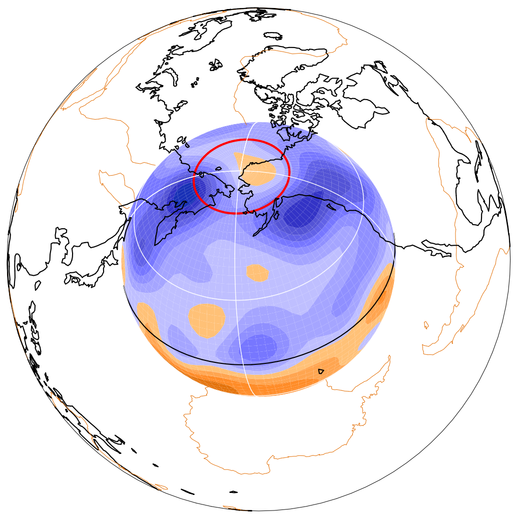
LEOPACK



arrows_z_eq_merid4

Constant z, equatorial, or meridian plot with arrows

Steven J. Gibbons, Oslo

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

1 arrows_z_eq_merid4

Source code is in

`LEOPACK_DIR/GPROGRAMS/arrows_z_eq_merid4.f`

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

`arrows_z_eq_merid4` plots - from a solution in standard format (i.e. `.ints`, `.vecs` and `.xarr` files) - a section through the spherical shell at either constant ϕ (does NOT contour the section $\phi + \pi$ simultaneously), constant θ (most usefully at the equator) or constant z. Arrows indicating the direction of flow in this section may also be plotted.

A typical input file is

```
* input file for arrows_z_eq_merid4
example_aOUTPUT                          : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 40   175   3                            : NRAD  NTHE   NNDS
 0.10  0.90   0.10   0.90    7.0   1.0   : xleft xright ybot ytop rwidth rratio
 0.0000   2.0000     2    0.5            : tfirst  tlast  iview  coord
 14   6    3    5   0                    : nlev idev  icont  icomp   ias
   20.0 165.0 1.0   1.00   3             : huepos, hueneg, csat, scal, iw
 120.0   0.0   0.75  5    1              : rphue rpsat rplight ipw ips
   0.0   0.0   0.75  5    4              : rnhue rnsat rnlight inw ins
  4   0.20   1.2                         : npba  rlong  rhead
   1       0.0       0.0                 : icontour  valmin   valmax
```

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 `arrows_z_eq_merid4` 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.

- `xleft`. Position within the output device of the left border of figure. (i.e. defines size of the left hand side margin.) Must be in the range $[0, 1]$.

- `xright`. Position within the output device of the right border of figure. (i.e. defines size of the right hand side margin.) Must be in the range $[0, 1]$.

- `ybot`. Position within the output device of the lower border of figure. (i.e. defines size of the bottom margin.) Must be in the range $[0, 1]$.

- `ytop`. Position within the output device of the upper border of figure. (i.e. defines size of the top margin.) Must be in the range $[0, 1]$.

- `rwidth`. Width in inches of output device (i.e. horizontal dimension of postscript of gif file.)

- `rratio`. Ratio of height to width for the output device.

- `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. For example, if `iview` is set to 2 or 3 and we wish to display a full "round the sphere" section, then `tfirst` and `tlast` are respectively set to `0.0` and `2.0`, giving a range $0 \leq \phi \leq 2\pi$. If we wish to plot an equatorial section between longitudes 45° and 135° only, then we set `tfirst = 0.25` and `tlast = 0.75`. 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). `arrows_z_eq_merid4` cannot extend a meridian section beyond a half-plane.

- `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` $\rightarrow$ portrait gif file.
  `idev = 5` $\rightarrow$ landscape colour postscript file.
  `idev = 6` $\rightarrow$ portrait colour postscript file.

- `icont`. Specification of what to display.
  `icont = 1` $\rightarrow$ coloured contours (shading) without contour lines or arrows.
  `icont = 2` $\rightarrow$ arrows of flow without contours.
  `icont = 3` $\rightarrow$ arrows of flow superimposed upon coloured contours (shading).
  `icont = 4` $\rightarrow$ arrows of flow superimposed upon contour lines.

- `icomp`. Field component to be displayed in the contour plot. 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` $\rightarrow$ radial velocity, v_r.
  `icomp = 2` $\rightarrow$ theta velocity, v_θ.
  `icomp = 3` $\rightarrow$ phi velocity, v_ϕ.

`icomp = 4` $\rightarrow$ radial magnetic field, B_r.
`icomp = 5` $\rightarrow$ theta magnetic field, B_θ.
`icomp = 6` $\rightarrow$ phi magnetic field, B_ϕ.
`icomp = 7` $\rightarrow$ temperature, T.

If `iview = 1`, then the following are options:
`icomp = 8` $\rightarrow$ heat-flux: $-(dT/dr)$.
`icomp = 9` $\rightarrow$ upwelling: $-(v_r/dr)$.
`icomp = 10` $\rightarrow$ magnetic ϕ stream function:

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

(See [ZB87]).
`icomp = 11` $\rightarrow$ velocity ϕ stream function:

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

(See [ZB87]).

If `iview = 2`, then the following are options:
`icomp = 11` $\rightarrow$ 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 (7) 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 (7) 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.

- `rphue`. Hue value for numbers greater than zero for functions which are to be contoured using lines (i.e for option `icont = 4` but otherwise not referred to). Number between `0` and `360`. See Appendix (2) and Figure (7) for details.

- `rpsat`. Saturation value for contour lines with positive values. Only referred to for option `icont = 4`. Number between `0.0` and `1.0` `rpsat = 1.0` implies full colour. `rpsat = 0.0` means monochrome and, in this case, the value `rphue` becomes irrelevant.

- `rplight`. Lightness value for contour lines with positive values. Only referred to for option `icont = 4`. Number between `0.0` and `1.0` with black at lightness 0.0 and white at lightness 1.0.

- `ipw`. Width value for contour lines with positive values. Only referred to for option `icont = 4`. Integer, with `1` being the thinnest available.

- `ips`. Style value for contour lines with positive values. Only referred to for option `icont = 4`. The following options are available:
  `ips = 1` → full line.
  `ips = 2` → long dashes.
  `ips = 3` → dash-dot-dash-dot.
  `ips = 4` → dotted.
  `ips = 5` → dash-dot-dot-dot.

- `rnhue`. Hue value for numbers less than zero for functions which are to be contoured using lines (i.e for option `icont = 4` but otherwise not referred to). Number between `0` and `360`. See Appendix (2) and Figure (7) for details.

- `rnsat`. Saturation value for contour lines with negative values. Only referred to for option `icont = 4`. Number between `0.0` and `1.0` `rnsat = 1.0` implies full colour. `rnsat = 0.0` means monochrome and, in this case, the value `rphue` becomes irrelevant.

- `rnlight`. Lightness value for contour lines with negative values. Only referred to for option `icont = 4`. Number between `0.0` and `1.0` with black at lightness 0.0 and white at lightness 1.0.

- `inw`. Width value for contour lines with negative values. Only referred to for option `icont = 4`. Integer, with `1` being the thinnest available.

- `ins`. Style value for contour lines with negative values. Only referred to for option `icont = 4`. The following options are available:
  `ins = 1` $\rightarrow$ full line.
  `ins = 2` $\rightarrow$ long dashes.
  `ins = 3` $\rightarrow$ dash-dot-dash-dot.
  `ins = 4` $\rightarrow$ dotted.
  `ins = 5` $\rightarrow$ dash-dot-dot-dot.

- `npba`. Number of points between arrows. The starting point of an arrow is drawn every `npba` points in both radius and angle. The lower the value for `npba`, the more arrows there will be. If arrows are too concentrated in both directions, increase `npba`. If arrows are too concentrated in radius only, then decrease `NRAD` with the same `npba` value, or increase `npba` and increase `NTHE` proportionately. If arrows are too concentrated in angle only, then decrease `NTHE` with the same `npba` value, or increase `npba` and increase `NRAD` proportionately. A certain amount of trial and error is required in selecting verb+npba+, `NRAD` and `NTHE` such that a good spread of arrows is achieved.

- `rlong`. Length of longest arrow (which automatically corresponds to the greatest flow). All other arrows are scaled relative to this.

- `rhead`. Size of the biggest arrowhead.

- `icontour`. Chooses automatic or manual scaling of contours.
  `icontour = 1` $\rightarrow$ contours are scaled automatically and the values `valmin` and `valmax` become irrelevant.
  `icontour = 2` $\rightarrow$ contours are scaled between the following values, `valmin` and `valmax`.

- `valmin`. User-imposed minimum value for contour function. Only referred to if `icontour = 2`.

- `valmax`. User-imposed maximum value for contour function. Only referred to if `icontour = 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 = 148, NHMAX = 3800, 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 arrows_z_eq_merid4

The directory

`$LEOPACK_DIR/SAMPLERUNS/ARROWS_Z_EQ_MERID4`

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 arrows_z_eq_merid4
example_aOUTPUT                         : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints  : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs  : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr  : radialfile
 40   175   3                           : NRAD  NTHE   NNDS
 0.10  0.90   0.10   0.90    7.0   1.0  : xleft xright ybot ytop rwidth rratio
 0.0000   2.0000     2    0.5           : tfirst  tlast  iview  coord
 14   6    3    5   0                   : nlev idev  icont  icomp   ias
   20.0 165.0 1.0   1.00   3            : huepos, hueneg, csat, scal, iw
```

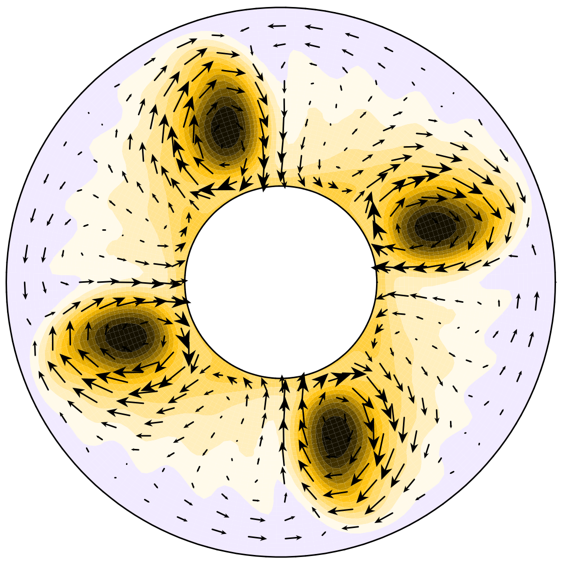


Figure 1: Output from arrows_z_eq_merid4 with example_a.input (Section 1.2.1).

```
 120.0   0.0   0.75  5    1              : rphue rpsat rplight ipw ips
   0.0   0.0   0.75  5    4              : rnhue rnsat rnlight inw ins
  4   0.20   1.2                         : npba  rlong  rhead
   1        0.0       0.0                : icontour  valmin   valmax
```

This example is clearly an equatorial section showing arrows of flow superimposed upon contours of B_θ for the Case 1 benchmark dynamo. Notice how the concentrations of the θ field component are encircled by flow.

1.2.2 Example b

```
* input file for arrows_z_eq_merid4
example_bOUTPUT                          : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 40   100   3                            : NRAD  NTHE   NNDS
 0.10  0.90   0.10   0.90    4.0   2.0   : xleft xright ybot ytop rwidth rratio
 -0.500    0.500     2    0.5            : tfirst  tlast  iview  coord
 14   6    4    7    0                   : nlev idev  icont  icomp   ias
   20.0 165.0 1.0   1.00   4             : huepos, hueneg, csat, scal, iw
 120.0   0.0   0.65  4    1              : rphue rpsat rplight ipw ips
   0.0   0.0   0.65  4    4              : rnhue rnsat rnlight inw ins
  4   0.16   2.0                         : npba  rlong  rhead
   1        0.0       0.0                : icontour  valmin   valmax
```

We consider now only a half of the equatorial plane of the same solution as in the previous section. This time, we have set `icont` to 4 and so only draw contour lines rather than contour shading. Setting the `rpsat` and `rnsat` to zero, we make it essentially a grayshade plot. The contour lines represent temperature.

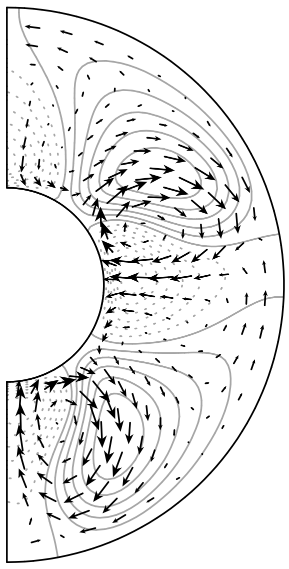


Figure 2: Output from arrows_z_eq_merid4 with example_b.input (Section 1.2.2).

1.2.3 Example c

```
* input file for arrows_z_eq_merid4
example_cOUTPUT                          : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 40   160   3                            : NRAD  NTHE   NNDS
 0.10  0.90   0.10   0.90    6.0   1.0   : xleft xright ybot ytop rwidth rratio
  0.000    1.950     3    0.500          : tfirst  tlast  iview  coord
 14   6    2    7   0                    : nlev idev  icont  icomp   ias
   20.0 165.0 1.0   1.00   4             : huepos, hueneg, csat, scal, iw
 120.0   0.0   0.65  4    1              : rphue rpsat rplight ipw ips
   0.0   0.0   0.65  4    4              : rnhue rnsat rnlight inw ins
  4   0.16   1.2                         : npba  rlong  rhead
   1        0.0       0.0                : icontour  valmin   valmax
```

Setting `icont = 2` means that we only consider the flow and no other function. This is at a constant z of 0.5. Inspecting the `.xarr` file which the solution consists of, reveals the inner core has a radius of $r_{\rm i} \approx 0.538$, and so we are only just slicing the top of the inner core. Note the difference in flow inside and outside of the tangent cylinder. Just to show how it is done, we omit the last 9° in longitude.

1.2.4 Example d

```
* input file for arrows_z_eq_merid4
example_dOUTPUT                          : Filename stem
```

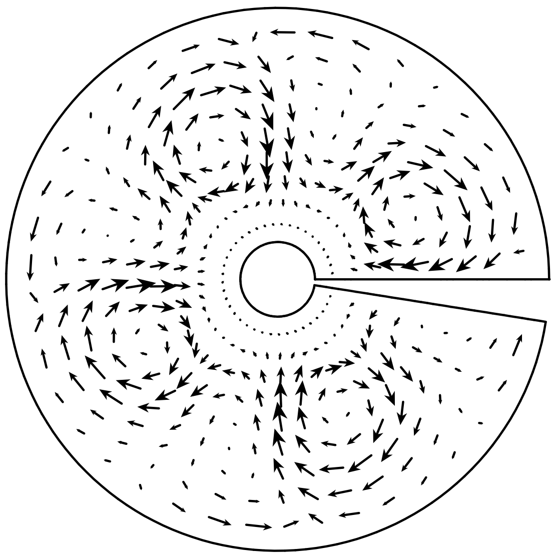


Figure 3: Output from arrows_z_eq_merid4 with example_c.input (Section 1.2.3).

```
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 40   160   3                            : NRAD  NTHE   NNDS
 0.10  0.90   0.10   0.90    6.0   1.0   : xleft xright ybot ytop rwidth rratio
  0.000    1.950     3    0.500          : tfirst  tlast  iview  coord
 -1   6    1    6   0                    : nlev idev  icont  icomp   ias
   20.0 165.0 1.0   1.00   4             : huepos, hueneg, csat, scal, iw
 120.0   0.0   0.65  4    1              : rphue rpsat rplight ipw ips
   0.0   0.0   0.65  4    4              : rnhue rnsat rnlight inw ins
  4   0.16   1.2                         : npba  rlong  rhead
   1        0.0       0.0                : icontour  valmin   valmax
```

Exactly the same as the previous example in terms of the section plotted. This is contours of B_ϕ with no arrows.

1.2.5 Example e

```
* input file for arrows_z_eq_merid4
example_eOUTPUT                          : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 40    90   3                            : NRAD  NTHE   NNDS
 0.10  0.90   0.10   0.90    4.0   2.0   : xleft xright ybot ytop rwidth rratio
 -0.49999  0.49999   1    0.000          : tfirst  tlast  iview  coord
 14   6    3    6   0                    : nlev idev  icont  icomp   ias
  120.0 240.0 1.0   1.00   4             : huepos, hueneg, csat, scal, iw
 120.0   0.0   0.65  4    1              : rphue rpsat rplight ipw ips
   0.0   0.0   0.65  4    4              : rnhue rnsat rnlight inw ins
  4   0.12   2.0                         : npba  rlong  rhead
   1        0.0       0.0                : icontour  valmin   valmax
```

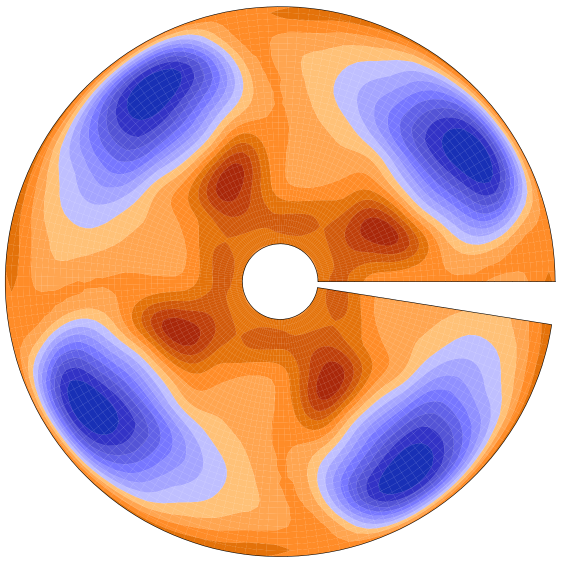


Figure 4: Output from arrows_z_eq_merid4 with example_d.input (Section 1.2.4).

A full meridian section at $\phi = 0$ showing arrows of flow superimposed upon contours of B_ϕ. The arrows are moving inwards, consistently with the direction of flow in the equatorial plane (Figures 1 and 2).

1.2.6 Example f

```
* input file for arrows_z_eq_merid4
example_fOUTPUT                         : Filename stem
../../EXAMPLES/FUNDAMENTALS/case1.ints   : integers
../../EXAMPLES/FUNDAMENTALS/case1.vecs   : vectorfile
../../EXAMPLES/FUNDAMENTALS/case1.xarr   : radialfile
 40    90   3                            : NRAD  NTHE   NNDS
 0.10  0.90   0.10   0.90    4.0   2.0   : xleft xright ybot ytop rwidth rratio
 -0.49999  0.49999   1    0.250          : tfirst  tlast  iview  coord
 14   6    4   10   0                    : nlev idev  icont  icomp   ias
  120.0 240.0 1.0    1.00    4           : huepos, hueneg, csat, scal, iw
 120.0    1.0    0.5    3    1           : rphue rpsat rplight ipw ips
   0.0    1.0    0.5    3    1           : rnhue rnsat rnlight inw ins
  4   0.12    2.0                        : npba  rlong  rhead
   1        0.0       0.0                : icontour  valmin   valmax
```

We have moved around to longitude $\phi = 45^\circ$ and plotted contours of the ϕ magnetic streamfunction (i.e. a representation of poloidal magnetic field lines) over arrows of flow. The arrows are moving outwards, consistently with the direction of flow in the equatorial plane (Figures 1 and 2).

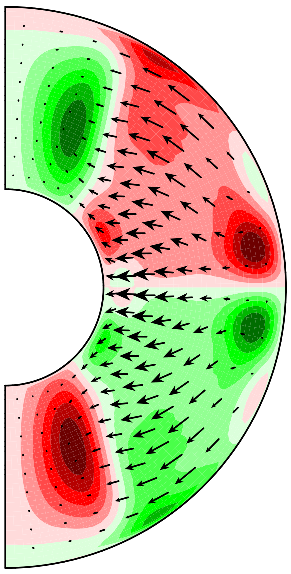


Figure 5: Output from arrows_z_eq_merid4 with example_e.input (Section 1.2.5).

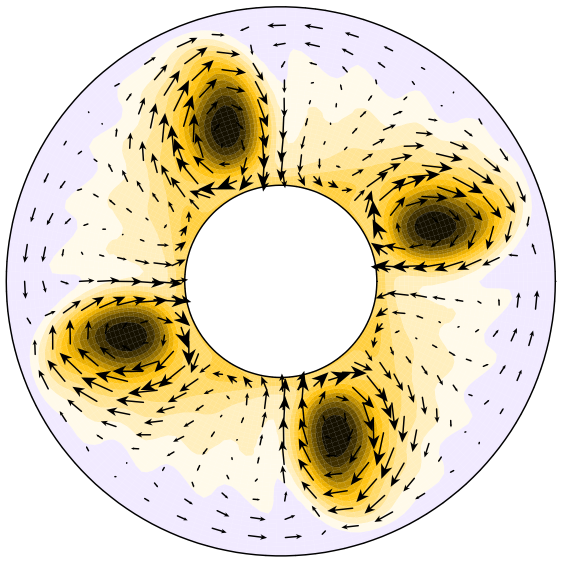


Figure 6: Output from arrows_z_eq_merid4 with example_f.input (Section 1.2.6).

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 $[0, 1]$.

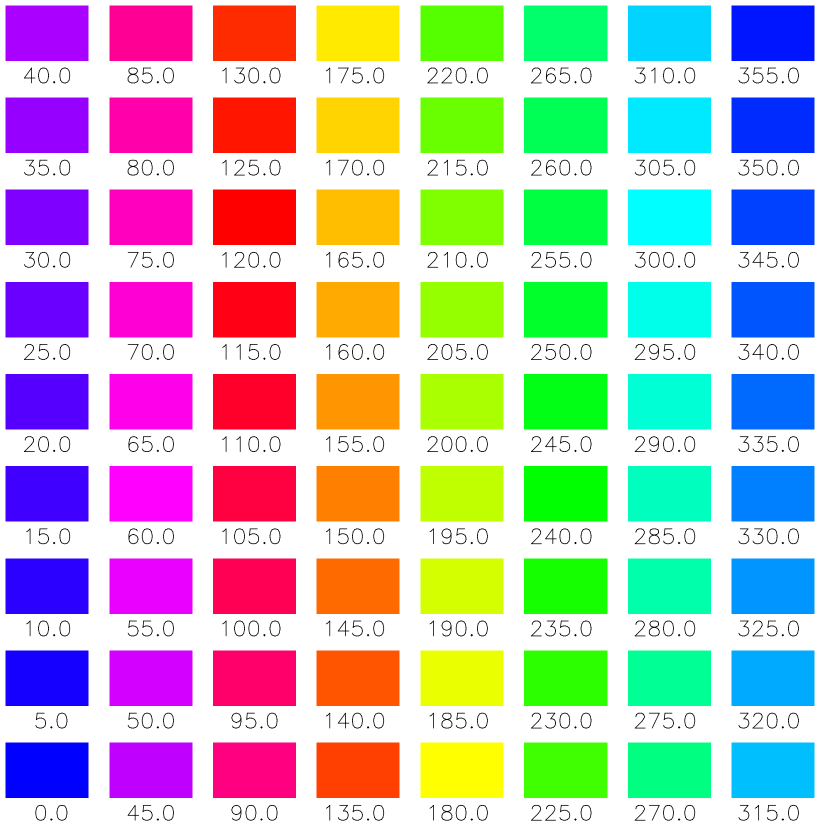


Figure 7: Colours as described by the integer HUE in the HLS (Hue, Light and Saturation) colour scheme.

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 (7).
- `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.