```
    //Method: OutBudgetsMonteC
    //This program randomizes the number of storms sampled for
    //  determining samping strategy using a Monte Carlo approach.
    //This program determines chemical outputs using the combination outp
    //  which uses streamwater samples and stream stage (to determine dis
    //Fluxes are generated daily and summed for the period.
    //Concentrations are calculated using concentraion/discharge/season m
    //  generated for each solute.  These models are stored in the file
    //
    //Subroutine(s) Used:
    //  Residuals
    //  pan rjday
    //  fracday
    //  pan caldate
    //  qtrap
    //  linterp
    //  resids search
    //  pinterpolate
    //  stgtodis
    //  concmodel
    //  avgdischarge
    //  getratingcurve
    //  chemconversion
    //  center window
    //  MyMessage
    //  OutBudgetModelsOutput
    //
    //Layout(s) Used:
    //  [File_Attributes];"frmOutBudMonteC"
    //
    //Modified for completion of method estimate  - 10 February 1999 bta
    //Modified for monthly output  - 2 January, 2004 bta
    //Written:   8 February 1999 bta
    //Last Modified: 7 January, 2004 bta
    //
  C_INTEGER($h;$i;$j;$m;NumResid)
  C_BOOLEAN($CFlag)
  $SRunDate:=Current date
  $SRunTime:=Current time
  $tab:=Char(9)
  $cr:=Char(13)
    //$CFlag is a flag to indicate whether or not to continue the program
    //  False - Do Not Continue; True - Continue.
```

```
$CFlag:=True
  //EPS is the desired fractional accuracy.
  //  Convergence is when change in estimate is less than current estim
  //  fractional accuracy (ESP)
EPS:=0.002
  //JMIN and JMAX are minimum and maximum number of iterations:
  //First iteration are the endpoints for the day.
  //Second iteration divides the day in half.
  //Third iteration divides the day in quarters, ect...
JMIN:=6
JMAX:=13
  //Always use residuals
NoResids:=0
  //Get Chemistry
NeedChem:=True
  //Chemistry data always from flat file, not [Chemistry_Login] file
  //  because need storm numbers:
i4DChem:=0
  //Randomization using storm numbers
bStorms:=True
  //Get dates and filename for results file from user
center window (370;430;2)
DIALOG([File_Attributes];"frmOutBudMonteC")
CLOSE WINDOW
If (bOK=1)
    //Calculation methods:
  If (CombModel=1)
    $sMethod:=1
    NoModel:=0
  Else
    $sMethod:=2
    NoModel:=1
  End if
  If (PeriodModel=1)
    $eMethod:=2
  Else
    $eMethod:=1
  End if
Else
  $CFlag:=False
End if
  //
  //Continue if everything is okay...
```

```
If ($CFlag)
    //Start and end times in Panola Running JDays
  $sjday:=pan rjday (sdate)
  $ejday:=pan rjday (edate)
    //Number of days to run:
  $days:=$ejday-$sjday
    //Calculate Residuals and such
    //  Pass the start date
  $CFlag:=Residuals
End if
  //
  //Continue if everything is okay...
If ($CFlag)
    //Store Residual Arrays to choose for randomization
  ARRAY REAL($iRJDays;NumResid)
  COPY ARRAY(RJDays;$iRJDays)
  ARRAY REAL($iResids;cNSolutes;NumResid)
  COPY ARRAY(Resids;$iResids)
  NResids:=NumResid
    //Number of storms to use:
  $NStormsUse:=Round(NActStorms*PctStorms/100;0)
  If ($NStormsUse>NStorms)
    $NStormsUse:=NStorms
    NRuns:=1
    MMessage:="Percentage of storms to randomly include is greater than
    MMessage:=MMessage+" uploaded, will include all storms uploaded and
    MyMessage
  End if
    //Create output document and export header info:
  $tab:=Char(9)
  $cr:=Char(13)
  OBMDoc:=Create document(OutFileName)
    //Export titles:
  $oString:="Output Budgets Monte Carlo method started on "+String($SRu
  $oString:=$oString+"For period "+String(pan caldate ($sjday))+" - "+S
  $oString:=$oString+"Output file name: "+OutFileName+"."+$cr
  $oString:=$oString+"Chemical samples imported from file "+FNChemistry
  $oString:=$oString+"Fractional Accuracy for convergence is "+String(E
  $oString:=$oString+"Minimum number of intervals for convergence is "+
  $oString:=$oString+"Maximum number of intervals for convergence is "+
  SEND PACKET(OBMDoc;$oString)
    //Use subroutine for outputing model parameters into file
  OutBudgetModelsOutput ($sjday;$ejday)
```

```
$oString:="Number of output budget runs is "+String(NRuns)+"."+$cr
$oString:=$oString+"Number of storms to randomly include in each run
$oString:=$oString+"Actual number of storms in the period (including
   //Headers:
$oString:=$oString+"Run"+$tab+"Method"+$tab+"Solute"+$tab+"Month"+$ta
$oString:=$oString+"Total Mass Flux (µeq/m^2)"+$tab
$oString:=$oString+"Sum of Concentration/Discharge/Season Model Mass
$oString:=$oString+"Sum of Residual Mass Flux from Sample Corrections
$oString:=$oString+"Sum of Absolute Value of Residual Mass Flux (µeq/
$oString:=$oString+"Average Daily Flow Calculated from Iterations (l/
$oString:=$oString+"Maximum Number of Iterations"+$cr
SEND PACKET(OBMDoc;$oString)
   //Arrays for randomizing storms:
ARRAY INTEGER($StormOrder;NStorms)
ARRAY INTEGER($StormRandom;NStorms)
   //Time start of Iterations
$SIterDate:=Current date
$SIterTime:=Current time
   //Loop for each randomization:
For ($h;1;NRuns)
  //Screen Message:
  OBMess1:="Run "+String($h)+" of "+String(NRuns)+"."+$cr
     //Estimated finish time
  If ($h>1)
    OBMess1:=OBMess1+"Completion of method estimated at "+String(pan ca
  End if
     //Randomize Storms
  For ($i;1;NStorms)
    $StormOrder{$i}:=$i
    $StormRandom{$i}:=Random
  End for
     //Order by random number so now $StormOrder is random
  SORT ARRAY($StormRandom;$StormOrder;>)
     //
     //Loop for Calculation Method:
  For ($m;$sMethod;$eMethod)
       //Set Calculation Method
    If ($m=1)
      NoModel:=0
      OBMess:=OBMess1+"Method is combined."+$cr
    Else
      NoModel:=1
      OBMess:=OBMess1+"Method is period-weighted (linear interpolation)
```

```
End if
   //Arrays to hold values for current randomization:
ARRAY REAL(RJDays;NResids)
ARRAY REAL(Resids;cNSolutes;NResids)
   //Put samples for current randomization into arrays:
NumResid:=0
For ($i;1;NResids)
  Case of
         //Not a storm
    : (aStormNum{$i}=0)
      NumResid:=NumResid+1
      RJDays{NumResid}:=$iRJDays{$i}
      For ($j;1;cNSolutes)
        If ($m=1)
          Resids{$j}{NumResid}:=$iResids{$j}{$i}
        Else
          If (Conc{$j}{$i}#<>Missing)
            Resids{$j}{NumResid}:=-Conc{$j}{$i}
          Else
            Resids{$j}{NumResid}:=<>Missing
          End if
        End if
      End for
         //Storm, only use if order number is less than or equal to $N
    : ($StormOrder{aStormNum{$i}}<=$NStormsUse)
      NumResid:=NumResid+1
      RJDays{NumResid}:=$iRJDays{$i}
      For ($j;1;cNSolutes)
        If ($m=1)
          Resids{$j}{NumResid}:=$iResids{$j}{$i}
        Else
          If (Conc{$j}{$i}#<>Missing)
            Resids{$j}{NumResid}:=-Conc{$j}{$i}
          Else
            Resids{$j}{NumResid}:=<>Missing
          End if
        End if
      End for
  End case
End for
   //Free up memory by removing unused section of arrays RJDays, Res
If (NResids>NumResid)
  DELETE FROM ARRAY(RJDays;NumResid+1;NResids-NumResid)
  For ($j;1;cNSolutes)
    DELETE FROM ARRAY(Resids{$j};NumResid+1;NResids-NumResid)
  End for
End if
   //
   //Now ready to do calculations:
   //Loop for each solute
For ($j;1;cNSolutes)
     //Initialize variables for output
  $NDays:=0
  $TotalFlux:=0
  $ModFlux:=0
  $ResidFlux:=0
  $AResidFlux:=0
  $DailyFlow:=0
  $MaxIt:=0
  $NewMonth:=True
     //Loop for DAILY OUTPUT BUDGETS
     //  Panola Running JDays
  For ($i;1;$days)
       //Make record for daily chemical fluxes:
    CurrentDay:=pan caldate ($sjday+$i-1)
       //Check to see if new month
    If ($NewMonth & (OutMonth=1))
         //Initialize variables for monthly output
      $MNDays:=0
      $MTotalFlux:=0
      $MModFlux:=0
      $MResidFlux:=0
      $MAResidFlux:=0
      $MDailyFlow:=0
      $MMaxIt:=0
      $NewMonth:=False
      $Month:=Month of(CurrentDay)
      $MSDay:=CurrentDay
    End if
       //Subroutine qtrap is the driver for the integration routine t
       //$S is in µeq
    $S:=qtrap (($sjday+$i-1);($sjday+$i);$j)
       //Converged
    If ($S#<>Missing)
         //Convert Fluxes from µeq/watershed/day to µeq/m2/day
      $NDays:=$NDays+1
```

```
            $TotalFlux:=$TotalFlux+($S/Area)
            $ModFlux:=$ModFlux+(bsModel/Area)
            $ResidFlux:=$ResidFlux+(bsResid/Area)
            $AResidFlux:=$AResidFlux+(bsAResid/Area)
            $DailyFlow:=$DailyFlow+bsQinst
            If (NIterations>$MaxIt)
              $MaxIt:=NIterations
            End if
               //Monthly totals
            If (OutMonth=1)
              $MNDays:=$MNDays+1
              $MTotalFlux:=$MTotalFlux+($S/Area)
              $MModFlux:=$MModFlux+(bsModel/Area)
              $MResidFlux:=$MResidFlux+(bsResid/Area)
              $MAResidFlux:=$MAResidFlux+(bsAResid/Area)
              $MDailyFlow:=$MDailyFlow+bsQinst
              If (NIterations>$MMaxIt)
                $MMaxIt:=NIterations
              End if
            End if
          End if
             //If end of month, export monthly sums:
          If ((OutMonth=1) & ($Month#Month of(CurrentDay+1)))
            $NewMonth:=True
               //Export data:
            $oString:=String($h)+$tab
            If ($m=1)
              $oString:=$oString+"Combination"
            Else
              $oString:=$oString+"Period-Weighted"
            End if
            $oString:=$oString+$tab+cSolutes{$j}+$tab+String($MSDay)+$tab+S
            $oString:=$oString+$tab+String($MModFlux)+$tab+String($MResidFl
            $oString:=$oString+$tab+String($MDailyFlow/$MNDays)+$tab+String
            SEND PACKET(OBMDoc;$oString)
          End if
        End for
           //Export data:
        $oString:=String($h)+$tab
        If ($m=1)
          $oString:=$oString+"Combination"
        Else
          $oString:=$oString+"Period-Weighted"
        End if
        $oString:=$oString+$tab+cSolutes{$j}+$tab+"Period"+$tab+String($N
        $oString:=$oString+$tab+String($ModFlux)+$tab+String($ResidFlux)+
        $oString:=$oString+$tab+String($DailyFlow/$NDays)+$tab+String($Ma
        SEND PACKET(OBMDoc;$oString)
      End for
    End for
      //Time Estimates of method completion:
    $EstEDate:=days ($SIterDate;$SIterTime;Current date;Current time)/$h
  End for
    //Final Output Time Stats:
  $ERunDate:=Current date
  $ERunTime:=Current time
  $RunTime:=Round(days ($SRunDate;$SRunTime;$ERunDate;$ERunTime)*24;2)
  $NActRuns:=NRuns*cNSolutes*($eMethod-$sMethod+1)
  $TimePerRun:=Round($RunTime/$NActRuns;4)
  $oString:="Method finished at "+String($ERunDate;1)+" at "+String($ER
  $oString:=$oString+"Method calculated "+String($NActRuns)+" runs in "
  $oString:=$oString+"Average time per run is "+String($TimePerRun)+" h
  SEND PACKET(OBMDoc;$oString)
  CLOSE DOCUMENT(OBMDoc)
End if
```