10/10/2000

PARALLELIZED VERSION OF ORBIT

CONTENT:

1. INTRODUCTION

As of October 2000, the Orbit code can be compiled and run on a parallel computer that provides a working MPI (Message Passing Interface) library. All the platform-specific options and informations on how the code is compiled are found in the file "Makefile". You may have to change the location of the libraries to reflect your system.

2. HOW TO BUILD (COMPILE+LINK) ORBIT AND EQS

Single processor Orbit, NO MPI (default)

```
     % gmake orbit    or just   % gmake
```

Single processor debugging version of Orbit, NO MPI

```
     % gmake DEBUG=y orbit
```

Multiprocessor version of Orbit

```
     % gmake MPI=y orbit
```

Multiprocessor debugging version of Orbit

```
     % gmake MPI=y DEBUG=y orbit
```

Building Eqs, single processor only:

```
     % gmake eqs
```

Debugging version of eqs

```
     % gmake DEBUG=y eqs
```

3. HOW TO RUN MEX2EQS/EQS/ORBIT

MEX2EQS allows you to load a toroidal MHD equilibrium from a variety of sources including from TRANSP data stored on the MDSPlus tree. Because MEX2EQS uses a shared library of MDSPlus calls, make sure that your LD_LIBRARY_PATH contains /usr/local/mdsplus/lib. Put the following in your .cshrc or .bashrc file:

```
% setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/usr/local/mdsplus/lib
```

under csh, or equivalently under bash

```
% export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/mdsplus/lib
```

To run MEX2EQS type

```
      % mex2eqs
```

and answer the various questions regarding the input equilibrium format, the MDSPlus tree/node/server and time slice information if applicable. Mex2eqs also supports the EFIT-EQDSK (or geqdsk) file format (select option 3). A geqdsk file, gbeta8.8, is provided in the distribution. MEX2EQS will create a file map01.cdf that mimics the old file produced by MAPMC, except that it is much lighter. You can plot its content from within MATLAB (type matlab) by loading the script map01.m at the MATLAB prompt:

```
      >> map01
```

This will display the mesh packing (which should be linear), various profiles and the Jacobian projected on the grid (the colours should align on the grid for Boozer coordinates). To plot |B| as a function on the poloidal plane, you could type

```
      >> figure, pcolor(x,z,b)
      >> shading('interp'), axis('image'), colorbar('vert')
      >> title('|B|')
```

Next you probably need to produce the cubic spline coefficients of the equilibrium (file spdata), by typing

```
% eqs
```

This will produce the file spdata needed by ORBIT. To view the result type

```
      % idt emeta &
```

or to send the result to the printer

```
      % ctrans -d ps.mono emeta | lpr -h -Pb143-dup
```

If compiled for single processor runs, which is the default when you do simply gmake, Orbit is launched the same way as before:

```
     % orbit
```

If compiled for multiprocessor runs, the easiest way to launch the code is to use the provided script "runorbit":

```
     % runorbit   or  runorbit 1    --> single processor run
     % runorbit num_proc   --> multiprocessor run if num_proc > 1
```

This script will take care of running the Perl script "orbout.pl", if needed, to merge all the results into the one familiar file "orbout".

If the script does not recognize your system, you will have to explicitly call the "mpirun" command:

```
     % mpirun -np num_proc [-machinefile file] orbit
```

where "num_proc" is the number of processors that you want to use for the run. The "-machinefile" option may be useful on a PC cluster of type "beowulf" to pick the hosts on which you want the code to run (see your local mpirun man page for more details).

By running "mpirun" explicitly, you will also have to run the Perl script "orbout.pl" if you want to merge all the results into one "orbout" file. See the beginning of the file "orbout.pl" for details about the script.

```
     % orbout.pl num_proc
```

4. THINGS TO KNOW TO CHANGE ORBIT

- DO NOT use write(6,...) or write(*,...) anymore. You need to use:
  ```
           write(myfile,...) ...
  ```

- In subroutine "bootrec" (record.f), nprt0 has been changed to nprt in loop 40 to handle the multiprocess calculation. A few source lines after, nprt is changed to nprt0 in the calculation of zv(kplt).

- Added variable "savenprt" in subroutine "pdist" (orbplot.f) to save the value of nprt. Also write out value of time(nob) instead of time(1) after the call to field.

- time(nob) is again used instead of time(1) in subroutine "plost", "wrt", and "wrt" (all in orbplot.f).

- replaced g(k) by gfun(px) in subroutine momf() (orbplot.f) in loop 483.

- orbit.f is now orbit.F since it includes some preprocessor statements.
  The same is true for ranff.f which is now ranff.F.

5. MORE INFORMATION

===========

All the codes for Orbit and Orbit3d in directories
pub/white/Orbit and pub/white/Orbit3d which are public.

```
        to get file, type
        ftp ftp.pppl.gov
        user: anonymous
        pass:   email address
        cd /pub/white/dir
        ls    - to see all files in dir
        get file
        get README.orbit
```

Files for the tokamak version orbit.f from Orbit

```
Makefile         eqs.f            math.f           record.f
bzio.f           eqsub.f          o.cln            step.f
bzio_dummy.f     ezcdf.f90        orbcom           stochastic.f
collisions.f     fshell.c         orbit.F          torsup.f
deposit.f        functions.f      orbplot.f        tv80_wrappers.f
diagnostic.f     icrfrot.f        perturb.f
dispersion.f     initial.f        ranff.f
dskinCDF.f90
```

Files for the Stellarator (or Tokamak) version orbit3d.f from Orbit3d

```
Makefile     eq3d         eqspline     o.cln        ranff_f90.f
allplot.f    eq3d.f       m0tok        orbit3d.f    sub3d.f
```

```
bzio.f       eqstell      m1tok       ranff.f
```

For a Stellarator or Tokamak

2.1 You can produce analytic second order Shafranov equilibria with eqs.f by selecting numeric = 0 at the beginning. See subroutine tok0 for analytic equilibria input.

2.2 Set the major radius (magnetic axis!) in centimeters at the beginning of eqs.f and the ripple choice krip. I have ripple models for ITER, Tore Supra, TFTR, and NSTX in eqs.f. Sorry, you will have to make your own for other machines. If no ripple is desired krip=0 skips it. The stellarator version eq3d.f does not have ripple, but the field input is in terms of harmonics, so it can be added. The magnetic axis location is essential since the code units are defined by the major axis and the gyro radius. The gyro radius is calculated later in orbit from the value of z, energy, B, and proton mass.

2.3 Type "gmake" to build both eqs (or eq3d) and orbit (or orbit3d), or "gmake eqs" (or "gmake eq3d") to build only eqs (eq3d). Now run eqs by simply typing "eqs" to produce the spline data set spdata. For Orbit3d, simply type "eq3d" to produce the file eqspline. It also produces an output file and, in the case of an analytic Shafranov equilibrium, a plot file gmeta, giving equilibria characteristics. Copy and store spdata or eqspline as eqs1001, or whatever, because otherwise it will be written over the next time you run produce an equilibrium spline file.

3. Guiding center analysis

3.1 Orbit.f takes as input spdata. Orbit3d takes eqspline. In the main of orbit.f there are many options, for single particle run, the addition of perturbations, different particle distributions, collisions, drag, subsidary options such as stochastic loss calculations, ripple contours, the trapped passing boundary, etc.

The guiding center equations used are given in
White, Phys Fluids 2, 845 1990. Begin by plotting the equilibrium and
a sample single particle orbit. Set nploteq = 1 (equilibrium plot)
and nplot = 1 (single particle orbit).

3.2 Examine the equilibrium plot in gmeta. If the outside surface
of the equilibrium is not well represented,
the spline dimension is too small. At the beginning of eqs.f,
the parameters lsp and lst are respectively the poloidal flux and
poloidal angle spline dimensions.
Increase lst. Limits on these dimensions are governed by the common
blocks spline, used by both orbit.f and eqs.f. Likewise, if
the magnetic axis location or plasma position are poor, increase
lsp. The accuracy of the representation depends on the plasma shape.

3.3. Output from orbit is a plot file,
gmeta, a data file orbout, and other data files for lost particle
distributions, etc., which can be constructed and set as desired,
usually in the routine pdist (particle distribution) or plost
(lost particle distribution).

3.4 There are all kinds of working switches in the code, for example
in the alpha particle distribution routine alphdep, ntrap = 1 can be
used to produce only a trapped distribution. These switches have
not been moved into an input file because I modify them all the time,
and produce new ones constantly. I try to put write statements
with them, so always check orbout to be sure you are doing what you
want to do, i.e. that the switches are set the way you want. A good
expedient is to always do a short run with few particles to check
what you are doing before starting a long run.

```
cc   r. b. white   princeton, jan 1982
cc   files needed:common blocks spline, equilibrium spdata written by spline.f
cc   shell sho will run and make plots
cc   nplot =1 gives single particle orbit data,
cc   data recorded at intervals dt1
cc   nplot=2 banana tip precession plot
cc   nplot =3 gives poincare  map
cc   nplot =4 gives precession and loss plot
cc   nplot=5 gives particle loss data, file data, write statement in reduce
cc   nplot=6 ripple loss calculation
ccc   nplot=7  boosted collisional ripple loss
ccc  , more analysis with lost.f after changing data to losdata
ccc  loss condition adjustable, see loss condition
cc   col = the collision frequency
cc   ekev is particle energy in kev,
cc   bkg is b at the magnetic axis in kgauss
ccc  field is b(pol,thet) + rpl(pol,thet)sin(N*zeta), rpl not usual ripple!
cc   zprt is the charge, and prot the mass in proton units
cc   rmaj is the major axis in cm- given in equilibrium file
cc   dele is the allowed fractional energy change per step
cc   the time step dt is adjusted accordingly,
cc   to run at dt0 set dele > 1.
cc   nprt is the number of particles.
cc   trun is total run time. need ten steps per transit time, tran
cc    nploteq=1 plots some equilibrium functions, 2 more, 0 none
ccc
ccc--computing time(Cray A): alpha particles in 272cm 50kg TFTR equilibrium
ccc --A run with 1000 particles 100 transits
ccc--with-ripple takes 38 sec
ccc--time-is 53% field, 31% onestep
```

2.5 Single particle orbits (nplot = 1)

nplot = 1 runs single particle orbits, giving plots of time history.
The first plot shows the accuracy of energy conservation.
The time step is controlled by the energy conservation, using the
limiter dele, normally set to about 5.e-8.
If time dependent MHD modes are used (set npert = 1, and set the
amplitudes and frequencies in subroutine amp1) energy is not conserved,
and a fixed time step must be used. dele > 1 accomplishes this, the
time step is dt0, which can be adjusted.

A typical output file (orbout) produced by nplot=1:

```
#################
 orbits.f, read eqdata -lsp,lst,lq,le,lr
  31  61   4   8
 plasma volume,bax   1.5554E+03  9.9881E-01
  Last flux surface is wall
  equilibrium plotted, sub plotf
     1  deposit   pol,thet,ptch   3.89E-01   0.00E+00   3.50E-01
 ended  1.00E+03 steps    0 lost     1 at time=trun
   code orbit.f  nplot=  1
equilibrium  m0iter1
 pq1,p1,polo,p2  1.37E-03  1.30E-03  3.89E-01  1.30E+00
 rq1,rw,eps,xc  3.92E-02  1.47E+00  1.55E-01  9.48E+00
 q0,qed,qw,bax  4.70E+00  5.52E+00  5.52E+00  8.82E-01
 ped,pw,pvol     1.30E+00   1.30E+00   1.56E+03
 shafranov shift/xc     8.32E-02
 ITER ripple  d0,n=  3.75E-06  20
 xrip,wrip,brip  6.75E+02  5.35E+01  2.68E-01
particles
 uniform dist   p1,p2  1.296E-03  1.296E+00
 engn,nprt,ftrap   6.66E-04       1   0.00E+00
 col,drag  0.00E+00  0.00E+00
times
 dt(nob),tran,dt0 4.8E+01 1.6E+03 3.3E+01
 time,ntor,dele  3.27E+04      20  5.00E-08
 dt1,nstep  1.63E-03      1003
             cgs units
equilibrium  m0iter1
 energy=1.50E+03 kev,   mass=4.00E+00 proton
 charge z= 2.00E+00
 b on axis =  4.86E+01 KG
 lft,cent,axis,rt 6.44E+02 8.70E+02 9.49E+02 1.10E+03 cm
 gyro=3.66E+00 cm, gyro/raxis 3.85E-03
 gyro=2.33E+08 rad per sec,   tran=7.00E-06 sec
 velocity=8.51E+08 cm/sec,    beta axis=1.26E-01
#################
```

Most output lines are self explanatory.
The first line is the reading of the spline data, with spline dimensions given. The plasma volume, bax are the volume in the units of the numerical equilibrium and the magnetic field on axis in the same units.
"Last flux surface is wall" refers to the criterion for particle loss, it can be changed, see subroutine wallset.
"1 deposit   pol,thet,ptch ..." gives initial particle data
"ended 1.00E+03 steps   0 lost   1 at time=trun" end of time step
pq1,p1,polo,p2 - poloidal flux values
rq1,rw,eps,xc - minor radius at q=1, last flux surface, aspect ratio, axis
q0,qed,qw,bax - q on axis, plasma edge, last flux surface, B on axis
ped,pw,pvol - poloidal flux at plasma edge, last flux surface, volume
engn,nprt,ftrap - normalized particle energy, number, fraction trapped
col,drag - collisions and drag
dt(nob),tran,dt0 - time step, transit time, initial time step
time,ntor,dele - final time, number of toroidal transits, energy conservation
dt1,nstep - recording interval, number of steps

2.6 Particle loss analysis (nplot = 5)

nplot = 5 is for doing particle loss analysis. There are several subroutines for loading different types of Monte-Carlo distributions. The subroutine pdist will give plots of particle distributions. It can be called at any time during a run, but normally only at the beginning and end. Inside the subroutine eject, near the beginning, there is a write statement which writes out individual particle loss data, including the particle number. To observe a particular loss orbit leave the initial nplot =5 load parameters the way they were, and activate runone using this particle number, using the call in the main, just before the time step loop. The routine runone loads the particle you have selected into position 1, and then switches to nplot =1, giving a plot of that particular orbit.
Inside plot5 there are several optional calls

```
  call sigma  - gives a statistical error analysis of particle loss
  call pdist - plots the particle distribution functions
  call plost - plots the lost particle distribution functions
```

```
call mupzeta(1) initial particle positions, pzeta, mu plane
call mupzeta(0) final particle positions, pzeta, mu plane
call dump0 - writes out a file for lost particle analysis, to be
used with lost.f
```

A typical output file (orbout) produced by nplot = 5:

```
 orbits.f, read eqdata -lsp,lst,lq,le,lr
  31  61   4   8
 plasma volume,bax   1.5554E+03  8.8181E-01
  Last flux surface is wall
 alphdep, pitch < 1     100 times
 alphas deposited (1-(r/a)**2)**p, p=  3.0000E+00
   500 nprt -loss   225  ptch,thet,x,t  3.48E-01 -2.18E-01  1.00E+03  4.96E+00
   499 nprt -loss   112  ptch,thet,x,t -1.17E-02 -6.39E-02  1.08E+03  2.00E+01
   497 nprt -loss   412  ptch,thet,x,t  5.00E-01 -3.95E-01  9.14E+02  2.54E+01
   482 nprt -loss   119  ptch,thet,x,t  4.81E-01 -4.22E-02  1.09E+03  1.77E+01
   465 nprt -loss   251  ptch,thet,x,t  5.65E-02 -5.25E-02  1.09E+03  4.64E+01
 ended  1.13E+04 steps    5 lost   495 at time=trun
 sigma-  mean, deviation   1.0000E-02  4.4274E-03
 pdist called time(1) =   5.3392E+04
  <pol/pw>, <(pol/pw)**2>   1.6372E-01  5.5420E-02
    500 part.      117 trap       3 trap-lost asym=  1.02E-01
 plost called time(1) =   5.3392E+04
 time of last loss   4.9598E+04
 plost, |thet|<pi confined,lost      111       5
 plost, |thet|>pi confined,lost      384       0
 mupzeta called time(1) =   5.3392E+04 init=    1
  mupplane called points   495
  mupplane called points     5
   code orbits.f  nplot=  5
equilibrium  m0iter1
 pq1,p1,polo,p2  1.37E-03  1.30E-03  3.89E-01  1.30E+00
 rq1,rw,eps,xc  3.92E-02  1.47E+00  1.55E-01  9.48E+00
 q0,qed,qw,bax  4.70E+00  5.52E+00  5.52E+00  8.82E-01
 ped,pw,pvol     1.30E+00   1.30E+00   1.56E+03
 shafranov shift/xc     8.32E-02
 ITER ripple  d0,n=  3.75E-06  20
 xrip,wrip,brip  6.75E+02  5.35E+01  2.68E-01
particles
 alpha dist
 engn,nprt,ftrap   1.55E-03     500   2.34E-01
 col,drag  0.00E+00  0.00E+00
times
 dt(nob),tran,dt0 5.3E+00 1.1E+03 2.1E+01
 time,ntor,dele  5.34E+04      50  5.00E-08
 dt1,nstep  1.07E-03     11306
             cgs units
equilibrium  m0iter1
 energy=3.50E+03 kev,   mass=4.00E+00 proton
 charge z= 2.00E+00
 b on axis =  4.86E+01 KG
 lft,cent,axis,rt 6.44E+02 8.70E+02 9.49E+02 1.10E+03 cm
 gyro=5.58E+00 cm, gyro/raxis 5.88E-03
 gyro=2.33E+08 rad per sec,   tran=4.58E-06 sec
 velocity=1.30E+09 cm/sec,    beta axis=1.26E-01


 alphdep, pitch < 1     100 times
 alphas deposited (1-(r/a)**2)**p, p=  3.0000E+00
```

This is the particle deposition record, see alphdep and other
Monte-Carlo deposition routines.

```
   500 nprt -loss   225  ptch,thet,x,t  3.48E-01 -2.18E-01  1.00E+03  4.96E+00
   499 nprt -loss   112  ptch,thet,x,t -1.17E-02 -6.39E-02  1.08E+03  2.00E+01
   497 nprt -loss   412  ptch,thet,x,t  5.00E-01 -3.95E-01  9.14E+02  2.54E+01
   482 nprt -loss   119  ptch,thet,x,t  4.81E-01 -4.22E-02  1.09E+03  1.77E+01
   465 nprt -loss   251  ptch,thet,x,t  5.65E-02 -5.25E-02  1.09E+03  4.64E+01
 ended  1.13E+04 steps    5 lost   495 at time=trun
```

This is the particle loss record, giving the total particles, the particle
lost, and its pitch, theta, major radius position and time in transit times.
To observe a particular loss orbit, activated runone (see call in main)

using the particle number.

```
sigma-  mean, deviation   1.0000E-02  4.4274E-03
```

This is the fraction lost and statictical error, see routine sigma

```
pdist called time(1) =   5.3392E+04
  <pol/pw>, <(pol/pw)**2>   1.6372E-01  5.5420E-02
     500 part.      117 trap       3 trap-lost asym=  1.02E-01
```

This is a call to pdist, the plot of the particle distribution, with loss estimate and distribution asymmetry parameter.

```
plost called time(1) =   5.3392E+04
 time of last loss   4.9598E+04
 plost, |thet|<pi confined,lost      111       5
 plost, |thet|>pi confined,lost      384       0
```

Lost particle plot and analysis

```
 mupzeta called time(1) =   5.3392E+04 init=    1
  mupplane called points   495
  mupplane called points     5
```

This is the call to the plot of the space of canonical toroidal momentum and magnetic moment, showing domains and the distribution of confined particles (495) and lost particles (5). init=1 shows the initial distribution, ionit=0 shows the final distribution. See subroutine plot5 for calls.
See White et. al. Physics of Plasmas 3, 3043, 1996, for a description of the plots.