CDFTOOLS

http://servforge.legi.grenoble-inp.fr/projects/CDFTOOLS
http://www-meom.hmg.inpg.fr/CDFTOOLS/cdftools-2.1.html

Xianmin Hu
University of Alberta

CDFTOOLS on knossos

- path
  - local server: /mnt/storage0/xhu/PROGRAM/CDFTOOLS/
  - jasper: /home/xianmin/CDFTOOLS
- based on version 2.1
- CDFTOOLS 3.0 has many differences (coding mainly)

cdftransportizFinalSP

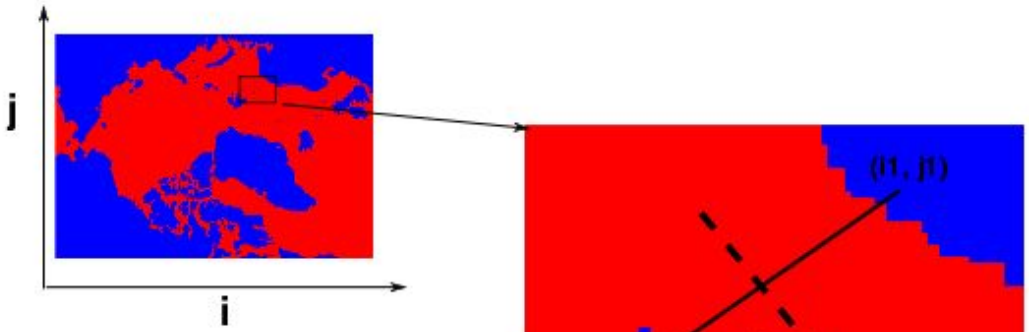
- compute volume, salt, and heat flux through a given section
- required files:
  - mask.nc (3d tmask, umask and vmask)
  - mesh_hgr.nc (horizontal mesh information: coordinates.nc)
  - mesh_zgr.nc: (contains vertical coordinate information: mbathy, e3t_ps, e3w_ps, gdpet_0, gdeptw_0, e3t_0, e3t_w, ...)
  - gridT-file, gridU-file and gridV-file
  - section.dat

CLIPPED VERSION

- cdftransportizClip
  - set the sub-domain for each section
  - use very small memory but may be slower if too many sections within a relative small area
- cdftransportizClipII
  - set the sub-domain based on all sections
  - relatively less memory usage and cpu cost

section.dat

section1-name
i0,i1,j0,j1
section2-name
i0,i1,j0,j1
….
EOF



if net volume flux > 0
the net flow goes from left to right
net volume flux < 0
the net flow goes from right to left

i0 could be smaller than i1
j0 could be smaller than j1

example:

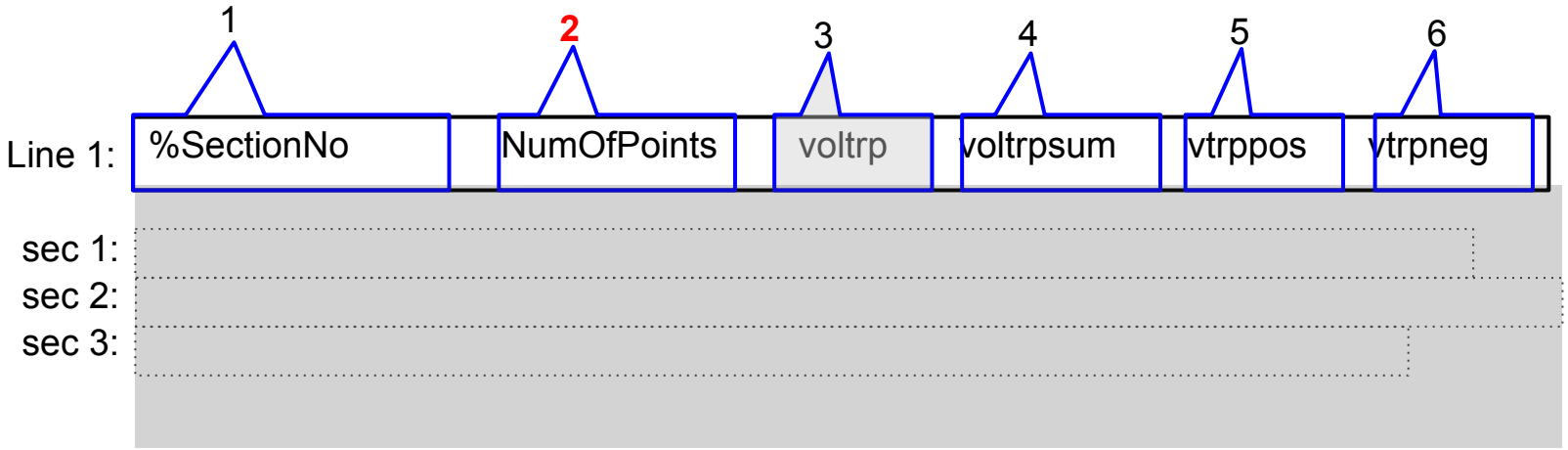
```
gridT file: NAA-EXH18_y1970m01d05_gridT.nc     # temperature, salinity
gridU file: NAA-EXH18_y1970m01d05_gridU.nc   # zonal (i-) velocity
gridV file: NAA-EXH18_y1970m01d05_gridV.nc    # meridinal (j-) velocity

cdftransportizFinalSP -short NAA-EXH18_y1970m01d05 < section.dat
```

cdftransportizFinalSP Output

- screen output (could be turned off by “-Q”)
- text files
  - section_trp.dat (section information and summaries)
  - vtrp.txt: volume flux
  - htrp.txt: heat flux
  - strp.txt: salt flux
- batch script example

  /mnt/storage1/xhu/nemo_script/GetSectionOceanFlux_ANHA.sh
  /lustre/home/xianmin/CREG012/CREG012-EXH003-S/PBSJobGetSectionVolIASP.sh

vtrp.txt (strp.txt, htrp.txt)



1. sectionNo (based on the order in section.dat)
2. number of model points along the section (used for calculation), **field 2**
3. vertically integrated volume transport on each model point along the section (**length is given by field 2**)
4. the net volume transport
5. the positive component
6. the negative component

to combine the output, see this slide

Not all lines are of same length ==> not ready for reading in matlab

cdficetrpFinal

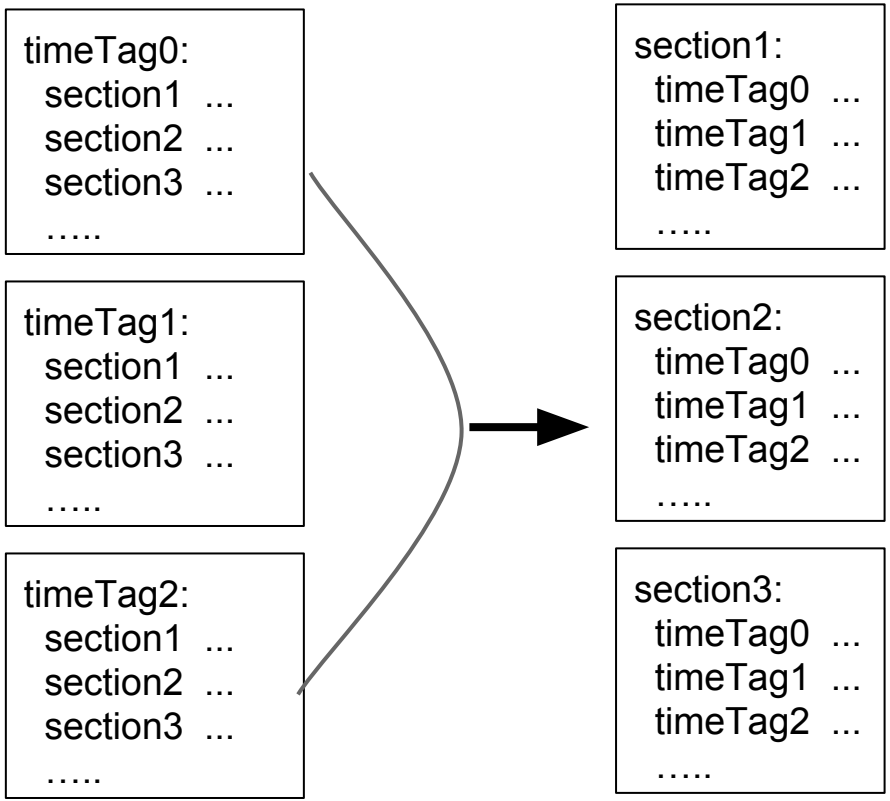
- compute ice flux through a given section
- required files:
  - mesh_hgr.nc, icemod file, section.dat
- output
  - section_trp.dat and ice_trp.txt
  - ice_trp.txt contains only **FOUR** fields:

    sectionNo netFlux positiveFlux negativeFlux

- batch script example

  /mnt/storage1/xhu/nemo_script/GetSectionIceFlux_ANHA.sh

Combine all trp records of a specific section



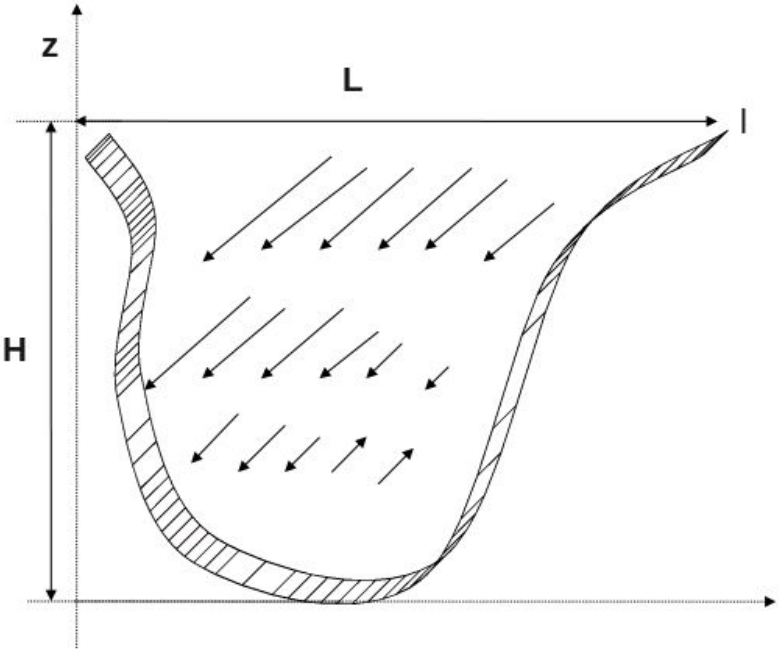
/mnt/storage1/xhu/nemo_script/combineSectionVHSIFlux.sh

Freshwater Flux

$$FW(t) = \int_{l_0}^{l_1} \int_{-H}^{0} \left(\frac{S_{ref} - S(t,z,l)}{S_{ref}} * V_{normal}(t,z,l)\right) dz\, dl$$

$$= \int_{l_0}^{l_1} \int_{-H}^{0} V_{normal}(t,z,l)\, dz\, dl - \frac{1}{S_{ref}} \int_{l_0}^{l_1} \int_{-H}^{0} S(t,z,l) * V_{normal}(t,z,l)\, dz\, dl$$

$$= FLUX_{vol} - \frac{FLUX_{salt}}{S_{ref}}$$



- NAA

  knossos: /mnt/storage0/xhu/NAA-I/mesh_mask_arc4.nc (contains everything)

  ln -s /mnt/storage0/xhu/NAA-I/mesh_mask_arc4.nc mask.nc

  ln -s /mnt/storage0/xhu/NAA-I/mesh_mask_arc4.nc mesh_hgr.nc

  ln -s /mnt/storage0/xhu/NAA-I/mesh_mask_arc4.nc mesh_zgr.nc

  section.data example:

  /mnt/storage0/xhu/NAA-EXH18-S/section_final.dat

- ANHA4

  jasper: /home/xianmin/ANHA4-I/mesh_mask_anha4.nc (contains mask.nc and mesh_zgr.nc)

  ln -s mesh_mask_anha4.nc mask.nc

  ln -s mesh_mask_anha4.nc mesh_zgr.nc

  ln -s mesh_mask_anha4.nc mesh_hgr.nc

  local server: /mnt/storage1/xhu/ANHA4-I/{ANHA4_mask.nc, ANHA4_mesh_hgr.nc, ANHA4_zgr.nc}

- ANHA12

  jasper: /lustre/home/xianmin/CREG012-I/ (ANHA12-EXH003, old bathymetry)

  /lustre/home/xianmin/ANHA12-I/mesh_zgr_2014.nc (ANHA12-EXH006, new bathymetry)

  local server: /mnt/storage1/xhu/ANHA12-I/

compute the fluxes of a specific water mass through a section

command:

cdftransportizTSmass gridTfile gridUfile gridVfile -tmin min-temp -tmax max-temp &
-smin min-salinity -smax max-salinity

water mass:

tmin<t<=tmax, smin<s<=smax (default: tmin=-100, tmax=100; smin=0, smax=100)

result:

section_TSmasstrp.dat (contains volume, salt and heat fluxes)

example:

/mnt/storage1/xhu/nemo_script/vol_watermass/getWatermassFlux.sh

example output

```
% Transport along a section by levels: CAA_42 DavisStrait
% No.  IMIN IMAX JMIN JMAX
% No. LONmin LATmin LONmax LATmax
% Top(m)  Bottom(m) flux (m^3/s, kg/s, w/s)  pos-comp   neg-comp
  1   369   393     73     98
  1   -61.82 66.35   -53.13     66.27
      0.00  5750.00          -303992.21      563066.92     -867059.14   volume
      0.00  5750.00        -10405899.45    19456183.82   -29862083.27   salt
      0.00  5750.00  -4621594089088.00  6149212849536.00  -10770806938624.00   heat
```

NET

due to positive vol_flux

due to negative vol_flux

compute sigmai using cdfsigi

command:

cdfsigi t-file ref_depth

t-file: the gridT file from NEMO output,

providing vosaline (salinity) and votemper (temperature)

ref_depth: in meter

output: a similar nc files with vosigmai (kg m^{-3})

bash scripts:

for a single file:

GetSigi.sh CaseTagWithTimeTag gridT ref_depth

batch mode:

JobGetSigi.sh

compute potential density using cdfsig0

command:

cdfsig0 t-file

t-file: the gridT file from NEMO output,
providing vosaline and votemper

output: sig0.nc with vosigma0 (kg/m^3)

upgraded version:

salinity in a different file (e.g., GLORYS2):

cdfsig0 t-file s-file

compute sigma0 within a sub-domain:

cdfsig0 t-file imin imax jmin jmax

or

cdfsig0 t-file s-file imin imax jmin jmax

project quantities onto isopycnals

command:

cdfrhoproj var-name rho-file var-file [var-grid-type]

var-name: the variable name of quantity in var-file

rho-file: the file contains rho [computed with cdfsig0)

var-file: the file contains var-name

var-grid-type: T (default), U, V

* isopycnal levels are defined in a file named **rho_lev** in current directory

Output (var-file.interp):

* on T-point

* ***varname*****:** interpolated variable

vodepiso: depths of isopycnals

rho_lev example:

3 ← number of isopycnals

25.5

26.5

27.5

integrate quantities between isopycnals

command:

cdfsigintegr var-name rho-file var-file [var-grid-type]

var-name: the variable name of quantity in var-file

rho-file: the file contains rho [computed with cdfsig0)

var-file: the file contains var-name

var-grid-type: T (default), U, V

* isopycnal levels are defined in a file named rho_lev in current directory

Output (var-file.integr):

* on T-point
* **inv*varname*:** integrated variable

isothick: isopycnal thickness

vodepiso: depths of isopycnals

rho_lev example:

3 ← number of isopycnals

25.5

26.5

27.5

maximum salinity between isopycnals

command:

cdfsigMaxS rho-file s-file rho0 rho1

rho-file: the file contains rho [computed with cdfsig0)

s-file: the file contains salinity, *vosaline*

rho0 rho1: defined the isopycnal layer

Output (s-file.maxS):

* on T-point

* **maxvoslaine:** maximum salinity

  isothick: isopycnal thickness

  vodepiso: depths of isopycnals

Note: if rho1 interface is not found but rho0 exists, maximum salinity is based on salinity from rho0 to bottom)

compute volume transport for each density class across a section

command:

cdfsigtrp tfile ufile vfile sigma_min sigma_max nbins [options]

tfile: gridT file contains the 3d temperature and salinity

ufile: gridU file contains the 3d zonal velocity

vfile: gridV file contains the 3d meridional velocity

nbins: number of bins between the given sigma range

options: -print ⇒ extra information on screen; *-file ⇒ extra information in cdfsigtrp_print*

* need mesh_hgr.nc and mesh_zgr.nc in current directory
* section information is defined in dens_section.dat

output: trpsig.txt:

* 1st column: sigma
* 2nd column: transports ($\mathbf{m^3s^{-1}}$) through 1st section
* 3rd column: transports ($\mathbf{m^3s^{-1}}$)through 2nd section

…...

dens_section.dat

ZonalAtlanticN ← section name

178,376,305,305 ← imin, imax,jmin,jmax

ZonalAtlanticS

72,371,228,228

EOF

heat content within a basin (given depth range)

command:

cdfheatcBasin tfile bsmaskfile dep0 dep1

tfile: the gridT file which contains temperature field, votemper

bsmaskfile: 2d basin mask file (tmask=1 within the basin)

dep0: depth of the top layer (of interested water mass)

dep1: depth the bottom layer

if dep0=dep1=0
⇒ whole water column

Output (on screen):

heat content in each layer (GJ)

total heat content
[sum(rho0*cp*t*vol)] in GJ

```
MESH_ZGR V3 detected
 Heat Content  at level          1 (   3.0467727      m)   -13664699218.750000     surface =    618303.25000000000      km^2
 Heat Content  at level          2 (   9.4540491      m)   -13946753906.250000     surface =    618303.25000000000      km^2
 Heat Content  at level          3 (   16.363966      m)   -14377369140.625000     surface =    618303.25000000000      km^2
 Heat Content  at level          4 (   23.898710      m)   -15600402343.750000     surface =    617739.00000000000      km^2
 Heat Content  at level          5 (   32.209290      m)   -17586396484.375000     surface =    617176.25000000000      km^2
                                 6 (   41.481853      m)   -20131568359.375000     surface =    615512.68750000000      km^2
                                 7 (   51.945129      m)   -22656150390.625000     surface =    613841.37500000000      km^2
                                 8 (   63.879051      m)   -23998103515.625000     surface =    610938.50000000000      km^2
                                 9 (   77.624512      m)   -23038201171.875000     surface =    607086.31250000000      km^2
                                10 (   93.594124      m)   -16012053710.937500     surface =    601878.12500000000      km^2
                                11 (   112.28349      m)   -3141271972.6562500     surface =    594340.00000000000      km^2
 Heat Content  at level         12 (   134.28227      m)    12555369140.625000     surface =    586355.87500000000      km^2
 Heat Content  at level         13 (   160.28400      m)    32574658203.125000     surface =    574088.12500000000      km^2
 Heat Content  at level         14 (   191.09251      m)    56923863281.250000     surface =    554927.25000000000      km^2
 Heat Content  at level         15 (   227.62332      m)    84476226562.500000     surface =    530785.56250000000      km^2
 Heat Content  at level         16 (   270.89621      m)   -51300707031.250000     surface =    502671.37500000000      km^2
 Total Heat content :   -48923560058.593750      GJ (1 GJ = 1e9 Joules)
 Total Heat content/volume :   -367428.45916449401      Joules/m3
```

Heat and salt contents in the mixed layer

command:

cdfmxlbasinST tfile bsmaskfile [rho0]

tfile: the gridT file which contains mixed layer depth, temperature, salinity fields, votemper

bsmaskfile: 2d basin mask file (tmask=1 within the basin)

Output (on screen):

```
npiglo=         568
npjglo=         400
npk    =          46
MESH_ZGR V3 detected
Mixed Layer ST:  avgH (m)  vol (km^3)   HeatContent (TJ)   SaltContent (kt)
                  20.283012403104262       4429.9241038331866       -3629017.9905816060       141233026.50000000
```

averaged MLD (m)

volume (km^3)

total heat content [sum(rho0*cp*t*vol)] in TJ

total salt content [sum(s*vol)] in kt

Heat and salt contents in mixed layer

command:

cdfmxlbasinST

extra options:

-d: debug mode
-rho rho: density used in heat content calculation, default is 1000 kg m^{-3}
-bsMaskVarName varName: mask variable name in the basin mask file
-region imin imax jmin jmax: set the sub dataset region to reduce memory usage

-tref : reference temperature, default is zero
-sref: reference salinity (for freshwater content),
if negative values, only relatively fresher water mass is considered

Variable vertical-averaging

command:

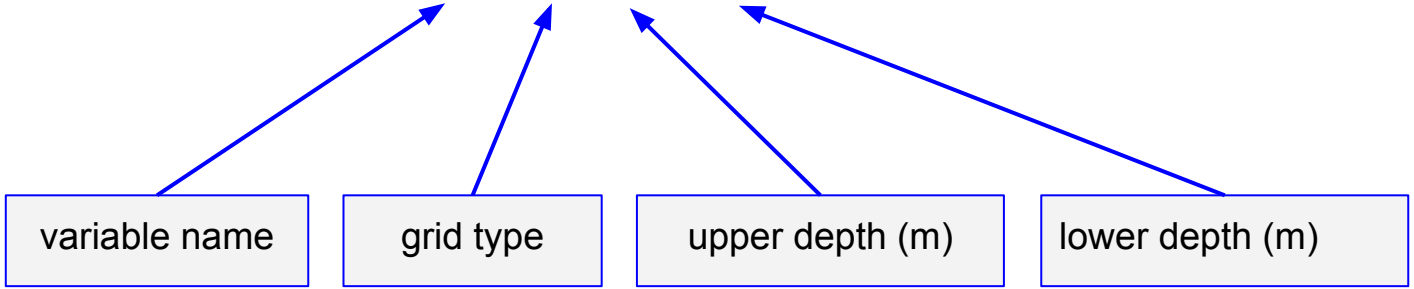
cdfvertmean datafile varname T|U|V|W z1 z2

T|U|V|W: grid-type of the variable

output: vertmean.nc (variable name: **sovertmean**)

example (top-55m salinity):

```
timeTag="y2002m01d05"
tfile="ANHA4-EXH001_${timeTag}_gridT.nc"
cdfvertmean    ${tfile}    vosaline    T    0    55   && mv vertmean.nc ncfile/ANHA4-EXH001_${timeTag}_S_0_55.nc
```



find closest i, j of given location (lon,lat)

command:

cdffindij start-lon end-lon start-lat end-lat [coordinate-file] [point-type]

point-type: T|U|V|F

example:

```
[xhu@knossos ANHA4-I]$ cdffindij -90 -90 75 75 ANHA4_coordinates.nc T
#  rdis=    5.129 km
#  rdis=    5.129 km
       168       168       552       552
  -89.8503  -89.8503   74.9750   74.9750
```

input order matters!

find the lon and lat of given location (i,j)

command:

cdfwhereij start-i end-i start-j end-j [coordinate-file] [point-type]

point-type: T|U|V|F

example:

```
[xhu@knossos ANHA4-I]$ cdfwhereij  168 168 552 552 ANHA4_coordinates.nc T
 Type of point   : T
   I J zoom      :    168   168   552   552
   LON LAT zoom  :   -89.850  -89.850   74.975   74.975
```

compute the linear trend of input files

command:

cdflinreg input-files

```
tfileStr=""
YS=2002; YE=2010;
CFEXP="ANHA4-TEST7"
for cYear in `seq ${YS} ${YE}`
do
        tfileStr="${tfileStr} ${CFEXP}_y${cYear}_annual_gridT.nc"
done
eval "cdflinreg ${tfileStr} && mv linreg.nc ts_linreg_${YS}_${YE}.nc"
```

remove the linear trend at each point

```
[ -e linreg_detrend_0000.nc ] && rm -f linreg_detrend_0000.nc
YS=2003
YE=2010
CFEXP="ANHA4-TEST7"
tfile0=${CFEXP}_y2002_annual_gridT.nc
linregfile="ts_linreg_2002_2010.nc"

for cYear in `seq ${YS} ${YE}`
do
   tfile=${CFEXP}_y${cYear}_annual_gridT.nc
   savefile="${CFEXP}_y${cYear}_annual_detrendS.nc"
   if [ -e ${savefile} ]; then
      echo "${savefile} exists. SKIP year: ${cYear}"
   else
     cdfdetrend vosaline ${linregfile} ${tfile0} ${tfile}
     if [ -e linreg_detrend_0000.nc ]; then
        tmpncfile="mydeflated_`date +%Y%m%d%H%M%S`$$.nc"
        nccopy -s -d 9 linreg_detrend_0000.nc ${tmpncfile} && mv ${tmpncfile} ${savefile} && rm -f linreg_detrend_0000.nc
    else
        echo "linreg_detrend_0000.nc is not produced! Must be something wrong!"
        exit
     fi
   fi
done
```

momentum terms computation

continue….