PARALLEL NETCDF output and VAPOR visualization of EULAG simulations

Zbigniew P. Piotrowski[1]

[1]Institute of Geophysics, University of Warsaw, Poland

Technical issues of Eulag simulations

- EULAG data I/O – alternatives to the Fortran tape
  - NETCDF input/output
  - Parallel NETCDF input/output
- Performance tests and scaling analysis with TAU
- Petascale computing visualization tool – VAPOR

NETCDF

- Popular - plenty of manipulation and visualization software available
- Possibility of including comments and descriptions in the data file
- Explicit information of the file contents, including fields and number of timesteps
- Simple data transport between Little Endian (e.g. PC) and Big Endian (e.g. IBM)
- Easier data access: Compact Disc versus Tape
- Serial write (whole matrices written by one proc) and “parallel serial” (matrix chunks written by each proc. with postprocessing needed) available

Parallel versions of NETCDF

- Easy to use parallel version of the NETCDF package
- Build on the top of MPI I/O technology
- Two branches currently available – Parallel NETCDF and NETCDF 4.0
- Competitive performance to the EULAG parallel Fortran tape and parallel write of serial NETCDF chunks
- No postprocessing necessary as in simultaneous serial NETCDF write

NETCDF output

Netcdf file create:

```
iret = nf_create('example.nc', NF_CLOBBER,nfd)
```

Defining dimension (x,y,z,t):

```
ier = nf_def_dim(nfd, 'x',n, nhand)
ier = nf_def_dim(nfd, 'y',m, mhand)
ier = nf_def_dim(nfd, 'z',l, lhand)
ier = nf_def_dim(nfd, 't',NF_UNLIMITED,thand)
```

Storing dimensions handles in the array:

```
hdimarr(1)=nhand
hdimarr(2)=mhand
hdimarr(3)=lhand
hdimarr(4)=thand
ndim=4
nfp=NF_REAL
```

Defining variables:

```
ier = nf_def_var(nfd,'x', nfp,pstart,nhand)
ier = nf_def_var(nfd,'w',nfp,ndim,whand)
```

Defining the position and the size of variable record:

```
istart(1)=1
istart(2)=1
istart(3)=1
istart(4)=iframe
icnt(1)=n
icnt(2)=m
icnt(3)=l
icnt(4)=1
```

End of the defining section:

```
ier = nf_enddef(nfd)
```

Writing data into the netcdf file:

```
ier = nf_put_vara_real(nfd,whand,istart,icnt,w)
```

Printing the debugging information about succes or fail

```
print *,'Put:',nf_strerror(ier)
ier = nf_close(nfd)
```

PNETCDF output

```
integer*8 xhand,whand
```

integer*8 pn,pm,pl,pt,podim,pndim

```
integer*8 nfd
integer nhand,mhand,lhand,thand
nfp = NF_REAL
pn=n
pm=m
pl=l
pt=(nt/noutp)+1
podimt=1
pndim=4
pstart=1
```

```
ier =                    nfmpi_create
( MPI_COMM_WORLD,    'example.nc',
. NF_CLOBBER,MPI_INFO_NULL,nfd)
```

```
ier=nfmpi_def_dim(nfd,"x",pn,nhand)
ier=nfmpi_def_dim(nfd,"y",pm,mhand)
ier=nfmpi_def_dim(nfd,"z",pl,lhand)
ier=nfmpi_def_dim(nfd,"t",pt,thand)
 hdimarr(1)=nhand
 hdimarr(2)=mhand
 hdimarr(3)=lhand
 hdimarr(4)=thand
```

```
     ier=nfmpi_def_var(nfd,"x"
,nfp,podim,nhand,xhand)
     ier=nfmpi_def_var
    (nfd,"w",nfp,pndim,hdimarr,whand)
     ier=nfmpi_enddef(nfd)
     ier=
nfmpi_put_vara_real_all(nfd,xhand,pstart,pn,x)
```

istart(1)=(npos-1)*np+1
istart(2)=(mpos-1)*mp+1

```
istart(3)=1
istart(4)=iframe
```

icnt(1)=np
icnt(2)=mp

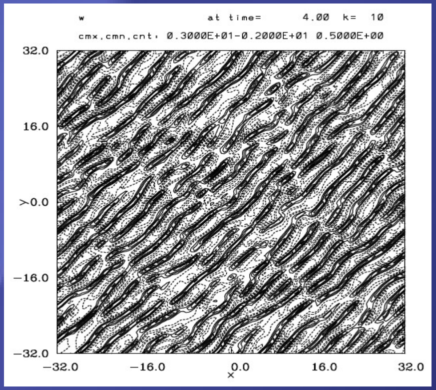
```
icnt(3)=1
icnt(4)=1

call pnettrans(w,data)
ier = nfmpi_put_vara_real_all(nfd,whand,
. istart,icnt,data)
 ier = nfmpi_close(nfd)
```

Serial versus Parallel NETCDF

- OPEN and CREATE calls include MPI constants
- Function calls start with nf_ versus nfmpi_
- Parallel NETCDF extremely sensitive to the data types of functions arguments – needs careful coding
- Data written in chunks in Parallel Netcdf but result is identical to serial write
- Admins don't like locking frontends with postproc.
- **Two times faster than the parallel Fortran tape ! (*BlueGene 256 proc.*)**

Tau parallel profiler

NCAR Graphics



- Traditional tool for EULAG visualization
- “Manual” modification of output impossible
- Hard to learn for beginners and hard to modify by experts
- Master code standarization and comments absolutely necessary
- Further separation of analysis from plotting with the use of NETCDF and NCL ?

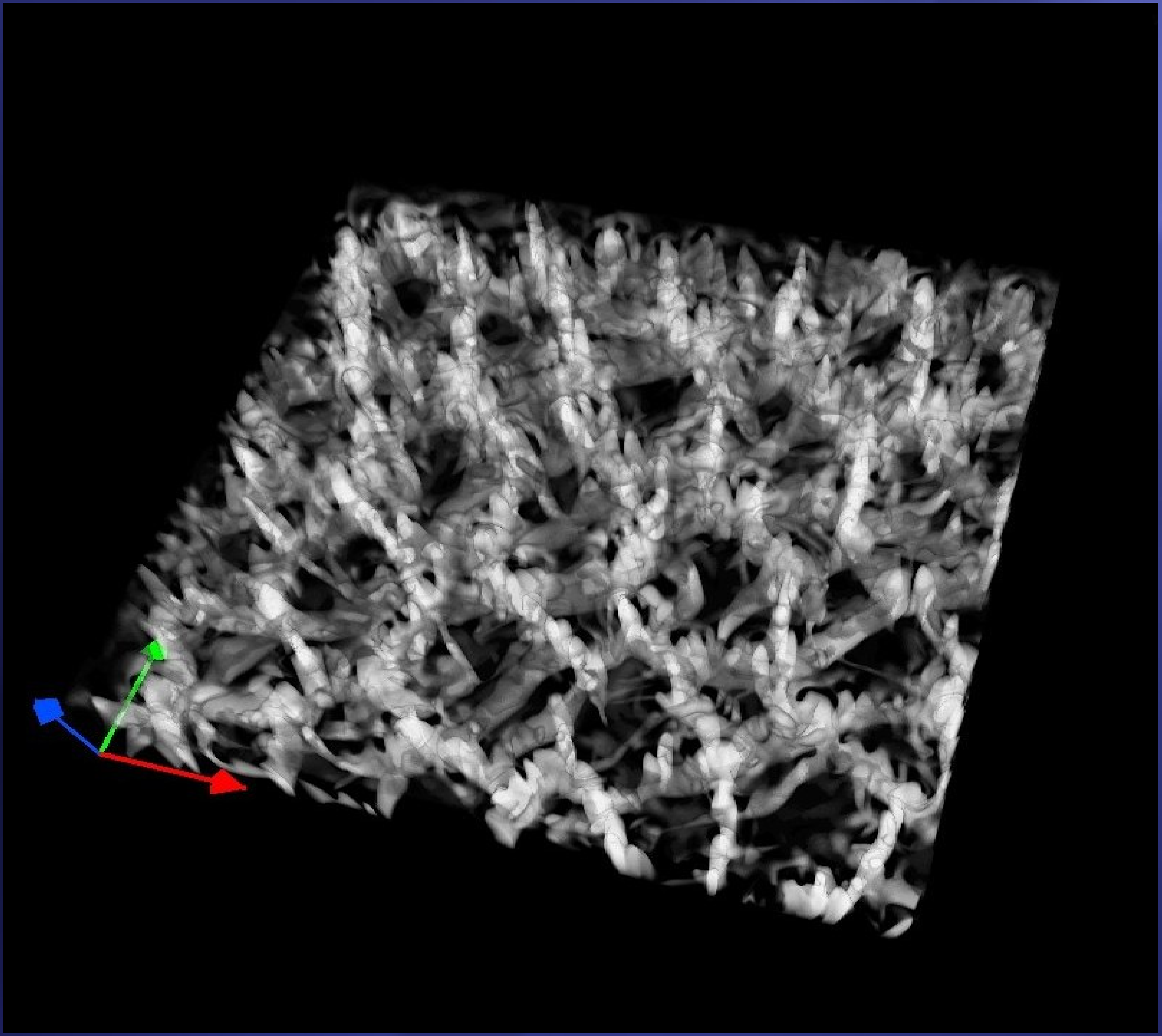
Vapor:
Visualization and Analysis Platform

- Desktop visualization and analysis of tera-scale sized data sets
- Direct Volume Rendering of variables
- Isosurfaces, crossections and probes
- Streamlines, trajectories
- IDL interface
- Simple conversion from raw data or NETCDF file

NETCDF TO VAPOR CONVERSION

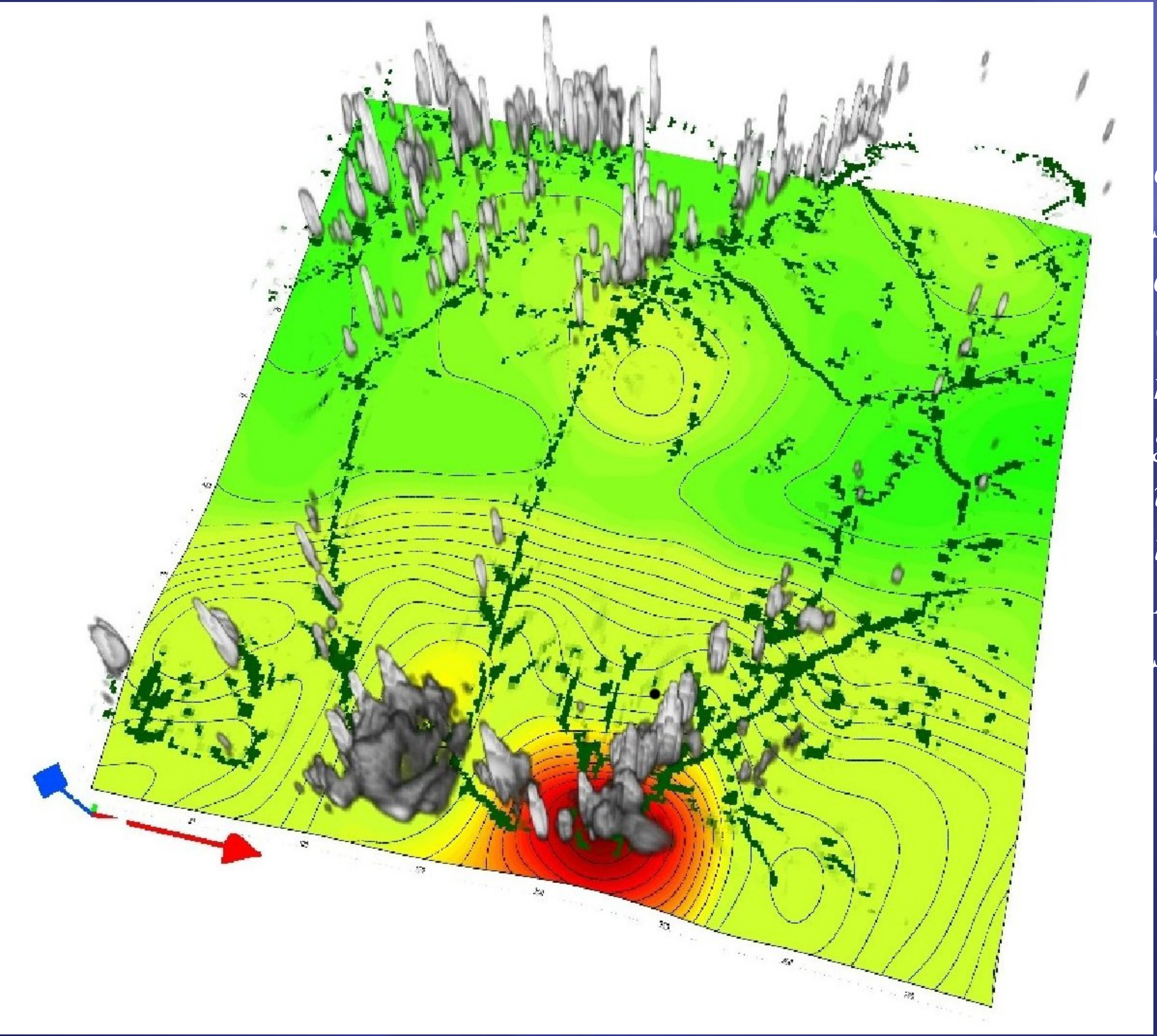
```
source /PATH-TO_VAPOR/vapor-setup.csh
setenv PATH /PATH-TO_VAPOR/bin:$PATH
vdfcreate -dimensio 256x256x128
-numts 13 -level 3 -extents 0:0:0:6000:6000:6000
-varnames w:ELEVATION -gridtype layered Polska.vdf
@ n = 0
while ($n < 13)
ncdf2vdf
-ts $n -varname u -dimnames x:y:z -cnstname t -cnstvals $n
Polska.vdf  ./Polska.nc
        ncdf2vdf -ts $n -varname w -dimnames x:y:z -cnstname t
-cnstvals $n Polska.vdf ./Polska.nc
        ncdf2vdf -ts $n -varname ELEVATION -dimnames x:y:z
-cnstname t -cnstvals $n Polska.vdf ./Polska.nc
@ n += 1
end
```

Example: idealized flow



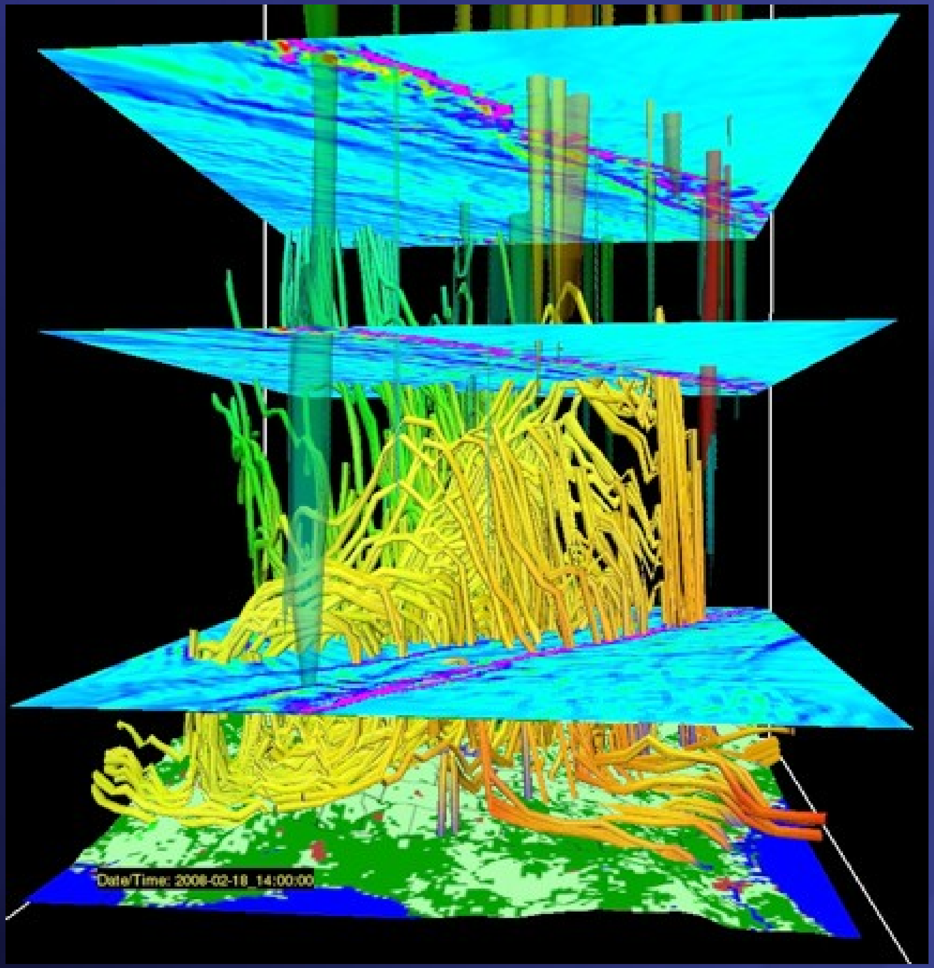
Structure of thermal convection over heated plate. Vertical velocities after 6h of simulated time are shown within the PBL depth. Bright and dark volumes denote updrafts and downdrafts, respectively.

Examples: realistic flow

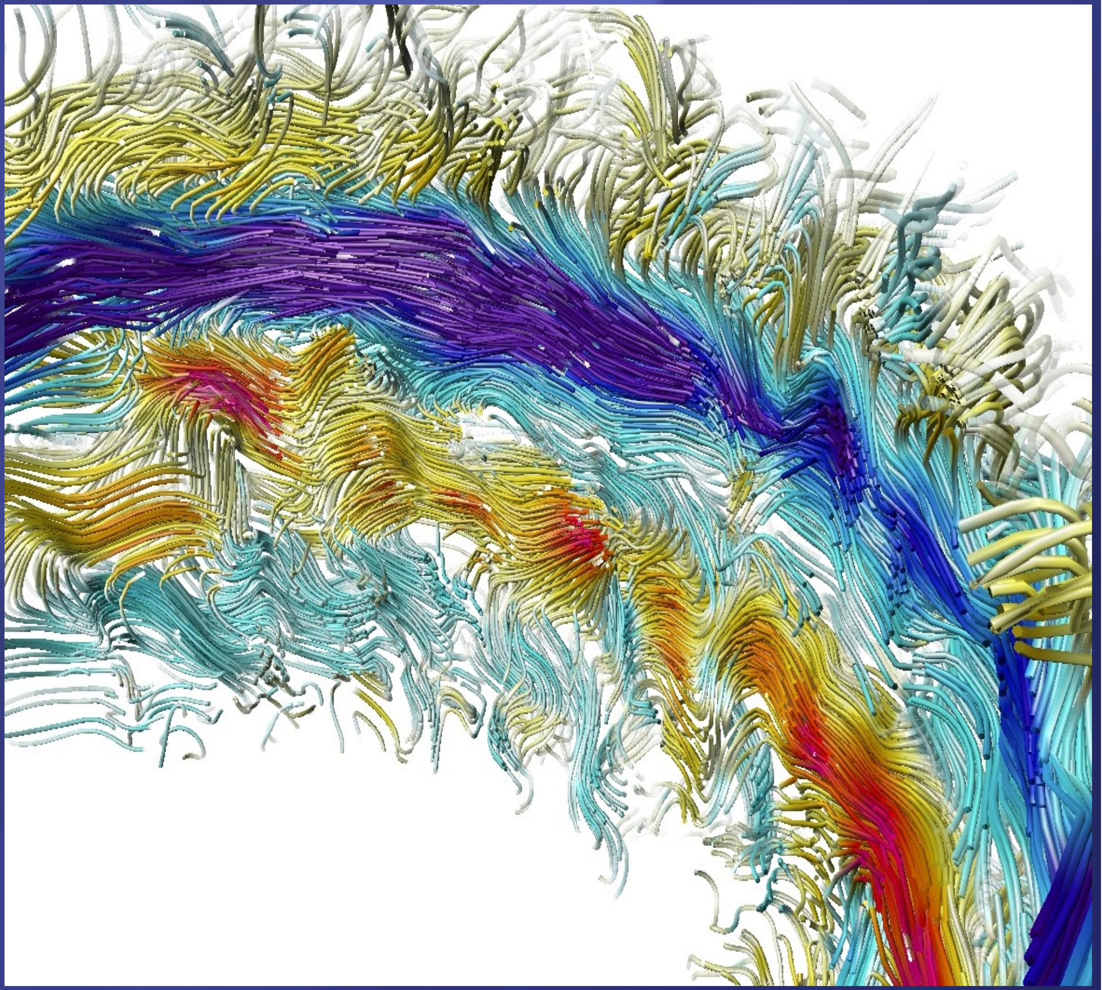


Vertical velocities after 6h of simulated time are shown within the PBL depth.
Grey iso-surfaces represent clouds, and dark green patterns mark updrafts at boundary layer top.
Isolines and other colors show the topography.

Examples from www.vapor.ucar.edu :



Passage of a cold front on 18th February 2008 over Georgia. The convergence of moisture (colored flow lines with red corresponding to high moisture content and green color with low moisture content) along the front is presented using three dimensional vorticity. Thara Prabhakaran, University of Georgia



Magnetic field lines from stellar convection simulations on a sphere. Image courtesy of Ben Brown, University of Colorado.

Special thanks to NCAR for
supporting computations
and
MetStrom for workshop support