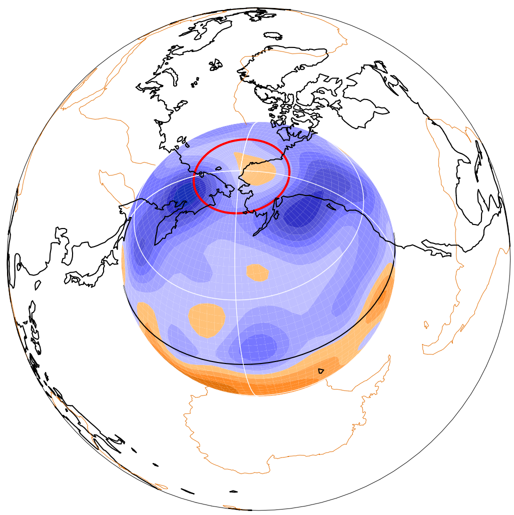
LEOPACK



ps_2plot_z_eq_merid2

Two side by side plots of constant z or meridian slices

Steven J. Gibbons, Oslo

Original document: November 21^{st}, 2001. Updated: October 29^{th}, 2022.

1 ps_2plot_z_eq_merid2

Source code is in

`LEOPACK_DIR/GPROGRAMS/ps_2plot_z_eq_merid2.f`

although subprograms from `LEOPACK_DIR/GSUBS`, `LEOPACK_DIR/SUBS` and `LEOPACK_DIR/LINALG` are also required.

`ps_2plot_z_eq_merid2` is a rather crude device for doing 2 side-by-side semi-circular contour plots of different functions from the same solution vector in standard format (i.e. `.ints`, `.vecs` and `.xarr` files). As the program stands, it is rather inflexible in terms of sizing and scaling plots and I recommend using `arrows_z_eq_merid4`

A typical input file is

```
* input file for ps_2plot_z_eq_merid2
*
example_aOUTPUT                          : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 60    60   3                            : NRAD  NTHE   NNDS
  -0.4999 0.49999    1    0.5            : tfirst  tlast  iview  coord
 14   6    1    4   6   0                : nlev idev  icont  icomp1 icomp2 ias
 120.0   0.0  1.0   1.00   1             : huepos, hueneg, csat, scal, iw
  1      0.0     0.0                     : icontour1  valmin1   valmax1
  1      0.0     0.0                     : icontour2  valmin2   valmax2
```

Any line in the input file beginning with an asterisk, `*`, is ignored by the program and can thus be used to enter comments and notes.

The inputs in the above file are as follows

- `Filename stem`. First characters in output files to be generated by current run. Running `ps_2plot_z_eq_merid2` with the above input file will create either the file `example_aOUTPUT.ps` or `example_aOUTPUT.gif`: depending upon the value of the integer flag `idev`.

- `integers`: name of already existing indices file describing solution.

- `vectorfile`: name of already existing vector file describing solution. Must contain the same number of radial functions as indicated in the `.ints` file.

- `radialfile`: name of already existing radial spacings file describing solution. Must contain the same number of radial grid nodes as indicated in the `.vecs` file.

- `NRAD`: Number of evenly spaced radial grid nodes to be used for polar contour plot. Note that this can be set irrespective of how many (arbitrarily spaced) grid nodes there were in the original solution vector. The solution is interpolated onto the specified regular mesh on reading in the solution.

- `NTHE`: Number of evenly spaced angular grid nodes to be used for polar contour plot. This can be set irrespective of the spectral resolution of the original solution vector.
  If `iview = 1`, then we are doing a meridian section and `NTHE` refers to points in latitude.
  If `iview = 2`, then we are doing a constant θ section and `NTHE` refers to points in longitude.
  If `iview = 3`, then we are doing a constant ϕ section and `NTHE` refers to points in the cylindrical polar angle.

- `NNDS`. Number of nodes for interpolating radial functions. `3` is a suggested value since no great accuracy is required here.

- `tfirst`: The lowest value of the angle to be contoured. This variable is always multiplied by π when plotting is performed to give a value in radians.

- `tlast`: The highest value of the angle to be contoured. This variable is always multiplied by π when plotting is performed to give a value in radians.
  If `iview` is set to 1, we are performing a meridian section and `tfirst` and `tlast` must be in the range $(-0.49999, 0.49999)$ (always smaller than 0.5 to avoid singular behaviour at the poles). Otherwise, I suggest `tfirst = -5.0`, `tlast = 5.0`.

- `iview`. The section to be studied. Options are:

  1. Meridian section at fixed value ϕ which is specified by setting `coord` to ϕ/π. If we want to contour from latitude λ_1 to latitude λ_2 then set `tfirst` to λ_1/π and set `tlast` to λ_2/π.
  2. Constant theta section where θ is specified by setting `coord` to θ/π. This is almost always an equatorial sectionm with `coord` set to `0.5`. If we want to contour from longitude ϕ_1 to longitude ϕ_2 then set `tfirst` to ϕ_1/π and set `tlast` to ϕ_2/π.
  3. Constant z section where z, the height above the equator is the value of `coord`. `tfirst` and `tlast` as for `iview = 2`.

- `coord`. Fixed value set. Takes on a different meaning depending upon the flag `iview`. (See above.)

- `nlev`. The number of contour levels required. There is a special value `nlev = -1` which applies a 16 contour level Red/Green/Blue coefficient set provided by Andy Jackson.

- `idev`. Device number. Can take the following values:-
  `idev = 1` $\rightarrow$ landscape gif file.

`idev = 2` → portrait gif file.
`idev = 5` → landscape colour postscript file.
`idev = 6` → portrait colour postscript file.

- `icont`. Specification of what to display.
  `icont = 1` → coloured contours (shading) without contour lines.
  `icont = 2` → contour lines without colours.
  `icont = 3` → contour lines and colours.
- `icomp1`. Field component to be displayed in the left-most contour plot.
- `icomp2`. Field component to be displayed in the right-most contour plot. The values of both `icomp1` and `icomp2` are taken from the following options:

  The value specified by `icomp_` varies depending upon which section (value of `iview`) is specified. The following values are common to all values of `iview`:
    `icomp_ = 1` → radial velocity, v_r.
    `icomp_ = 2` → theta velocity, v_θ.
    `icomp_ = 3` → phi velocity, v_ϕ.
    `icomp_ = 4` → radial magnetic field, B_r.
    `icomp_ = 5` → theta magnetic field, B_θ.
    `icomp_ = 6` → phi magnetic field, B_ϕ.
    `icomp_ = 7` → temperature, T.
  If `iview = 1`, then the following are options:
    `icomp_ = 8` → heat-flux: $-(dT/dr)$.
    `icomp_ = 9` → upwelling: $-(v_r/dr)$.
    `icomp_ = 10` → magnetic ϕ stream function:

$$F_\theta = \frac{r}{\sin\theta}\frac{\partial^P B}{\partial\phi}$$

  (See [ZB87]).
    `icomp_ = 11` → velocity ϕ stream function:

$$F_\theta = \frac{r}{\sin\theta}\frac{\partial^P v}{\partial\phi}$$

  (See [ZB87]).
  If `iview = 2`, then the following are options:
    `icomp_ = 11` → magnetic θ stream function:

$$F_\theta = \frac{r}{\sin\theta}\frac{\partial^P B}{\partial\phi}$$

(See [ZB87]).

`icomp_ = 12` $\rightarrow$ velocity θ stream function:

$$F_\theta = \frac{r}{\sin\theta}\frac{\partial^P v}{\partial\phi}$$

(See [ZB87]).

If `iview = 3`, then the following are options:

`icomp_ = 8` $\rightarrow$ heat-flux: $-(dT/dr)$.

`icomp_ = 9` $\rightarrow$ upwelling: $-(v_r/dr)$.

- `ias`. Axisymmetric only flag.
  `ias = 0` $\rightarrow$ full 3D solution is used.
  `ias = 1` $\rightarrow$ only the axisymmetric part is used.

- `huepos`. Hue value for numbers greater than zero for functions which are to be contoured using fill (i.e for options `icont = 1` and `icont = 3`, but otherwise not referred to). Number between `0` and `360`. See Appendix (2) and Figure (4) for details.

- `hueneg`. Hue value for numbers less than zero for functions which are to be contoured using fill (i.e for options `icont = 1` and `icont = 3`, but otherwise not referred to). Number between `0` and `360`. See Appendix (2) and Figure (4) for details.

- `csat`. Saturation value for shaded contours (i.e for options `icont = 1` and `icont = 3`, but otherwise not referred to). Number between `0.0` and `1.0` `csat = 1.0` implies full colour. `csat = 0.0` means monochrome and, in this case, the values `huepos` and `hueneg` become irrelevant and a grey-shade plot results with the most negative value as white and the most positive value as black. See Appendix (2) for details.

- `scal`. A very crude means of lightening a dark plot. Normal value is `scal = 1`, but reducing this (e.g. `scal = 0.7`) may give a better picture.

- `iw`. Width of lines used to draw arrows. Integer, with `1` being the thinnest available.

- `icontour1/2`. Chooses automatic or manual scaling of contours. 1 applies to the left-hand plot, 2 to the right-hand plot.
  `icontour1/2 = 1` $\rightarrow$ contours are scaled automatically and the values `valmin` and `valmax` become irrelevant.
  `icontour1/2 = 2` $\rightarrow$ contours are scaled between the following values, `valmin` and `valmax`.

- `valmin`. User-imposed minimum value for contour function. Only referred to if `icontour1/2 = 2`.
- `valmax`. User-imposed maximum value for contour function. Only referred to if `icontour1/2 = 2`.

1.1 Run-time limitations

Several parameters are set at the outset which limit the physical size of the problem.

```
      INTEGER NRMAX, NTMAX, NLEVM, LHMAX, NHMAX, ISVMAX, NNDM,
     1        NPMAX
      PARAMETER ( NRMAX = 250, NTMAX = 250, NLEVM = 20,
     1            LHMAX = 160, NHMAX = 3000, ISVMAX = NRMAX*NHMAX,
     2            NNDM = 6, NPMAX = (LHMAX+1)*(LHMAX+2)/2 )
```

If the values are insufficient, then change them and recompile.

- `NRMAX` is the maximum permitted number of radial grid nodes.
- `NTMAX` is the maximum permitted number of grid nodes in angle.
- `NLEVM` is the maximum permitted number of contour levels.
- `LHMAX` is the highest permitted spherical harmonic degree, l.
- `NHMAX` is the highest permitted number of spherical harmonic radial functions.
- `NNDM` is the highest permitted value of `nnds`.

1.2 Sample runs of ps_2plot_z_eq_merid2

The directory

`$LEOPACK_DIR/SAMPLERUNS/PS_2PLOT_Z_EQ_MERID2`

contains example input files only. Do not under any circumstances edit these files. They refer to other (solution vector) files which are elsewhere in the distribution and provide a relative path to avoid unnecessary duplication of files. The outputs from the different files are displayed here rather than left in the directory.

1.2.1 Example a

```
* input file for ps_2plot_z_eq_merid2
*
example_aOUTPUT                          : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 60    60   3                            : NRAD  NTHE   NNDS
  -0.4999 0.49999    1    0.5            : tfirst  tlast  iview  coord
 14   6    1    4    6   0               : nlev idev  icont  icomp1 icomp2 ias
 120.0   0.0  1.0   1.00   1             : huepos, hueneg, csat, scal, iw
  1      0.0     0.0                     : icontour1  valmin1   valmax1
  1      0.0     0.0                     : icontour2  valmin2   valmax2
```

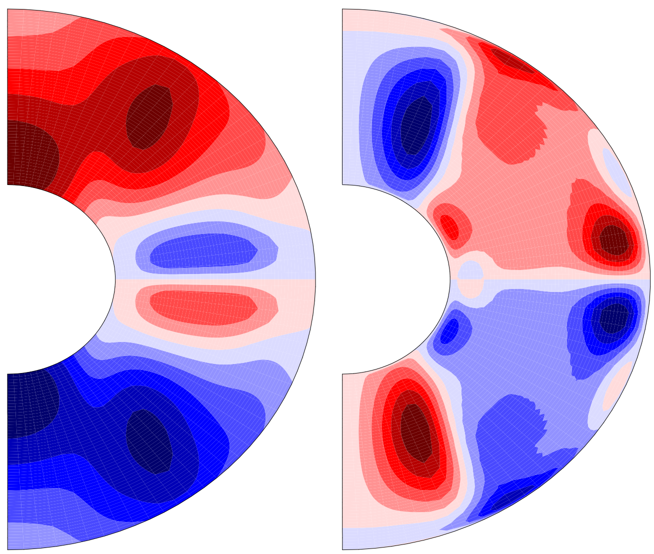


Figure 1: Output from ps_2plot_z_eq_merid2 with example_a.input

1.2.2 Example b

```
* input file for ps_2plot_z_eq_merid2
*
example_bOUTPUT                          : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 60    60   3                            : NRAD  NTHE   NNDS
  -0.5     0.5        2    0.5           : tfirst  tlast  iview  coord
 14   6    3    1    7   0               : nlev idev  icont  icomp1 icomp2 ias
 120.0 240.0  1.0   1.00   3             : huepos, hueneg, csat, scal, iw
  1      0.0     0.0                     : icontour1  valmin1   valmax1
  1      0.0     0.0                     : icontour2  valmin2   valmax2
```

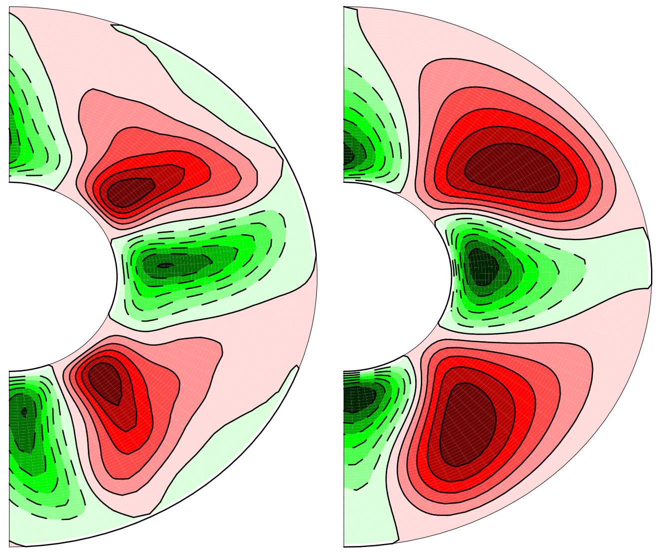


Figure 2: Output from ps_2plot_z_eq_merid2 with example_b.input

1.2.3 Example c

```
* input file for ps_2plot_z_eq_merid2
*
example_cOUTPUT                              : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints       : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs       : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr       : radialfile
 60    60   3                                : NRAD  NTHE   NNDS
  -0.5    0.5         3    0.5               : tfirst  tlast  iview  coord
 14   6    3    4   6   0                    : nlev idev  icont  icomp1 icomp2 ias
 120.0   0.0  0.0   1.00   1                 : huepos, hueneg, csat, scal, iw
  1      0.0     0.0                         : icontour1  valmin1   valmax1
  1      0.0     0.0                         : icontour2  valmin2   valmax2
```

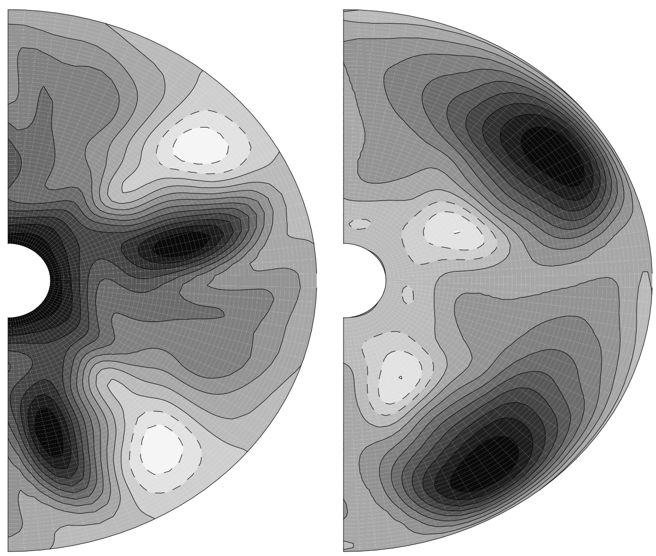


Figure 3: Output from ps_2plot_z_eq_merid2 with example_c.input

2 The HLS colour scheme

When plotting using the PGPLOT software, a colour is specified by either one of the two calls

```
CALL PGSHLS( IND, CH, CL, CS )
```

or

```
CALL PGSCR( IND, CR, CG, CB )
```

The integer `IND` is the index of the colour being applied. `CR`, `CG` and `CB` are respectively the red, green and blue values in the ranges $[0, 1]$.

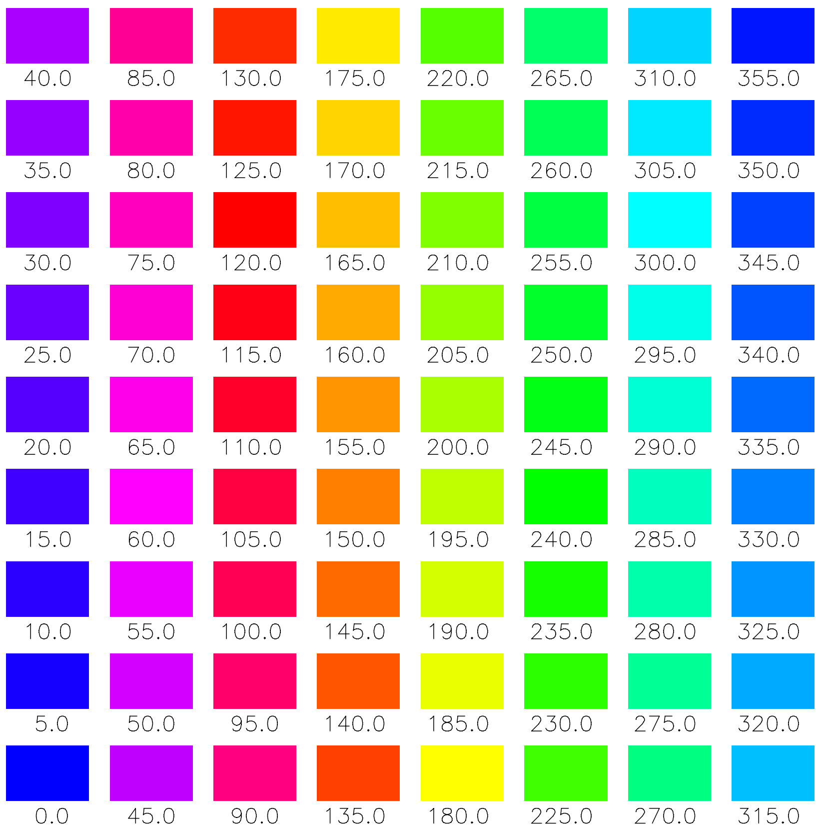


Figure 4: Colours as described by the integer HUE in the HLS (Hue, Light and Saturation) colour scheme.

The alternative HLS (Hue, Light and Saturation) system takes three real values

- `CH`. Hue. This is an angle between zero and 360 degrees which specifies the colour. Red is 120, Green is 240 and Blue is 0 (or 360). The full spectrum, in intervals of 5 degrees, is displayed in Figure (4).
- `CL`. Light. Ranges from 0.0 to 1.0 with black at lightness 0.0 and white at lightness 1.0.
- `CS`. Saturation. Ranges from 0.0 (grey) to 1.0 (pure colour). Hue is irrelevant when saturation is 0.0

I opted for the HLS system for the general graphics system - not because I thought the results were better - but because it is simply much easier to apply. I generally set one hue value for positive values and one for negative values and then vary the lightness as a function of the numbers being plotted.

Other users may find this colour scheme unappealing and so are welcome to devise a better way of assigning colours to contour levels! I did a job for Andy Jackson last year, for which he gave me a set of 16 red, green and blue (RGB) coefficients. This scheme is very nice and so I have implemented it in the majority of the codes as a special value of `NLEV` (the number of contour levels). Setting `NLEV = -1` should implement this colour scheme, resulting in 16 contour levels. I never got round to implementing any more general RGB scheme.

References

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