DRAFT

PRINCETON UNIVERSITY, PPPL, Mechanical Engineering Division

TO: M. Zarnstorff DATE: 29 February 2000

FROM: I. Zatz SUBJECT: ORBIT Code Application for Stellerator Optimizer

---

A stripped down collisionless version of the ORBIT code developed by R. White has been modified for stellerator optimizer applications. This memo will serve as a 'user's guide' for the code, whose results have been independently computed and benchmarked by M. Redi. A comparison of these results is also contained herein.

Introduction

This version of the ORBIT code, developed specifically for inclusion with the stellerator optimizer, contains several flexibility modifications not available with the baseline version of ORBIT. The code, as presently configured, can launch all particles at a specified surface, or launch them all randomly. Particle orbits are tracked until either a specified fraction of the particles have been lost or until a maximum specified number of toroidal transits have occurred. Additionally, an upper and lower annular boundary can be specified which defines particle loss. By default, each particle is launched with a random pitch, but if desired, a fixed pitch for the particles can be specified.

Compiling the Code

The following files are required for ORBIT compilation:
bzio.f
eq3d
eq3d.f
o.cln
orbit3d.f
sub3d.f

The script file 'orbf' is used to perform the complete fortran 90 compilation. The code should compile and run on both killeen at nersc and the hecate origin2000 at princeton.

Executing the Code

The script file 'orbx' is used to execute the ORBIT code. The execute line is of the form: ./orbx *ft* – where *ft* is the stellerator configuration indicated in the VMEC output file Bmns.*ft* (which needs to be present for ORBIT execution along with an input file called 'inorbit'). The *ft* string will appear on all files generated by ORBIT during its execution. While several output files are generated, two in particular are of primary importance - 'orbout.*ft*' contains the complete text of the ORBIT output while 'orbsum.*ft*' is a brief summary file containing parameter values required by the next step in the optimizer sequence. Those values include the number of particles at the start of the analysis, the number lost at the end of the analysis, the maximum number of transits requested, the number of transits completed at the end of the analysis, and an ORBIT termination code indicating the reason ORBIT stopped execution. The 'orbout.*ft*' file contains a particle-by-particle table indicating when a particle is lost. Particle pitch, and a breakdown of whether particles are lost to the inner or outer annulus are indicated as well as energy and time parameters for the full analysis.

DRAFT

The input file 'inorbit' provides ten parameters to the ORBIT analysis on two lines of input as follows:

First input card:

ntor	- maximum number of toroidal transits to be run
psiz	- normalized flux value (0-1) to launch particles at
	- (-1) a single random value at which to launch all particles
	- (2) a different random value is chosen for each particle (at present the only distribution available is uniform between zero and one)
ptchz	- particle pitch specified (-1 to 1)
	- otherwise each particle has random pitch
smax	- normalized (0-1) upper annular bound for particle loss
smin	- normalized (0-1) lower annular bound for particle loss

Second input card:

nprt	- number of particles
zprt	- particle charge
prot	- particle mass in proton units
ekev	- particle energy in kev
pcnt	- stop run when percentage of particles lost exceeds integer value

At present, a fixed random number seed is built into the code. In order to change this seed, change the value of variable jdmax located in subroutine dep in file sub3d.f (currently set to the default value of 100). Re-compilation of the code would then be required.

Benchmarking Comparisons

Three collisionless stellerator configurations were evaluated with the ORBIT code – c10, c82 and i63. All were run with 1024 alpha particles for 4000 toroidal transits at an energy level of 8.06 kev. The 'inorbit' file corresponding to these runs is -
4000, 0.25, 2., 1., 0.
1024, 2., 4., 8.06, 99

Two different random seeds were used for the c82 runs to verify the consistency of the results. The following table summarizes the collisionless particle and energy losses as well as the error bands:

	c10	c82	i63
Particle loss (%)	24.5 ± 1.5 %	25.1 ± 1.6 % 25.7 ± 1.6 %	29.2 ± 1.7 %
Energy loss (%)	24.5	25.1 25.7	29.2

M. Redi performed an independent analysis of these configurations. These calculations used a model alpha-particle source (parabolic**9) and has the option of including pitch angle (PA) scattering. The above baseline configuration runs were validated against the ORBIT simulations of birth energy alpha loss from reactor-size scaled stellerator configurations run by Redi. The two sets of simulations are comparable as 8 keV alphas in the stellerator have the same ratio Larmor radius to device size as do 3.5 MeV alphas in a reactor scaled stellerator. Once scaled for these differences in initial conditions, the

particle loss results compare favorably as shown in the table below. Not only do the summaries show good correlation, but the losses as a function of time do as well. Additional analyses for these configurations, which included pitch angle scattering, were also performed to indicate the contribution of collision effects.

	c10	c82	i63
Particle loss (%)	23.7 ± 1.6 %	23.3	25.1
Energy loss (%)	23.4	23.0	24.9
E loss w/ PA Scattering (%)	36	36	36