Structure of the CATHY model (31/08/12)

Note: this flowchart does not consider EnKF/SIR DA, soil deformation, chord slope, and lookup table options (i.e., only the cases ENKF=false, IPEAT=0, KSLOPE=0, and IVGHU≠-1 are considered here)

CATHY.H - *setting of array dimensions for proper memory allocation during code compilation*
cathy.fnames - *names of the input and ouput files (see also block_data.f and openio.f)*

CATHY_MAIN - *main program*
- OPENIO - *define and open I/O files*
- DATIN - *read in (some of the) input data*
  - if DEM=true
    - RAST_INPUT_DEM - *read in DEM (raster map of surface topography)*
    - RAST_INPUT_LZ - *read in "zones" (material types) raster map*
    - if FL3D=true and IVERT=true
      - RAST_INPUT_DEM - *read in raster map of catchment base (eg, bedrock topography)*
    - RAST_INPUT_LZ - *read in lakes raster map*
    - INDEX_DEM - *numbering of catchment cells (index matrices INDCEL and INDCELWL)*
    - if FL3D=true and IVERT=true
      - ASSIGN_DEM - *assign DEM values to "elevation" storage arrays*
      - TRIANGOLI - *triangulate DEM surface cells for the subsurface grid (TRIANG array)*
        - TPNODI2D - *assign elevation values from elements (triangles) to nodes (of the surface grid)*
  - if SURF=true
    - RAST_INPUT_REAL (13 times) - *read real raster input data for surface solver (drainage direction weights, local slopes, path lengths, roughness coefficients, surface water widths, "b^1" and "y^1" scaling exponents, rivulets per cell)*
    - RAST_INPUT_INT (twice) - *read integer raster input data for surface solver (drainage directions)*
- if FL3D=true
  - GRDSYS - *generate 3D grid and set up system matrices*
    - RIORD - *sort node indices for each element in TRIANG into ascending order*
    - AREA2D - *calculate area assigned to each surface node of the subsurface grid*
      - AREAS - *calculate area of a triangular element*
    - GEN3D - *generate 3D subsurface grid: element connectivities array TETRA(5,NT) for NT tetrahedra (TETRA(5,i) is the material type for element i)*
      - NODELT (3 times) - *node-to-element averaging (in this case for transferring 3D nodal coordinates X,Y,Z to centroidal coordinates XC,YC,ZC)*
    - if IPRT1=-1
      - NODELT (3 times) - *node-to-element averaging (as above, but in this case with 3D coordinates X,Y,Z read in as input)*
    - if IOPT=1
      - RIORD - *sort node indices for each element in TETRA into ascending order (not necessary for Newton case - nonsymmetric system matrices)*
    - VOLBAS - *calculate volumes and basis function coefficients*
      - VOLUMS - *calculate volume of an element*
      - BASIS6 - *calculate local basis function coefficients (divided by 6)*
    - LOCMAS - *set up LMASS, the part of the local mass matrix which is constant for all elements (i.e. without the storage coefficient and without the volume term)*
    - if IOPT=1 *(Picard iteration case)*
      - STRPIC - *set up JA (column indices for system matrix storage) and TOPOL (pointer to first nonzero element of each row stored in system matrices; diagonal entry in this symmetric case), and determine NTERM (# of nonzero elements in system matrices)*

CHKPIC - *check that the pointers and indices for system matrix storage have been properly set up*
TETPIC - *set up TETJA(4,4,NT), the index within JA of each component of the 4 x 4 local system matrices (upper triangle only for this symmetric case)*

else *(IOPT=2 - Newton iteration case)*
STRNEW - *set up JA and TOPOL, and determine NTERM*
CHKNEW - *check for errors in set up of system matrix storage*
TETNEW - *set up TETJA(4,4,NT)*
TOPIA - *set up IA (row indices for system matrix storage)*

if FL3D=false
INIT_SURF - *initialization for surface solver module for the case where only surface routing is being simulated*
EFFONE - *initialization for first time step for surface routing only case*

if FL3D=true
INITAL - *further input, and initialization for ICs, BCs, nudging, parameters, and misc variables*
ICVHE, ICVHWT, ICVDWT - *various options for setting up vertical hydrostatic equilibrium initial conditions (ICs)*
INIT1 - *initialization of various counters, flags, and arrays*
INIT_SR - *initialization for surface solver module for the case where coupled surface--subsurface flow is being simulated*
BCONE (twice) - *initialization for non-atmospheric, non-seepage face (nansf) Dirichlet and Neumann boundary conditions (BCs)*
RDNDBC - *read nodal information for nansf BCs*
READBC - *read BC values for nansf BCs*
ATMONE - *read and initialize atmospheric BCs (atmbc); also first time step check for atmbc switching*
SFVONE - *read and initialize seepage face BCs (sfbc)*
SFINIT - *calculate seepage face exit points based on initial pressure heads*
NUDONE - *read and initialize nudging parameters*
NUDLOCATE - *locate the nudging observation points within the grid*
WEIGHT - *calculate weighted pressure heads ($P^{k+\theta} = \theta P^{k+1} + (1-\theta)P^k$, where "k" is the time level); note: oddity of doing this at "time 0" owes to the subtlety of atmbc switching performed above in ATMONE*
MBINIT - *initialize mass balance and hydrograph parameters for first time level*
TPNODI - *distribute on a nodal basis parameters which are input on an element basis (POROS and ELSTOR); also, set up TP, which gives the # of elements connecting to each node*
CHPARM (also CHTANP and CHTANN if KSLOPE=4) - *calculate moisture curve parameters for the various IVGHU cases*
CHVELO - *calculate SW and CKRW needed for storage and velocity calculations and for output*
STORCAL - *calculate volume of water in the subsurface*
if IPRT≥2
NODELT - *perform averaging to transfer nodal CKRW to elemental CKRWE*
VEL3D - *calculate Darcy velocities at the centroid of each tetrahedral element*
VNOD3D - *perform averaging to transfer elemental Darcy velocities to each node*
RECHARGE - *calculate groundwater recharge at initial conditions*
DETOUT - *detailed output at initial conditions*
CHECK - *check if arrays are dimensioned correctly (with respect to maximum values declared in CATHY.H)*

"**start-time-loop**" - *start of new time step handle (this is a "DO WHILE (TIME.LE.TIMESTOP)" in current version of the code, signalling the "interior loop on time", an artifact needed for EnKF/SIR DA)*

if FL3D=true
BCNXT (twice) - *input (if necessary) and process nansf Dirichlet and Neumann BCs*
RDNDBC - *read nodal information for nansf BCs*
READBC - *read BC values for nansf BCs*
SFVNXT or SFVREC - *input (if necessary) and process seepage face BCs*
ATMNXT - *input (if necessary) and process atmospheric BCs*
SWITCH_OLD or ADRSTN - *check for switching of atmospheric BCs*

```
   NUDNXT - input (if necessary) and process nudging observation data
   WEIGHT - calculate weighted pressure heads
100 - back-stepping handle
if SURF=true and FL3D=true and PONDING=true (coupled surface--subsurface flow case)
   SURF_ROUTE - surface routing module
      NOD_CELL - transfer nodal overland flux values computed by the subsurface flow module to cell-based values needed by the surface
                 routing module
      TRANSFER_F3D_SURF - further transfer of overland flux information from subsurface flow module to surface routing module
      ROUTE - solve the surface routing equations
         SERBATOIO or MC (twice) - compute inflow and outflow hydrographs for reservoir or regular surface cells; SERBATOIO calls CALCOLI,
                                   MC solves the Muskingum-Cunge scheme
      ALTEZZE - perform a water balance at the surface cells to calculate the ponding levels (water heights)
      TRANSFER_SURF_F3D - prepare to transfer ponding information computed by the surface routing module (surface water heights and water
                          levels in reservoirs) to the subsurface flow module
      CELL_NOD - transfer cell-based ponding values computed by the surface routing module to node-based values needed by the subsurface
                 flow module
   PONDUPD - update surface BCs for FLOW3D to reflect ponding situation
else if FL3D=false (surface routing only case)
   ROUTE - solve the surface routing equations
      SERBATOIO or MC (twice) - compute inflow and outflow hydrographs for reservoir or regular surface cells; SERBATOIO calls CALCOLI, MC
                                solves the Muskingum-Cunge scheme
   ALTEZZE - perform a water balance at the surface cells to calculate the ponding levels (water heights)
if FL3D=true
   FLOW3D - subsurface flow module
      200_FLOW3D - start of new iteration handle
      if IOPT=1
         PICARD - Picard iteration to linearize Richards' equation
            PICUNS - calculate unsaturated zone characteristic curves, Picard case
               CHPIC0 - calculate soil moisture characteristics, KSLOPE=0 and Picard case
               NODELT - perform averaging to transfer nodal CKRW to elemental CKRWE
               NODELT - perform averaging to transfer nodal ETAI to elemental ETAE
            ASSPIC - assemble global stiffness and mass matrices from the local contributions of each element, Picard case
            RHSPIC - assemble RHS vector, without contributions of the unsaturated zone gravitational term, the boundary conditions, and
                     the nudging term; Picard case
            CFMATP - assemble global LHS system matrix from the stiffness and mass matrices, Picard case
            RHSGRV - add contribution of the unsaturated zone gravitational term to the RHS vector
            NUDPIC - add nudging contribution to the RHS vector, Picard case
               NUDCPT - evaluate model (computed) results at the nudging observation points
            BCPIC - impose boundary conditions, and save diagonal elements of LHS system matrix corresponding to Dirichlet nodes; Picard
                    case
            SYMSLV - solve the symmetric algebraic system of equations resulting from Picard linearization
               SYMSLV subroutines - SYMSLV and the subroutines it calls are in the linear solver package "solscal-extended"
            SHLPIC - restore diagonal elements of LHS system matrix corresponding to Dirichlet nodes and set solution at Dirichlet nodes to
                     the prescribed values, Picard case
      else
         NEWTON - Newton iteration to linearize Richards' equation
            NEWUNS - calculate unsaturated zone characteristic curves, Newton case
               CHNEW0 - calculate soil moisture characteristics, KSLOPE=0 and Newton case
```

NODELT - *perform averaging to transfer nodal CKRW to elemental CKRWE*
NODELT - *perform averaging to transfer nodal ETAI to elemental ETAE*
ASSNEW - *assemble global stiffness and mass matrices from the local contributions of each element, and also the derivative term components of the Jacobian; Newton case*
RHSNEW - *assemble RHS vector, without contributions of the unsaturated zone gravitational term and the boundary conditions; Newton case*
CFMATN - *assemble global LHS system matrix (the Jacobian) from the stiffness and mass matrices and the derivative term components of the Jacobian, Newton case*
RHSGRV - *add contribution of the unsaturated zone gravitational term to the RHS vector*
BCNEW - *impose boundary conditions, and save diagonal elements of LHS system matrix corresponding to Dirichlet nodes; Newton case*
NSYSLV - *solve the nonsymmetric algebraic system of equations resulting from Newton linearization*
NSYSLV subroutines - *NSYSLV and the subroutines it calls are in the linear solver package "solscal-extended"*
SHLNEW - *restore diagonal elements of LHS system matrix corresponding to Dirichlet nodes and set solution at Dirichlet nodes to the prescribed values, Newton case*

MASBAL - *mass balance calculation*
if IOPT=1
BKPIC - *back-calculate fluxes at all Dirichlet BC nodes, Picard case*
else
BKNEW - *back-calculate fluxes at all Dirichlet BC nodes, Newton case*
CHMASS - *calculate general storage term at previous time level*
CHMASS - *calculate general storage term at current time level*
STORMB - *calculate volume of change in storage between current and previous time levels*
FLUXMB - *calculate total inflow and outflow fluxes at the current time level*

if NLRELX=1
RELAX - *apply nonlinear relaxation with constant relaxation parameter OMEGA (Ω): $P^{m+\Omega} = \Omega P^{m+1} + (1-\Omega)P^{m}$, where "m" is the iteration level and the time level throughout is k+1*

else if NLRELX=2
RELXOM - *calculate iteration-dependent relaxation parameter OMEGA*
RELAX - *apply nonlinear relaxation with just-computed OMEGA value*

CONVER - *check for nonlinear convergence and switching of atmospheric and seepage face BCs*
NORMS - *calculate nonlinear convergence and residual error norms*
SWITCH_OLD or SWITCH - *check for switching of atmospheric BCs; note: whether atmbc switching check is done at every iteration or not depends on setting of parameter TOLSWI*
EXTALL or EXTONE - *calculate new position of the exit point along each seepage face*
EXTCVG - *check for seepage face exit point convergence*

if NORMCV=false and ITAGEN=true *(convergence not achieved, and we have not attained the maximum number of iterations ITUNS; note: there are actually a couple of other conditions that also need to be satisfied in order to proceed to the next iteration, namely the linear solver must not have failed, the nonlinear iterations must not be diverging, and, optionally, the seepage face exit points have not all converged)*
WEIGHT - *calculate weighted pressure heads*
go to 200_FLOW3D

if BCKSTP=true
BKSTEP - *reset variables and arrays for back-stepping*
BCNXT or BCBAK (twice) - *special handling of Dirichlet and Neumann BCs for back-stepping case*
RDNDBC - *read nodal information for nansf BCs*
READBC - *read BC values for nansf BCs*
SFVBAK or SFVREC - *special handling of seepage face BCs for back-stepping case*

```
            ATMNXT or ATMBAK - special handling of atmospheric BCs for back-stepping case
            SWITCH_OLD or ADRSTN - check for switching of atmospheric BCs
            WEIGHT - calculate weighted pressure heads
         go to 100
   CHVELO - calculate SW and CKRW needed for storage and velocity calculations and for output
   STORCAL - calculate volume of water in the subsurface
   if FL3D=true
      if IPRT≥2
         NODELT - perform averaging to transfer nodal CKRW to elemental CKRWE
         VEL3D - calculate Darcy velocities at the centroid of each tetrahedral element
         VNOD3D - perform averaging to transfer elemental Darcy velocities to each node
   if SURF=true
      DETOUTQ - discharge output
   if FL3D=true
      HGRAPH - hydrograph calculation (atmospheric component)
      RECHARGE - calculate groundwater recharge
      SAT_FRAC - calculate fraction of the surface that is Horton or Dunne saturated or ponded
      DETOUT - detailed output at intermediate time (TIMPRT time values)
      if NUDN>0
         NUDCPT - time series output of model results at the nudging observation points
   if FL3D=true
      DTSTAT - update time step statistics
      if NOBACK=true (no more back-stepping is possible)
         go to 300
      if TIME>=TMAX (end of simulation)
         go to 300 (this is a "GO TO 200" in current version of the code, but this is because of artifact of "interior" and "exterior" time
                    loops needed for EnKF/SIR DA)
      TIMUPD - update TIME and time step size for the next time step
      TIMNXT - update or re-initialize counters for next time level
   if FL3D=false
      if TIME≥TMAX (end of simulation)
         go to 300
      TIMUPDSUP - update TIME and time step size for the next time step for surface routing only case
      EFFNXT - update parameters and arrays for next time step for surface routing only case
   go to "start-time-loop" (this is an "END DO" just before "C END of THE INTERIOR TIME LOOP" in current version of the code, because of
                            EnKF/SIR DA artifact)
   300 - end of simulation handle
   if FL3D=true
      DETOUT - detailed output at final time
   CLOSIO - close I/O files
end of CATHY_MAIN
```