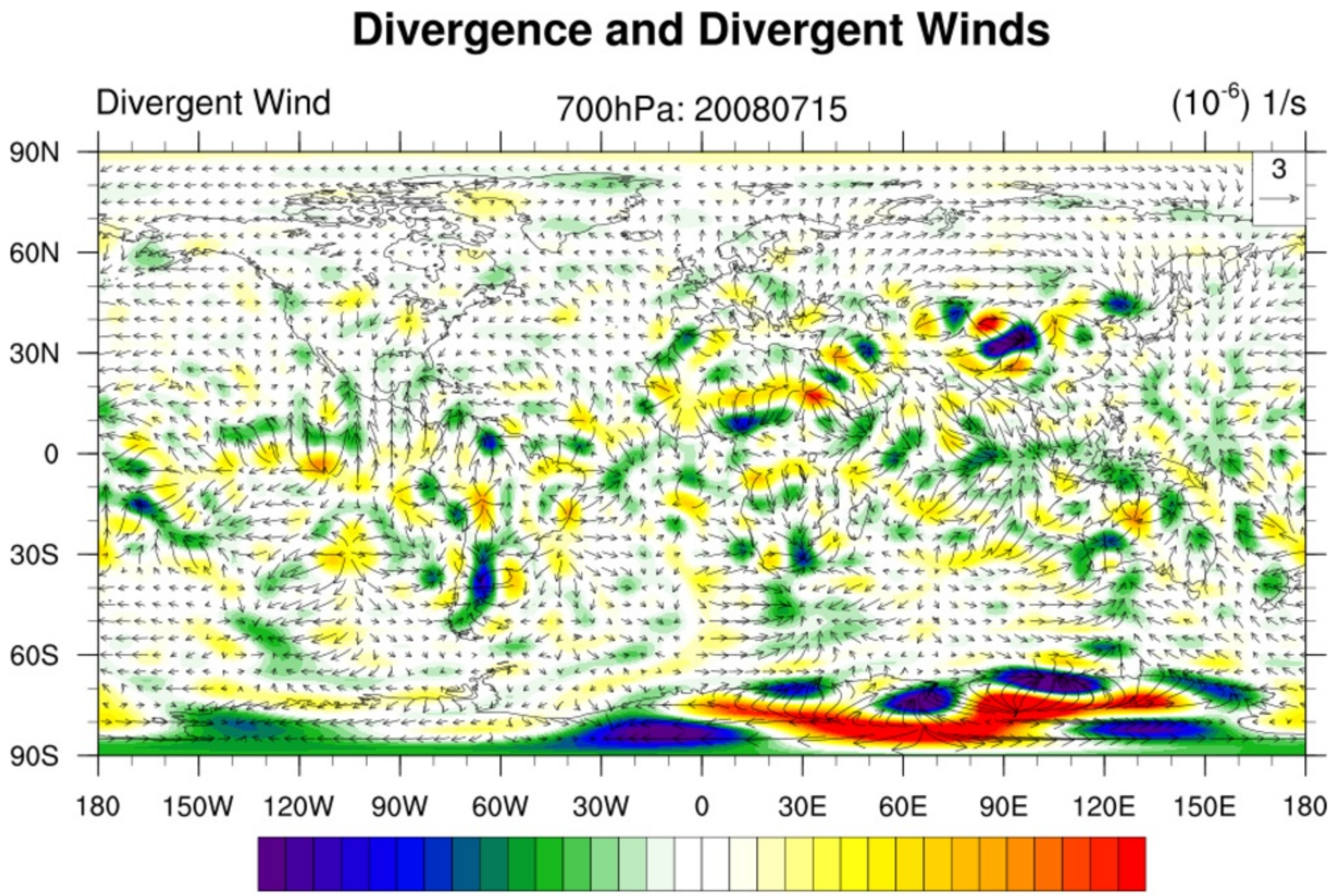
水平输送水汽通量matlab,NCL专辑水汽通量散度以及垂直积分可视化

mfc_div_1.ncl: Calculate various divergence and moisture quantities including Vertically Integrated Moisture Flux Convergence (VIMFC). VIMFC has a high correlation with frontal and convective activity. Positive values indicate net precipitation. The following equation is implemented within mfc_div_1.ncl

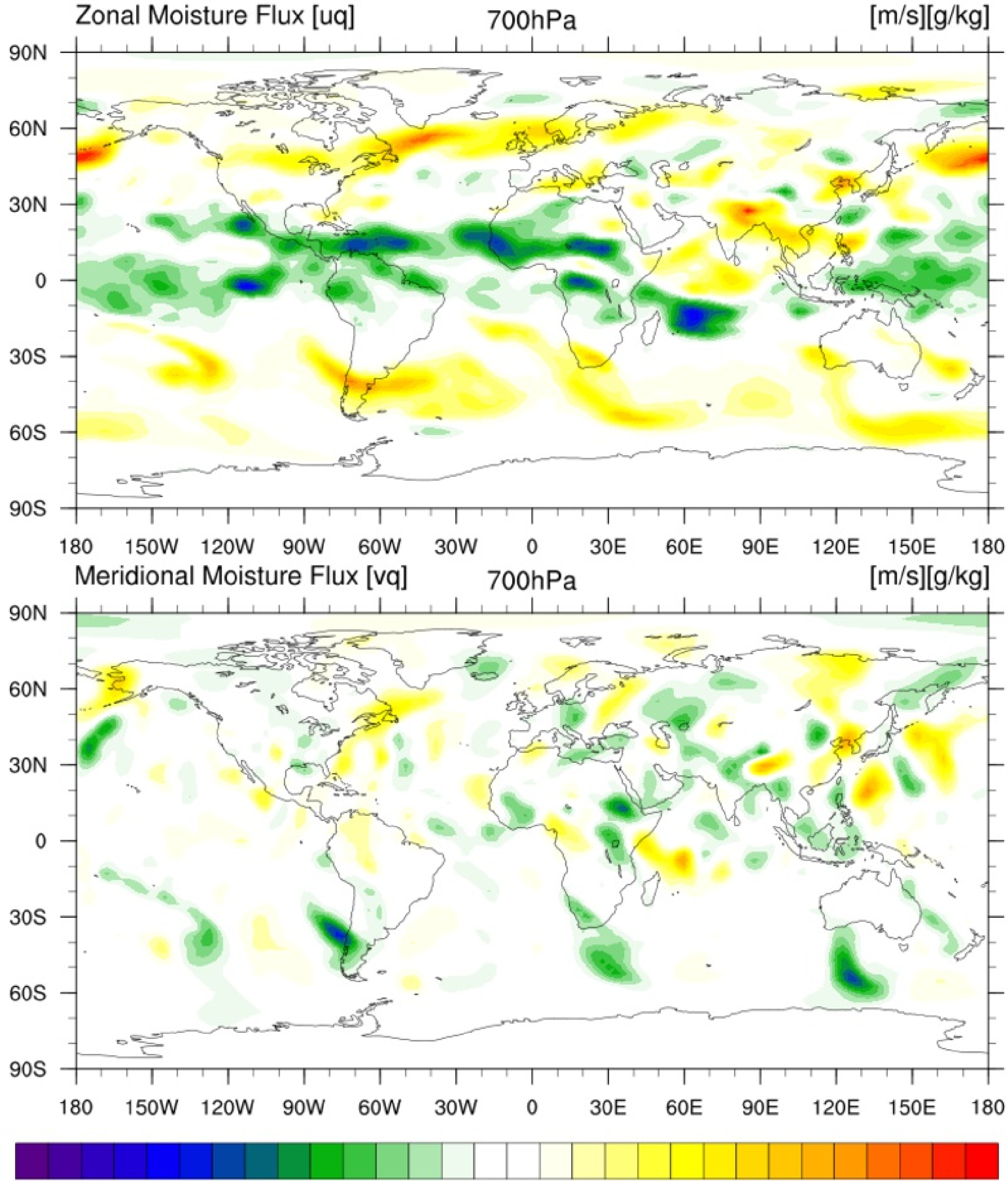
$$\mathrm{MFC} = -\int_0^{p_s} \nabla_p \cdot (\mathbf{u}\, q)\frac{\mathrm{d}p}{g} = P - E + \frac{\partial w}{\partial t} \qquad (1)$$

where MFC is the horizontal moisture flux convergence integrated from the surface to the top of the atmosphere, $\nabla_p \cdot ()$ the horizontal divergence in pressure coordinates, $\mathbf{u} = (u, v)$ the horizontal wind vector, q the specific humidity, P the precipitation rate, E the evaporation rate, and $w = \int_0^{p_s} q\frac{\mathrm{d}p}{g}$ the total precipitable water. With negligible storage $\partial w/\partial t$ (e.g., Figure 1a), the dominant balance in the SASM region is between MFC and net precipitation (P-E). Thus, positive (negative) values of MFC correspond to positive (negative) net

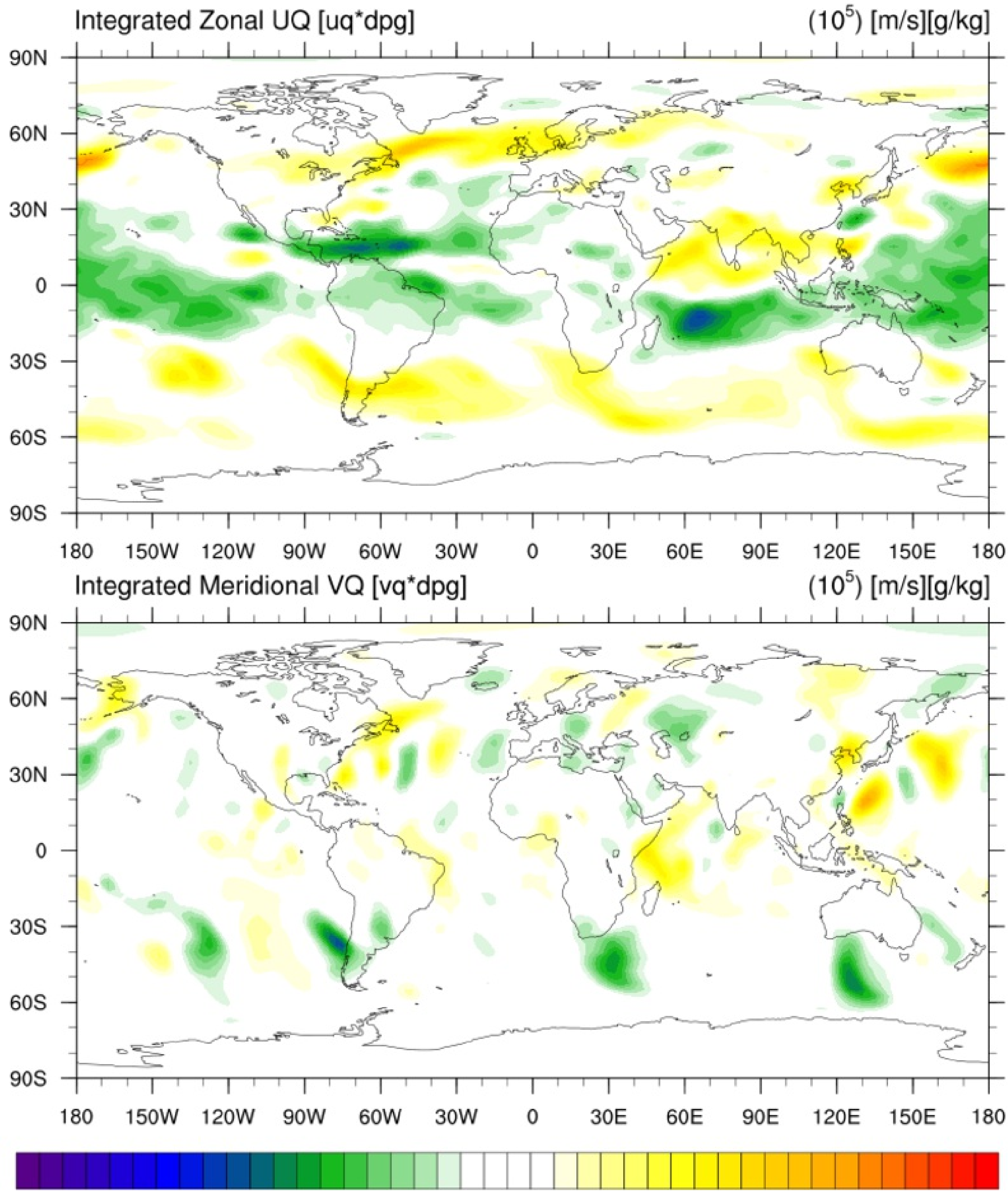
This example uses uv2dvF_Wrap [uv2dvF] because the grid is a global fixed grid. For global gaussian, uv2dvG_Wrap [uv2dvG] should be used. For a regional grid uv2dv_cfd should be used.



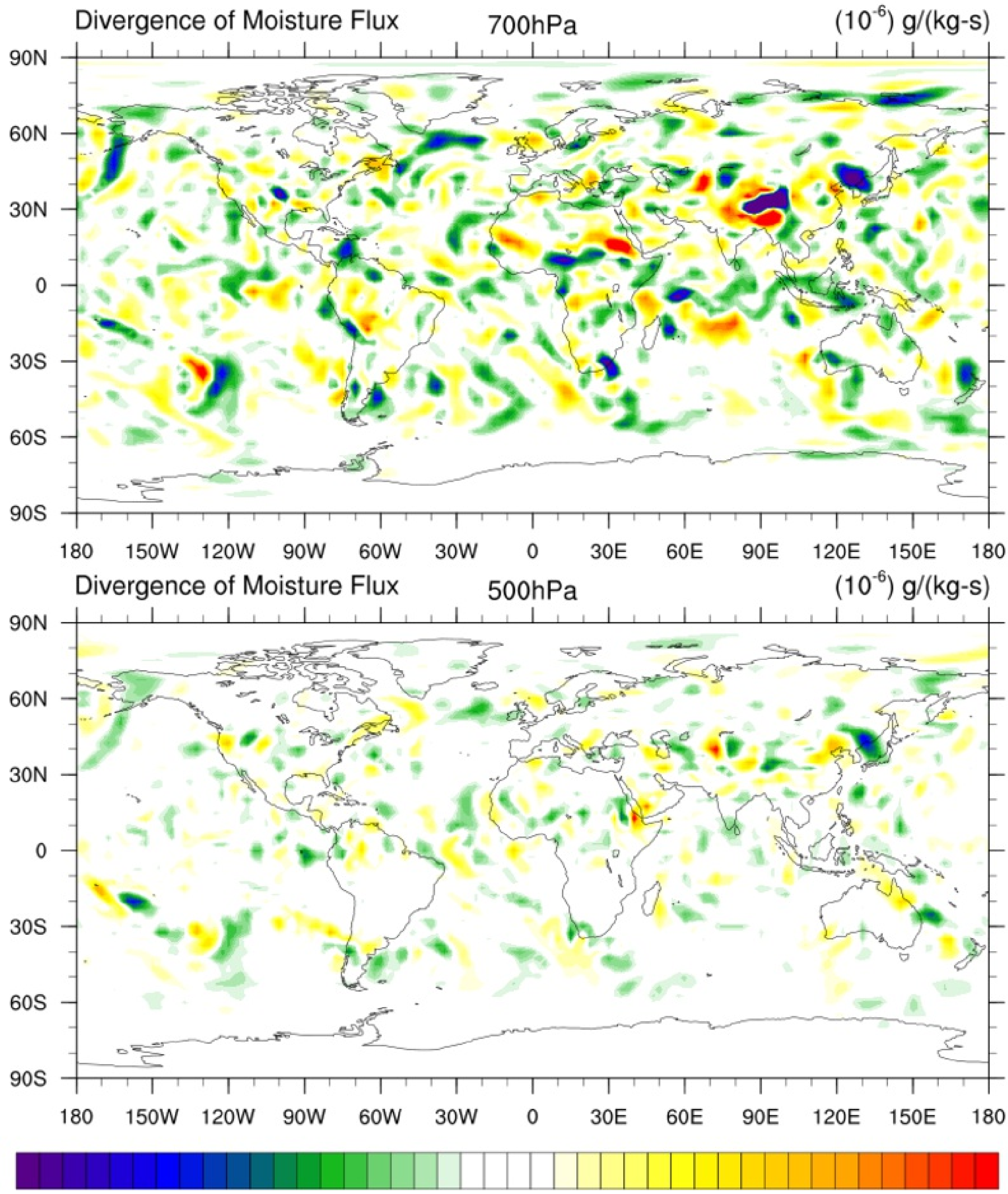
20080715: Unweighted Moisture Flux Components

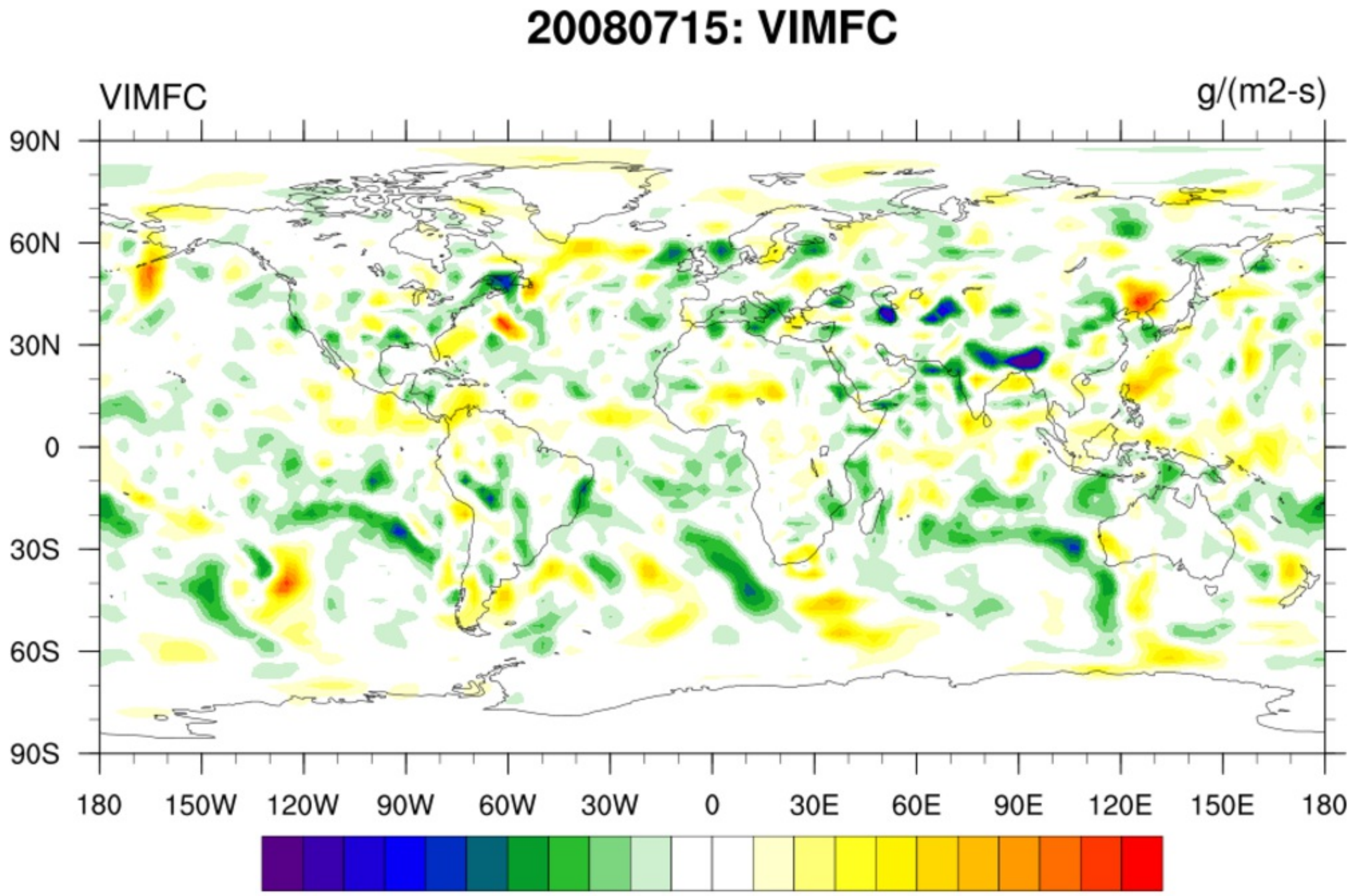


20080715: Mass Wgt. Component Moisture Flux



20080715: Divergence of Moisture Flux





mfc_div_1.ncl

```
;*************************************************
; mfc_div_1.ncl
;
; Concepts illustrated:
;   - Read daily mean wind components, humidity and sfc. pressure
;     from different files
;   - Reorder the input (N==>S) grid order to (S==>N) via NCL syntax  ::-1
;   - Calculate mass weightined layer thickness [units="kg/m2"]
;   - Calculate moisture flux [uq, vq]
;   - Calculate moisture flux divergence using spherical harmonics
;   - Integrate the moisture flux divergence using mass weighting
;   - Plot a number of quantities
;*************************************************
;---Calculate the Horizontal Moisture Flux Convergence [MFC]
;*************************************************
;---High frequency source data: hourly/3hr/6hr/12hr/daily .... NOT monthly values
;---References:
```

```
;=========================================================================
;   Data Source: ESRL Physical Sciences Division
;   NCEP Reanalysis data provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA,
;=========================================================================
ptop = 300              ; 'shum' upper level
ptop@units = "hPa"
g    = 9.80665          ; m/s2
date  = 20080715        ; NH summer
;---ESRL: CDC data
diri = "./"
filq = "shum.2008.nc"   ; daily data for current year [366 days]
filu = "uwnd.2008.nc"
filv = "vwnd.2008.nc"
filps= "pres.sfc.2008.nc"
pthu = diri+filu
pthv = diri+filv
pthq = diri+filq
pthps= diri+filps
fu   = addfile(pthu ,"r")
fv   = addfile(pthv ,"r")
fq   = addfile(pthq ,"r")
fps  = addfile(pthps,"r")
;---Time
ymd  = cd_calendar(fu->time, -2)    ; ymd[*]: human readable
nt   = ind(ymd.eq.date)             ; date for plotting and testing
TEST = True
if (.not.TEST) then                 ; all times
u    = fu->uwnd(:,{1000:ptop},:,:)  ; m/s, (time,level,lat,lon)
v    = fv->vwnd(:,{1000:ptop},:,:)
q    = fq->shum                     ; [kg/kg], 1000-300 levels only
ps   = fps->pres                    ; Pa=>[kg/(m-s2)], (time,lat,lon)
else                                ; one time step; keep time dimension [ nt:nt: ]
u    = fu->uwnd(nt:nt,{1000:ptop},:,:); m/s, (time,level,lat,lon)
v    = fv->vwnd(nt:nt,{1000:ptop},:,:)
```

```
q    = fq->shum(nt:nt,:,:,:)         ; [kg/kg], 1000-300 levels only

ps   = fps->pres(nt:nt,:,:)          ; Pa=>[kg/(m-s2)], (time,lat,lon)

nt   = 0                             ; only one time step

end if

;---Vertical levels

ptop = ptop*100

ptop@units = "Pa"

plev = q&level                       ; hPa

plev = plev*100                      ; [100000,...,30000] Pa [kg/(m-s2)]

plev@units = "Pa"

;---Change [kg/kg] to [g/kg]; not necessary: but common units for q

q    = q*1000

q@units = "g/kg"

;---Divergence function [used later] requires S->N grid order

u    = u(:,:,::-1,:)

v    = v(:,:,::-1,:)

q    = q(:,:,::-1,:)

ps   =ps(:,  ::-1,:)

;---Layer thickness: ; Pa=>[kg/(m-s2)], (time,level,lat,lon)

;---Mass weighting:  (dp/g) => [Pa/(m/s2)] => (Pa-s2)/m => [kg/(m-s2)][s2/m] =>  (kg/m2)

dp   = dpres_plevel_Wrap(plev, ps, ptop, 0) ; Pa; layar thickness

dpg  = dp/g

dpg@long_name = "Layer Mass Weighting"

dpg@units     = "kg/m2"                     ; dp/g, Pa/(m s-2), reduce to kg m-2

;---Moisture flux components at each pressure level

uq   = u*q                                  ; (:,:,:,:)

uq@long_name = "Zonal Moisture Flux [uq]"

uq@units = "["+u@units+"]["+q@units+"]"     ; [m/s][g/kg]

copy_VarCoords(u,uq)                        ; (time,level,lat,lon)

vq   = v*q                                  ; (:,:,:,:)

vq@long_name = "Meridional Moisture Flux [vq]"

vq@units = "["+v@units+"]["+q@units+"]"

copy_VarCoords(v,vq)                        ; (time,level,lat,lon)

PRINT_RAW = True
```

```
if (PRINT_RAW) then
printVarSummary(q)                ; (time,level,lat,lon); g/kg
printMinMax(q,0)
print("-----")
printVarSummary(u)                ; (time,level,lat,lon); m/s
printMinMax(u,0)
print("-----")
printVarSummary(v)
printMinMax(v,0)
print("-----")
printVarSummary(ps)               ; (time,lat,lon); Pa => kg/(m-s2)
printMinMax(ps,0)
print("-----")
printVarSummary(uq)               ; (time,level,lat,lon); (m/s)(g/kg)
printMinMax(uq,0)
print("-----")
printVarSummary(vq)
printMinMax(vq,0)
print("-----")
printVarSummary(dp)               ; (time,level,lat,lon); Pa => kg/(m-s2)
printMinMax(dp,0)
print("-----")
; examine layer thickness at selected locations
print(dp(nt,:,{40},{180}))    ; mid-Pacific
print(dp(nt,:,{40},{255}))    ; Boulder, CO
print("-----")
end if
;---Integrated mass weighted moisture flux components
uq_dpg = uq*dpg          ; mass weighted 'uq'; [m/s][g/kg][kg/m2]=>[m/s][g/kg]
iuq    = dim_sum_n(uq_dpg, 1)
iuq@long_name = "Integrated Zonal UQ [uq*dpg]"
iuq@LONG_NAME = "Sum: Mass Weighted Integrated Zonal Moisture Flux [uq*dpg]"
iuq@units     = "[m/s][g/kg]"
copy_VarCoords(u(:,0,:,:), iuq); (time,lat,lon)
```

```
delete(uq_dpg)

vq_dpg = vq*dpg                 ; mass weighted 'vq'; [m/s][g/kg][kg/m2]=>[m/s][g/kg]

ivq    = dim_sum_n(vq_dpg, 1)

ivq@long_name = "Integrated Meridional VQ [vq*dpg]"

ivq@LONG_NAME = "Sum: Mass Weighted Integrated Meridional Moisture Flux [vq*dpg]"

ivq@units     = "[m/s][g/kg]"

copy_VarCoords(v(:,0,:,:), ivq); (time,lat,lon)

delete(vq_dpg)

;---Divergence of moisture flux: uv2dvF => global 'fixed' rectilinear grid

duvq  = uv2dvF_Wrap(uq, vq)    ; (time,level,lat,lon)

duvq@long_name = "Divergence of Moisture Flux"

duvq@units     = "g/(kg-s)"    ; (1/m)*[(m/s)(g/kg)] => [g/(kg-s)]

;---Mass weighted integration [sum] of the divergence of moisture flux

duvq_dpg = duvq*dpg            ;  [g/(kg-s)][kg/m2] => [g/(m2-s)]

iduvq    = dim_sum_n(duvq_dpg, 1)

iduvq@long_name = "Integrated Mass Wgt MFC"

iduvq@LONG_NAME = "Integrated Mass Weighted Moisture Flux Convergence"

iduvq@units     = "g/(m2-s)"

copy_VarCoords(u(:,0,:,:), iduvq)      ; (time,lat,lon)

delete(duvq_dpg)

VIMFC =  iduvq           ; keep meta data

VIMFC = -VIMFC           ; Note the preceding -1  [negative precedes integration]

VIMFC@long_name = "VIMFC"

;---Another way to compute Integrated divergence of moisture flux [iduvq_1]

;;IDUVQ = wgt_vertical_n(duvq, dp, 2, 1)

;;iduvq_0 = IDUVQ[0]

;;iduvq_0 = iduvq_0/g                   ; complete mass weighting

;;iduvq_0@long_name = "Average Mass Weighted MFC"

;;iduvq_0@LONG_NAME = "Average Mass Weighted Moisture Flux Convergence"

;;iduvq_0@units     = "g/(m2-s)"

;;iduvq_1 = IDUVQ[1]                    ; same as iuvq_sum

;;iduvq_1 = iduvq_1/g

;;iduvq_1@long_name = "Integrated MFC"

;;iduvq_1@LONG_NAME = "Integrated Moisture Flux Convergence"
```

```
;;iduvq_1@units     = "g/(m2-s)"

PRINT_RESULT = True

if (PRINT_RESULT) then

printVarSummary(iuq)                  ; (time,lat,lon)

printMinMax(iuq,0)

print("-----")

printVarSummary(ivq)                  ; (time,lat,lon)

printMinMax(ivq,0)

print("-----")

printVarSummary(duvq)                 ; (time,lev,lat,lon)

printMinMax(duvq,0)

print("-----")

printVarSummary(iduvq)                ; (time,lat,lon)

printMinMax(iduvq,0)

print("-----")

;;  printVarSummary(iduvq_0)             ; (time,lat,lon)

;;  printMinMax(iduvq_0,0)

;;  print("-----")

;;  printVarSummary(iduvq_1)             ; (time,lat,lon)

;;  printMinMax(iduvq_1,0)

;;  print("-----")

end if

;*************************************************

; Calculate divergence: Use Wrap to include meta data

; Calculate divergent wind components; used for graphics

;*************************************************

div = uv2dvF_Wrap(u,v)                 ; u,v ==> divergence; (:,:,:,:)

ud  = new ( dimsizes(u), typeof(u), "No_FillValue")

vd  = new ( dimsizes(v), typeof(v), "No_FillValue")

dv2uvf(div,ud,vd)                      ; divergence ==> divergent components

copy_VarCoords(u, ud )

copy_VarCoords(u, vd )

ud@long_name  = "Zonal Divergent Wind"

ud@units      = u@units
```

```
vd@long_name  = "Meridional Divergent Wind"
vd@units      = v@units
if (PRINT_RESULT) then
printVarSummary(ud)                 ; (time,level,lat,lon)
printMinMax(ud,0)
print("-----")
printVarSummary(vd)                 ; (time,level,lat,lon)
printMinMax(vd,0)
print("-----")
end if
;***********************************************
; plot results
;***********************************************
scl5  = 1e5                                ; arbitrary: used for nicer plot values
sclab5= "(10~S~-5~N~)"                     ; used later
SCLAB5= "(10~S~5~N~)"
scl6  = 1e6
sclab6= "(10~S~-6~N~)"
SCLAB6= "(10~S~6~N~)"
plot := new(2,graphic)
wks   = gsn_open_wks("png","mfc_div")        ; send graphics to PNG file
resd                 = True
resd@cnFillOn        = True                  ; color
resd@cnLinesOn       = False                 ; turn off contour lines
resd@cnLevelSelectionMode = "ManualLevels"   ; set manual contour levels
resd@cnMinLevelValF  = -15.                  ; set min contour level
resd@cnMaxLevelValF  =  15.                  ; set max contour level
resd@cnLevelSpacingF =   1.                  ; set contour spacing
;resd@cnFillPalette   = "cmocean_balance"     ; NCL 6.5.0
resd@cnFillPalette   = "ViBlGrWhYeOrRe"
resd@mpFillOn        = False                 ; turn off map fill
resd@vcRefMagnitudeF = 3.                    ; make vectors larger
resd@vcRefLengthF    = 0.025                 ; reference vector length
resd@vcGlyphStyle    = "CurlyVector"         ; turn on curly vectors
```

```
resd@vcMinDistanceF  = 0.010                ; thin the vectors

resd@vcRefAnnoOrthogonalPosF = -1.0         ; move ref vector up

resd@gsnLeftString   = "Divergent Wind"

resd@gsnScalarContour= True                 ; vectors over contours

LEVP = 700

DIV  = div(nt,{LEVP},:,:)                   ; keep meta data

DIV  = DIV*scl6                             ; nicer numbers

resd@tiMainString    = "Divergence and Divergent Winds"

resd@gsnCenterString = LEVP+"hPa: "+date

resd@gsnRightString  = sclab6+" "+div@units

dplt = gsn_csm_vector_scalar_map(wks,ud(nt,{LEVP},:,:),vd(nt,{LEVP},:,:),DIV,resd)

;--- Moisture Transport [uq, vq] at a specified pressure level

res                  = True             ; plot mods desired

res@gsnDraw          = False            ; don't draw yet

res@gsnFrame         = False            ; don't advance frame yet

res@cnFillOn         = True             ; turn on color

res@cnLinesOn        = False            ; turn off contour lines

res@cnLineLabelsOn   = False            ; turn off contour lines

res@cnFillPalette    = "ViBlGrWhYeOrRe" ; set White-in-Middle color map

res@lbLabelBarOn     = False            ; turn off individual cb's

res@mpFillOn         = False            ; turn off map fill

; Use a common scale

res@cnLevelSelectionMode = "ManualLevels"; manual set levels so lb consistent

res@cnMaxLevelValF       =  140.0       ; max level

res@cnMinLevelValF       = -res@cnMaxLevelValF     ; min level

res@cnLevelSpacingF      =   10.0       ; contour interval

LEVP    = 700

res@gsnCenterString      = LEVP+"hPa"

plot(0) = gsn_csm_contour_map(wks,uq(nt,{LEVP},:,:),res)

plot(1) = gsn_csm_contour_map(wks,vq(nt,{LEVP},:,:),res)

resP                     = True              ; modify the panel plot

resP@gsnPanelMainString  = date+": Unweighted Moisture Flux Components"

resP@gsnPanelLabelBar    = True              ; add common colorbar

gsn_panel(wks,plot,(/2,1/),resP)             ; now draw as one plot
```