# Section 5

# SUBROUTINES

In this section, subroutines are presented in alphabetical order and are described following a standard format. A short general description of the subroutine is given first followed by a list of the routines it is called from and any routines it calls. Arguments are then listed in the order they appear in the code with input variables shown in **UPPER CASE** and output variables in **lower case**. Arguments that serve as both input and output are shown in lower case appended with an asterisk, and those that are only used internally (neither input or output) are [**BRACKETED**]. Definitions of the arguments are given in section 6. Common blocks used by the subroutine are listed next, followed by definition of any data in DATA statements. Finally, a detailed description is provided for subroutines that perform many significant tasks, and usage notes are included for a few routines.

## 5.1 ADDALL

Subroutine ADDALL presets pointer addresses of all variables on all domains in common blocks associated with pointers and puts them into the **IAXALL (NUMVAR, MAXNES)** array.

ADDALL is called from the main program.

Arguments: None.

Common blocks: ADDR0, HUGE, and NESLEV.

Data: None.

---

## 5.2 ADDRX1C

Subroutine ADDRX1C sets pointer addresses specific to a given domain and variable by equivalencing the pointers to the common block addresses.

ADDRX1C is called from the main program and from subroutines CHKNST, INIT, INITNEST, INITSAV, NSTLEV1, NSTLEV2, NSTLEV3, OUTPUT, and SHUTDO.

Arguments: **IARR**.

Common blocks: ADDR1, ADDR2, ADDR3, ADDR4, ADDR5, ADDR6, ADDR7, ADDR8, ADDRNC, ADDRSP, ADDRV, HUGE, NESLEV, NHCNS, NONHYD, NONHYDB, RADIAT, and UPRAD.

Data: None.

---

## 5.3 ADDRX1N

Subroutine ADDRX1N sets pointer addresses as in ADDRX1C, except in this case, it is for a second domain needed at the same time as the first domain handled by ADDRX1C.

ADDRX1N is called from the main program and from subroutines CHKNST, INITNEST, INITSAV, NSTLEV1, NSTLEV2, and NSTLEV3.

Arguments: **IARR**.

Common blocks: ADDRN1, ADDRN2, ADDRN3, ADDRN4, ADDR5N, ADDR6N, ADDR7N, ADDR8N, ADDNSP, ADDRNN, ADDRVN, HUGE, NESLEV, NNCNS, NNNHYD, NNNHYDB, RADIATN, and UPRAD.

Data: None.

## 5.4 ARAMB

Subroutine ARAMB sets up matrices for a linear solver routine (ZX4LP) that returns mass fluxes as solutions for the Arakawa-Schubert convection scheme.

ARAMB is called from subroutine ARASCH. It calls subroutine ZX4LP.

Arguments: **[ITEST], DRW, IW1, DDSOL, DPSOL, [DB], [DC], [DA], [IW], IWW, MBOTH, NIMSL, [ITEST2], IA, MINP, XK, F, xmb, KB, [IER], DTIME, FTEST, KNUM.**

Common blocks: None.

Data: None.

## 5.5 ARAOUT

Subroutine ARAOUT calculates feedback effects of clouds for the Arakawa-Schubert convection scheme.

ARAOUT is called from subroutine ARASCH.

Arguments: **[XMC], XMB, ZU, DELT, DELQ, KDIM, outtem, outq, ier, PW, pre, PWD, ZD, EDT, KNUM.**

Common blocks: None.

Data: None.

## 5.6 ARAOUTS

Subroutine ARAOUTS calculates feedback effects of clouds for the shallow convection scheme.

ARAOUTS is called from subroutine SHALLOW.

Arguments: **[XMC], XMB, ZU, DELT, DELQ, KDIM, outtem, outq, ier, pre, KNUM.**

Common blocks: None.

Data: None.

## 5.7 ARASCH

Subroutine ARASCH is the Arakawa-Schubert cumulus parameterization routine. Although it is the main routine for the Arakawa-Schubert scheme, it calls several other related subroutines.

ARASCH is called from subroutine CUPARA3. It calls subroutines ARAMB, ARAOUT, CLODWD, CLOUDW, ENTR, HEIPRE, KERHEL, MAXIM, MINIM, MOIENE, PRECIP, SOUNDD, and ZUNC.

Arguments: **TI, QI, Z1, TIO, QIO, PIO, KLEV, pre, PI, outtem, outq, DTIME, KBMAX, PCUT, C0, FTEST, PSUR, PSURO, ierrt, KDET, VSP.**

Common blocks: HUGE, NESLEV, PARAM2, PARAM3, PARFDDA, and PMOIST.

Data: **RAD (KNUM)** /0., 20000., 2000., 400., 267., 200./ cloud updraft radius (m).

---- Program Detail ----

ARASCH computes the moisture cycle using an Arakawa-Schubert-type cumulus parameterization scheme that includes downdrafts. The sequence of the basic subroutine tasks is:

1. Initialize some of the arrays.
2. Compute the heights of the sigma levels (HEIPRE).
3. Calculate the moist static energy **HE** and the saturation moist static energy **HES** (MOIENE).
4. Determine the level with the highest moist static energy (MAXIM).
5. Determine the cloud base and level of largest negative buoyancy (MINIM).
6. Begin in-cloud calculations, use **LP** do-loop where **LP** is cloud type.

a) Calculate moist static energy and cloud top (ENTR).

b) Compute normalized mass-flux profile (ZUNC).

c) Calculate the amount of moisture in the updraft (PRECIP).

d) Calculate the precipitation efficiency in terms of windshear.

e) Determine downdraft thermodynamic properties and the normalized mass flux (SOUNDD).

f) Calculate efficiency of generation of kinetic energy for the updraft (CLOUDW) and for the downdraft (CLODWD).

7. Begin new in-cloud and modified environmental calculations, loop again over cloud type.

   a) Calculate cloud effect on environment (KERHEL).

   b) Call HEIPRE and MOIENE for heights and moist static energy.

   c) Call ENTR, ZUNC, PRECIP, and SOUNDD for the moist static energy (in cloud), normalized mass-flux profile, amount of moisture in the updraft, and the downdraft thermodynamic properties.

   d) Call CLOUDW and CLODWD for updraft and downdraft cloud-work functions.

8. Calculate the mass fluxes (ARAMB).

## 5.8 BDYIN

Subroutine BDYIN reads in the coarsest-grid boundary conditions (boundary data input file).

BDYIN is called from the main program and from subroutine INIT.

Arguments: **IUNIT, tbdybe, bdytim, IX, JX.**

Common blocks: ADDR0, ADDR3, ADDR4, HUGE, NESLEV, and NONHYDB.

Data: None.

## 5.9 BDYOVL1 [multi-tasked]

Subroutine BDYOVL1 adjusts overlapping grid-point values on and near the nested boundaries for overlapping nests.

BDYOVL1 is called from subroutines NSTLEV1, NSTLEV2, and NSTLEV3.

Arguments: **NESCOU.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR3, ADDR4, ADDRN1, ADDRN2, ADDRN3, ADDRN4, ADDRNN, HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, and PBLDIM.

Data: None.

## 5.10 BDYUV [multi-tasked]

Subroutine BDYUV sets boundary values of U, V according to the type of boundary conditions (fixed or time dependent) specified by the input variable **IBOUDY** defined in PARAM.

BDYUV is called from subroutine BDYVAL. It calls subroutine DOTS.

Arguments: **IB, DTB.**

Common blocks: ADDR1, ADDR2, ADDR3, ADDR4, HUGE, MIC, NESLEV, PARAM3, and PBLDIM.

Data: None.

## 5.11 BDYVAL [multi-tasked]

Subroutine BDYVAL sets boundary values of the water substance variables equal to 0 if the wind flow is into the domain (inflow), or equal to the value of the first interior point if outflow.

BDYVAL is called from the main program. It calls subroutine BDYUV.

Arguments: **IN, NN, IEXEC.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR3, ADDR4, HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, and PBLDIM.

Data: None.

## 5.12 BLBRGD

Subroutine BLBRGD is an analysis-nudging FDDA routine that calculates the new values of **TIMB** and **TIME**, and position counters **NTB** and **NTE**, for bracketing **XTIME** with two surface analyses.

BLBRGD is called from subroutine BLNUDGD.

Arguments: **NV, MVAR, MTIM, XTIME, SFCTIM, timb, time, ntb, nte*, IQCHK, IN, KTAU, KTAUR, IFREST**.

Common blocks: None.

Data: None.

## 5.13 BLKPBL

Subroutine BLKPBL calculates the PBL tendencies using a bulk aerodynamic parameterization.

BLKPBL is called from subroutines SOLVE1 and SOLVE3. It calls subroutines SFCRAD and SLAB.

Arguments: **IYY, JXX, KZZ, J, INEST, NST, uten*, vten*, qcten*, scr1*, scr2*.**

Common blocks: ADDR1, ADDR2, ADDR4, ASSEL, HUGE, MIC, NESLEV, NHCNS, NHCNST, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: **KZO** /1./ background diffusion coefficient ($m^2$ $s^{-1}$). **RIC** /3.05/ critical bulk Richardson number for Deardorff scheme. **SZKM** /1600./ constant used to compute the eddy diffusion coefficients ($m^2$).

---- Program Detail ----

BLKPBL uses wind speed, PBL height, and the bulk Richardson number to compute the sensible heat, latent heat, and momentum fluxes from the ground. The sequence of the basic subroutine tasks is:

1. Compute wind speed and height for the lowest sigma level.
2. Compute the PBL height and bulk Richardson number.
3. Compute the exchange coefficient for momentum, heat, and moisture.
4. Compute sensible and latent heat flux (**ISFFLX** = 1).
5. Compute ground temperature from surface energy budget (**ITGFLG** = 1).
6. Add terrain effect to the exchange coefficient and compute momentum fluxes.
7. Compute the vertical mixing of momentum, heat, and moisture.

---- User Note ----

Some of the storage spaces for the high-resolution PBL are used to store the variables in BLKPBL. They are: **REGIME, PBL, MOL, HOL, HFX, QFX, ZOL, DUM1, DUM2, SCR1, SCR2. SCR1** is used to sum temperature tendency, and **SCR2** is used to sum moisture tendency. **DUM1** is used to compute average **PSB**, and **DUM2** is the average horizontal wind speed.

## 5.14 BLNUDGD

Subroutine BLNUDGD is an analysis-nudging FDDA routine that computes the surface-analysis nudging term in the PBL for a general model variable.

BLNUDGD is called from subroutine NUDGD. It calls subroutines BLBRGD, INTPSGD, and MAPSMP.

Arguments: **MTIM, MVAR, J, XB, xten*, pten*, PSTF, PSTO, GX, ID, WXY, WXY2, blwnv*, WPBL, TFAC, ZFAC, GP, IVAR, IN, KPBLT, SFCOBS, SFCTIM, QSATF, IQCHK, timb*, time*, ntb*, nte*, [SCR2D], [BLDUM2D], blpsoc*, blpsod*, REGJ, ZNTJ.**

Common blocks: ADDR4, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

---- Program detail ----

BLNUDGD computes the surface-analysis nudging term within the PBL of the tendency equation (**XTEN**) for **XB**, where **XB** is a general model field at time t-1. The sequence of basic subroutine tasks is:

1. If necessary, calculate new values of **TIMB** and **TIME**, and position counters **NTB** and **NTE**, which bracket **XTIME** with two surface analyses (BLBRGD).
2. If nudging p* from surface-analysis files, perform temporal interpolation (INTPSGD).
3. Calculate the storage array, **BLWNV**, for the horizontal weighting function based on surface data density.
4. Adjust the observed surface wind for the lowest model layer using similarity relationships based on roughness.
5. Adjust observed $q_v$ if supersaturated.
6. Compute analysis-nudging term of the tendency equation within the PBL.
   a) If **ID** = 0, compute the normal nudging term for U, V, T, or $q_v$.
   b) If **ID** = 1, compute the nudging term in the continuity (**PTEN**) equation in the hydrostatic version of the model.
   c) If **ID** = 3, compute the moisture nudging term using observed precipitation data.

## 5.15 BLW

Subroutine BLW is an analysis-nudging FDDA routine that computes the horizontal weighting function for surface-analysis nudging within the PBL, based on surface data density (**BLWXY**).

BLW is called from subroutine IN4DGD. It calls subroutine SMTHER.

Arguments: **IL, JL, WTTOP, blxwy*, [BLDUM2D], RINBLW, DX, TIMANLS, MDATES, IN**.

Common blocks: HUGE and NESLEV.

Data: None.

## 5.16 BUFSLGD

Subroutine BUFSLGD is an analysis-nudging FDDA routine that "decouples" (e.g., u from p*u) and interpolates observed analyzed values for a general model variable to time t-1 for use in calculating the analysis-nudging term.

BUFSLGD is called from subroutines SOLVE1 and SOLVE3.

Arguments: **J, XOB, XOBTEN, xobjk, PSTO, FDTIM, NV, IN.**

Common blocks: ADDR4, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

## 5.17 BUFVDGD

Subroutine BUFVDGD is an analysis-nudging FDDA routine that computes the observed minus model forecast vorticity difference, **VORDIF**, for use in the analysis nudging of vorticity.

BUFVDGD is called from subroutine SETFD.

Arguments: **UOB, VOB, UOBTEN, VOBTEN, UB, VB, PSTO, FDTIM, MSD, MSX, vordif, PSTF, IN**.

Common blocks: ADDR4, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

## 5.18 CADJMX

Subroutine CADJMX sets up the indices and matrices needed for the Gaussian elimination method used for the dry-convective adjustment.

CADJMX is called from subroutine CONVAD. It calls subroutine GAUSS.

Arguments: **KB, KE, PI, S, nadj, DSIGMA, DTHDPC, P, [AX], [B], [C], D,** y, KL, MKX.

Common blocks: None.

Data: None.

## 5.19 CHKNST

Subroutine CHKNST checks whether a nest should be activated and initialized (INITNEST and OUTTAP), moved and initialized (INITNEST and OUTTAP), or deactivated.

CHKNST is called from the main program and from subroutine INIT. It calls subroutines ADDRX1C, ADDRX1N, INITNEST, OUTTAP, and SPINIT.

Arguments: **iexec, NESCOU.**

Common blocks: ADDR0, ADDR4, ADDRNN, HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: None.

## 5.20 CLODWD

Subroutine CLODWD calculates the efficiency of generation of kinetic energy for downdrafts in the Arakawa-Schubert convection scheme.

CLODWD is called from subroutine ARASCH.

Arguments: **L1, KS, KMIN, HCD, QES, HES, ZU, Z, TEMPP, KDIM, axd, KNUM.**

Common blocks: None.

Data: None.

## 5.21 CLOUDW

Subroutine CLOUDW calculates the efficiency of generation of kinetic energy for updrafts in the Arakawa-Schubert convection scheme.

CLOUDW is called from subroutine ARASCH.

Arguments: **L1, HC, QES, HES, ZU, Z, TEMPP, KDIM, ax, KBCON, KNUM, KTOP.**

Common blocks: None.

Data: None.

## 5.22 CLOUDWS

Subroutine CLOUDWS calculates the efficiency of generation of kinetic energy for shallow clouds in the shallow convection scheme.

CLOUDWS is called from subroutine SHALLOW.

Arguments: **L1, HC, QES, HES, ZU, Z, TEMPP, KDIM, ax, KBCON, KNUM, KTOP.**

Common blocks: None.

Data: None.

## 5.23 CONADV

Subroutine CONADV computes the amount of dry air and water substance advected through the east-west and north-south lateral boundaries. The advection unit is converted to kg.

CONADV is called from the main program.

Arguments: **ID, IM.**

Common blocks: ADDR1, ADDR2, ADDR4, HUGE, NESLEV, and PARAM3.

Data: None.

## 5.24 CONMAS

Subroutine CONMAS computes the total amount of dry air and water substance in the domain. Total mass (kg), including advection, evaporation, and rainfall are calculated.

CONMAS is called from the main program. It calls subroutine TMASS.

Arguments: **ID, IM, MC.**

Common blocks: ADDR1, ADDR2, ADDR4, HUGE, NESLEV, and PARAM3.

Data: None.

## 5.25 CONV3

Subroutine CONV3 is an analysis-nudging FDDA routine that buffers a 2-D array **SLAB(IMAX, JMAX)** into a 4-D array **SFCOBS(MTIM, MVAR, IMAX, JMAX)**, and also places the model-relative time of the 2-D array into the 1-D array **SFCTIM**.

CONV3 is called from subroutine IN4DGD.

Arguments: **SLAB, sfcobs, MIX, MJX, TIMANL, sfctim, MTIM, MVAR, IMAX, JMAX, NT, NV.**

Common blocks: None.

Data: None.

## 5.26 CONVAD

Subroutine CONVAD is the main routine used for the dry-convective adjustment.

CONVAD is called from subroutines SOLVE1 and SOLVE3. It calls subroutine CADJMX.

Arguments: **ILXM, KL, NUMNES, J.**

Common blocks: HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, PBLDIM and PMOIST.

Data: **DTDPC** /-0.03/ critical lapse rate (K $cb^{-1}$).

---- Program Detail ----

CONVAD performs the dry convective adjustment and mixes water vapor whenever the vertical derivative of potential temperature exceeds the critical lapse rate. The sequence of the basic subroutine tasks is:

1. Compute the potential temperature.
2. Determine if multiple-layer adjustment is needed by comparing the derivative over two or more layers to a critical lapse rate, **DTDPC** = -0.03 K $cb^{-1}$.
3. Perform layer adjustment by calling CADJMX.

## 5.27 CUP

Subroutine CUP is the Grell cumulus parameterization routine.

CUP is called from subroutine CUPARA2. It calls subroutine MAXIMI and MINIMI.

Arguments: **QCRIT, T, Q, Z1, TN, QO, PO, pre*, P, outtem*, outq*, DTIME, PSUR, VSP, ISTART, IEND, KDET.**

Common blocks: HUGE, NESLEV, PARAM3, and PMOIST.

Data: None.

---- Program Detail ----

CUP computes the moisture cycle using the Grell cumulus parameterization scheme that includes downdrafts. The sequence of the basic subroutine tasks is:

1. Compute the heights of the sigma levels.
2. Calculate the moist static energy **HE** and the saturation moist static energy **HES**.
3. Determine the level with the highest moist static energy (MAXIMI).
4. Determine the cloud base, cloud top, and level of largest negative buoyancy (MINIMI).
5. Calculate the available buoyant energy.
6. Downdraft calculations.
   a) Calculate the precipitation efficiency in terms of windshear.
   b) Determine thermodynamic profiles in downdrafts.
   c) Calculate normalized downdraft strength in terms of precipitation efficiency and normalized condensation and precipitation.
   d) Calculate effects on the environment per unit mass.
7. Recalculate the environmental thermodynamic profiles, available buoyant energy.
8. Calculate mass fluxes and feedbacks

## 5.28 CUPARA1

Subroutine CUPARA1 is the Kuo-Anthes cumulus parameterization routine.

CUPARA1 is called from subroutines SOLVE1 and SOLVE3.

Arguments: **J, IN, qvten*, tten*.**

Common blocks: ADDR1, ADDR2, ADDR4, ASSEL, HUGE, MIC, NESLEV, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM and PMOIST.

Data:
**CDSCLD** /0.3/ critical cloud depth in sigma units. If the depth is less than the critical value, the convection is suppressed.
**DLT** /3.0/ temperature difference (K) used to allow over-shooting of cloud top above level of zero buoyancy.
**PERQ** /1.0E-3/ perturbation mixing ratio (kg $kg^{-1}$) to initiate the cloud.
**PERT** /1.0/ perturbation temperature (K) to initiate the cloud.

---- Program Detail ----

CUPARA1 computes the moisture cycle using a Kuo-Anthes type cumulus parameterization scheme. The sequence of the basic subroutine tasks is:

1. Compute the moisture convergence in a column.
2. Determine if convection exists by comparing the total moisture convergence to the threshold convergence, QDCRIT = 3.0E-7.

3. Compute cloud base and top, and check stability for convection potential.
4. If convection exists, compute convective flux of water vapor and latent heat.
5. If convection does not exist, compute vertical flux divergence term.

## 5.29 CUPARA2

Subroutine CUPARA2 sets up the variables for the Grell convection scheme (CUP).

CUPARA2 is called from subroutines SOLVE1 and SOLVE3. It calls subroutine CUP.

Arguments: **TA, QVA, PSA, tten*, qten*, rainc*, J, HT, UA, VA, INEST.**

Common blocks: ADDR4, HUGE, NESLEV, NONHYD, PARAM2, PARAM3, PARFDDA, and PMOIST.

Data: None.

## 5.30 CUPARA3

Subroutine CUPARA3 sets up the variables for the Arakawa-Schubert convection scheme (ARASCH).

CUPARA3 is called from subroutine SOLVE1. It calls subroutine ARASCH.

Arguments: **TA, QVA, PSA, tten*, qten*, rainc*, J, HT, UA, VA, NUMNES.**

Common blocks: ADDR4, ARASCH, HUGE, NESLEV, NONHYD, PARAM2, PARAM3, PARFDDA, and PMOIST.

Data: None.

----User Note----

Set **IARASC** = 1 for this option.

## 5.31 DECPU

Subroutine DECPU decouples the U, V variables from p*.

DECPU is called from subroutines SOLVE1 and SOLVE3.

Arguments: **KZZ, ften*, FB, PSB, XK, MSF, [SCR], C203, J, IEND, JEND, IND, INEST.**

Common blocks: ADDR3, HUGE, and NESLEV.

Data: None.

## 5.32 DIFFU

Subroutine DIFFU computes the diffusion term for p*U and p*V using a fourth-order scheme for the interior and a second-order scheme for the boundaries. The diffusion term is calculated on a constant sigma surface.

DIFFU is called from subroutines SOLVE1 and SOLVE3.

Arguments: **C203, FB, FTEN, IEND, IND, INEST, J, JEND, MSF, PSB, SCR, and XK.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.33 DIFFUT

Subroutine DIFFUT computes the diffusion term for p*T. The routine computes the derivatives of T along the sigma surfaces.

DIFFUT is called from subroutines SOLVE1 and SOLVE3.

Arguments: **KZZ, ften*, FB, PSB, XK, C201, J, IEND, JEND, INEST.**

Common blocks: HUGE, NESLEV, and PARAM3.

Data: None.

## 5.34 DIVG [multi-tasked]

Subroutine DIVG computes the divergence from U and V for split-explicit scheme.

DIVG is called from subroutine SPSTEP2.

Arguments: **[U], [V], z, MSFD, MSFX, DX2, KL, IMIN, JMIN, IMAX, JMAX.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.35 DOTS

Subroutine DOTS converts cross-point arrays to dot-point arrays using a four-point average for the interpolation.

DOTS is called from subroutines BDYUV, INITNEST, INTPSGD, SETUPGD, and SOLVE3.

Arguments: **SLAB1, slab2, IS1, IS2, ID3, ID4.**

Common blocks: None.

Data: None.

## 5.36 ENTR

Subroutine ENTR calculates the in-cloud moist static energy and cloud top for the Arakawa-Schubert convection scheme.

ENTR is called from subroutine ARASCH.

Arguments: **CD, KBC, H, hc, HSAT, ENT, KDIM, LP, KB, P, HKB, KNUM, ktop.**

Common blocks: None.

Data: None.

## 5.37 ENTRS

Subroutine ENTRS calculates the in-cloud moist static energy and cloud top for the shallow convection scheme.

ENTRS is called from subroutine SHALLOW.

Arguments: **CD, KBC, H, hc, HSAT, ENT, KDIM, LP, KB, P, HKB, KNUM, ktop.**

Common blocks: None.

Data: None.

## 5.38 EQUATE

Subroutine EQUATE puts select values of a 3-D input array **FIN(IXI, JXI, KXI)** into select locations of a 3-D output array **FOUT(IXO, JXO, KXO).**

EQUATE is called from subroutines IN4DGD, INITNEST, OUTTAP and RDINIT.

Arguments: **FIN, IXI, JXI, KXI, fout, IXO, JXO, KXO.**

Common blocks: None.

Data: None.

## 5.39 ERROB

Subroutine ERROB is an observation-nudging FDDA routine that calculates the difference between the observation and the model forecast at the observation location for all model variables. The difference for each variable and observation location is stored in array, **ERRF**, and used for observation nudging within the specified time window (**TWINDO**).

ERROB is called from subroutine SETFD.

Arguments: **IN, UB, VB, TB, QVB, PSB.**

Common blocks: ADDR4, ADDR8, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

## 5.40 EXAINT

Subroutine EXAINT is called whenever a nest is initialized or moved. It sets up input arrays for the interpolation routine SINT.

EXAINT is called from subroutines INITNEST, STOTNDI, and STOTNDT. It calls subroutine SINT.

Arguments: **TA, IYY, JXX, tan, IYYN, JXXN, ISOUTH, JWEST, ICRSDOT.**

Common blocks: DEPAR2, HUGE, and NESLEV.

Data: None.

## 5.41 EXCHANI

Subroutine EXCHANI exchanges boundary values between nests for particular **I**-index values from **JSTARTN** to **JENDN**.

EXCHANI is called from subroutine BDYOVL1.

Arguments: **IARG, IARGN, IOVE, JSTART, JSTARTN, JENDN, K, ARR, arrn*, ARRB, arrnb*, arrbb*, arrbt*, KZZ, NSPG, DT, ISTOP.**

Common blocks: None.

## 5.42 EXCHANJ

Subroutine EXCHANJ exchanges boundary values between nests for particular **J**-index values from **ISTARTN** to **IENDN**.

EXCHANJ is called from subroutine BDYOVL1.

Arguments: **ARG, JARGN, JOVE, ISTART, ISTARTN, IENDN, K, ARR, arrn*, ARRB, arrnb*, arrbb*, arrbt*, KZZ, NSPG, DT, ISTOP.**

Common blocks: None.

Data: None.

## 5.43 EXMOISR

Subroutine EXMOISR is an explicit scheme for computing the moisture tendencies that includes an ice option that allows for supercooled liquid water.

EXMOISR is called from subroutines SOLVE1 and SOLVE3.

Arguments: **J, IN, IM, qcten*, qrten*, qiten*, qniten*, qvten*, tten*, IST, IEN.**

Common blocks: ADDR1, ADDR2, ADDR4, JRG, MIC, NHCNST, NONHYD, PARAM3, PBLDIM, and PMOIST.

Data: None.

---- Program detail ----

EXMOISR computes the tendencies of water vapor, cloud water, cloud ice, snow, and rainwater using cloud microphysics parameterization for the autoconversion, accretion, evaporation, deposition/sublimation, melting, and freezing processes. This routine allows for supercooled liquid water and has more sophisticated melting processes than EXMOISS. The sequence of the basic subroutine tasks is:

1. Calculate the saturation vapor pressure.
2. Compute the initiation of cloud ice (**PRI**), freezing of cloud droplets (**NUFCI**), accretion of cloud ice by snow (**PRAI**), collection of cloud water by snow (**PSACW**), deposition/sublimation of cloud ice (**PRD**), autoconversion of cloud ice to snow (**PRCI**), and deposition/sublimation of snow (**PREI**).
3. Compute the autoconversion of cloud to rain (**PRC**), accretion of cloud by rain (**PRA**), and evaporation of rain (**PRE**).
4. Compute melting of snow (**PSMLT**) and evaporation of melting snow (**PMLTEV**).
5. Calculate tendencies ensuring positive water/ice after adding them.
6. Calculate condensation term for vapor to cloud.
7. Compute the fallout terms (with split time steps) for rain and snow.
8. Include effects of melting and freezing of cloud ice.

----User Note----

**IICE** = 1 to use this option, and **IEXMS** = 1

## 5.44 EXMOISS

Subroutine EXMOISS is an explicit scheme for computing the moisture tendencies that includes a simple ice option.

EXMOISS is called from subroutines SOLVE1 and SOLVE3.

Arguments: **J, IN, IM, qcten*, qrten*, qvten*, tten*, SCR9, IST, IEN.**

Common blocks: ADDR1, ADDR2, ADDR4, ASSEL, HUGE, MIC, NESLEV, NHCNS, NHCNST, NONHYD, PARAM3, PBLDIM, and PMOIST.

Data: None.

---- Program detail ----

EXMOISS computes the tendencies of water vapor, cloud water, cloud ice, snow, and rainwater using cloud microphysics parameterization for the autoconversion, accretion, evaporation, deposition/sublimation, melting and freezing processes. The sequence of the basic subroutine tasks is:

1. Calculate the saturation vapor pressure
2. Compute the autoconversion (**PRC**), accretion (**PRA**), and evaporation (**PRE**) terms without ice (all production terms are based on variables at time t-1).
3. Compute the initiation of cloud ice (**PRI**) and deposition/sublimation of cloud ice (**PRD**) terms.
4. Re-adjust **PRC, PRA, PRI,** and **PRD** if total production is greater than that available.
5. Compute p*T, $p^*q_v$, and $p^*q_c$ (at time t+1) without the condensation term, and $p^*q_r$ without the fallout term.
6. Compute the condensation and fallout (with split time steps) terms.
7. Include effects of freezing and melting of particles depending on the sign of the vertical motion between sigma layers.

---- User note ----

Initially, **TAOUT, QAOUT, SCR2, SCR3, SCR6,** and **SCR8** are used to store 2-D slices of T, $q_v$, $q_c$, $q_r$, pressure, and air density at time t-1. Set **IEMXS** = 1 for this option.

## 5.45 FDAOFF

Subroutine FDAOFF resets the analysis-nudging and observation-nudging FDDA switches for a particular domain (**NUMNES**).

FDAOFF is called from subroutine OUTPUT.

Arguments: **NUMNES.**

Common blocks: HUGE, NESLEV, PARAM2 and PARFDDA.

Data: None.

## 5.46 FEEDBK [multi-tasked]

Subroutine FEEDBK computes the feedback effects from the nested domain to the coarse domain.

FEEDBK is called from subroutines NSTLEV1, NSTLEV2, and NSTLEV3. It calls subroutines MDOTS, SEAPRS, SMTHER, and SMT2.

Arguments: **INEST, IYY, JXX.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR4, ADDRN1, ADDRN2, ADDRN4, ADDRNN, ADDRV, ADDRVN, HUGE, MIC, NESLEV, NONHYD, NNNHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: None.

---- Program detail ----

FEEDBK interpolates (if IFEED = 1) coarse-grid values in the nested domain from the nested values using a nine-point averaging technique. If IFEED > 1, it sets coarse values to coincident nested values. The order of variables for the calculations are: p*, $T_g$, T, W, and p′ (if nonhydrostatic), $q_v$, $q_c$, $q_r$, $q_i$, $q_s$, U, and V. The interpolated data are smoothed using a 2-pass smoother-desmoother (SMTHER) if IFEED = 3.

## 5.47 FILL

Subroutine FILL puts select values of a 2-D input array **HT(MIX, MJX)** into select locations of a 2-D output array **HSCR1(IYYN, JXXN).**

FILL is called from subroutine INITNEST.

Arguments: **HT, hscr1, MIX, MJX, IYYN, JXXN, IEND, JEND.**

Common blocks: None.

Data: None.

## 5.48 FILSLB

Subroutine FILSLB overwrites new nest values (of a moving nest) with old nest values in regions where they overlap.

FILSLB is called from subroutine INITNEST.

Arguments: **anew, AOLD, ISOUTH, ISOUTHO, JWEST, JWESTO, MIX, MJX, ICRSDOT.**

Common blocks: None.

Data: None.

## 5.49 GAUSS

Subroutine GAUSS is a routine used for the dry-convective adjustment. It employs the Gaussian elimination method to solve a matrix system.

GAUSS is called from subroutine CADJMX.

Arguments: **AX, B, C, [D], y*, LM, MKX.**

Common blocks: None.

Data: None.

## 5.50 HADV

Subroutine HADV computes the horizontal advection (flux-divergence) terms using second-order finite differences. The upstream values are optionally used for the $q_c$ and $q_r$ tendencies to ensure positive definiteness.

HADV is called from subroutines SOLVE1 and SOLVE3.

Arguments: **ften*, UA, VA, F, MSF, DXX, J, IND, IN.**

Common blocks: ADDR4, HUGE, NESLEV, and PARAM3.

Data: None.

## 5.51 HEIPRE

Subroutine HEIPRE calculates heights for the deep (Arakawa-Schubert) and shallow convection schemes.

HEIPRE is called from subroutines ARASCH and SHALLOW.

Arguments: **P, z, T, Z1, KDIM, KS, PSURF.**

Common blocks: None.

Data: None.

## 5.52 HIRPBL

Subroutine HIRPBL is a high-resolution PBL routine based on a model developed by Blackadar.

HIRPBL is called from subroutines SOLVE1 and SOLVE3. It calls subroutines SFCRAD and SLAB.

Arguments: **IYY, JXX, KZZ, J, IN, NST, uten*, vten*, qcten*, scr1*, scr2*, [KEPBLH], [KWPBLH], qiten*, qniten*.**

Common blocks: ADDR1, ADDR2, ADDR4, ASSEL, HUGE, MIC, NESLEV, NHCNS, NHCNST, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data:
**C1** /0.2721655/ constant ($\sqrt{2/27}$) used in Priestley's equation.
**C2** /-0.33333/ constant used in Priestley's equation.
**CZO** /0.032/ constant used to compute roughness length over water.
**ENTRMT** /0.2/ entrainment coefficient used for free convection.
**KA** /2.4E-5/ background molecular diffusivity ($m^2$ $s^{-1}$) used to compute latent heat flux.
**KZO** /1.0/ background diffusion coefficient ($m^2$ $s^{-1}$).
**OZO** /1.0E-4/ constant (m) used to compute roughness length over water.
**SZKM** /1600./ constant ($m^2$) used to compute the eddy diffusion coefficients.

---- Program Detail ----

HIRPBL uses horizontal winds, temperature, cloud water, and water vapor at half-sigma levels to compute the bulk Richardson number of surface layers to determine stability classes that define basic parameters affecting the PBL forecast. The algorithms are written to allow the PBL to be described as a discrete, variable number of model layers of arbitrary thickness. The sequence of the basic subroutine tasks is:

1. Convert ground temperature to potential temperature.
2. Decouple flux-form variables to give U, V, T, potential temperature, virtual potential temperature, virtual temperature, $q_v$, and $q_c$ at cross points at time step **KTAU**-1.
3. Compute the height of the full- and half-sigma levels and the layer thickness.

4. Calculate the critical Richardson number.
5. Begin iteration loop steps.
6. Initialize the vertical tendencies and calculate the bulk Richardson number of the surface layer.
7. Diagnose basic parameters for the appropriate stability classes
   a) stable (night) conditions,
   b) damped mechanical convection,
   c) forced convection, and
   d) free convection.
8. Calculate the frictional velocity and surface radiation (SFCRAD).
9. Compute the surface sensible and latent heat fluxes, and momentum fluxes.
10. Compute ground temperature (SLAB).
11. Compute PBL prognostic terms (explicit computation of surface layer values for classes 1-3, and PBL computations free convection class.)
12. Get PBL tendencies above the surface layer and sum the fluxes.
13. Compute the weighting factor (**WGT**) for simulating reduced mixing rates near the top of the PBL.
14. Compute the surface layer tendencies.
15. Apply vertical diffusion (include moist vertical diffusion if **IMVDIF** = 1).
16. End iteration loop.
17. Convert all tendencies to flux form and store in **SCR1** and **SCR2** (T and $q_v$).
18. Apply Asselin filter to ground temperature and update it.
19. Convert ground potential temperature to temperature.

---- User Note ----

**SCR1, SCR2, SCR3,** and **SCR4** store **TTEN**, **QVTEN**, temperature, and virtual temperature respectively. **DUM1** is used in computing the height of the full- and half-sigma levels.

## 5.53 IN4DGD

Subroutine IN4DGD is an analysis-nudging FDDA routine that reads in the surface and upper-air analyses which are used for analysis nudging during the period **XTIM1** to **XTIM2**.

IN4DGD is called from subroutines INIT, INITNEST, and SETFD. It calls subroutines BLW, CONV3, EQUATE, JULGMT, and NOPRO.

Arguments: **IN, IYY, JXX, KZZ.**

Common blocks: ADDR4, ADDR5, ADDR6, ADDR7, CFD, CFDDAGD, HEADER, HEADERC,HUGE, NESLEV, PARAM2, PARAM3, PARFDDA, and SIZE.

Data: None.

---- Program detail ----

IN4DGD reads in upper-air and surface analyses files and selects data based on **TIMANL**, **DIFTIM, TIMANLS** and **DIFTIMS**. The sequence of the basic subroutine tasks is:

1. Compute **TIMANL**, the model-relative time of the input data (JULGMT), and check for consistency with **XTIME**, the current time step model time.
2. Read in observed upper-air analyses **UBO, VBO, TBO, QBO** and **PSO** at two time levels (**XTIM1** and **XTIM2**).
3. Calculate observed tendencies **UBOTEN, VBOTEN, QBOTEN, TBOTEN,** and **PSOTEN** from **XTIM1** to **XTIM2**.
4. Check whether surface analysis nudging is to be employed. If yes:
   a) Read in surface analyses between **XTIM1** and **XTIM2** and put into **DUM2D** (EQUATE).
   b) Compute horizontal weighting function (BLW) which is used for surface-analysis nudging within the PBL.
   c) Buffer 2-D array **DUM2D** into 4-D array **SFCOBS** (CONV3).
5. Print information concerning the details of the parameter choices related to analysis nudging.

## 5.54 IN4DOB

Subroutine IN4DOB is an observation-nudging FDDA routine that reads in the observations (all variables) used for observation nudging within the specified time window (**TWINDO**).

IN4DOB is called from subroutines INIT, INITNEST, and SETFD.

Arguments: **INEST, XTIME, KTAU, KTAUR, DTMIN.**

Common block: ADDR8, HUGE, NESLEV, PARAM2, PARAM3, and PARFDDA.

Data: **IEOF** /0/ flag indicator (0 or 1) for end of file. **NMOVE** /0/ initialize (set to 0) number of observations to be removed. **NVOLA** input logical unit number constant for **NVOL**, where **NVOL** = **NVOLA** + **INEST**-1.

---- Program detail ----

IN4DOB reads in observation data files and selects only observations that are contained within the designated time window. The sequence of the basic subroutine tasks is:

1. Define time window limits in model-relative times.
2. Check for old observations that can be discarded.
3. Read in all observations within the time window.
4. Convert the observation time from Julian date and **GMT** form to model-relative (forecast) time.
5. Determine if maximum number of observations exceeds allowable observation array size. If so, stop and increase size parameter **NIOBF**, or optionally, decrease size of time window (generally not a good idea).
6. Print information concerning the details of parameter choices related to observational nudging.

## 5.55 INIT

Subroutine INIT gets (or specifies) the coarse-grid initial conditions and boundary conditions.

INIT is called from the main program. It calls subroutines ADDRX1C, BDYIN, CHKNST, IN4DGD, IN4DOB, OUTTAP, RDINIT, SOLAR1, and TMASS.

Arguments: **IEXEC, IX, JX.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR4, ADDR5, ADDR6, ADDR7, ADDR8, ADDRNN, ASSEL, CFD, CFDDAGD, HUGE, LANDUSE, MIC, NESLEV, NHCNS, NHCNST, NONHYD, NONHYDB, PARAM2, PARAM3, PARFDDA, and PBLDIM.

---- Program detail ----

INIT reads and stores initial and boundary conditions for the coarse-grid domain, and initializes some of the input parameter variables. The sequence of the basic subroutine tasks is:

1. Variables used to store rainwater, cloudwater, snow and ice are initialized to 0.
2. Total mass and PBL variables are initialized to 0.
3. Read in the initial conditions (RDINIT).
4. Invert the map scale factor and set the Asselin filter mask.
5. Calculate the total mass of dry air and water substance (TMASS).
6. Specify surface parameters according to land-use category and read in the slab temperature.

7. If FDDA is employed, call subroutine that reads in the observations (for observation nudging) and/or analysis (for analysis nudging).
8. Initialize nests (CHKNST).
9. Read in boundary conditions (BDYIN).
10. Compute solar declination and set up the output times.

## 5.56 INITNEST [multi-tasked]

Subroutine INITNEST interpolates (or reads in) the nested-grid initial conditions.

INITNEST is called from subroutine CHKNST. It calls subroutines ADDRX1C, ADDRX1N, DOTS, DOTS2, EQUATE, EXAINT, FILL, FILSLB, IN4DGD, IN4DOB, OVLCHK, RDINIT, SEAPRS, SUBCH, and TMASS.

Arguments: **IYY, JXX, KZZ, IYYN, JXXN, KZZN, IYYM, JXXM, KZZM, IYYNM, JXXNM, KZZNM, NESTII, NESTJJ, NESCOU, HTNO, INSTES, ISOUTHO, JWESTO.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR4, ADDR5, ADDR6, ADDR7, ADDR8, ADDRN1, ADDRN2, ADDRN4, ADDRNN, ADDRV, ADDRVN, ASSEL, CFD, CFDDAGD, HUGE, LANDUSE, MIC, NESLEV, NHCNS, NHCNST, NNCNS, NNNHYD, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

---- Program detail ----

INITNEST interpolates (or reads in) and stores the initial conditions for the coarse domain, and initializes some of the input parameter variables. The sequence of the basic subroutine tasks is:

1. Adjust corner points of the nest, if a moving nest (SUBCH).
2. Initialize and set up the coordinates of the nest.
3. Define some constants for grid size and time step.
4. Check if the nest is to be overlaid with other nest (OVLCHK).
5. Recalculate sea-level pressure on coarse domain, interpolate them to the nest, and calculate $p_s$ on the nest based on nested terrain values. Note: EXAINT is used for all interpolations in INITNEST.
6. Interpolate U, V, T, $q_v$, $q_c$, and $q_r$ arrays.
7. Interpolate some PBL variables.
8. Interpolate nonhydrostatic arrays: reference temperature, p′, and W.
9. Interpolate 2-D terrestrial arrays.
10. Get total mass of dry air and water substance in the nested domain (TMASS).

11. Initialize some arrays pertaining to the Asselin filter.
12. Specify some surface parameters according to land-use category.
13. Read in nested initial conditions (RDINIT) if you already have a nested initial conditions file.
14. Initialize some nested FDDA parameters.
15. If FDDA, get observations and/or analyses needed (IN4DOB or IN4DGD).

## 5.57 INITSAV

Subroutine INITSAV calls the routine SAVREAD that reads restart file information needed for the initialization procedure for a restart. It also initializes some arrays pertaining to the Asselin filter.

INITSAV is called from the main program. It calls subroutines ADDRX1C, ADDRX1N, SAVREAD, SOLAR1, and SPINIT.

Arguments: **iexec.**

Common blocks: ADDR0, ADDR4, ADDRN4, ADDRNN, ASSEL, HUGE, NESLEV, PARAM2, PARAM3, PARFDDA, and PMOIST.

Data: None.

## 5.58 INTPSGD

Subroutine INTPSGD is an analysis-nudging FDDA routine used for temporal and dot-point interpolation of observed p*.

INTPSGD is called from subroutines BLNUDGD and SETFD. It calls subroutine DOTS

Arguments: **PSO, PSOTEN, psoc, psod, FDTIM, IN.**

Common blocks: ADDR4, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

## 5.59 INVMTX

Subroutine INVMTX is used to invert some matrices for the split-explicit scheme.

INVMTX is called from subroutine VMODES.

Arguments: **A, NA, v, NV, N, D, IP, ier.**

Common blocks: None.

Data: None.

## 5.60 JULGMT

Subroutine JULGMT is an FDDA routine that converts standard date and time information (**MDATE**) from the input data into model-relative time (**TIMANL**) and/or the Julian day and **GMT** form used by the FDDA routines.

JULGMT is called from subroutines IN4DGD and SETUPGD.

Arguments: **MDATE, julgmtn*, timanl*, JULDAY, GMT, IND.**

Common blocks: None.

Data: None.

## 5.61 KERHEL

Subroutine KERHEL calculates net cloud effect per unit mass on the environment for the Arakawa-Schubert convection scheme.

KERHEL is called from subroutine ARASCH.

Arguments: **VAR, R, LP, ZU, HKB, HC, della, P, Z, KB, KDIM, xvar, MBDT, KBEG, xhkb, CD, HCD, ZD, ITEST, KDET, EDT, PSU, KNUM, KTOP.**

Common blocks: None.

Data: None.

## 5.62 KERHELS

Subroutine KERHELS calculates net cloud effect per unit mass on the environment for the shallow convection scheme.

KERHELS is called from subroutine SHALLOW.

Arguments: **VAR, R, LP, ZU, HKB, HC, della, P, Z, KB, KDIM, xvar, MBDT, KBEG, xhkb, CD, KDET, PSU, KNUM, KTOP.**

Common blocks: None.

Data: None.

## 5.63 LWRAD

Subroutine LWRAD calculates the longwave component of the atmospheric radiative temperature tendencies and surface flux.

LWRAD is called from subroutines SOLVE1 and SOLVE3.

Arguments: **J**

Common blocks: ADDR1, ADDR2, ADDR4, HUGE, NESLEV, NHCNS, NHCNST, NONHYD, PARAM3, and RADIAT.

Data: None.

---- Program Detail ----

1. Presets column variables and radiation constants.
2. Calculates various path lengths for each model level.
3. Calculates downward fluxes for each model level by upward integration of path lengths and applying broadband emissivity, overlap approximation for $CO_2$ and assuming radiative grey clouds and precip.
4. Calculates upward fluxes for each model level by downward integration, accounting for ground emissivity.
5. Calculate longwave heating rate from radiative flux divergence.
6. Save downward longwave flux at surface (**GLW**) for ground heat budget.

----User Note----

Set **IRDDIM** = 1 for this option.

## 5.64 MAPSMP

Subroutine MAPSMP is one of the two main printer output routines for MM5. It prints select values of two-dimensional, five-significant-digit data fields.

MAPSMP is called from subroutines BLNUDGD, NUDGD, NUDOB, and OUTPRT. It calls subroutine VTRAN.

Arguments: **FLD, IYY, JXX, IA, IB, INY, JA, JB, JNX, CONST, ICHOS, NAME, TIME.**

Common blocks: None.

Data: KSIGT /5/ number of significant digits for data printout.

## 5.65 MAXIM

Subroutine MAXIM determines the level of the maximum value of a 1-D vertical **ARRAY(KDIM)**.

MAXIM is called from subroutines ARASCH and SHALLOW.

Arguments: **ARRAY, KDIM, KS, KE, max.**

Common blocks: None.

Data: None.

## 5.66 MAXIMI

Subroutine MAXIMI determines the level of the maximum value of each vertical string of a 2-D slice **ARRAY(MIX, MKX)**.

MAXIMI is called from subroutine CUP.

Arguments: **ARRAY, MIX, MKX, KS, KE, max, ISTART, IEND.**

Common blocks: None.

Data: None.

## 5.67 MINIM

Subroutine MINIM determines the level of the minimum value of a 1-D vertical **ARRAY(KDIM)**.

MINIM is called from subroutines ARASCH and SHALLOW.

Arguments: **ARRAY, KDIM, KS, KEND, kt.**

Common blocks: None.

Data: None.

## 5.68 MINIMI

Subroutine MINIMI determines the level of the minimum value of each vertical string of a 2-D slice **ARRAY(MIX, MKX).**

MINIMI is called from subroutine CUP.

Arguments: **ARRAY, MIX, MJX, KS, KEND, kt, ISTART, IEND.**

Common blocks: None.

Data: None.

## 5.69 MM5

MM5 is the main driver program for the predictive component of the MM5 modeling system.

MM5 calls subroutines ADDALL, ADDRX1C, ADDRXIN, BDYIN, BDYVAL, CHKNST, CONADV, CONMAS, INIT, INITSAV, NESLEV1, OUTPUT, PARAM, SHUTDO, SOLVE1, SOLVE3, SPINIT, SPLITF, and STOTNDI.

Arguments: None.

Common blocks: ADDR0, ADDR4, ADDRN4, ASSEL, HUGE, NESLEV, PARAM2, PARAM3, and PARFDDA.

Data: None.

---- Program detail ----

The sequence of the program tasks is:

1. Preset pointer addresses of all variables on all domains (ADDALL).
2. Set pointer addresses specific to a given domain and variable (ADDRX1C).
3. Setup (or define) model parameters (PARAM**)**.
4. If **IFREST**=.T.; read restart file (INITSAV), initialize some arrays, and check whether a nest should be moved, activated, or initialized (CHKNST).
5. Call the routine (INIT) that gets the initial coarse-grid conditions.
6. If **INHYD**=0, initialize arrays for the split-explicit scheme and perform split-explicit calculations at the initial time (SPINIT).
7. Begin forecast loop and read in coarse-grid boundary conditions (BDYIN).
8. Set boundary values for the water-substance variables (BDYVAL).
9. Set pointer addresses specific to a second domain if needed at the same time as the first domain (ADDRX1N).
10. If nests exist, interpolate and store nested-grid values from coarse-grid variables (STOTNDI) at $t - \Delta t$.
11. Compute the tendencies of the prognostic variables (call SOLVE1 for the hydrostatic version of the model, and SOLVE3 for the nonhydrostatic version).
12. If **INHYD**=0, perform the split-explicit time integration (SPLITF).
13. Compute amount of dry air and water substance advected through the lateral boundaries (CONADV).

14. Compute total amount of dry air and water substance in the domain.
15. Get the nested-domain tendencies (NSTLEV1).
16. Organize output for post-processor analyses (OUTPUT).
17. Shutdown forecast if out of CPU time or if domains are incompatible (SHUTDO).
18. Check whether a nest should be moved, activated, or initialized (CHKNST).
19. Continue forecast loop.

## 5.70 MOIENE

Subroutine MOIENE calculates the moist-static energy for the deep (Arakawa-Schubert) and shallow convection schemes.

MOIENE is called from subroutines ARASCH and SHALLOW.

Arguments: **T, Q, Z, h, KDIM, KS.**

Common blocks: None.

Data: None.

## 5.71 NOPRO

Subroutine NOPRO is an analysis-nudging FDDA routine that uses the information (defined in PARAM) on what times (**IDDATE**) and variables (**IDCHK**), from the surface-analysis nudging files, to withhold from processing and subsequent use for surface-analysis nudging within the PBL.

NOPRO is called from subroutine IN4DGD.

Arguments: **MDATES, IDDATE, IDCHK, MCHA, MVAR, idhk.**

Common blocks: None.

Data: None.

## 5.72 NSTLEV1

Subroutine NSTLEV1 is a driver subroutine for cycling through the subroutines that calculate the tendencies for all the nests at level one. The routine is similar to the main program since most of the subroutines called are the same as those called from the main program.

NSTLEV1 is called from the main program. It calls subroutines ADDRX1C, ADDRX1N, BDYOVL1, FEEDBK, NSTLEV2, SOLVE1, SOLVE3, SPINIT, SPLITF, STOTNDI, and STOTNDT.

Arguments: **ICOARS, IEXEC.**

Common blocks: ADDR0, ADDR3, ADDR4, ADDRN4, HUGE, NESLEV, NONHYDB, PARAM2, PARAM3, and PARFDDA.

Data: None.

## 5.73 NSTLEV2

Subroutine NSTLEV2 is a driver subroutine for cycling through the subroutines that calculate the tendencies for all the nests at level two. The routine is similar to the main program since most of the subroutines called are the same as those called from the main program.

NSTLEV2 is called from subroutine NSTLEV1. It calls subroutines ADDRX1C, ADDRX1N, BDYOVL1, FEEDBK, NSTLEV3, SOLVE1, SOLVE3, SPINIT, SPLITF, STOTNDI, and STOTNDT.

Arguments: **ICOARS, IEXEC.**

Common blocks: ADDR0, ADDR4, ADDRN4, HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, and PBLDIM.

Data: None.

## 5.74 NSTLEV3

Subroutine NSTLEV3 is a driver subroutine for cycling through the subroutines that calculate the tendencies for all the nests at level three. The routine is similar to the main program since most of the subroutines called are the same as those called from the main program.

NSTLEV3 is called from subroutine NSTLEV2. It calls subroutines ADDRX1C, ADDRX1N, BDYOVL1, FEEDBK, SOLVE1, SOLVE3, SPINIT, SPLITF, STOTNDI, and STOTNDT.

Arguments: **ICOARS, IEXEC.**

Common blocks: ADDR0, ADDR4, ADDRN4, HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, and PBLDIM.

Data: None.

## 5.75 NUDGD

Subroutine NUDGD is an analysis-nudging FDDA routine that computes the analysis-nudging term for a general model variable.

NUDGD is called from subroutines SETFD, SOLVE1 and SOLVE3. It calls subroutines BLNUDGD and MAPSMP.

Arguments: **MTIM, MVAR, INVAR, J, XB, XOBJK, xten*, pten*, PSTF, PSTO, GX, ID, WXY, WXY2, TFAC, ZFAC, GP, GR, MSD, IVAR, VORDIF, IN, KPBLT, BLWXY, blwnv*, SFCOBS, SFCTIM, QSATF, IQCHK, timb*, time*, ntb*, nte*, [SCR2D], [BLDUM2D], blpsoc*, blpsod*, REGJ, ZNTJ.**

Common blocks: ADDR4, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

---- Program detail ----

NUDGD computes the analysis-nudging term of the tendency equation (**XTEN**) for **XB**, where **XB** is a general model field at time t-1. The sequence of basic subroutine tasks is:

1. If **ID** = 1, compute the analysis-nudging term in the continuity (**PTEN**) equation (hydrostatic version of the model, and whether the p* fields from the surface-analysis or upper-air files are used for p* nudging depends on the **IPSTR** switch).
2. Define the vertical weighting function/partitioning factors (WPBL) between the surface-analysis nudging in the PBL and the analysis nudging of 3-D fields above the PBL. The partitioning is based on the current time step model PBL depth.
3. Compute surface-analysis nudging term of the tendency equation in the PBL (BLNUDGD).
   a) If **ID** = 0, compute the normal nudging term for U, V, T, or $q_v$.
   b) If **ID** = 3, compute the moisture nudging term using observed precipitation data.
4. Compute analysis-nudging term of the tendency equation above the PBL.
   a) If **ID** = 0, compute the normal nudging term for U, V, T, or $q_v$.
   b) If **ID** = 2, compute the rotational (vorticity) nudging term for U and V.
   c) If **ID** = 3, compute the moisture nudging term using observed precipitation data.

## 5.76 NUDGE

Subroutine NUDGE applies the relaxation boundary conditions to the local tendency term, **FTEN**. The final tendency is computed by a Newtonian term and a diffusion term.

NUDGE is called from subroutines SOLVE1 and SOLVE3.

Arguments: **IP, FCOEF, GCOEF, TBE, XT, ften*, FEB, FWB, FSB, FNB, FEBT, FWBT, FSBT, FNBT, FB, C203, KD, IE, JE, J.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.77 NUDOB

Subroutine NUDOB is an observational-nudging FDDA routine that computes the observations-nudging term for a general model variable.

NUDOB is called from subroutines SOLVE1 and SOLVE3. It calls subroutine MAPSMP

Arguments: **J, IVAR, aten*, pten*, IN, A, ISOUTH, JWEST, UB, VB, TB, QVB, PSB, PSBD, KPBLT, PTOP.**

Common blocks: ADDR4, ADDR8, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

---- Program detail ----

NUDOB computes the observation-nudging term of the tendency equation for a general model variable using a one-pass calculation of the weighting factors. The sequence of basic subroutine tasks is:

1. Compute the observation-nudging vertical weighting function **WPBL**, which is based on the current time step model PBL depth.
2. Compute the locations of the observations with respect to current mesh on either dot or cross points, depending on variable type.
3. Compute model p* at the observation locations for later use in the modified Cressman function used to spread observational information near the surface along constant sigma surfaces.
4. Begin outer loop for **NSTA** observations and check for proximity of each observation to grid point value.
5. Compute the temporal, horizontal and vertical weighting functions used to spread each surface observation along constant sigma near the surface, or to spread each observation above the surface along constant pressure (hydrostatic) or height (nonhydrostatic) levels.
6. Nudge model solution to surface-layer observations (default: winds only) on sigma levels.
   a) Loop through grid points within influence range of each surface-layer observation and compute the weighting sums over all observations.
7. Nudge model solution to observations above the model surface layer on pressure levels in **IWTSIG** = 0, or sigma levels if **IWTSIG** = 1:

a) Loop through grid points within influence range and compute the weighting sums over all observations.

8. Apply these sums of nudging weights to get the observation-nudging tendency **ATEN** (general model variable nudging tendency for **UTEN, VTEN TTEN, QVTEN** or **PTEN**).

   NOTE: For efficiency, these sums of weights used for observation nudging do not need to be computed every model timestep(see description of parameter **IONF**).

## 5.78 OUTPRT

Subroutine OUTPRT is one of the two main printer output routines for MM5.

OUTPRT is called from subroutine OUTPUT. It calls subroutine MAPSMP.

Arguments: **IEXEC, N.**

Common blocks: ADDR1, ADDR2, ADDR3, ADDR4, HUGE, MIC, NESLEV, NONHYD, NONHYDB, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data:
**IXN** /2/ sampling interval in I-direction.
**JXN** /2/ sampling interval in J-direction.
**KXN** /1/ sampling interval in K-direction.
**SVP**11 /0.611/ constant used for calculation of the saturation vapor pressure (cb).
**SVP**12 /19.84659/ constant used for calculation of the saturation vapor pressure.
**SVP**13 /5418.12/ constant used for calculation of the saturation vapor pressure (K).

---- Program detail ----

OUTPRT uses MAPSMP to print output data at time intervals specified in PARAM. The sequence of the basic subroutine tasks is:

1. Print north-south cross sections of U, V, T, W, $q_v$, $q_c$, and $q_r$.
2. Print horizontal slices:
   a) Latitude, longitude, terrain, Coriolis parameter, and map-scale factor.
   b) Surface albedo, surface roughness, moisture availability, thermal inertia, and surface emissivity.
   c) $p_s$ and $p_s$ change.
   d) U, V, T, relative humidity, $q_c$, and $q_r$.
   e) PBL variables.
   f) Ground temperature, convective, non-convective, and total rainfall.

## 5.79 OUTPUT

Subroutine OUTPUT is the driver subroutine for output from MM5. It calls OUTTAP (write forecast data to output file), OUTSAV (write save file), and OUTPRT (primary printout routine).

OUTPUT is called from the main program. It calls subroutines ADDRX1C, FDAOFF, OUTPRT, OUTSAV, and OUTTAP.

Arguments: **IEXEC, NESCOUO.**

Common blocks: ADDR0, ADDR4, HUGE, MIC, NESLEV, PARAM2, PARFDDA, and PBLDIM.

Data: None.

## 5.80 OUTSAV

Subroutine OUTSAV writes the output data arrays **(ALLARR (IRHUGE),** and **INTALL (IIHUGE))** to a save file for each domain.

OUTSAV is called from subroutines OUTPUT and SHUTDO.

Arguments: **IUTSAV, ALLARR, IRHUGE, INTALL, IIHUGE.**

Common blocks: None.

Data: None.

## 5.81 OUTTAP

Subroutine OUTTAP writes output data to a standard output file.

OUTTAP is called from subroutines CHKNST, INIT, and OUTPUT. It calls subroutines DELTATIM and EQUATE.

Arguments: **IUTL, NESCOU, IX, JX, KZZ**.

Common blocks: ADDR0, ADDR1, ADDR2, ADDR4, ADDRNN, ADDRV, HEADER, HUGE, MIC, NESLEV, NHCNS, NHCNST, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, V9COM, and V9COMC.

Data: None.

---- Program detail ----

OUTTAP first writes a record header that includes coordinate information, nested information, and constants that define the nonhydrostatic reference state. Next, the 3-D forecast arrays are written followed by the 2-D forecast arrays, and finally the 2-D terrestrial arrays. EQUATE is used throughout to ensure that arrays from nested domains having different grid size are properly written to the output file.

## 5.82 OVLCHK

Subroutine OVLCHK calculates which i, j boundary values come from another nest.

OVLCHK is called from subroutine INITNEST.

Arguments: **L.**

Common blocks: ADDRNN, HUGE, NESLEV, PARAM2, and PARFDDA.

Data: None.

## 5.83 PARAM

Subroutine PARAM defines the model parameters.

PARAM is called from the main program.

Arguments: None.

Common blocks: ADDR0, ADDR4, ADDR5, ADDR6, ADDR7, ADDR8, ARASCH, CFD, CFDDAGD, DEPAR2, HEADER, HUGE, NESLEV, PARAM2, PARAM3, PARFDDA, PMOIST, V9COM, and V9COMC.

Data:
**MMD** (12) /31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31/ number of days in month.
**QDCRIT** /3.0E-7/ precipitation threshold for moisture convergence (cb kg $K^{-1}$ $s^{-1}$) in the Kuo-Anthes scheme.
**VQRANG** /5.0E-4/ range limit on vertical eddy flux of water vapor (cb $s^{-1}$) in the Kuo-Anthes scheme.

[Summer Data Values]
**ALBD** (13) /18., 17., 19., 16., 12., 14., 8., 14., 25., 15., 55., 12., 20./ surface albedo (percent).
**SLMO** (13) /.05,.30, .15, .30, .30, .35, 1.0, .50, .02, .50, .95, .50, .15/ surface moisture availability as a fraction of one.
**THERIN** (13) /3., 4., 3., 4., 4., 5., 6., 6., 2., 5., 5., 5., 3./ thermal inertia (0.01 cal $cm^{-2}$ $K^{-1}$ $s^{-0.5}$).
**SFEM** (13) /.88, .92, .92, .93, .95, .95, .98, .95, .85, .92, .95, .95, .92/ surface emissivity at 9 micrometers.
**SFZO** (13) /50., 15., 12., 50., 50., 40., 0.01, 20., 10., 10., 5., 50., 15./ surface roughness length (cm).

**SFHC** (13) /18.5E5, 25.0E5, 20.8E5, 25.0E5, 29.2E5, 41.8E5, 99999., 29.2E5, 12.0E5, 99999., 99999., 29.2E5, 25.0E5/ surface heat capacity per unit volume (J $m^{-3}$ $K^{-1}$).

[Winter Data Values]

**ALBD** (13) /18., 23., 23., 17., 12., 14., 8., 14., 25., 70., 70., 12., 20./ surface albedo (percent).
**SLMO** (13) /.10, .60, .30, .60, .60, .70, 1.0, .75, .05, .90, .95, .50, .15/ surface moisture availability as a fraction of one.
**THERIN** (13) /3., 4., 4., 5., 5., 6., 6., 6., 2., 5., 5., 5., 3./ thermal inertia (0.01 cal $cm^{-2}$ $K^{-1}$ $s^{-0.5}$).
**SFEM** (13) /.88, .92, .92, .93, .95, .95, .98, .95, .85, .92, .95, .95, .92/ surface emissivity at 9 micrometers.
**SFZO** (13) /50., 5., 10., 50., 50., 40., 0.01, 20., 10., 10., 5., 50., 15./ surface roughness length (cm).
**SFHC** (13) /18.5E5, 25.0E5, 20.8E5, 25.0E5, 29.2E5, 41.8E5, 99999., 29.2E5, 12.0E5, 99999., 99999., 29.2E5, 25.0E5/ surface heat capacity per unit volume (J $m^{-3}$ $K^{-1}$).
**SCFX** (13) /.1, .25, .25, .2, .1, .22, 0., .15, .25, 0., 0., 0., 0./ scale factor for calculating the new albedo due to snow-cover effects.

---- Program detail ----

PARAM defines many of the model parameters regarding various options and contains extensive comments pertaining to model dimensions, FDDA options physical options, time and date, and output parameters. The sequence of the basic subroutine tasks is:

1. Set default values for namelist $OPARAM, $PPARAM, and $LPARAM options including nests.
2. Set FDDA default values for namelist $FPARAM options including nests.
3. Read in user-defined namelist values from $OPARAM, $LPARAM, $PPARAM, and $FPARAM
4. Define some FDDA parameters not included in the namelist.
5. Reset some of the coarse and nested domain options.
6. Specify the Julian date and compute the half-sigma levels.
7. Set the constants used in the model.
8. Set (or calculate) **DT** for the nonhydrostatic version and compute the time steps for radiation computations.
9. Compute the vertical interpolation coefficients for T and $q_v$, and determine the sigma levels for the lower bounds of low, middle, and high clouds.
10. Specify the coefficients for the sponge and nudging boundary conditions.

11. Read in transmissions if the ground temperature is to be calculated from the surface energy budget.
12. Printout parameter specifications for the model run.

## 5.84 PARAMR

Subroutine PARAMR sets constants for use with the EXMOISR routine.

PARAMR is called from subroutine PARAM.

Arguments: None.

Common blocks: JRG.

Data: None.

## 5.85 PRECIP

Subroutine PRECIP calculates the moisture properties of the cloud updraft for the deep (Arakawa-Schubert) and shallow convection schemes.

PRECIP is called from subroutines ARASCH and SHALLOW.

Arguments: **CD, KB, KBCON, KDIM, R, HC, HES, T, QE, QES, pw, qc, qrc, Z, LP, P, QKB, PCUT, C0, ZU, KNUM, KTOP.**

Common blocks: None.

Data: None.

## 5.86 PRINTSP

Subroutine PRINTSP prints out select values of the correction terms (for T, p*, U, and V) for the split-explicit scheme.

PRINTSP is called from subroutine SPSTEP2.

Arguments: **ARRAY, IX, JX, KX, IINC, JINC, KREF, NAME.**

Common blocks: None.

Data: None.

## 5.87 QSATGD

Subroutine QSATGD is an analysis-nudging FDDA routine that computes the model-simulated saturation mixing ratio for use in analysis nudging.

QSATGD is called from subroutines SOLVE1 and SOLVE3.

Arguments: **J, qsatf, TB, PSB, PPB, A, PTOP.**

Common blocks: ADDR4, HUGE, NESLEV, and PMOIST.

Data: None.

## 5.88 RDINIT

Subroutine RDINIT reads in the initial condition data.

RDINIT is called from subroutines INIT and INITNEST. It calls subroutines EQUATE and SHUTDO.

Arguments: **IUNITH, IUNITNH, IX, JX.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR4, ADDRNN, ASSEL, HEADER, HEADERC, HUGE, MIC, NESLEV, NHCNS, NHCNST, NONHYD, NONHYDB, PARAM2, PARAM3, PARFDDA, PBLDIM, SIZE.

Data: None.

## 5.89 SAVREAD

Subroutine SAVREAD reads a restart input file and puts the data into data arrays **(ALLARR(IRHUGE)** and **INTALL (IIHUGE)**) for a given domain.

SAVREAD is called from subroutine INITSAV.

Arguments: **ALLARR, IHUGE, INTALL, IIHUGE, IUTSAV.**

Common blocks: None.

Data: None.

## 5.90 SEAPRS

Subroutine SEAPRS computes sea-level pressure using the relationship: T1/T2 =(p1/p2)**(gamma*R/g), where gamma is the standard atmospheric lapse rate.

SEAPRS is called from subroutines FEEDBK, INITNEST, STOTNDI and STOTNDT.

Arguments: **T, TER, PS, SIG, IYY, JXX, KZZ, PTOP, slp, IDIM, JDIM, ts, t0.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.91 SETFD

Subroutine SETFD is a driver subroutine for calls to analysis-nudging and observation-nudging FDDA routines that operate on the entire grid, rather than one particular north-south slice, and can be performed prior to the major east-west J-loops in the SOLVE routines.

SETFD is called from subroutines SOLVE1 and SOLVE3. It calls subroutines BUFVDGD, ERROB, IN4DGD, IN4DOB, INTPSGD, NUDGD, and SETUPGD.

Arguments: **INEST, fdtim, [INFR], [IDARST].**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR3, ADDR4, ADDR5, ADDR6, ADDR7, ADDR8, ASSEL, CFD, CFDDAGD, HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: None.

---- Program detail ----

The sequence of the basic subroutine tasks is:

1. If necessary, read new analysis data for analysis nudging if **I4D** = 1, (IN4DGD).
2. Perform temporal and dot-point interpolation of p* (INTPSGD).
3. Set up temporal and spatial weighting functions for analysis nudging (SETUPGD).
4. Compute observed and model (forecast) vorticity difference if **IROT** = 1 (BUFVDGD).
5. Call analysis-nudging routines (NUDGD) for first north-south slice to perform necessary setup for subsequent north-south slices.
6. Read observation data for observation nudging if **I4DI** = 1, (IN4DGD).
7. Calculate difference between observation and model forecast at the observation locations within the specified time window (**TWINDO**) if **I4DI** = 1 (ERROB).

## 5.92 SETUPGD

Subroutine SETUPGD is an analysis-nudging FDDA routine that defines temporal, horizontal and vertical weighting functions used for analysis-nudging FDDA. It also computes model p* on dot points (**PSBD**) if nudging wind.

SETUPGD is called from subroutine SETFD. It calls subroutines DOTS, JULGMT, and UNITY.

Arguments: **PSB, psbdt, IN.**

Common blocks: ADDR4, ADDR5, ADDR6, ADDR7, CFD, CFDDAGD, HUGE, NESLEV, PARAM2, PARAM3, and PARFDDA.

Data: None.

---- Program detail ----

SETUPGD defines the time (**TFAC**), horizontal (**WDT, WCS, BLWDT, BLWCS**), and vertical (**ZFAC**) weighting functions for analysis-nudging. It also computes the horizontal weighting factor for moisture nudging, **WQ**, based on observed precipitation, if **IMOISP** = 1.

## 5.93 SFCRAD

Subroutine SFCRAD calculates the shortwave and longwave irradiances incident on the surface.

SFCRAD is called from subroutines BLKPBL and HIRPBL. It calls subroutine TRANSM.

Arguments: **IYY, JXX, J, INEST.**

Common blocks: ADDR1, ADDR2, ADDR4, HUGE, MIC, NESLEV, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data:
**CLW** (3) /0.06, 0.22, 0.26/ coefficients used to compute cloud-cover fraction for low, middle, and high clouds.
**CSW** /4.183E+6/ heat capacity of water per unit volume (J $m^{-3}$ $K^{-1}$).
**LAMG** (4) /1.407E-8, -1.455E-5, 6.29E-3, 0.16857/ constants used to compute soil thermal conductivity in moist soil.
**TAC** (3) /0.98, 0.85, 0.80/ absorption transmission coefficients for low, middle, and high clouds.
**TSC** (3) /0.80, 0.60, 0.48/ scattering transmission coefficients for low, middle, and high clouds.

---- Program detail ----

The sequence of the basic subroutine tasks is:

1. Determine the cloud fractions of low, middle, and high clouds and calculate the amount of precipitable water.
2. From (1) above, calculate the long-wave irradiance incident from the ground.
3. Calculate the short-wave irradiance from scattering transmissivities obtained from TRANSM.

4. Compute the slab thermal capacity based on soil heat capacity, soil conductivity, and moisture availability.

## 5.94 SHALLCU

Subroutine SHALLCU is a small driver subroutine for the shallow convection scheme (SHALLOW) and it sets up a few of the variables.

SHALLCU is called from subroutines SOLVE1 and SOLVE3. It calls subroutine SHALLOW.

Arguments: **tten*, qten*, J, NUMNES, SCR1, SCR2, ISTART, IEND.**

Common blocks: ADDR1, ADDR2, ADDR4, HUGE, NESLEV, PARAM2, PARAM3, PARFDDA, and PMOIST.

Data: None.

## 5.95 SHALLOW

Subroutine SHALLOW is the shallow cumulus parameterization routine. Although it is the main routine for the shallow convection scheme, it calls several other related subroutines.

SHALLOW is called from subroutine SHALLCU. It calls subroutines ARAOUTS, CLOUDWS, ENTRS, HEIPRE, KERHELS, MAXIM, MINIM, MOIENE, PRECIP, and ZUNC.

Arguments: **TI, QI, Z1, TIO, QIO, PIO, KLEV, PRE, PI, outtem, outq, DTIME, KBMAX, PCUT, C0, PSUR, ierrt, RADS, xmb.**

Common blocks: HUGE, NESLEV, PARAM2, PARAM3, PARFDDA, and PMOIST.

Data: None.

---- Program Detail ----

SHALLOW computes the moisture cycle using a shallow cumulus parameterization scheme. The sequence of the basic subroutine tasks is:

1. Initialize some of the arrays.
2. Compute the heights of the sigma levels (HEIPRE).
3. Calculate the moist static energy **HE** and the saturation moist static energy **HES** (MOIENE).
4. Determine the level with the highest moist static energy (MAXIM) and cloud base.
5. Begin in-cloud calculations.
   a) Calculate moist static energy and cloud top (ENTRS).
   b) Compute normalized mass-flux profile (ZUNC).

c) Calculate the moisture properties (non-precipitating) in the shallow updraft (PRECIP).

d) Calculate efficiency of generation of kinetic energy for the shallow updraft (CLOUDWS).

6. Begin new in-cloud and modified environmental calculations.

a) Calculate shallow cloud effect on environment (KERHELS).

b) Call HEIPRE and MOIENE for heights and moist static energy.

c) Call ENTRS, ZUNC, PRECIP, and CLOUDWS for the moist static energy (in cloud), normalized mass-flux profile, amount of moisture in the shallow updraft, and updraft cloud-work function.

7. Calculate feedback effects of clouds to the environment (ARAOUTS).

## 5.96 SHUTDO

Subroutine SHUTDO shuts down a forecast if running out of CPU time or domains are not assigned correctly. A save file is written (calls OUTSAV).

SHUTDO is called from the main program and subroutine RDINIT. It calls subroutines ADDRX1C and OUTSAV.

Arguments: None.

Common blocks: ADDR0, ADDR4, HUGE, MIC, NESLEV, PARAM2, PARFDDA, and PBLDIM.

Data: None.

## 5.97 SINT

Subroutine SINT interpolates nested-grid values from coarse-grid values using a positive-definite advection scheme.

SINT is called from subroutine EXAINT.

Arguments: **xf*, N, M, N1STAR, N1END, N2STAR, N2END.**

Common blocks: DEPAR2 and HUGE.

Data: **EP** /1.0E-10/ machine precision value.

## 5.98 SINTX

Subroutine SINTX interpolates nested-grid boundary values from coarse-grid values (in the x-direction) using a positive-definite advection scheme.

SINTX is called from subroutines STOTNDI and STOTNDT.

Arguments: **xf*, N, M, N1STAR, N1END, JXWANT, ICOMPS, ICRSDOT.**

Common blocks: DEPAR2 and HUGE.

Data: **EP** /1.0E-10/ machine precision value.

## 5.99 SINTY

Subroutine SINTY interpolates nested-grid boundary values from coarse-grid values (in the y-direction) using a positive-definite advection scheme.

SINTY is called from subroutines STOTNDI and STOTNDT.

Arguments: **xf*, N, M, IYWANT, N2STAR, N2END, ICOMPS, ICRSDOT.**

Common blocks: DEPAR2 and HUGE.

Data: **EP** /1.0E-10/ machine precision value.

## 5.100 SLAB

Subroutine SLAB calculates the ground temperature tendency from the residual of the surface energy budget. For points over water, the ground temperature tendency is zero; if snow cover effects are considered, the ground temperature is limited to less than 273.16 K for grid points having snow cover.

SLAB is called from subroutines BLKPBL and HIRPBL.

Arguments: **DELTSM, J, INEST.**

Common blocks: ADDR1, ADDR2, ADDR4, HUGE, MIC, NESLEV, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: None.

## 5.101 SMTHER

Subroutine SMTHER is a 2-pass smoother-desmoother routine designed to dampen the short wavelength components.

SMTHER is called from subroutines BLW and FEEDBK.

Arguments: **slab*, IS1, IS2, NPASS, IST, IE, JST, JE.**

Common blocks: None.

Data: None.

## 5.102 SOLAR1

Subroutine SOLAR1 computes the solar declination angle from the Julian date.

SOLAR1 is called from subroutines INIT, INITSAV, SOLVE1 and SOLVE3.

Arguments: **XTIME, IEXEC.**

Common blocks: HUGE, NESLEV, and PARAM3.

Data: None.

## 5.103 SOLVE1 [multi-tasked]

Subroutine SOLVE1 computes the tendencies of the prognostic variables for the hydrostatic version (**INHYD** = 0) of MM5.

SOLVE1 is called from the main program and from subroutines NSTLEV1, NSTLEV2, and NSTLEV3. It calls subroutines BLKPBL, BUFSLGD, CUPARA1, CUPARA2, CUPARA3, DECPU, DIFFU, DIFFUT, EXMOISR, EXMOISS, HADV, HIRPBL, MDOTS, NUDGD, NUDGE, NUDOB, QSATGD, SETFD, SHALLCU, SOLAR1, SPONGE, and VADV.

Arguments: **iexec*, INEST, and NN.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR3, ADDR4, ADDR5, ADDR6, ADDR7, ADDR8, ASSEL, CFD, CFDDAGD, HUGE, MIC, NESLEV, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: **DTMITE** /25./ time constant (s) used to compute PBL iteration loop **MITERO**.

---- Program Detail ----

SOLVE1 and SOLVE3 are the main forecast subroutines of MM5. SOLVE1 computes the tendencies of the prognostic variables p*, U, V, $q_v$ and other moisture variables, and T. For each time step, SOLVE1 is used one time for the large domain and **IRAX** (the ratio between the coarse and fine grids) times for a nested domain. In a case where a nested domain is nested within a bigger nest, the biggest nest is considered the large domain. The sequence of the basic subroutine tasks is:

1. Decouple the boundary values of the **UA** and **VA** arrays.
2. Multiply **UA** and **VA** by the inverse of the mapscale factor and initialize some arrays to zero for time t+1.
3. Calculate the boundary values of U, V, $q_v$, p*, T, $q_r$, $q_c$, ice ($q_i$), and snow ($q_{ni}$) for time t+1, and compute QDOT at the boundaries.
4. If employing FDDA (**IFDDA** = 1), call SETFD for initial setup.
5. Compute the pressure tendency and vertical sigma velocity (**QDOT**).
6. Apply sponge or relaxation boundary conditions to the pressure tendency (SPONGE or NUDGE).
7. If **IFDDA** = 1, compute the nudging term for p* (NUDOB, NUDGD).
8. Get the forecast pressure at time t+1 (add the tendency) and compute the Bleck noise parameters.
9. Compute geopotential height at the half-sigma levels.
10. Decouple U and V (DECPU), and compute the horizontal diffusion coefficients. Calculate horizontal diffusion and horizontal advection of momentum.
11. Decouple temperature and moisture.
12. Get the temperature tendency:
    a) Compute horizontal and vertical advection (HADV and VADV).
    b) Compute the adiabatic and radiational (LWRAD and SWRAD) cooling terms.
13. Get contributions to the moisture tendency from cumulus parameterizations and advection of water vapor.
    a) If ICUPA = 2, use Anthes-Kuo scheme (CUPARA1).
    b) If ICUPA = 3, use Grell scheme (CUPARA2).
    c) If ICUPA = 4, use Arakawa Schubert scheme (CUPARA3).
14. Compute some horizontal and vertical flux divergence terms needed for the explicit moisture scheme (HADV and VADV).
15. Calculate diffusion (DIFFUT) for moisture and temperature.
16. Compute PBL fluxes (BLKPBL if **IBLTYP** = 1, or HIRPBL if **IBLTYP** = 2).
17. Get the contribution to the moisture tendency from shallow cumulus (call SHALLCU if **ISHALLO** = 1).
18. Add horizontal diffusion and PBL tendencies for T and $q_v$ to **TTEN** and **QVTEN**.
19. Apply sponge or relaxation boundary conditions to T and $q_v$.
20. If **IFDDA** = 1, get FDDA nudging terms for temperature and moisture from observations (NUDOB) and/or from an analysis (NUDGD).
21. Call EXMOISR or EXMOISS for contributions to the moisture tendency from explicit moisture schemes.

22. Get the forecast T, $q_v$, $q_c$, $q_r$, and $q_i$ at time t+1 (add the tendencies).
23. Get the U and V tendencies.
    a) If **IFDDA** = 1, get FDDA nudging terms for U and V wind components from observations (NUDOB) and/or from an analysis (NUDGD).
    b) Compute the pressure gradient, Coriolis, and geopotential gradient terms.
    c) Compute the vertical flux divergence terms (**VADV**), and add the diffusion and PBL tendencies for U and V to **UTEN** and **VTEN**.
    d) Apply sponge or relaxation boundary conditions to U and V.
    e) Get the forecast U and V at time t+1 (add the tendencies).
24. Perform time-smoothing operations. Update interior of arrays using Asselin filter.
25. Compute the non-convective precipitation and increment the forecast time.
26. Printout noise parameters and number of convective points.
27. Recalculate solar declination angle if forecast time is greater than 24 hours (SOLAR1).

----User Note----

Set **INHYD** = 0 for this option

## 5.104 SOLVE3 [multi-tasked]

Subroutine SOLVE3 computes the tendencies of the prognostic variables for the nonhydrostatic version (**INHYD** = 1) of MM5.

SOLVE3 is called from the main program and from subroutines NSTLEV1, NSTLEV2, and NSTLEV3. It calls subroutines BLKPBL, BUFSLGD, CUPARA1, CUPARA2, CUPARA3, DECPU, DIFFU, DIFFUT, DOTS, EXMOISR, EXMOISS, HADV, HIRPBL, NUDGD, NUDGE, NUDOB, QSATGD, SETFD, SHALLCU, SOLAR1, SOUND, and VADV.

Arguments: **iexec*, INEST, and NN.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR3, ADDR4, ADDR5, ADDR6, ADDR7, ADDR8, ASSEL, CFD, CFDDAGD, HUGE, MIC, NESLEV, NHCNS, NHCNST, NONHYD, NONHYDB, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: **DTMITE** /25./ time constant (s) used to compute PBL iteration loop **MITERO**.

---- Program Detail ----

SOLVE3 and SOLVE1 are the main forecast subroutines of MM5. SOLVE3 computes the tendencies of the prognostic variables p', U, V, W, $q_v$, T and other moisture variables. For each time step, SOLVE3 is used one time for the large domain and **IRAX** (the ratio between the coarse and fine grids) times for a nested domain. In a case where a nested domain is nested within a bigger nest, the biggest nest is considered the large domain. The sequence of the basic subroutine tasks is:

1. Decouple the boundary values of the **UA** and **VA** arrays.
2. Multiply **UA** and **VA** by the inverse of the mapscale factor and initialize some arrays to zero for time t+1.
3. Calculate the boundary values of U, V, $q_v$, p*, T, $q_r$, $q_c$, ice ($q_i$), and snow ($q_{ni}$) for time t+1.
4. If employing FDDA (**IFDDA** = 1), call SETFD for initial setup.
5. Calculate **QDOT** and the total divergence **DIVX**.
6. Decouple U and V (DECPU), and compute the horizontal diffusion coefficients. Calculate horizontal diffusion and horizontal advection of momentum.
7. Decouple temperature, p′, and moisture.
8. Get the temperature, p′, and W tendencies:
   a) Compute horizontal and vertical advection (HADV and VADV).
   b) Compute the adiabatic and radiational (LWRAD and SWRAD) cooling terms for T and buoyancy term for W.
   c) Remove the total divergence term in temperature, moisture, W, and p′.
9. Get contributions to the moisture tendency from cumulus parameterizations and advection of water vapor:
   a) If **ICUPA** = 2, use Anthes-Kuo scheme (CUPARA1).
   b) If **ICUPA** = 3, use Grell scheme (CUPARA2).
   c) If **ICUPA** = 4, use Arakawa-Schubert scheme (CUPARA3).
10. Compute some horizontal and vertical flux divergence terms needed for the explicit moisture scheme (HADV and VADV).
11. Calculate diffusion (DIFFUT) for moisture, T, and W.
12. Compute PBL fluxes (BLKPBL if **IBLTYP** = 1, or HIRPBL if **IBLTYP** = 2).
13. Get the contribution to the moisture tendency from shallow cumulus (call SHALLCU if **ISHALLO** = 1).
14. Add horizontal diffusion and PBL tendencies for T and $q_v$ to **TTEN** and **QVTEN**.
15. Apply relaxation boundary conditions to T and $q_v$.
16. If **IFDDA** = 1, get FDDA nudging terms for temperature and moisture from observations (NUDOB) and/or from an analysis (NUDGD).
17. Set **PPTENS** and **WTENS** for SOUND.

18. Call EXMOISR or EXMOISS for contributions to the moisture tendency from explicit moisture schemes.
19. Get the forecast T, $q_v$, $q_c$, $q_r$, $q_{ni}$, and $q_i$ at time t+1 (add the tendencies).
20. If **IFDDA** = 1, get FDDA nudging terms for U and V wind components from observations (NUDOB) and/or from an analysis (NUDGD).
21. Compute the Coriolis terms.
22. Compute the vertical flux divergence terms (VADV), and add divergence term, diffusion and PBL tendencies for U and V to **UTEN** and **VTEN**.
23. Apply relaxation boundary conditions to U and V.
24. Set **UTENS** and **VTENS** for SOUND.
25. Perform time-smoothing operations on temperature and moisture. Update interior of arrays using asselin filter.
26. Call SOUND which considers the effects of soundwaves for short time steps and includes the interaction of p and momentum.
27. Compute the non-convective precipitation and increment the forecast time.
28. Recalculate solar declination angle if forecast time is greater than 24 hours (SOLAR1).

----User Note----

Set **INHYD** = 1 for this option.

## 5.105 SOUND [multi-tasked]

Subroutine SOUND considers the effects of soundwaves for short time steps and includes the interaction of pressure and momentum.

SOUND is called from subroutine SOLVE3.

Arguments: **IYY, JXX, ub*, vb*, tb*, PSADOT, ua*, va*, ta*, QVB, PSA, HT, MSFD, MSFX, DX, DTL, INEST, KTAU.**

Common blocks: ASSEL, HUGE, NESLEV, NHCNS, NHCNST, NONHYD, NONHYDB, PARAM2, PARAM3, PARFDDA, PMOIST and UPRAD.

Data: None.

---- Program detail ----

SOUND, which is needed only for the nonhydrostatic version of MM5, considers the effects on soundwaves for short time steps. The sequence of the basic subroutine tasks is:

1. Set some constants and calculate initial arrays for the short time step.
2. Begin short time steps.

3. Calculate the divergence damping term (added temporarily) and the U, V pressure gradients.
4. Establish the U and V boundary conditions and subtract the divergence damping term.
5. Begin the main **I, J** (do 400) loop.
6. Begin the vertical momentum/pressure calculation, column by column.
7. Calculate implicit W equation scheme coefficients **AA**, **B**, and **C** (Ikawa method.)
8. Get RHS for implicit scheme and calculate vertical momentum.
9. Calculate pressure and add the pressure correction (dp'/dt) to temperature.
10. End main **I**, **J** loop.
11. Establish W and p' boundary conditions.
12. End short time steps.
13. Transfer **XXA** values to **XXB** and apply time filter.

## 5.106 SOUNDD

Subroutine SOUNDD determines the downdraft thermodynamic properties for the Arakawa-Schubert convection scheme and calculates the normalized mass flux.

SOUNDD is called from subroutine ARASCH.

Arguments: **HE, hcd, Z, KDIM, KMIN, PW, LP, HES, QES, QE, T, itest*, qrcd*, pwd*, zd, edt, KDET, KNUM.**

Common blocks: None.

Data: None.

## 5.107 SPCORT [multi-tasked]

Subroutine SPCORT adds the correction term **DDSUM** to temperature and p* for the split-explicit scheme.

SPCORT is called from subroutine SPLITF.

Arguments: **ta*, psa*, tb*, psb*, DDSUM, AM, AN, GNUHF, KL, ILXM, JLXM, INEST.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.108 SPCORU [multi-tasked]

Subroutine SPCORU adds the correction term **DHSUM** to U and V for the split- explicit scheme.

SPCORU is called from subroutine SPLITF.

Arguments: **ua*, va*, PSDOT, ub*, vb*, MSFD, DHSUM, ZMATX, GNUHF, DX2, KL, ILX, JLX, INEST.**

Common blocks: HUGE and NESLEV.

Data: None.

---

## 5.109 SPDIFF [multi-tasked]

Subroutine SPDIFF is a small routine used to compute the difference between arrays **IN1** and **IN2** for the split-explicit scheme.

SPDIFF is called from subroutine SPLITF.

Arguments: **out, IN2, IN1, IY, JX, KZ, NT2, NT1, NT.**

Common blocks: HUGE and NESLEV.

Data: None.

---

## 5.110 SPDIVG [multi-tasked]

Subroutine SPDIVG calculates the divergence from U and V in vertical mode space for the split-explicit scheme.

SPDIVG is called from subroutine SPINIT and SPLITF.

Arguments: **u*, v*, deld*, ZMATXR, MSFD, MSFX, DX2, IX, JX, KL, IMIN, JMIN, IMAX, JMAX.**

Common blocks: HUGE and NESLEV.

Data: None.

---

## 5.111 SPGEOP [multi-tasked]

Subroutine SPGEOP calculates the geopotential in vertical mode space for split-explicit scheme.

SPGEOP is called from subroutine SPINIT and SPLITF.

Arguments: **T, PS, delh, BZ, CZ, SIGMAH, PD, PTOP, KL, ILX, JLX.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.112 SPGRAD [multi-tasked]

Subroutine SPGRAD computes the gradient of geopotential for the split-explicit scheme.

SPGRAD is called from subroutine SPSTEP2.

Arguments: **PHI, gx, gy, MSFX, DX2, KL, ILX, JLX.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.113 SPINIT

Subroutine SPINIT is an initialization routine for the split-explicit scheme. The purpose of the split-explicit scheme is to minimize the computation cost while, at the same time, maintaining model numerical stability. The momentum, pressure and temperature tendencies are computed on 2 time scales with the fastest gravity modes handled with a short time step. The technique still allows the bulk of the model tendency calculations to be performed using a longer time step, and thus no significant increase in the number of required computations.

SPINIT is called from the main program and from subroutines CHKNST, INITSAV, NSTLEV1, NSTLEV2, and NSTLEV3. It calls subroutines SPDIVG, SPGEOP, and VMODES.

Arguments: **PTOP, SIGMA, KV1, IX, JX.**

Common blocks: ADDR1, ADDR2, ADDR4, ADDRSP, blank, HUGE, and NESLEV.

Data: None.

---- Program Detail ----

SPINIT initializes a number of arrays for the split-explicit scheme and performs calculations at the initial time. The sequence of the basic subroutine tasks is:

1. Calculate a mean surface pressure and mean vertical temperature profile (**PS** and **TBARH**) for use in VMODES.
2. Determine the vertical modes of the model (VMODES).
3. Compute the **BZ** and **CZ** arrays needed for transformation of temperature from sigma space to vertical mode space for calculation of geopotential in vertical mode space.

4. Calculate the divergence in vertical mode space (SPDIVG).
5. Calculate the geopotential in vertical mode space (SPGEOP).

## 5.114 SPLITF [multi-tasked]

Subroutine SPLITF is a driver subroutine for split-explicit time integration.

SPLITF is called from the main program and from subroutines NSTLEV1, NSTLEV2, and NSTLEV3. It calls subroutines SPCORT, SPCORU, SPDIFF, SPDIVG, SPGEOP, SPSTEP2, and XTDOT.

Arguments: **GNUHF, INEST.**

Common blocks: ADDR1, ADDR2, ADDR4, ADDRSP, blank, HUGE, MIC, NESLEV, and PBLDIM.

Data: None.

---- Program Detail ----

SPLITF calls routines that compute the time derivative of divergence and geopotential and add the correction terms to T, p*, U, and V. The sequence of the basic subroutine tasks is:

1. Interpolate p from cross point to dot point (XTDOT).
2. Call SPDIVG twice to calculate the divergence **DELD** at time t (use **UA** and **VA**) and at time t-1 (use **UB** and **VB**).
3. Call SPDIFF to get the **DELD** difference between time t and t-1.
4. Call SPGEOP twice to calculate the geopotential **DELH** at time t (use **TA** and **PSA**) and at time t-1 (use **TB** and **PSB**).
5. Call SPDIFF to get the **DELH** difference between time t and t-1.
6. Perform time integration for the small time steps for fast gravity wave modes (SPSTEP2) and return the correction terms **DDSUM** and **DHSUM** that have been summed over the small time steps.
7. Add the correction **DDSUM** to temperature and p* (SPCORT).
8. Add the correction **DHSUM** to U and V (SPCORU).

## 5.115 SPONGE

Subroutine SPONGE applies the sponge boundary conditions to the local tendency term, **FTEN**. The final tendency is computed using a weighted average of large-scale (or observed) and model-predicted tendencies.

SPONGE is called from subroutine SOLVE1.

Arguments: **IP, WG, ften*, FEBT, FWBT, FNBT, FSBT, KD, IE, JE, J.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.116 SPSTEP2 [multi-tasked]

Subroutine SPSTEP2 is a time integration routine for the split-explicit scheme.

SPSTEP2 is called from subroutine SPLITF. It calls subroutines DIVG, PRINTSP, and SPGRAD.

Arguments: **ddsum, dhsum, [DELD], [DELH], HBAR, PSA, PSDOT, MSFD, MSFX, [WORK], DX2, DTAU, M, IX, JX, ILX, JLX, ILXM, JLXM, INEST.**

Common blocks: HUGE and NESLEV.

Data: None.

---- Program detail ----

SPSTEP2 performs time integration for the small time steps for fast gravity wave modes in the split explicit scheme. It returns the correction terms for temperature and p* (**DDSUM**) and U and V (**DHSUM**) that have been summed over the small time steps. It calls SPGRAD to compute the gradient of geopotential needed for **DHSUM** calculations, and it calls DIVG to compute the divergence needed for the **DDSUM** calculations.

## 5.117 STOTNDI [multi-tasked]

Subroutine STOTNDI interpolates and stores nested-grid values from coarse-grid variables at the nested-grid boundaries for time t-1. These values are needed later in STOTNDT.

STOTNDI is called from the main program and from subroutines NSTLEV1, NSTLEV2, and NSTLEV3. It calls subroutines EXAINT, MDOTS, SEAPRS, SINTX, and SINTY.

Arguments: **INEST, IYYN, JXXN, IYY, JXX.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR4, ADDRN1, ADDRN2, ADDRN3, ADDRN4, ADDRNN, DEPAR2, HUGE, MIC, NESLEV, NNNHYD, NNNHYDB, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: None.

---- Program detail ----

STOTNDI interpolates values of p*, T, $q_v$, $q_c$, $q_r$, $q_i$, $q_{ni}$, U, V, and (if using the nonhydrostatic version) vertical velocity and the pressure perturbation from the coarse grid variables at the nest interface. The interpolation of the variables at time t-1 is accomplished using a positive-definite advection scheme (SINTX for the x-direction and SINTY for the y-direction).

## 5.118 STOTNDT [multi-tasked]

Subroutine STOTNDT interpolates nested-grid values from coarse-grid variables at the nested-grid boundaries for time t+1. It then also uses the interpolated values for time t-1 (obtained from STOTNDI) to calculate the nested-grid boundary tendencies.

STOTNDT is called from subroutines NSTLEV1, NSTLEV2, and NSTLEV3. It calls subroutines EXAINT, MDOTS, SEAPRS, SINTX, and SINTY.

Arguments: **INEST, IYYN, JXXN, IYY, JXX.**

Common blocks: ADDR0, ADDR1, ADDR2, ADDR4, ADDRN1, ADDRN2, ADDRN3, ADDRN4, ADDRNN, DEPAR2, HUGE, MIC, NESLEV, NNNHYD, NNNHYDB, NONHYD, PARAM2, PARAM3, PARFDDA, PBLDIM, and PMOIST.

Data: None.

---- Program detail ----

STOTNDT interpolates values of p*, T, $q_v$, $q_c$, $q_r$, $q_i$, $q_{ni}$, U, V, and (if using the nonhydrostatic version) vertical velocity and the pressure perturbation from the coarse grid variables at the nested interface. The interpolation of the variables at time t+1 is accomplished using a positive-definite advection scheme (SINTX for the x-direction and SINTY for the y-direction. It then uses these values along with the values for time t-1 to calculate the nested-grid tendencies at the interface.

## 5.119 SUBCH

Subroutine SUBCH adjusts the corner points of a nest within a moving nest.

SUBCH is called from subroutine INITNEST.

Arguments: **NUMNE, NUMN, NESTII, NESTJJ, ISOUTHO, JWESTO.**

Common blocks: ADDRN4, ADDRNN, HUGE, and NESLEV.

Data: None.

## 5.120 SWRAD

Subroutine SWRAD calculates the shortwave component of the atmospheric radiative temperature tendencies and surface flux.

SWRAD is called from subroutines SOLVE1 and SOLVE3.

Arguments: **J**

Common blocks: ADDR1, ADDR2, ADDR4, HUGE, NESLEV, NHCNS, NHCNST, NONHYD, PARAM3, and RADIAT.

Data: **ALBTAB** (4, 5) look-up table of cloud albedo as a function of liquid water path and zenith angle.
**ABSTAB** (4, 5) look-up table of cloud absorption as a function of liquid water path and zenith angle.
**XMUVAL** (4) cosine (zenith angle) for look-up tables.

----User Note----

Set **IRDDIM** = 1 for this option.

---- Program Detail ----

1. Preset column variables and calculates solar zenith angle.
2. Calculate various path lengths for each model level.
3. Calculate downward flux for each model level by integration from top down. Flux depleted by absorption and scattering in clear air and clouds.
4. Absorption in each layer determines heating rate.
5. Save downward shortwave flux at surface (**GSW**) for ground heat budget.

## 5.121 TMASS

Subroutine TMASS computes the total mass (kg) of dry air and water substance within the domain.

TMASS is called from subroutines CONMAS, INIT, and INITNEST.

Arguments: **tdrym, tqmass, tvmass, tcmass, trmass, IL, JL, KL, IDRY, IMOIST, PSA, QVA, QCA, QRA, DSIGMA, DX, G.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.122 TRANSM

Subroutine TRANSM uses a lookup table to compute the shortwave transmissivity. The values (which are a function of path length and precipitable water at standard atmospheric pressure) are interpolated and then adjusted for actual pressure.

TRANSM is called from subroutine SFCRAD.

Arguments: **[PATH], PRW, ftabs, ftscat, fbscat, [PSB], J, IYY.**

Common blocks: HUGE, MESLEV, NONHYD, and PARAM3.

Data: None.

## 5.123 UNITY

Subroutine UNITY sets the values of a 2-D array **SLAB1(MIX, MJX)** TO 1.

UNITY is called from subroutine SETUPGD.

Arguments: **slab1, IMAX, JMAX.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.124 VADV

Subroutine VADV computes the vertical advection (flux-divergence) terms for **QVA**, **QCA, QRA, UA, VA,** and nonhydrostatic W and p′. The interpolation is based on linear potential temperature variation for temperature and logarithmic variation for $q_v$.

VADV is called from subroutines SOLVE1 and SOLVE3.

Arguments: **KZZ, ften*, FA, QDOT, MSF, PSA, [F], J, IND, IN.**

Common blocks: ADDR4, HUGE, NESLEV, and PARAM3.

Data: None.

## 5.125 VCHEKE

Subroutine VCHEKE checks that eigenvalues are real and positive valued for computations of **HBAR** for the split-explicit scheme.

VCHEKE is called from subroutine VMODES.

Arguments: **ER, EI, NK, numerr*, ANAME.**

Common blocks: None.

Data: None.

## 5.126 VCHEKI

Subroutine VCHEKI checks for linear algebra errors in the computations of matrix **ZMATX** for the split-explicit scheme.

VCHEKI is called from subroutine VMODES.

Arguments: **IER, numerr*, ANAME.**

Common blocks: None.

Data: None.

## 5.127 VCHEKT

Subroutine VCHEKT checks if the mean vertical temperature profile **TBARH** is stable for the split-explicit scheme.

VCHEKT is called from subroutine VMODES.

Arguments: **TBARH, SIGMAH, SIGMAF, XKAPPA, PT, PD, NK, numerr*.**

Common blocks: None.

Data: None.

## 5.128 VMODES

Subroutine VMODES determines the vertical modes of the model for the split-explicit scheme.

VMODES is called from subroutine SPINIT. It calls subroutines INVMTX, VCHEKE, VCKEKI, VCHEKT, VMULTM, VNORML, VORDER, VPRNTM, VPRNTV, and VTLAPS.

Arguments: **LSTAND, SIGMAF, KV1**

Common blocks: ADDR1, ADDR4, ADDRSP, blank, HUGE, and NESLEV.

Data: **LPRINT /.FALSE./ true** denotes linearization about standard atmosphere, **false** denotes linearization about horizontal mean of data.

---- Program Detail ----

VMODES determines the equivalent depths of the vertical modes of the model and also calculates matrices needed for transformation from sigma space to vertical mode space. The sequence of the basic subroutine tasks is:

1. Set up the arrays describing the vertical structure and compute a standard atmospheric temperature environment (VTLAPS) if **LSTAND** = **TRUE**.
2. Check if the vertical temperature structure **TBARH** is stable (VCHEKT).
3. Define diagonal elements for determination of the thermodynamic matrix.
4. Compute the transform from divergence to sigma dot (VMULTM).
5. Define matrices for linearized determination of geopotential--compute the hydrostatic matrices **HYDROS** and **HYDROC**.
6. Get equivalent depths of vertical modes (**HBAR**) and determine the matrix **ZMATX** which is used for transforming divergence from sigma to vertical mode space.
7. Call VCHEKI and VCHEKE to check for linear algebra errors in the **ZMATX** computations and for eigenvalue errors in the **HBAR** computations.
8. Adjust the components of **HBAR** so that they are ordered from largest value to smallest value (VORDER).
9. Normalize the columns of **ZMATX** such that the component with the largest absolute value is positive and the sum of the mass-weighted squares is one (VNORML).
10. Call INVTMX to compute the inverse of **ZMATX** (**ZMATXR**) and **HYDROS** (**HYDROR**).
11. If **LPRINT** is **TRUE**, print out some VMODES arrays (**VPRNTV** and **VPRNTM**).

## 5.129 VMULTM

Subroutine VMULTM calls four entries which perform matrix multiplication, addition, and subtraction operations for the split-explicit scheme.

VMULTM is called from subroutine VMODES.

Arguments: **a1*, A2, A3, NK.**

Common blocks: None.

Data: None.

## 5.130 VNORML

Subroutine VNORML normalizes the columns of the matrix **ZMATX** for the split- explicit scheme such that the component with the largest absolute value is positive and the sum of the mass-weighted squares is one.

VNORML is called from subroutine VMODES.

Arguments: **z*, S, NK, NK1.**

Common blocks: None.

Data: None.

## 5.131 VORDER

Subroutine VORDER adjusts the components of HBAR (equivalent depths of vertical modes) for the split-explicit scheme so that they are ordered from largest value to smallest value.

VORDER is called from subroutine VMODES.

Arguments: **z*, hbar*, [WZ], [WH], NK.**

Common blocks: None.

Data: None.

## 5.132 VPRNTM

Subroutine VPRNTM prints some 2-D VMODES arrays for the split-explicit scheme.

VPRNTM is called from subroutine VMODES.

Arguments: **A, N1, N2, NAM.**

Common blocks: None.

Data: None.

## 5.133 VPRNTV

Subroutine VPRNTV prints some 1-D VMODES arrays for the split-explicit scheme.

VPRNTV is called from subroutine VMODES.

Arguments: **A, N, NAM.**

Common blocks: None.

Data: None.

## 5.134 VTLAPS

Subroutine VTLAPS computes a standard atmosphere temperature environment for the split-explicit scheme (if **LSTAND** = **TRUE**).

VTLAPS is called from subroutine VMODES.

Arguments: **t, SIGMA, R, PT, PD, NK.**

Common blocks: None.

Data: None.

## 5.135 VTRAN

Subroutine VTRAN transposes the x<-->y orientation of a 2-D array.

VTRAN is called from subroutine MAPSMP.

Arguments: **fyx*, IMAX, JMAX.**

Common blocks: None.

Data: None.

## 5.136 XTDOT

Subroutine XTDOT interpolates P from cross point to dot point for the split-explicit scheme.

XTDOT is called from subroutine SPLITF.

Arguments: **PX, pd, NI, NJ, NI1, NJ1.**

Common blocks: HUGE and NESLEV.

Data: None.

## 5.137 ZUNC

Subroutine ZUNC computes the normalized mass-flux profiles for the deep (Arakawa-Schubert) and shallow convection schemes.

ZUNC is called from subroutines ARASCH and SHALLOW.

Arguments: **KBEG, zu, KB, R, Z, KDIM, LP, KNUM, KTOP**

Common blocks: None.

Data: None.

### 5.138 ZX4LP

Subroutine ZX4LP is used for the Arakawa-Schubert convection scheme to solve a linear programming problem.

ZX4LP is called from subroutine ARAMB.

Arguments: **A, IA, B, C, N, M1, M2, S, psol, DSOL, RW, IW, ier.**

Common blocks: WORKSP.

Data: None.