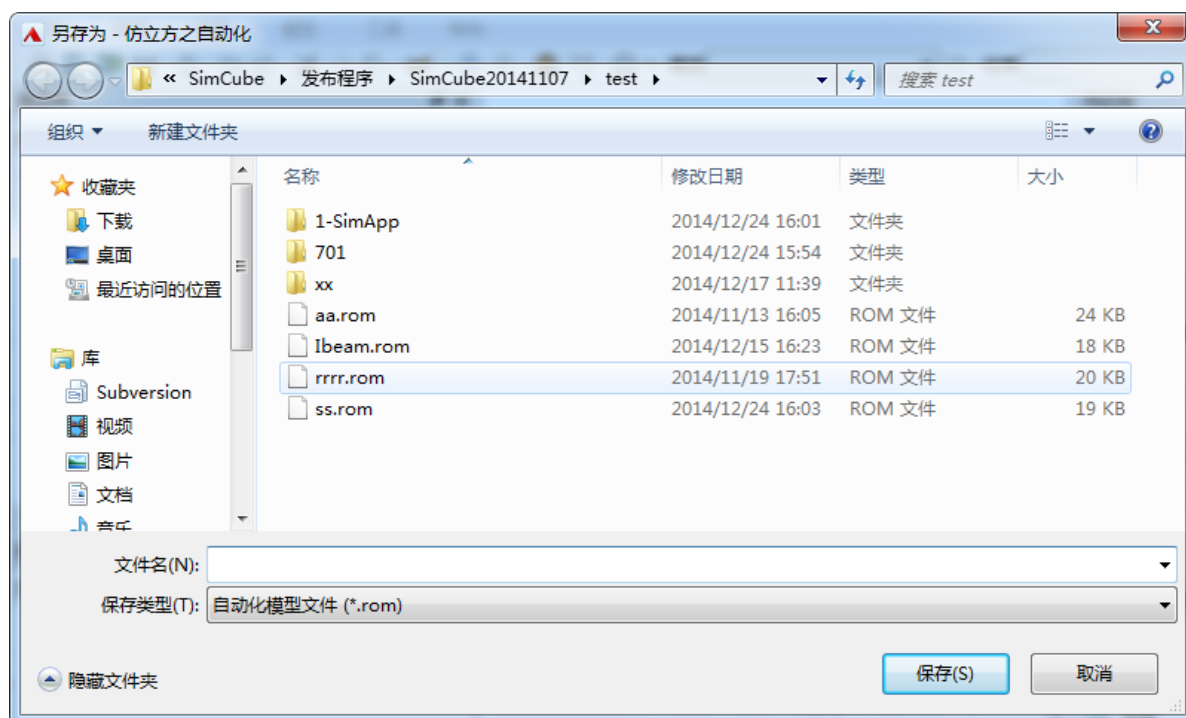


参数提取完成后，在界面中显示如下图所示。



1.2.2.3 运行模型

点击“运行模型”，会要求首先保存一个自动化模型 rom 文件，在弹出的对话框中输入模型名称，如下图所示。



点击“保存”，程序开始计算，计算完成后，可以看到计算的输出值，如下图所示。

