```
//Method: Residuals
//Subroutine for loading and calculating residuals to concentration m
//
//Subroutine used by:
//  SS OutputBudget
//  SS ModelConc
//  Kalman Smoothing
//  OutBudgetsMonteC
//  AutoCorrelation
//  SampleResids
//
//Passed Parameters:
//$0 - $CFlag
//  0 - Do not continue
//  1 - Continue
//
//Global Variables needed by subroutine:
//  cFileNum - current file number
//  NoModel
//  NoResids
//  NeedChem
//  i4DChem
//  cNSolutes - # of solutes
//  bStorms - true for OutBudgetsMonteC method
//Global Arrays created here:
//  Model - array with model parameters
//  RJDays - array with dates of resiuals
//  Conc - array with concentrations
//  Resids - array with resiudals
//  ModelValid - array whether model valid for current time period
//Global Variables created here:
//  NStorms
//
//Layout(s) Used:
//  [Solute Info];"ChemSelect"
//  [Chemistry_Login];"Chemistry"
//
//Subroutine(s) Used:
//  ConcModelChecker
//  getratingcurve
//  MyMessage
//  center window
//  chemconversion
```

```
    //  pan rjday
    //  fracday
    //  concmodel
    //
    //Format of Chemistry Flat File (No Header Information):
    //  Sorted by Date and Time, tab delimited
    //  Maximum number of records currently 15000.
    //  1 - Date (MM/DD/YY)
    //  2 - Time (00:00:00)
    //  2.5 - Storm Number (0 if baseflow, consecutively numbered for sto
    //      Column only read in if bStorms = true.
    //  3 - Solute 1 (µeq/l)
    //  4 - Solute 2 (µeq/l) ...
    //
    //Fixed for multiple models with daily physical series now stored in
    //  - 9/18/2013
    //Last Modified: 9/16/14 bta
  $tab:=Char(9)
  $cr:=Char(13)
  C_LONGINT($h;$i;$j)
  C_BOOLEAN(bStorms)
  $CFlag:=True
  QUERY([File_Attributes];[File_Attributes]File_Number=cFileNum)
    //Get area of basin from [File_Attributes] file:
    //Watershed area in Ha converted to m^2
  Area:=[File_Attributes]Area_in_Ha*10000
  If (Area=0)
   MMessage:="No watershed area for "+[File_Attributes]File_Name+" file!
   MyMessage
   $CFlag:=False
  End if
    //Continue if everything is okay...
  If ($CFlag)
     //If models are being used, allow only solutes which contain models
   If (NoModel=0)
    QUERY([ConcModel];[ConcModel]File_Number=cFileNum)
    RELATE ONE SELECTION([ConcModel];[Solute_Info])
      //Search out just solutes which are chemistry:
   Else
    QUERY([Solute_Info];[Solute_Info]Solute_Number>0;*)
    QUERY([Solute_Info]; & [Solute_Info]Units#" ")
   End if
   If (Records in selection([Solute_Info])=0)
```

```
    MMessage:="There are no solutes with models entered in [ConcModel] t
    MyMessage
    $CFlag:=False
  End if
End if
  //Continue if everything is okay...
If ($CFlag)
   //Get user to choose the solutes to do calcuations for:
 center window (540;380;2)
 DIALOG([Solute_Info];"ChemSelect")
 CLOSE WINDOW
 If ((bOK=0) | (cNSolutes=0))
  $CFlag:=False
 End if
End if
  //Continue if everything is okay...
If ($CFlag)
   //Check to see if concentration models are valid for chosen series
 If (NoModel=0)
  ConcModelChecker
  $CFlag:=ModelsValid
 End if
End if
  //Continue if everything is okay...
If ($CFlag)
   //
   //Get Chemistry Data:
   //(Not just using concentration/discharge model)
 If (NeedChem)
    //User wants to use chemistry data in [Chemistry_Login] file:
  If (i4DChem=1)
     //Choose chemical samples in [Chemistry_Login] file
   DEFAULT TABLE([Chemistry_Login])
   ALL RECORDS([Chemistry_Login])
   CREATE SET([Chemistry_Login];"NeedSet")
   FORM SET OUTPUT([Chemistry_Login];"Chemistry")
   ChemHeader:="Select Samples to use to Correct Models in Output Budg
   ChemHeader:=ChemHeader+"then Click the 'Continue' Button."
     //Change Menu Bars:
   SET MENU BAR(7)
   DISPLAY SELECTION([Chemistry_Login];*)
   SET MENU BAR(1)
     //Check to see if user hit cancel or no records selected:
```

```
    NumResid:=Records in selection([Chemistry_Login])
    If ((bCancel=1) | (NumResid=0))
      $CFlag:=False
    Else
       //Calculate the residuals for all streamwater samples used
       //  and put results into an array Conc{ion}{sample #}.
       //  Columns of array are Solute1, Solute2 ...
      ARRAY REAL(RJDays;NumResid)
      ARRAY REAL(Conc;cNSolutes;NumResid)
      ORDER BY([Chemistry_Login];[Chemistry_Login]Collect_RJDay;>)
      SELECTION TO ARRAY([Chemistry_Login]Collect_RJDay;RJDays)
      For ($i;1;cNSolutes)
         //Currently assumes offset of chemistry in [Chemistry_Login] i
        SELECTION TO ARRAY(Field(Table(->[Chemistry_Login]);cSolFldNum{$
        QUERY([Solute_Info];[Solute_Info]Solute_Number=cSolFldNum{$i}-14
          //Loop for conversions into µeq/L:
        For ($j;1;NumResid)
          Conc{$i}{$j}:=chemconversion (Conc{$i}{$j};[Solute_Info]Units;[
        End for
      End for
    End if
      //
      //User wants to upload a flat file with chem data in it:
  Else
    mMessage:="In the following open file dialog, open the sample chemi
    mMessage:=mMessage+"date (m/d/yy), time (h:mm:ss), followed by each
    MyMessage
    $Doc:=Open document("";"TEXT")
      //Dimension for array (Max # of Chem Records!)
    NumResid:=15000
    ARRAY REAL(RJDays;NumResid)
    ARRAY REAL(Conc;cNSolutes;NumResid)
    If (bStorms)
      ARRAY INTEGER(aStormNum;NumResid)
      NStorms:=0
    End if
      //$i is counter for actual # of records read:
    $i:=0
      //Compiler Directives:
    C_STRING(12;$vDate;$vTime;$vConc;$vDayVar)
      //Check to see if document was opened:
    If ($Doc=?00:00:00?)
      $CFlag:=False
```

```
Else
 FNChemistry:=Document
 Repeat
    //Get variables $vDate; $vTime:
  RECEIVE PACKET($Doc;$vDate;$tab)
    //End of File; Make For loop end by changing counter:
  If (OK=1)
     //Sucessfully received data:
     //Counter for actual number of chemical records uploaded:
   $i:=$i+1
   MESSAGE("Reading in sample "+String($i)+".")
   RECEIVE PACKET($Doc;$vTime;$tab)
   RJDays{$i}:=pan rjday (Date($vDate))+fracday (Time($vTime))
     //Receive Storm Number
   If (bStorms)
    RECEIVE PACKET($Doc;$vConc;$tab)
    aStormNum{$i}:=Num($vConc)
      //Determine number of storms
    If (aStormNum{$i}>NStorms)
     NStorms:=aStormNum{$i}
    End if
   End if
   For ($j;1;cNSolutes-1)
    RECEIVE PACKET($Doc;$vConc;$tab)
    Conc{$j}{$i}:=Num($vConc)
   End for
     //Last Solute
   RECEIVE PACKET($Doc;$vConc;$cr)
   Conc{cNSolutes}{$i}:=Num($vConc)
  End if
 Until (OK=0)
   //Close Document:
 CLOSE DOCUMENT($Doc)
   //Free up memory by removing unused section of arrays RJDays Cor
 DELETE FROM ARRAY(RJDays;$i+1;NumResid-$i)
 If (bStorms)
  DELETE FROM ARRAY(aStormNum;$i+1;NumResid-$i)
 End if
 For ($j;1;cNSolutes)
  DELETE FROM ARRAY(Conc{$j};$i+1;NumResid-$i)
 End for
 NumResid:=$i
End if
```

```
    End if
      //If there are no models, assume all negative concentrations are n
      //  half the detection limit, use half detection limit for concent
    If (NoModel=1)
      For ($i;1;NumResid)
        For ($j;1;cNSolutes)
          If ((Conc{$j}{$i}#<>Missing) & (Conc{$j}{$i}<0))
            Conc{$j}{$i}:=Abs(Conc{$j}{$i})
          End if
        End for
      End for
    End if
  End if
End if
  //
  //Continue if everything is okay...
If ($CFlag)
    //Use date of first sample if there are residuals, otherwise use st
  If (NoResids=0)
    $SDate:=RJDays{1}
  Else
    $SDate:=pan rjday (sdate)+fracday (stime)
  End if
    // Get constants for stage-discharge relationships from the [Rating
  If ([File_Attributes]File_Type_Code="SS")
    getratingcurve (cFileNum;$SDate)
  End if
    //
    //Put initial CONCENTRATION MODEL parameters into array Model if
    //  a model is used.
    //Arrays for Models!
    //  Model parameters (make whether or not a model)
  ARRAY REAL(Model;cNSolutes;30)
    //  Model Valid flag
  ARRAY BOOLEAN(ModelValid;cNSolutes)
  If (NoModel=0)
    //Arrays for Models!
    //  Model start and end period
    ARRAY REAL(ConcModSRJDay;cNSolutes)
    ARRAY REAL(ConcModERJDay;cNSolutes)
    //  Flow model types
    ARRAY STRING(10;FlowModel;cNSolutes;2)
    //  Number of daily physical series and daily physical series numb
```

```
ARRAY INTEGER(ConcModDailyNumSer;cNSolutes)
ARRAY INTEGER(ConcModDailySerNum;cNSolutes;6)
  //Loop to get initial information on models:
For ($i;1;cNSolutes)
   //Subrountine to update current concentration model parameters
 concmodelfinder ($SDate;$i)
End for
  //
  //This section commented out because went to storing daily data in
  //  Much easier to lookup, especially with models using different
  //    changing models through time (complicated to figure out what
  //    uploaded for each run)
  //NOTE: Currently assumes (programmed) same daily file across all
  //See what daily files need to read in from user
  //Start and end times in Panola Running JDays
  //$sjday:=pan rjday (sdate)
  //$ejday:=pan rjday (edate)
  //ARRAY BOOLEAN(DailyVarPres;3)
  //ARRAY STRING(36;$DailyVarName;3)
  //NDailyVar:=0
  //mMessage:="In the following open file dialog, open the daily var
  //mMessage:=mMessage+"date (m/d/yy) followed by each daily variabl
  //For ($h;1;3)
  //DailyVarPres{$h}:=False
  //Loop for each solute used
  //For ($i;1;cNSolutes)
  //Search out records for solutes; time period, and daily variable
  //QUERY([ConcModel];[ConcModel]File Number=cFileNum;*)
  //QUERY([ConcModel]; & [ConcModel]Solute Number=cSolFldNum{$i}-14;
  //QUERY([ConcModel]; & [ConcModel]Model SRJDay<$ejday;*)
  //QUERY([ConcModel]; & [ConcModel]Model ERJDay>$sjday;*)
  //QUERY([ConcModel]; & Field(Table(->[ConcModel]);$h+21)#0)
  //Case of
  //: ($h=1)
  //QUERY([ConcModel]; & [ConcModel]DayVarMult1#0)
  //: ($h=2)
  //QUERY([ConcModel]; & [ConcModel]DayVarMult2#0)
  //Else
  //QUERY([ConcModel]; & [ConcModel]DayVarMult3#0)
  //End case
  //$NModels:=Records in selection([ConcModel])
  //If there were any records found, then upload daily variable for
  //If ($NModels>0)
```

```
//DailyVarPres{$h}:=True
//NDailyVar:=NDailyVar+1
//Put in daily variable name from first record
//2 fields deleted 35 & 36, hence label fields are 38-40
//$DailyVarName{$h}:=Field(Table(->[ConcModel]);$h+37)->
//mMessage:=mMessage+"Daily variable "+String(NDailyVar)+" is "+$D
//Force end of loop so don't add any more
//$i:=cNSolutes
//End if
//End for
//End for
//If need to read in daily variables:
//If (NDailyVar>0)
//Arrays to read in
//$DayVarDim:=20000
//ARRAY REAL(DailyVarRJDay;$DayVarDim)
//ARRAY REAL(DailyVars;NDailyVar;$DayVarDim)
//MyMessage
//$Doc:=Open document("";"TEXT")
//Check to see if document was opened:
//If ($Doc=?00:00:00?)
//$CFlag:=False
//Else
//  Panola Running JDays
//$i:=0
//Repeat
//Get variables $vDate;
//RECEIVE PACKET($Doc;$vDate;$tab)
//End of File; Make For loop end by changing counter:
//If (OK=1)
//$i:=$i+1
//Sucessfully received data:
//MESSAGE("Reading in sample "+String($i)+".")
//DailyVarRJDay{$i}:=pan rjday (Date($vDate))
//Loop for each daily file variable
//NOTE: Currently assumes (programmed) same daily file across all
//$DayVarCount:=0
//For ($h;1;3)
//If the day variable is present
//If (DailyVarPres{$h})
//$DayVarCount:=$DayVarCount+1
//If ($DayVarCount=NDailyVar)
//RECEIVE PACKET($Doc;$vDayVar;$cr)
```

```
    //Else
    //RECEIVE PACKET($Doc;$vDayVar;$tab)
    //End if
    //DailyVars{$DayVarCount}{$i}:=Num($vDayVar)
    //End if
    //End for
    //End if
    //Until (OK=0)
    //Close Document:
    //CLOSE DOCUMENT($Doc)
    //Free up memory by removing unused section of arrays
    //DELETE FROM ARRAY(DailyVarRJDay;$i+1;$DayVarDim-$i)
    //For ($h;1;NDailyVar)
    //DELETE FROM ARRAY(DailyVars{$h};$i+1;$DayVarDim-$i)
    //End for
    //End if
    //End if
End if
  //
  //Calculate residuals if using residuals:
If ((NoResids=0) & ($CFlag))
 ARRAY REAL(Resids;cNSolutes;NumResid)
   //Calculate the residuals for each streamwater sample:
 For ($i;1;NumResid)
  MESSAGE("Calculating residual(s) for sample "+String($i)+" of "+Str
    //Loop for each solute
  For ($j;1;cNSolutes)
     //Calculate predicted concentration at Resids{1}{$i} for ion $j
     //The # 0 in $3 indicates indicates that residuals are being ca
     //  In which case, predicted concentration will be calulated by
   If (Conc{$j}{$i}#<>Missing)
      //Calculate predicted only if NoModel = 0, otherwise residual
    If (NoModel=0)
     $Predicted:=concmodel (RJDays{$i};$j;0;0)
       //Valid model should be selected for solute and time, check t
       //  negative concentrations, otherwise adjust concentration.
       //If don't allow Negative Concentrations (e.g. for ANC), Nega
       //  negative half the detection limit, use half detection lim
     If ((Conc{$j}{$i}<0) & (Model{$j}{9}=0))
      Conc{$j}{$i}:=Abs(Conc{$j}{$i})
     End if
    Else
     $Predicted:=0
```

```
        End if
          //Residuals defined as (Predicted - Observed)
        If ($Predicted#<>Missing)
         Resids{$j}{$i}:=$Predicted-Conc{$j}{$i}
        Else
         Resids{$j}{$i}:=<>Missing
        End if
       Else
        Resids{$j}{$i}:=<>Missing
       End if
      End for
     End for
       //Reset start & end rjdays and make models invalid to force new mo
       //  Can do this here because both methods that use concmodel use t
       //  Concmodel method always assumes that day is increasing - hence
     For ($i;1;cNSolutes)
      If (NoModel=0)
       ModelValid{$i}:=False
       ConcModSRJDay{$i}:=-1000000
       ConcModERJDay{$i}:=-1000000
      Else
       ModelValid{$i}:=True
      End if
     End for
    End if
  End if
  $0:=$CFlag
```