

JMCT-FISPACT耦合程序用户手册

注：如果您使用的是windows下的二进制发布包，您可以跳过步骤一，直接从步骤二开始。如果您希望直接运行源码，那么请从步骤一开始。

一、运行环境

本程序需要在Python环境下运行。Python是当今最流行的程序设计语言，它是一种解释型、交互式、面向对象、动态数据类型的高级程序设计语言。我们需要配置好合适的Python环境以供本程序正常运行。

1. Python的安装

Python的[官方网站](#)给出了Python的详细介绍，并提供了在各个平台下的安装版本和下载渠道。用户可以在自己计算机的终端输入"python"命令查看计算机是否已安装Python，以及安装的版本（进入终端：Windows下，进入"命令提示符"；类Unix下，进入"Terminal"）。

注意到Python3和Python2是不兼容的，我们建议安装Python3.6及以上版本。

在Linux平台下安装Python:

linux一般会自带一个python3和一个python2，无需额外安装

在终端键入

```
python3 -V
```

查看当前默认的python版本，不低于3.6为佳

linux下强烈建议使用python虚拟环境以不影响系统正常功能，创建并使用虚拟环境查看[这里](#)

您也可以使用Anaconda虚拟环境，Anaconda教程查看[这里](#)

在Windows平台下安装Python:

通过浏览器访问 Python[官方网站](#)

- 在下载列表中选择Window平台安装包，包的格式为：python-XYZ.msi 文件，XYZ 为你要安装的版本号。
- 下载后，运行下载包，进入Python安装向导，按提示操作。

安装非常简单，我们建议勾选上"添加到环境变量"选项，以便从命令提示符进入Python交互式界面。

在MAC Os平台下安装 Python:

较新的Mac Os系统都自带Python环境；也可以在[这里](#)下载最新版本安装。

2. 配置所需要的软件包

运行本软件，需要配置 **pyqt5** 和 **JFlink** 这两个python的软件包。可以采取pip安装的方式来获取所需要的包。其中 **JFlink** 为源代码的一部分，会随软件包提供，也可用以下方式从[pypi](#)下载。

pip是Python官方推荐的包管理工具。某些编译环境（如Visual Studio）也内置有便捷的包管理工具，从那里安装也可以取得同样的效果。

pip安装:

在python3.6以上的版本里, pip包已包含在其中并被安装好。在终端中键入"pip"命令, 即可查看到pip命令的简要用法。

在正常的情况下, 从操作系统进入终端后, 键入:

```
pip install somepackage
```

即可安装所需要的包, 其中"somepackage"是所需要的包的名称。

1. 如果您希望使用程序包自带的JFlink包

在这里, 我们需要的包是pyqt5和lxml, 所以需要分别键入命令:

windows:

```
pip install lxml  
pip install pyqt5
```

linux:

```
sudo apt install pip3  
pip3 install lxml  
pip3 install pyqt5
```

2. 如果您希望从pypi获取最新的JFlink包

在这里, 我们需要的包是pyqt5和JFlink(事实上, lxml会被自动安装), 所以需要分别键入命令:

windows:

```
pip install JFlink  
pip install pyqt5
```

linux:

```
sudo apt install pip3  
pip3 install JFlink  
pip3 install pyqt5
```

至此, 本程序的运行环境配置完成。

二、运行耦合程序

在我们提供的软件包里面, 进入JFmddler-master目录, 运行main.py, 即可进入程序窗口页面。

1、JMCT-FISP的转换

打开程序后，在"文件转换"选项下选中"JMCTtoFISPACT"，然后进行相关的配置（黑体为必选项，下同）：

- **JMCT输出文件位置：**

这里是JMCT输出文件所在的目录。在系统目录中进入到所需要转换的JMCT输出文件（文件格式为.OUT）所在的目录并选中它即可配置完成。此文件将在接下来的计算中被读取。设置好的目录应类似于："C:/JMCT-FISPACT/JFmiddler-master/testcase/input/neutron.OUT"

- **GDML文件位置：**

这里是GDML模型文件所在的位置，文件格式为.JDML。此文件描述了粒子输运模型的几何信息和材质信息，将在接下来的计算中被读取。设置的方法同上例，设置好的目录应类似于："C:/JMCT-FISPACT/JFmiddler-master/testcase/input/Model_test.gdml"

- **FISPACT工作目录：**

参见 [Q&A](#)。

- **读取模板：**

此模式下，点击"读取模板"，4个FISPACT输入文件模板就被导入并覆盖默认的模板，然后将显示在下方的文本框中前四个标签内，可以方便地编辑和修改。

注意：

FISPACT模板文件应放置在FISPACT工作目录下

- **光子单位时间产额：**

光子单位时间产额是光子源的固有属性，用于计算光子通量。将已知的光子单位时间产额以科学记数法输入，形如：7.8E+18即可。

完成以上设置后，点击"开始"按钮即可开始进行文件转换，转换进度在下方进度条显示。详细的操作步骤都将在下方的文本框中显示。

2、FISP-JMCT的转换

打开程序后，在"文件转换"选项下选中"FISPtOJMCT"，然后进行相关的配置：

- **JMCT输出文件位置：**

这里的设置与JMCT-FISP完全相同，参见"二、运行耦合程序-JMCTtoFISP-JMCT输出文件位置"。如在连续的计算中，提前选中了"复用文件同步"，则此项会自动设置完成。

- **FISPACT工作目录：**

参见 [Q&A](#)。

- **JMCT模板文件位置：**

这里需要设置JMCT模板文件的位置，进入其所在的目录并选中JMCT模板文件(文件格式：.input)即可，设置好的目录形如"C:/JMCT-FISPACT/JFmiddler-master/testcase/input/model.in"

JMCT模板文件记录了粒子输运模型的几何信息，将在接下来的计算中被读取。

- **光谱无穷大等效值：**

光子无穷大等效值用于从fispact到jmct的计算中.fispact输出的光谱最高能量为 $14+\infty$ ，但是jmct中光谱输入值必须为有限值，所以正无穷必须用某个等效值代替，默认值为20.

- 读取模板：

此模式下，点击"读取模板"，JMCT输入文件模板就被导入，然后将显示在下方的文本框中jmct标签内，可以方便地编辑和修改。

完成以上设置后，点击【开始】按钮即可开始进行文件转换，转换进度在下方进度条显示。

3、其他选项

- 复用文件同步：

在连续的计算中，若选中"复用文件同步"，则在不同的文件转换和程序调用中使用到的同一文件、目录将被自动设置好。

- 保留JMCT模板：

若勾选此选项，则完成计算后，原先的JMCT模板将依然被保留，新生成的JMCT输入文件，其文件名将会增加_new后缀。

- 清屏：

在右页面，有一个【清屏】按钮，用于将当前所有操作日志清空。

- 文本框之一：

collapx.iarrayx.iinput.iprintlib.ijmct

```
MONITOR 1
COLLAPSE 175
FISPACT|
* {title}
END
* END OF RUN
/*
```

这个文本框负责显示一些预读取的文件信息，其中：collapx.i; arrayx.i; input.i; printlib.i 均为FISPACT输出文件的文件模板，jmct中显示的是JMCT的模板文件，后者需点击"导入模板文件"后方可显示。

- 文本框之二：

位于文本框之一的下方。启动程序后，操作日志将自动在此文本框中被显示。

三、程序调用

在启动程序后，点击"程序调用"栏目。

1、调用FISPACT

点击"CallFISP", 并进行相关设置:

- **FISPACT执行文件:**
这里应设置为FISPACT可执行文件的位置。进入FISPACT程序的工作目录, 选中它的可执行文件即可。
- **EAF数据库目录:**
这里应设置为FISPACT运行所需要的EAF数据库所在的目录, 进入目录并选中即可。
- **FISPACT工作目录:**
这里应设置为FISPACT主程序所在的目录。若在此之前进行过文件转换操作并勾选了"生成调用语句", 则此目录将被自动设置好。
- **能群:**
这里选择的是不同的能群划分格式, 我们可以选取五种低能标准能群: WIMS(69), GAM-II(100),XMAS(172),VITAMIN-J(175), TRIPOLI(315)和两种高能标准能群: VITAMIN-J+(211)和TRIPOLI+(351)。
- **权函数:**
权函数和能群一起描述粒子群的性质, 有FLA/FLT、FIS、FUS三种权函数供选择。
- **粒子类型:**
这里选取的是粒子的类型, 有光子 (p)、中子 (n) 和 (d) 三种粒子选项。

配置完成后, 点击"开始"即可调用FISPACT程序。

2、调用JMCT

点击"CallJMCT",并进行相关设置:

- **JMCT输入文件:**
这里需要设置JMCT的输入文件, 文件格式为 *.input*。
点击"开始"即可调用JMCT。

四、相关文件说明

1、JMCT模板文件

本模板文件用于生成JMCT输入文件, 建议后缀名为 *.input*, 模板文件应与最终生成的输入文件保持完全一致, 除非必要的地方使用关键字代替。

可用关键字:

关键字	描述
{source}	应位于Source代码块中, 程序将用计算出的粒子源信息替代此处

2、FISPACT模板文件

本模板文件用于生成FISPACT输入文件, 建议后缀名为*.i*, 模板文件应与最终生成的输入文件保持完全一致, 除非必要的地方使用关键字代替。

可用关键字:

关键字	可用模板文件	描述
{title}	all	默认标题，一般为【Irradiation of ...】
{mass}	input.i	计算出的物质质量，一般不使用或用于MASS标签
{density}	input.i	计算出的物质密度，一般用于DENSITY标签
{flux}	input.i	计算出的中子通量，一般用于输运过程的FLUX标签
{elements}	input.i	计算出的元素信息，一般用于MASS标签，在{mass} 之后

3、ini文件

wiz.ini为程序记录的某些可能需要多次使用的信息，所在位置应为"./tmp/wiz.ini"，在程序启动时读取，程序关闭时储存，一个典型的wiz.ini文件如下所示：

```
[turn]
jpathu =
gpathu =
fpathu =
genrate =
fpathd =
jpathd =
jmodel =

[call]
fispact =
eaf =
case =
jpath =

[global]
logs = 1
```

关键字	自动生成	描述
jpathu	是	储存【JMCT-FISP的转换】中的 【JMCT输出文件位置】
gpathu	是	储存【JMCT-FISP的转换】中的 【GDML文件位置】
fpathu	是	储存【JMCT-FISP的转换】中的 【FISPACT工作目录】
generate	是	储存【JMCT-FISP的转换】中的 【光子单位时间产额】
fpathd	是	储存【FISP-JMCT的转换】中的 【FISPACT工作目录】
jpathd	是	储存【FISP-JMCT的转换】中的 【JMCT输出文件位置】
jmodel	是	储存【FISP-JMCT的转换】中的 【JMCT模板文件位置】
fispact	是	储存【调用FISPACT】中的 【FISPACT执行文件】
eaf	是	储存【调用FISPACT】中的 【EAF数据库目录】
case	是	储存【调用FISPACT】中的 【FISPACT工作目录】
jpath	是	储存【调用JMCT】中的 【JMCT输入文件】
logs	否	用户指定的最大日志储存数量，若超出此值，最旧的日志文件将会被新日志替代

4、log文件

日志文件位于"./tmp/*.log"，*部分一般含有保存日志的时间，日志内保存全部当次运行的操作日志。

注意：使用【重置】清除掉的操作日志不会被保存。

六、开发手册

如果您只是需要使用软件，可以跳过此部分

此部分对于软件包的二次开发有一定帮助

整个耦合软件由后端逻辑软件包JFlink与前端GUI工具JFwizard构成

1、软件架构

后端软件包JFlink

```
.
+-- call.py
+-- model.py
+-- read.py
+-- write.py
+-- __init__.py
```

模块	描述	接口
call	用于调用JMCT和FISPACT的工具集	jmct/fisp
model	用于记录转换过程中的类和必要的常量	无
read	用于读取JMCT和FISPACT输出文件的工具集	readf/readg/readf
write	用于生成JMCT和FISPACT输入文件的工具集	writej/writef
__init__	软件包入口	无

前端界面JFwizard

```
.
+-- tmp
|   +-- wiz.ini
|   +-- log-2018-03-04-11-03-23-857.log
+-- testcase
+-- ICON.ico
+-- main.py
+-- static.py
+-- window.py
+-- worker.py
```


文件/文件夹	文件夹	必需	描述
tmp	是	否	程序运行过程生成的临时文件放置位置
wiz.ini	否	否	程序记录的某些可能需要多次使用的信息
log-2018-03-04-11-03-23-857.log	否	否	程序生成的日志文件，可能不止一个
testcase	是	否	测试文件放置位置
ICON.ico	否	是	软件的图标文件
main.py	否	是	软件的入口文件
static.py	否	是	软件的静态界面文件
window.py	否	是	软件的动态界面文件
worker.py	否	是	软件定义的多线程任务

2、JFlink软件包接口详解

```
call.jmct(info, jinput, gpath='')
```

无返回值

参数	类型	描述	默认值
info	function	回调函数，用于输出提示信息	无
jinput	string	JMCT输入文件.input位置	无
gpath	string	GDML结构文件.gdml位置	""

```
call.fisp(self, info, env, group, indir: Path, _outdir=None)
```

无返回值

参数	类型	描述	默认值
self	class	传入一个类用于临时储存信息	无
info	function	回调函数，用于输出提示信息	无
env	2-elements-list	[fpath, epath]储存可执行文件的二元列表	无
group	3-elements-list	[p, g, w]储存粒子信息的三元列表	无
indir	string	【调用FISPACT】中的 【FISPACT工作目录】	无
_outdir	string	FISPACT中定义的work directory	None

```
read.readj(path, funcTime=None, funcOne=None, interval=100)
```

返回值neutron附在表格最后

参数	类型	描述	默认值
path	string	.input文件的路径	无
funTime	function	回调函数，用于通知上层函数总共需处理的条目数	None
funcOne	function	回调函数，用于通知上层函数目前处理的条目数	None
interval	int	调用funcOne的次数	100
neutron	model.Data	从.input文件中提取出的物质信息	无

```
read.readg(path, funcTime=None, funcOne=None, interval=100)
```

返回值allStructure附在表格最后

参数	类型	描述	默认值
path	string	.gdml文件的路径	无
funTime	function	回调函数，用于通知上层函数总共需处理的条目数	None
funcOne	function	回调函数，用于通知上层函数目前处理的条目数	None
interval	int	调用funcOne的次数	100
allStructure	dict	从.gdml文件中提取出的空间信息	无

```
read.readf(path, maxFlag=20., funcTime=None, funcOne=None, interval=100)
```

返回值allDistributions附在表格最后

参数	类型	描述	默认值
path	string	FISPACT工作目录路径	无
funTime	function	回调函数，用于通知上层函数总共需处理的条目数	None
funcOne	function	回调函数，用于通知上层函数目前处理的条目数	None
interval	int	调用funcOne的次数	100
allDistributions	dict	从FISPACT输出文件.o中提取出的光谱信息	无

```
write.writej(path, text, neutron, allDistributions, split, funcTime=None, funcOne=None, interval=100)
```

无返回值

参数	类型	描述	默认值
path	string	生成的JMCT输入文件.input路径	无
text	string	JMCT模板文件内容	无
neutron	model.Data	物质信息，来自之前的JMCT输入文件.input	无
allDistributions		光谱信息，来自FISPACT输出文件.o	无
split	string	缩进使用内容	无
funTime	function	回调函数，用于通知上层函数总共需处理的条目数	None
funcOne	function	回调函数，用于通知上层函数目前处理的条目数	None
interval	int	调用funcOne的次数	100

```
write.writef(path, genRate, neutron, allStructure, _inputText = defaultInput, _collapxText = defaultCollapx, _arrayxText = defaultArrayx, _printlibText = defaultPrintlib, funcTime=None, funcOne=None, interval=100)
```

无返回值

参数	类型	描述	默认值
path	string	FISPACT工作目录路径	无
genRate	float/int	光子产生速度	无
neutron	model.Data	物质信息，来自之前的JMCT输入文件.input	无
allStructure	dict	从.gdml文件中提取出的空间信息	无
_inputText	string	input.i模板文件内容	model.defaultInput
_collapxText	string	collapx.i模板文件内容	model.defaultInput
_arrayxText	string	arrayx.i模板文件内容	model.defaultArrayx
_printlibText	string	printlib.i模板文件内容	model.defaultPrintlib
funTime	function	回调函数，用于通知上层函数总共需处理的条目数	None
funcOne	function	回调函数，用于通知上层函数目前处理的条目数	None
interval	int	调用funcOne的次数	100

七、Q & A

Q：如何在启动时不显示cmd终端？

A：将程序运行方式由python修改为pythonw。

Q：FISPACT工作目录指的是哪一级？

A：指最顶层包含所有材料的一级目录。

如以下结构中，FISPACT工作目录指的是fisp。

```
.
+-- fisp
|   +-- AL6061
|       |   +-- arrayx.i
|       |   +-- collapx.i
|       |   +-- input.i
|       |   +-- printlib.i
|       |   +-- fluxes
|   +-- CU
|       |   +-- arrayx.i
|       |   +-- collapx.i
|       |   +-- input.i
|       |   +-- printlib.i
|       |   +-- fluxes
|   +-- SS316
|       |   +-- arrayx.i
|       |   +-- collapx.i
|       |   +-- input.i
|       |   +-- printlib.i
|       |   +-- fluxes
```

Q：源代码丢失了怎么办？

A：JFwizard和JFlink均为开源软件，可以从[github](#)或pypi获取

- [JFwizard主页](#)
- [JFlink主页](#)
- JFlink可以[如此](#)从pypi获取

Q：运行二进制程序出现类似于【无法启动此程序，因为计算机中丢失api-ms-win-crt-runtime-l1-1-0.dll】或类似错误，这是为什么？

A：一般在win7或更低版本windows下运行程序会出现这种问题。因为程序在win10下打包，会依赖一些win10下的动态链接库。建议使用win10运行，但如果您坚持使用win7，您可以选择以下任何一种做法：

- 安装KB2999226、KB3118401更新之一
- 安装Visual Studio 2015
- 使用源码方式运行