

# LEOPACK



## **cicm\_arrows\_z\_eq\_merid4**

Conducting Inner Core and Mantle constant  $z$ , equatorial,  
or meridian plot with arrows

Steven J. Gibbons, Oslo

Original document: November 21<sup>st</sup>, 2001. Updated: October 29<sup>th</sup>, 2022.

# 1 `cicm_arrows_z_eq_merid4`

Source code is in

`LEOPACK_DIR/GPROGRAMS/cicm_arrows_z_eq_merid4.f`

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

This program is essentially the equivalent to `arrows_z_eq_merid4` for solutions where the magnetic field extends into a conducting inner core. Solutions computed with programs such as `cicubcdts2` may be displayed using this program.

It requires a solution in standard format describing the velocity and temperature (i.e. a `.intsv`, `.vecsv` and `.xarrv` file) and a solution in standard format describing the magnetic field (i.e. a `.intsm`, `.vecsm` and `.xarrm` file). It can plot a section through the spherical shell at either constant  $\phi$  (does NOT contour the section  $\phi + \pi$  simultaneously), constant  $\theta$  (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 cic_arrows_z_eq_merid4
*
example_eOUTPUT      : Filename stem
../CICUBCDTS2/example_b.intsv : vel integers
../CICUBCDTS2/example_b.vecsv : vel vectorfile
../CICUBCDTS2/example_b.xarrv : vel radialfile
../CICUBCDTS2/example_b.intsm : mag integers
../CICUBCDTS2/example_b.vecsm : mag vectorfile
../CICUBCDTS2/example_b.xarrm : mag radialfile
40 160 3              : NRAD NTHE NNDS
0.02 0.98 0.02 0.48 7.0 1.0 : xleft xright ybot ytop rwidth rratio
0.0000 1.0000 3 0.25 : tfirst tlast iview coord
14 6 3 7 0           : nlev idev icont icomp ias
120.0 0.0 1.0 0.75 4 : 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 rnlght inw ins
4 0.14 1.4           : 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 `cic_arrows_z_eq_merid4` with the above input file will create either the file `example_eOUTPUT.ps` or `example_eOUTPUT.gif`: depending upon the value of the integer flag `idev`.
- **vel integers:** name of already existing indices file describing velocity.

- **vel vectorfile**: name of already existing vector file describing velocity. Must contain the same number of radial functions as indicated in the `.intsv` file.
- **vel radialfile**: name of already existing radial spacings file describing velocity. Must contain the same number of radial grid nodes as indicated in the `.vecsv` file.
- **mag integers**: name of already existing indices file describing magnetic field.
- **mag vectorfile**: name of already existing vector file describing magnetic field. Must contain the same number of radial functions as indicated in the `.intsm` file.
- **mag radialfile**: name of already existing radial spacings file describing magnetic field. Must contain the same number of radial grid nodes as indicated in the `.vecsm` 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  $\theta$  section and **NTHE** refers to points in longitude.  
 If `iview = 3`, then we are doing a constant  $\phi$  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 or 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  $\pi$  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  $\pi$  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  $45^\circ$  and  $135^\circ$  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  $\phi$  which is specified by setting **coord** to  $\phi/\pi$ . If we want to contour from latitude  $\lambda_1$  to latitude  $\lambda_2$  then set **tfirst** to  $\lambda_1/\pi$  and set **tlast** to  $\lambda_2/\pi$ .
  2. Constant theta section where  $\theta$  is specified by setting **coord** to  $\theta/\pi$ . This is almost always an equatorial section with **coord** set to 0.5. If we want to contour from longitude  $\phi_1$  to longitude  $\phi_2$  then set **tfirst** to  $\phi_1/\pi$  and set **tlast** to  $\phi_2/\pi$ .
  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 → landscape colour postscript file.

idev = 6 → portrait colour postscript file.

- **icont.** Specification of what to display.
    - icont = 1 → coloured contours (shading) without contour lines or arrows.
    - icont = 2 → arrows of flow without contours.
    - icont = 3 → arrows of flow superimposed upon coloured contours (shading).
    - icont = 4 → 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 → radial velocity,  $v_r$ .
    - icomp = 2 → theta velocity,  $v_\theta$ .
    - icomp = 3 → phi velocity,  $v_\phi$ .
    - icomp = 4 → radial magnetic field,  $B_r$ .
    - icomp = 5 → theta magnetic field,  $B_\theta$ .
    - icomp = 6 → phi magnetic field,  $B_\phi$ .
    - 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  $\phi$  stream function:

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

(See [ZB87]).

icomp = 11 → velocity  $\phi$  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 → magnetic  $\theta$  stream function:

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

(See [ZB87]).

`icom` = 12  $\rightarrow$  velocity  $\theta$  stream function:

$$F_\theta = \frac{r}{\sin \theta} \frac{\partial^2 v}{\partial \phi^2}$$

(See [ZB87]).

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

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

`icom` = 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 (2) 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 (2) 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 (2) 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 (2) 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 → full line.  
**ins** = 2 → long dashes.  
**ins** = 3 → dash-dot-dash-dot.  
**ins** = 4 → dotted.  
**ins** = 5 → 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 → contours are scaled automatically and the values **valmin** and **valmax** become irrelevant.  
**icontour** = 2 → 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, NHVMAX, NHMMAX,
1          IVVMAX, IVMMAX, NNDM, NPMAX
      PARAMETER ( NRMAX = 250, NTMAX = 250, NLEVM = 20,
1          LHMAX = 148, NHVMAX = 3800, NHMMAX = 3800,
2          IVVMAX = NRMAX*NHVMAX, IVMMAX = NRMAX*NHMMAX,
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.



- NLEV is the maximum permitted number of contour levels.
- LHMAX is the highest permitted spherical harmonic degree,  $l$ .
- NHVMAX is the highest permitted number of velocity/temperature spherical harmonic radial functions.
- NHMMAX is the highest permitted number of magnetic field spherical harmonic radial functions.
- NNNDM is the highest permitted value of `nnds`.

## 1.2 Sample runs of `cic_arrows_z_eq_merid4`

The directory

`$LEOPACK_DIR/SAMPLERUNS/CIC_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 cic_arrows_z_eq_merid4
*
example_aOUTPUT          : Filename stem
../CICUBCDTS2/example_b.intsv : vel integers
../CICUBCDTS2/example_b.vecsv : vel vectorfile
../CICUBCDTS2/example_b.xarrv : vel radialfile
../CICUBCDTS2/example_b.intsm : mag integers
../CICUBCDTS2/example_b.vecsm : mag vectorfile
../CICUBCDTS2/example_b.xarrm : mag radialfile
60 160 3                  : NRAD NTHE NNDS
0.02 0.98 0.02 0.98 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 iconv icomp ias
160.0 330.0 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          : rn timer rnsat rnlght inw ins
4 0.16 1.4                : npba rlong rhead
1 0.0 0.0                  : iconv valmin valmax
```

---

The first example shows coloured contours of  $B_\theta$  in the equatorial plane for a conducting inner-core solution with fixed heat-flux at the outer boundary. (The solution is very similar to, and based on, the Case 1 benchmark solution ([CAC<sup>+</sup>01]) - only with altered thermal boundary conditions and treatment for the inner core.

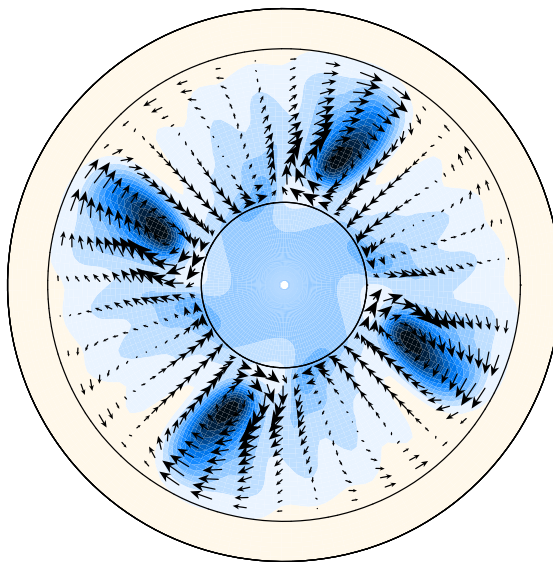


Figure 1: Output from `cic_arrows_z_eq_merid4` with `example_a.input` (Section 1.2.1).

## 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  $[0, 1]$ .

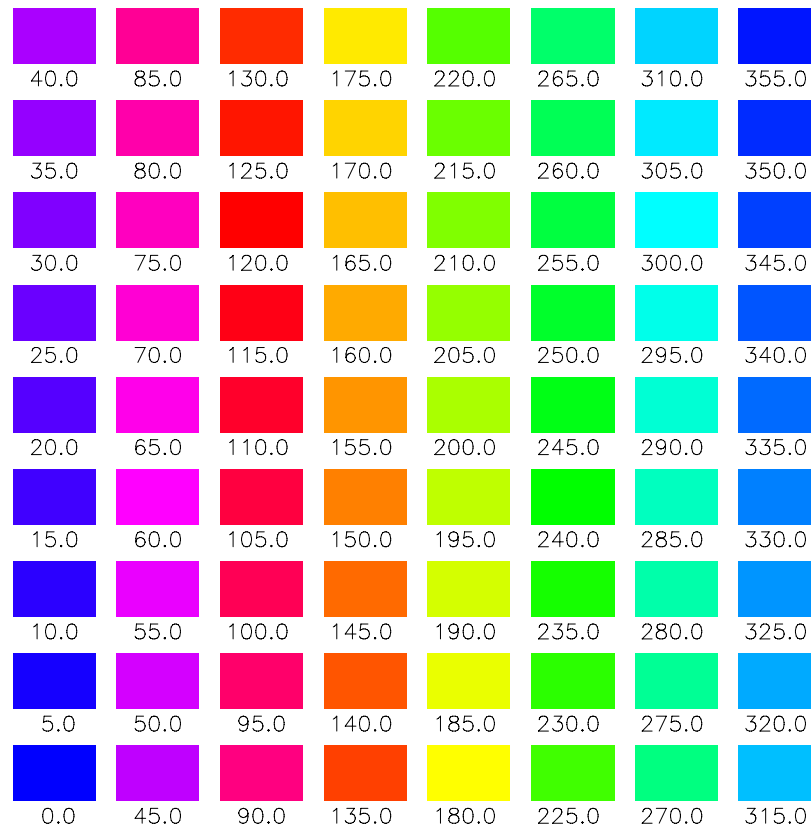


Figure 2: Colours as described by the integer HUE in the HLS (Hue, Light and Saturation) colour scheme.

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 (2).
- **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

- [CAC<sup>+</sup>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.