

LabVIEW 与 MATLAB 混合编程之 COM

目录

一、引文	2
二、工具	2
三、MatLab 生成 COM 组件	3
四、LabVIEW 中调用 COM 组件	5
五、多输入多输出的 MATLAB 函数	10
六、多函数的 MATLAB 程序	11
七、一个 com 调用多个函数---多个函数一次封装	13
八、最后四点比较有用的说明	16

一、引文

LabVIEW 适合开发设备采集和控制程序，虽然也提供了大量的数学运算函数，但与 MatLab 的功能不可同日而语。如果需要在 LabVIEW 中调用 MatLab 程序，有几种方法：1.MatLab 脚本节点 2.DLL 3.COM。

MatLab 脚本节点最容易实现，但前提是目标机器上必须安装 MatLab，如果是简单的就用 MatLab 里的几条语句，不值得这样做；DLL 是利用编译器将 MatLab 程序封装成函数库，但 MatLab 编译器直接封装出来的 DLL 在 LabVIEW 中基本不可用，因为 MatLab 封装出来的 DLL 是基于 mxArray 数据类型的，LabVIEW 没有或找不到与之对应的数据类型（用 DLL 一个可行的办法是实用 VC/VC++进行二次封装，但费时费力）；COM 也是 MatLab 能够封装成的一种形式，而且是基于对象的，可以在 LabVIEW 中成功调用。

本文主要介绍利用 COM 实现 LabVIEW 和 MatLab 混合编程。

二、工具

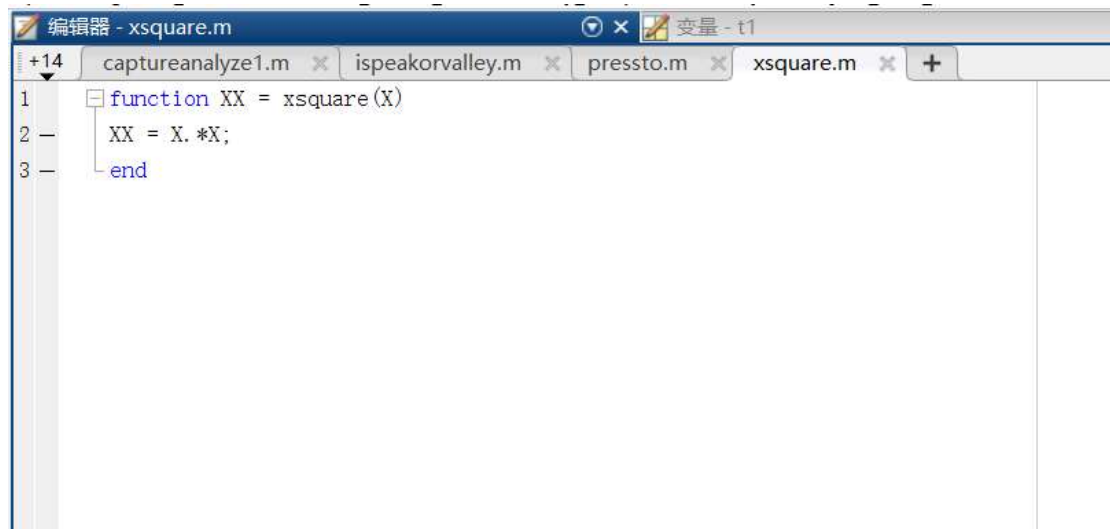
LabVIEW 2012 及更新版本（我使用的 LabVIEW2018 32-bit，保存为前期版本至 2012）

MatLab R2015a 及更新版本（我使用的 MATLAB R2019a 64-bit，如果是 R2015a 以及更早的 32-bit 可以把代码复制，自己建脚本文件，自行生成 dll）

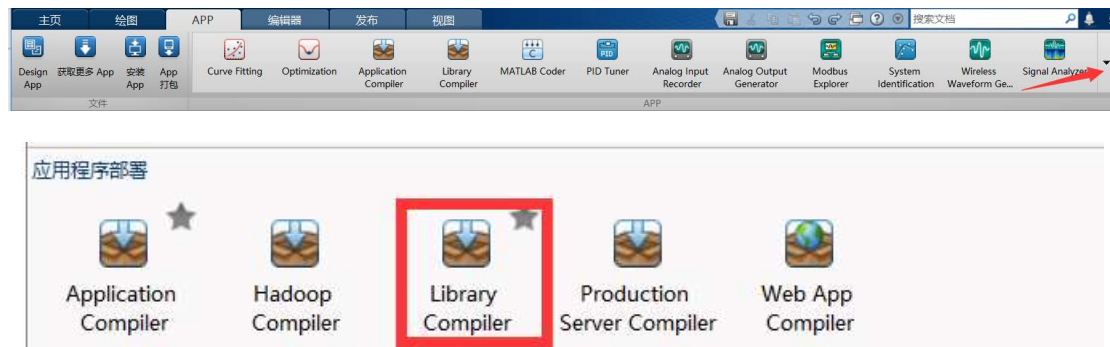
三、MatLab 生成 COM 组件

1、MatLab 中新建脚本文件，写入函数。这里简单的写一个求平方的函数作为例子（单输入单输出）。

```
function XX = xsquare(X)
XX = X.*X;
end
```

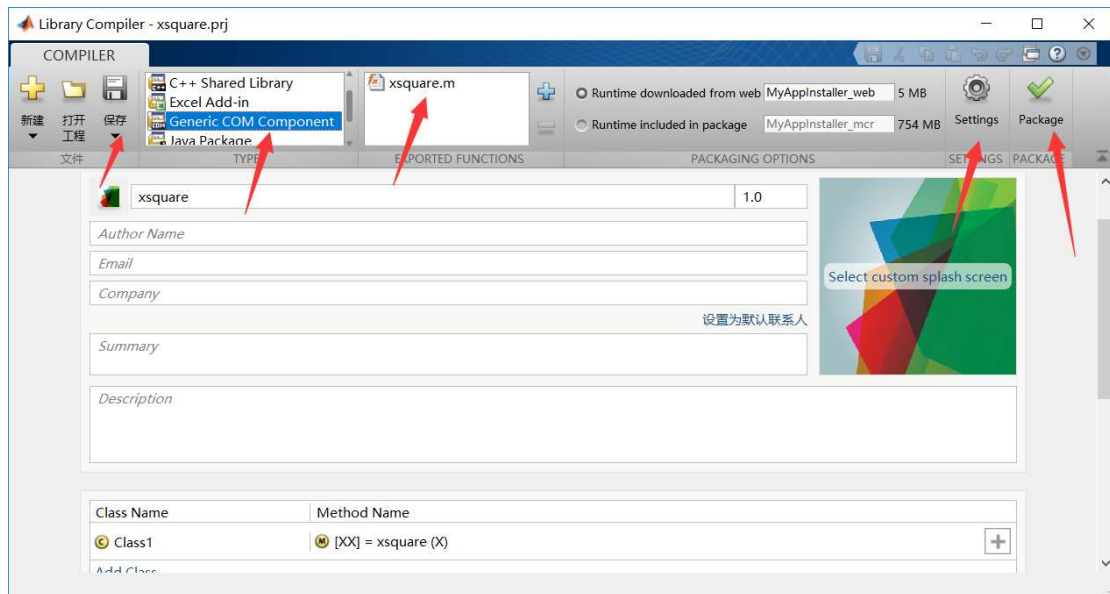


2、点击“应用程序”->“应用程序编译器”。

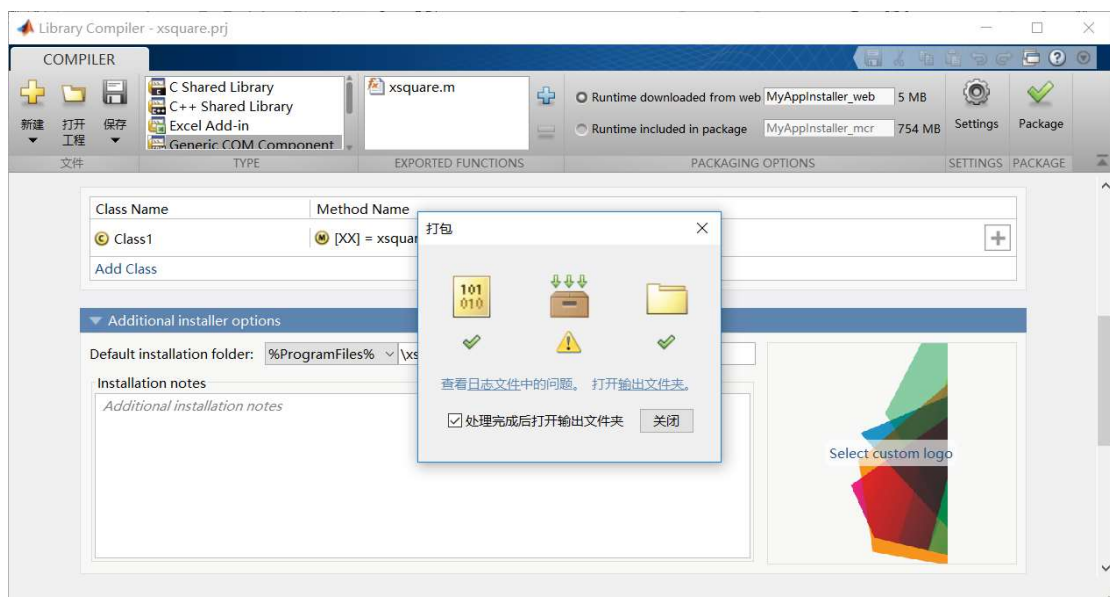


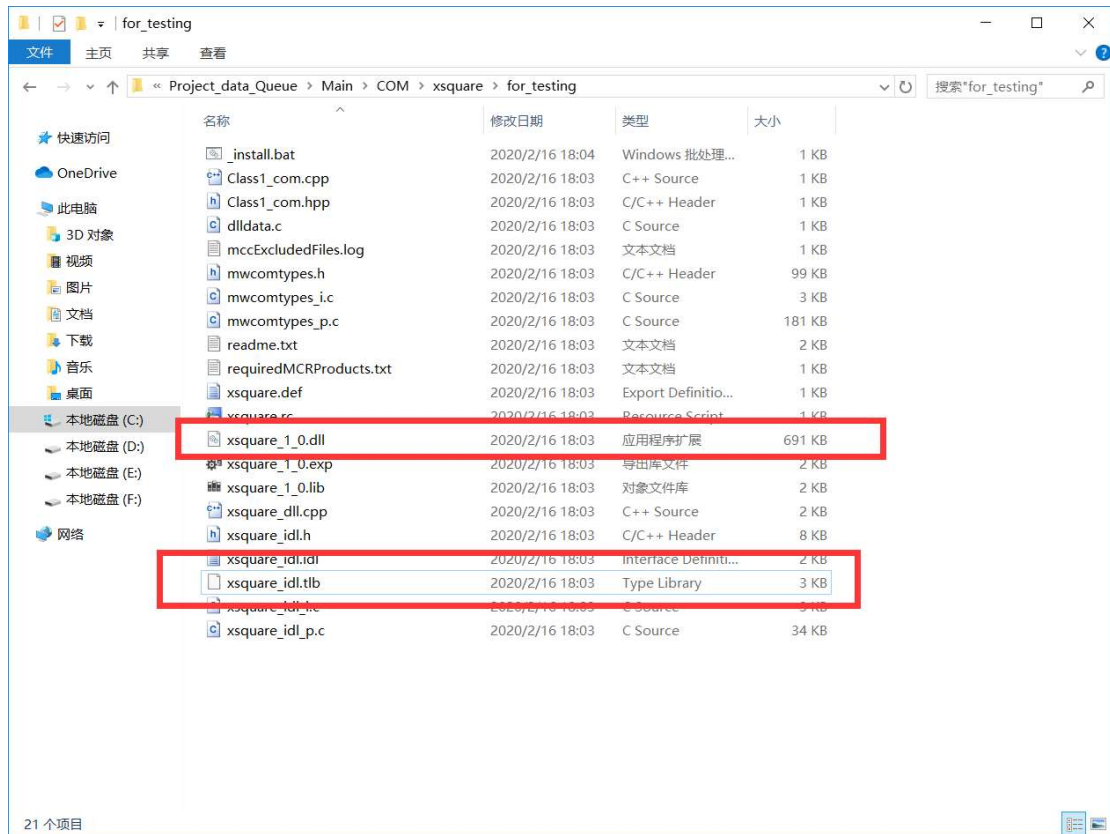
3、在 DEPLOYMENT 窗口中，依次点击“NEW”->“Library Project”->“Generic COM component”-> 添加需要的 .m 文件 ->“Package”。

或者添加写的函数.m → 将项目另存为一个新的.prj → 选择 Generic COM component → Package。（推荐）



4、等待打包完成，MatLab 会自动跳转到生成的文件夹，在 for_testing 文件夹下可看到生成的.dll 文件。





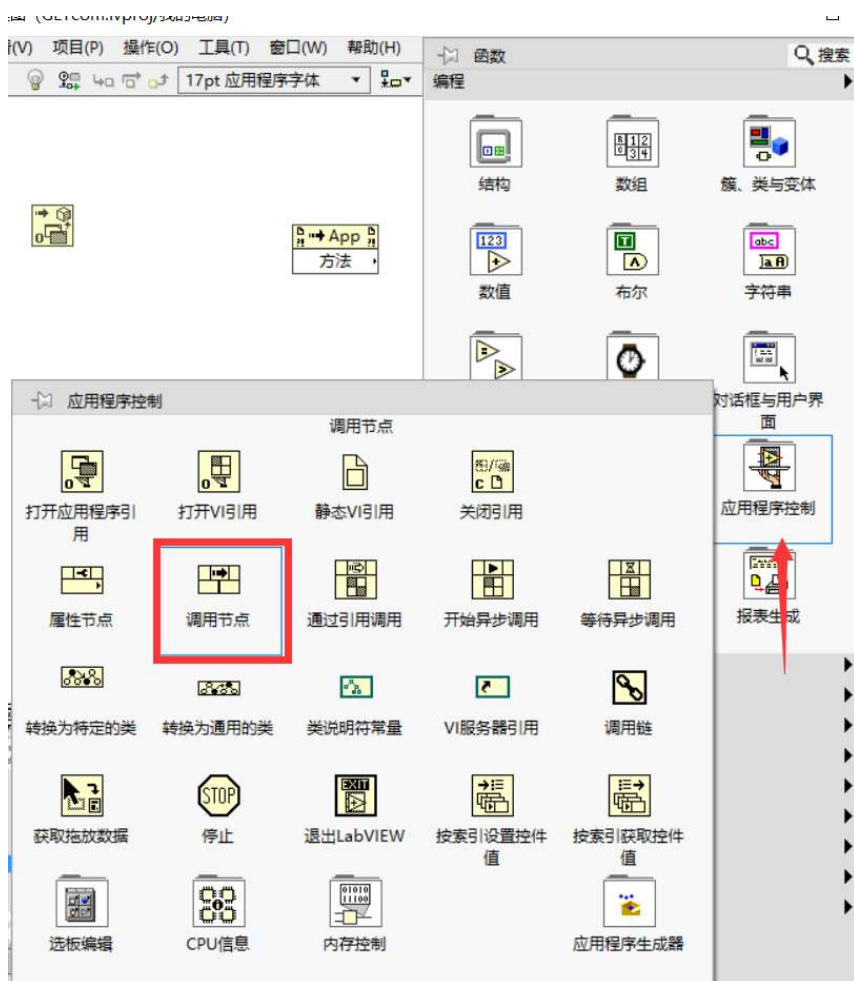
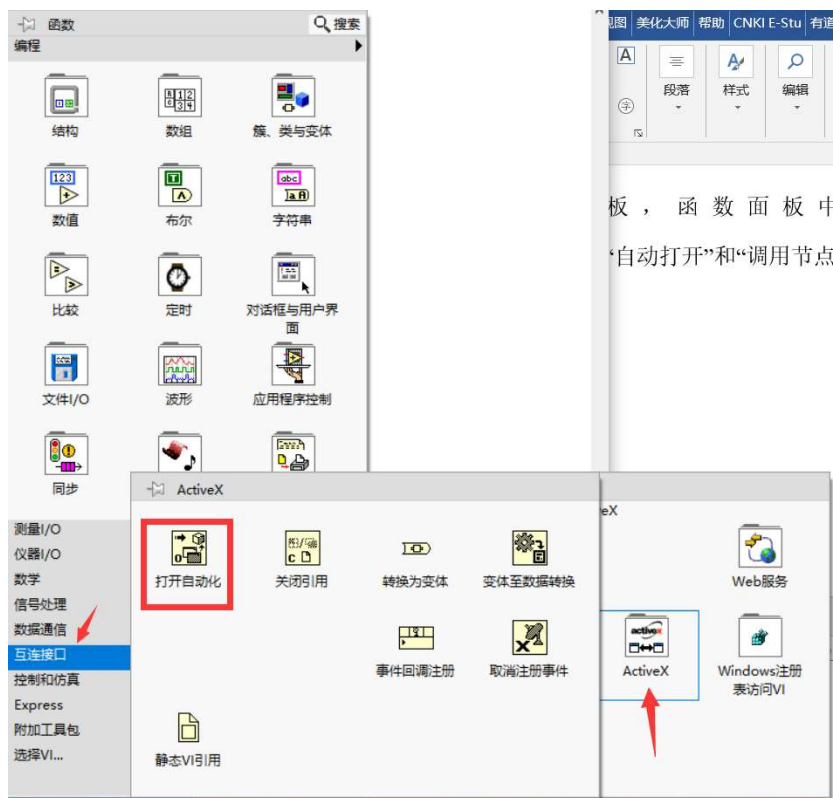
我在编译的时候有警告，经过查看，

警告: Component "xsquare_1_0" will be registered for the current user only.

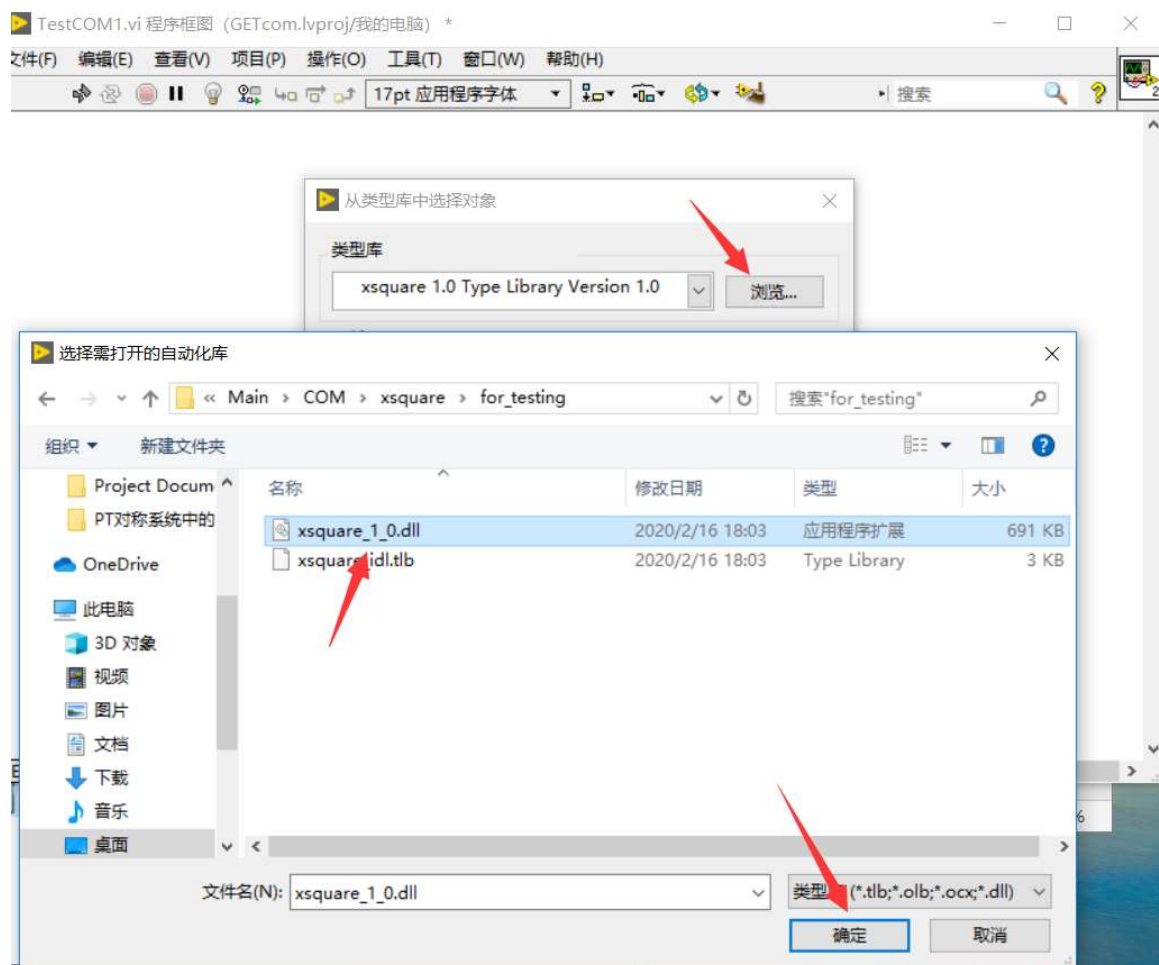
对于 COM 的调用没有影响。

四、LabVIEW 中调用 COM 组件

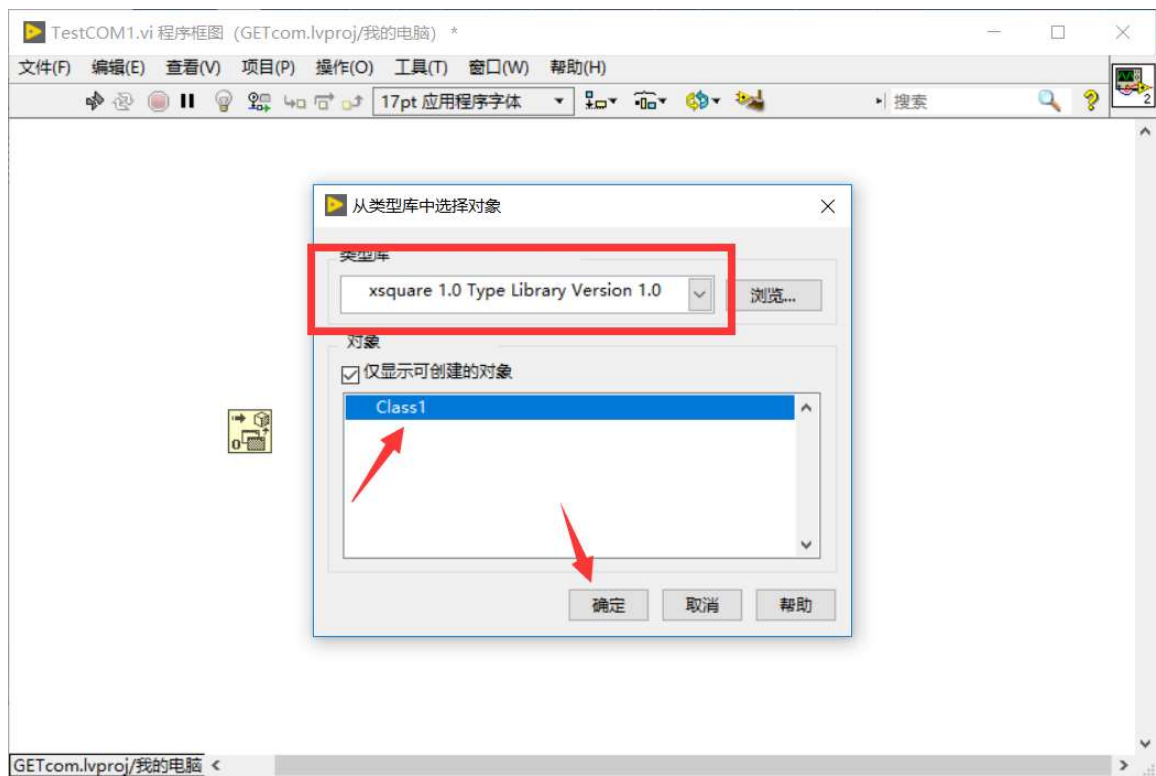
1、LabVIEW 后面板，函数面板中依次点击“Connectivity”->“ActiveX”，将“自动打开”和“调用节点”函数拖入后面板。



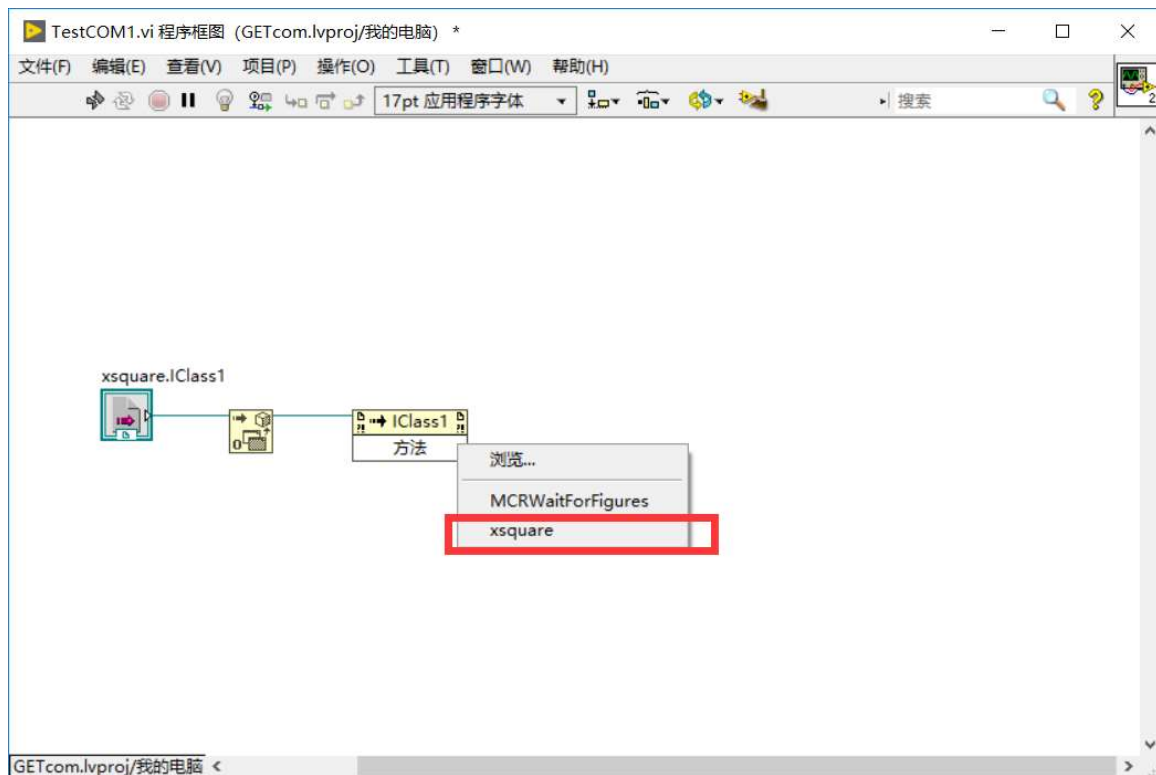
2、在“自动打开”函数上右击，浏览到 MatLab 生成的.dll 文件。



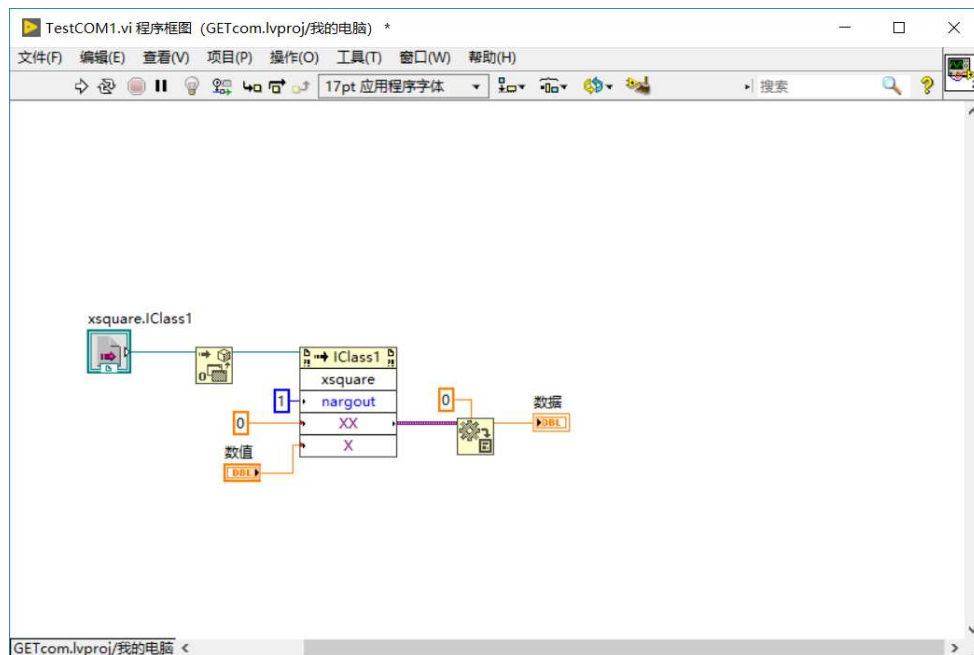
3、已自动定位到类库，选择库中的对象，确定。



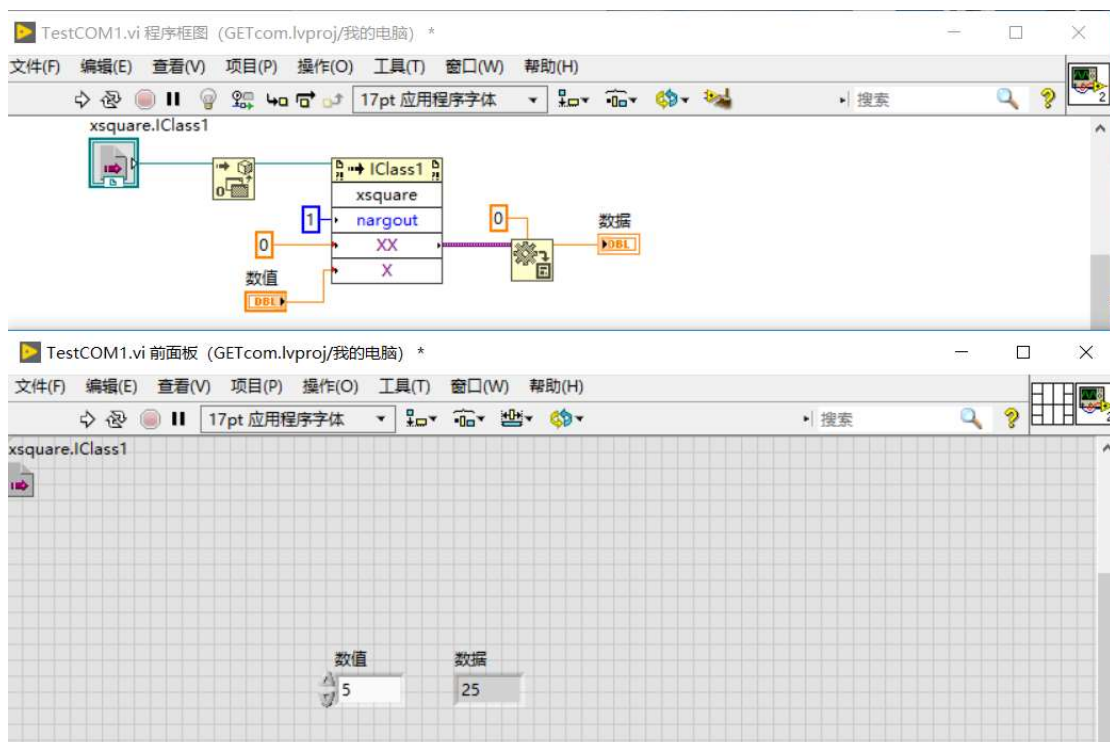
4、连接调用节点的引线，调用节点的方法中选择需要的函数（本例中是 xsquare 函数）。



5、给函数的输入变量和输出变量赋值，可以看到变量的名称与我们在 MatLab 中建立的函数的变量名称完全一致，但多了一个变量 `nargout`，这个变量是指输出变量的数量，一般为定义函数有几个输出就为几，本例中为 1。



6、前面板输入参数，运行，结果完全正确。



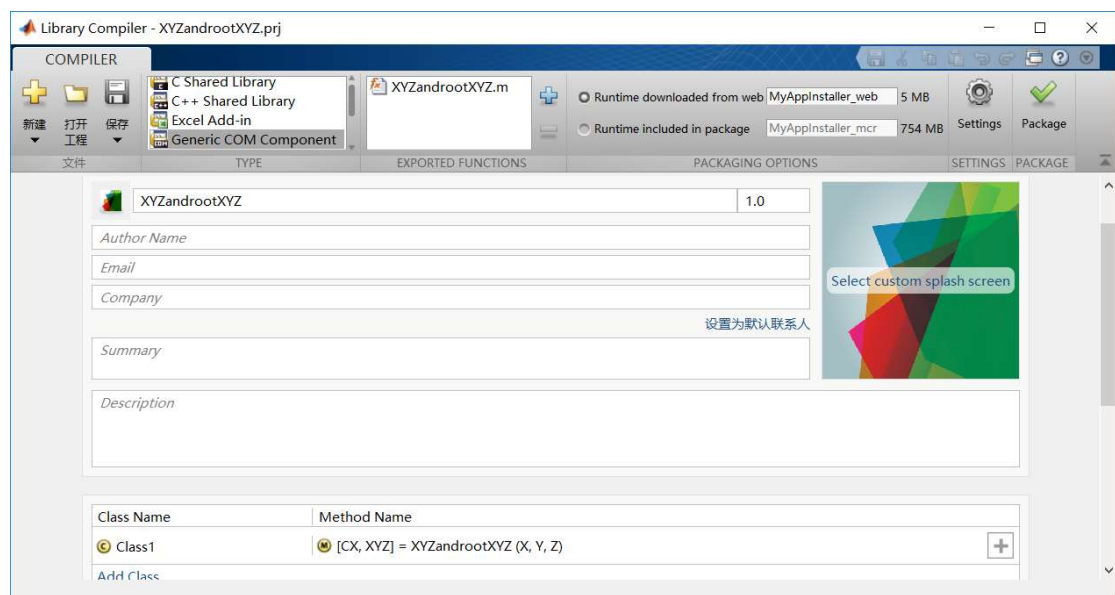
五、多输入多输出的 MATLAB 函数

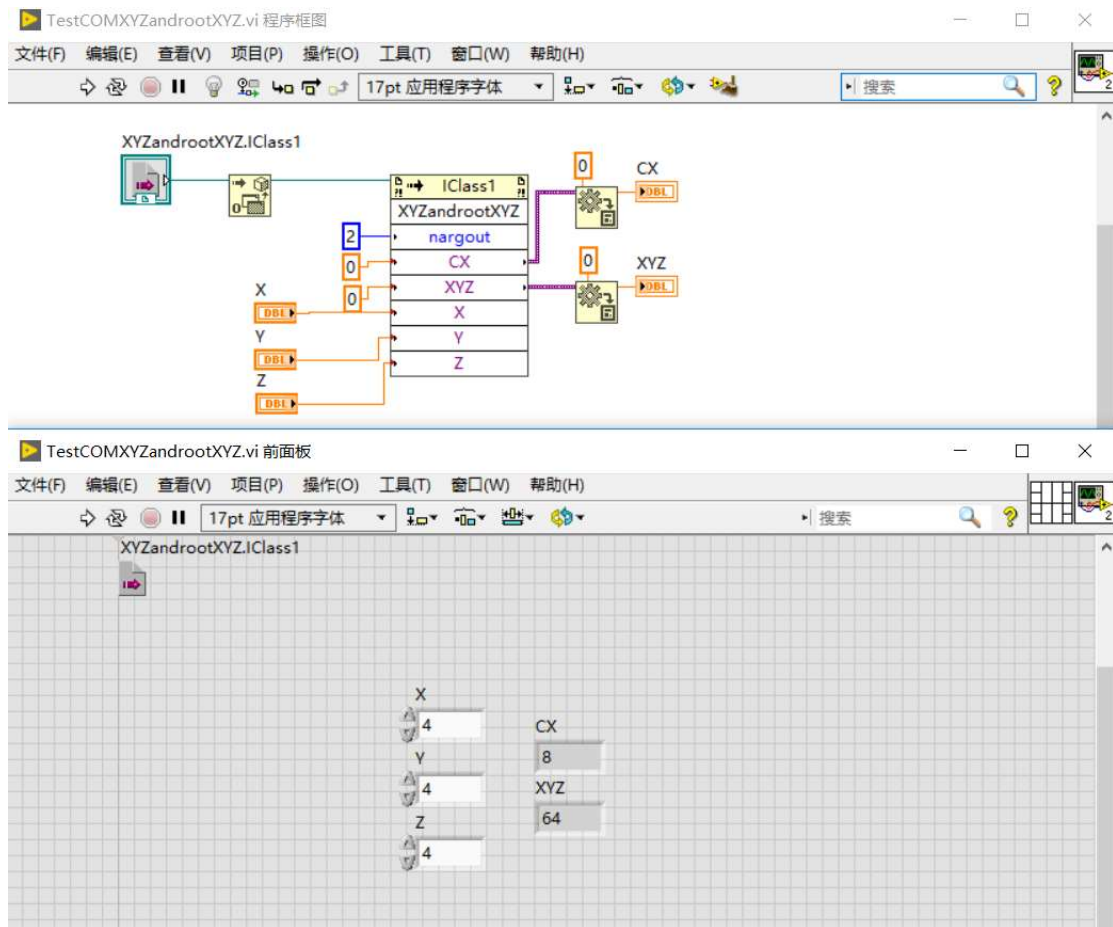
下面写一个多输入输出的函数 demo（多输入多输出）

```
% test function
% [CX ,XYZ ]= XYZandrootXYZ(4,4,4)
% We can get CX=8, XYZ=64

function [CX ,XYZ ]= XYZandrootXYZ(X,Y,Z)
% main function
XYZ = X.*Y.*Z;
CX=sqrt(XYZ);
end
```

```
1 % test function
2 % [CX ,XYZ ] = XYZandrootXYZ(4,4,4)
3 % We can get CX = 8, XYZ = 64
4
5 function [CX ,XYZ ] = XYZandrootXYZ(X, Y, Z)
6 % main function
7 XYZ = X.*Y.*Z;
8 CX = sqrt(XYZ);
9 end
10
11
```





最后，我们得出来一种多输入输出的函数调用的办法：在一个脚本文件函数有多个输入和输出，正常生成 MATLAB 的 COM dll，进行编译后形成一个类。在 LabVIEW 中调用时与平时不一样是对于 nargout 等于输出变量的个数即要用变体将变量输出出来的个数。

六、多函数的 MATLAB 程序

下面写一个多函数一输出的 demo

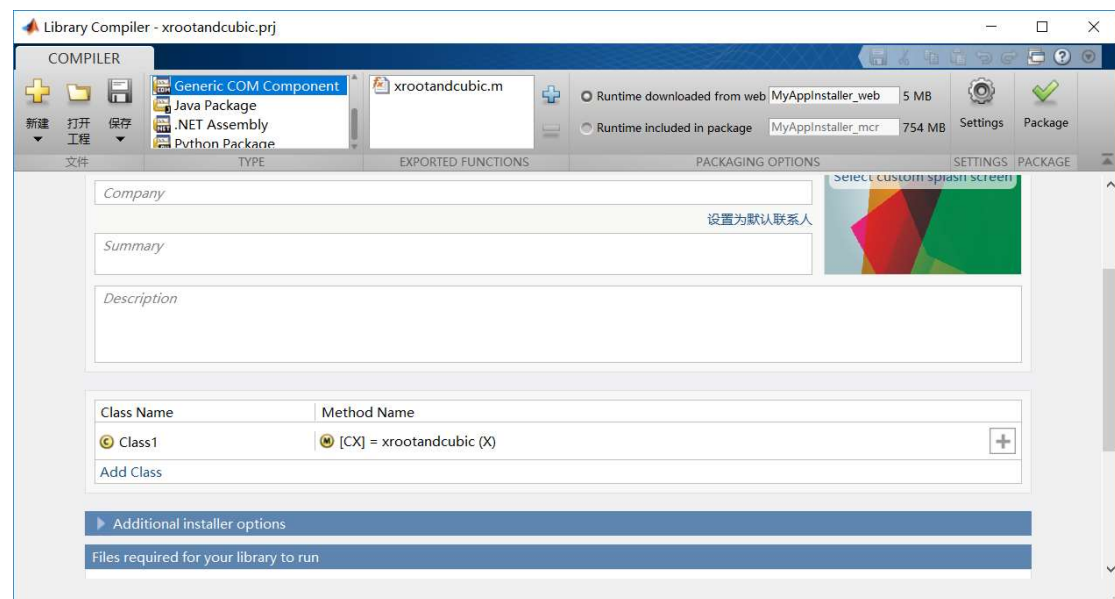
```
% test function
% CX=xrootandcubic(4)
% We can get CX=8

function CX = xrootandcubic(X)
% main function
XXX = X.*X.*X;
```

```
CX=myroot (XXX) ;
end
```

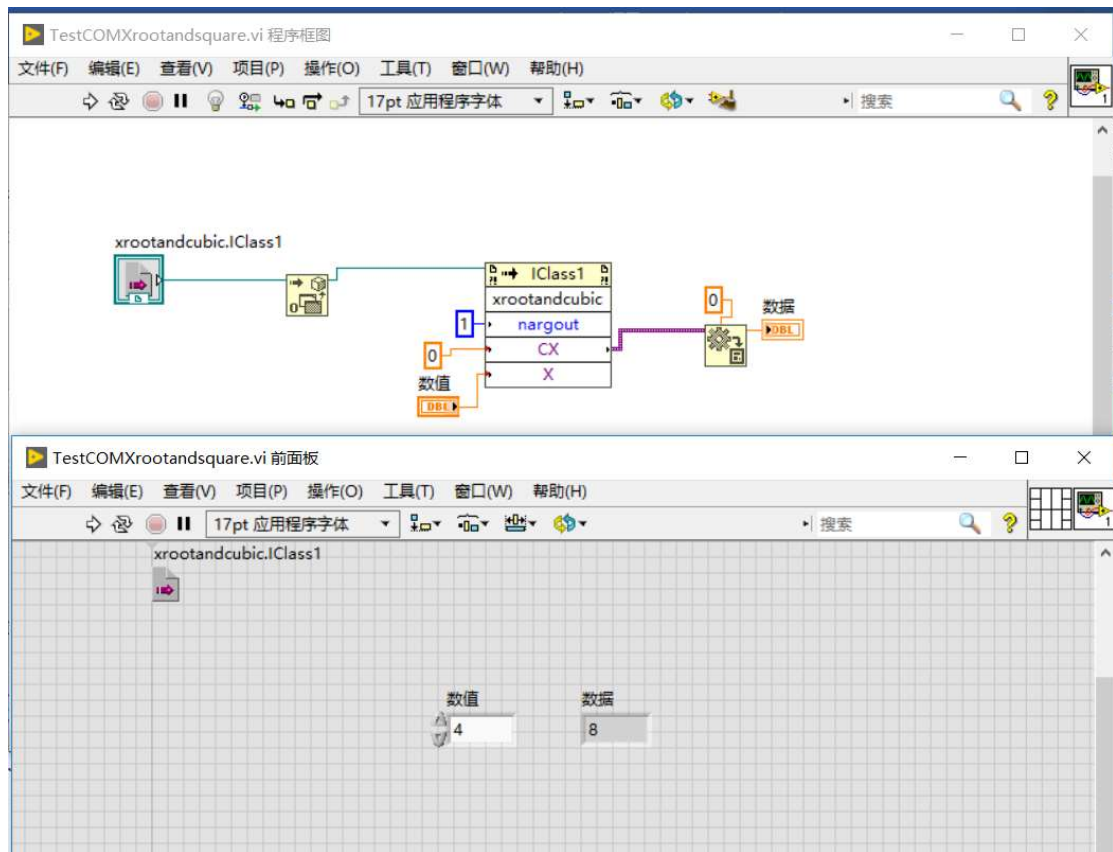
```
function rootX=myroot (Y)
% called function
rootX=sqrt (Y) ;
end
```

```
1 % test function
2 % CX=xrootandcubic(4)
3
4 function CX = xrootandcubic(X)
5 % main function
6 XXX = X.*X.*X;
7 CX=myroot(XXX);
8 end
9
10 function rootX=myroot(Y)
11 % called function
12 rootX=sqrt(Y);
13 end
```



此时 Class Name 中只有一个类，里边只有一个 Method Name，
function rootX=myroot(Y)不会显示。

生成成功后按照步骤四进行 LabVIEW 调用。



最后,我们得出来一种多函数调用的办法:在一个脚本文件(.m)中放入多个函数,名字用主函数命名,其他被调用函数放在末尾。进行编译后形成一个类 Class Name 且只有一个 Method Name。在 LabVIEW 中调用时与平时一样。

七、一个 com 调用多个函数---多个函数一次封装

这里的函数是不存在调用关系的(存在调用关系也可以这样用,亲测可用,我们这里举例以不存在调用关系的单个函数为例)。

```
% mysquare.m
function XX = mysquare(X)
% square
XX = X.*X;
end
```

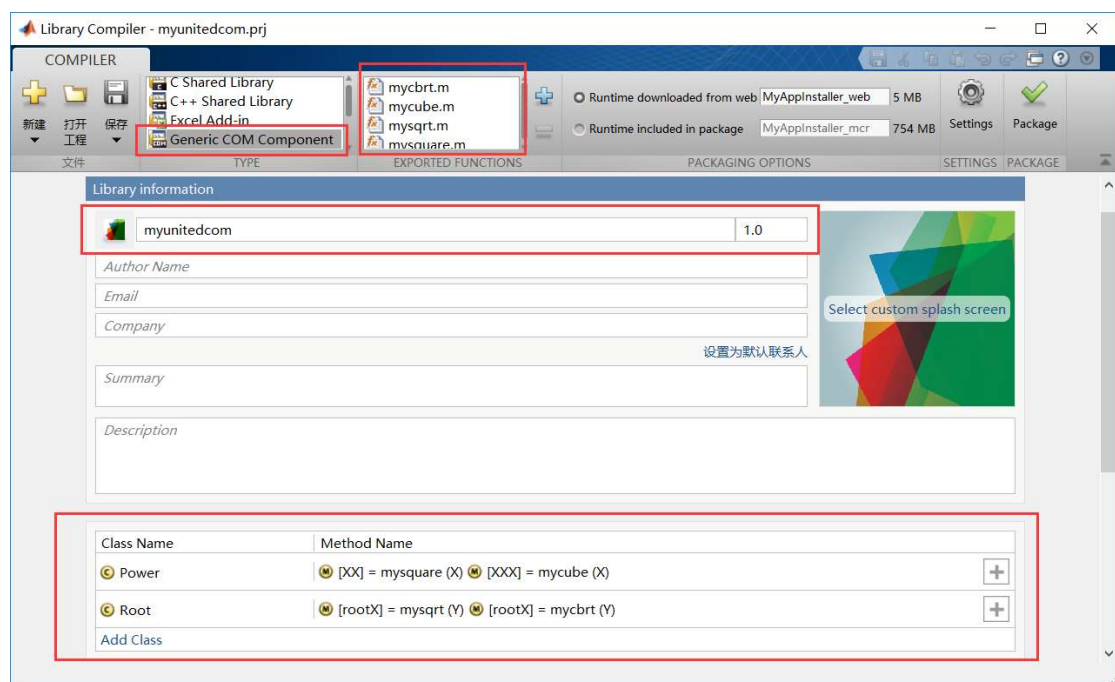
```
% mycube.m
function XXX = mycube(X)
% cube
XXX = X.*X.*X;
end
```

```
% mysqrt.m
function rootX=mysqrt(Y)
% square root
rootX=sqrt(Y);
end
```

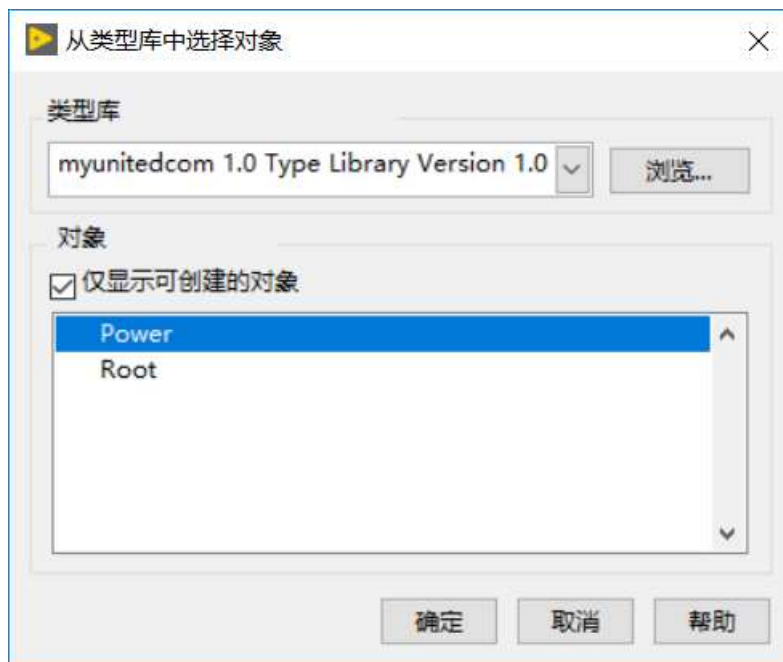
```
% mycbirt.m
function rootX=mycbirt(Y)
% cube root
rootX = nthroot(Y,3);
end
```

将函数分两类，分别为乘方(mysquare、mycube)和开方(mysqrt、mycbirt)，进行 COM 编译。

Tips: 可改 Class Name、Method Name 和 Name of the library 加强可读性。如下图：

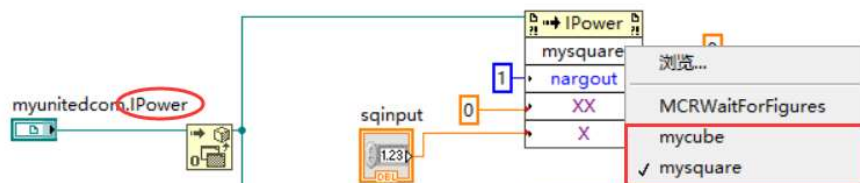


我们生成成功后在 LabVIEW 中进行调用，可知共分为两个对象 (Class)，如下图：

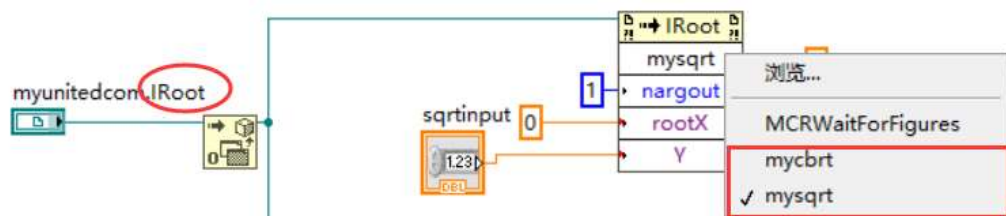


并且每个 Class 里有两个调用节点(Method)，如下图：

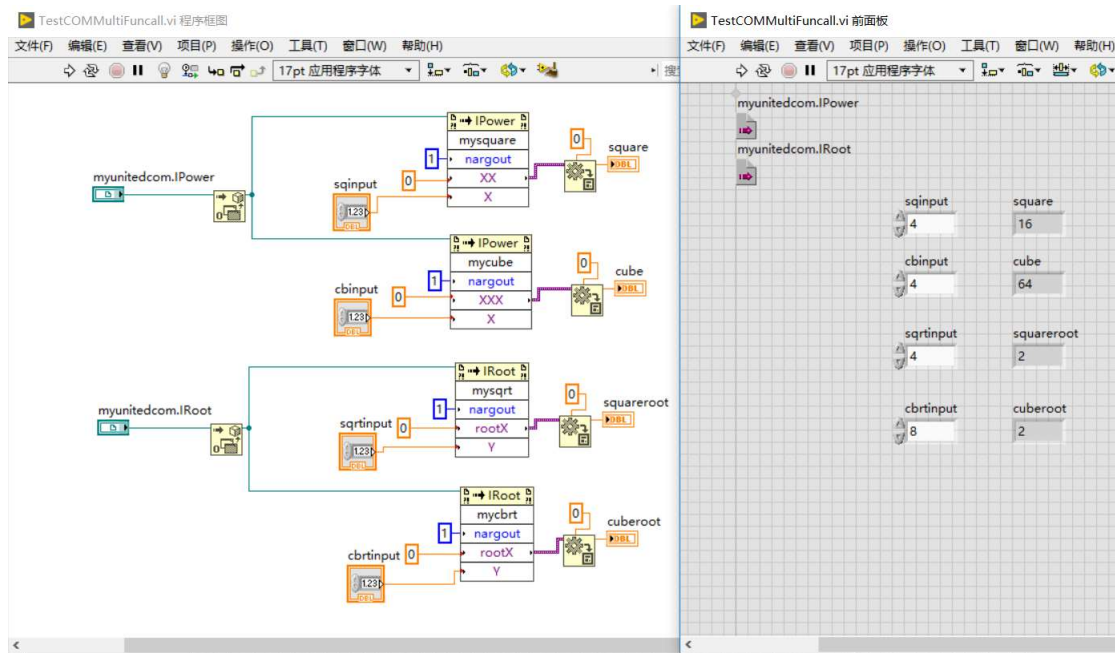
Power: mysquare、mycube



Root: mysqrt、mycbrt



将所有调用节点都调用出来，如下图：



MATLAB 的 Class Name 对应的 LabVIEW 的类型库中可选择的对象 Class； MATLAB 的 Method Name 对应的 LabVIEW 的调用节点；不用区分主函数和被调用函数。

我们根据这个想法可以在实际运用中对 Class 和 Mehtod 进行分类，并封装到一个 com 中。

八、最后四点比较有用的说明

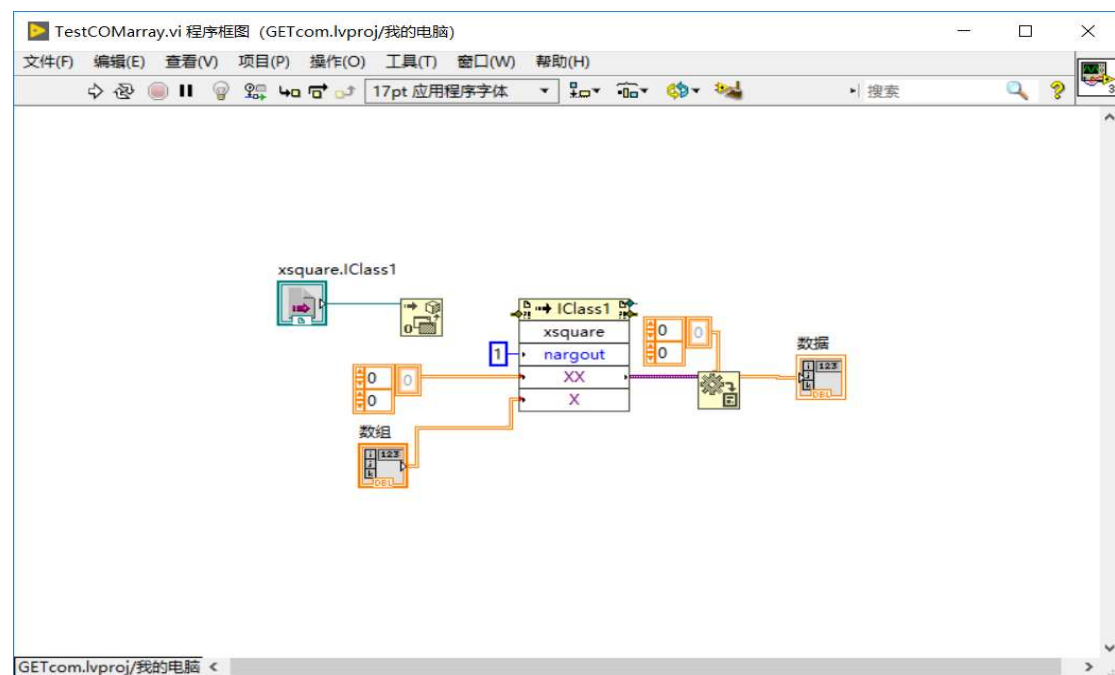
1、MATLAB 生成 Generic COM Component 后只需要留下的是
`...\for_redistribution_files_only\xrootandcubic_1_0.dll`
 （for_redistribution_files_only 里要调用的 dll）与 `...\for_redistribution\MyAppInstaller_web.exe` 和 `\for_redistribution_files_only\install.bat`。

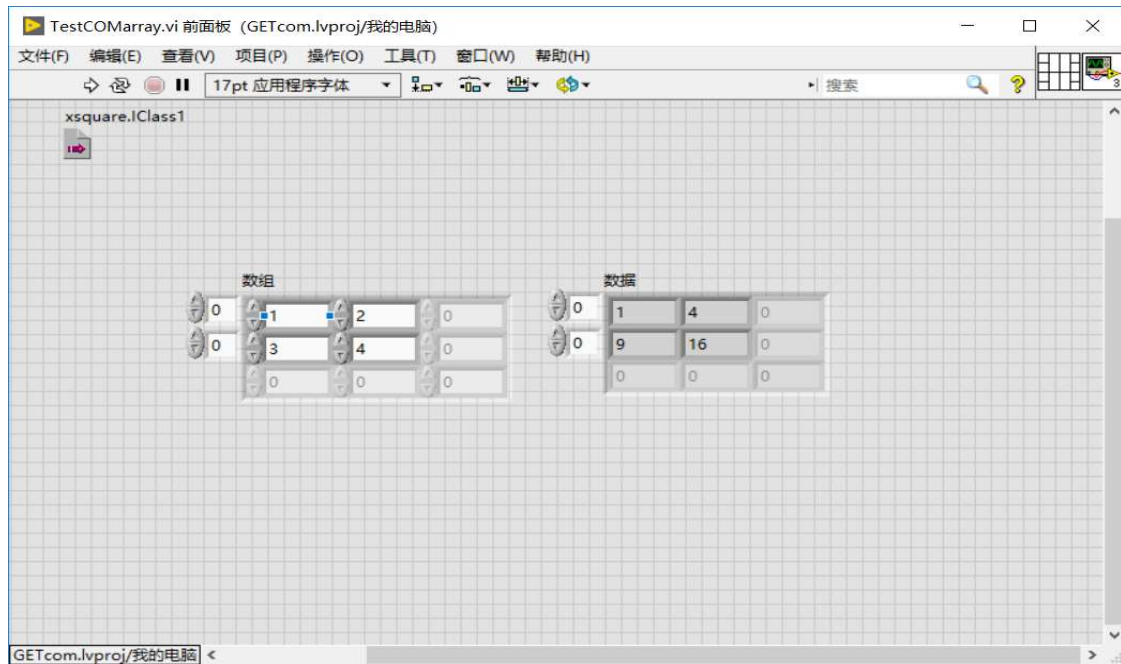
其中 dll 用来调用，exe 和 bat 用来更换电脑后，安装 dll 的运行环境。

2、如果要使用数组作为变量进行计算，必须使用二维数组！即使是一维数组也要转化成二维数组！这是 MatLab 本身的数据类型所决定的。

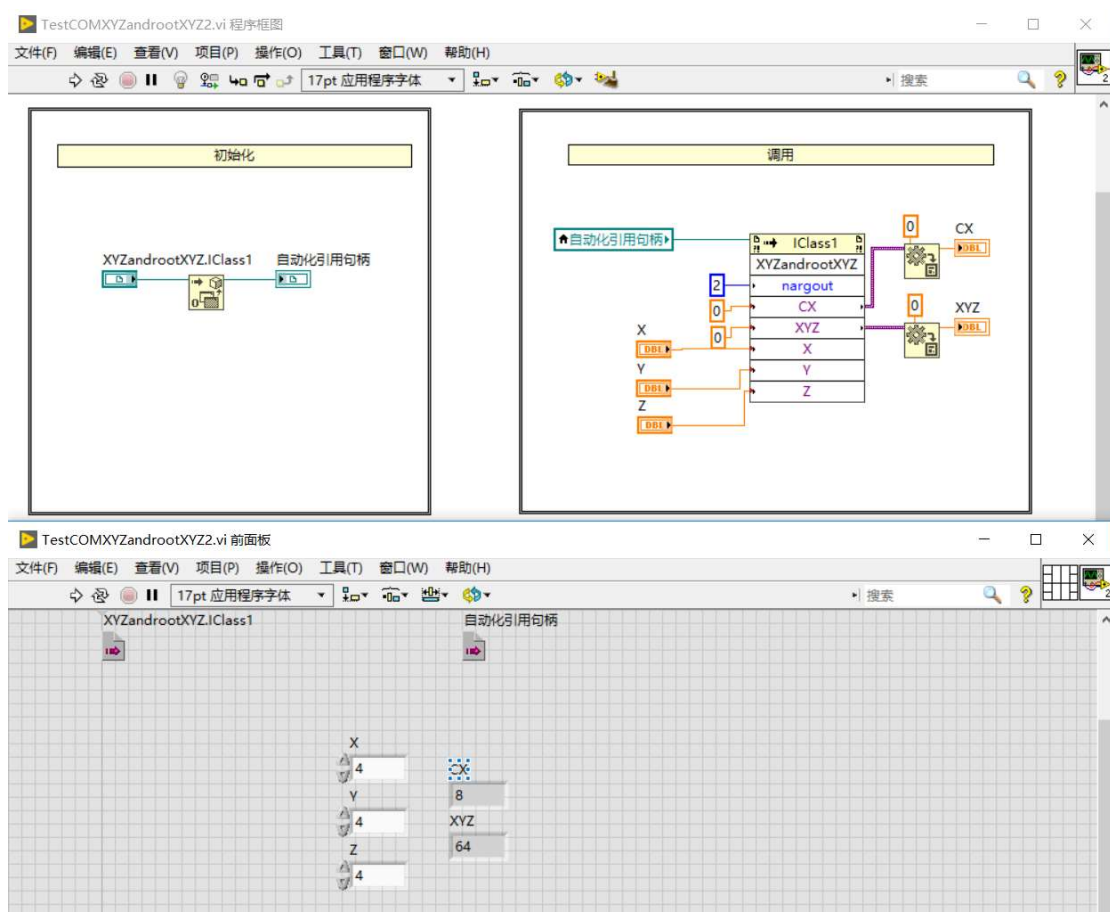
注：此时 MATLAB 的 function 也不用变，沿用原来的 dll，不用定义 X 和 XX 的数据类型。只需要把输入的 X 定义为数组，输出的 XX 也定义为数组即可。

```
function XX = xsquare(X)
XX = X.*X;
end
```





3、对于运行速度方面的优化，COM 只需要自动打开一次，如果这个类需要反复调用，只需要将调用节点放在循环内，这样会加快了运行速度。其中最费时间的步骤就是打开 COM。



对这个 LabVIEW 程序进行优化：分成初始化和调用两部分。

这样我们能够有效的融合在生产者消费者模型中，在程序运行之初的消费者循环中初始化条件下进行自动打开类。在消费者循环中运算的条件下进行调用类计算。

4、最后，小声的说一句：

作者：时断

LabVIEW 开发交流群：240870480，可加群互相交流。

第一次写这方面文档有笔误和写的词不达意之处见谅。

参考来源：

<https://jingyan.baidu.com/article/5970355287bd448fc0074086.html>

如果想要加快开发速度，可使用 NI 的 LabVIEW Comms，详情见 https://www.ni.com/zh-cn/support/documentation/supplemental/18/interfacing-labview-communications-to-matlab--software.html?_t=b1581293779