```
Attribute VB_Name = "QWASI"

Sub QWASIcalcs()

' Canadian Environmental Modelling Centre
' Trent University
' Peterborough, Ontario, K9J 7B8, Canada
' http://www.trentu.ca/cemc/

' Original Program coded by Eva Webster and Heather Haertel.
' MSExcel version modifies original code - modifications by Liisa Reid, May 2007

' Based on the following publications.

' Mackay, D. 2001. "Multimedia Environmental Models: The Fugacity Approach -
' Second edition", Lewis  Publishers, Boca Raton. pp. 201-213.

' Mackay, D., Joy, M., Paterson, S. 1983. A Quantitative Water, Air, Sediment
' Interaction (QWASI) Fugacity Model for Describing The Fate of Chemicals in
' Lakes. Chemosphere. 12: 981-997.

' Mackay, D., Paterson, S.,  Joy, M. 1983. A Quantitative Water, Air,
' Sediment Interaction (QWASI) Fugacity Model for Describing the Fate of
' Chemicals in Rivers. Chemosphere. 12: 1193-1208.

' With a Great Lakes parameterization as described in:
'
' Webster, E., Lian, L., Mackay, D. 2005. Application of the Quantitative Water Air
' Sediment Interaction (QWASI) Model to the Great Lakes. Report to the Lakewide
' Management Plan (LaMP) Committee CEMC Report 200501. Trent University,
' Peterborough, Ontario.

' Other related publications:

' Mackay, D., Diamond, M. 1989. Application of the QWASI (Quantitative Water
' Air Sediment Interaction) Fugacity Model to the Dynamics of Organic and
' Inorganic Chemicals in Lakes. Chemosphere. 18: 1343-1365.

' Mackay, D. 1989. Modelling the Long Term Behaviour of an Organic Contaminant in
' a Large Lake: Application to PCBs in Lake Ontario. J. Great Lakes Res. 15:
' 283-297.

' Diamond, M., Mackay, D., Poulton, D. and Stride, F. 1994. Development of a Mass
' Balance Model of the Fate of 17 Chemicals in the Bay of Quinte. J. Great
' Lakes Res. 20: 643-666.

' Diamond, M.L., Mackay, D., Poulton, D.J., and Stride, F.A. 1996. Assessing
' Chemical Behavior and Developing Remedial Actions Using a Mass Balance Model
' of Chemical Fate in the Bay of Quinte. Wat. Res. 30: 405-421.

' Lun, R., Lee, K., De Marco, L., Nalewajko, C. and Mackay, D. 1998. A Model of
' the Fate of Polycyclic Aromatic Hydrocarbons in the Saguenay Fjord. Environ.
' Toxicol. and Chem. 17: 333-341.
```

```
' Woodfine D.G., Seth R., Mackay D., Havas M. 2000. Simulating the Response of
' Metal Contaminated Lakes to Reductions in Atmospheric Loading Using a
' Modified QWASI Model. Chemosphere. 41: 1377-1388.

' Mackay, D., Hickie, B. 2000. Mass Balance Model of Sources, Transport, and
' Fate of PAHs in Lac Saint Louis, Quebec. Chemosphere.41: 681-692.

' Milford, L. 2002. A Multi-Segment Modelling Approach to Describe the Fate
' of PCBs in Two River Systems in Southern Ontario. Masters thesis, Watershed
' Ecosystems Graduate Program, Trent University, Peterborough, Ontario.

' Warren, C.S., Mackay, D., Bahadur, N.P., Boocock, G.B. 2002. A Suite of
' Multi-Segment Fugacity Models Describing the Fate of Organic Contaminants
' in Aquatic Systems: Application to the Rihand Reservoir, India. Water
' Research 36: 4341-4355.

'
************************************************************************************************

' NOTES:

' Numbering schemes for arrays Environmental Media defined as follows:
' for bulk phases
' 1 = water; 2 = sediment; 3 = inflow water; 4 = air

' for subcompartments
' 1 = water
' 2 = solids (i.e., aerosols, water particles, Sediment solids)
' 3 is not used
' 4 = vapour (i.e., air without aerosols)
' 5 = resuspended sediment

' There are 16 transfer processes
' 1 for Sediment burial (B)
' 2 for Sediment transformation (S)
' 3 for Sediment resuspension (R)
' 4 for water-to-Sediment diffusion (WD)
' 5 for Sediment-to-water diffusion (SD)
' 6 for Sediment deposition (D)
' 7 for water transformation (W)
' 8 for volatilization (V)
' 9 for absorption (A)
' 10 for water outflow (J)
' 11 for water particle outflow (Y)
' 12 for rain dissolution (M)
' 13 for wet particle deposition (C)
' 14 for dry particle deposition (Q)
' 15 for water inflow (I)
' 16 for water particle inflow (X)
' (Letters refer to Mackay, 2001)

'
************************************************************************************************
' The Calculations
```

```
'
***************************************************************************************************
'############################################################################
'## DURING DYMAMIC CALCULATIONS ONLY DO THIS SECTION ONCE TO INITIALIZE    ##
'## VALUES, OTHERWISE JUMP OVER IT                                          ##
'############################################################################
If Dynamic = True And (tim > (timestep - (0.001 * timestep))) Then
    GoTo cont
End If


nom$(1) = "Water"
nom$(2) = "Sediment"
nom$(3) = "Inflow Water"
nom$(4) = "Air"

If InitialTimestepCalc = True Then
    model$ = ActiveCell.Parent.Parent.name
End If

'Set Environmental Parameters
Workbooks(model$).Activate
Sheets("Environment").Select

EnvironmentName = [b2] '1 Name

'Dimensions
Area(1) = [B4] '2 Water Surface Area (m2)
Area(2) = Area(1) ' Sediment surface area (m2)
Volume(1) = [B5] '3 Water Volume(m3)
Depth(1) = Volume(1) / Area(1)
Depth(2) = [b6] '4 Sediment Active Layer Depth (m)
Volume(2) = Area(2) * Depth(2)

' Concentration of Solids
ParticConc(4) = [b8] '5 Aerosol Particles(ug / m3)
ParticConc(3) = [b9] '6 inflow Particles(mg / L)
ParticConc(1) = [b10] '7 Suspended Sediment(mg / L)
baseParticConc(1) = ParticConc(1) ' store this for later use as we change ParticConc(1) during
RO events
VolFract(2) = [b11] '8 Sediment Particles(m3 / m3)

'Density of Solids (kg/m3)
Density_kg(1) = [b13] '9 in Water
Density_kg(2) = [b14] '10 in Sediment
Density_kg(4) = [b15] '11 in Aerosols
Density_kg(3) = Density_kg(1)
Density_kg(5) = Density_kg(2)

'Organic Carbon Fraction of Solids
OrgCarbFract(1) = [b17] '12 in Water
OrgCarbFract(2) = [b18] '13 in Sediment
OrgCarbFract(3) = [b19] '14 in Inflow
OrgCarbFract(5) = [b20] '15 in Resuspended Sediment
```

```
'Flows
GRate(15) = Val([B22].Value) '16 River water inflow (m3/h)
GRate(10) = [B23] '17 Water outflow(m3 / h)
GRate_n(6) = [B24] '18 Deposition rate of solids to sediment (g/m2)
GRate_n(1) = [b25] '19 Burial Rate of Solids (g/m2)
GRate_n(3) = [b26] '20 Resuspension Rate of Solids (g/m2)

'sediment loss rate - it's specified as a 90% clearance time
sedRate = Log(10) / ([b35] * 24)

'Mass Transfer Coefficients
GRate_n(14) = [b28] '21 Aerosol Dry Deposition (m/h)
ScavengRatio = [b29] '22 Scavenging Ratio (vol air/vol rain)
GRate_n(12) = [b30] '23 Rain Rate(m / y)
MTCairSide = [b31] '24 Volatilization (air side) (m/h)
MTCwaterSide = [b32] '25 Volatilization (water side) (m/h)
GRate_n(4) = [b33] '26 Sediment-Water diffusion (m/h)



'Environmental Properties
'Calculate density in g/m3
Density(1) = Density_kg(1) * kg2g ' suspended particles
Density(2) = Density_kg(2) * kg2g ' sediment solids
Density(3) = Density_kg(3) * kg2g ' same value as (1)
Density(4) = Density_kg(4) * kg2g ' aerosols
Density(5) = Density_kg(5) * kg2g ' resuspended Particles

'Calculate solid volume fractions
VolFract(1) = ParticConc(1) / Density(1)             ' ParticConc(1) is in mg/L
VolFract(3) = ParticConc(3) / Density(3)             ' ParticConc(3) is in mg/L
VolFract(4) = ParticConc(4) / (Density(4) * g2ug)    ' ParticConc(4) is in ug/m3; volume
fraction particles in atmosphere

Area(2) = Area(1)                                    ' sediment surface area (m2)
Depth(1) = Volume(1) / Area(1)                       ' water depth (metres)
Volume(2) = Area(2) * Depth(2)                       ' sediment active volume (cubic metres)

'volume of suspended Particles and water solution
VolSub(1, 2) = VolFract(1) * Volume(1)
VolSub(1, 1) = Volume(1) - VolSub(1, 2)

'volume of sediment solids and pore water
VolSub(2, 2) = Volume(2) * VolFract(2)
VolSub(2, 1) = Volume(2) - VolSub(2, 2)

'Chemical Inputs
Sheets("Chemical").Select
ChemicalType = [B3]

'#################################
'##      CHEMICAL TYPE 1        ##
'#################################
```

```
If ChemicalType = 1 Then
    ChemicalName = [b2] 'Chemical Name
    Chemtype = [B3] 'Chemical Type
    ChemPropTemp_C = [B4] 'Property Temperature C
    MolMass = [B5] 'Molecular Mass(g / mol)
    MeltPoint_C = [b6] 'Melting Point(C)
    WatSol = [b7] 'Solubility (g / m3)
    VapPress = [b8] 'Vapour Pressure(Pa)

    'partitioning
    LKow = [b10] 'logKOW
    LKaw = [b11] 'logKAW
    LKoa = [b12] 'logKOA

    'Degradation half - lives(h)
    HalfLife(1) = [b14] 'Water
    HalfLife(2) = [b15] 'Sediment

End If
'#################################
'##     CHEMICAL TYPE 1 END    ##
'#################################

'#################################
'##     CHEMICAL TYPE 2        ##
'#################################
If ChemicalType = 2 Then
    ChemicalName = [b2] 'Chemical Name
    Chemtype = [B3] 'Chemical Type
    ChemPropTemp_C = [B4] 'Property Temperature C
    MolMass = [B5] 'Molecular Mass(g / mol)
    MeltPoint_C = [b6] 'Melting Point(C)

    'Degradation half - lives(h)
    HalfLife(1) = [b14] 'Water
    HalfLife(2) = [b15] 'Sediment

    'Type II partitioning
    Kaw = [b17] 'Air-Water KAW (dimensionless)
    KQW = [b18] 'AerosolWater KQW(dimensionless)
    PartitionCoef_Lkg(2, 1) = [b19] 'Sediment -Water(L / kg)
    PartitionCoef_Lkg(1, 1) = [b20] 'Suspended Sediment - Water(L / kg)
    PartitionCoef_Lkg(5, 1) = [b21] 'Resuspended Sediment - Water(L / kg)
End If
'#################################
'##     CHEMICAL TYPE 2 END    ##
'#################################

'unit conversions on chemical input values

' data temperature in degrees Kelvin
ChemPropTemp_K = ChemPropTemp_C + CelsiusToKelvin

'chemical type 1 or 2 calculations
```

```
'################################
'##     CHEMICAL TYPE 1        ##
'################################
If ChemicalType = 1 Then    ' type 1 specific calculations begin here

    ' melting point in degrees Kelvin
    MeltPoint_K = MeltPoint_C + CelsiusToKelvin

    'fugacity ratio
    If MeltPoint_C > ChemPropTemp_C Then
        FugRatio = Exp(6.79 * (1 - MeltPoint_K / ChemPropTemp_K))
    Else
        FugRatio = 1!
    End If

    VapPressLiquid = VapPress / FugRatio            ' subcooled liquid vapor pressure
    WatSol_mol = WatSol / MolMass                  ' solubility (mol/m3)
    HenryLaw = VapPress / WatSol_mol                ' Henry's Law constant (Pa m3/mol)
    Kow = 10 ^ LKow                                 ' octanol-water partition coefficient

    Kaw = HenryLaw / (GasConst * ChemPropTemp_K)    ' air-water

    ' calculate partition coefficients in units of L/kg
    If [AGRO!P7] Then
        Koc = [Chemical!B22]     'use Koc direct input instead of Kow equation (QWASI only,
        Foodweb still uses LKow)
    Else
        Koc = 0.41 * Kow                                ' organic carbon-water (the constant
        0.41 has units of L/kg)
        [Chemical!B22] = Koc     'output calculated Koc so user knows what it was
    End If
    PartitionCoef_Lkg(1, 1) = Koc * OrgCarbFract(1) ' water particles - water
    PartitionCoef_Lkg(2, 1) = Koc * OrgCarbFract(2) ' sediment solids - water
    PartitionCoef_Lkg(5, 1) = Koc * OrgCarbFract(5) ' resuspended Particles solids - water

    'Set Emissions
    If EmisDyn = False Then
        Sheets("Emissions").Select
        Emis_kg(1) = [b9]

        ConcBulk_n(3) = [b12] 'ng/L, ug/m3
        ConcBulk_n(4) = [b11] 'ug/m3
    End If

    'Set emission and runoff time periods
    Sheets("AGRO").Select
    EmissTime_h = [f23] 'hr
    RunoffTime_h = [f24] 'hr

cont:
    '############################################################################
    '## cont:                                                                 ##
    '##                                                                       ##
    '## JUMP INTO HERE DURING DYNAMIC ITERATIONS                              ##
```

```
    '## (GET HERE NORMALLY IN SS AND DURING INITIALIZATION)                ##
    '##                                                                    ##
    '##                                                                    ##
    '#######################################################################

    'if running dynamic mode for SS parameters do not redefine params
    If EmisDyn = False Then GoTo ssparams
    '#######################################################################
    '##                                                                    ##
    '## AND THEN FIND WAYS OF JUMPING OUT AGAIN ... IF RUNNING IN DYNAMIC MODE ##
    '##                                                                    ##
    '#######################################################################

    'Calculate fluctuating D- and Z-values if running Dynamic model
    'If Dynamic = True And tim = timestep Then Call QWASI.Dyn_Params
    If Dynamic = True Then
        If timLoop = 1 Then
            Call QWASI.Dyn_Params ' sets up parameters on first loop
        End If
        If sedResuspPerCent > 0 Then
            'now some parameters change per timestep with the settling sediment
            Call QWASI.Dyn_Sed_Params
            Call QWASI.Dyn_Sed_Calcs
            Call QWASI.Dyn_Sed_ZD
        End If
        'Calculate dynamic parameters every day now that runoff may be applied
        'over sub-daily period
        Call QWASI.Dyn_Params
        GoTo SkipZDcalcs 'and we still jump over the steady state calcs below
    End If

ssparams:
    '#######################################################################
    '## ssparams:                                                          ##
    '##                                                                    ##
    '## DEFINE Z VALUES FOR STEADY STATE ONLY                              ##
    '##  - SKIPPED IF EMISSIONS ARE DYNAMIC                                ##
    '##                                                                    ##
    '#######################################################################


    'Calculate Z values for subcompartments
    ZvalueSub(4, 4) = 1 / GasConst / ChemPropTemp_K ' Z for air
    ZvalueSub(1, 1) = 1 / HenryLaw                  ' Z for water
    ZvalueSub(1, 2) = Koc * OrgCarbFract(1) * Density(1) * ZvalueSub(1, 1) / g2ug 'Z for water
    particles
    ZvalueSub(3, 2) = Koc * OrgCarbFract(3) * Density(1) * ZvalueSub(1, 1) / g2ug 'Z for inflow
    water particles
    ZvalueSub(2, 2) = Koc * OrgCarbFract(2) * Density(2) * ZvalueSub(1, 1) / g2ug 'Z for
    sediment particles
    ZvalueSub(1, 5) = Koc * OrgCarbFract(5) * Density(5) * ZvalueSub(1, 1) / g2ug 'Z for resusp
    sediment particles
    ZvalueSub(4, 2) = 6000000! / VapPressLiquid * ZvalueSub(4, 4)  'Z for aerosol particles
```

```
    'dimensionless partition coefficients
    PartitionCoef(1, 1) = ZvalueSub(1, 2) / ZvalueSub(1, 1) ' particles - water
    PartitionCoef(2, 1) = ZvalueSub(2, 2) / ZvalueSub(1, 1) ' sed solids - water
    PartitionCoef(5, 1) = ZvalueSub(1, 5) / ZvalueSub(1, 1) ' resus sed - water
    PartitionCoef(4, 2) = ZvalueSub(4, 2) / ZvalueSub(4, 4) ' aerosol - air

    'overall air - water mass transfer coefficient (m/h)
    GRate_n(8) = 1 / (1 / MTCwaterSide + 1 / (MTCairSide * Kaw))
End If
'#################################
'##     CHEMICAL TYPE 1 END    ##
'#################################

If ChemicalType = 2 Then 'Chemical type is type 2 then calculate type 2 specific calculations
end of type 1 specific

    'calculate the dimensionless partition coefficients
    PartitionCoef(1, 1) = PartitionCoef_Lkg(1, 1) * Density_kg(1) / 1000   ' water particles -
    water
    PartitionCoef(2, 1) = PartitionCoef_Lkg(2, 1) * Density_kg(2) / 1000   ' sediment solids -
    water
    PartitionCoef(5, 1) = PartitionCoef_Lkg(5, 1) * Density_kg(1) / 1000   ' resuspended
    Particles solids - water

    'dimensional to dimensionless
    ZvalueSub(1, 1) = 1 'water
    ZvalueSub(4, 4) = ZvalueSub(1, 1) * Kaw                  ' air
    ZvalueSub(4, 2) = ZvalueSub(1, 1) * KQW                  ' aerosol water
    ZvalueSub(1, 2) = ZvalueSub(1, 1) * PartitionCoef(1, 1) ' Z(water solids) = Z(water
    solution) x K(solids in water, water)
    ZvalueSub(2, 2) = ZvalueSub(1, 1) * PartitionCoef(2, 1) ' particles water
    ZvalueSub(3, 2) = ZvalueSub(1, 1) * PartitionCoef(1, 1) ' Z(inflow water solids) = Z(water
    solution) x K(solids in water, water)
    ZvalueSub(1, 5) = ZvalueSub(1, 1) * PartitionCoef(5, 1) ' resuspended water

    GRate_n(8) = Kaw / (Kaw / MTCwaterSide + 1 / MTCairSide)

    If ZvalueSub(4, 4) = 0 Then
            KQA = 0
        Else
            KQA = KQW / Kaw
    End If ' ZvalueSub(4,4) = 0

End If

'Reaction Rates
'ReactRate(2) = NatLog2 / HalfLife(2)                  ' sediment reaction rate constant (1/h)
'ReactRate(1) = NatLog2 / HalfLife(1)                  ' water reaction rate constant (1/h)
'Reaction Rates
' *******************************************************
' * 6/19/2007 added test to allow                       *
' * zero halflife specification to mean zero degradation*
' *******************************************************
```

```
' sediment reaction rate constant (1/h)
If HalfLife(2) = 0 Then
    ReactRate(2) = 0
Else
    ReactRate(2) = NatLog2 / HalfLife(2)
End If

' water reaction rate constant (1/h)
If HalfLife(1) = 0 Then
    ReactRate(1) = 0
Else
    ReactRate(1) = NatLog2 / HalfLife(1)
End If


' Transfer rates in m3/h
GRate(1) = GRate_n(1) * Area(1) / HoursPerDay / Density(2)   ' sediment burial
GRate(2) = Volume(2) * ReactRate(2)                          ' reaction in sediment
GRate(3) = GRate_n(3) * Area(1) / HoursPerDay / Density(2)   ' sediment resuspension
GRate(4) = GRate_n(4) * Area(2)                              ' sed-water diffusion
GRate(6) = GRate_n(6) * Area(1) / HoursPerDay / Density(1)   ' sediment deposition
GRate(7) = Volume(1) * ReactRate(1)                          ' reaction in water
GRate(8) = GRate_n(8) * Area(1)                              ' volatilization
GRate(11) = GRate(10) * VolFract(1)                          ' particle outflow
GRate(12) = GRate_n(12) * Area(1) / HoursPerYear             ' rain rate from m/year to m3/h
GRate(13) = GRate(12) * VolFract(4) * ScavengRatio           ' wet particle deposition
GRate(14) = GRate_n(14) * Area(1) * VolFract(4)              ' dry particle deposition
GRate(16) = GRate(15) * VolFract(3)                          ' particle inflow

GRateNetDepo = GRate(14) + GRate(13)                         ' Total deposition rate of aerosols

'Calculated Z values for bulk media
ZvalueBulk(1) = ZvalueSub(1, 1) * (1 - VolFract(1)) + ZvalueSub(1, 2) * VolFract(1) 'Z for bulk
water
ZvalueBulk(2) = ZvalueSub(1, 1) * (1 - VolFract(2)) + ZvalueSub(2, 2) * VolFract(2) 'Z for bulk
sediment

ZvalueBulk(3) = ((GRate(15) - GRate(16)) * ZvalueSub(1, 1) + GRate(16) * ZvalueSub(3, 2)) /
GRate(15) 'Z Value for bulk inflow water

ZvalueBulk(4) = ZvalueSub(4, 4) * (1 - VolFract(4)) + VolFract(4) * ZvalueSub(4, 2)  'Z for
bulk air

' some temporary variable to treat volatilization - fix this later. - December 19, 2006 ----
'Public VolatDair As Single, VolatDwater As Single
VolatDair = Area(1) * MTCairSide * ZvalueSub(4, 4)
VolatDwater = Area(1) * MTCwaterSide * ZvalueSub(1, 1)

'Calculation of D values
Dvalue(1) = GRate(1) * ZvalueSub(2, 2)          ' burial
Dvalue(2) = GRate(2) * ZvalueBulk(2)            ' sediment transformation
Dvalue(3) = GRate(3) * ZvalueSub(1, 5)          ' sediment resuspension
Dvalue(4) = GRate(4) * ZvalueSub(1, 1)          ' sediment-water diffusion
Dvalue(6) = GRate(6) * ZvalueSub(1, 2)          ' sediment deposition
```

```
Dvalue(7) = GRate(7) * ZvalueBulk(1)             ' water transformation
Dvalue(8) = GRate(8) * ZvalueSub(1, 1)           ' volatilization
Dvalue(10) = GRate(10) * ZvalueSub(1, 1)         ' water outflow
Dvalue(11) = GRate(11) * ZvalueSub(1, 2)         ' water particle outflow
Dvalue(12) = GRate(12) * ZvalueSub(1, 1)         ' rain dissolution
Dvalue(13) = GRate(13) * ZvalueSub(4, 2)         ' wet particle deposition
Dvalue(14) = GRate(14) * ZvalueSub(4, 2)         ' dry particle deposition
Dvalue(15) = GRate(15) * ZvalueSub(1, 1)         ' water inflow
Dvalue(16) = GRate(16) * ZvalueSub(3, 2)         ' water particle inflow

' Calculate total D values not previously calculated
DAirToWater = Dvalue(8) + Dvalue(14) + Dvalue(13) + Dvalue(12) ' total of air to water D values
DInflowToWater = Dvalue(15) + Dvalue(16) ' total of inflow D values
DsystemLoss(1) = Dvalue(7) + Dvalue(8) + Dvalue(10) + Dvalue(11) ' transformation in water +
volatilization + water and particle outflows
DWaterToSed = Dvalue(4) + Dvalue(6) 'sed-water diffusion + sed dep
Dtotal(1) = DsystemLoss(1) + DWaterToSed
DsystemLoss(2) = Dvalue(1) + Dvalue(2) ' sed burial + transform in sed
DSedToWater = Dvalue(3) + Dvalue(4) 'sed resus + sed-water diff
Dtotal(2) = DsystemLoss(2) + DSedToWater

If keepcalcdtimestepconstant = False Then
    Call Main.CalcMinTimestep
End If

If InitialTimestepCalc = True Then
    GoTo inittimestepcalc
End If

CalcResponse

SkipZDcalcs:
    '#############################################################################
    '## SkipZDcalcs:                                                             ##
    '##                                                                          ##
    '## JUMP INTO HERE DURING DYNAMIC ITERATIONS                                 ##
    '##                                                                          ##
    '#############################################################################

If EmisDyn = True And yearly = 0 Then
    GoTo endsim
End If


If keepcalcdtimestepconstant = False Then
    Call Main.CalcMinTimestep
End If

If InitialTimestepCalc = True Then
    GoTo inittimestepcalc
End If

Emis(4) = Emis_kg(4) * kg2g / HoursPerYear / MolMass    ' discharge rate mol/h to air
'LEP modification distribute drift over first N hours of day 7/24/2013
```

```
'If within the runoff time period, distribute daily rate over runoff time
If ModuloTime_h <= EmissTime_h Then
    Emis(1) = Emis_kg(1) * kg2g / (365 * EmissTime_h) / MolMass
'Outside the runoff time period, rates are zero
Else
    Emis(1) = 0
End If

Emis(2) = Emis_kg(2) * kg2g / HoursPerYear / MolMass  ' discharge rate mol/h to sediment

ConcBulk(3) = ConcBulk_n(3) / (MolMass * g2ug)       ' inflow water concn mol/m3
ConcBulk(4) = ConcBulk_n(4) / (MolMass * g2ug)       ' air concn mol/m3

'Fugacity(3) = ConcBulk(3) / ZvalueBulk(3)
'Fugacity(4) = ConcBulk(4) / ZvalueBulk(4)
' Calculate inflow and air fugacities
'6/19/2007 added test for zero ZvalueBulk
For i = 3 To 4
    If ZvalueBulk(i) = 0 Then
        Fugacity(i) = 0
    Else
        Fugacity(i) = ConcBulk(i) / ZvalueBulk(i)
    End If
Next i

'##############################################################
'##                                                          ##
'## DO THE DYNAMIC CALCULATIONS FOR FUGACITY                 ##
'## based on new D & Z VALUES                                ##
'##                                                          ##
'##############################################################

If Dynamic = True Then
    'by now we must have the Dvalues calculated correctly
    'specifically we need DAirToWater, DInflowToWater,DSedToWater,DTotal(1) for (1) i.e. the
    water
    'and DWaterToSed,DTotal(2) for (2) i.e. the sediment
    Call QWASI_Dyn_Fugacity
    GoTo skipfugcalcs
    'AND JUMP OVER THESE STEADY STATE FUGACITY CALCS
    'BELOW HERE
End If


' Calculate steady-state water and sediment fugacities
Fugacity(1) = (Emis(1) + Fugacity(3) * DInflowToWater + Fugacity(4) * DAirToWater) _
        / (DsystemLoss(1) + (DWaterToSed * DsystemLoss(2) / Dtotal(2)))
Fugacity(2) = Fugacity(1) * DWaterToSed / Dtotal(2)


'concentrations in bulk phases in mol/m3
' check that solubility is not exceeded ---- added January 19, 2006 ----
' fugacity of water is limited by the dissolved phase LKR June 2007
'inventory of water is limited by the dissolved phase LKR June 28 2007
```

```
'define maximum inventory of the water compartment
Inv_mol_1_max = (WatSol_mol / ZvalueSub(1, 1)) * ZvalueBulk(1) * Volume(1)

'check to see if solubility is exceeded
If (Fugacity(1) * ZvalueSub(1, 1)) > WatSol_mol Then
    dInv_mol = Inv_mol(1) - Inv_mol_1_max
    Inv_mol(5) = Inv_mol(5) + dInv_mol '((Fugacity(1) * ZvalueSub(1, 1) - WatSol_mol) *
    Volume(1))
    Inv_mol(1) = Inv_mol(1) - dInv_mol

    Fugacity(1) = WatSol_mol / ZvalueSub(1, 1)
    'Inv_mol(1) = Fugacity(1) * ZvalueBulk(1) * Volume(1)
Else
    ' no correction needed - no pure chemical present
    Fugacity(5) = 0
    Inv_mol(5) = 0
End If

skipfugcalcs:
    '############################################################################
    '## skipfugcalcs:                                                          ##
    '##                                                                        ##
    '## JUMP BACK IN HERE FOR DYNAMIC CALCULATIONS                             ##
    '##                                                                        ##
    '############################################################################

ConcBulk(1) = Fugacity(1) * ZvalueBulk(1) ' concn in bulk water
ConcBulk(2) = Fugacity(2) * ZvalueBulk(2) ' concn in bulk sediment

'concentrations in bulk phases in kg/m3
ConcBulk_kg(1) = ConcBulk(1) * MolMass / kg2g ' concn in bulk water
ConcBulk_kg(2) = ConcBulk(2) * MolMass / kg2g ' concn in bulk sediment
ConcBulk_kg(3) = ConcBulk(3) * MolMass / kg2g ' concn in bulk water
ConcBulk_kg(4) = ConcBulk(4) * MolMass / kg2g ' concn in bulk water

'concentrations in bulk phases with alternate units
ConcBulk_n(1) = ConcBulk(1) * MolMass * g2ug ' concn in bulk water in units ng/L
ConcBulk_n(2) = ConcBulk(2) * MolMass * g2ng ' concn in bulk sediment in units of ng/m3

'concentrations in subcompartments in mol/m3
ConcSub(1, 1) = Fugacity(1) * ZvalueSub(1, 1) ' concn in water solution
ConcSub(1, 2) = Fugacity(1) * ZvalueSub(1, 2) ' concn on particulates
ConcSub(2, 1) = Fugacity(2) * ZvalueSub(1, 1) ' concn in sed pore water
ConcSub(2, 2) = Fugacity(2) * ZvalueSub(2, 2) ' concn on sediment solids
ConcSub(3, 1) = Fugacity(3) * ZvalueSub(1, 1) ' concn in inflow water solution
ConcSub(3, 2) = Fugacity(3) * ZvalueSub(3, 2) ' concn in inflow water particles
ConcSub(4, 2) = Fugacity(4) * ZvalueSub(4, 2) ' conc on aerosols
ConcSub(4, 4) = Fugacity(4) * ZvalueSub(4, 4) ' conc in air vapour

'concentrations in subcompartments in kg/m3
ConcSub_kg(1, 1) = ConcSub(1, 1) * MolMass / kg2g ' concn in water solution
ConcSub_kg(1, 2) = ConcSub(1, 2) * MolMass / kg2g ' concn on particulates
ConcSub_kg(2, 1) = ConcSub(2, 1) * MolMass / kg2g ' concn in sed pore water
```

```
ConcSub_kg(2, 2) = ConcSub(2, 2) * MolMass / kg2g ' concn on sediment solids
ConcSub_kg(3, 1) = ConcSub(3, 1) * MolMass / kg2g ' concn in inflow water solution
ConcSub_kg(3, 2) = ConcSub(3, 2) * MolMass / kg2g ' concn in inflow water particles
ConcSub_kg(4, 2) = ConcSub(4, 2) * MolMass / kg2g ' conc on aerosols
ConcSub_kg(4, 4) = ConcSub(4, 4) * MolMass / kg2g ' conc in air vapour


'concentrations in subcompartments in alternate units
ConcSub_n(1, 2) = ConcSub(1, 2) * MolMass * g2ng / Density(1) ' concn on particulates ng/g
ConcSub_n(2, 1) = ConcSub(2, 1) * MolMass * g2ug              ' concn in sed pore water - units
of ng/L
ConcSub_n(2, 2) = ConcSub(2, 2) * MolMass * g2ng / Density(2) ' ng/g sed solids

'concentrations on solid fractions in terms of bulk volumes
ConcPerVolBulk(1) = ConcSub(1, 2) * VolFract(1) ' concentration on particles per unit volume of
water
ConcPerVolBulk(2) = ConcSub(2, 2) * VolFract(2) ' concentration on solids per unit volume of
sediment
ConcPerVolBulk(3) = ConcSub(3, 2) * VolFract(3) ' concentration on particles per unit volume of
inflow water
ConcPerVolBulk(4) = ConcSub(4, 2) * VolFract(4) ' concentration on aerosols per unit volume of
air

'rain concentration
ConcRain = Fugacity(4) * (ZvalueSub(1, 1) + VolFract(4) * ScavengRatio * ZvalueSub(4, 2))
ConcRain_kg = ConcRain * MolMass / kg2g
ConcRain_n = ConcRain * MolMass * g2ug ' ng/L

'amounts in subcompartments in mol/m3
'LEP note: be careful when using AmountSub for diagnostics, VolSub is not a dynamic variable!
'It most likely should be, not sure why original AGRO doesn't update it with PRZM sediment
AmountSub(1, 1) = ConcSub(1, 1) * VolSub(1, 1) ' amount in water solution
AmountSub(1, 2) = ConcSub(1, 2) * VolSub(1, 2) ' amount on water particles
AmountSub(2, 1) = ConcSub(2, 1) * VolSub(2, 1) ' amount in sediment pore water
AmountSub(2, 2) = ConcSub(2, 2) * VolSub(2, 2) ' amount on sediment solids

'amounts in subcompartments in kg/m3
AmountSub_kg(1, 1) = AmountSub(1, 1) * MolMass / kg2g ' amount in water solution
AmountSub_kg(1, 2) = AmountSub(1, 2) * MolMass / kg2g ' amount on water particles
AmountSub_kg(2, 1) = AmountSub(2, 1) * MolMass / kg2g ' amount in sediment pore water
AmountSub_kg(2, 2) = AmountSub(2, 2) * MolMass / kg2g ' amount on sediment solids

'calculate amounts in bulk water and sediment (i.e., wet weight sediment) in mol and kg
For medium = 1 To 2
    AmountBulk(medium) = ConcBulk(medium) * Volume(medium)
    AmountBulk_kg(medium) = AmountBulk(medium) * MolMass / kg2g
Next medium



If AmountBulk(1) = Inv_mol(1) Then
    che = True
End If

'add in pure phase to inventory calculations (and percents) LKR - June 26 2007
```

```
AmountBulk(5) = Inv_mol(5)
AmountBulk_kg(5) = AmountBulk(5) * MolMass / kg2g

AmountTotal = AmountBulk(1) + AmountBulk(2) + AmountBulk(5)
AmountTotal_kg = AmountTotal * MolMass / kg2g

If AmountTotal_kg = 0 Then GoTo skippct

AmountBulk_percent(1) = 100 * AmountBulk_kg(1) / AmountTotal_kg
AmountSub_percent(1, 1) = 100 * AmountSub_kg(1, 1) / AmountTotal_kg
AmountSub_percent(1, 2) = 100 * AmountSub_kg(1, 2) / AmountTotal_kg
AmountBulk_percent(2) = 100 * AmountBulk_kg(2) / AmountTotal_kg
AmountSub_percent(2, 1) = 100 * AmountSub_kg(2, 1) / AmountTotal_kg
AmountSub_percent(2, 2) = 100 * AmountSub_kg(2, 2) / AmountTotal_kg
AmountBulk_percent(5) = 100 * AmountBulk_kg(5) / AmountTotal_kg
AmountTotal_percent = AmountBulk_percent(1) + AmountBulk_percent(2) + AmountBulk_percent(5)

skippct:

'6/22/2007 to allow zero values in denominator when
'various processes are not operational (zeroed out)
For i = 1 To 4
    If ZvalueBulk(i) = 0 Then
        AmountSorbed(i) = 0
    Else
        AmountSorbed(i) = 100 * VolFract(i) * ZvalueSub(i, 2) / ZvalueBulk(i)
    End If
Next i

'Calculate fluxes in mol/h
Flux(1) = Dvalue(1) * Fugacity(2) ' sediment burial
Flux(2) = Dvalue(2) * Fugacity(2) ' sediment transformation
Flux(3) = Dvalue(3) * Fugacity(2) ' sediment resuspension
Flux(4) = Dvalue(4) * Fugacity(1) ' water to sediment diffusion
Flux(5) = Dvalue(4) * Fugacity(2) ' sediment to water diffusion
Flux(6) = Dvalue(6) * Fugacity(1) ' sediment deposition
Flux(7) = Dvalue(7) * Fugacity(1) ' water transformation
Flux(8) = Dvalue(8) * Fugacity(1) ' water to air volatilization
Flux(9) = Dvalue(8) * Fugacity(4) ' air to water volatilization
Flux(10) = Dvalue(10) * Fugacity(1) ' outflow water
Flux(11) = Dvalue(11) * Fugacity(1) ' outflow particles
Flux(12) = Dvalue(12) * Fugacity(4) ' rain
Flux(13) = Dvalue(13) * Fugacity(4) ' wet deposition
Flux(14) = Dvalue(14) * Fugacity(4) ' dry deposition
Flux(15) = Dvalue(15) * Fugacity(3) ' inflow water
Flux(16) = Dvalue(16) * Fugacity(3) ' inflow particles

'convert fluxes from mol/h to kg/year
molh2kgy = MolMass * HoursPerYear / 1000
For Process = 1 To 16
    Flux_kg(Process) = Flux(Process) * molh2kgy
Next Process

'sums of fluxes in kg/y
```

```
FluxAir2Water_kg = Flux_kg(9) + Flux_kg(12) + Flux_kg(13) + Flux_kg(14)
FluxWater2Air_kg = Flux_kg(8)
FluxWater2Sed_kg = Flux_kg(4) + Flux_kg(6)
FluxSed2Water_kg = Flux_kg(3) + Flux_kg(5)

FluxInflow_kg = Flux_kg(15) + Flux_kg(16)
FluxOutflow_kg = Flux_kg(10) + Flux_kg(11)
FluxTransform_kg = Flux_kg(2) + Flux_kg(7)

' Chemical flux totals
' system
SystemInFlux_kg = Emis_kg(4) + Emis_kg(1) + FluxInflow_kg + FluxAir2Water_kg
SystemOutFlux_kg = FluxOutflow_kg + FluxTransform_kg + Flux_kg(1) + FluxWater2Air_kg

' chemical flux totals for water compartment
WaterInFlux_kg = Emis_kg(1) + FluxInflow_kg + FluxAir2Water_kg + FluxSed2Water_kg
WaterOutFlux_kg = FluxOutflow_kg + FluxWater2Air_kg + FluxWater2Sed_kg + Flux_kg(7)

' chemical flux totals for sediment compartment
SedInFlux_kg = Emis_kg(2) + FluxWater2Sed_kg
SedOutFlux_kg = FluxSed2Water_kg + Flux_kg(2) + Flux_kg(1)

'unit conversions
FluxAir2Water = FluxAir2Water_kg / molh2kgy
FluxWater2Air = FluxWater2Air_kg / molh2kgy
FluxWater2Sed = FluxWater2Sed_kg / molh2kgy
FluxSed2Water = FluxSed2Water_kg / molh2kgy

FluxInflow = FluxInflow_kg / molh2kgy
FluxOutflow = FluxOutflow_kg / molh2kgy
FluxTransform = FluxTransform_kg / molh2kgy

SystemInFlux = SystemInFlux_kg / molh2kgy
SystemOutFlux = SystemOutFlux_kg / molh2kgy

WaterInFlux = WaterInFlux_kg / molh2kgy
WaterOutFlux = WaterOutFlux_kg / molh2kgy

SedInFlux = SedInFlux_kg / molh2kgy
SedOutFlux = SedOutFlux_kg / molh2kgy

' residence times
If AmountTotal_kg = 0 Then GoTo skiprestimes

ResTimeWater_h = AmountBulk(1) / WaterOutFlux
If SedOutFlux = 0 Then GoTo nosedoutyet
ResTimeSed_h = AmountBulk(2) / SedOutFlux
nosedoutyet:
ResTimeOverall_h = AmountTotal / SystemOutFlux

ResTimeWater_d = ResTimeWater_h / HoursPerDay
ResTimeSed_d = ResTimeSed_h / HoursPerDay
ResTimeOverall_d = ResTimeOverall_h / HoursPerDay
```

```
ResTimeWater_y = ResTimeWater_h / HoursPerYear
ResTimeSed_y = ResTimeSed_h / HoursPerYear
ResTimeOverall_y = ResTimeOverall_h / HoursPerYear

' residence time in the system calculated as net system losses
' not the residence time in each medium but the residence time in the system
' based on the Whelan report
ResTimeNetWater_h = AmountBulk(1) / (FluxOutflow + FluxWater2Air + Flux(7))
If (Flux(1) + Flux(2)) = 0 Then GoTo nonetressed
ResTimeNetSed_h = AmountBulk(2) / (Flux(1) + Flux(2))
ResTimeNetSystem_h = 100 / (AmountBulk_percent(1) / ResTimeNetWater_h + AmountBulk_percent(2) /
ResTimeNetSed_h)
nonetressed:

ResTimeNetWater_d = ResTimeNetWater_h / HoursPerDay
ResTimeNetSed_d = ResTimeNetSed_h / HoursPerDay
ResTimeNetSystem_d = ResTimeNetSystem_h / HoursPerDay

ResTimeNetWater_y = ResTimeNetWater_h / HoursPerYear
ResTimeNetSed_y = ResTimeNetSed_h / HoursPerYear
ResTimeNetSystem_y = ResTimeNetSystem_h / HoursPerYear

skiprestimes:
inittimestepcalc:

If foody = True Then
    Call FoodWeb.Input_foodweb
End If
nofoodwebtemp:

If InitialTimestepCalc = True Then
    GoTo inittimestepcalcs
End If
If Dynamic = False Then
    Call Outputs.OutputSteady
End If
endsim:
inittimestepcalcs:
End Sub

Sub CalcResponse()

'Calculate response times for water processes
RespondTimeWater_h(3) = Volume(1) * ZvalueBulk(1) / Dvalue(3)  ' sediment resuspension
RespondTimeWater_h(4) = Volume(1) * ZvalueBulk(1) / Dvalue(4) ' sediment-water diffusion
RespondTimeWater_h(6) = Volume(1) * ZvalueBulk(1) / Dvalue(6)  ' sediment deposition
'6/19/2007 allow zero degradation
If Dvalue(7) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(7) = 100000000000#
Else
    RespondTimeWater_h(7) = Volume(1) * ZvalueBulk(1) / Dvalue(7) ' water transformation
End If

If Dvalue(8) = 0 Then ' set the response time to arbitarily large value if D is zero
```

```
    RespondTimeWater_h(8) = 100000000000#
Else
    RespondTimeWater_h(8) = Volume(1) * ZvalueBulk(1) / Dvalue(8)  ' volatilization
End If

'6/19/2007 to allow zero outflow
If Dvalue(10) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(10) = 100000000000#
Else
    RespondTimeWater_h(10) = Volume(1) * ZvalueBulk(1) / Dvalue(10) ' water outflow
End If

'6/19/2007 to allow zero denominator
If Dvalue(11) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(11) = 100000000000#
Else
    RespondTimeWater_h(11) = Volume(1) * ZvalueBulk(1) / Dvalue(11) ' water particle outflow
End If


If Dvalue(12) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(12) = 100000000000#
Else
    RespondTimeWater_h(12) = Volume(1) * ZvalueBulk(1) / Dvalue(12)  ' rain dissolution
End If
If Dvalue(13) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(13) = 100000000000#
Else
    RespondTimeWater_h(13) = Volume(1) * ZvalueBulk(1) / Dvalue(13) ' wet particle deposition
End If
If Dvalue(14) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(14) = 100000000000#
Else
    RespondTimeWater_h(14) = Volume(1) * ZvalueBulk(1) / Dvalue(14)  ' dry particle deposition
End If

'6/19/2007 to allow zero inflow
If Dvalue(15) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(15) = 100000000000#
Else
    RespondTimeWater_h(15) = Volume(1) * ZvalueBulk(1) / Dvalue(15) ' water inflow
End If

'6/19/2007 to allow zero inflow
If Dvalue(16) = 0 Then ' set the response time to arbitarily large value if D is zero
    RespondTimeWater_h(16) = 100000000000#
Else
    RespondTimeWater_h(16) = Volume(1) * ZvalueBulk(1) / Dvalue(16)  ' water particle inflow
End If

' unit conversions
RespondTimeWater_d(3) = RespondTimeWater_h(3) / HoursPerDay
RespondTimeWater_d(4) = RespondTimeWater_h(4) / HoursPerDay
RespondTimeWater_d(6) = RespondTimeWater_h(6) / HoursPerDay
```

```
RespondTimeWater_d(7) = RespondTimeWater_h(7) / HoursPerDay
RespondTimeWater_d(8) = RespondTimeWater_h(8) / HoursPerDay
For Process = 10 To 16
    RespondTimeWater_d(Process) = RespondTimeWater_h(Process) / HoursPerDay
Next Process

RespondTimeWater_y(3) = RespondTimeWater_h(3) / HoursPerYear
RespondTimeWater_y(4) = RespondTimeWater_h(4) / HoursPerYear
RespondTimeWater_y(6) = RespondTimeWater_h(6) / HoursPerYear
RespondTimeWater_y(7) = RespondTimeWater_h(7) / HoursPerYear
RespondTimeWater_y(8) = RespondTimeWater_h(8) / HoursPerYear
For Process = 10 To 16
    RespondTimeWater_y(Process) = RespondTimeWater_h(Process) / HoursPerYear
Next Process


' response times for sediment processes
'6/19/2007 to allow zero dvalues
For i = 1 To 6
    If i = 5 Then
        i = 6
    End If
    ' Dvalue(1)  ' burial
    ' Dvalue(2)  ' sediment transformation
    ' Dvalue(3)  ' sediment resuspension
    ' Dvalue(4)  ' sediment-water diffusion
    ' Dvalue(6)  ' sediment deposition
    If Dvalue(i) = 0 Then
        RespondTimeSed_h(i) = 100000000000000# 'arbitrarily long time
    Else
        RespondTimeSed_h(i) = Volume(2) * ZvalueBulk(2) / Dvalue(i)  ' sediment transformation
    End If
Next i

' unit conversions
RespondTimeSed_d(1) = RespondTimeSed_h(1) / HoursPerDay
RespondTimeSed_d(2) = RespondTimeSed_h(2) / HoursPerDay
RespondTimeSed_d(3) = RespondTimeSed_h(3) / HoursPerDay
RespondTimeSed_d(4) = RespondTimeSed_h(4) / HoursPerDay
RespondTimeSed_d(6) = RespondTimeSed_h(6) / HoursPerDay

RespondTimeSed_y(1) = RespondTimeSed_h(1) / HoursPerYear
RespondTimeSed_y(2) = RespondTimeSed_h(2) / HoursPerYear
RespondTimeSed_y(3) = RespondTimeSed_h(3) / HoursPerYear
RespondTimeSed_y(4) = RespondTimeSed_h(4) / HoursPerYear
RespondTimeSed_y(6) = RespondTimeSed_h(6) / HoursPerYear




End Sub

Sub Dyn_Params()
```

```
'Read in and calculate new Z- and D- values for each dynamic timestep
If ChemicalType = 1 Then
    'Calculate Z values for subcompartments
    ZvalueSub(4, 4) = 1 / GasConst / ChemPropTemp_K ' Z for air
    ZvalueSub(1, 1) = 1 / HenryLaw                  ' Z for water
    ZvalueSub(1, 2) = Koc * OrgCarbFract(1) * Density(1) * ZvalueSub(1, 1) / g2ug 'Z for water
    particles
    ZvalueSub(3, 2) = Koc * OrgCarbFract(3) * Density(1) * ZvalueSub(1, 1) / g2ug 'Z for inflow
    water particles
    ZvalueSub(2, 2) = Koc * OrgCarbFract(2) * Density(2) * ZvalueSub(1, 1) / g2ug 'Z for
    sediment particles
    ZvalueSub(1, 5) = Koc * OrgCarbFract(5) * Density(5) * ZvalueSub(1, 1) / g2ug 'Z for resusp
    sediment particles
    ZvalueSub(4, 2) = 6000000! / VapPressLiquid * ZvalueSub(4, 4)  'Z for aerosol particles

    'dimensionless partition coefficients
    PartitionCoef(1, 1) = ZvalueSub(1, 2) / ZvalueSub(1, 1) ' particles - water
    PartitionCoef(2, 1) = ZvalueSub(2, 2) / ZvalueSub(1, 1) ' sed solids - water
    PartitionCoef(5, 1) = ZvalueSub(1, 5) / ZvalueSub(1, 1) ' resus sed - water
    PartitionCoef(4, 2) = ZvalueSub(4, 2) / ZvalueSub(4, 4) ' aerosol - air

    'overall air - water mass transfer coefficient (m/h)
    GRate_n(8) = 1 / (1 / MTCwaterSide + 1 / (MTCairSide * Kaw))
End If
If ChemicalType = 2 Then 'Chemical type is type 2 then calculate type 2 specific calculations
end of type 1 specific

    'calculate the dimensionless partition coefficients
    PartitionCoef(1, 1) = PartitionCoef_Lkg(1, 1) * Density_kg(1) / 1000   ' water particles -
    water
    PartitionCoef(2, 1) = PartitionCoef_Lkg(2, 1) * Density_kg(2) / 1000   ' sediment solids -
    water
    PartitionCoef(5, 1) = PartitionCoef_Lkg(5, 1) * Density_kg(1) / 1000   ' resuspended
    Particles solids - water

    'dimensional to dimensionless
    ZvalueSub(1, 1) = 1 'water
    ZvalueSub(4, 4) = ZvalueSub(1, 1) * Kaw                 ' air
    ZvalueSub(4, 2) = ZvalueSub(1, 1) * KQW                 ' aerosol water
    ZvalueSub(1, 2) = ZvalueSub(1, 1) * PartitionCoef(1, 1) ' Z(water solids) = Z(water
    solution) x K(solids in water, water)
    ZvalueSub(2, 2) = ZvalueSub(1, 1) * PartitionCoef(2, 1) ' particles water
    ZvalueSub(3, 2) = ZvalueSub(1, 1) * PartitionCoef(1, 1) ' Z(inflow water solids) = Z(water
    solution) x K(solids in water, water)
    ZvalueSub(1, 5) = ZvalueSub(1, 1) * PartitionCoef(5, 1) ' resuspended water

    GRate_n(8) = Kaw / (Kaw / MTCwaterSide + 1 / MTCairSide)

    If ZvalueSub(4, 4) = 0 Then
            KQA = 0
        Else
            KQA = KQW / Kaw
    End If ' ZvalueSub(4,4) = 0
```

```
End If

'Else ' Chemical is Type 3

'End If ' chemical specific type calculations

'Reaction Rates
' *******************************************************
' * 6/19/2007 added test to allow                       *
' * zero halflife specification to mean zero degradation*
' *******************************************************

If [AGRO!P6] Then 'adjust degradation rate by environmental temperature
    pondTemp = getCurrentPondTemp()
    ' sediment reaction rate constant (1/h)
    ReactRate(2) = adjustReactRate(HalfLife(2), [Chemical!B24], WorksheetFunction.Max(4#,
    pondTemp))
    ' water reaction rate constant (1/h)
    ReactRate(1) = adjustReactRate(HalfLife(1), [Chemical!B25], WorksheetFunction.Max(0#,
    pondTemp))
Else 'no temperature adjustment
    ' sediment reaction rate constant (1/h)
    If HalfLife(2) = 0 Then
        ReactRate(2) = 0
    Else
        ReactRate(2) = NatLog2 / HalfLife(2)
    End If

    ' water reaction rate constant (1/h)
    If HalfLife(1) = 0 Then
        ReactRate(1) = 0
    Else
        ReactRate(1) = NatLog2 / HalfLife(1)
    End If
End If

' Transfer rates in m3/h
GRate(1) = GRate_n(1) * Area(1) / HoursPerDay / Density(2)   ' sediment burial
GRate(2) = Volume(2) * ReactRate(2)                           ' reaction in sediment
GRate(3) = GRate_n(3) * Area(1) / HoursPerDay / Density(2)   ' sediment resuspension
GRate(4) = GRate_n(4) * Area(2)                               ' sed-water diffusion

GRate(7) = Volume(1) * ReactRate(1)                           ' reaction in water
GRate(8) = GRate_n(8) * Area(1)                               ' volatilization

'Get time-variable rates
Workbooks(model$).Activate
'this option allows user of original PRZM data stored in the model, rather than
'the processed/massaged data in QWASI units, EmisDynDirect uses actually P2E values
baseFlowParticlesVolume = (baseFlowParticles * 1000# / (1000000# * Density_kg(3))) * baseFlow
If EmisDynDirect Then
    Sheets("GetPRZM_Files").Select
    FieldArea = [b2]
    [a3].Select
```

```
    runoffdepth2 = ActiveCell.Offset(PRZMsimday, 7)
    erosoilloss2 = ActiveCell.Offset(PRZMsimday, 9)
    precipdata2 = ActiveCell.Offset(PRZMsimday, 11)

    'If within the runoff time period, distribute daily rate over runoff time
    If ModuloTime_h <= RunoffTime_h Then
        PRZM_In_W = runoffdepth2 * FieldArea * 10000 / (100 * RunoffTime_h) 'cm/day to m3/(hr
        of runofftime)
        PRZM_In_P = erosoilloss2 * FieldArea * 1000# / (Density_kg(3) * RunoffTime_h)
        'tonnes/Ha.day to m3/(hr of runofftime)
    'Outside the runoff time period, rates are zero
    Else
        PRZM_In_W = 0
        PRZM_In_P = 0
    End If

    'Rain rate
    GRate(12) = (precipdata2 / (100 * 24)) * Area(1) / HoursPerYear
    PRZM_Tot_Inflow = baseFlow + baseFlowParticlesVolume + PRZM_In_W + PRZM_In_P ' total inflow
    m3/h
Else
    Sheets("PRZMforInput").Select
    [a3].Select

    'If within the runoff time period, distribute daily rate over runoff time
    If ModuloTime_h <= RunoffTime_h Then
        PRZM_In_W = (ActiveCell.Offset(PRZMsimday, 9) - baseFlow) * (24 / RunoffTime_h)
        PRZM_In_P = (ActiveCell.Offset(PRZMsimday, 10) - baseFlowParticlesVolume) * (24 /
        RunoffTime_h)
    'Outside the runoff time period, rates are zero
    Else
        PRZM_In_W = 0
        PRZM_In_P = 0
    End If
    GRate(12) = ActiveCell.Offset(PRZMsimday, 15)                        ' rain rate from m/year to
    m3/h
    PRZM_Tot_Inflow = baseFlow + baseFlowParticlesVolume + PRZM_In_W + PRZM_In_P ' total inflow
    m3/h
End If
'Avoid division by zero in particulate volume fraction calc
If PRZM_Tot_Inflow = 0 Then
    PRZM_VolFractP = 0
Else
    PRZM_VolFractP = (PRZM_In_P + baseFlowParticlesVolume) / PRZM_Tot_Inflow
End If
VolFract(3) = PRZM_VolFractP
GRate(15) = PRZM_In_W + baseFlow
GRate(10) = GRate(15)
'GRate(16) = GRate(15) * VolFract(3)                          ' particle inflow
GRate(16) = PRZM_Tot_Inflow * VolFract(3)                    ' particle inflow, LEP 2.27.14
GRate(11) = GRate(10) * VolFract(1)                          ' particle outflow with initial conc
GRate(6) = GRate_n(6) * Area(1) / HoursPerDay / Density(1)   ' sediment deposition
newSedDep = GRate(6) * Density(1) * HoursPerDay / Area(1)    ' allows output to show current
```

```
rate useful for confirming variable or fixed sedimentation scheme

If sedResuspPerCent > 0 Then
    Call QWASI.Dyn_Sed_Params
    '##############################################
    '##  scheme to implement use of p2e transfer ##
    '##  file sediment values in ADDITION to the ##
    '##  default sediment values for this envir. ##
    '## 1.2.8 version: implementing gradual decay##
    '## of the sediment load to mimic settling   ##
    '##############################################
End If

'attempt at sediment balance - not accomplished LKR May 29'07
'delta_inflowSed_Vol = timestep * (GRate(16) - GRate(11))     'volume sediment entering pond
'VolFract_old(1) = VolFract(1)
'VolFract(1) = ((VolFract_old(1) * Volume(1)) + delta_inflowSed_Vol) / Volume(1)

'rateconst_6 = GRate(6) / (VolFract_old(1) * Volume(1))
'GRate(6) = rateconst_6 * VolFract(1) * Volume(1)

GRate(13) = GRate(12) * VolFract(4) * ScavengRatio           ' wet particle deposition
GRate(14) = GRate_n(14) * Area(1) * VolFract(4)              ' dry particle deposition

GRateNetDepo = GRate(14) + GRate(13)                         ' Total deposition rate of aerosols

yearly = ActiveCell.Offset(PRZMsimday, 1)
monthy = ActiveCell.Offset(PRZMsimday, 2)
dayy = ActiveCell.Offset(PRZMsimday, 3)

If yearly = 0 Then
    GoTo endsimparams
End If


'Calculated Z values for bulk media
ZvalueBulk(1) = ZvalueSub(1, 1) * (1 - VolFract(1)) + ZvalueSub(1, 2) * VolFract(1) 'Z for bulk
water
ZvalueBulk(2) = ZvalueSub(1, 1) * (1 - VolFract(2)) + ZvalueSub(2, 2) * VolFract(2) 'Z for bulk
sediment
'ZvalueBulk(3) = (GRate(15) * ZvalueSub(1, 1) + GRate(16) * ZvalueSub(3, 2)) / (GRate(15) +
GRate(16)) 'Z Value for bulk inflow water
'6/19/2007 added test to allow zvaluebulk(3) to be correctly
'calculated if there is zero inflow
If GRate(15) + GRate(16) = 0 Then
    ZvalueBulk(3) = 0
Else
    'ZvalueBulk(3) = ((GRate(15) - GRate(16)) * ZvalueSub(1, 1) + GRate(16) * ZvalueSub(3, 2))
    / GRate(15) 'Z Value for bulk inflow water
    ZvalueBulk(3) = (1 - VolFract(3)) * ZvalueSub(1, 1) + VolFract(3) * ZvalueSub(3, 2) ' LEP
    3/6/2014 Z Value for bulk inflow water
End If
ZvalueBulk(4) = ZvalueSub(4, 4) * (1 - VolFract(4)) + VolFract(4) * ZvalueSub(4, 2)  'Z for
bulk air
```

```
' some temporary variable to treat volatilization - fix this later. - December 19, 2006 ----
'Public VolatDair As Single, VolatDwater As Single
VolatDair = Area(1) * MTCairSide * ZvalueSub(4, 4)
VolatDwater = Area(1) * MTCwaterSide * ZvalueSub(1, 1)

'Calculation of D values
Dvalue(1) = GRate(1) * ZvalueSub(2, 2)           ' burial
Dvalue(2) = GRate(2) * ZvalueBulk(2)             ' sediment transformation
Dvalue(3) = GRate(3) * ZvalueSub(1, 5)           ' sediment resuspension
Dvalue(4) = GRate(4) * ZvalueSub(1, 1)           ' sediment-water diffusion
Dvalue(6) = GRate(6) * ZvalueSub(1, 2)           ' sediment deposition
Dvalue(7) = GRate(7) * ZvalueBulk(1)             ' water transformation
Dvalue(8) = GRate(8) * ZvalueSub(1, 1)           ' volatilization
Dvalue(10) = GRate(10) * ZvalueSub(1, 1)         ' water outflow
Dvalue(11) = GRate(11) * ZvalueSub(1, 2)         ' water particle outflow
Dvalue(12) = GRate(12) * ZvalueSub(1, 1)         ' rain dissolution
Dvalue(13) = GRate(13) * ZvalueSub(4, 2)         ' wet particle deposition
Dvalue(14) = GRate(14) * ZvalueSub(4, 2)         ' dry particle deposition
Dvalue(15) = GRate(15) * ZvalueSub(1, 1)         ' water inflow
Dvalue(16) = GRate(16) * ZvalueSub(3, 2)         ' water particle inflow

' Calculate total D values not previously calculated
DAirToWater = Dvalue(8) + Dvalue(14) + Dvalue(13) + Dvalue(12) ' total of air to water D values
DInflowToWater = Dvalue(15) + Dvalue(16) ' total of inflow D values
DsystemLoss(1) = Dvalue(7) + Dvalue(8) + Dvalue(10) + Dvalue(11) ' transformation in water +
volatilization + water and particle outflows
DWaterToSed = Dvalue(4) + Dvalue(6) 'sed-water diffusion + sed dep
Dtotal(1) = DsystemLoss(1) + DWaterToSed
DsystemLoss(2) = Dvalue(1) + Dvalue(2) ' sed burial + transform in sed
DSedToWater = Dvalue(3) + Dvalue(4) 'sed resus + sed-water diff
Dtotal(2) = DsystemLoss(2) + DSedToWater

If tim = 0 Then GoTo endsimparams

'Below calculations determine bulk inflow concentration from PRZM inputs
'Dynamic emissions option must be true (emissions come from PRZM)
If EmisDyn = True Then
    'PRZM inputs read directly from P2E data in GetPRZM_Files worksheet
    'Active sheet and cell specified in conditional block above
    If EmisDynDirect Then
        apprate2 = ActiveCell.Offset(PRZMsimday, 4)
        appeff2 = ActiveCell.Offset(PRZMsimday, 5)
        pctdrift2 = ActiveCell.Offset(PRZMsimday, 6)
        Emis_kg(1) = apprate2 * (pctdrift2 / 100) * (Area(1) / 10000) * 365 'apprate in
        kg/Ha.day to kg/year (changing each day)

        'If within the runoff time period, distribute daily rate over runoff time
        If ModuloTime_h <= RunoffTime_h Then
            runofflux2 = ActiveCell.Offset(PRZMsimday, 8) / RunoffTime_h 'g/cm2/hr of runoff
            period
            eropestflux2 = ActiveCell.Offset(PRZMsimday, 10) / RunoffTime_h 'g/cm2/hr of runoff
            period
        'Outside the runoff time period, rates are zero
```

```
        Else
            runoffflux2 = 0
            eropestflux2 = 0
        End If

        If PRZM_Tot_Inflow = 0 Then
            PRZM_VolFractW = 0
            PRZM_VolFractP = 0
            PRZM_In_P_conc = 0
            PRZM_ConcSub(3, 1) = 0
            PRZM_ConcSub(3, 2) = 0
        Else
            PRZM_VolFractW = (PRZM_In_W + baseFlow) / PRZM_Tot_Inflow
            PRZM_VolFractP = (PRZM_In_P + baseFlowParticlesVolume) / PRZM_Tot_Inflow
            PRZM_In_P_conc = (baseFlowParticlesVolume + PRZM_In_P) * Density_kg(3) * 1000000# /
            ((GRate(15) + PRZM_In_W) * 1000#) 'concentration of solids in mg/L
            PRZM_ConcSub(3, 1) = runoffflux2 * 1000000000# * 10000# * FieldArea * 10000 / ((
            PRZM_In_W + baseFlow) * 1000#) 'concentration in ng/L inflow water phase
            PRZM_ConcSub(3, 2) = eropestflux2 * 1000000000# * 10000# * FieldArea * 10000 / ((
            PRZM_In_P + baseFlowParticlesVolume) * 1000#) 'concentration in ng/L inflow
            particulate phase
        End If

        PRZM_ConcBulk(3) = PRZM_VolFractW * PRZM_ConcSub(3, 1) + PRZM_VolFractP * PRZM_ConcSub(3
        , 2)
        ConcBulk_n(3) = PRZM_ConcBulk(3)

    'PRZM inputs read from PRZMforInput worksheet
    Else
        Emis_kg(1) = ActiveCell.Offset(PRZMsimday, 4)
        'If within the runoff time period, distribute daily concentration over runoff time
        If ModuloTime_h <= RunoffTime_h Then
            ConcBulk_n(3) = (ActiveCell.Offset(PRZMsimday, 7) * (ActiveCell.Offset(PRZMsimday, 9
            ) + ActiveCell.Offset(PRZMsimday, 10)) * 24) / (PRZM_Tot_Inflow * RunoffTime_h)
        'Outside the runoff time period, rates are zero
        Else
            ConcBulk_n(3) = 0
        End If

    End If
    ConcBulk_n(4) = 0 'no PRZM output for concentration in Air
End If

endsimparams:

CalcResponse

End Sub

Sub QWASI_Dyn_Fugacity()

'calculate change in inventory and change in fugacity in each
'compartment for each timestep
```

```
'Calculate input per timestep to each medium
'air does not change

'water
delta_inv_mol(1) = (Emis(1) + Fugacity(4) * DAirToWater + Fugacity(3) * DInflowToWater + Fug_old
(2) * DSedToWater - Fug_old(1) * Dtotal(1)) * timestep

'sediment
delta_inv_mol(2) = ((Fug_old(1) * DWaterToSed) - (Fug_old(2) * Dtotal(2))) * timestep

'Calculate new inventories in mol

'water
Inv_mol(1) = Inv_old(1) + delta_inv_mol(1)

'sediment
Inv_mol(2) = Inv_old(2) + delta_inv_mol(2)

'Calculate new fugacities for rate calculations in next timestep
For medium = 1 To 2
    Fugacity(medium) = Inv_mol(medium) / (Volume(medium) * ZvalueBulk(medium))
Next medium

'Fugacities for incoming media (inflow water and air) are already set
'define maximum inventory of the water compartment
Inv_mol_1_max = (WatSol_mol / ZvalueSub(1, 1)) * ZvalueBulk(1) * Volume(1)

'concentrations in bulk phases in mol/m3
' check that solubility is not exceeded ---- added January 19, 2006 ----
'fugacity of water is limited by the dissolved phase LKR - June 2007
'refinement of calculations - Don Mackay and Liisa Reid June 27 2007


'check for pure phase presence
If Inv_mol(5) = 0 Then

'check to see if solubility is exceeded
    If (Fugacity(1) * ZvalueSub(1, 1)) > WatSol_mol Then
        dInv_mol = Inv_mol(1) - Inv_mol_1_max
        Inv_mol(5) = Inv_mol(5) + dInv_mol '((Fugacity(1) * ZvalueSub(1, 1) - WatSol_mol) *
        Volume(1))
        Inv_mol(1) = Inv_mol(1) - dInv_mol

        Fugacity(1) = WatSol_mol / ZvalueSub(1, 1)
        'Inv_mol(1) = Fugacity(1) * ZvalueBulk(1) * Volume(1)
    Else
        ' no correction needed - no pure chemical present
        Fugacity(5) = 0
        Inv_mol(5) = 0
    End If

Else

'check to see if solubility is exceeded
```

```
    If (Fugacity(1) * ZvalueSub(1, 1)) > WatSol_mol Then
        dInv_mol = Inv_mol(1) - Inv_mol_1_max
        Inv_mol(5) = Inv_mol(5) + dInv_mol
        Inv_mol(1) = Inv_mol(1) - dInv_mol

        Fugacity(1) = WatSol_mol / ZvalueSub(1, 1)
    Else
       'check to see that inv_mol(5) does not dip below zero if it is decreasing
        dInv_mol = Inv_mol(1) - Inv_mol_1_max

        If Inv_mol(5) + dInv_mol < 0 Then
            dInv_mol = -1 * Inv_mol(5)
            Inv_mol(5) = Inv_mol(5) + dInv_mol
            Inv_mol(1) = Inv_mol(1) - dInv_mol
            Fugacity(1) = (Inv_mol(1) / Volume(1)) / ZvalueBulk(1)
        Else
            Inv_mol(5) = Inv_mol(5) + dInv_mol
            Fugacity(1) = WatSol_mol / ZvalueSub(1, 1)
            Inv_mol(1) = Fugacity(1) * ZvalueBulk(1) * Volume(1)
        End If

    End If



End If

End Sub

Sub Dyn_Sed_Params()
'###############################################
'##  scheme to implement use of p2e transfer ##
'##  file sediment values in ADDITION to the ##
'##  default sediment values for this envir. ##
'## 1.2.8 version: implementing gradual decay##
'## of the sediment load to mimic settling   ##
'###############################################
'this is now handled in a separate sub-routine called every timestep instead of every day
'If sedResuspPerCent > 0 And timestep > 0 Then
If timestep > 0 Then
    'note deposition = resuspension + burial
    'i.e. GRate(6)   = GRate(3)     + GRate(1)
    'GRate(6) is the total deposition, partitioned between burial and resuspension in the pond
    ' it excludes inflow and outflow normally
    ' this scheme starts to make it include the inflow sediments
    '           First add inflow particle volume to deposition
    '           6/16/2008 telecon with SHJ
    '           This is an EITHER OR situation
    '           if we implement PRZM varying sediment values
    '           we have to turn off the default values in the model
    '           in order for their still to be non-zero (possible model breakage if zero)
    '           we set the base deposition rate to be 2 g/m2/day and add in any P2E (GRATE16)
    values
```

```
    ' attempt at implementing a settling time scheme June 2008
    ' First estimate mass per m2 of system of sediment in the water column
    '  contributing to the total are
    '   base suspended sediment concentration
    '   incoming sediment in base inflow
    '   incoming sediment in runoff
    '   losses in outflow
    '   losses through deposition
    '   increases through re-suspension (which we're going to set at a percentage of deposition)
    ' let's consider this on a system basis then convert to per sq meter basis for GRate calcs


    'sedInWaterTot = ParticConc(1) * Volume(1) 'mg/L * m3 == g/m3 * m3 = g SEDIMENT SUSPENDED
    ' set outside main loop
    ' then add the incoming sediment, less outflow
    '                 base amt    + inflow m3/hr * density g/m3 * hours (and then take off
    particle outflow)
    ''''''''''''''''**************'''''''''''''''''''''''''**************''''''''''''''''''''''''''**************
    'sedRate (sedimentation rate) must be specified in hours^-1
    deltaSed = sedInWaterTot * (1 - Exp(-sedRate))
    'deltaSed is now in g for the system/hr
    'check that deltaSed doesn't drop us below our initial base level based on
    'the supplied suspended sediment concentration
    'deltased is in g/hr
    'we want this to be the actual change, actual change depends on what's lost
    '- that's the burial
    '  so set this into GRate(1) converting back from g/hr to m3/hr
    '  add in the base 1 g/m2/day burial rate
    GRate(1) = deltaSed / Density(1) + 1 * Area(1) / HoursPerDay / Density(2)
    'calculate GRate(6) deposition from GRate(1) burial by re-arranging original equation
    GRate(6) = GRate(1) * 100 / (100 - sedResuspPerCent)
    'and calculate GRate(3) resuspension the usual way
    GRate(3) = GRate(6) * sedResuspPerCent / 100
    'recalculate particle concentration
End If
'##############################################
'##  end of scheme to modify sedimentation   ##
'##  based on varying PRZM events            ##
'##############################################

End Sub





Sub Dyn_Sed_Calcs()
'this keeps track of the sediment (extra sediment above base susp solids) in the system
'needs to be separate from calculation of the parameters
If timestep > 0 Then
    If sedInWaterTot = 0 And GRate(16) > GRate(11) Then
        'we haven't had an event yet (so sedInWaterTot=0) ..
        ' and we have an event now because GRate(16)>GRate(11)
        ' so set sedinwatertot for the first time
        sedInWaterTot = (GRate(16) - GRate(11)) * Density(1) * timestep
```

```
    Else
        sedInWaterTot = sedInWaterTot + (GRate(16) - GRate(11)) * Density(1) * timestep
    End If
    'deltaSed is calculated in Dyn_Sed_Params
    'deltaSed is now in g for the system/hr
    'check that deltaSed doesn't drop us below our initial base level based on
    'the supplied suspended sediment concentration
    If sedInWaterTot - deltaSed * timestep < 0 Then
        sedInWaterTot = 0
        'reset the GRates appropriately
        GRate(1) = 1 * Area(1) / HoursPerDay / Density(2)
        GRate(3) = 1 * Area(1) / HoursPerDay / Density(2)
        GRate(6) = 2 * Area(1) / HoursPerDay / Density(2)
        ParticConc(1) = baseParticConc(1)
    Else
        sedInWaterTot = sedInWaterTot - deltaSed * timestep ' g
        ParticConc(1) = sedInWaterTot / Volume(1) + baseParticConc(1) ' mg/L (assumes g/m3 =
        mg/L)
    End If
    'recalculate volume fraction of solids in the water compartment
    VolFract(1) = ParticConc(1) / Density(1)              ' ParticConc(1) is in mg/L
End If
newSedDep = GRate(6) * Density(1) * HoursPerDay / Area(1)

End Sub

Sub Dyn_Sed_ZD()
    'Calculated Z values for bulk media
    ZvalueBulk(1) = ZvalueSub(1, 1) * (1 - VolFract(1)) + ZvalueSub(1, 2) * VolFract(1) 'Z for
    bulk water
    ZvalueBulk(2) = ZvalueSub(1, 1) * (1 - VolFract(2)) + ZvalueSub(2, 2) * VolFract(2) 'Z for
    bulk sediment
    'ZvalueBulk(3) = (GRate(15) * ZvalueSub(1, 1) + GRate(16) * ZvalueSub(3, 2)) / (GRate(15) +
    GRate(16)) 'Z Value for bulk inflow water
    '6/19/2007 added test to allow zvaluebulk(3) to be correctly
    'calculated if there is zero inflow
    If GRate(15) + GRate(16) = 0 Then
        ZvalueBulk(3) = 0
    Else
        'ZvalueBulk(3) = ((GRate(15) - GRate(16)) * ZvalueSub(1, 1) + GRate(16) * ZvalueSub(3,
        2)) / GRate(15) 'Z Value for bulk inflow water
        ZvalueBulk(3) = (1 - VolFract(3)) * ZvalueSub(1, 1) + VolFract(3) * ZvalueSub(3, 2) '
        LEP 3/6/2014 Z Value for bulk inflow water
    End If
    ZvalueBulk(4) = ZvalueSub(4, 4) * (1 - VolFract(4)) + VolFract(4) * ZvalueSub(4, 2)  'Z for
    bulk air

    ' some temporary variable to treat volatilization - fix this later. - December 19, 2006 ----
    'Public VolatDair As Single, VolatDwater As Single
    VolatDair = Area(1) * MTCairSide * ZvalueSub(4, 4)
    VolatDwater = Area(1) * MTCwaterSide * ZvalueSub(1, 1)

    'Calculation of D values
    Dvalue(1) = GRate(1) * ZvalueSub(2, 2)           ' burial
```

```
    Dvalue(2) = GRate(2) * ZvalueBulk(2)            ' sediment transformation
    Dvalue(3) = GRate(3) * ZvalueSub(1, 5)          ' sediment resuspension
    Dvalue(4) = GRate(4) * ZvalueSub(1, 1)          ' sediment-water diffusion
    Dvalue(6) = GRate(6) * ZvalueSub(1, 2)          ' sediment deposition
    Dvalue(7) = GRate(7) * ZvalueBulk(1)            ' water transformation
    Dvalue(8) = GRate(8) * ZvalueSub(1, 1)          ' volatilization
    Dvalue(10) = GRate(10) * ZvalueSub(1, 1)        ' water outflow
    Dvalue(11) = GRate(11) * ZvalueSub(1, 2)        ' water particle outflow
    Dvalue(12) = GRate(12) * ZvalueSub(1, 1)        ' rain dissolution
    Dvalue(13) = GRate(13) * ZvalueSub(4, 2)        ' wet particle deposition
    Dvalue(14) = GRate(14) * ZvalueSub(4, 2)        ' dry particle deposition
    Dvalue(15) = GRate(15) * ZvalueSub(1, 1)        ' water inflow
    Dvalue(16) = GRate(16) * ZvalueSub(3, 2)        ' water particle inflow

    ' Calculate total D values not previously calculated
    DAirToWater = Dvalue(8) + Dvalue(14) + Dvalue(13) + Dvalue(12) ' total of air to water D 
    values
    DInflowToWater = Dvalue(15) + Dvalue(16) ' total of inflow D values
    DsystemLoss(1) = Dvalue(7) + Dvalue(8) + Dvalue(10) + Dvalue(11) ' transformation in water 
    + volatilization + water and particle outflows
    DWaterToSed = Dvalue(4) + Dvalue(6) 'sed-water diffusion + sed dep
    Dtotal(1) = DsystemLoss(1) + DWaterToSed
    DsystemLoss(2) = Dvalue(1) + Dvalue(2) ' sed burial + transform in sed
    DSedToWater = Dvalue(3) + Dvalue(4) 'sed resus + sed-water diff
    Dtotal(2) = DsystemLoss(2) + DSedToWater


End Sub
```