[日期]

Limin YU

ECUST

2018-09-06

ORBIT使用手册

程序包安装 3

代码简介 5

参数输入 6

数据输出 10

模型 11

示例 13

附录－运行mex2eqs，产生map01.cdf 14

附录－中性束离子分布 17

# 程序包安装

1、**软件包及下载地址**

编译该程序需要如下五个软件包：

* Orbit: <ftp://ftp.pppl.gov/pub/white/Orbit>
* Netcdf: <http://www.unidata.ucar.edu/packages/netcdf/>
* Ezcdf: <http://w3.pppl.gov/rib/repositories/NTCC/catalog/Asset/EZcdf.html>
* Pspline: <http://w3.pppl.gov/ntcc/PSPLINE/>
* ncl\_ncarg-5.1.0: <http://www.ncl.ucar.edu/Download/>

2、**软件包安装**

以上五个软件包，以源代码方式安装，过程如下：

* Netcdf安装(netcdf-3.6.1)

tar -zxvf netcdf-3.6.1.tar.gz （解压缩）

./configure --prefix=/users/hostname/opt/netcdf-3.6.1（确定安装路径）

Make

make install

其中，configure：获取系统的环境变量，生成配置好的Makefile文件，Make 之前须准备好编译器：g++, ifort或pgf90, configure 会自动找到这些编译器的路径；

./configure --prefix=/users/hostname/opt/netcdf-3.6.1：指定需安装的路径，如果不方便在机器上找到相应的软件包，一般地，可以在自己的目录下，新建一个opt文件夹，专门放置安装个人所需的软件；

Make : 进行编译

Make install：进行安装，路径configure时指定的，默认路径/usr/local安装

* ezcdf安装

新建ezcdf文件夹，把ezcdf.tar.gz放入该文件夹

安装同上

tar -zxvf ezcdf.tar.gz （解压缩）

设置makefile的头文件Make.local：

ezcdf下的share文件夹：

cp Make.local.sample Make.local

make NETCDF\_DIR=/users/hostname/opt/netcdf-3.6.1 FORTRAN\_VARIANT=Porland

或make LIBROOT=/users/hostname/opt/netcdf-3.6.1 FORTRAN\_VARIANT=Porland

编译：指定netcdf路径，其默认路径为 (/usr/local) 以及指定编译器(应与编译netcdf用到的编译器,以及编译ORBIT程序包的编译器都一致)

编译后生成LINUX文件夹，包含lib, mod, test, obj/ezcdf四个次文件夹

最后，把LINUX中的文件安装至指定路径中去，默认为/usr/ntcc

make install PREFIX=/users/hostname/opt/ezcdf

* Pspline安装

tar -zxvf pspline.tar （解压缩）

确定安装路径

./configure - -prefix=/usr1/ntcc

使用其中默认/usr/ntcc（先在usr下建立ntcc）在./share中的Make.local.sample中加

入NETCDF\_DIR=/users/hostname/opt/netcdf-3.6.1，重命名为Make.local。如果用

porland编译器，键入：

make FC=pgf90

make install PREFIX=/usr/ntcc/

* ncl\_ncarg-5.1.0安装

tar -zxvf ncl\_ncarg\_src-5.1.0.tar

确定安装路径，使用其中默认/usr/local/ncarg（先在usr下建立ncarg）

./Configure –v -- prefix=/usr1/local

Make Everything >& make–output &

#开始安装， 安装需要十几分钟且没有安装过程详情显示，要有显示需：

tail –f make-output

库文件在/usr1/local/ncarg/lib

3、**编译和运行orbit**

1. 修改Makefie，具体如下：

# compiler dependencies

F90 = pgf90指定fortran编译命令

NTCCHOME:=/usr/ntcc

NETCDFHOME:=/usr1/local

NCARG\_ROOT:=/usr/local/ncarg

1. 清理旧的编译文件：键入make clean
2. 生成orbit/eqs可执行程序： 键入make FC=pgf90 orbit/eqs
3. 运行eqs：键入 ./eqs

#生成eqs可执行文件，用平衡数据文件map01.cdf通过spline插值来产生orbit所需的平衡数据spdata或用解析平衡通过spline插值来产生orbit所需的平衡数据spdata。其中，map01.cdf由平衡代码，如TRANSP, EEFIT, JSOLVER，产生的平衡数据文件，用mex2eqs软件，转化而得的。

1. 生成orbit：键入make FC=pgf90
2. 运行orbit：键入 ./orbit
3. 神马服务器运行orbit：qsub job.pbs

# 代码简介

Orbit由美国普林斯顿大学PPPL实验室Roscoe B.White开发，是一个根据环位形下导心方程，编写的导心轨道代码，能得到粒子在磁面坐标下位置和速度。导心方程只依赖于磁场大小。因此，Orbit 只能接受的扰动形式为.这种形式能产生垂直磁面的分量，后者是产生磁岛（magnetic islands）的原因，同时也是无散度的,非常适合低beta MHD 扰动。Orbit利用2维样条插值法表示给定轴对称磁平衡的B，以及波纹场和MHD扰动，然后利用四阶Runge-Cutta方法计算导心方程。Orbit结构是由一个主程序orbit.F以及若干个子程序构成。其中，orbit.f中,有一个name list和供选择运行方式，具体诊断，数据存储和不同输出的开关。子程序包括：initial.f, deposit.f, collisions.f, perturb.f, step.f, record.f, and orbplot.f。输出数据由２部分文件存储，orbit.out和作图数据\*.plt。orbit.out输出粒子的信息，给定的平衡数据，和记录运行状态的数据，包括运行中断原因。Orbit可执行程序是利用Makefile脚本编译而成，适用各种常用f90编译器。重新编译规定删除所有的输出文件.out和.plt，因此，有用数据需要另行保存。

**Orbit计算流程：**

# 参数输入

1. **平衡数据**

Orbit程序所需的平衡文件以样条数据形式保存，文件名为spdata。Spdata由平衡程序该平衡eqs.f生成。Eqs.f程序支持2类平衡，即为解析平衡和数值平衡。其中，数值平衡是建立在由Alex Pletzer设计算法、Doug Mccune实现的Mex2eqs转化程序。Mex2eqs读入TRANSP或EFIT数据(gfile)，然后产生平衡文件map01.cdf，之后又被Eqs.f读入生成样条数据文件spdata。而解析平衡不需要生成map01.cdf文件。所有的平衡数据是不包括波纹场。如果要加入波纹场效应，设置eqs.f，自带TFTR, Tore Supra, ITER，NSTX，Ignitor装置的波纹场，其他装置实际波纹场的数据必须根据给定的表达式来进行拟合，修改eqsub.f中的波纹场。

* 设置eqs.f
  + 解析平衡：numeric=0
    - 环位形tok： 大环半径，反向环径比，q剖面
  + 数值平衡：numeric=1

1. **运行模式—orbit.F**

Orbit运行前需设置：

* 运行模式： nplot, nploteq
* 粒子数目：nprt
* 运行时间：ntor
* 扰动：npert

主程序Orbit.f提供了nplot=1-14，12种运行模式，产生12种不同的输出。设置运行模式时，除了需设置nplot的值，还需设置粒子、平衡磁场、扰动场，碰撞等参数。

* 单粒子轨迹计算 nplot=1

计算一个给定电荷、质量、磁面位置polo、能量E、投掷角pchi的粒子运动轨迹。计算结果需调用诊断程序rcrd1得到。时间步长为dt1，默认值为0.01\*tran，通行时间的1/100，在initial.f中设置。时间步长dt0=tran/N，对于低频加扰动，一般取N=200结果收敛。默认产生数据包括：时间, 能量变化, 时间步长, *,*  可以通过修改record.f中的子程序rcrd1来产生数据。每隔ndum步纪录数据，比如，ndum=mod(nstep,10)或ndum=mode(nstep,1)。调用shelldep(tdum)来设定单粒子初始轨道。其中，tdum为极向角，zet(1)为环向角位置。运行结束以后，这些数据保存至traj1.plt和traj2.plt中。作图软件默认为supermongo。相应的作图程序为：trajxz.p作轨道的极向投影图和trajt.p作上述各量随时间演化的图。

* 局部扩散和损失计算 nplot=2

计算初始均匀分布于某个磁面上单能粒子群扩散和损失。计算结果需调用诊断程序rcrd2得到。默认产生数据为：均方位移随时演化,和平均投掷角分布以及平均能量变化。时间间隔默认为一次运行产生200个点，设置记录数据的时间间隔的变量为nskip，在initial.f子程序set1之中。运行结束后，数据保存于diffusion.plt，损失粒子数据保存于lost.plt，初始粒子数据保存在dist.plt，最终粒子数据保存在distf.plt，损失、初始、最终粒子的分布可以用supermongo的作图程序：histogram.p。并且，nplot=2运行模式也提供单粒子轨迹，具体：运行完一次多粒子模式后，得到想要分析的粒子的序号，赋值给name0，调用subroutine runone，粒子轨道数据也保存于traj1.plt，traj2.plt，类似nplot=1模式。其它多离子模式下，也可以调用runone，得到某个序号粒子的轨道。注意，运行多粒子模式时，利用subroutine reduce, eject来获得剩余的约束离子，减少运行时间。此外，可以选sampledep、shelldep、btail等，

* 庞加莱截面计算 nplot=3

计算投射角pitch=1，低能量，均匀分布于p1和p2之间磁面的粒子群轨道，来模拟磁力线轨迹。计算结果调用诊断程序rcrd3。记录一个粒子每次经过截面的位置，通过子程序plot3保存至poincare.plt中。庞加莱截面作图程序：poin.p，得到扰动的在极向截面上结构。

* 扰动引起的分布变化计算 nplot=4

计算在扰动模，比如TAE，影响下的粒子分布变化。该分布变化按照初始分布的粒子经过一定时间持续运动后，然后，统计接下来每一时刻的粒子分布，得到时间平均后的粒子分布。持续时间由子程序rcrd4设定。数据保存于Distave.plt

* 局部自举电流计算 nplot=5

计算局部自举电流的。See chapter 8 and 文献 Wu and White (1993).电流值用园截面下小碰撞频率的理论值归一。作图程序为：boot.p

* 稳态扩散计算 nplot=6

计算一个帐篷型的粒子分布。边界处粒子一直被中心处的粒子替换，直至达到稳定。通过分布梯度和流速可以得到局部扩散，用tent.p作图。

* 粒子飞行分布函数计算 nplot=7

计算飞行距离。一个飞行定义为投掷角保持符号不变的一个运动周期。See publication with Spizzo in Phys Plasmas 2007, and PPCF in 2009.作图程序为flight.p

* 波的激发计算 nplot=8

计算波或本征模与离子的能量交换。数据保存于excite.plt。其中，P,E,mu分别代表初始的离子角动量，能量(keV)，磁矩；dE/E, pol分别代表时间平均后的能量变化和位置，时间为ntor。

* 动理学庞加莱截面计算 nplot=10

计算来动理学庞加莱截面确定时变波与粒子的共振。通过调用deposit.f中poinkdep产生初始均匀分布于附近一定区域的粒子。通过调用record.f中rcrd10来收集每次经过的粒子数据，保存于poink.plt。作图用：poinkin.p。

* 相矢量的转动计算 nplot=12

计算时变波与粒子的共振。调用deposit.f中wdep2产生磁矩为固定，均匀分布E,P平面的粒子。调用rcrdrot收集粒子数据，保存worm.out和slope.out。作图用：peplane.p。See papers on Modification of particle distributions by MHD

instabilities, R. B. White 2011, listed in attached file.

* “退火”计算 nplot=14

计算由于相矢量转动产生的连续”退火”对初始粒子分布的改变。调用choas.f中anneal2d进行“退火”计算。调用stoch2d读取数据worm.2.sv和slope.2.sv进行计算。这里.2表示E,P平面中的磁矩值。初始分布和最终分布保存于dist0.plt和distf.plt，作图用：histogram2.p。See papers on Modification of particle distributions by MHD instabilities, R. B. White 2011,listed in attached file.

**粒子参数设置表**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 参数名 | 符号 | | | | 初始设定值 | | | | | 备注 | | |
| 粒子数目 Nprt | | | 1000 | | | | |  | | | | |
| 粒子电荷 | Zprt | | | | 1.D0 | | | | | 以质子电荷为单位 | | |
| 粒子质量 | Prot | | | | 1.D0 | | | | | 以质子质量为单位 | | |
| 粒子能量 | Ekev | | | | 20.D0 | | | | | 以电子伏特为单位 | | |
| 随机数种子 | Iseed | | | | 0 | | | | | 用于分布和散射操作 | | |
| 模拟时长 | Ntor | | | | 6 | | | | | 以绕环轴次数为单位 | | |
| 磁场强度 | Bkg | | | | 3.557 | | | | | 单位kGs，可有起伏 | | |
| 扰动加载 | | Npert | | | 0/1(无/有扰动) | | | | | |  |
| 扰动势振幅 | | Pamp | | | 单位Kev | | | | npert=0时无效 | | | | |
| 分布类型 | | Ndist | | | 4 =alphas 1=shell distrib 2=sampledep,read fromTRANSP data，3=poincare | | | |  | | | | |
| 磁面（flux surface） | | Polo | | | | 0.5\*pw,pw为最后磁面 | | | Nplot=1,2时初始化 | | | | |
| 最小表面 | | P1 | | | | .6\*pw | | | Poincare模式 | | | | |
| 最大表面 | | P2 | | | | .95\*pw | | | Poincare模式 | | | | |
| 投掷角pitch | | Pchi | | | | .6D0 | | | 单粒子模式 | | | | |
| 能量守恒标度 | | Dele | | | | 5.D-8，用于校验能量守恒是否成立 | | | 每时间步长能量  起伏限制。 | | | | |
| 对同步时间，dele必须大于1以强制使用initial.f中的dt0为同一时间步长（200步每绕环运动） | | | | 对非时间依赖过程，能量不守恒，置dele>1,  Energy conservation should be controlled using zero frequency. | | | nplot=4时 dele为初值  若用瞬时分布测定，则 需dele >1来固定时间步长 | | | | | | |
|  |  | | | |  | | | | |  | | |

**碰撞参数设置表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 参量名 | | 参数 | 初始设定值 | | 备注 | |
| 碰撞类型 | | Ncol | 0=无碰 2=完整情形(profiles)见collisions.f中scatr, 函数denb,deni,tempi,tempe | | 1=能量依赖性 | |
| 倾角散射频率 | | Col | 1.d0/(50\*tran) | | 碰撞频率为穿行时间tran的1/50 | |
| 拖曳频率 | | Drag | 0.d0/(200\*tran) | | 减速拖曳为穿行时间tran的1/50 | |
| 背景等离子质量 | Massb | | | 2 | |  |
| 背景等离子电荷 | Chgb | | | 1 | |  |
| 杂质存在设置 | | Imp | 0=无杂质 | | 1=有杂质,类型1 | |
| 杂质质量 | Massi | | | 5 | |  |
| 杂质电荷 | Chgi | | | 5 | | ion energy |
| 粒子能量 | | Eion | 1.1 | | in kev for plots | |

1. **粒子载入**

几种不同分布的粒子通过设置变量ndist来开关程序deposit.f得到。其中，shelldep表示单能粒子群均匀分布于某个磁面polo上；poindep表示粒子群均匀分布于磁面p1和p2之间；sampledep表示从TRANSP中直接读入粒子分布，该文件为fbm.dat；alpha表示给alpha粒子的分布。任何在位置、能量以及投射角的粒子分布都能利用蒙托卡洛取样办法生成。

设置ndist=2，调用sampledep，读入TRANSP直接产生粒子分布，该文件名为fbm.dat，格式为简单的ASCII，脚本getfh4orbit.scr用于TRANSP生成粒子束数据(见附录)。

调用epfdep，读入高能量粒子来计算频率分布，默认初始没有给定ptch值。

1. **扰动载入**

Orbit中扰动形式为：

假定为绝热不变量。每一时间步，由于粒子处于不同位置，扰动场通过程序perturb.f中的子程序ptrb1计算一次代入到磁场中。注意，对于MHD扰动，平行电场为0。因此，需要引入一个静电势来抵消MHD扰动的作用，使得平行电场为0。子程序ptrb1有关于该静电扰动的开关。扰动也具有2种形式。解析的形式通过子程序amp1调用插值程序spln以及ampa调用插值程序splna来得到。数值形式使用readptrb子程序来读取NOVA等程序产生的扰动的数据ptrb.dat。程序中给出一个TAE扰动样例文件ptrb.dat，格式为简单的ASCII。扰动结果保存于文件harmonics.plt，查看可利用harmonics.p作图。

另外，Orbit提供两种扰动形式的载入，即通过或扰动位移来加入扰动。子程序ampa载入解析，子程序ampx载入解析；子程序readptrbx载入数值，子程序readptrba载入数值。其中，解析扰动结构自带三种，包括Alfven模式，MHD／电阻模式以及波纹场，极向模数和环向模数通过mmode和nmode给定，幅度通过amp给定，频率通过omegv给定，实际频率为omegv\*omeg0/(2.D3\*pi),单位为kHz。而数值扰动的结构由文件输入，具体格式见样例：Xin08w.1267E+01，包括表头中格点，模数的注释，一个x坐标一维数组，按极向模数升序排列的所有模式径向结构一维数组。幅度也通过amp给定。注意amp对应于输入文件中最大模式的幅度，以此作为定标，其他幅值根据模式结构和该幅度定出。频率也通过omegv给定。

1. **碰撞设置**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参量名 | | 参数 | 初始设定值 | | 备注 |
| 碰撞类型 | | Ncol | 0=无碰 | | 1=能量依赖性 |
| 2=完整情形(profiles)见collisions.f中scatr, 函数denb,deni,tempi,tempe | | | | | |
| 倾角散射频率 | | Col | 1.d0/(50\*tran) | | 碰撞频率为穿行时间tran的1/50 |
| 拖曳频率 | | Drag | 0.d0/(200\*tran) | | 减速拖曳为穿行时间tran的1/50 |
| 背景等离子质量 | Massb | | | 2 | |
| 背景等离子电荷 | Chgb | | | 1 | |
| 杂质存在设置 | | Imp | 0=无杂质 | | 1=有杂质,类型1 |
| 杂质质量 | Massi | | | 5 | |
| 杂质电荷 | Chgi | | | 5 | |
| 粒子能量 | | Eion | 1.1 | | ion energy in kev for plots |

# 

# 数据输出

**数据输出子程序(orbplot.f)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 子过程 | 记录内容 | | Orbplot.F对应plot输出 | |
| Plot1 | | 在dt1的时间间隔内，单粒子的数据 | | traj1.plt,traj2.plt |
| Plot2 | 平均磁面移出量的平方(mean square displacement from flux surface)，平均倾斜角(pitch)和能量变化 | | Diffusion.plt | |
| Plot3 | 当时庞加莱图数据 | | Poincare.plt | |
| Plot4 | 在程序开始后的特定时间（specfied time），展示分布的时间平均值 | | Distave.plt | |
| Plot5 | 自举电流-时间 | | Boot.plt | |
| Plot6  Pdist | 自行按需要对应  nplot=6设置  初始分布函数，  最终分布函数，  损失分布函数  包括粒子的pol  thet,zet,x,z,  en,ptch,P,P0,  mu\*B,polo,V,k | | Dist.plt，distf.plt, lost.plt  其中,得到ptch有两种方式，可以默认，也可以用计算公式，为ptch(k)= rho(k)\*b(k) /sqrt(2\*en (k))， 导致ptch>1.0，不要采用该式，直接用程序update后的ptch(k) | |

**supermonogo数据处理程序[[1]](#footnote-1)**

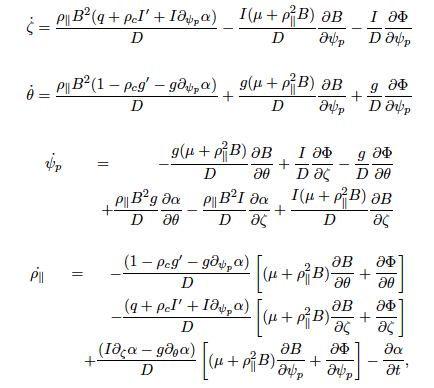
|  |  |  |  |
| --- | --- | --- | --- |
| 程序名 | 备注 | 程序名 | 备注 |
| Diff.p | 扩散 | Equilibrium.p | 平衡形状 |
| Dist.p | 初始、最终以及平均粒子在位形空间分布 | Profile.p | 各种平衡剖面 |
| Harmonics.p | 扰动的谐波 | Histogram.p | 柱型统计图 |
| Poin.p | 庞加莱截面 | Scatr.p | 高级碰撞算符剖面 |
| Lost.p | 粒子损失－时间 | Poinkin.p | 动理学庞加莱截面 |
| Trajxz.p | 极向截面上的粒子轨迹 | Muplane.p | 显示通行、捕获粒子P,区域的图形 |
| Trajtop.p | 环向的粒子轨迹 | Boot.p | 自举电流－时间 |
| Trajt.p | 粒子轨迹数据－时间 | Peplane.p | 显示通行、捕获粒子P,E区域的图形 |
| Distave.p | 平均分布 |  |  |
|  |  |  |  |

# 模型

1. **包含扰动的Hamiltonian量**

定义

则Hamiltonian共轭量：. 是磁面内环向电流。

1. **运动方程**
2. **物理量与变量**

* 求解运动方程－step.f

|  |  |  |  |
| --- | --- | --- | --- |
| 物理量 | 变量 | 物理量 | 变量 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| *k*th particle |  |  |  |

* 添加扰动－perturb.f

|  |  |  |  |
| --- | --- | --- | --- |
| 物理量 | 变量 | 物理量 | 变量 |
|  |  |  |  |
|  |  |  |  |

三种解析扰动形式为：

1. Alfven mode

|  |  |
| --- | --- |
|  |  |

1. MHD mode
2. Ripple

# 示例

1. **nplot=1**

|  |  |
| --- | --- |
| gfile(Efit) |  |
| 磁场 |  |
| 能量 |  |
| 初始位置 |  |
| 投掷角 | ，0.8226 |



# 附录－运行mex2eqs，产生map01.cdf

* **运行mex2eqs**
  + **路径：**/home/plsm/orbit/cc/LINUX/test

**注：mex2eqs安装于Ubuntu10.10虚拟机中，xplasma路径：**/home/plsm/orbit/dd/LINUX/test

* + **mex2eqs的运行示例：**







# 附录－中性束离子分布

对于中性束注入，Orbit需要读入TRANSP产生的中性束离子的数据，作为样本，从而生成中性束离子分布。过程如下：首先，设置TRANSP参数表单变量OUTTIM，即指定对时间片，运行TRANSP，产生额外输出runID.DATAn，这里n的取值可以是1～9，取决于表单中设置的时间片数量。每个DATAn文件对应与设置的第n个时间片。这些文件包含NUBEAM模块中性离子的分布等细节。然后，键入get\_fbm命令读入，经计算处理后，输出N个离子的文件fbm\_dist.dat，每个离子给定能量，投掷角，径向位置和极向位置。

* **TRANSP生成runID.DATAn和fbm\_dist.dat**

DMC 14 Aug 2000

updated 24 Oct 2008

updated 05 Jan 2009, M. Gorelenkova(MG)

\*\*\*\*\*\*\*MG

The distribution function calculation is independent of the NLFBMFLR switch,

and have the code accumulate two copies of the distribution function,

both the particle version (FBM\_PTCL) and the guiding center version (FBM) .

We will still use NLFBMFLR, but only to control which copy of the

distribution function is used to compute the beam-beam neutrons

in ../nubeam/bbneut.for and associated routines.

\*\*\*\*\*\*\*MG

The fortran program `get\_fbm' has been ported to unix and upgraded

to allow greater flexibility of access to fast ion distributions.

In particular, beam and fusion product distribution functions (now),

and RF tail ion distribution functions (eventually we hope) can be

accessed, at any radial or poloidal location.

Averaging and integration features allow reconstruction of such

quantities as the flux surface averaged total fast ion density

as a function of flux surface label, or, at a particular flux

surface, the poloidal variation of the fast ion density. Or the

density integration can be restricted to particles within a certain

energy or vpll/v range. Etc., etc.

The `get\_fbm' program can also write the distribution function data

out to a NetCDF file, which will be a more convenient format for

separately written software using this data, than the "encoded ascii"

legacy format used in the .DATAn files output directly by TRANSP.

\*\* a note on units & normalization \*\*

`get\_fbm' reads the TRANSP COMMON array fbm from an ACfile (see

Heidbrink's note, below). The units of FBM are

#/cm3/eV/[delta(solid-angle)/4pi]

but in TRANSP & NUBEAM, "delta(solid-angle)/4pi" is equivalent to

1/[number of vpll/v bins].

------------------------------------------------------------------------

(Discussion of delta(solid-angle) on a unit sphere in velocity space)

In terms of angles on a unit sphere in velocity space:

vpll/v = cos(alpha)

vperp/v = sin(alpha)

delta(vpll/v) = delta(cos(alpha))

For contribution to surface area on the sphare, there is a factor

2\*pi\*sin(alpha) = 2\*pi\*(vperp/v) since vperp/v is the radius of a

circle on the unit sphere in 3d v-space. But the differential area

on the surface of the sphere also has a factor of differential width

dw: dA=dw\*2\*pi\*sin(alpha). Corresponding to delta(cos(alpha)) this

width dw of the differential band around the circle at radius

2\*pi\*sin(alpha) has a factor of 1/(sin(alpha)) which cancels out the

factor due to the radius of the circle. You can convince yourself

of this with a picture of the sphere showing the relation of dw to

delta(cos(alpha))=delta(vpll/v).

You get integral[(vpll/v = -1 to 1){d(vpll/v)\*2pi}] = 4pi = surface

area of unit sphere. The contribution of each pitch zone delta(vpll/v)

is equal at delta(solid angle)/4pi = (1/nznbma) in terms of

TRANSP/NUBEAM's FBM data.

(End of discussion of delta(solid-angle)).

------------------------------------------------------------------------

Given the reconstructed TRANSP COMMON arrays (e.g. in get\_fbm.for), the

number of particles of fast specie 'js' in spatial zone 'jz', is given

by:

!

! nznbma = # of vpll/v bins

! nznbmea(js) = # of energy bins

! (in general, different fast species have different energy grids)

! bmvol(iz) = volume of iz'th spatial zone

!

znsum=0.0 ! sum to contain total number of particles this zone

!

do ia=1,nznbma

do ie=1,nznbmea(js)

dE=efbmb(ie+1,js)-efbmb(ie,js)

znsum=znsum+fbm(ie,ia,iz,js)\*dE\*bmvol(iz)/nznbma

enddo

enddo

Questions: send email to dmccune (dmccune@pppl.gov).

`get\_fbm' enhancements can be requested.

--------------------------------------------------------

Note by W. Heidbrink (1997)

Date: Tue, 12 Aug 1997 10:27:29 -0700 (PDT)

From: <HEIDBRINK@GAV.GAT.COM>

Subject: Beam distribution function in TRANSP

To: GREENFIEL@GAV.GAT.COM, HEIDBRINK@GAV.GAT.COM, LAZARUS@GAV.GAT.COM,

MURAKAMI@GAV.GAT.COM, TERPSTRA@GAV.GAT.COM, PETTY@GAV.GAT.COM,

FOREST@GAV.GAT.COM, STALLARD@GAV.GAT.COM, BaylorLR@ORNL.GOV

TRANSP FAST-ION DISTRIBUTION FUNCTION

W.W. Heidbrink

TRANSP calculates many quantities derived from the fast-ion distribution

function "f" but does not ordinarily save "f" itself. Using tools

originally written by McCune and at JET, Ted Terpstra and I have created

programs to view "f" at DIII-D.

-----------------------------------------------------------------------------

1) Add lines to namelist. You must instruct TRANSP to dump its memory

at certain selected times. This creates a so-called ACFILE in the

appropriate result area. Here are the lines I inserted in

71524A01TR.DAT:

!

! Dump common block:

!

!SELOUT='XXX' !list of TRANSP common symbols to output (default=all)

SELAVG='FBM BMVOL BDENS2 EBA2PL EBA2PP' !quantities to average

AVGTIM=0.030 !averaging time (to reduce MC noise)

MTHDAVG=2 !2 for averaging after each MC timestep

OUTTIM=1.875,2.150 !up to 5 times that AC files are written

------------------------------------------------------------------------

2) Run TRANSP. If all goes well, you'll end up with files like

$TRANSPROOT/result/D3D.91/71524A01.DATA1

$TRANSPROOT/result/D3D.91/71524A01.DATA2

with one file for each OUTTIM you selected.

-----------------------------------------------------------------------

3) Use your chosen method for accessing the fbm data; use get\_fbm to

convert the data to NetCDF format if needed.

----------------------------------------------------------------------------

4) Use your chosen method for accessing neutron emission rates 2d profiles

and convert the data to NetCDF format.

----------------------------------------------------------------------------

5) Physics note: the "fbm" in TRANSP actually has a complicated

spatial index label that moves around radially and poloidally. There

are codes that only bother to return "f" at the outer midplane.

This nearly always is sufficient to reconstruct the full distribution

function, because almost every orbit crosses the outer midplane.

But not quite. There are some "inner" orbits (close to the x-point

"pinch" orbit in phase space) that always have R<R\_0. These are

extremely rare for DIII-D beam ions, but could be important if we ever

pull out an enormous rf tail.

============================================================================

D. McCune: Caution: NLFBMFLR = .TRUE. is the namelist default!!

\*\*\*\*\* See comments by MG

In NUBEAM,

NLFBMFLR = .TRUE. means: accumulate fbm(rho,theta,E,vpll/v) at particle

positions;

NLFBMFLR = .FALSE. means: accumulate fbm(rho,theta,E,vpll/v) at the orbit

guiding centers.

\*\*\*\*\*

The default is .TRUE. because of the history of use of fbm for synthetic

diagnostics code, but it is not necessarily the best choice. In particular,

fbm has no gyro-phase dimension, and so, the accumulation of data at

particle locations that correspond to loss orbits for some gyro-phases (or

equivalently, some guiding center locations that could contribute) but

not for others, will be accumulated mostly from non-loss orbits, but, the

resultant anisotropy in the direction of contributing ions at (E,vpll/v) at

the spatial location cannot be reconstructed!

For NLFBMFLR=.FALSE., an FLR step needs to be taken to get to particle

positions e.g. for charge exchange eflux diagnostics simulation based on fbm.

Existing codes may not do this.

Of course, if the finite larmor radius is small, the switch will have little

effect. But if it is not, e.g. in STs or other low field experiments, it is

important to consider the proper setting of NLFBMFLR, which depends on the

use to be made of the "fbm" data.

============================================================================

D. McCune: Description of iregular 2d spatial grid over which the

"fbm" data is written...

The MCgrid, or "Monte Carlo grid", is an irregular 2d spatial grid that

was developed originally for capture of 2d binned data in a Monte Carlo

calculation (NUBEAM). It is used for fast ion distribution functions and

for certain spatially 2d profiles output by NUBEAM-- such as, beam halo

thermal neutral sources, beam-target and beam-beam fusion rates.

The grid is aligned with flux coordinates. It is constructed of "zone

rows" that are equally spaced in rho=sqrt(Psi\_tor/Psi\_tor\_at\_bdy); each

zone row is subdivided into a different number of zones equispaced in

the equilibrium poloidal angle coordinate theta, with fewer subdivisions

for zone rows near the axis, and more for zone rows in the edge. The

result is a set of zones all with roughly equal volume and cross sectional

area, as is desirable for consistency of Monte Carlo summation statistics.

The number of poloidal zones per zone row is linear in the zone row index:

typically, 4 in the 1st zone row adjacent to the magnetic axis, 8 in the

next row out, 12 in the next row, etc.

There are two variants, according as the underlying MHD equilibrium is

updown symmetric, or updown asymmetric.

As internally stored, the first row layout is this for the updown SYMMETRIC

variant:

\_\_\_\_\_

/ | \

/ 2 | 1 \

|\_\_\_|\_\_\_|

with theta=0 on the large major radius side; increasing to theta=pi on

the large major radius side covering the upper half of the plasma cross

section above the midplane. The next row out would have 4 zones, the

next 6 zones, etc.

And this for the updown ASYMMETRIC variant:

\_\_\_\_\_

/ | \

/ 4 | 3 \

|\_\_\_|\_\_\_|

| | |

\ 1 | 2 /

\\_\_|\_\_/

with theta=-pi on the lower branch on the small major radius side, theta=0

on the large major radius side, and theta=pi on the upper branch on the small

major radius side. The next zone row contains 8 zones, the next 12 zones,

and so on.

Poloidal zones are stored contiguously, with the poloidal zone index

increasing with increasing theta coordinate, theta being oriented

counter-clockwise in the plasma cross section drawn to the right of

the machine axis of symmetry.

--------------------------------------------------------

Spatial resolution of fbm output from TRANSP

There are two relevant TRANSP namelist controls:

NZONE\_NB (default =20) -- base radial resolution for NUBEAM

rule: divisible by 10

NZONE\_FB (default =10) -- radial resolution for fbm

rule: NZONE\_FB an even divisor of NZONE\_NB

Thus for NZONE\_NB=20, legal values of NZONE\_FB are 5, 10, and 20.

For NZONE\_NB=40, legal values of NZONE\_FB are 5, 10, and 20 and 40.

The number of poloidal zones per radial row depends on the radial row index:

NZONE\_FB = 10

row index updown symmetric updown asymmetric

1 2 4

2 4 8

3 6 12

4 8 16

5 10 20

6 12 24

7 14 28

8 16 32

9 18 36

10 20 40

----- -----

total 110 220

NZONE\_FB = 20

arithmetic progression extended through 20 zone row indices. The total

number of zones is NZONE\_FB\*(NZONE\_FB+1) for the updown symmetric case, and

twice that for the updown asymmetric case: 420 and 840, respectively, for

NZONE\_FB = 20.

Some variants of the MC grid include extra zone rows that extend beyond

the plasma boundary.

* **利用runID.DATAn，使用get\_fbm生成fbm\_dist.dat相关操作**

#! /bin/csh

# script to read the particle coordinates from TRANSP dumped files

# aand to write them into a file, which ORBIT can read

# the output file has information for each particle in the form of its

# position as columns: R Z Pitch Energy

# see comments in this file on the input options

# N.N.Gorelenkov, PPPL

#

# NTCC executable file name to read the distribution function

set EXE = get\_fbm

if(-e inp\_param) mv inp\_param inp\_param.old

# shot number

cat > inp\_param <<endcom

116313H22

endcom

# enter the directory, which must contain that run results (check it with ls first)

cat >> inp\_param <<endcom

/p/transparch/result/NSTX/05/ q

endcom

# enter the file number such as 1 for 66957B01.DATA1 or 2 for 66957B01.DATA2

# these are different files for different times of dumping the particles

cat >> inp\_param <<endcom

1

endcom

# specie number, 1 for beams (check for others with TRANSP people)

cat >> inp\_param <<endcom

1 n w n

endcom

# Give the number of particles. I suggest to give no more

# then you need, otherwise the ASCII file may be huge.

# But if you need all the particles and you give say 100000000

# only what it actually contains will be written.

# Also i do not take random particle position in rho, theta, v||/v, E

cat >> inp\_param <<endcom

100000 n y y y y

endcom

# enter the filename for the file

cat >> inp\_param <<endcom

fbm\_dist.dat q q

endcom

# enter the rest

cat >> inp\_param <<endcom

dat q q q

endcom

# run the program with input selected

${EXE} < inp\_param

# exit

exit 0

* 神马服务器中**利用runID.DATAn，使用get\_fbm生成fbm\_dist.dat相关操作**

**进入transp组：/scratch/transp/result/yulimin/backup/63887, 键入get\_fbm, 打开运行get\_fbm工具**

**GET\_FBM: ENTER TRANSP RUN ID (MAX 8 CHARS)**

**63887A01**

**unixpath: enter path string or option code:**

**Q**

**GET\_FBM: ENTER FILE ID (# BTW 0 AND 9):**

**1**

**choose species # or option code:**

**1**

**get\_fbm: enter option code:**

**w**

**get\_fbm: enter option code:**

**c**

**get\_fdep: give random number seed ?(Y/N)**

**Y**

**get\_fdep: enter seed (odd integer):**

**100001**

**modify energy range for sampling? (Y/N)**

**n**

**get\_fdep: vary rho(Y/N)?:**

**y**

**get\_fdep:vary theta(Y/N):**

**y**

**get\_fdep: vary vpll/v(Y/N):**

**y**

**get\_fdep: vary E(Y/N)**

**y**

**unixfile(fbm\_dist): enter file spec or “r”:**

**fbm\_dist.dat**

**get\_fbm:enter option code:**

**q**

**打开fbm\_dist.dat，头语句更改为5 100001, 并保存成为orbit所需的fbm\_dist.dat.**

1. **注：也可以用其他常用的作图软件画图输出，表中的作图程序容易修改成其他作图软件命令的程序。** [↑](#footnote-ref-1)