# Spin Tracking for RCS Lattice using Zgoubi

Vahid Ranjbar

# Outline

- Preparing Zgoubi input files
  - Design in MADX
  - Translation of MADX output to zgoubi.dat
- Running Zgoubi on Cluster
  - Python Script to manage Parallelism
  - Python Script to manage output

# Designing in MADX

- I run my Spin and Twiss parameter optimization python script to generate my final MADX sequence file.
- I have a special MADX input file to generate twiss output appropriate for the Zgoubi MADX translator:
  - Use the following format:

    ```
    select, flag=twiss, column = l, angle, k1l,k2l,k3l,k4l, tilt, e1,e2, h1,h2, s, alfx,betx, alfy,bety, x,y, Dx,Dpx, Dy,Dpy, mux,muy, hkick, vkick,ksi,name, keyword;
    ```

  - If you are exploring misalignments you can include all of that in the MADX script:

    ```
    SELECT, FLAG = ERROR, CLASS = quadrupole;
    EALIGN,  DY:= 0.0001*TGAUSS(2);
    SELECT, FLAG = ERROR, CLASS = sbend;
    EALIGN,  DPSI:= 0.0025*TGAUSS(2);
    ```

    Save errors using:  `ESAVE,  FILE=err.file;`

  - If you can also perform the orbit corrections in MADX and save the corrector settings:

    ```
    CORRECT, FLAG=line, PLANE = y, MODE = svd,cond=1, CLIST = corry.out;
    ```

# Translating into a Zgoubi.dat file

- Francois wrote a translator in fortran which I modified a bit to read in misalignments and corrector strengths.
- It takes in three files generated by MADX:
  - madzg.in → twiss output
  - err.file → misalignments
  - corry.out → corrector values
- It generates a file called trad.out which is a zgoubi readable file (like zgoubi.dat) it is set up to perform a Zgoubi TWISS calculation.
- To do real tracking you need to modify it adding by hand the RF cavity:

```
'CAVITE' accelerating cavity
2
3841.699163  360.0 circumf. H
0.33500E+07  2.8
```

- And modifying the number of passes:

```
'FAISCEAU'                                                        4216
'MARKER' #End                                                        4217
'SRPRNT’
'REBELOTE'
16000 0.2 99
'END’
```

- You Probably also want to modify the particle input to reflect the initial distribution you want.

# Tracking on NERSC Cluster: Files needed

- prun -- batch file
- mwrapper.py -- python wrapper for handling MPI and zgoubi. creates directories called Run# with # equal to the rank to a total equaling the total number of process requested. The final output zgoubi.fai will be in this directory.
- zgoubi.Distgen.dat -- sample zgoubi.dat file to be run on single node or just at the normal command line to generate a 10K particle distribution named dist.out
- zgoubi.dat -- sample zgoubi.dat file for reading in the dist.out 20 particles per node (in the line after 3 the 1 20 tell it to read from line 1 to 20 this is then iterated up by 20 for each rank). If you want fewer particles per rank you need to change this line. Also you should check to see if your version of zgoubi is handling the 'ranks' correctly since this is my version.
- firstpass.py, lastpass.py, read_fai.py -- some python scripts for reading output data from all directories for a zgoubi run might need to be modified a bit for your needs

# Prun:

```
#!/bin/bash -l
#SBATCH -p regular
#SBATCH -N 2
#SBATCH -t 07:30:00
#SBATCH -J zgoubi
#SBATCH -L SCRATCH
#SBATCH -C knl,quad,cache

module load python
module load mpi4py
srun -n 32 -c 16  python-mpi mwrapper.py ../zgoubi
```

# Mwrapper.py

```
#!/usr/bin/env python
from mpi4py import MPI
from subprocess import call
import sys
exctbl = sys.argv[1]
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
myDir = "Run"+str(rank)
cmd = "mkdir "+myDir+"; cp zgoubi.dat "+myDir+"/. ; cd "+myDir+" ; "+exctbl+" -rank
"+str(rank)+" > outfile"
sts = call(cmd,shell=True)
comm.Barrier()
```

# zgoubi.Distgen.dat: Used to generate initial distribution

```
 Generated using MADX -> Zgoubi translator

'MCOBJET'                                                    1

13.34255292E+02     reference momentum (MeV/c) = 39.9999674E+01  ,  G.gamma = 90.7753087E-02

3

10000

2 2 2 2 2 2

0. 0. 0. 0. 0. 1.

3.345e-6 5.0461 7.02625E-8 2

4.1758e-06 20.76 7.02625E-8 2

0    35.     0.5e-6  1

123456 234567 345678


'PARTICUL'                                                   2

5.10998928E-01 1.60217653e-19 1.15965213627E-03 0. 0.

'SPNTRK'                                                    3

 3

'FAISCEAU'                                                   4

'FAISTORE'                                                   5

 dist.out   none

 1

'END'
```

# zgoubi.dat

```
Generated using MADX -> Zgoubi translator
'OBJET'                                                                          1
13.34255292E+02        reference momentum (MeV/c) = 39.9999674E+01  , G.gamma = 90.7753087E-02
3.0
1 20 1
1 1  1
1. 1. 1. 1. 0. 1. 0. '*'
0. 0. 0. 0. 0. 0. 0.
0
../dist.out
```

# lastpass.py

```
for k in range(0,NT-1):
 Fname = 'Run'+str(k)+'/zgoubi.fai'
 i=0

 for line in open(Fname):
   i = i +1
   if i > 4:
     listWords = line.split()
     gamma = float(listWords[24])/(pmass*1000)
     if int(listWords[37]) == 16001:
      # runNo        IPASS,   Ggamma, Y, T, Z, P, RET,DPR,SX,SY,SZ
      print k, listWords[37],G*gamma,listWords[9],listWords[10],listWords[11],listWords[12],listWords[33],listWords[34],listWords[19],listWords[20],listWords[21]
```