Manaus-K34, Santarém-K67, and Santarém-K83, hereinafter referred to as K34, K67, and K83

**SAM First Run Notes: LBA-August**

Original path that I copied data: /rigel/glab/users/tmp

Two paths of my directory: /rigel/glab/users/yh3019 /rigel/home/yh3019

The path of Adam’s data: /rigel/glab/users/akm2203/bridges/data/SAM6.11.0

-How to calculate height z?

use dp/dz=-g to alculate dz and then get z.

-prm file

&SGS\_TKE

dosmagor = .true.

/

&PARAMETERS

caseid = '382x576x64\_250m\_250m\_5s\_213day\_LBA'

nrestart    = 0,

restart\_sep = .true.,

output\_sep  = .true.,

nstop       = 17280,

nprint      = 360,

nstat       = 360,

nstatfrq    = 15,

CEM  = .true.,

LAND = .true.,

z0 = 2.35,

dosgs           = .true.,

dodamping = .true.,

doupperbound    = .true.,

docloud         = .true.,

doprecip        = .true.,

docoriolis = .false.,

dolongwave = .true.,

doshortwave     = .true.,

doradforcing    = .false.,

doseasons = .true.

nrad            = 40,

SFC\_FLX\_FXD     = .true.,

SFC\_TAU\_FXD     = .false.,

dosurface = .true.,

dolargescale    = .false.,

dosfcforcing    = .true.,

donudging\_uv    = .false.,

donudging\_tq    = .true.,

tautqls         = 14400.,

doSAMconditionals     = .true.

dosatupdnconditionals = .true.

dx              = 250.,

dy              = 250.,

dt              = 5.,

day0            = 213.,

latitude0 = -2.5,

longitude0 = -60.,

nsave2D         = 360,

nsave2Dstart    = 1440,

nsave2Dend = 15840,

save2Dsep = .false.

save2Davg = .true.

nsave3D         = 360,

nsave3Dstart    = 1440,

nsave3Dend = 15840,

save3Dbin = .false.

/

prescribed snd and sfc, no prescription of radiation and large scale forcing

snd vertical layers: 52

sfc day period: 0~0.99

dolongwave     = .true.

doshortwave   = .true.

-nelapse

-Domain setting:

nx\_gl = 384

ny\_gl = 576

nz\_gl = 64

nsubdomains\_x  = 4

nsubdomains\_y  = 6

-

PS:

Initial snd file has three values in the very beginning: surface height, vertical layers and surface pressure; initial sfc file shows the diurnal cycle of

xarray.Dataset {

dimensions:

time = 142 ;

z = 64 ;

variables:

float32 z(z) ;

z:units = m ;

z:long\_name = height ;

float32 time(time) ;

time:units = day ;

time:long\_name = time ;

float32 p(z) ;

p:units = mb ;

p:long\_name = pressure ;

float32 SST(time) ;

SST:long\_name = SST ;

SST:units = K ;

float32 Ps(time) ;

Ps:long\_name = Surface Pressure ;

Ps:units = ;

float32 CLDSHD(time) ;

CLDSHD:long\_name = Shaded Cloud Fraction ;

CLDSHD:units = ;

float32 AREAPREC(time) ;

AREAPREC:long\_name = Surface Precip. Fraction ;

AREAPREC:units = ;

float32 CLD245(time) ;

CLD245:long\_name = Cloud Fraction above 245K level ;

CLD245:units = ;

float32 WMAX(time) ;

WMAX:long\_name = Maximum Updraft Velocity ;

WMAX:units = m/s ;

float32 UMAX(time) ;

UMAX:long\_name = Maximum Horizontal Wind ;

UMAX:units = m/s ;

float32 PREC(time) ;

PREC:long\_name = Surface Precipitation ;

PREC:units = mm/day ;

float32 LHF(time) ;

LHF:long\_name = Latent Heat Flux ;

LHF:units = W/m2 ;

float32 SHF(time) ;

SHF:long\_name = Sensible Heat Flux ;

SHF:units = W/m2 ;

float32 PW(time) ;

PW:long\_name = Precipitable Water ;

PW:units = mm ;

float32 PWOBS(time) ;

PWOBS:long\_name = Observed Precipitable Water ;

PWOBS:units = mm ;

float32 CWP(time) ;

CWP:long\_name = Cloud Water Path ;

CWP:units = g/m2 ;

float32 IWP(time) ;

IWP:long\_name = Ice Water Path ;

IWP:units = g/m2 ;

float32 RWP(time) ;

RWP:long\_name = Rain Water Path ;

RWP:units = g/m2 ;

float32 SWP(time) ;

SWP:long\_name = Snow Water Path ;

SWP:units = g/m2 ;

float32 GWP(time) ;

GWP:long\_name = Grauple Water Path ;

GWP:units = g/m2 ;

float32 CAPE(time) ;

CAPE:long\_name = CAPE ;

CAPE:units = J/kg ;

float32 CAPEOBS(time) ;

CAPEOBS:long\_name = CAPEOBS ;

CAPEOBS:units = J/kg ;

float32 CIN(time) ;

CIN:long\_name = CIN ;

CIN:units = J/kg ;

float32 CINOBS(time) ;

CINOBS:long\_name = CINOBS ;

CINOBS:units = J/kg ;

float32 LWNS(time) ;

LWNS:long\_name = Net LW flux at sfc ;

LWNS:units = W/m2 ;

float32 LWNT(time) ;

LWNT:long\_name = Net LW flux at Top-of-Model) ;

LWNT:units = W/m2 ;

float32 LWNTOA(time) ;

LWNTOA:long\_name = Net LW flux at TOA ;

LWNTOA:units = W/m2 ;

float32 LWNSC(time) ;

LWNSC:long\_name = Net LW flux at sfc (Clear Sky) ;

LWNSC:units = W/m2 ;

float32 LWNTOAC(time) ;

LWNTOAC:long\_name = Net LW flux at TOA (Clear Sky) ;

LWNTOAC:units = W/m2 ;

float32 LWDS(time) ;

LWDS:long\_name = Downward LW flux at sfc ;

LWDS:units = W/m2 ;

float32 SWNS(time) ;

SWNS:long\_name = Net SW flux at sfc ;

SWNS:units = W/m2 ;

float32 SWNT(time) ;

SWNT:long\_name = Net SW flux at Top-of-Model ;

SWNT:units = W/m2 ;

float32 SWNTOA(time) ;

SWNTOA:long\_name = Net SW flux at TOA ;

SWNTOA:units = W/m2 ;

float32 SWNSC(time) ;

SWNSC:long\_name = Net SW flux at sfc (Clear Sky) ;

SWNSC:units = W/m2 ;

float32 SWNTOAC(time) ;

SWNTOAC:long\_name = Net SW flux at TOA (Clear Sky) ;

SWNTOAC:units = W/m2 ;

float32 SWDS(time) ;

SWDS:long\_name = Downward SW flux at sfc ;

SWDS:units = W/m2 ;

float32 SOLIN(time) ;

SOLIN:long\_name = Incoming SW flux at TOA ;

SOLIN:units = W/m2 ;

float32 SSTOBS(time) ;

SSTOBS:long\_name = Observed SST ;

SSTOBS:units = K ;

float32 LHFOBS(time) ;

LHFOBS:long\_name = Observed Latent Heat Flux ;

LHFOBS:units = W/m2 ;

float32 SHFOBS(time) ;

SHFOBS:long\_name = Observed Sensible Heat Flux ;

SHFOBS:units = SHFOBS ;

float32 CLDLOW(time) ;

CLDLOW:long\_name = Low Cloud Fraction ;

CLDLOW:units = ;

float32 CLDMID(time) ;

CLDMID:long\_name = Middle Cloud Fraction ;

CLDMID:units = ;

float32 CLDHI(time) ;

CLDHI:long\_name = High Cloud Fraction ;

CLDHI:units = ;

float32 ISCCPTOT(time) ;

ISCCPTOT:long\_name = ISCCP Total Cloud Fraction (tau > 0.3) ;

ISCCPTOT:units = ;

float32 ISCCPLOW(time) ;

ISCCPLOW:long\_name = ISCCP Low Cloud Fraction (tau > 0.3) ;

ISCCPLOW:units = ;

float32 ISCCPMID(time) ;

ISCCPMID:long\_name = ISCCP Middle Cloud Fraction (tau > 0.3) ;

ISCCPMID:units = ;

float32 ISCCPHGH(time) ;

ISCCPHGH:long\_name = ISCCP High Cloud Fraction (tau > 0.3) ;

ISCCPHGH:units = ;

float32 MODISTOT(time) ;

MODISTOT:long\_name = MODIS Total Cloud Fraction ;

MODISTOT:units = ;

float32 MODISLOW(time) ;

MODISLOW:long\_name = MODIS Low Cloud Fraction ;

MODISLOW:units = ;

float32 MODISMID(time) ;

MODISMID:long\_name = MODIS Middle Cloud Fraction ;

MODISMID:units = ;

float32 MODISHGH(time) ;

MODISHGH:long\_name = MODIS High Cloud Fraction (tau > 0.3) ;

MODISHGH:units = ;

float32 MISRTOT(time) ;

MISRTOT:long\_name = MISR Total Cloud Fraction ;

MISRTOT:units = ;

float32 MODISREL(time) ;

MODISREL:long\_name = MODIS Effective Radius (Liquid) ;

MODISREL:units = mkm ;

float32 MODISREI(time) ;

MODISREI:long\_name = MODIS Effective Radius (Ice) ;

MODISREI:units = mkm ;

float32 MODISLWP(time) ;

MODISLWP:long\_name = MODIS Liquid Water Path ;

MODISLWP:units = g/m2 ;

float32 MODISIWP(time) ;

MODISIWP:long\_name = MODIS Ice Water Path ;

MODISIWP:units = g/m2 ;

float32 ISCCPTB(time) ;

ISCCPTB:long\_name = ISCCP Brightness Temperature ;

ISCCPTB:units = K ;

float32 ISCCPTBCLR(time) ;

ISCCPTBCLR:long\_name = ISCCP Brightness Temperature (Clear Sky) ;

ISCCPTBCLR:units = K ;

float32 MODISTOTL(time) ;

MODISTOTL:long\_name = MODIS Total Fraction (Liquid) ;

MODISTOTL:units = ;

float32 MODISTOTI(time) ;

MODISTOTI:long\_name = MODIS Total Fraction (Ice) ;

MODISTOTI:units = ;

float32 ISCCPTAU(time) ;

ISCCPTAU:long\_name = ISCCP Optical Path ;

ISCCPTAU:units = ;

float32 ISCCPALB(time) ;

ISCCPALB:long\_name = ISCCP Cloud Albedo ;

ISCCPALB:units = ;

float32 ISCCPPTOP(time) ;

ISCCPPTOP:long\_name = ISCCP Cloud-Top Pressure ;

ISCCPPTOP:units = mb ;

float32 MODISTAU(time) ;

MODISTAU:long\_name = MODIS Cloud Optical Path ;

MODISTAU:units = ;

float32 MODISTAUL(time) ;

MODISTAUL:long\_name = MODIS Cloud Optical Path (Liquid) ;

MODISTAUL:units = ;

float32 MODISTAUI(time) ;

MODISTAUI:long\_name = MODIS Cloud Optical Path (Ice) ;

MODISTAUI:units = ;

float32 MODISPTOP(time) ;

MODISPTOP:long\_name = MODIS Cloud-Top Pressure ;

MODISPTOP:units = mb ;

float32 MISRZTOP(time) ;

MISRZTOP:long\_name = MISR Cloud-Top Height ;

MISRZTOP:units = km ;

float32 ZINV(time) ;

ZINV:long\_name = GCSS Inversion Height ;

ZINV:units = km ;

float32 ZINV2(time) ;

ZINV2:long\_name = GCSS Variance of the Inversion Height ;

ZINV2:units = km2 ;

float32 ZCT(time) ;

ZCT:long\_name = GCSS Mean Cloud-top Height ;

ZCT:units = km ;

float32 ZCT2(time) ;

ZCT2:long\_name = GCSS Variance of Cloud-top Height ;

ZCT2:units = km2 ;

float32 ZCTMAX(time) ;

ZCTMAX:long\_name = GCSS Maximum Cloud-top Height ;

ZCTMAX:units = km ;

float32 ZCB(time) ;

ZCB:long\_name = GCSS Mean Cloud-base Height ;

ZCB:units = km ;

float32 ZCB2(time) ;

ZCB2:long\_name = GCSS Variance of Cloud-base Height ;

ZCB2:units = km ;

float32 ZCBMIN(time) ;

ZCBMIN:long\_name = GCSS Minimum Cloud-base Height ;

ZCBMIN:units = km ;

float32 LWP(time) ;

LWP:long\_name = GCSS Liquid Water Path ;

LWP:units = g/m2 ;

float32 LWP2(time) ;

LWP2:long\_name = GCSS Variance of Liquid Water Path ;

LWP2:units = (g/m2)^2 ;

float32 PRECMN(time) ;

PRECMN:long\_name = GCSS Precipitation Rate ;

PRECMN:units = mm/d ;

float32 PREC2(time) ;

PREC2:long\_name = GCSS Variance of Precipitation Rate ;

PREC2:units = (mm/d)^2 ;

float32 PRECMAX(time) ;

PRECMAX:long\_name = GCSS Maximum Precipitation Rate ;

PRECMAX:units = mm/d ;

float32 NCMN(time) ;

NCMN:long\_name = GCSS Mean Drop Number Comcentration ;

NCMN:units = #/cm3 ;

float32 NRMN(time) ;

NRMN:long\_name = GCSS Mean Rain Number Comcentration ;

NRMN:units = #/cm3 ;

float32 AREAPRTHR(time) ;

AREAPRTHR:long\_name = GCSS Precip. over threshold Area Fraction ;

AREAPRTHR:units = ;

float32 PRES(time, z) ;

PRES:long\_name = Pressure ;

PRES:units = mb ;

float32 U(time, z) ;

U:long\_name = x wind component ;

U:units = m/s ;

float32 V(time, z) ;

V:long\_name = y wind component ;

V:units = m/s ;

float32 UOBS(time, z) ;

UOBS:long\_name = Observed x wind component ;

UOBS:units = m/s ;

float32 VOBS(time, z) ;

VOBS:long\_name = Observed y wind component ;

VOBS:units = m/s ;

float32 WOBS(time, z) ;

WOBS:long\_name = Observed large-scale vert. velocity ;

WOBS:units = m/s ;

float32 RHO(time, z) ;

RHO:long\_name = Air density ;

RHO:units = kg/m3 ;

float32 MSE(time, z) ;

MSE:long\_name = Moist static energy ;

MSE:units = K ;

float32 DSE(time, z) ;

DSE:long\_name = Dry static energy ;

DSE:units = K ;

float32 SSE(time, z) ;

SSE:long\_name = Saturation static energy ;

SSE:units = K ;

float32 THETAE(time, z) ;

THETAE:long\_name = Equivalent (generalized) potential temperature ;

THETAE:units = K ;

float32 THETA(time, z) ;

THETA:long\_name = Potential temperature ;

THETA:units = K ;

float32 THETAV(time, z) ;

THETAV:long\_name = Virtual potential temperature ;

THETAV:units = K ;

float32 THETAL(time, z) ;

THETAL:long\_name = Liquid water potential temperature ;

THETAL:units = K ;

float32 TABS(time, z) ;

TABS:long\_name = Absolute temperature ;

TABS:units = K ;

float32 TABSOBS(time, z) ;

TABSOBS:long\_name = Observed Absolute temperature ;

TABSOBS:units = K ;

float32 TL(time, z) ;

TL:long\_name = Liquid water static energy ;

TL:units = K ;

float32 QT(time, z) ;

QT:long\_name = Total water (no rain/snow included) ;

QT:units = g/kg ;

float32 QV(time, z) ;

QV:long\_name = Water vapor ;

QV:units = g/kg ;

float32 QVOBS(time, z) ;

QVOBS:long\_name = Observed Water vapor ;

QVOBS:units = g/kg ;

float32 TTEND(time, z) ;

TTEND:long\_name = Observed Large-Scale Temperature Tendency ;

TTEND:units = K/day ;

float32 QTEND(time, z) ;

QTEND:long\_name = Observed Large-Scale Moisture Tendency ;

QTEND:units = g/kg/day ;

float32 QCL(time, z) ;

QCL:long\_name = Cloud water ;

QCL:units = g/kg ;

float32 QCI(time, z) ;

QCI:long\_name = Cloud ice ;

QCI:units = g/kg ;

float32 QPL(time, z) ;

QPL:long\_name = Rain content ;

QPL:units = g/kg ;

float32 QPI(time, z) ;

QPI:long\_name = Snow content ;

QPI:units = g/kg ;

float32 QN(time, z) ;

QN:long\_name = Cloud water and cloud ice ;

QN:units = g/kg ;

float32 QP(time, z) ;

QP:long\_name = Rain and Snow ;

QP:units = g/kg ;

float32 QSAT(time, z) ;

QSAT:long\_name = Saturation mixing ratio ;

QSAT:units = g/kg ;

float32 QCOND(time, z) ;

QCOND:long\_name = Total Condensate ;

QCOND:units = g/kg ;

float32 PRECIP(time, z) ;

PRECIP:long\_name = Precipitation flux ;

PRECIP:units = mm/day ;

float32 RELH(time, z) ;

RELH:long\_name = Relative humidity ;

RELH:units = per cent ;

float32 TLFLUX(time, z) ;

TLFLUX:long\_name = Liquid water static energy flux (Total) ;

TLFLUX:units = W/m2 ;

float32 TLFLUXS(time, z) ;

TLFLUXS:long\_name = Liquid water static energy flux (SGS) ;

TLFLUXS:units = W/m2 ;

float32 TVFLUX(time, z) ;

TVFLUX:long\_name = Buoyancy flux (Resolved) ;

TVFLUX:units = W/m2 ;

float32 QCFLUX(time, z) ;

QCFLUX:long\_name = Liquid water flux (Resolved) ;

QCFLUX:units = W/m2 ;

float32 QIFLUX(time, z) ;

QIFLUX:long\_name = Ice flux (Resolved) ;

QIFLUX:units = W/m2 ;

float32 UW(time, z) ;

UW:long\_name = x-momentum flux (Total) ;

UW:units = m2/s2 ;

float32 UWSB(time, z) ;

UWSB:long\_name = x-momentum flux (SGS) ;

UWSB:units = m2/s2 ;

float32 VW(time, z) ;

VW:long\_name = y-momentum flux (Total) ;

VW:units = m2/s2 ;

float32 VWSB(time, z) ;

VWSB:long\_name = y-momentum flux (SGS) ;

VWSB:units = m2/s2 ;

float32 RADLWUP(time, z) ;

RADLWUP:long\_name = Upward longwave radiative flux ;

RADLWUP:units = W/m2 ;

float32 RADLWDN(time, z) ;

RADLWDN:long\_name = Downward longwave radiative flux ;

RADLWDN:units = W/m2 ;

float32 RADSWUP(time, z) ;

RADSWUP:long\_name = Upward shortwave radiative flux ;

RADSWUP:units = W/m2 ;

float32 RADSWDN(time, z) ;

RADSWDN:long\_name = Downward shortwave radiative flux ;

RADSWDN:units = W/m2 ;

float32 RADQRLW(time, z) ;

RADQRLW:long\_name = Longwave heating rate ;

RADQRLW:units = K/day ;

float32 RADQRSW(time, z) ;

RADQRSW:long\_name = Shortwave heating rate ;

RADQRSW:units = K/day ;

float32 RADQR(time, z) ;

RADQR:long\_name = Radiative heating rate ;

RADQR:units = K/day ;

float32 RADQRS(time, z) ;

RADQRS:long\_name = Clear-sky Radiative heating rate ;

RADQRS:units = K/day ;

float32 RADQRC(time, z) ;

RADQRC:long\_name = Cloudy-sky Radiative heating rate ;

RADQRC:units = K/day ;

float32 Q1C(time, z) ;

Q1C:long\_name = Apparent heat source: Q1 - QR ;

Q1C:units = K/day ;

float32 Q2(time, z) ;

Q2:long\_name = Apparent moisture sink: Q2 ;

Q2:units = K/day ;

float32 U2(time, z) ;

U2:long\_name = Variance of the x wind component ;

U2:units = m2/s2 ;

float32 V2(time, z) ;

V2:long\_name = Variance of the y wind component ;

V2:units = m2/s2 ;

float32 W2(time, z) ;

W2:long\_name = Variance of the z wind component ;

W2:units = m2/s2 ;

float32 TL2(time, z) ;

TL2:long\_name = Variance of l.w.stat.energy ;

TL2:units = K2 ;

float32 TQ(time, z) ;

TQ:long\_name = CoVariance of HL and QT ;

TQ:units = K2 ;

float32 QT2(time, z) ;

QT2:long\_name = Variance of total water ;

QT2:units = g2/kg2 ;

float32 QC2(time, z) ;

QC2:long\_name = Variance of cloud water ;

QC2:units = g2/kg2 ;

float32 QI2(time, z) ;

QI2:long\_name = Variance of cloud ice ;

QI2:units = g2/kg2 ;

float32 QS2(time, z) ;

QS2:long\_name = Variance of saturation mixing ratio ;

QS2:units = g2/kg2 ;

float32 W3(time, z) ;

W3:long\_name = Third moment of the vertical velocity ;

W3:units = m3/s3 ;

float32 AUP(time, z) ;

AUP:long\_name = domain fraction occupied by updrafts ;

AUP:units = ;

float32 WSKEW(time, z) ;

WSKEW:long\_name = Vertical velocity skewness W3/(W2)^3/2 ;

WSKEW:units = ;

float32 TKE(time, z) ;

TKE:long\_name = Turbulent kinetic energy (Resolved) ;

TKE:units = m2/s2 ;

float32 TKES(time, z) ;

TKES:long\_name = Turbulent kinetic energy (SGS) ;

TKES:units = m2/s2 ;

float32 TK(time, z) ;

TK:long\_name = Eddy viscosity ;

TK:units = m2/s ;

float32 TKH(time, z) ;

TKH:long\_name = Eddy diffusivity ;

TKH:units = m2/s ;

float32 HLADV(time, z) ;

HLADV:long\_name = Advective Transport of HL ;

HLADV:units = K/day ;

float32 HLDIFF(time, z) ;

HLDIFF:long\_name = Diffusive Transport of HL ;

HLDIFF:units = K/day ;

float32 HLLAT(time, z) ;

HLLAT:long\_name = Latent Heating of HL ;

HLLAT:units = K/day ;

float32 HLRAD(time, z) ;

HLRAD:long\_name = Radiative Heating of HL ;

HLRAD:units = K/day ;

float32 SHEAR(time, z) ;

SHEAR:long\_name = Shear production of TKE (resolved) ;

SHEAR:units = m2/s3 ;

float32 SHEARS(time, z) ;

SHEARS:long\_name = Shear production of TKE (SGS) ;

SHEARS:units = m2/s3 ;

float32 BUOYA(time, z) ;

BUOYA:long\_name = Buoyancy production of TKE (resolved) ;

BUOYA:units = m2/s3 ;

float32 BUOYAS(time, z) ;

BUOYAS:long\_name = Buoyancy production of TKE (SGS) ;

BUOYAS:units = m2/s3 ;

float32 ADVTR(time, z) ;

ADVTR:long\_name = Turbulent advective transport of TKE (Resolved) ;

ADVTR:units = m2/s3 ;

float32 PRESSTR(time, z) ;

PRESSTR:long\_name = Pressure transport of TKE (Resolved) ;

PRESSTR:units = m2/s3 ;

float32 ADVTRS(time, z) ;

ADVTRS:long\_name = Turbulent+pressure transport of TKE (SGS) ;

ADVTRS:units = m2/s3 ;

float32 DIFTR(time, z) ;

DIFTR:long\_name = SGS transport of TKE (Resolved) ;

DIFTR:units = m2/s3 ;

float32 DISSIP(time, z) ;

DISSIP:long\_name = Dissipation (Resolved) ;

DISSIP:units = m2/s3 ;

float32 DISSIPS(time, z) ;

DISSIPS:long\_name = Dissipation (SGS) ;

DISSIPS:units = m2/s3 ;

float32 WUADV(time, z) ;

WUADV:long\_name = WU advection (Resolved) ;

WUADV:units = m2/s3 ;

float32 WVADV(time, z) ;

WVADV:long\_name = WV advection (Resolved) ;

WVADV:units = m2/s3 ;

float32 WUPRES(time, z) ;

WUPRES:long\_name = WU pressure (Resolved) ;

WUPRES:units = m2/s3 ;

float32 WVPRES(time, z) ;

WVPRES:long\_name = WV pressure (Resolved) ;

WVPRES:units = m2/s3 ;

float32 WUANIZ(time, z) ;

WUANIZ:long\_name = WU return to anisotropy (Resolved) ;

WUANIZ:units = m2/s3 ;

float32 WVANIZ(time, z) ;

WVANIZ:long\_name = WV return to anisotropy (Resolved) ;

WVANIZ:units = m2/s3 ;

float32 WUSHEAR(time, z) ;

WUSHEAR:long\_name = WU shear(Resolved) ;

WUSHEAR:units = m2/s3 ;

float32 WVSHEAR(time, z) ;

WVSHEAR:long\_name = WV shear (Resolved) ;

WVSHEAR:units = m2/s3 ;

float32 WUBUOY(time, z) ;

WUBUOY:long\_name = WU buoyancy (Resolved) ;

WUBUOY:units = m2/s3 ;

float32 WVBUOY(time, z) ;

WVBUOY:long\_name = WV buoyancy (Resolved) ;

WVBUOY:units = m2/s3 ;

float32 WUDIFF(time, z) ;

WUDIFF:long\_name = WU diffusion (Resolved) ;

WUDIFF:units = m2/s3 ;

float32 WVDIFF(time, z) ;

WVDIFF:long\_name = WV diffusion (Resolved) ;

WVDIFF:units = m2/s3 ;

float32 W2ADV(time, z) ;

W2ADV:long\_name = WU advection (Resolved) ;

W2ADV:units = m2/s3 ;

float32 W2PRES(time, z) ;

W2PRES:long\_name = W2 pressure (Resolved) ;

W2PRES:units = m2/s3 ;

float32 W2REDIS(time, z) ;

W2REDIS:long\_name = W2 pressure redistribution(Resolved) ;

W2REDIS:units = m2/s3 ;

float32 W2BUOY(time, z) ;

W2BUOY:long\_name = W2 buoyancy (Resolved) ;

W2BUOY:units = m2/s3 ;

float32 W2DIFF(time, z) ;

W2DIFF:long\_name = W2 diffusion (Resolved) ;

W2DIFF:units = m2/s3 ;

float32 TWGRAD(time, z) ;

TWGRAD:long\_name = Gradient production of l.w.s.e flux ;

TWGRAD:units = m s-2 K ;

float32 TWADV(time, z) ;

TWADV:long\_name = Transport of l.w.s.e flux ;

TWADV:units = m s-2 K ;

float32 TWDIFF(time, z) ;

TWDIFF:long\_name = Diffusion of l.w.s.e flux ;

TWDIFF:units = m s-2 K ;

float32 TWBUOY(time, z) ;

TWBUOY:long\_name = Buoyancy production of l.w.s.e flux ;

TWBUOY:units = m s-2 K ;

float32 TWPRES(time, z) ;

TWPRES:long\_name = Pressure production of l.w.s.e flux ;

TWPRES:units = m s-2 K ;

float32 TWPREC(time, z) ;

TWPREC:long\_name = Precip. production of l.w.s.e flux ;

TWPREC:units = m s-2 K ;

float32 QWGRAD(time, z) ;

QWGRAD:long\_name = Gradient production of total water flux ;

QWGRAD:units = m s-2 K ;

float32 QWADV(time, z) ;

QWADV:long\_name = Transport of total water flux ;

QWADV:units = m s-2 K ;

float32 QWDIFF(time, z) ;

QWDIFF:long\_name = Diffusion of total water flux ;

QWDIFF:units = m s-2 K ;

float32 QWBUOY(time, z) ;

QWBUOY:long\_name = Buoyancy production of total water flux ;

QWBUOY:units = m s-2 ;

float32 QWPRES(time, z) ;

QWPRES:long\_name = Pressure production of total water flux ;

QWPRES:units = m s-2 ;

float32 QWPREC(time, z) ;

QWPREC:long\_name = Precip. production of total water flux ;

QWPREC:units = m s-2 ;

float32 T2ADVTR(time, z) ;

T2ADVTR:long\_name = Transport of l.w.s.e variance ;

T2ADVTR:units = K2/s ;

float32 T2GRAD(time, z) ;

T2GRAD:long\_name = Gradient production of l.w.s.e variance ;

T2GRAD:units = K2/s ;

float32 T2DISSIP(time, z) ;

T2DISSIP:long\_name = Dissipation of l.w.s.e variance ;

T2DISSIP:units = K2/s ;

float32 T2DIFTR(time, z) ;

T2DIFTR:long\_name = SGS transport of l.w.s.e variance ;

T2DIFTR:units = K2/s ;

float32 T2PREC(time, z) ;

T2PREC:long\_name = Precipitation production of l.w.s.e variance ;

T2PREC:units = K2/s ;

float32 Q2ADVTR(time, z) ;

Q2ADVTR:long\_name = Transport of total water variance ;

Q2ADVTR:units = 1/s ;

float32 Q2GRAD(time, z) ;

Q2GRAD:long\_name = Gradient production of total water variance ;

Q2GRAD:units = 1/s ;

float32 Q2DISSIP(time, z) ;

Q2DISSIP:long\_name = Dissipation of total water variance ;

Q2DISSIP:units = 1/s ;

float32 Q2DIFTR(time, z) ;

Q2DIFTR:long\_name = SGS transport of total water variance ;

Q2DIFTR:units = 1/s ;

float32 Q2PREC(time, z) ;

Q2PREC:long\_name = Precipitation production of total water variance ;

Q2PREC:units = 1/s ;

float32 HYDRO(time, z) ;

HYDRO:long\_name = Total fraction of hydrometeors ;

HYDRO:units = ;

float32 MCUP(time, z) ;

MCUP:long\_name = Updraft cloud mass flux ;

MCUP:units = kg/m2/s ;

float32 MCDNS(time, z) ;

MCDNS:long\_name = Downdraft saturated cloud mass flux ;

MCDNS:units = kg/m2/s ;

float32 MCDNU(time, z) ;

MCDNU:long\_name = Downdraft unsaturated mass flux ;

MCDNU:units = kg/m2/s ;

float32 MC(time, z) ;

MC:long\_name = Cloud mass flux ;

MC:units = kg/m2/s ;

float32 CORECL(time, z) ;

CORECL:long\_name = Cloudy Updraft core fraction ;

CORECL:units = ;

float32 COREDNCL(time, z) ;

COREDNCL:long\_name = Cloudy Downdraft core fraction ;

COREDNCL:units = ;

float32 MCRUP(time, z) ;

MCRUP:long\_name = Updraft core mass flux ;

MCRUP:units = kg/m2/s ;

float32 MCRDNS(time, z) ;

MCRDNS:long\_name = Downdraft cloud core mass flux ;

MCRDNS:units = kg/m2/s ;

float32 MCRDNU(time, z) ;

MCRDNU:long\_name = Downdraft unsaturated core mass flux ;

MCRDNU:units = kg/m2/s ;

float32 MCR(time, z) ;

MCR:long\_name = Core mass flux ;

MCR:units = kg/m2/s ;

float32 QVTEND(time, z) ;

QVTEND:long\_name = Large-Scale Vertical Advection Moisture Tendency ;

QVTEND:units = g/kg/day ;

float32 QHTEND(time, z) ;

QHTEND:long\_name = Large-Scale Horizontal Advection Moisture Tendency ;

QHTEND:units = g/kg/day ;

float32 QNUDGE(time, z) ;

QNUDGE:long\_name = Large-Scale Moisture Nudging ;

QNUDGE:units = g/kg/day ;

float32 THTEND(time, z) ;

THTEND:long\_name = Large-Scale Horizontal Advection Temperature Tendency ;

THTEND:units = K/day ;

float32 TVTEND(time, z) ;

TVTEND:long\_name = Large-Scale Vertical Advection Temperature Tendency ;

TVTEND:units = K/day ;

float32 TNUDGE(time, z) ;

TNUDGE:long\_name = Large-Scale Temperature Nudging ;

TNUDGE:units = K/day ;

float32 WSTAR3(time, z) ;

WSTAR3:long\_name = Value at inversion is w\*^3 (conv. vel. scale) ;

WSTAR3:units = m3/s3 ;

float32 UADV(time, z) ;

UADV:long\_name = Resolved zonal momentum flux convergence ;

UADV:units = m/s/day ;

float32 VADV(time, z) ;

VADV:long\_name = Resolved meridioinal momentum flux convergence ;

VADV:units = m/s/day ;

float32 UDIFF(time, z) ;

UDIFF:long\_name = Subgrid zonal momentum flux convergence ;

UDIFF:units = m/s/day ;

float32 VDIFF(time, z) ;

VDIFF:long\_name = Subgrid meridioinal momentum flux convergence ;

VDIFF:units = m/s/day ;

float32 UNUDGE(time, z) ;

UNUDGE:long\_name = Zonal velocity nudging ;

UNUDGE:units = m/s/day ;

float32 VNUDGE(time, z) ;

VNUDGE:long\_name = Meridional velocity nudging ;

VNUDGE:units = m/s/day ;

float32 ULSADVV(time, z) ;

ULSADVV:long\_name = Large-scale vertical advection of Zonal velocity ;

ULSADVV:units = m/s/day ;

float32 VLSADVV(time, z) ;

VLSADVV:long\_name = Large-scale vertical advection of meridional velocity ;

VLSADVV:units = m/s/day ;

float32 USTOR(time, z) ;

USTOR:long\_name = Zonal velocity storage ;

USTOR:units = m/s/day ;

float32 VSTOR(time, z) ;

VSTOR:long\_name = Meridional velocity storage ;

VSTOR:units = m/s/day ;

float32 UTENDCOR(time, z) ;

UTENDCOR:long\_name = Zonal velocity tendency due to Coriolis effect ;

UTENDCOR:units = m/s/day ;

float32 URESID(time, z) ;

URESID:long\_name = Zonal velocity budget residual ;

URESID:units = m/s/day ;

float32 VTENDCOR(time, z) ;

VTENDCOR:long\_name = Meridional velocity budget residual ;

VTENDCOR:units = m/s/day ;

float32 VRESID(time, z) ;

VRESID:long\_name = Meridional velocity budget residual ;

VRESID:units = m/s/day ;

float32 HLSTOR(time, z) ;

HLSTOR:long\_name = Liquid-ice static energy storage ;

HLSTOR:units = K/day ;

float32 QTSTOR(time, z) ;

QTSTOR:long\_name = Total water storage ;

QTSTOR:units = K/day ;

float32 RADQRCLW(time, z) ;

RADQRCLW:long\_name = Clearsky longwave heating rate ;

RADQRCLW:units = K/d ;

float32 RADQRCSW(time, z) ;

RADQRCSW:long\_name = Clearsky shortwave heating rate ;

RADQRCSW:units = K/d ;

float32 CLD(time, z) ;

CLD:long\_name = cloud Fraction ;

CLD:units = ;

float32 WCLD(time, z) ;

WCLD:long\_name = Mean W in cloud ;

WCLD:units = m/s ;

float32 UCLD(time, z) ;

UCLD:long\_name = Mean U in cloud ;

UCLD:units = m/s ;

float32 VCLD(time, z) ;

VCLD:long\_name = Mean V in cloud ;

VCLD:units = m/s ;

float32 MSECLD(time, z) ;

MSECLD:long\_name = Mean moist static energy in cloud ;

MSECLD:units = K ;

float32 DSECLD(time, z) ;

DSECLD:long\_name = Mean dry static energy in cloud ;

DSECLD:units = K ;

float32 TLCLD(time, z) ;

TLCLD:long\_name = Mean liquid-ice static energy in cloud ;

TLCLD:units = K ;

float32 TACLD(time, z) ;

TACLD:long\_name = Mean TABS in cloud ;

TACLD:units = K ;

float32 TVCLD(time, z) ;

TVCLD:long\_name = Mean THETAV in cloud ;

TVCLD:units = K ;

float32 TVCLDA(time, z) ;

TVCLDA:long\_name = Mean THETAV anomaly in cloud ;

TVCLDA:units = K ;

float32 QTCLD(time, z) ;

QTCLD:long\_name = Mean QT in cloud ;

QTCLD:units = g/kg ;

float32 QNCLD(time, z) ;

QNCLD:long\_name = Mean QN in cloud ;

QNCLD:units = g/kg ;

float32 QPCLD(time, z) ;

QPCLD:long\_name = Mean QP in cloud ;

QPCLD:units = g/kg ;

float32 WCLDA(time, z) ;

WCLDA:long\_name = W in cloud averaged over the whole domain ;

WCLDA:units = m/s ;

float32 TLWCLD(time, z) ;

TLWCLD:long\_name = TLW in cloud averaged over the whole domain ;

TLWCLD:units = Km/s ;

float32 TVWCLD(time, z) ;

TVWCLD:long\_name = TVW in cloud averaged over the whole domain ;

TVWCLD:units = Km/s ;

float32 QTWCLD(time, z) ;

QTWCLD:long\_name = QTW in cloud averaged over the whole domain ;

QTWCLD:units = g/kg m/s ;

float32 QCWCLD(time, z) ;

QCWCLD:long\_name = QCW in cloud averaged over the whole domain ;

QCWCLD:units = g/kg m/s ;

float32 QIWCLD(time, z) ;

QIWCLD:long\_name = QIW in cloud averaged over the whole domain ;

QIWCLD:units = g/kg m/s ;

float32 HFCLD(time, z) ;

HFCLD:long\_name = Mean Frozen MSE in cloud ;

HFCLD:units = K ;

float32 HFCLDA(time, z) ;

HFCLDA:long\_name = Mean Frozen MSE anomaly in cloud ;

HFCLDA:units = K ;

float32 UCLDA(time, z) ;

UCLDA:long\_name = Mean U anomaly in cloud ;

UCLDA:units = m/s ;

float32 VCLDA(time, z) ;

VCLDA:long\_name = Mean V anomaly in cloud ;

VCLDA:units = m/s ;

float32 UPGFCLD(time, z) ;

UPGFCLD:long\_name = Zonal pressure gradient in cloud ;

UPGFCLD:units = m/s2 ;

float32 VPGFCLD(time, z) ;

VPGFCLD:long\_name = Meridional pressure gradient in cloud ;

VPGFCLD:units = m/s2 ;

float32 WPGFCLD(time, z) ;

WPGFCLD:long\_name = Vertical pressure gradient in cloud ;

WPGFCLD:units = m/s2 ;

float32 UWCLD(time, z) ;

UWCLD:long\_name = UW in cloud ;

UWCLD:units = m2/s2 ;

float32 VWCLD(time, z) ;

VWCLD:long\_name = VW in cloud ;

VWCLD:units = m2/s2 ;

float32 UWSBCLD(time, z) ;

UWSBCLD:long\_name = Subgrid UW in cloud ;

UWSBCLD:units = m2/s2 ;

float32 VWSBCLD(time, z) ;

VWSBCLD:long\_name = Subgrid VW in cloud ;

VWSBCLD:units = m2/s2 ;

float32 MFCLD(time, z) ;

MFCLD:long\_name = Mass flux in cloud averaged over the whole domain ;

MFCLD:units = kg/m2/s ;

float32 MFHCLD(time, z) ;

MFHCLD:long\_name = RHO\*W\*HF in cloud averaged over the whole domain ;

MFHCLD:units = K kg/m2/s ;

float32 MFHCLDA(time, z) ;

MFHCLDA:long\_name = RHO\*W\*HF anomaly in cloud averaged over the whole domain ;

MFHCLDA:units = K kg/m2/s ;

float32 MFTLCLD(time, z) ;

MFTLCLD:long\_name = RHO\*W\*TL in cloud averaged over the whole domain ;

MFTLCLD:units = K kg/m2/s ;

float32 MFTLCLDA(time, z) ;

MFTLCLDA:long\_name = RHO\*W\*TL anomaly in cloud averaged over the whole domain ;

MFTLCLDA:units = K kg/m2/s ;

float32 MFTVCLD(time, z) ;

MFTVCLD:long\_name = RHO\*W\*TV in cloud averaged over the whole domain ;

MFTVCLD:units = K kg/m2/s ;

float32 MFTVCLDA(time, z) ;

MFTVCLDA:long\_name = RHO\*W\*TV anomaly in cloud averaged over the whole domain ;

MFTVCLDA:units = K kg/m2/s ;

float32 MFQTCLD(time, z) ;

MFQTCLD:long\_name = RHO\*W\*QT in cloud averaged over the whole domain ;

MFQTCLD:units = g/m2/s ;

float32 MFQTCLDA(time, z) ;

MFQTCLDA:long\_name = RHO\*W\*QT anomaly in cloud averaged over the whole domain ;

MFQTCLDA:units = g/m2/s ;

float32 RUWCLD(time, z) ;

RUWCLD:long\_name = RHOUW in cloud averaged over the whole domain ;

RUWCLD:units = kg/m/s2 ;

float32 RVWCLD(time, z) ;

RVWCLD:long\_name = RHOVW in cloud averaged over the whole domain ;

RVWCLD:units = kg/m/s2 ;

float32 RWWCLD(time, z) ;

RWWCLD:long\_name = RHOWW in cloud averaged over the whole domain ;

RWWCLD:units = kg/m/s2 ;

float32 COR(time, z) ;

COR:long\_name = core Fraction ;

COR:units = ;

float32 WCOR(time, z) ;

WCOR:long\_name = Mean W in core ;

WCOR:units = m/s ;

float32 UCOR(time, z) ;

UCOR:long\_name = Mean U in core ;

UCOR:units = m/s ;

float32 VCOR(time, z) ;

VCOR:long\_name = Mean V in core ;

VCOR:units = m/s ;

float32 MSECOR(time, z) ;

MSECOR:long\_name = Mean moist static energy in core ;

MSECOR:units = K ;

float32 DSECOR(time, z) ;

DSECOR:long\_name = Mean dry static energy in core ;

DSECOR:units = K ;

float32 TLCOR(time, z) ;

TLCOR:long\_name = Mean liquid-ice static energy in core ;

TLCOR:units = K ;

float32 TACOR(time, z) ;

TACOR:long\_name = Mean TABS in core ;

TACOR:units = K ;

float32 TVCOR(time, z) ;

TVCOR:long\_name = Mean THETAV in core ;

TVCOR:units = K ;

float32 TVCORA(time, z) ;

TVCORA:long\_name = Mean THETAV anomaly in core ;

TVCORA:units = K ;

float32 QTCOR(time, z) ;

QTCOR:long\_name = Mean QT in core ;

QTCOR:units = g/kg ;

float32 QNCOR(time, z) ;

QNCOR:long\_name = Mean QN in core ;

QNCOR:units = g/kg ;

float32 QPCOR(time, z) ;

QPCOR:long\_name = Mean QP in core ;

QPCOR:units = g/kg ;

float32 WCORA(time, z) ;

WCORA:long\_name = W in core averaged over the whole domain ;

WCORA:units = m/s ;

float32 TLWCOR(time, z) ;

TLWCOR:long\_name = TLW in core averaged over the whole domain ;

TLWCOR:units = Km/s ;

float32 TVWCOR(time, z) ;

TVWCOR:long\_name = TVW in core averaged over the whole domain ;

TVWCOR:units = Km/s ;

float32 QTWCOR(time, z) ;

QTWCOR:long\_name = QTW in core averaged over the whole domain ;

QTWCOR:units = g/kg m/s ;

float32 QCWCOR(time, z) ;

QCWCOR:long\_name = QCW in core averaged over the whole domain ;

QCWCOR:units = g/kg m/s ;

float32 QIWCOR(time, z) ;

QIWCOR:long\_name = QIW in core averaged over the whole domain ;

QIWCOR:units = g/kg m/s ;

float32 HFCOR(time, z) ;

HFCOR:long\_name = Mean Frozen MSE in core ;

HFCOR:units = K ;

float32 HFCORA(time, z) ;

HFCORA:long\_name = Mean Frozen MSE anomaly in core ;

HFCORA:units = K ;

float32 UCORA(time, z) ;

UCORA:long\_name = Mean U anomaly in core ;

UCORA:units = m/s ;

float32 VCORA(time, z) ;

VCORA:long\_name = Mean V anomaly in core ;

VCORA:units = m/s ;

float32 UPGFCOR(time, z) ;

UPGFCOR:long\_name = Zonal pressure gradient in core ;

UPGFCOR:units = m/s2 ;

float32 VPGFCOR(time, z) ;

VPGFCOR:long\_name = Meridional pressure gradient in core ;

VPGFCOR:units = m/s2 ;

float32 WPGFCOR(time, z) ;

WPGFCOR:long\_name = Vertical pressure gradient in core ;

WPGFCOR:units = m/s2 ;

float32 UWCOR(time, z) ;

UWCOR:long\_name = UW in core ;

UWCOR:units = m2/s2 ;

float32 VWCOR(time, z) ;

VWCOR:long\_name = VW in core ;

VWCOR:units = m2/s2 ;

float32 UWSBCOR(time, z) ;

UWSBCOR:long\_name = Subgrid UW in core ;

UWSBCOR:units = m2/s2 ;

float32 VWSBCOR(time, z) ;

VWSBCOR:long\_name = Subgrid VW in core ;

VWSBCOR:units = m2/s2 ;

float32 MFCOR(time, z) ;

MFCOR:long\_name = Mass flux in core averaged over the whole domain ;

MFCOR:units = kg/m2/s ;

float32 MFHCOR(time, z) ;

MFHCOR:long\_name = RHO\*W\*HF in core averaged over the whole domain ;

MFHCOR:units = K kg/m2/s ;

float32 MFHCORA(time, z) ;

MFHCORA:long\_name = RHO\*W\*HF anomaly in core averaged over the whole domain ;

MFHCORA:units = K kg/m2/s ;

float32 MFTLCOR(time, z) ;

MFTLCOR:long\_name = RHO\*W\*TL in core averaged over the whole domain ;

MFTLCOR:units = K kg/m2/s ;

float32 MFTLCORA(time, z) ;

MFTLCORA:long\_name = RHO\*W\*TL anomaly in core averaged over the whole domain ;

MFTLCORA:units = K kg/m2/s ;

float32 MFTVCOR(time, z) ;

MFTVCOR:long\_name = RHO\*W\*TV in core averaged over the whole domain ;

MFTVCOR:units = K kg/m2/s ;

float32 MFTVCORA(time, z) ;

MFTVCORA:long\_name = RHO\*W\*TV anomaly in core averaged over the whole domain ;

MFTVCORA:units = K kg/m2/s ;

float32 MFQTCOR(time, z) ;

MFQTCOR:long\_name = RHO\*W\*QT in core averaged over the whole domain ;

MFQTCOR:units = g/m2/s ;

float32 MFQTCORA(time, z) ;

MFQTCORA:long\_name = RHO\*W\*QT anomaly in core averaged over the whole domain ;

MFQTCORA:units = g/m2/s ;

float32 RUWCOR(time, z) ;

RUWCOR:long\_name = RHOUW in core averaged over the whole domain ;

RUWCOR:units = kg/m/s2 ;

float32 RVWCOR(time, z) ;

RVWCOR:long\_name = RHOVW in core averaged over the whole domain ;

RVWCOR:units = kg/m/s2 ;

float32 RWWCOR(time, z) ;

RWWCOR:long\_name = RHOWW in core averaged over the whole domain ;

RWWCOR:units = kg/m/s2 ;

float32 CDN(time, z) ;

CDN:long\_name = downdraft core Fraction ;

CDN:units = ;

float32 WCDN(time, z) ;

WCDN:long\_name = Mean W in downdraft core ;

WCDN:units = m/s ;

float32 UCDN(time, z) ;

UCDN:long\_name = Mean U in downdraft core ;

UCDN:units = m/s ;

float32 VCDN(time, z) ;

VCDN:long\_name = Mean V in downdraft core ;

VCDN:units = m/s ;

float32 MSECDN(time, z) ;

MSECDN:long\_name = Mean moist static energy in downdraft core ;

MSECDN:units = K ;

float32 DSECDN(time, z) ;

DSECDN:long\_name = Mean dry static energy in downdraft core ;

DSECDN:units = K ;

float32 TLCDN(time, z) ;

TLCDN:long\_name = Mean liquid-ice static energy in downdraft core ;

TLCDN:units = K ;

float32 TACDN(time, z) ;

TACDN:long\_name = Mean TABS in downdraft core ;

TACDN:units = K ;

float32 TVCDN(time, z) ;

TVCDN:long\_name = Mean THETAV in downdraft core ;

TVCDN:units = K ;

float32 TVCDNA(time, z) ;

TVCDNA:long\_name = Mean THETAV anomaly in downdraft core ;

TVCDNA:units = K ;

float32 QTCDN(time, z) ;

QTCDN:long\_name = Mean QT in downdraft core ;

QTCDN:units = g/kg ;

float32 QNCDN(time, z) ;

QNCDN:long\_name = Mean QN in downdraft core ;

QNCDN:units = g/kg ;

float32 QPCDN(time, z) ;

QPCDN:long\_name = Mean QP in downdraft core ;

QPCDN:units = g/kg ;

float32 WCDNA(time, z) ;

WCDNA:long\_name = W in downdraft core averaged over the whole domain ;

WCDNA:units = m/s ;

float32 TLWCDN(time, z) ;

TLWCDN:long\_name = TLW in downdraft core averaged over the whole domain ;

TLWCDN:units = Km/s ;

float32 TVWCDN(time, z) ;

TVWCDN:long\_name = TVW in downdraft core averaged over the whole domain ;

TVWCDN:units = Km/s ;

float32 QTWCDN(time, z) ;

QTWCDN:long\_name = QTW in downdraft core averaged over the whole domain ;

QTWCDN:units = g/kg m/s ;

float32 QCWCDN(time, z) ;

QCWCDN:long\_name = QCW in downdraft core averaged over the whole domain ;

QCWCDN:units = g/kg m/s ;

float32 QIWCDN(time, z) ;

QIWCDN:long\_name = QIW in downdraft core averaged over the whole domain ;

QIWCDN:units = g/kg m/s ;

float32 HFCDN(time, z) ;

HFCDN:long\_name = Mean Frozen MSE in downdraft core ;

HFCDN:units = K ;

float32 HFCDNA(time, z) ;

HFCDNA:long\_name = Mean Frozen MSE anomaly in downdraft core ;

HFCDNA:units = K ;

float32 UCDNA(time, z) ;

UCDNA:long\_name = Mean U anomaly in downdraft core ;

UCDNA:units = m/s ;

float32 VCDNA(time, z) ;

VCDNA:long\_name = Mean V anomaly in downdraft core ;

VCDNA:units = m/s ;

float32 UPGFCDN(time, z) ;

UPGFCDN:long\_name = Zonal pressure gradient in downdraft core ;

UPGFCDN:units = m/s2 ;

float32 VPGFCDN(time, z) ;

VPGFCDN:long\_name = Meridional pressure gradient in downdraft core ;

VPGFCDN:units = m/s2 ;

float32 WPGFCDN(time, z) ;

WPGFCDN:long\_name = Vertical pressure gradient in downdraft core ;

WPGFCDN:units = m/s2 ;

float32 UWCDN(time, z) ;

UWCDN:long\_name = UW in downdraft core ;

UWCDN:units = m2/s2 ;

float32 VWCDN(time, z) ;

VWCDN:long\_name = VW in downdraft core ;

VWCDN:units = m2/s2 ;

float32 UWSBCDN(time, z) ;

UWSBCDN:long\_name = Subgrid UW in downdraft core ;

UWSBCDN:units = m2/s2 ;

float32 VWSBCDN(time, z) ;

VWSBCDN:long\_name = Subgrid VW in downdraft core ;

VWSBCDN:units = m2/s2 ;

float32 MFCDN(time, z) ;

MFCDN:long\_name = Mass flux in downdraft core averaged over the whole domain ;

MFCDN:units = kg/m2/s ;

float32 MFHCDN(time, z) ;

MFHCDN:long\_name = RHO\*W\*HF in downdraft core averaged over the whole domain ;

MFHCDN:units = K kg/m2/s ;

float32 MFHCDNA(time, z) ;

MFHCDNA:long\_name = RHO\*W\*HF anomaly in downdraft core averaged over the whole domain ;

MFHCDNA:units = K kg/m2/s ;

float32 MFTLCDN(time, z) ;

MFTLCDN:long\_name = RHO\*W\*TL in downdraft core averaged over the whole domain ;

MFTLCDN:units = K kg/m2/s ;

float32 MFTLCDNA(time, z) ;

MFTLCDNA:long\_name = RHO\*W\*TL anomaly in downdraft core averaged over the whole domain ;

MFTLCDNA:units = K kg/m2/s ;

float32 MFTVCDN(time, z) ;

MFTVCDN:long\_name = RHO\*W\*TV in downdraft core averaged over the whole domain ;

MFTVCDN:units = K kg/m2/s ;

float32 MFTVCDNA(time, z) ;

MFTVCDNA:long\_name = RHO\*W\*TV anomaly in downdraft core averaged over the whole domain ;

MFTVCDNA:units = K kg/m2/s ;

float32 MFQTCDN(time, z) ;

MFQTCDN:long\_name = RHO\*W\*QT in downdraft core averaged over the whole domain ;

MFQTCDN:units = g/m2/s ;

float32 MFQTCDNA(time, z) ;

MFQTCDNA:long\_name = RHO\*W\*QT anomaly in downdraft core averaged over the whole domain ;

MFQTCDNA:units = g/m2/s ;

float32 RUWCDN(time, z) ;

RUWCDN:long\_name = RHOUW in downdraft core averaged over the whole domain ;

RUWCDN:units = kg/m/s2 ;

float32 RVWCDN(time, z) ;

RVWCDN:long\_name = RHOVW in downdraft core averaged over the whole domain ;

RVWCDN:units = kg/m/s2 ;

float32 RWWCDN(time, z) ;

RWWCDN:long\_name = RHOWW in downdraft core averaged over the whole domain ;

RWWCDN:units = kg/m/s2 ;

float32 SUP(time, z) ;

SUP:long\_name = saturated updrafts Fraction ;

SUP:units = ;

float32 WSUP(time, z) ;

WSUP:long\_name = Mean W in saturated updrafts ;

WSUP:units = m/s ;

float32 USUP(time, z) ;

USUP:long\_name = Mean U in saturated updrafts ;

USUP:units = m/s ;

float32 VSUP(time, z) ;

VSUP:long\_name = Mean V in saturated updrafts ;

VSUP:units = m/s ;

float32 MSESUP(time, z) ;

MSESUP:long\_name = Mean moist static energy in saturated updrafts ;

MSESUP:units = K ;

float32 DSESUP(time, z) ;

DSESUP:long\_name = Mean dry static energy in saturated updrafts ;

DSESUP:units = K ;

float32 TLSUP(time, z) ;

TLSUP:long\_name = Mean liquid-ice static energy in saturated updrafts ;

TLSUP:units = K ;

float32 TASUP(time, z) ;

TASUP:long\_name = Mean TABS in saturated updrafts ;

TASUP:units = K ;

float32 TVSUP(time, z) ;

TVSUP:long\_name = Mean THETAV in saturated updrafts ;

TVSUP:units = K ;

float32 TVSUPA(time, z) ;

TVSUPA:long\_name = Mean THETAV anomaly in saturated updrafts ;

TVSUPA:units = K ;

float32 QTSUP(time, z) ;

QTSUP:long\_name = Mean QT in saturated updrafts ;

QTSUP:units = g/kg ;

float32 QNSUP(time, z) ;

QNSUP:long\_name = Mean QN in saturated updrafts ;

QNSUP:units = g/kg ;

float32 QPSUP(time, z) ;

QPSUP:long\_name = Mean QP in saturated updrafts ;

QPSUP:units = g/kg ;

float32 WSUPA(time, z) ;

WSUPA:long\_name = W in saturated updrafts averaged over the whole domain ;

WSUPA:units = m/s ;

float32 TLWSUP(time, z) ;

TLWSUP:long\_name = TLW in saturated updrafts averaged over the whole domain ;

TLWSUP:units = Km/s ;

float32 TVWSUP(time, z) ;

TVWSUP:long\_name = TVW in saturated updrafts averaged over the whole domain ;

TVWSUP:units = Km/s ;

float32 QTWSUP(time, z) ;

QTWSUP:long\_name = QTW in saturated updrafts averaged over the whole domain ;

QTWSUP:units = g/kg m/s ;

float32 QCWSUP(time, z) ;

QCWSUP:long\_name = QCW in saturated updrafts averaged over the whole domain ;

QCWSUP:units = g/kg m/s ;

float32 QIWSUP(time, z) ;

QIWSUP:long\_name = QIW in saturated updrafts averaged over the whole domain ;

QIWSUP:units = g/kg m/s ;

float32 HFSUP(time, z) ;

HFSUP:long\_name = Mean Frozen MSE in saturated updrafts ;

HFSUP:units = K ;

float32 HFSUPA(time, z) ;

HFSUPA:long\_name = Mean Frozen MSE anomaly in saturated updrafts ;

HFSUPA:units = K ;

float32 USUPA(time, z) ;

USUPA:long\_name = Mean U anomaly in saturated updrafts ;

USUPA:units = m/s ;

float32 VSUPA(time, z) ;

VSUPA:long\_name = Mean V anomaly in saturated updrafts ;

VSUPA:units = m/s ;

float32 UPGFSUP(time, z) ;

UPGFSUP:long\_name = Zonal pressure gradient in saturated updrafts ;

UPGFSUP:units = m/s2 ;

float32 VPGFSUP(time, z) ;

VPGFSUP:long\_name = Meridional pressure gradient in saturated updrafts ;

VPGFSUP:units = m/s2 ;

float32 WPGFSUP(time, z) ;

WPGFSUP:long\_name = Vertical pressure gradient in saturated updrafts ;

WPGFSUP:units = m/s2 ;

float32 UWSUP(time, z) ;

UWSUP:long\_name = UW in saturated updrafts ;

UWSUP:units = m2/s2 ;

float32 VWSUP(time, z) ;

VWSUP:long\_name = VW in saturated updrafts ;

VWSUP:units = m2/s2 ;

float32 UWSBSUP(time, z) ;

UWSBSUP:long\_name = Subgrid UW in saturated updrafts ;

UWSBSUP:units = m2/s2 ;

float32 VWSBSUP(time, z) ;

VWSBSUP:long\_name = Subgrid VW in saturated updrafts ;

VWSBSUP:units = m2/s2 ;

float32 MFSUP(time, z) ;

MFSUP:long\_name = Mass flux in saturated updrafts averaged over the whole domain ;

MFSUP:units = kg/m2/s ;

float32 MFHSUP(time, z) ;

MFHSUP:long\_name = RHO\*W\*HF in saturated updrafts averaged over the whole domain ;

MFHSUP:units = K kg/m2/s ;

float32 MFHSUPA(time, z) ;

MFHSUPA:long\_name = RHO\*W\*HF anomaly in saturated updrafts averaged over the whole domain ;

MFHSUPA:units = K kg/m2/s ;

float32 MFTLSUP(time, z) ;

MFTLSUP:long\_name = RHO\*W\*TL in saturated updrafts averaged over the whole domain ;

MFTLSUP:units = K kg/m2/s ;

float32 MFTLSUPA(time, z) ;

MFTLSUPA:long\_name = RHO\*W\*TL anomaly in saturated updrafts averaged over the whole domain ;

MFTLSUPA:units = K kg/m2/s ;

float32 MFTVSUP(time, z) ;

MFTVSUP:long\_name = RHO\*W\*TV in saturated updrafts averaged over the whole domain ;

MFTVSUP:units = K kg/m2/s ;

float32 MFTVSUPA(time, z) ;

MFTVSUPA:long\_name = RHO\*W\*TV anomaly in saturated updrafts averaged over the whole domain ;

MFTVSUPA:units = K kg/m2/s ;

float32 MFQTSUP(time, z) ;

MFQTSUP:long\_name = RHO\*W\*QT in saturated updrafts averaged over the whole domain ;

MFQTSUP:units = g/m2/s ;

float32 MFQTSUPA(time, z) ;

MFQTSUPA:long\_name = RHO\*W\*QT anomaly in saturated updrafts averaged over the whole domain ;

MFQTSUPA:units = g/m2/s ;

float32 RUWSUP(time, z) ;

RUWSUP:long\_name = RHOUW in saturated updrafts averaged over the whole domain ;

RUWSUP:units = kg/m/s2 ;

float32 RVWSUP(time, z) ;

RVWSUP:long\_name = RHOVW in saturated updrafts averaged over the whole domain ;

RVWSUP:units = kg/m/s2 ;

float32 RWWSUP(time, z) ;

RWWSUP:long\_name = RHOWW in saturated updrafts averaged over the whole domain ;

RWWSUP:units = kg/m/s2 ;

float32 SDN(time, z) ;

SDN:long\_name = saturated downdrafts Fraction ;

SDN:units = ;

float32 WSDN(time, z) ;

WSDN:long\_name = Mean W in saturated downdrafts ;

WSDN:units = m/s ;

float32 USDN(time, z) ;

USDN:long\_name = Mean U in saturated downdrafts ;

USDN:units = m/s ;

float32 VSDN(time, z) ;

VSDN:long\_name = Mean V in saturated downdrafts ;

VSDN:units = m/s ;

float32 MSESDN(time, z) ;

MSESDN:long\_name = Mean moist static energy in saturated downdrafts ;

MSESDN:units = K ;

float32 DSESDN(time, z) ;

DSESDN:long\_name = Mean dry static energy in saturated downdrafts ;

DSESDN:units = K ;

float32 TLSDN(time, z) ;

TLSDN:long\_name = Mean liquid-ice static energy in saturated downdrafts ;

TLSDN:units = K ;

float32 TASDN(time, z) ;

TASDN:long\_name = Mean TABS in saturated downdrafts ;

TASDN:units = K ;

float32 TVSDN(time, z) ;

TVSDN:long\_name = Mean THETAV in saturated downdrafts ;

TVSDN:units = K ;

float32 TVSDNA(time, z) ;

TVSDNA:long\_name = Mean THETAV anomaly in saturated downdrafts ;

TVSDNA:units = K ;

float32 QTSDN(time, z) ;

QTSDN:long\_name = Mean QT in saturated downdrafts ;

QTSDN:units = g/kg ;

float32 QNSDN(time, z) ;

QNSDN:long\_name = Mean QN in saturated downdrafts ;

QNSDN:units = g/kg ;

float32 QPSDN(time, z) ;

QPSDN:long\_name = Mean QP in saturated downdrafts ;

QPSDN:units = g/kg ;

float32 WSDNA(time, z) ;

WSDNA:long\_name = W in saturated downdrafts averaged over the whole domain ;

WSDNA:units = m/s ;

float32 TLWSDN(time, z) ;

TLWSDN:long\_name = TLW in saturated downdrafts averaged over the whole domain ;

TLWSDN:units = Km/s ;

float32 TVWSDN(time, z) ;

TVWSDN:long\_name = TVW in saturated downdrafts averaged over the whole domain ;

TVWSDN:units = Km/s ;

float32 QTWSDN(time, z) ;

QTWSDN:long\_name = QTW in saturated downdrafts averaged over the whole domain ;

QTWSDN:units = g/kg m/s ;

float32 QCWSDN(time, z) ;

QCWSDN:long\_name = QCW in saturated downdrafts averaged over the whole domain ;

QCWSDN:units = g/kg m/s ;

float32 QIWSDN(time, z) ;

QIWSDN:long\_name = QIW in saturated downdrafts averaged over the whole domain ;

QIWSDN:units = g/kg m/s ;

float32 HFSDN(time, z) ;

HFSDN:long\_name = Mean Frozen MSE in saturated downdrafts ;

HFSDN:units = K ;

float32 HFSDNA(time, z) ;

HFSDNA:long\_name = Mean Frozen MSE anomaly in saturated downdrafts ;

HFSDNA:units = K ;

float32 USDNA(time, z) ;

USDNA:long\_name = Mean U anomaly in saturated downdrafts ;

USDNA:units = m/s ;

float32 VSDNA(time, z) ;

VSDNA:long\_name = Mean V anomaly in saturated downdrafts ;

VSDNA:units = m/s ;

float32 UPGFSDN(time, z) ;

UPGFSDN:long\_name = Zonal pressure gradient in saturated downdrafts ;

UPGFSDN:units = m/s2 ;

float32 VPGFSDN(time, z) ;

VPGFSDN:long\_name = Meridional pressure gradient in saturated downdrafts ;

VPGFSDN:units = m/s2 ;

float32 WPGFSDN(time, z) ;

WPGFSDN:long\_name = Vertical pressure gradient in saturated downdrafts ;

WPGFSDN:units = m/s2 ;

float32 UWSDN(time, z) ;

UWSDN:long\_name = UW in saturated downdrafts ;

UWSDN:units = m2/s2 ;

float32 VWSDN(time, z) ;

VWSDN:long\_name = VW in saturated downdrafts ;

VWSDN:units = m2/s2 ;

float32 UWSBSDN(time, z) ;

UWSBSDN:long\_name = Subgrid UW in saturated downdrafts ;

UWSBSDN:units = m2/s2 ;

float32 VWSBSDN(time, z) ;

VWSBSDN:long\_name = Subgrid VW in saturated downdrafts ;

VWSBSDN:units = m2/s2 ;

float32 MFSDN(time, z) ;

MFSDN:long\_name = Mass flux in saturated downdrafts averaged over the whole domain ;

MFSDN:units = kg/m2/s ;

float32 MFHSDN(time, z) ;

MFHSDN:long\_name = RHO\*W\*HF in saturated downdrafts averaged over the whole domain ;

MFHSDN:units = K kg/m2/s ;

float32 MFHSDNA(time, z) ;

MFHSDNA:long\_name = RHO\*W\*HF anomaly in saturated downdrafts averaged over the whole domain ;

MFHSDNA:units = K kg/m2/s ;

float32 MFTLSDN(time, z) ;

MFTLSDN:long\_name = RHO\*W\*TL in saturated downdrafts averaged over the whole domain ;

MFTLSDN:units = K kg/m2/s ;

float32 MFTLSDNA(time, z) ;

MFTLSDNA:long\_name = RHO\*W\*TL anomaly in saturated downdrafts averaged over the whole domain ;

MFTLSDNA:units = K kg/m2/s ;

float32 MFTVSDN(time, z) ;

MFTVSDN:long\_name = RHO\*W\*TV in saturated downdrafts averaged over the whole domain ;

MFTVSDN:units = K kg/m2/s ;

float32 MFTVSDNA(time, z) ;

MFTVSDNA:long\_name = RHO\*W\*TV anomaly in saturated downdrafts averaged over the whole domain ;

MFTVSDNA:units = K kg/m2/s ;

float32 MFQTSDN(time, z) ;

MFQTSDN:long\_name = RHO\*W\*QT in saturated downdrafts averaged over the whole domain ;

MFQTSDN:units = g/m2/s ;

float32 MFQTSDNA(time, z) ;

MFQTSDNA:long\_name = RHO\*W\*QT anomaly in saturated downdrafts averaged over the whole domain ;

MFQTSDNA:units = g/m2/s ;

float32 RUWSDN(time, z) ;

RUWSDN:long\_name = RHOUW in saturated downdrafts averaged over the whole domain ;

RUWSDN:units = kg/m/s2 ;

float32 RVWSDN(time, z) ;

RVWSDN:long\_name = RHOVW in saturated downdrafts averaged over the whole domain ;

RVWSDN:units = kg/m/s2 ;

float32 RWWSDN(time, z) ;

RWWSDN:long\_name = RHOWW in saturated downdrafts averaged over the whole domain ;

RWWSDN:units = kg/m/s2 ;

float32 ENV(time, z) ;

ENV:long\_name = unsaturated environment Fraction ;

ENV:units = ;

float32 WENV(time, z) ;

WENV:long\_name = Mean W in unsaturated environment ;

WENV:units = m/s ;

float32 UENV(time, z) ;

UENV:long\_name = Mean U in unsaturated environment ;

UENV:units = m/s ;

float32 VENV(time, z) ;

VENV:long\_name = Mean V in unsaturated environment ;

VENV:units = m/s ;

float32 MSEENV(time, z) ;

MSEENV:long\_name = Mean moist static energy in unsaturated environment ;

MSEENV:units = K ;

float32 DSEENV(time, z) ;

DSEENV:long\_name = Mean dry static energy in unsaturated environment ;

DSEENV:units = K ;

float32 TLENV(time, z) ;

TLENV:long\_name = Mean liquid-ice static energy in unsaturated environment ;

TLENV:units = K ;

float32 TAENV(time, z) ;

TAENV:long\_name = Mean TABS in unsaturated environment ;

TAENV:units = K ;

float32 TVENV(time, z) ;

TVENV:long\_name = Mean THETAV in unsaturated environment ;

TVENV:units = K ;

float32 TVENVA(time, z) ;

TVENVA:long\_name = Mean THETAV anomaly in unsaturated environment ;

TVENVA:units = K ;

float32 QTENV(time, z) ;

QTENV:long\_name = Mean QT in unsaturated environment ;

QTENV:units = g/kg ;

float32 QNENV(time, z) ;

QNENV:long\_name = Mean QN in unsaturated environment ;

QNENV:units = g/kg ;

float32 QPENV(time, z) ;

QPENV:long\_name = Mean QP in unsaturated environment ;

QPENV:units = g/kg ;

float32 WENVA(time, z) ;

WENVA:long\_name = W in unsaturated environment averaged over the whole domain ;

WENVA:units = m/s ;

float32 TLWENV(time, z) ;

TLWENV:long\_name = TLW in unsaturated environment averaged over the whole domain ;

TLWENV:units = Km/s ;

float32 TVWENV(time, z) ;

TVWENV:long\_name = TVW in unsaturated environment averaged over the whole domain ;

TVWENV:units = Km/s ;

float32 QTWENV(time, z) ;

QTWENV:long\_name = QTW in unsaturated environment averaged over the whole domain ;

QTWENV:units = g/kg m/s ;

float32 QCWENV(time, z) ;

QCWENV:long\_name = QCW in unsaturated environment averaged over the whole domain ;

QCWENV:units = g/kg m/s ;

float32 QIWENV(time, z) ;

QIWENV:long\_name = QIW in unsaturated environment averaged over the whole domain ;

QIWENV:units = g/kg m/s ;

float32 HFENV(time, z) ;

HFENV:long\_name = Mean Frozen MSE in unsaturated environment ;

HFENV:units = K ;

float32 HFENVA(time, z) ;

HFENVA:long\_name = Mean Frozen MSE anomaly in unsaturated environment ;

HFENVA:units = K ;

float32 UENVA(time, z) ;

UENVA:long\_name = Mean U anomaly in unsaturated environment ;

UENVA:units = m/s ;

float32 VENVA(time, z) ;

VENVA:long\_name = Mean V anomaly in unsaturated environment ;

VENVA:units = m/s ;

float32 UPGFENV(time, z) ;

UPGFENV:long\_name = Zonal pressure gradient in unsaturated environment ;

UPGFENV:units = m/s2 ;

float32 VPGFENV(time, z) ;

VPGFENV:long\_name = Meridional pressure gradient in unsaturated environment ;

VPGFENV:units = m/s2 ;

float32 WPGFENV(time, z) ;

WPGFENV:long\_name = Vertical pressure gradient in unsaturated environment ;

WPGFENV:units = m/s2 ;

float32 UWENV(time, z) ;

UWENV:long\_name = UW in unsaturated environment ;

UWENV:units = m2/s2 ;

float32 VWENV(time, z) ;

VWENV:long\_name = VW in unsaturated environment ;

VWENV:units = m2/s2 ;

float32 UWSBENV(time, z) ;

UWSBENV:long\_name = Subgrid UW in unsaturated environment ;

UWSBENV:units = m2/s2 ;

float32 VWSBENV(time, z) ;

VWSBENV:long\_name = Subgrid VW in unsaturated environment ;

VWSBENV:units = m2/s2 ;

float32 MFENV(time, z) ;

MFENV:long\_name = Mass flux in unsaturated environment averaged over the whole domain ;

MFENV:units = kg/m2/s ;

float32 MFHENV(time, z) ;

MFHENV:long\_name = RHO\*W\*HF in unsaturated environment averaged over the whole domain ;

MFHENV:units = K kg/m2/s ;

float32 MFHENVA(time, z) ;

MFHENVA:long\_name = RHO\*W\*HF anomaly in unsaturated environment averaged over the whole domain ;

MFHENVA:units = K kg/m2/s ;

float32 MFTLENV(time, z) ;

MFTLENV:long\_name = RHO\*W\*TL in unsaturated environment averaged over the whole domain ;

MFTLENV:units = K kg/m2/s ;

float32 MFTLENVA(time, z) ;

MFTLENVA:long\_name = RHO\*W\*TL anomaly in unsaturated environment averaged over the whole domain ;

MFTLENVA:units = K kg/m2/s ;

float32 MFTVENV(time, z) ;

MFTVENV:long\_name = RHO\*W\*TV in unsaturated environment averaged over the whole domain ;

MFTVENV:units = K kg/m2/s ;

float32 MFTVENVA(time, z) ;

MFTVENVA:long\_name = RHO\*W\*TV anomaly in unsaturated environment averaged over the whole domain ;

MFTVENVA:units = K kg/m2/s ;

float32 MFQTENV(time, z) ;

MFQTENV:long\_name = RHO\*W\*QT in unsaturated environment averaged over the whole domain ;

MFQTENV:units = g/m2/s ;

float32 MFQTENVA(time, z) ;

MFQTENVA:long\_name = RHO\*W\*QT anomaly in unsaturated environment averaged over the whole domain ;

MFQTENVA:units = g/m2/s ;

float32 RUWENV(time, z) ;

RUWENV:long\_name = RHOUW in unsaturated environment averaged over the whole domain ;

RUWENV:units = kg/m/s2 ;

float32 RVWENV(time, z) ;

RVWENV:long\_name = RHOVW in unsaturated environment averaged over the whole domain ;

RVWENV:units = kg/m/s2 ;

float32 RWWENV(time, z) ;

RWWENV:long\_name = RHOWW in unsaturated environment averaged over the whole domain ;

RWWENV:units = kg/m/s2 ;

float32 QTFLUX(time, z) ;

QTFLUX:long\_name = Total (resolved + subgrid) total water (vapor+cloud) flux ;

QTFLUX:units = W/m2 ;

float32 QTO(time, z) ;

QTO:long\_name = TOTAL WATER (VAPOR + CLOUD LIQUID) ;

QTO:units = g/kg ;

float32 QTO2(time, z) ;

QTO2:long\_name = Variance of TOTAL WATER (VAPOR + CLOUD LIQUID) ;

QTO2:units = (g/kg)^2 ;

float32 QTOADV(time, z) ;

QTOADV:long\_name = Tendency of TOTAL WATER (VAPOR + CLOUD LIQUID) due to resolved vertical advectio ;

QTOADV:units = g/kg/day ;

float32 QTODIFF(time, z) ;

QTODIFF:long\_name = Tendency of TOTAL WATER (VAPOR + CLOUD LIQUID) due to vertical SGS transport ;

QTODIFF:units = g/kg/day ;

float32 QTOLSADV(time, z) ;

QTOLSADV:long\_name = Tendency of TOTAL WATER (VAPOR + CLOUD LIQUID) due to large-scale vertical advec ;

QTOLSADV:units = g/kg/day ;

float32 QTOMPHY(time, z) ;

QTOMPHY:long\_name = Tendency of TOTAL WATER (VAPOR + CLOUD LIQUID) due to microphysical processes ;

QTOMPHY:units = g/kg/day ;

float32 QTOSED(time, z) ;

QTOSED:long\_name = Tendency of TOTAL WATER (VAPOR + CLOUD LIQUID) due to sedimentation ;

QTOSED:units = g/kg/day ;

float32 QTOFLXR(time, z) ;

QTOFLXR:long\_name = Resolved flux of TOTAL WATER (VAPOR + CLOUD LIQUID) ;

QTOFLXR:units = W/m2 ;

float32 QTOFLXS(time, z) ;

QTOFLXS:long\_name = Subgrid flux of TOTAL WATER (VAPOR + CLOUD LIQUID) ;

QTOFLXS:units = W/m2 ;

float32 QTOSDFLX(time, z) ;

QTOSDFLX:long\_name = Sedimentation flux of TOTAL WATER (VAPOR + CLOUD LIQUID) ;

QTOSDFLX:units = W/m2 ;

float32 QR(time, z) ;

QR:long\_name = RAIN ;

QR:units = g/kg ;

float32 QRADV(time, z) ;

QRADV:long\_name = Tendency of RAIN due to resolved vertical advection ;

QRADV:units = g/kg/day ;

float32 QRDIFF(time, z) ;

QRDIFF:long\_name = Tendency of RAIN due to vertical SGS transport ;

QRDIFF:units = g/kg/day ;

float32 QRLSADV(time, z) ;

QRLSADV:long\_name = Tendency of RAIN due to large-scale vertical advection ;

QRLSADV:units = g/kg/day ;

float32 QRMPHY(time, z) ;

QRMPHY:long\_name = Tendency of RAIN due to microphysical processes ;

QRMPHY:units = g/kg/day ;

float32 QRSED(time, z) ;

QRSED:long\_name = Tendency of RAIN due to sedimentation ;

QRSED:units = g/kg/day ;

float32 QRFLXR(time, z) ;

QRFLXR:long\_name = Resolved flux of RAIN ;

QRFLXR:units = W/m2 ;

float32 QRFLXS(time, z) ;

QRFLXS:long\_name = Subgrid flux of RAIN ;

QRFLXS:units = W/m2 ;

float32 QRSDFLX(time, z) ;

QRSDFLX:long\_name = Sedimentation flux of RAIN ;

QRSDFLX:units = W/m2 ;

float32 QRFRAC(time, z) ;

QRFRAC:long\_name = RAIN FRACTION ;

QRFRAC:units = 1 ;

float32 TAUQR(time, z) ;

TAUQR:long\_name = Approx optical depth of RAIN ;

TAUQR:units = 1 ;

float32 QROEFFR(time, z) ;

QROEFFR:long\_name = Mixing ratio of RAIN over effective radius, EFFR = QR/QROEFFR ;

QROEFFR:units = g/kg/micro ;

float32 NR(time, z) ;

NR:long\_name = RAIN NUMBER CONCENTRATION ;

NR:units = #/cm3 ;

float32 NRADV(time, z) ;

NRADV:long\_name = Tendency of RAIN NUMBER CONCENTRATION due to resolved vertical advection ;

NRADV:units = #/cm3/day ;

float32 NRDIFF(time, z) ;

NRDIFF:long\_name = Tendency of RAIN NUMBER CONCENTRATION due to vertical SGS transport ;

NRDIFF:units = #/cm3/day ;

float32 NRLSADV(time, z) ;

NRLSADV:long\_name = Tendency of RAIN NUMBER CONCENTRATION due to large-scale vertical advection ;

NRLSADV:units = #/cm3/day ;

float32 NRMPHY(time, z) ;

NRMPHY:long\_name = Tendency of RAIN NUMBER CONCENTRATION due to microphysical processes ;

NRMPHY:units = #/cm3/day ;

float32 NRSED(time, z) ;

NRSED:long\_name = Tendency of RAIN NUMBER CONCENTRATION due to sedimentation ;

NRSED:units = #/cm3/day ;

float32 NRFLXR(time, z) ;

NRFLXR:long\_name = Resolved flux of RAIN NUMBER CONCENTRATION ;

NRFLXR:units = #/m2/s ;

float32 NRFLXS(time, z) ;

NRFLXS:long\_name = Subgrid flux of RAIN NUMBER CONCENTRATION ;

NRFLXS:units = #/m2/s ;

float32 NRSDFLX(time, z) ;

NRSDFLX:long\_name = Sedimentation flux of RAIN NUMBER CONCENTRATION ;

NRSDFLX:units = #/m2/s ;

float32 QI(time, z) ;

QI:long\_name = CLOUD ICE ;

QI:units = g/kg ;

float32 QIADV(time, z) ;

QIADV:long\_name = Tendency of CLOUD ICE due to resolved vertical advection ;

QIADV:units = g/kg/day ;

float32 QIDIFF(time, z) ;

QIDIFF:long\_name = Tendency of CLOUD ICE due to vertical SGS transport ;

QIDIFF:units = g/kg/day ;

float32 QILSADV(time, z) ;

QILSADV:long\_name = Tendency of CLOUD ICE due to large-scale vertical advection ;

QILSADV:units = g/kg/day ;

float32 QIMPHY(time, z) ;

QIMPHY:long\_name = Tendency of CLOUD ICE due to microphysical processes ;

QIMPHY:units = g/kg/day ;

float32 QISED(time, z) ;

QISED:long\_name = Tendency of CLOUD ICE due to sedimentation ;

QISED:units = g/kg/day ;

float32 QIFLXR(time, z) ;

QIFLXR:long\_name = Resolved flux of CLOUD ICE ;

QIFLXR:units = W/m2 ;

float32 QIFLXS(time, z) ;

QIFLXS:long\_name = Subgrid flux of CLOUD ICE ;

QIFLXS:units = W/m2 ;

float32 QISDFLX(time, z) ;

QISDFLX:long\_name = Sedimentation flux of CLOUD ICE ;

QISDFLX:units = W/m2 ;

float32 QIFRAC(time, z) ;

QIFRAC:long\_name = CLOUD ICE FRACTION ;

QIFRAC:units = 1 ;

float32 TAUQI(time, z) ;

TAUQI:long\_name = Approx optical depth of CLOUD ICE ;

TAUQI:units = 1 ;

float32 QIOEFFR(time, z) ;

QIOEFFR:long\_name = Mixing ratio of CLOUD ICE over effective radius, EFFR = QI/QIOEFFR ;

QIOEFFR:units = g/kg/micro ;

float32 NI(time, z) ;

NI:long\_name = CLOUD ICE NUMBER CONCENTRATION ;

NI:units = #/cm3 ;

float32 NIADV(time, z) ;

NIADV:long\_name = Tendency of CLOUD ICE NUMBER CONCENTRATION due to resolved vertical advection ;

NIADV:units = #/cm3/day ;

float32 NIDIFF(time, z) ;

NIDIFF:long\_name = Tendency of CLOUD ICE NUMBER CONCENTRATION due to vertical SGS transport ;

NIDIFF:units = #/cm3/day ;

float32 NILSADV(time, z) ;

NILSADV:long\_name = Tendency of CLOUD ICE NUMBER CONCENTRATION due to large-scale vertical advection ;

NILSADV:units = #/cm3/day ;

float32 NIMPHY(time, z) ;

NIMPHY:long\_name = Tendency of CLOUD ICE NUMBER CONCENTRATION due to microphysical processes ;

NIMPHY:units = #/cm3/day ;

float32 NISED(time, z) ;

NISED:long\_name = Tendency of CLOUD ICE NUMBER CONCENTRATION due to sedimentation ;

NISED:units = #/cm3/day ;

float32 NIFLXR(time, z) ;

NIFLXR:long\_name = Resolved flux of CLOUD ICE NUMBER CONCENTRATION ;

NIFLXR:units = #/m2/s ;

float32 NIFLXS(time, z) ;

NIFLXS:long\_name = Subgrid flux of CLOUD ICE NUMBER CONCENTRATION ;

NIFLXS:units = #/m2/s ;

float32 NISDFLX(time, z) ;

NISDFLX:long\_name = Sedimentation flux of CLOUD ICE NUMBER CONCENTRATION ;

NISDFLX:units = #/m2/s ;

float32 QS(time, z) ;

QS:long\_name = SNOW ;

QS:units = g/kg ;

float32 QSADV(time, z) ;

QSADV:long\_name = Tendency of SNOW due to resolved vertical advection ;

QSADV:units = g/kg/day ;

float32 QSDIFF(time, z) ;

QSDIFF:long\_name = Tendency of SNOW due to vertical SGS transport ;

QSDIFF:units = g/kg/day ;

float32 QSLSADV(time, z) ;

QSLSADV:long\_name = Tendency of SNOW due to large-scale vertical advection ;

QSLSADV:units = g/kg/day ;

float32 QSMPHY(time, z) ;

QSMPHY:long\_name = Tendency of SNOW due to microphysical processes ;

QSMPHY:units = g/kg/day ;

float32 QSSED(time, z) ;

QSSED:long\_name = Tendency of SNOW due to sedimentation ;

QSSED:units = g/kg/day ;

float32 QSFLXR(time, z) ;

QSFLXR:long\_name = Resolved flux of SNOW ;

QSFLXR:units = W/m2 ;

float32 QSFLXS(time, z) ;

QSFLXS:long\_name = Subgrid flux of SNOW ;

QSFLXS:units = W/m2 ;

float32 QSSDFLX(time, z) ;

QSSDFLX:long\_name = Sedimentation flux of SNOW ;

QSSDFLX:units = W/m2 ;

float32 QSFRAC(time, z) ;

QSFRAC:long\_name = SNOW FRACTION ;

QSFRAC:units = 1 ;

float32 TAUQS(time, z) ;

TAUQS:long\_name = Approx optical depth of SNOW ;

TAUQS:units = 1 ;

float32 QSOEFFR(time, z) ;

QSOEFFR:long\_name = Mixing ratio of SNOW over effective radius, EFFR = QS/QSOEFFR ;

QSOEFFR:units = g/kg/micro ;

float32 NS(time, z) ;

NS:long\_name = SNOW NUMBER CONCENTRATION ;

NS:units = #/cm3 ;

float32 NSADV(time, z) ;

NSADV:long\_name = Tendency of SNOW NUMBER CONCENTRATION due to resolved vertical advection ;

NSADV:units = #/cm3/day ;

float32 NSDIFF(time, z) ;

NSDIFF:long\_name = Tendency of SNOW NUMBER CONCENTRATION due to vertical SGS transport ;

NSDIFF:units = #/cm3/day ;

float32 NSLSADV(time, z) ;

NSLSADV:long\_name = Tendency of SNOW NUMBER CONCENTRATION due to large-scale vertical advection ;

NSLSADV:units = #/cm3/day ;

float32 NSMPHY(time, z) ;

NSMPHY:long\_name = Tendency of SNOW NUMBER CONCENTRATION due to microphysical processes ;

NSMPHY:units = #/cm3/day ;

float32 NSSED(time, z) ;

NSSED:long\_name = Tendency of SNOW NUMBER CONCENTRATION due to sedimentation ;

NSSED:units = #/cm3/day ;

float32 NSFLXR(time, z) ;

NSFLXR:long\_name = Resolved flux of SNOW NUMBER CONCENTRATION ;

NSFLXR:units = #/m2/s ;

float32 NSFLXS(time, z) ;

NSFLXS:long\_name = Subgrid flux of SNOW NUMBER CONCENTRATION ;

NSFLXS:units = #/m2/s ;

float32 NSSDFLX(time, z) ;

NSSDFLX:long\_name = Sedimentation flux of SNOW NUMBER CONCENTRATION ;

NSSDFLX:units = #/m2/s ;

float32 QG(time, z) ;

QG:long\_name = GRAUPEL ;

QG:units = g/kg ;

float32 QGADV(time, z) ;

QGADV:long\_name = Tendency of GRAUPEL due to resolved vertical advection ;

QGADV:units = g/kg/day ;

float32 QGDIFF(time, z) ;

QGDIFF:long\_name = Tendency of GRAUPEL due to vertical SGS transport ;

QGDIFF:units = g/kg/day ;

float32 QGLSADV(time, z) ;

QGLSADV:long\_name = Tendency of GRAUPEL due to large-scale vertical advection ;

QGLSADV:units = g/kg/day ;

float32 QGMPHY(time, z) ;

QGMPHY:long\_name = Tendency of GRAUPEL due to microphysical processes ;

QGMPHY:units = g/kg/day ;

float32 QGSED(time, z) ;

QGSED:long\_name = Tendency of GRAUPEL due to sedimentation ;

QGSED:units = g/kg/day ;

float32 QGFLXR(time, z) ;

QGFLXR:long\_name = Resolved flux of GRAUPEL ;

QGFLXR:units = W/m2 ;

float32 QGFLXS(time, z) ;

QGFLXS:long\_name = Subgrid flux of GRAUPEL ;

QGFLXS:units = W/m2 ;

float32 QGSDFLX(time, z) ;

QGSDFLX:long\_name = Sedimentation flux of GRAUPEL ;

QGSDFLX:units = W/m2 ;

float32 QGFRAC(time, z) ;

QGFRAC:long\_name = GRAUPEL FRACTION ;

QGFRAC:units = 1 ;

float32 TAUQG(time, z) ;

TAUQG:long\_name = Approx optical depth of GRAUPEL ;

TAUQG:units = 1 ;

float32 QGOEFFR(time, z) ;

QGOEFFR:long\_name = Mixing ratio of GRAUPEL over effective radius, EFFR = QG/QGOEFFR ;

QGOEFFR:units = g/kg/micro ;

float32 NG(time, z) ;

NG:long\_name = GRAUPEL NUMBER CONCENTRATION ;

NG:units = #/cm3 ;

float32 NGADV(time, z) ;

NGADV:long\_name = Tendency of GRAUPEL NUMBER CONCENTRATION due to resolved vertical advection ;

NGADV:units = #/cm3/day ;

float32 NGDIFF(time, z) ;

NGDIFF:long\_name = Tendency of GRAUPEL NUMBER CONCENTRATION due to vertical SGS transport ;

NGDIFF:units = #/cm3/day ;

float32 NGLSADV(time, z) ;

NGLSADV:long\_name = Tendency of GRAUPEL NUMBER CONCENTRATION due to large-scale vertical advection ;

NGLSADV:units = #/cm3/day ;

float32 NGMPHY(time, z) ;

NGMPHY:long\_name = Tendency of GRAUPEL NUMBER CONCENTRATION due to microphysical processes ;

NGMPHY:units = #/cm3/day ;

float32 NGSED(time, z) ;

NGSED:long\_name = Tendency of GRAUPEL NUMBER CONCENTRATION due to sedimentation ;

NGSED:units = #/cm3/day ;

float32 NGFLXR(time, z) ;

NGFLXR:long\_name = Resolved flux of GRAUPEL NUMBER CONCENTRATION ;

NGFLXR:units = #/m2/s ;

float32 NGFLXS(time, z) ;

NGFLXS:long\_name = Subgrid flux of GRAUPEL NUMBER CONCENTRATION ;

NGFLXS:units = #/m2/s ;

float32 NGSDFLX(time, z) ;

NGSDFLX:long\_name = Sedimentation flux of GRAUPEL NUMBER CONCENTRATION ;

NGSDFLX:units = #/m2/s ;

float32 QC(time, z) ;

QC:long\_name = Cloud liquid water mass mixing ratio ;

QC:units = g/kg ;

float32 TAUQC(time, z) ;

TAUQC:long\_name = Approx optical depth of cloud liquid water ;

TAUQC:units = 1 ;

float32 QCOEFFR(time, z) ;

QCOEFFR:long\_name = Mixing ratio of QC over effective radius, EFFR = QC/QCOEFFR ;

QCOEFFR:units = g/kg/micro ;

float32 QVCLD(time, z) ;

QVCLD:long\_name = Water vapor mixing ratio in cloud ;

QVCLD:units = kg/kg ;

float32 QCCLD(time, z) ;

QCCLD:long\_name = Cloud liquid water mixing ratio in cloud ;

QCCLD:units = kg/kg ;

float32 QTOCLD(time, z) ;

QTOCLD:long\_name = TOTAL WATER (VAPOR + CLOUD LIQUID) in cloud ;

QTOCLD:units = g/kg ;

float32 QRCLD(time, z) ;

QRCLD:long\_name = RAIN in cloud ;

QRCLD:units = g/kg ;

float32 NRCLD(time, z) ;

NRCLD:long\_name = RAIN NUMBER CONCENTRATION in cloud ;

NRCLD:units = #/cm3 ;

float32 QICLD(time, z) ;

QICLD:long\_name = CLOUD ICE in cloud ;

QICLD:units = g/kg ;

float32 NICLD(time, z) ;

NICLD:long\_name = CLOUD ICE NUMBER CONCENTRATION in cloud ;

NICLD:units = #/cm3 ;

float32 QSCLD(time, z) ;

QSCLD:long\_name = SNOW in cloud ;

QSCLD:units = g/kg ;

float32 NSCLD(time, z) ;

NSCLD:long\_name = SNOW NUMBER CONCENTRATION in cloud ;

NSCLD:units = #/cm3 ;

float32 QGCLD(time, z) ;

QGCLD:long\_name = GRAUPEL in cloud ;

QGCLD:units = g/kg ;

float32 NGCLD(time, z) ;

NGCLD:long\_name = GRAUPEL NUMBER CONCENTRATION in cloud ;

NGCLD:units = #/cm3 ;

float32 QVCOR(time, z) ;

QVCOR:long\_name = Water vapor mixing ratio in core ;

QVCOR:units = kg/kg ;

float32 QCCOR(time, z) ;

QCCOR:long\_name = Cloud liquid water mixing ratio in core ;

QCCOR:units = kg/kg ;

float32 QTOCOR(time, z) ;

QTOCOR:long\_name = TOTAL WATER (VAPOR + CLOUD LIQUID) in core ;

QTOCOR:units = g/kg ;

float32 QRCOR(time, z) ;

QRCOR:long\_name = RAIN in core ;

QRCOR:units = g/kg ;

float32 NRCOR(time, z) ;

NRCOR:long\_name = RAIN NUMBER CONCENTRATION in core ;

NRCOR:units = #/cm3 ;

float32 QICOR(time, z) ;

QICOR:long\_name = CLOUD ICE in core ;

QICOR:units = g/kg ;

float32 NICOR(time, z) ;

NICOR:long\_name = CLOUD ICE NUMBER CONCENTRATION in core ;

NICOR:units = #/cm3 ;

float32 QSCOR(time, z) ;

QSCOR:long\_name = SNOW in core ;

QSCOR:units = g/kg ;

float32 NSCOR(time, z) ;

NSCOR:long\_name = SNOW NUMBER CONCENTRATION in core ;

NSCOR:units = #/cm3 ;

float32 QGCOR(time, z) ;

QGCOR:long\_name = GRAUPEL in core ;

QGCOR:units = g/kg ;

float32 NGCOR(time, z) ;

NGCOR:long\_name = GRAUPEL NUMBER CONCENTRATION in core ;

NGCOR:units = #/cm3 ;

float32 QVCDN(time, z) ;

QVCDN:long\_name = Water vapor mixing ratio in downdraft core ;

QVCDN:units = kg/kg ;

float32 QCCDN(time, z) ;

QCCDN:long\_name = Cloud liquid water mixing ratio in downdraft core ;

QCCDN:units = kg/kg ;

float32 QTOCDN(time, z) ;

QTOCDN:long\_name = TOTAL WATER (VAPOR + CLOUD LIQUID) in downdraft core ;

QTOCDN:units = g/kg ;

float32 QRCDN(time, z) ;

QRCDN:long\_name = RAIN in downdraft core ;

QRCDN:units = g/kg ;

float32 NRCDN(time, z) ;

NRCDN:long\_name = RAIN NUMBER CONCENTRATION in downdraft core ;

NRCDN:units = #/cm3 ;

float32 QICDN(time, z) ;

QICDN:long\_name = CLOUD ICE in downdraft core ;

QICDN:units = g/kg ;

float32 NICDN(time, z) ;

NICDN:long\_name = CLOUD ICE NUMBER CONCENTRATION in downdraft core ;

NICDN:units = #/cm3 ;

float32 QSCDN(time, z) ;

QSCDN:long\_name = SNOW in downdraft core ;

QSCDN:units = g/kg ;

float32 NSCDN(time, z) ;

NSCDN:long\_name = SNOW NUMBER CONCENTRATION in downdraft core ;

NSCDN:units = #/cm3 ;

float32 QGCDN(time, z) ;

QGCDN:long\_name = GRAUPEL in downdraft core ;

QGCDN:units = g/kg ;

float32 NGCDN(time, z) ;

NGCDN:long\_name = GRAUPEL NUMBER CONCENTRATION in downdraft core ;

NGCDN:units = #/cm3 ;

float32 QVSUP(time, z) ;

QVSUP:long\_name = Water vapor mixing ratio in saturated updrafts ;

QVSUP:units = kg/kg ;

float32 QCSUP(time, z) ;

QCSUP:long\_name = Cloud liquid water mixing ratio in saturated updrafts ;

QCSUP:units = kg/kg ;

float32 QTOSUP(time, z) ;

QTOSUP:long\_name = TOTAL WATER (VAPOR + CLOUD LIQUID) in saturated updrafts ;

QTOSUP:units = g/kg ;

float32 QRSUP(time, z) ;

QRSUP:long\_name = RAIN in saturated updrafts ;

QRSUP:units = g/kg ;

float32 NRSUP(time, z) ;

NRSUP:long\_name = RAIN NUMBER CONCENTRATION in saturated updrafts ;

NRSUP:units = #/cm3 ;

float32 QISUP(time, z) ;

QISUP:long\_name = CLOUD ICE in saturated updrafts ;

QISUP:units = g/kg ;

float32 NISUP(time, z) ;

NISUP:long\_name = CLOUD ICE NUMBER CONCENTRATION in saturated updrafts ;

NISUP:units = #/cm3 ;

float32 QSSUP(time, z) ;

QSSUP:long\_name = SNOW in saturated updrafts ;

QSSUP:units = g/kg ;

float32 NSSUP(time, z) ;

NSSUP:long\_name = SNOW NUMBER CONCENTRATION in saturated updrafts ;

NSSUP:units = #/cm3 ;

float32 QGSUP(time, z) ;

QGSUP:long\_name = GRAUPEL in saturated updrafts ;

QGSUP:units = g/kg ;

float32 NGSUP(time, z) ;

NGSUP:long\_name = GRAUPEL NUMBER CONCENTRATION in saturated updrafts ;

NGSUP:units = #/cm3 ;

float32 QVSDN(time, z) ;

QVSDN:long\_name = Water vapor mixing ratio in saturated downdrafts ;

QVSDN:units = kg/kg ;

float32 QCSDN(time, z) ;

QCSDN:long\_name = Cloud liquid water mixing ratio in saturated downdrafts ;

QCSDN:units = kg/kg ;

float32 QTOSDN(time, z) ;

QTOSDN:long\_name = TOTAL WATER (VAPOR + CLOUD LIQUID) in saturated downdrafts ;

QTOSDN:units = g/kg ;

float32 QRSDN(time, z) ;

QRSDN:long\_name = RAIN in saturated downdrafts ;

QRSDN:units = g/kg ;

float32 NRSDN(time, z) ;

NRSDN:long\_name = RAIN NUMBER CONCENTRATION in saturated downdrafts ;

NRSDN:units = #/cm3 ;

float32 QISDN(time, z) ;

QISDN:long\_name = CLOUD ICE in saturated downdrafts ;

QISDN:units = g/kg ;

float32 NISDN(time, z) ;

NISDN:long\_name = CLOUD ICE NUMBER CONCENTRATION in saturated downdrafts ;

NISDN:units = #/cm3 ;

float32 QSSDN(time, z) ;

QSSDN:long\_name = SNOW in saturated downdrafts ;

QSSDN:units = g/kg ;

float32 NSSDN(time, z) ;

NSSDN:long\_name = SNOW NUMBER CONCENTRATION in saturated downdrafts ;

NSSDN:units = #/cm3 ;

float32 QGSDN(time, z) ;

QGSDN:long\_name = GRAUPEL in saturated downdrafts ;

QGSDN:units = g/kg ;

float32 NGSDN(time, z) ;

NGSDN:long\_name = GRAUPEL NUMBER CONCENTRATION in saturated downdrafts ;

NGSDN:units = #/cm3 ;

float32 QVENV(time, z) ;

QVENV:long\_name = Water vapor mixing ratio in unsaturated environment ;

QVENV:units = kg/kg ;

float32 QCENV(time, z) ;

QCENV:long\_name = Cloud liquid water mixing ratio in unsaturated environment ;

QCENV:units = kg/kg ;

float32 QTOENV(time, z) ;

QTOENV:long\_name = TOTAL WATER (VAPOR + CLOUD LIQUID) in unsaturated environment ;

QTOENV:units = g/kg ;

float32 QRENV(time, z) ;

QRENV:long\_name = RAIN in unsaturated environment ;

QRENV:units = g/kg ;

float32 NRENV(time, z) ;

NRENV:long\_name = RAIN NUMBER CONCENTRATION in unsaturated environment ;

NRENV:units = #/cm3 ;

float32 QIENV(time, z) ;

QIENV:long\_name = CLOUD ICE in unsaturated environment ;

QIENV:units = g/kg ;

float32 NIENV(time, z) ;

NIENV:long\_name = CLOUD ICE NUMBER CONCENTRATION in unsaturated environment ;

NIENV:units = #/cm3 ;

float32 QSENV(time, z) ;

QSENV:long\_name = SNOW in unsaturated environment ;

QSENV:units = g/kg ;

float32 NSENV(time, z) ;

NSENV:long\_name = SNOW NUMBER CONCENTRATION in unsaturated environment ;

NSENV:units = #/cm3 ;

float32 QGENV(time, z) ;

QGENV:long\_name = GRAUPEL in unsaturated environment ;

QGENV:units = g/kg ;

float32 NGENV(time, z) ;

NGENV:long\_name = GRAUPEL NUMBER CONCENTRATION in unsaturated environment ;

NGENV:units = #/cm3 ;

// global attributes:

:SAM version = ;

:caseid = 382x576x64\_250m\_5s\_213d\_20d\_edu ;

}None