**使用说明**

1. **二维全matlab版本程序**

直接运行即可，可自动完成前处理、时间推进和后处理的全过程。

1. **三维全matlab版本程序**

直接运行即可，可自动完成前处理、时间推进和后处理的全过程。

1. **二维matlab与fortran配合版本程序**
2. 运行matlab程序的前处理部分，将得到一系列的txt文件，这些文件默认会放在matlab代码所在的文件夹中
3. 将txt文件全部拷贝到fortran代码所在的文件夹中，运行fortran代码，将会得到 output.txt 文件，此文件将默认放在Fortran代码所在的文件夹中
4. 将output.txt 文件拷贝到matlab代码所在文件夹中，运行matlab代码的后处理和绘图部分，即可将求得的结果可视化
5. **三维matlab与fortran配合版本程序**

（1）运行matlab程序的前处理部分，将得到一系列的txt文件，这些文件默认会放在matlab代码所在的文件夹中

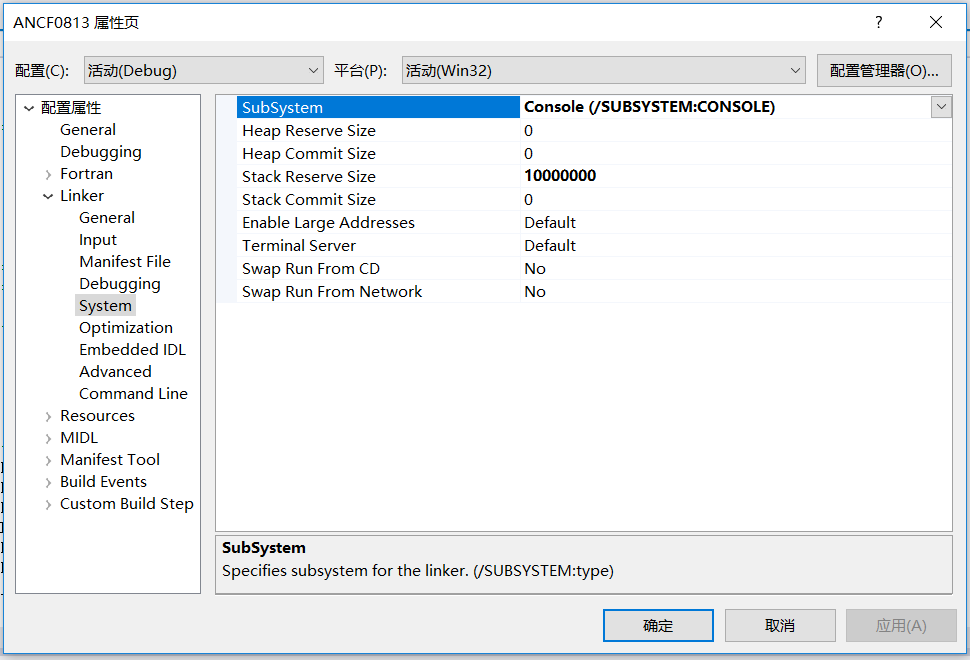
（2）将txt文件全部拷贝到fortran代码所在的文件夹中，运行fortran代码，将会得到 output.txt 文件，此文件将默认放在Fortran代码所在的文件夹中

（3）将output.txt 文件拷贝到matlab代码所在文件夹中，运行matlab代码的后处理和绘图部分，即可将求得的结果可视化

注意，因为三维的矩阵比较大，在读入的时候可能会出现内存溢出的情况，报错为Program Exception - stack overflow，对此可以进行以下操作：

项目->属性->Linker->System->Stack Resrve Size->10000000

如图：



（没有必要必须是10000000，只要是一个很大的数就可以）

**五、二维程序静态算例**

整套验证在matlab中实现，直接运行即可

**六、三维程序静态算例**

整套验证在matlab中实现，直接运行即可

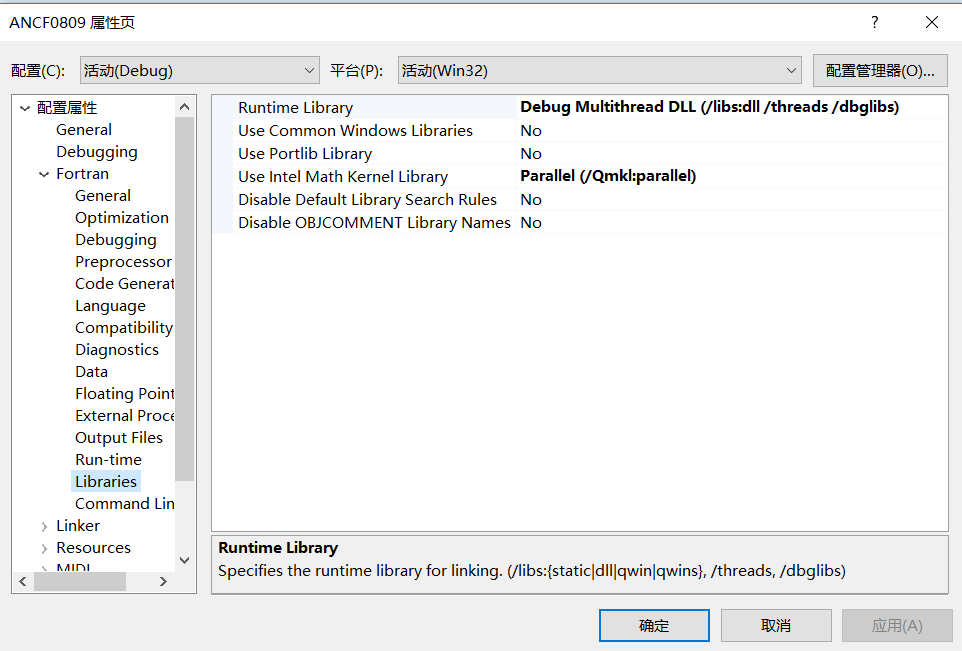
**七、其他扩展内容**

（1）MKL库的添加方法：

从fortran页面选择

项目->属性->Fortran->Libraries->Use Intel Math Kernel Library->Parallel

即可，如下图：



（2）Open MP的环境配置及简单操作方法可以参考网址：

<http://blog.sina.com.cn/s/blog_aed5bd1d0102vpk8.html>

另外有<https://blog.csdn.net/dengm155/article/details/78837408> 也可以作为参考。

但是在我们的尝试中使用openMP似乎没有变快。