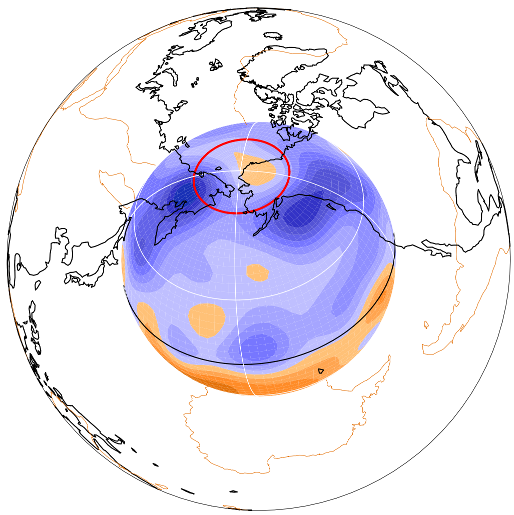
LEOPACK



sbrlinonsd

Solid **B**ody **R**otation **LIN**ear **ONS**et of Thermal Convection **D**rifting Frame Solve

Steven J. Gibbons, Oslo

Original document: November 21$^{\text{st}}$, 2001. Updated: October 24$^{\text{th}}$, 2022.

sectionsbrlinonsd

Solid **B**ody **R**otation **LIN**ear **ONS**et of Thermal Convection **D**rifting Frame Solve

Solves iteratively for a critical Rayleigh number, c_h^c in the onset of thermal convection problem

$$c_a \sigma \Theta \quad = \quad c_d \nabla^2 \Theta + b_1 u_r r + b_2 \frac{u_r}{r^2} \tag{1}$$

and

$$c_e \sigma \nabla \times \boldsymbol{u} \quad = \quad -c_g \nabla \times (\boldsymbol{k} \times \boldsymbol{u}) + c_h \nabla \times [\Theta \boldsymbol{r}] + c_i \nabla \times (\nabla^2 \boldsymbol{u}) \,. \tag{2}$$

which is fully described for the program `linons1`. This program is very similiar to `sbrlinons1` and differs only in that, instead of using one drifting longitudinal frame of reference for all iterations, it modifies the strength of the applied solid body rotation in order to solve the solution as being steady in the drifting frame. Most of the details for this program are omitted and the user is referred to that documentation.

The stand-alone source code version of the program is is compiled by typing

```
make sbrlinonsd
```

within this directory.

Once the executable is created, begin execution by typing

```
sbrlinonsd < inputfile
```

The inputs file must have the following format.

```
* input file for sbrlinonsd
example_aOUTPUT                           : ROOT
0.666666666   1.666666666  2   1   145 : RI, RO, IVELBC, ITHEBC, LU
 200.0  2  20    12   0.0010  1       : DRSV  NEV  NCV  MXATT CTOL  IA
*---------------------------------------------------------------------------
*     NR  ISP  LH  SYM  M  IOF  REYSB1  REYSB2
* CA    CB1   CB2   CD    CE   CG    CH1    CH2     CI
*---------------------------------------------------------------------------
      40   2   24   1   4   1   110.0   115.0
 0.01   1.0   0.0   1.0   1.0  1000.0 1930.0  1945.0  1.0
*---------------------------------------------------------------------------
      40   2   24   1   6   1  -17.0   -18.0
 0.1    1.0   0.0   1.0   1.0  1000.0 4420.0  4440.0  1.0
*---------------------------------------------------------------------------
      40   2   24   1   6   1  -5.0    -6.0
 1.0    1.0   0.0   1.0   1.0  1000.0 13150.0 13250.0  1.0
*---------------------------------------------------------------------------
      40   2   24   1   6   1  0.0     -1.0
 10.0   1.0   0.0   1.0   1.0  1000.0 20180.0 20300.0  1.0
*---------------------------------------------------------------------------
```

The details of the input file are almost identical to those for `sbrlinons1` except that instead of a single value, `REYSBR`, there are two distinct guesses, `REYSB1` and `REYSB2`. From the third iteration onwards, the imposed solid body rotation is altered in order to try and converge to a frame of reference in which the solution is stationary. Note that only the real part of the growth rate is used to vary the Rayleigh number; the imaginary part merely changes the frame of reference.

0.1 Subprograms required for sbrlinonsd

SUBS subroutines

```
fopen.f esnaas.f zcpaas.f nphpf.f gauwts.f schnla.f
vthmsr.f cindsw.f vecop.f svfdcf.f iocrbd.f hmfwt.f
svfwt.f xarrwt.f svprnt.f fclose.f fnamer.f ldgnmf.f
gfdcfd.f avmato.f sbrrfc.f sbrvmr.f vmeps.f evalas.f
evecex.f radvlf.f matop.f amlp.f  amlc.f  amccfo.f
amta.f  amcl.f  amhst.f iv0gto.f ivgt0o.f iv0cvo.f
ivcv0o.f bmrcop.f amsdea.f dvecz.f asvta.f asvcl.f
vesr.f  asvcpl.f amdlt.f amlica.f amccft.f corcoo.f
amhsar.f invgtt.f shveco.f vfdp.f  forsso.f innlca.f
invcvt.f vfcp.f  vf2qso.f asvdr.f matind.f vfcor.f
cubeop.f fftrlv.f powtwo.f
```

SUBS double precision function

```
pmm.f   pmm1.f  plm.f   dpmm.f  dpmm1.f dplm.f
emmult.f dl.f    sqrll1.f
```

SUBS integer function

```
indfun.f indshc.f
```

BLAS double precision function

```
dnrm2.f ddot.f  dasum.f
```

BLAS integer function

```
idamax.f
```

BLAS subroutines

```
daxpy.f dgemm.f dtrsm.f dgemv.f dswap.f dcopy.f
dger.f  dscal.f dtrmm.f dtbsv.f drot.f  dtrmv.f
```

ARPACK subroutines

```
dnaupd.f dneupd.f dnaup2.f dvout.f ivout.f second.f
dstatn.f dmout.f dgetv0.f dnaitr.f dnconv.f dneigh.f
dngets.f dnapps.f dlaqrb.f dsortc.f
```

LAPACK subroutines

```
dgetrf.f dgetri.f dgbtrf.f dgetf2.f dlaswp.f xerbla.f
dtrtri.f dgbtf2.f dgbtrs.f dlahqr.f dgeqr2.f dlacpy.f
dlaset.f dorm2r.f dtrevc.f dtrsen.f dtrti2.f dlabad.f
dlanv2.f dlarfg.f dlarf.f dlaln2.f dlacon.f dtrexc.f
dtrsyl.f dlarnv.f dlascl.f dlartg.f dlassq.f dladiv.f
dlaexc.f dlasy2.f dlaruv.f dlarfx.f
end{verbatim}

{\bf LAPACK double precision function}
\begin{verbatim}
dlapy2.f dlamch.f dlanhs.f dlange.f
```

LAPACK integer function

```
ilaenv.f
```

LAPACK logical function

```
lsame.f
```

0.2 Outputs from SBRLINONSD

These are identical to outputs from `sbrlinons1`.

0.3 Sample runs of sbrlinonsd

The directory

`$LEOPACK_DIR/SAMPLERUNS/SBRLINONSD`

contains example input files and model output. Do not under any circumstances edit these files, as these examples should serve as a control for the correct working of the code. After compiling the program, copy the `.input` files to another directory, run the code and confirm that the output agrees with that in the directory.

0.3.1 Example a

This is simply a confirmation of the results from `sbrlinons1` we have here reproduced very well the results of [ZB87]. The `.res` file reads

```
 4  24  40  1.00000000D+03  1.94374552D+03 -6.319775D-08 -3.54784917D-06  1.13416712D+02
 6  24  40  1.00000000D+03  4.43140102D+03 -2.483457D-05 -2.32864772D-05 -1.71162271D+01
 6  24  40  1.00000000D+03  1.32023484D+04  5.101902D-07  0.00000000D+00 -5.71674234D+00
 6  24  40  1.00000000D+03  2.02540012D+04 -5.012929D-04  0.00000000D+00 -6.77038814D-01
```

Column number 7 is the imaginary part of the growth rate and is very small (and identically zero in the last two cases). Our drifting solutions are therefore steady in the drifting frames of reference, defined by the imposition of a solid body rotation with strength defined by the number in column 8. The corresponding drift-rates, c, for our solutions are therefore, -113.4, 17.11, 5.717 and 0.6770, in good agreement with [ZB87].

0.3.2 Example b

We seek the critical Rayleigh number for the parameters described in [CAC$^+$01]. The file `example_bOUTPUT.res` reads

```
 4  36  50  2.00000000D+03  3.84810991D+04 -1.433039D-05 -2.49482699D-05 -9.08582326D+00
```

and we see that the critical Rayleigh number is approximately 38481 and the rolls are drifting prograde with a drift-rate of 9.086. I leave it as an exercise for the user to check critical Rayleigh numbers for other modes, m, and to check the numerical convergence of these results.

References

[CAC+01] U. R. Christensen, J. Aubert, P. Cardin, E. Dormy, S. Gibbons, G. A. Glatzmaier, E. Grote, Y. Honkura, C. Jones, M. Kono, M. Matsushima, A. Sakuraba, F. Takahashi, A. Tilgner, J. Wicht, and K. Zhang. A numerical dynamo benchmark. *Phys. Earth Planet. Inter.*, 128:25–34, 2001.

[ZB87] K. Zhang and F. H. Busse. On the onset of convection in rotating spherical shells. *Geophys. Astrophys. Fluid Dyn.*, 39:119–147, 1987.