Oculus: The Eye into Chaos

Dr. S.R. Hudson*
Princeton Plasma Physics Laboratory, PO Box 451, Princeton NJ 08543, USA

Prof. Y. Suzuki†
National Institute for Natural Sciences, National Institute for Fusion Sciences, 322-6 Oroshi, Toki, 509-5292, Japan
(Dated: March 21, 2018)

`Oculus` is a continually-under-development suite of diagnostic subroutines for non-integrable, toroidal magnetic fields used in the numerical simulation of magnetic confinement of fusion-research plasmas. `Oculus` is freely distributed, with the expectation that users will promptly inform the developer(s) of any errors.

Suggestions, requests *and contributions* are welcome, indeed encouraged! Subroutines, expanded documentation etc. will be developed on demand.

Quick Links

subroutine	task	available	documented
ga00aa	Locate the magnetic axis using fieldline following methods	✓	✓
tr00aa	Estimate the rotational-transform using fieldline following methods	✓	✓
bn00aa	Compute Fourier harmonics of $\mathbf{B}\cdot\mathbf{e}_\theta\times\mathbf{e}_\zeta$ on given surface	✓	✓

Contents

I. USER INPUTS AND COMPILATION INSTRUCTIONS

A. user supplied magnetic field

For the routines that require information regarding the magnetic field, the user must provide a subroutine, `bfield( RpZ, itangent, BRpZ, ifail )`, which returns the magnetic field, $\mathbf{B}$, in cylindrical coordinates, (R,ϕ,Z).

1. `RpZ(1:3)` is `real*8`; input;
   i. contains the R, ϕ and Z coordinates at which the field, and possibly the derivatives, are required.
2. `itangent` is integer; input;
   i. if `itangent=0` then only $\mathbf{B}$ is required;
   ii. if `itangent=1` then both $\mathbf{B}$ and its derivatives are required.
3. `BRpZ(1:3,0:3)` is `real*8`; output;
   i. The contravariant components of the magnetic field, namely $B^R\equiv\mathbf{B}\cdot\nabla R$, $B^\phi\equiv\mathbf{B}\cdot\nabla\phi$, and $B^Z\equiv\mathbf{B}\cdot\nabla Z$.
   ii. The required format is
   `BRpZ(1,0)` $=B^R$, `BRpZ(1,1)` $=\partial_R B^R$, `BRpZ(1,2)` $=\partial_\phi B^R$, `BRpZ(1,3)` $=\partial_Z B^R$,
   `BRpZ(2,0)` $=B^\phi$, `BRpZ(2,1)` $=\partial_R B^\phi$, `BRpZ(2,2)` $=\partial_\phi B^\phi$, `BRpZ(2,3)` $=\partial_Z B^\phi$,
   `BRpZ(3,0)` $=B^Z$, `BRpZ(3,1)` $=\partial_R B^Z$, `BRpZ(3,2)` $=\partial_\phi B^Z$, `BRpZ(3,3)` $=\partial_Z B^Z$.
   !!! Note that $B^\phi=\mathbf{B}\cdot\hat{\phi}/R$, and $\partial_R B^\phi=(\partial_R\mathbf{B}\cdot\hat{\phi}-B^\phi)/R$!!!
4. `ifail` is integer; output;
   i. returns an error flag;
   ii. `ifail=0` indicates that the calculation of $\mathbf{B}$ was successful.

For many of the following subroutines, the periodicity of the field will be exploited, by which it is meant that the magnetic field must satisfy

$$\mathbf{B}(R,\phi+\Delta\phi,Z)=\mathbf{B}(R,\phi,Z), \tag{1}$$

where $\Delta\phi\equiv 2\pi/$`Nfp`, and `Nfp` is an integer that must be provided as required.

*Electronic address: shudson@pppl.gov
†Electronic address: suzuki.yasuhiro@lhd.nifs.ac.jp

B. macro expansion and compilation

1. `Oculus` is available at `http://w3.pppl.gov/~shudson/Oculus/oculus.xxxxxxxx.tar`, where `xxxxxxxx` indicates the date=verson.
2. The `oculus.h` file is converted to `oculus.F90` via `m4 -P oculus.macros oculus.h > oculus.F90`.
3. **On compilation, it is required to convert single precision to double precision.**
4. Presently, the NAG library is required. (Replacement routines are presently being implemented.)
5. At some time in the future, the routines will be kept under version control (perhaps under `github`).
6. Please inform `shudson@pppl.gov` of any errors; and suggestions and requests are very welcome!

C. error flag

Each subroutine has an input integer `ifail`.

1. On input: `ifail` controls the degree of screen output;
2. for `ifail.ge.0`, operation is "quiet";
3. for `ifail.eq.0`, screen output is "terse";
4. for increasingly negative `ifail` the screen output is increasingly "noisy", which may be useful for debugging, for maximum screen output set `ifail=-9`.
5. for `ifail.eq.9`, internally allocated memory is deallocated and no action is taken;
6. On output, `ifail=0` for normal execution.

oculus.h : last modified on 2018-03-21