



QUAL2K:

**A Modeling Framework for Simulating River and Stream
Water Quality
(Version 2.11)**

Documentation



The Mystic River at Medford, MA

Steve Chapra, Greg Pelletier and Hua Tao

December 16, 2008

Chapra, S.C., Pelletier, G.J. and Tao, H. 2008. QUAL2K: A Modeling Framework for Simulating River and Stream Water Quality, Version 2.11: Documentation and Users Manual. Civil and Environmental Engineering Dept., Tufts University, Medford, MA., Steven.Chapra@tufts.edu

Disclaimer

The information in this document has been funded partly by the United States Environmental Protection Agency. It is currently being subjected to the Agency's peer and administrative review and has yet to be approved for publication as an EPA document. Mention of trade names or commercial products does not constitute endorsement or recommendation for use by the U.S. Environmental Protection Agency.

The QUAL2K model (Q2K) described in this manual must be used at the user's own risk. Neither the U.S. Environmental Protection Agency, Tufts University, the Washington Dept. of Ecology, nor the program authors can assume responsibility for model operation, output, interpretation or usage.

The creators of this program have used their best efforts in preparing this code. It is not absolutely guaranteed to be error free. The author/programmer makes no warranties, expressed or implied, including without limitation warranties of merchantability or fitness for any particular purpose. No liability is accepted in any event for any damages, including accidental or consequential damages, lost of profits, costs of lost data or programming materials, or otherwise in connection with or arising out of the use of this program.

1 INTRODUCTION

QUAL2K (or Q2K) is a river and stream water quality model that is intended to represent a modernized version of the QUAL2E (or Q2E) model (Brown and Barnwell 1987). Q2K is similar to Q2E in the following respects:

- One dimensional. The channel is well-mixed vertically and laterally.
- Branching. The system can consist of a mainstem river with branched tributaries.
- Steady state hydraulics. Non-uniform, steady flow is simulated.
- Diel heat budget. The heat budget and temperature are simulated as a function of meteorology on a diel time scale.
- Diel water-quality kinetics. All water quality variables are simulated on a diel time scale.
- Heat and mass inputs. Point and non-point loads and withdrawals are simulated.

The QUAL2K framework includes the following new elements:

- Software Environment and Interface. Q2K is implemented within the Microsoft Windows environment. Numerical computations are programmed in Fortran 90. Excel is used as the graphical user interface. All interface operations are programmed in the Microsoft Office macro language: Visual Basic for Applications (VBA).
- Model segmentation. Q2E segments the system into river reaches comprised of equally spaced elements. Q2K also divides the system into reaches and elements. However, in contrast to Q2E, the element size for Q2K can vary from reach to reach. In addition, multiple loadings and withdrawals can be input to any element.
- Carbonaceous BOD speciation. Q2K uses two forms of carbonaceous BOD to represent organic carbon. These forms are a slowly oxidizing form (slow CBOD) and a rapidly oxidizing form (fast CBOD).
- Anoxia. Q2K accommodates anoxia by reducing oxidation reactions to zero at low oxygen levels. In addition, denitrification is modeled as a first-order reaction that becomes pronounced at low oxygen concentrations.
- Sediment-water interactions. Sediment-water fluxes of dissolved oxygen and nutrients can be simulated internally rather than being prescribed. That is, oxygen (SOD) and nutrient fluxes are simulated as a function of settling particulate organic matter, reactions within the sediments, and the concentrations of soluble forms in the overlying waters.
- Bottom algae. The model explicitly simulates attached bottom algae. These algae have variable stoichiometry.
- Light extinction. Light extinction is calculated as a function of algae, detritus and inorganic solids.
- pH. Both alkalinity and total inorganic carbon are simulated. The river's pH is then computed based on these two quantities.
- Pathogens. A generic pathogen is simulated. Pathogen removal is determined as a function of temperature, light, and settling.
- Reach specific kinetic parameters. Q2K allows you to specify many of the kinetic parameters on a reach-specific basis.
- Weirs and waterfalls. The hydraulics of weirs as well as the effect of weirs and waterfalls on gas transfer are explicitly included.

2 GETTING STARTED

As presently configured, an Excel workbook serves as the interface for QUAL2K. That is, all input and output as well as model execution are implemented from within Excel. All interface functions are programmed in Excel's macro language: Visual Basic for Applications (VBA). All numerical calculations are implemented in Fortran 90 for speed of execution. The following material provides a step-by-step description of how the model can be set up on your computer and used to perform a simulation.

Step 1: Copy the file, Q2Kv2_11.zip, to a directory (e.g., C:\). When this file is unzipped, it will set up a subdirectory, Q2Kv2_11 which includes an Excel file (Q2KMasterv2_11.xls), and an executable file (Q2KFortran2_11.exe). The first is the Q2K interface that allows you to run Q2K and display its results. The second is the Fortran executable that actually performs the model computations. **These two files must always be in the same directory for the model to run properly.** Note that after you run the model, some assisting files will be automatically created by the Fortran executable file to exchange information with Excel.

NOTE: DO NOT DELETE THE .zip file. If for some reason, you modify Q2K in a way that makes it unusable, you can always use the zip file to reinstall the model.

Step 2: Create a subdirectory off of C:\Q2Kv2_11 called **DataFiles**.

Step 3: Open Excel and make sure that your macro security level is set to medium (Figure 1). This can be done using the menu commands: Tools → Macro → Security. Make certain that the Medium radio button is selected.

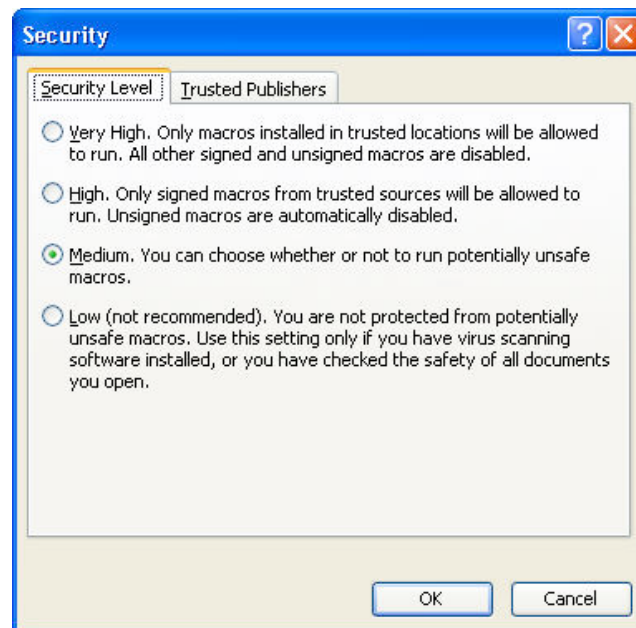


Figure 1 The Excel Macro Security Level dialogue box. In order to run Q2K, the Medium level of security should be selected.

Step 4: Open Q2KMasterFortranv2_11.xls. When you do this, the Macro Security Dialogue Box will be displayed (Figure 2).



Figure 2 The Excel Macro security dialogue box. In order to run Q2K, the **Enable Macros** button must be selected.

Click on the **Enable Macros** button.

Step 5: On the **QUAL2K Worksheet**, go to cell B10 and enter the path to the **DataFiles** directory: C:\QUAL2K\DataFiles as shown in Figure 3.


	A	B	C	D	E	F
1	QUAL2K FORTRAN					
2	Stream Water Quality Model					
3	Steve Chapra, Hua Tao and Greg Pelletier					
4	Version 2.11b8					
5						
6						
7	System ID:					
8	River name	Boulder Creek				Open Old File
9	Saved file name	BC092187v0211b8				
10	Directory where file saved	c:\Q2Kv2_11\DataFiles				Run Fortran
11	Month	9				
12	Day	21				
13	Year	1987				
14	Local time hours to UTC	-7				
15	Daylight savings time	Yes				
16	Calculation:					
17	Calculation step	0.1 hours				
18	Final time	30 day				
19	Solution method (integration)	Euler				
20	Solution method (pH)	Brent				
21	Time zone	Mountain Standard Time				
22	Program determined calc step	0.093750 hours				
23	Time of last calculation	0.17 minutes				
24	Time of sunrise	6:46 AM				
25	Time of solar noon	12:53 PM				
26	Time of sunset	6:59 PM				
27	Photoperiod	12.22 hours				

Figure 3 The QUAL2K Worksheet showing the entry of the file path into cell B10.

Step 6: Click on the **Run Fortran** button.

If the program does not work correctly...

There are two primary reasons why the program would not work properly. First, you may be using an old version of Microsoft Office. Although Excel is downwardly compatible for some earlier versions, Q2K will not work with very old versions.

Second, you may have made a mistake in implementing the preceding steps. A common mistake is to have mistyped the file path that you entered in cell B10. For example, suppose that you mistyped the path as C:\Q2KFortranv2_11\DataFiles. If this is the case, you will receive an error message (Figure 4).

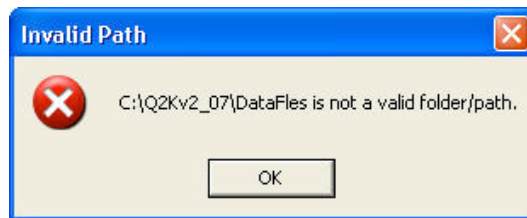


Figure 4 An error message that will occur if you type the incorrect file path into cell B10 on the QUAL2K Worksheet.

If this occurs, click OK. This will terminate the run and bring you back to the **QUAL2K Worksheet** where you can correct the file path entry.

If the program works correctly...

QUAL2K will begin to execute. A window will open showing the progress of the Fortran computations (Figure 5).

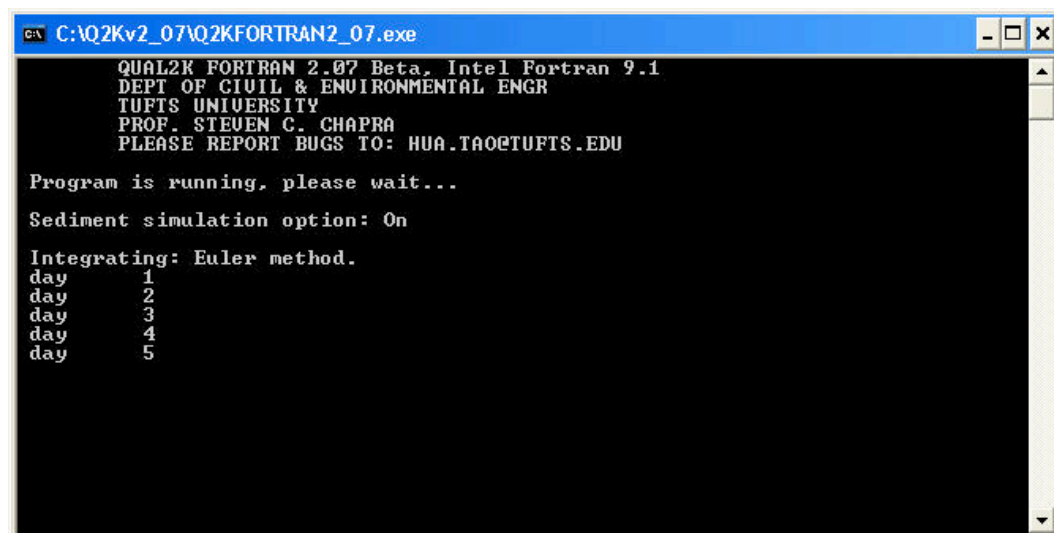
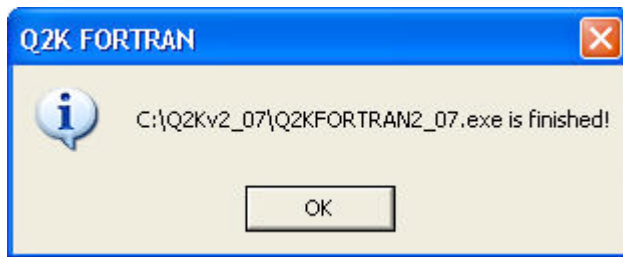
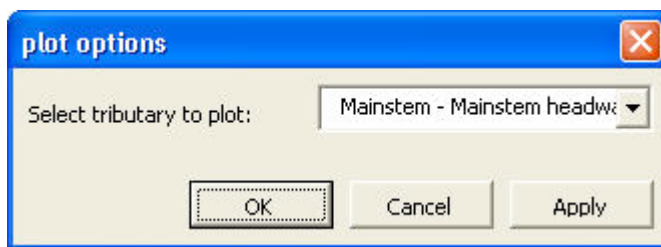


Figure 5 This window is displayed showing the progress of the model computations as executed in Fortran. It allows you to follow the progress of a model run.

The program is set up to simulate a fictitious river with a mainstem along with two tributaries. If the program works properly, the following dialogue box will appear when the run is completed:

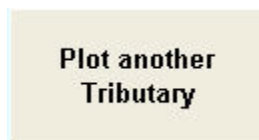


Press OK and the following dialogue box will be displayed:



This box allows you to choose the parts of the system that you want to plot. As shown, it defaults to the river's Mainstem. Press OK to see the travel time for the Mainstem. Note that all plots are updated when you press OK.

To switch to see the plots for one of the tributaries, you would press the button on the upper left of the screen



This causes the plot options dialogue box to be displayed. The pulldown can then be used to select another tributary.

Step 7: On the **QUAL2K Worksheet** click on the Open Old File button. Browse to get to the directory: C:\Q2KFortranv2_11\DataFiles. You should see that a new file has been created with the name that was specified in cell B9 (in the case of the example in Figure 3, BogusExample.q2k). Click on the Cancel button to return to Q2K.

Note that every time that Q2K is run, a data file will be created with the file name specified in cell B9 on the **QUAL2K Worksheet** (Figure 3). The program automatically affixes the extension .q2k to the file name. Since this will overwrite previous versions of the file, make certain to change the file name when you perform a new application.

Now that you have successfully run Q2K on your computer, the following pages are devoted to documenting the science that underlies the model.

3 SEGMENTATION AND HYDRAULICS

The model represents a river as a series of reaches. These represent stretches of river that have constant hydraulic characteristics (e.g., slope, bottom width, etc.). As depicted in Figure 6, the reaches are numbered in ascending order starting from the headwater of the river's main stem. Notice that both point and non-point sources and point and non-point withdrawals (abstractions) can be positioned anywhere along the channel's length.

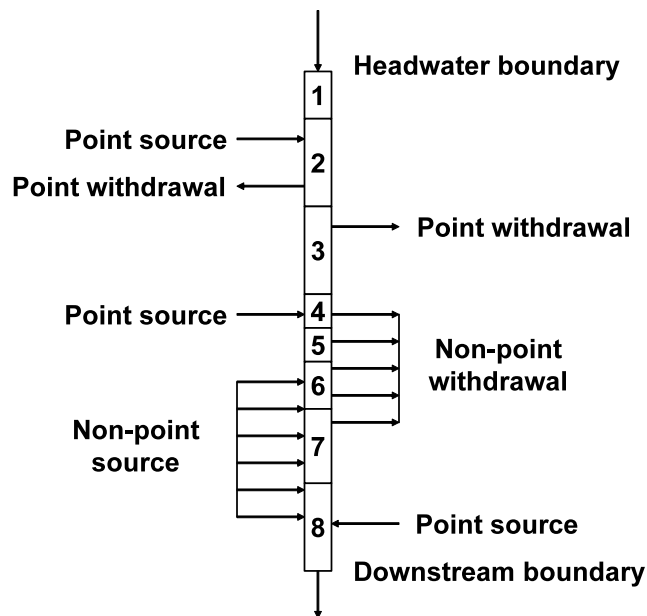


Figure 6 QUAL2K segmentation scheme for a river with no tributaries.

For systems with tributaries (Figure 7), the reaches are numbered in ascending order starting at reach 1 at the headwater of the main stem. When a junction with a tributary is reached, the numbering continues at that tributary's headwater. Observe that both the headwaters and the tributaries are also numbered consecutively following a sequencing scheme similar to the reaches. Note also that the major branches of the system (that is, the main stem and each of the tributaries) are referred to as segments. This distinction has practical importance because the software provides plots of model output on a segment basis. That is, the software generates individual plots for the main stem as well as each of the tributaries.

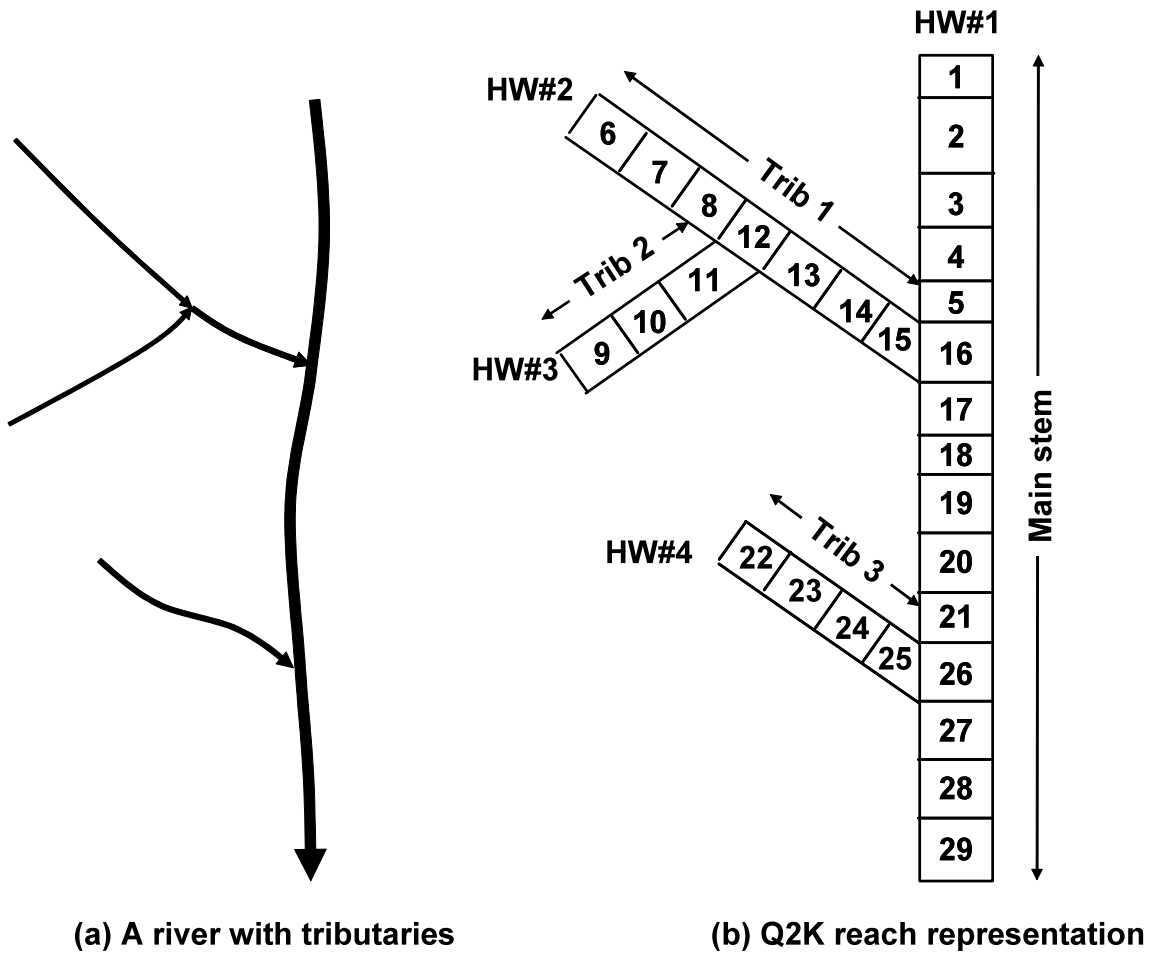


Figure 7 QUAL2K segmentation scheme for (a) a river with tributaries. The Q2K reach representation in (b) illustrates the reach, headwater and tributary numbering schemes.

Finally, any model reach can be further divided into a series of equally-spaced elements. As in Figure 8, this is done by merely specifying the number of elements that are desired.

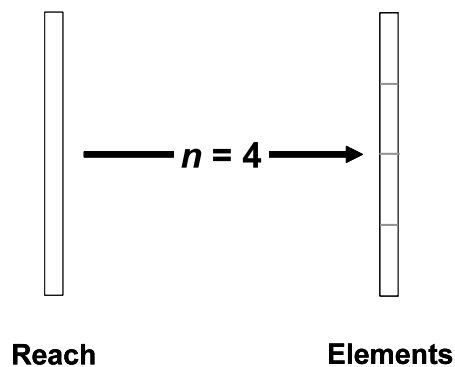


Figure 8 If desired, any model reach can be further subdivided into a series of n equal-length elements.

In summary, the nomenclature used to describe the way in which Q2K organizes river topology is as follows:

- Reach. A length of river with constant hydraulic characteristics.
- Element. The model's fundamental computational unit which consists of an equal length subdivision of a reach.
- Segment. A collection of reaches representing a branch of the system. These consist of the main stem as well as each tributary.
- Headwater. The upper boundary of a model segment.

3.1 Flow Balance

As described in the last section, Q2K's most fundamental unit is the element. A steady-state flow balance is implemented for each model element as (Figure 9)

$$Q_i = Q_{i-1} + Q_{in,i} - Q_{out,i} \quad (1)$$

where Q_i = outflow from element i into the downstream element $i + 1$ [m^3/d], Q_{i-1} = inflow from the upstream element $i - 1$ [m^3/d], $Q_{in,i}$ is the total inflow into the element from point and nonpoint sources [m^3/d], and $Q_{out,i}$ is the total outflow from the element due to point and nonpoint withdrawals [m^3/d]. Thus, the downstream outflow is simply the difference between inflow and source gains minus withdrawal losses.

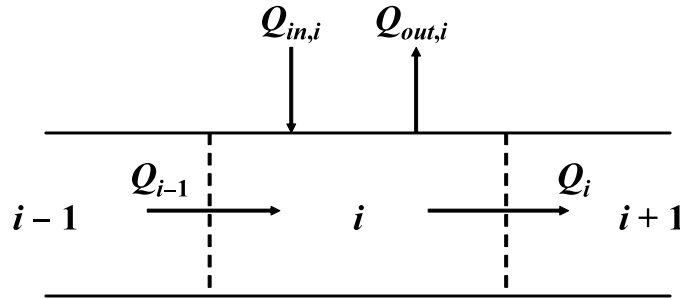


Figure 9 Element flow balance.

The total inflow from sources is computed as

$$Q_{in,i} = \sum_{j=1}^{psi} Q_{ps,i,j} + \sum_{j=1}^{npsi} Q_{nps,i,j} \quad (2)$$

where $Q_{ps,i,j}$ is the j th point source inflow to element i [m^3/d], psi = the total number of point sources to element i , $Q_{nps,i,j}$ is the j th non-point source inflow to element i [m^3/d], and $npsi$ = the total number of non-point source inflows to element i .

The total outflow from withdrawals is computed as

$$Q_{out,i} = \sum_{j=1}^{pai} Q_{pa,i,j} + \sum_{j=1}{npai} Q_{npa,i,j} \quad (3)$$

where $Q_{pa,i,j}$ is the j th point withdrawal outflow from element i [m^3/d], pai = the total number of point withdrawals from element i , $Q_{npa,i,j}$ is the j th non-point withdrawal outflow from element i [m^3/d], and $npai$ = the total number of non-point withdrawal flows from element i .

The non-point sources and withdrawals are modeled as line sources. As in Figure 10, the non-point source or withdrawal is demarcated by its starting and ending kilometer points. Its flow is then distributed to or from each element in a length-weighted fashion.

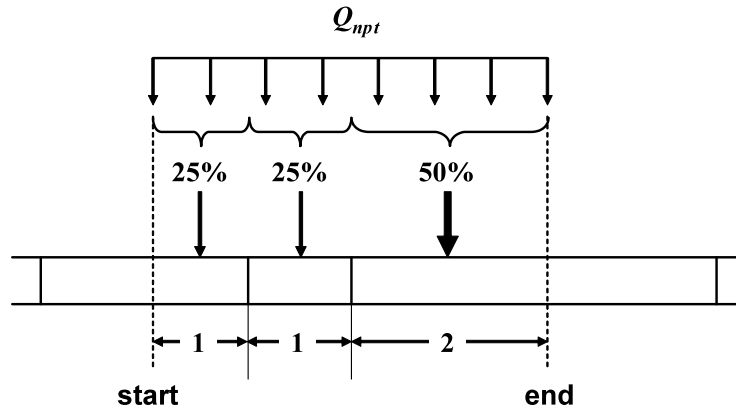


Figure 10 The manner in which non-point source flow is distributed to an element.

3.2 Hydraulic Characteristics

Once the outflow for each element is computed, the depth and velocity are calculated in one of three ways: weirs, rating curves, and Manning equations. The program decides among these options in the following manner:

- If weir height and width are entered, the weir option is implemented.
- If the weir height and width are zero and rating curve coefficients are entered (a and α), the rating curve option is implemented.
- If neither of the previous conditions is met, Q2K uses the Manning equation.

3.2.1 Weirs

Figure 11 shows how weirs are represented in Q2K. **Note that a weir can only occur at the end of a reach consisting of a single element.** The symbols shown in Figure 11 are defined as: H_i = the depth of the element upstream of the weir [m], H_{i+1} = the depth of the element downstream of the weir [m], $elev2_i$ = the elevation above sea level of the tail end of the upstream element [m], $elev1_{i+1}$ = the elevation above sea level of the head end of the downstream element [m], H_w = the height of the weir above $elev2_i$ [m], H_d = the drop between the elevation above sea level of the surface of element i and element $i+1$ [m], H_h = the head above the weir [m], B_w = the width of the weir [m]. Note that the width of the weir can differ from the width of the element, B_i .

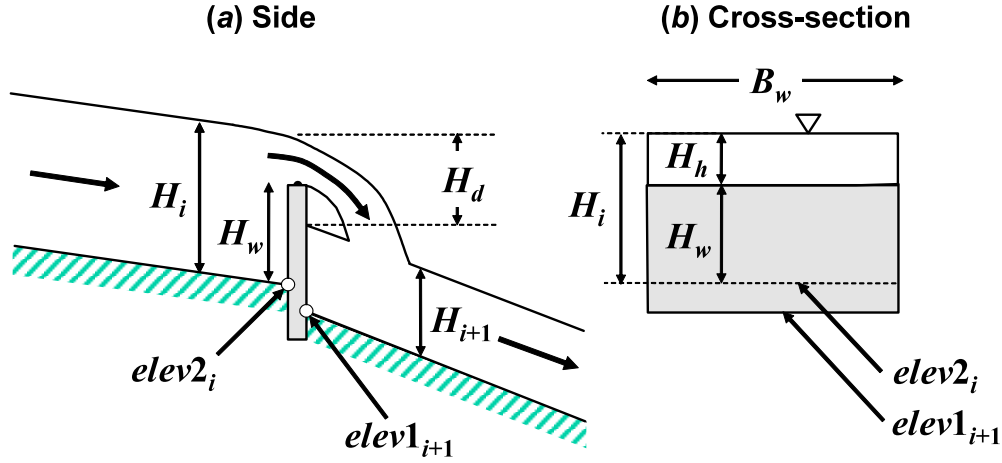


Figure 11 A sharp-crested weir occurring at the boundary between two reaches.

For a sharp-crested weir where $H_h/H_w < 0.4$, flow is related to head by (Finnemore and Franzini 2002)

$$Q_i = 1.83B_w H_h^{3/2} \quad (4)$$

where Q_i is the outflow from the element upstream of the weir in m^3/s , and B_w and H_h are in m. Equation (4) can be solved for

$$H_h = \left(\frac{Q_i}{1.83B_w} \right)^{2/3} \quad (5)$$

This result can then be used to compute the depth of element i ,

$$H_i = H_w + H_h \quad (6)$$

and the drop over the weir

$$H_d = elev2_i + H_i - elev1_{i+1} - H_{i+1} \quad (7)$$

Note that this drop is used to compute oxygen and carbon dioxide gas transfer due to the weir (see pages 57 and 63).

The cross-sectional area, velocity, surface area and volume of element i can then be computed as

$$A_{c,i} = B_i H_i \quad (8)$$

$$U_i = \frac{Q_i}{A_{c,i}} \quad (9)$$

$$A_{s,i} = B_i \Delta x_i$$

$$V_i = B_i H_i \Delta x_i$$

where B_i = the width of element i , Δx_i = the length of element i . Note that for reaches with weirs, the reach width must be entered. This value is entered in the column AA (labeled "Bottom Width") of the Reach Worksheet.

3.2.2 Rating Curves

Power equations (sometimes called Leopold-Maddox relationships) can be used to relate mean velocity and depth to flow for the elements in a reach,

$$U = aQ^b \quad (10)$$

$$H = \alpha Q^\beta \quad (11)$$

where a , b , α and β are empirical coefficients that are determined from velocity-discharge and stage-discharge rating curves, respectively. The values of velocity and depth can then be employed to determine the cross-sectional area and width by

$$A_c = \frac{Q}{U} \quad (12)$$

$$B = \frac{A_c}{H} \quad (13)$$

The surface area and volume of the element can then be computed as

$$A_s = B\Delta x$$

$$V = BH\Delta x$$

The exponents b and β typically take on values listed in Table 1. Note that the sum of b and β must be less than or equal to 1. If this is not the case, the width will decrease with increasing flow. If their sum equals 1, the channel is rectangular.

Table 1 Typical values for the exponents of rating curves used to determine velocity and depth from flow (Barnwell et al. 1989).

Equation	Exponent	Typical value	Range
$U = aQ^b$	b	0.43	0.4–0.6
$H = \alpha Q^\beta$	β	0.45	0.3–0.5

In some applications, you might want to specify constant values of depth and velocity that do not vary with flow. This can be done by setting the exponents b and β to zero and setting a equal to the desired velocity and α equal to the desired depth.

3.2.3 Manning Equation

Each element in a particular reach can be idealized as a trapezoidal channel (Figure 12). Under conditions of steady flow, the Manning equation can be used to express the relationship between flow and depth as

$$Q = \frac{S_0^{1/2}}{n} \frac{A_c^{5/3}}{P^{2/3}} \quad (14)$$

where Q = flow [m^3/s]¹, S_0 = bottom slope [m/m], n = the Manning roughness coefficient, A_c = the cross-sectional area [m^2], and P = the wetted perimeter [m].

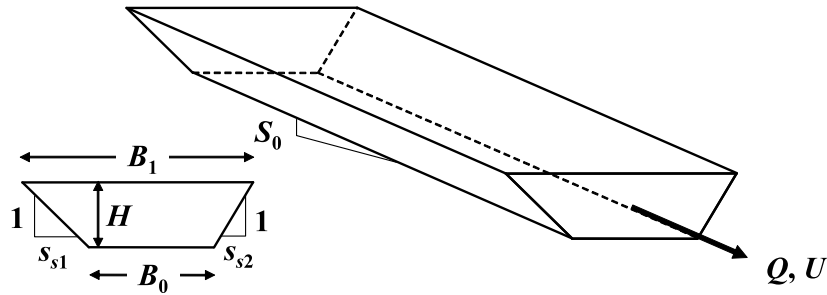


Figure 12 Trapezoidal channel.

The cross-sectional area of a trapezoidal channel is computed as

$$A_c = [B_0 + 0.5(s_{s1} + s_{s2})H]H \quad (15)$$

where B_0 = bottom width [m], s_{s1} and s_{s2} = the two side slopes as shown in Figure 12 [m/m], and H = element depth [m].

The wetted perimeter is computed as

$$P = B_0 + H\sqrt{s_{s1}^2 + 1} + H\sqrt{s_{s2}^2 + 1} \quad (16)$$

After substituting Eqs. (15) and (16), Eq. (14) can be solved iteratively for depth (Chapra and Canale 2006),

$$H_k = \frac{(Qn)^{3/5} \left(B_0 + H_{k-1}\sqrt{s_{s1}^2 + 1} + H_{k-1}\sqrt{s_{s2}^2 + 1} \right)^{2/5}}{S_0^{3/10} [B_0 + 0.5(s_{s1} + s_{s2})H_{k-1}]} \quad (17)$$

where $k = 1, 2, \dots, n$, where n = the number of iterations. An initial guess of $H_0 = 0$ is employed. The method is terminated when the estimated error falls below a specified value of 0.001%. The estimated error is calculated as

¹ Notice that time is measured in seconds in this and other formulas used to characterize hydraulics. This is how the computations are implemented within Q2K. However, once the hydraulic characteristics are determined they are converted to day units to be compatible with other computations.

$$\varepsilon_a = \left| \frac{H_{k+1} - H_k}{H_{k+1}} \right| \times 100\% \quad (18)$$

The cross-sectional area is determined with Eq. (15) and the velocity can then be determined from the continuity equation,

$$U = \frac{Q}{A_c} \quad (19)$$

The average element width, B [m], is computed as

$$B = \frac{A_c}{H} \quad (20)$$

The top width, B_1 [m], is computed as

$$B_1 = B_0 + (s_{s1} + s_{s2})H$$

The surface area and volume of the element can then be computed as

$$A_s = B_1 \Delta x$$

$$V = BH \Delta x$$

Suggested values for the Manning coefficient are listed in Table 2. Manning's n typically varies with flow and depth (Gordon et al. 1992). As the depth decreases at low flow, the relative roughness usually increases. Typical published values of Manning's n , which range from about 0.015 for smooth channels to about 0.15 for rough natural channels, are representative of conditions when the flow is at the bankfull capacity (Rosgen, 1996). Critical conditions of depth for evaluating water quality are generally much less than bankfull depth, and the relative roughness may be much higher.

Table 2 The Manning roughness coefficient for various open channel surfaces (from Chow et al. 1988).

MATERIAL	n
Man-made channels	
Concrete	0.012
Gravel bottom with sides:	
Concrete	0.020
mortared stone	0.023
Riprap	0.033
Natural stream channels	
Clean, straight	0.025-0.04
Clean, winding and some weeds	0.03-0.05
Weeds and pools, winding	0.05
Mountain streams with boulders	0.04-0.10
Heavy brush, timber	0.05-0.20

3.2.4 Waterfalls

In Section 3.2.1, the drop of water over a weir was computed. This value is needed in order to compute the enhanced reaeration that occurs in such cases. In addition to weirs, such drops can also occur at waterfalls (Figure 13). **Note that waterfalls can only occur at the end of a reach.**

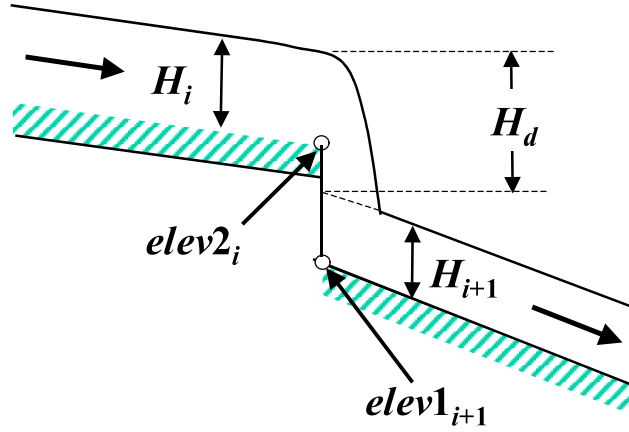


Figure 13 A waterfall occurring at the boundary between two reaches.

QUAL2K computes such drops for cases where the elevation above sea level drops abruptly at the boundary between two reaches. Equation (7) is used to compute the drop. It should be noted that the drop is only calculated when the elevation above sea level at the downstream end of a reach is greater than at the beginning of the next downstream reach; that is, $elev2_i > elev1_{i+1}$.

3.3 Travel Time

The residence time of each element is computed as

$$\tau_k = \frac{V_k}{Q_k} \quad (21)$$

where τ_k = the residence time of the k^{th} element [d], V_k = the volume of the k^{th} element [m^3] = $A_{c,k}\Delta x_k$, $A_{c,k}$ = the cross-sectional area of the k^{th} element [m^2], and Δx_k = the length of the k^{th} element [m]. These times are then accumulated to determine the travel time along each of the river's segments (that is, either the main stem or one of the tributaries). For example, the travel time from the headwater to the downstream end of the j th element in a segment is computed as,

$$t_{t,j} = \sum_{k=1}^j \tau_k \quad (22)$$

where $t_{t,j}$ = the travel time [d].

3.4 Longitudinal Dispersion

Two options are used to determine the longitudinal dispersion for a boundary between two elements. First, the user can simply enter estimated values on the **Reach Worksheet**. If the user

does not enter values, a formula is employed to internally compute dispersion based on the channel's hydraulics (Fischer et al. 1979),

$$E_{p,i} = 0.011 \frac{U_i^2 B_i^2}{H_i U_i^*} \quad (23)$$

where $E_{p,i}$ = the longitudinal dispersion between elements i and $i + 1$ [m^2/s], U_i = velocity [m/s], B_i = width [m], H_i = mean depth [m], and U_i^* = shear velocity [m/s], which is related to more fundamental characteristics by

$$U_i^* = \sqrt{g H_i S_i} \quad (24)$$

where g = acceleration due to gravity [$= 9.81 \text{ m/s}^2$] and S = channel slope [dimensionless].

After computing or prescribing $E_{p,i}$, the numerical dispersion is computed as

$$E_{n,i} = \frac{U_i \Delta x_i}{2} \quad (25)$$

The model dispersion E_i (i.e., the value used in the model calculations) is then computed as follows:

- If $E_{n,i} \leq E_{p,i}$, the model dispersion, E_i is set to $E_{p,i} - E_{n,i}$.
- If $E_{n,i} > E_{p,i}$, the model dispersion is set to zero.

For the latter case, the resulting dispersion will be greater than the physical dispersion. Thus, dispersive mixing will be higher than reality. It should be noted that for most steady-state rivers, the impact of this overestimation on concentration gradients will be negligible. If the discrepancy is significant, the only alternative is to make element lengths smaller so that the numerical dispersion becomes smaller than the physical dispersion.

4 TEMPERATURE MODEL

As in Figure 14, the heat balance takes into account heat transfers from adjacent elements, loads, withdrawals, the atmosphere, and the sediments. A heat balance can be written for element i as

$$\begin{aligned} \frac{dT_i}{dt} = & \frac{Q_{i-1}}{V_i} T_{i-1} - \frac{Q_i}{V_i} T_i - \frac{Q_{out,i}}{V_i} T_i + \frac{E'_{i-1}}{V_i} (T_{i-1} - T_i) + \frac{E'_i}{V_i} (T_{i+1} - T_i) \\ & + \frac{W_{h,i}}{\rho_w C_{pw} V_i} \left(\frac{\text{m}^3}{10^6 \text{ cm}^3} \right) + \frac{J_{a,i}}{\rho_w C_{pw} H_i} \left(\frac{\text{m}}{100 \text{ cm}} \right) + \frac{J_{s,i}}{\rho_w C_{pw} H_i} \left(\frac{\text{m}}{100 \text{ cm}} \right) \end{aligned} \quad (26)$$

where T_i = temperature in element i [$^{\circ}\text{C}$], t = time [d], E'_i = the bulk dispersion coefficient between elements i and $i + 1$ [m^3/d], $W_{h,i}$ = the net heat load from point and non-point sources into element i [cal/d], ρ_w = the density of water [g/cm^3], C_{pw} = the specific heat of water [cal/(g $^{\circ}\text{C}$)], $J_{a,i}$ = the air-water heat flux [cal/(cm^2 d)], and $J_{s,i}$ = the sediment-water heat flux [cal/(cm^2 d)].

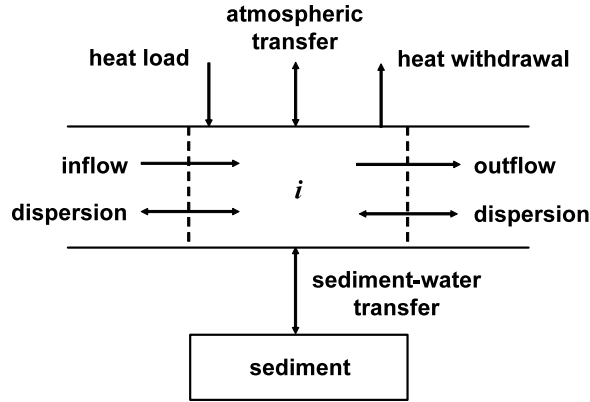


Figure 14 Heat balance for an element.

The bulk dispersion coefficient is computed as

$$E'_i = \frac{E_i A_{c,i}}{(\Delta x_i + \Delta x_{i+1})/2} \quad (27)$$

Note that two types of boundary condition are used at the river's downstream terminus: (1) a zero dispersion condition (natural boundary condition) and (2) a prescribed downstream boundary condition (Dirichlet boundary condition). The choice between these options is made on the **Downstream Worksheet**.

The net heat load from sources is computed as (recall Eq. 2)

$$W_{h,i} = \rho C_p \left[\sum_{j=1}^{psi} Q_{ps,i,j} T_{psi,j} + \sum_{j=1}^{npsi} Q_{nps,i,j} T_{npsi,j} \right] \quad (28)$$

where $T_{ps,i,j}$ is the temperature of the j th point source for element i [$^{\circ}\text{C}$], and $T_{nps,i,j}$ is the temperature of the j th non-point source temperature for element i [$^{\circ}\text{C}$].

4.1 Surface Heat Flux

As depicted in Figure 15, surface heat exchange is modeled as a combination of five processes:

$$J_h = I(0) + J_{an} - J_{br} - J_c - J_e \quad (29)$$

where $I(0)$ = net solar shortwave radiation at the water surface, J_{an} = net atmospheric longwave radiation, J_{br} = longwave back radiation from the water, J_c = conduction, and J_e = evaporation. All fluxes are expressed as $\text{cal}/\text{cm}^2/\text{d}$.

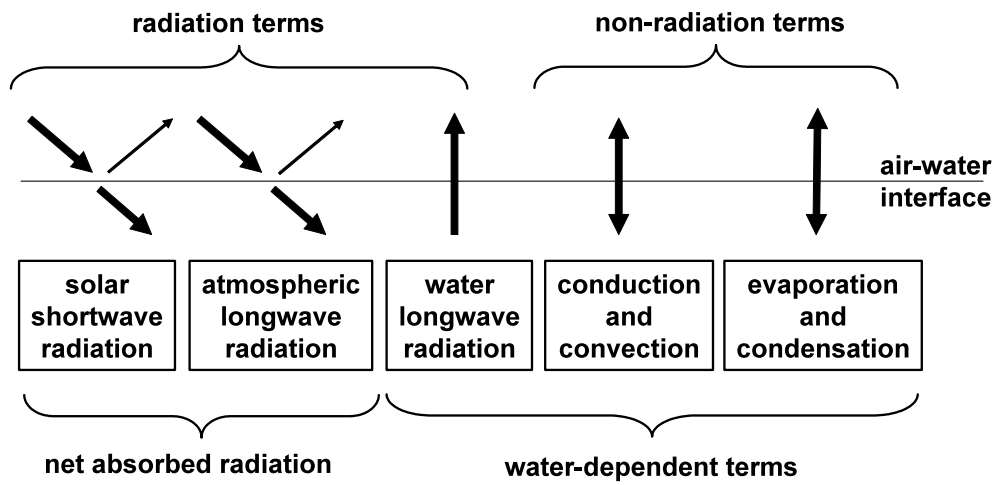


Figure 15 The components of surface heat exchange.

4.1.1 Solar Radiation

The model computes the amount of solar radiation entering the water at a particular latitude (L_{at}) and longitude (L_{lm}) on the earth's surface. This quantity is a function of the radiation at the top of the earth's atmosphere which is attenuated by atmospheric transmission, cloud cover, reflection, and shade,

$$I(0) = I_0 a_t a_c (1 - R_s) (1 - S_f) \quad (30)$$

extraterrestrial radiation atmospheric attenuation cloud attenuation reflection shading

where $I(0)$ = solar radiation at the water surface [$\text{cal}/\text{cm}^2/\text{d}$], I_0 = extraterrestrial radiation (i.e., at the top of the earth's atmosphere) [$\text{cal}/\text{cm}^2/\text{d}$], a_t = atmospheric attenuation, a_c = cloud attenuation, R_s = albedo (fraction reflected), and S_f = effective shade (fraction blocked by vegetation and topography).

Extraterrestrial radiation. The extraterrestrial radiation is computed as (TVA 1972)

$$I_0 = \frac{W_0}{r^2} \sin \alpha \quad (31)$$

where W_0 = the solar constant [1367 W/m² or 2823 cal/cm²/d], r = normalized radius of the earth's orbit (i.e., the ratio of actual earth-sun distance to mean earth-sun distance), and α = the sun's altitude [radians], which can be computed as

$$\sin \alpha = \sin \delta \sin L_{at} + \cos \delta \cos L_{at} \cos(\tau) \quad (32)$$

where δ = solar declination [radians], L_{at} = local latitude [radians], and τ = the local hour angle of the sun [radians].

The local hour angle in radians is given by

$$\tau = \left(\frac{trueSolarTime}{4} - 180 \right) \frac{\pi}{180} \quad (33)$$

where:

$$trueSolarTime = localTime + eqtime - 4 \times L_{lm} - 60 \times timezone \quad (34)$$

where *trueSolarTime* is the solar time determined from the actual position of the sun in the sky [minutes], *localTime* is the local time in minutes (local standard time), L_{lm} is the local longitude (positive decimal degrees for the western hemisphere), and *timezone* is the local time zone in hours relative to Greenwich Mean Time (e.g. -8 hours for Pacific Standard Time; the local time zone is selected on the **QUAL2K Worksheet**). The value of *eqtime* represents the difference between true solar time and mean solar time in minutes of time.

QUAL2K calculates the solar declination, hour angle, solar altitude, and normalized radius (distance between the earth and sun), as well as the times of sunrise and sunset using the Meeus (1999) algorithms as implemented by NOAA's Surface Radiation Research Branch (www.srrb.noaa.gov/highlights/sunrise/azel.html). The NOAA method for solar position that is used in QUAL2K also includes a correction for the effect of atmospheric refraction. The complete calculation method that is used to determine the solar position, sunrise, and sunset is presented in Appendix B.

The photoperiod f [hours] is computed as

$$f = t_{ss} - t_{sr} \quad (35)$$

where t_{ss} = time of sunset [hours] and t_{sr} = time of sunrise [hours].

Atmospheric attenuation. Various methods have been published to estimate the fraction of the atmospheric attenuation from a clear sky (a_t). Two alternative methods are available in QUAL2K to estimate a_t (Note that the solar radiation model is selected on the **Light and Heat Worksheet** of QUAL2K):

1) Bras (default)

The Bras (1990) method computes a_t as:

$$a_t = e^{-n_{fac} a_1 m} \quad (36)$$

where n_{fac} is an atmospheric turbidity factor that varies from approximately 2 for clear skies to 4 or 5 for smoggy urban areas. The molecular scattering coefficient (a_1) is calculated as

$$a_1 = 0.128 - 0.054 \log_{10} m \quad (37)$$

where m is the optical air mass, calculated as

$$m = \frac{1}{\sin \alpha + 0.15(\alpha_d + 3.885)^{-1.253}} \quad (38)$$

where α_d is the sun's altitude in degrees from the horizon $= \alpha \times (180^\circ/\pi)$.

2) Ryan and Stolzenbach

The Ryan and Stolzenbach (1972) model computes a_t from ground surface elevation and solar altitude as:

$$a_t = a_{tc} m \left(\frac{288 - 0.0065 \text{ elev}}{288} \right)^{5.256} \quad (39)$$

where a_{tc} is the atmospheric transmission coefficient (0.70-0.91, typically approximately 0.8), and $elev$ is the ground surface elevation in meters.

Direct measurements of solar radiation are available at some locations. For example, NOAA's Integrated Surface Irradiance Study (ISIS) has data from various stations across the United States (<http://www.atdd.noaa.gov/isis.htm>). The selection of either the Bras or Ryan-Stolzenbach solar radiation model and the appropriate atmospheric turbidity factor or atmospheric transmission coefficient for a particular application should ideally be guided by a comparison of predicted solar radiation with measured values at a reference location.

Cloud Attenuation. Attenuation of solar radiation due to cloud cover is computed with

$$a_c = 1 - 0.65 C_L^2 \quad (40)$$

where C_L = fraction of the sky covered with clouds.

Reflectivity. Reflectivity is calculated as

$$R_s = A \alpha_d^B \quad (41)$$

where A and B are coefficients related to cloud cover (Table 3).

Table 3 Coefficients used to calculate reflectivity based on cloud cover.

Cloudiness	Clear		Scattered		Broken		Overcast	
C_L	0		0.1-0.5		0.5-0.9		1.0	
Coefficients	A	B	A	B	A	B	A	B
	1.18	-0.77	2.20	-0.97	0.95	-0.75	0.35	-0.45

Shade. Shade is an input variable for the QUAL2K model. Shade is defined as the fraction of potential solar radiation that is blocked by topography and vegetation. An Excel/VBA program named ‘Shade.xls’ is available from the Washington Department of Ecology to estimate shade from topography and riparian vegetation (Ecology 2003). Input values of integrated hourly estimates of shade for each reach are entered on the **Shade Worksheet** of QUAL2K.

4.1.2 Atmospheric Long-wave Radiation

The downward flux of longwave radiation from the atmosphere is one of the largest terms in the surface heat balance. This flux can be calculated using the Stefan-Boltzmann law

$$J_{an} = \sigma (T_{air} + 273)^4 \varepsilon_{sky} (1 - R_L) \quad (42)$$

where σ = the Stefan-Boltzmann constant = 11.7×10^{-8} cal/(cm² d K⁴), T_{air} = air temperature [°C], ε_{sky} = effective emissivity of the atmosphere [dimensionless], and R_L = longwave reflection coefficient [dimensionless]. Emissivity is the ratio of the longwave radiation from an object compared with the radiation from a perfect emitter at the same temperature. The reflection coefficient is generally small and is assumed to equal 0.03.

The atmospheric longwave radiation model is selected on the **Light and Heat Worksheet** of QUAL2K. Three alternative methods are available for use in QUAL2K to represent the effective emissivity (ε_{sky}):

1) Brunt (default)

Brunt’s (1932) equation is an empirical model that has been commonly used in water-quality models (Thomann and Mueller 1987),

$$\varepsilon_{clear} = A_a + A_b \sqrt{e_{air}}$$

where A_a and A_b are empirical coefficients. Values of A_a have been reported to range from about 0.5 to 0.7 and values of A_b have been reported to range from about 0.031 to 0.076 mmHg^{-0.5} for a wide range of atmospheric conditions. QUAL2K uses a default mid-range value of $A_a = 0.6$ together with a value of $A_b = 0.031$ mmHg^{-0.5} if the Brunt method is selected on the **Light and Heat Worksheet**.

2) Brutsaert

The Brutsaert equation is physically-based instead of empirically derived and has been shown to yield satisfactory results over a wide range of atmospheric conditions of air temperature and humidity at intermediate latitudes for conditions above freezing (Brutsaert, 1982).

$$\varepsilon_{clear} = 1.24 \left(\frac{1.333224 e_{air}}{T_a} \right)^{1/7}$$

where e_{air} is the air vapor pressure [mm Hg], and T_a is the air temperature in °K. The factor of 1.333224 converts the vapor pressure from mm Hg to millibars. The air vapor pressure [in mm Hg] is computed as (Raudkivi 1979):

$$e_{air} = 4.596 e^{\frac{17.27 T_d}{237.3 + T_d}} \quad (43)$$

where T_d = the dew-point temperature [°C].

3) Koberg

Koberg (1964) reported that the A_a in Brunt's formula depends on both air temperature and the ratio of the incident solar radiation to the clear-sky radiation (R_{sc}). As in Figure 16, he presented a series of curves indicating that A_a increases with T_{air} and decreases with R_{sc} with A_b held constant at $0.0263 \text{ millibars}^{-0.5}$ (about $0.031 \text{ mmHg}^{-0.5}$).

The following polynomial is used in Q2K to provide a continuous approximation of Koberg's curves.

$$A_a = a_k T_{air}^2 + b_k T_{air} + c_k$$

where

$$a_k = -0.00076437 R_{sc}^3 + 0.00121134 R_{sc}^2 - 0.00073087 R_{sc} + 0.0001106$$

$$b_k = 0.12796842 R_{sc}^3 - 0.2204455 R_{sc}^2 + 0.13397992 R_{sc} - 0.02586655$$

$$c_k = -3.25272249 R_{sc}^3 + 5.65909609 R_{sc}^2 - 3.43402413 R_{sc} + 1.43052757$$

The fit of this polynomial to points sampled from Koberg's curves are depicted in Figure 16. Note that an upper limit of 0.735 is prescribed for A_a .

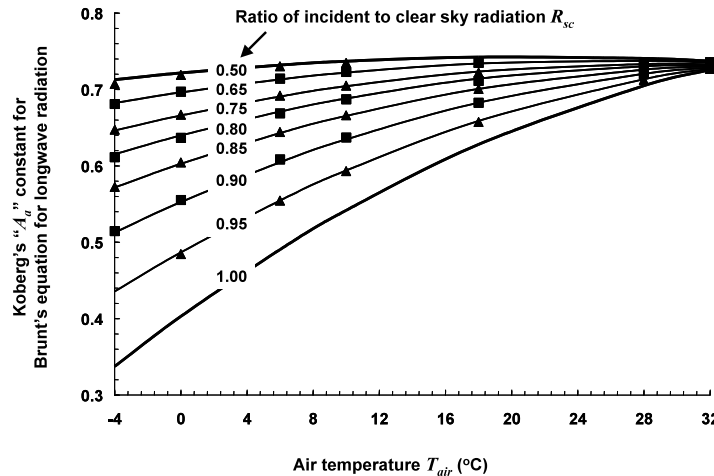


Figure 16 The points are sampled from Koberg's family of curves for determining the value of the A_a constant in Brunt's equation for atmospheric longwave radiation (Koberg, 1964). The lines are the functional representation used in Q2K.

For cloudy conditions the atmospheric emissivity may increase as a result of increased water vapor content. High cirrus clouds may have a negligible effect on atmospheric emissivity, but lower stratus and cumulus clouds may have a significant effect. The Koberg method accounts for the effect of clouds on the emissivity of longwave radiation in the determination of the A_a coefficient. The Brunt and Brutsaert methods determine emissivity of a clear sky and do not account for the effect of clouds. Therefore, if the Brunt or Brutsaert methods are selected, then the effective atmospheric emissivity for cloudy skies (ε_{sky}) is estimated from the clear sky emissivity by using a nonlinear function of the fractional cloud cover (C_L) of the form (TVA, 1972):

$$\varepsilon_{sky} = \varepsilon_{clear} (1 + 0.17 C_L^2) \quad (44)$$

The selection of the longwave model for a particular application should ideally be guided by a comparison of predicted results with measured values at a reference location. However, direct measurements are rarely collected. The Brutsaert method is recommended to represent a wide range of atmospheric conditions.

4.1.3 Water Long-wave Radiation

The back radiation from the water surface is represented by the Stefan-Boltzmann law,

$$J_{br} = \varepsilon \sigma (T + 273)^4 \quad (45)$$

where ε = emissivity of water (= 0.97) and T = the water temperature [$^{\circ}\text{C}$].

4.1.4 Conduction and Convection

Conduction is the transfer of heat from molecule to molecule when matter of different temperatures are brought into contact. *Convection* is heat transfer that occurs due to mass movement of fluids. Both can occur at the air-water interface and can be described by,

$$J_c = c_1 f(U_w) (T_s - T_{air}) \quad (46)$$

where c_1 = Bowen's coefficient (= 0.47 mmHg/ $^{\circ}\text{C}$). The term, $f(U_w)$, defines the dependence of the transfer on wind velocity over the water surface where U_w is the wind speed measured a fixed distance above the water surface.

Many relationships exist to define the wind dependence. Bras (1990), Edinger et al. (1974), and Shanahan (1984) provide reviews of various methods. Some researchers have proposed that conduction/convection and evaporation are negligible in the absence of wind (e.g. Marciano and Harbeck, 1952), which is consistent with the assumption that only molecular processes contribute to the transfer of mass and heat without wind (Edinger et al. 1974). Others have shown that significant conduction/convection and evaporation can occur in the absence of wind (e.g. Brady Graves and Geyer 1969, Harbeck 1962, Ryan and Harleman 1971, Helfrich et al. 1982, and

Adams et al. 1987). This latter opinion has gained favor (Edinger et al. 1974), especially for waterbodies that exhibit water temperatures that are greater than the air temperature.

Brady, Graves, and Geyer (1969) pointed out that if the water surface temperature is warmer than the air temperature, then “the air adjacent to the water surface would tend to become both warmer and more moist than that above it, thereby (due to both of these factors) becoming less dense. The resulting vertical convective air currents ... might be expected to achieve much higher rates of heat and mass transfer from the water surface [even in the absence of wind] than would be possible by molecular diffusion alone” (Edinger et al. 1974). Water temperatures in natural waterbodies are frequently greater than the air temperature, especially at night.

Edinger et al. (1974) recommend that the relationship that was proposed by Brady, Graves and Geyer (1969) based on data from cooling ponds, could be representative of most environmental conditions. Shanahan (1984) recommends that the Lake Hefner equation (Marciano and Harbeck, 1952) is appropriate for natural waters in which the water temperature is less than the air temperature. Shanahan also recommends that the Ryan and Harleman (1971) equation as recalibrated by Helfrich et al. (1982) is best suited for waterbodies that experience water temperatures that are greater than the air temperature. Adams et al. (1987) revisited the Ryan and Harleman and Helfrich et al. models and proposed another recalibration using additional data for waterbodies that exhibit water temperatures that are greater than the air temperature.

Three options are available on the **Light and Heat Worksheet** in QUAL2K to calculate $f(U_w)$:

1) Brady, Graves, and Geyer (default)

$$f(U_w) = 19.0 + 0.95U_w^2$$

where U_w = wind speed at a height of 7 m [m/s].

2) Adams 1

Adams et al. (1987) updated the work of Ryan and Harleman (1971) and Helfrich et al. (1982) to derive an empirical model of the wind speed function for heated waters that accounts for the enhancement of convection currents when the virtual temperature difference between the water and air ($\Delta\theta_v$ in degrees F) is greater than zero. Two wind functions reported by Adams et al., also known as the East Mesa method, are implemented in QUAL2K (wind speed in these equations is at a height of 2m).

This formulation uses an empirical function to estimate the effect of convection currents caused by virtual temperature differences between water and air, and the Harbeck (1962) equation is used to represent the contribution to conduction/convection and evaporation that is not due to convection currents caused by high virtual water temperature.

$$f(U_w) = 0.271 \sqrt{(22.4\Delta\theta_v^{1/3})^2 + (24.2A_{acres,i}^{-0.05}U_{w,mph})^2}$$

where $U_{w,mph}$ is wind speed in mph and $A_{acres,i}$ is surface area of element i in acres. The constant 0.271 converts the original units of $\text{BTU ft}^{-2} \text{ day}^{-1} \text{ mmHg}^{-1}$ to $\text{cal cm}^{-2} \text{ day}^{-1} \text{ mmHg}^{-1}$.

3) Adams 2

This formulation uses an empirical function of virtual temperature differences with the Marciano and Harbeck (1952) equation for the contribution to conduction/convection and evaporation that is not due to the high virtual water temperature

$$f(U_w) = 0.271 \sqrt{(22.4 \Delta \theta_v^{1/3})^2 + (17 U_{w, mph})^2}$$

Virtual temperature is defined as the temperature of dry air that has the same density as air under the in situ conditions of humidity. The virtual temperature difference between the water and air ($\Delta \theta_v$ in °F) accounts for the buoyancy of the moist air above a heated water surface. The virtual temperature difference is estimated from water temperature ($T_{w,f}$ in °F), air temperature ($T_{air,f}$ in °F), vapor pressure of water and air (e_s and e_{air} in mmHg), and the atmospheric pressure (p_{atm} is estimated as standard atmospheric pressure of 760 mmHg in QUAL2K):

$$\Delta \theta_v = \left(\frac{T_{w,f} + 460}{1 + 0.378 e_s / p_{atm}} - 460 \right) - \left(\frac{T_{air,f} + 460}{1 + 0.378 e_{air} / p_{atm}} - 460 \right) \quad (47)$$

The height of wind speed measurements is also an important consideration for estimating conduction/convection and evaporation. QUAL2K internally adjusts the wind speed to the correct height for the wind function that is selected on the **Light and Heat Worksheet**. The input values for wind speed on the **Wind Speed Worksheet** in QUAL2K are assumed to be representative of conditions at a height of 7 meters above the water surface. To convert wind speed measurements ($U_{w,z}$ in m/s) taken at any height (z_w in meters) to the equivalent conditions at a height of $z = 7$ m for input to the **Wind Speed Worksheet** of QUAL2K, the exponential wind law equation may be used (TVA, 1972):

$$U_w = U_{wz} \left(\frac{z}{z_w} \right)^{0.15} \quad (48)$$

For example, if wind speed data were collected from a height of 2 m, then the wind speed at 7 m for input to the **Wind Speed Worksheet** of QUAL2K would be estimated by multiplying the measured wind speed by a factor of 1.2.

4.1.5 Evaporation and Condensation

The heat loss due to evaporation can be represented by Dalton's law,

$$J_e = f(U_w)(e_s - e_{air}) \quad (49)$$

where e_s = the saturation vapor pressure at the water surface [mmHg], and e_{air} = the air vapor pressure [mmHg]. The saturation vapor pressure is computed as

$$e_{air} = 4.596 e^{\frac{17.27T}{237.3+T}} \quad (50)$$

4.2 Sediment-Water Heat Transfer

A heat balance for bottom sediment underlying a water element i can be written as

$$\frac{dT_{s,i}}{dt} = - \frac{J_{s,i}}{\rho_s C_{ps} H_{sed,i}} \quad (51)$$

where $T_{s,i}$ = the temperature of the bottom sediment below element i [$^{\circ}\text{C}$], $J_{s,i}$ = the sediment-water heat flux [$\text{cal}/(\text{cm}^2 \text{ d})$], ρ_s = the density of the sediments [g/cm^3], C_{ps} = the specific heat of the sediments [$\text{cal}/(\text{g } ^{\circ}\text{C})$], and $H_{sed,i}$ = the effective thickness of the sediment layer [cm].

The flux from the sediments to the water can be computed as

$$J_{s,i} = \rho_s C_{ps} \frac{\alpha_s}{H_{sed,i} / 2} (T_{si} - T_i) \times \frac{86,400 \text{ s}}{\text{d}} \quad (52)$$

where α_s = the sediment thermal diffusivity [cm^2/s].

The thermal properties of some natural sediments along with its components are summarized in Table 4. Note that soft, gelatinous sediments found in the deposition zones of lakes are very porous and approach the values for water. Some very slow, impounded rivers may approach such a state. However, rivers tend to have coarser sediments with significant fractions of sands, gravels and stones. Upland streams can have bottoms that are dominated by boulders and rock substrates.

Table 4 Thermal properties for natural sediments and the materials that comprise natural sediments.

Table 4. Thermal properties of various materials

Type of material	thermal conductivity		thermal diffusivity		ρ	C_p	ρC_p	reference
	w/m/°C	cal/s/cm/°C	m ² /s	cm ² /s	g/cm ³	cal/(g °C)	cal/(cm ³ °C)	
<i>Sediment samples</i>								
Mud Flat	1.82	0.0044	4.80E-07	0.0048			0.906	(1)
Sand	2.50	0.0060	7.90E-07	0.0079			0.757	"
Mud Sand	1.80	0.0043	5.10E-07	0.0051			0.844	"
Mud	1.70	0.0041	4.50E-07	0.0045			0.903	"
Wet Sand	1.67	0.0040	7.00E-07	0.0070			0.570	(2)
Sand 23% saturation with water	1.82	0.0044	1.26E-06	0.0126			0.345	(3)
Wet Peat	0.36	0.0009	1.20E-07	0.0012			0.717	(2)
Rock	1.76	0.0042	1.18E-06	0.0118			0.357	(4)
Loam 75% saturation with water	1.78	0.0043	6.00E-07	0.0060			0.709	(3)
Lake, gelatinous sediments	0.46	0.0011	2.00E-07	0.0020			0.550	(5)
Concrete canal	1.55	0.0037	8.00E-07	0.0080	2.200	0.210	0.460	"
Average of sediment samples:	1.57	0.0037	6.45E-07	0.0064			0.647	
<i>Miscellaneous measurements:</i>								
Lake, shoreline	0.59	0.0014						(5)
Lake soft sediments			3.25E-07	0.0033				"
Lake, with sand			4.00E-07	0.0040				"
River, sand bed			7.70E-07	0.0077				"
<i>Component materials:</i>								
Water	0.59	0.0014	1.40E-07	0.0014	1.000	0.999	1.000	(6)
Clay	1.30	0.0031	9.80E-07	0.0098	1.490	0.210	0.310	"
Soil, dry	1.09	0.0026	3.70E-07	0.0037	1.500	0.465	0.700	"
Sand	0.59	0.0014	4.70E-07	0.0047	1.520	0.190	0.290	"
Soil, wet	1.80	0.0043	4.50E-07	0.0045	1.810	0.525	0.950	"
Granite	2.89	0.0069	1.27E-06	0.0127	2.700	0.202	0.540	"
Average of composite materials:	1.37	0.0033	6.13E-07	0.0061	1.670	0.432	0.632	

(1) Andrews and Rodvey (1980)

(2) Geiger (1965)

(3) Nakshabandi and Kohnke (1965)

(4) Chow (1964) and Carslaw and Jaeger (1959)

(5) Hutchinson 1957, Jobson 1977, and Likens and Johnson 1969

(6) Cengel, Grigull, Mills, Bejan, Kreith and Bohn

Inspection of the component properties of Table 4 suggests that the presence of solid material in stream sediments leads to a higher coefficient of thermal diffusivity than that for water or porous lake sediments. In Q2K, we suggest a default value of 0.005 cm²/s for this quantity.

In addition, specific heat tends to decrease with density. Thus, the product of these two quantities tends to be more constant than the multiplicands. Nevertheless, it appears that the presence of solid material in stream sediments leads to a lower product than that for water or gelatinous lake sediments. In Q2K, we suggest default values of $\rho_s = 1.6 \text{ g/cm}^3$ and $C_{ps} = 0.4 \text{ cal/(g °C)}$. This corresponds to a product of 0.64 cal/(cm³ °C) for this quantity. Finally, as derived in Appendix C, the sediment thickness is set by default to 10 cm in order to capture the effect of the sediments on the diel heat budget for the water.

5 CONSTITUENT MODEL

5.1 Constituents and General Mass Balance

The model constituents are listed in Table 5.

Table 5 Model state variables

Variable	Symbol	Units*
Conductivity	s	μmhos
Inorganic suspended solids	m_i	mgD/L
Dissolved oxygen	o	mgO_2/L
Slowly reacting CBOD	c_s	mgO_2/L
Fast reacting CBOD	c_f	mgO_2/L
Organic nitrogen	n_o	$\mu\text{gN/L}$
Ammonia nitrogen	n_a	$\mu\text{gN/L}$
Nitrate nitrogen	n_n	$\mu\text{gN/L}$
Organic phosphorus	p_o	$\mu\text{gP/L}$
Inorganic phosphorus	p_i	$\mu\text{gP/L}$
Phytoplankton	a_p	$\mu\text{gA/L}$
Phytoplankton nitrogen	IN_p	$\mu\text{gN/L}$
Phytoplankton phosphorus	IP_p	$\mu\text{gP/L}$
Detritus	m_o	mgD/L
Pathogen	X	$\text{cfu}/100 \text{ mL}$
Alkalinity	Alk	mgCaCO_3/L
Total inorganic carbon	c_T	mole/L
Bottom algae biomass	a_b	mgA/m^2
Bottom algae nitrogen	IN_b	mgN/m^2
Bottom algae phosphorus	IP_b	mgP/m^2
Constituent i		
Constituent ii		
Constituent iii		

* $\text{mg/L} = \text{g}/\text{m}^3$; In addition, the terms D, C, N, P, and A refer to dry weight, carbon, nitrogen, phosphorus, and chlorophyll a , respectively. The term cfu stands for colony forming unit which is a measure of viable bacterial numbers.

For all but the bottom algae variables, a general mass balance for a constituent in an element is written as (Figure 17)

$$\frac{dc_i}{dt} = \frac{Q_{i-1}}{V_i} c_{i-1} - \frac{Q_i}{V_i} c_i - \frac{Q_{out,i}}{V_i} c_i + \frac{E'_{i-1}}{V_i} (c_{i-1} - c_i) + \frac{E'_i}{V_i} (c_{i+1} - c_i) + \frac{W_i}{V_i} + S_i \quad (53)$$

where W_i = the external loading of the constituent to element i [g/d or mg/d], and S_i = sources and sinks of the constituent due to reactions and mass transfer mechanisms [$\text{g}/\text{m}^3/\text{d}$ or $\text{mg}/\text{m}^3/\text{d}$].

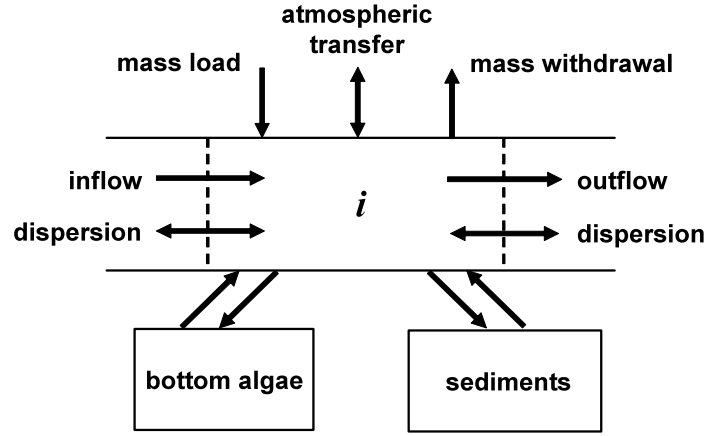


Figure 17 Mass balance.

The external load is computed as (recall Eq. 2),

$$W_i = \sum_{j=1}^{psi} Q_{ps,i,j} c_{psi,j} + \sum_{j=1}^{npsi} Q_{nps,i,j} c_{npsi,j} \quad (54)$$

where $c_{ps,i,j}$ is the j th point source concentration for element i [mg/L or $\mu\text{g/L}$], and $c_{nps,i,j}$ is the j th non-point source concentration for element i [mg/L or $\mu\text{g/L}$].

For bottom algae, the transport and loading terms are omitted,

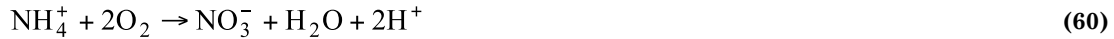
$$\frac{da_{b,i}}{dt} = S_{b,i} \quad (55)$$

$$\frac{dIN_b}{dt} = S_{bN,i} \quad (56)$$

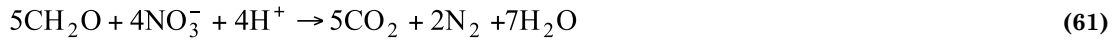
$$\frac{dIP_b}{dt} = S_{bP,i} \quad (57)$$

where $S_{b,i}$ = sources and sinks of bottom algae biomass due to reactions [mgA/m²/d], $S_{bN,i}$ = sources and sinks of bottom algae nitrogen due to reactions [mgN/m²/d], and $S_{bP,i}$ = sources and sinks of bottom algae phosphorus due to reactions [mgP/m²/d].

The sources and sinks for the state variables are depicted in Figure 18 (note that the internal levels of nitrogen and phosphorus in the bottom algae are not depicted). The mathematical representations of these processes are presented in the following sections.



Denitrification:



Note that a number of additional reactions are used in the model such as those involved with simulating pH and unionized ammonia. These will be outlined when these topics are discussed later in this document.

5.2.2 Stoichiometry of Organic Matter

The model requires that the stoichiometry of organic matter (i.e., phytoplankton and detritus) be specified by the user. The following representation is suggested as a first approximation (Redfield et al. 1963, Chapra 1997),

$$100 \text{ gD} : 40 \text{ gC} : 7200 \text{ mgN} : 1000 \text{ mgP} : 1000 \text{ mgA} \quad (62)$$

where gX = mass of element X [g] and mgY = mass of element Y [mg]. The terms D, C, N, P, and A refer to dry weight, carbon, nitrogen, phosphorus, and chlorophyll *a*, respectively. It should be noted that chlorophyll *a* is the most variable of these quantities with a range of approximately 500-2000 mgA (Laws and Chalup 1990, Chapra 1997).

These values are then combined to determine stoichiometric ratios as in

$$r_{xy} = \frac{\text{gX}}{\text{gY}} \quad (63)$$

For example, the amount of detritus (in grams dry weight or gD) that is released due to the death of a unit amount of phytoplankton (in milligrams of chlorophyll *a* or mgA) can be computed as

$$r_{da} = \frac{100 \text{ gD}}{1000 \text{ mgA}} = 0.1 \frac{\text{gD}}{\text{mgA}}$$

5.2.2.1 Oxygen Generation and Consumption

The model requires that the rates of oxygen generation and consumption be prescribed. If ammonia is the substrate, the following ratio (based on Eq. 58) can be used to determine the grams of oxygen generated for each gram of plant matter that is produced through photosynthesis.

$$r_{oca} = \frac{106 \text{ moleO}_2 (32 \text{ gO}_2/\text{moleO}_2)}{106 \text{ moleC} (12 \text{ gC}/\text{moleC})} = 2.67 \frac{\text{gO}_2}{\text{gC}} \quad (64)$$

If nitrate is the substrate, the following ratio (based on Eq. 59) applies

$$r_{ocn} = \frac{138 \text{ moleO}_2 (32 \text{ gO}_2/\text{moleO}_2)}{106 \text{ moleC} (12 \text{ gC}/\text{moleC})} = 3.47 \frac{\text{gO}_2}{\text{gC}} \quad (65)$$

Note that Eq. (58) is also used for the stoichiometry of the amount of oxygen consumed for plant respiration.

For nitrification, the following ratio is based on Eq. (60)

$$r_{on} = \frac{2 \text{ moleO}_2 (32 \text{ gO}_2/\text{moleO}_2)}{1 \text{ moleN} (14 \text{ gN}/\text{moleN})} = 4.57 \frac{\text{gO}_2}{\text{gN}} \quad (66)$$

5.2.2.2 CBOD Utilization Due to Denitrification

As represented by Eq. (61), CBOD is utilized during denitrification,

$$r_{ondn} = 2.67 \frac{\text{gO}_2}{\text{gC}} \frac{5 \text{ moleC} \times 12 \text{ gC}/\text{moleC}}{4 \text{ moleN} \times 14 \text{ gN}/\text{moleN}} \times \frac{1 \text{ gN}}{1000 \text{ mgN}} = 0.00286 \frac{\text{gO}_2}{\text{mgN}} \quad (67)$$

5.2.3 Temperature Effects on Reactions

The temperature effect for all first-order reactions used in the model is represented by

$$k(T) = k(20)\theta^{T-20} \quad (68)$$

where $k(T)$ = the reaction rate [1/d] at temperature T [°C] and θ = the temperature coefficient for the reaction.

5.3 Composite Variables

In addition to the model's state variables, Q2K also displays several composite variables that are computed as follows:

Total Organic Carbon (mgC/L):

$$TOC = \frac{c_s + c_f}{r_{oc}} + r_{ca}a_p + r_{cd}m_o \quad (69)$$

Total Nitrogen (µgN/L):

$$TN = n_o + n_a + n_n + IN_p \quad (70)$$

Total Phosphorus (µgP/L):

$$TP = p_o + p_i + IP_p \quad (71)$$

Total Kjeldahl Nitrogen (µgN/L):

$$TKN = n_o + n_a + IN_p \quad (72)$$

Total Suspended Solids (mgD/L):

$$TSS = r_{da}a_p + m_o + m_i \quad (73)$$

Ultimate Carbonaceous BOD (mgO₂/L):

$$CBOD_u = c_s + c_f + r_{oc}r_{ca}a_p + r_{oc}r_{cd}m_o \quad (74)$$

5.4 Relationship of Model Variables and Data

For all but slow and fast CBOD (c_f and c_s), there exists a relatively straightforward relationship between the model state variables and standard water-quality measurements. These are outlined next. Then we discuss issues related to the more difficult problem of measuring CBOD.

5.4.1 Non-CBOD Variables and Data

The following are measurements that are needed for comparison of non-BOD variables with model output:

TEMP = temperature (°C)
 TKN = total kjeldahl nitrogen (µgN/L) or TN = total nitrogen (µgN/L)
 NH4 = ammonium nitrogen (µgN/L)
 NO2 = nitrite nitrogen (µgN/L)
 NO3 = nitrate nitrogen (µgN/L)
 CHLA = chlorophyll *a* (µgA/L)
 TP = total phosphorus (µgP/L)
 SRP = soluble reactive phosphorus (µgP/L)
 TSS = total suspended solids (mgD/L)
 VSS = volatile suspended solids (mgD/L)
 TOC = total organic carbon (mgC/L)
 DOC = dissolved organic carbon (mgC/L)
 DO = dissolved oxygen (mgO₂/L)
 PH = pH
 ALK = alkalinity (mgCaCO₃/L)
 COND = specific conductance (µmhos)

The model state variables can then be related to these measurements as follows:

$$s = \text{COND}$$

$$m_i = \text{TSS} - \text{VSS} \text{ or } \text{TSS} - r_{dc}(\text{TOC} - \text{DOC})$$

$$o = \text{DO}$$

$$n_o = \text{TKN} - \text{NH4} - r_{na} \text{CHLA} \quad \text{or} \quad n_o = \text{TN} - \text{NO2} - \text{NO3} - \text{NH4} - r_{na} \text{CHLA}$$

$$n_a = \text{NH4}$$

$$n_n = \text{NO2} + \text{NO3}$$

$$p_o = TP - SRP - r_{pa} \text{ CHLA}$$

$$p_i = SRP$$

$$a_p = \text{CHLA}$$

$$m_o = VSS - r_{da} \text{ CHLA or } r_{dc} (\text{TOC} - \text{DOC}) - r_{da} \text{ CHLA}$$

$$pH = \text{PH}$$

$$Alk = \text{ALK}$$

5.4.2 Carbonaceous BOD

The interpretation of BOD measurements in natural waters is complicated by three primary factors:

- Filtered versus unfiltered. If the sample is unfiltered, the BOD will reflect oxidation of both dissolved and particulate organic carbon. Since Q2K distinguishes between dissolved (c_s and c_f) and particulate (m_o and a_p) organics, an unfiltered measurement alone does not provide an adequate basis for distinguishing these individual forms. In addition, one component of the particulate BOD, phytoplankton (a_p) can further complicate the test through photosynthetic oxygen generation.
- Nitrogenous BOD. Along with the oxidation of organic carbon (CBOD), nitrification also contributes to oxygen depletion (NBOD). Thus, if the sample (a) contains reduced nitrogen and (b) nitrification is not inhibited, the measurement includes both types of BOD.
- Incubation time. Short-term, usually 5-day, BODs are typically performed. Because Q2K uses ultimate CBOD, 5-day BODs must be converted to ultimate BODs in a sensible fashion.

We suggest the following as practical ways to measure CBOD in a manner that accounts for the above factors and results in measurements that are compatible with Q2K.

Filtration. The sample should be filtered prior to incubation in order to separate dissolved from particulate organic carbon.

Nitrification inhibition. Nitrification can be suppressed by adding a chemical inhibiting agent such as TCMP (2-chloro-6-(trichloro methyl) pyridine. The measurement then truly reflects CBOD. In the event that inhibition is not possible, the measured value can be corrected for nitrogen by subtracting the oxygen equivalents of the reduced nitrogen ($= r_{on} \times \text{TKN}$) in the sample. However, as with all such difference-based adjustments, this correction may exhibit substantial error.

Incubation time. The model is based on ultimate CBOD, so two approaches are possible: (1) use a sufficiently long period so that the ultimate value is measured, or (2) use a 5-day measurement and extrapolate the result to the ultimate. The latter method is often computed with the formula

$$CBODFNU = \frac{CBODFN5}{1 - e^{-k_1 5}} \quad (75)$$

where $CBODFNU^2$ = the ultimate dissolved carbonaceous BOD [mgO_2/L], $CBODFN5$ = the 5-day dissolved carbonaceous BOD [mgO_2/L], and k_1 = the CBOD decomposition rate in the bottle [$/d$].

It should be noted that, besides practical considerations of time and expense, there may be other benefits from using the 5-day measurement with extrapolation, rather than performing the longer-term CBOD. Although extrapolation does introduce some error, the 5-day value has the advantage that it would tend to minimize possible nitrification effects which, even when inhibited, can begin to be exerted on longer time frames.

If all the above provisions are implemented, the result should correspond to the model variables by

$$c_f + c_s = CBODFNU$$

Slow versus Fast CBOD. The final question relates to discrimination between fast and slow CBODU. Although we believe that there is currently no single, simple, economically-feasible answer to this problem, we think that the following 2 strategies represent the best current alternatives.

Option 1: Represent all the dissolved, oxidizable organic carbon with a single pool (fast CBOD). The model includes parameters to bypass slow CBOD. If no slow CBOD inputs are entered, this effectively drops it from the model. For this case,

$$c_f = CBODFNU$$

$$c_s = 0$$

Option 2: Use an ultimate CBOD measurement for the fast fraction and compute slow CBOD by difference with a DOC measurement. For this case,

$$c_f = CBODFNU$$

$$c_s = r_{oc}DOC - CBODFNU$$

Option 2 works very nicely for systems where two distinct types of CBOD are present. For example, sewage effluent and autochthonous carbon from the aquatic food chain might be considered as fast CBOD. In contrast, industrial wastewaters such as pulp and paper mill effluent or allochthonous DOC from the watershed might be considered more recalcitrant and hence could be lumped into the slow CBOD fraction. In such case, the hydrolysis rate converting the slow into the fast fraction could be set to zero to make the two forms independent.

For both options, the $CBODFNU$ can either be (a) measured directly using a long incubation time or (b) computed by extrapolation with Eq. 75. In both situations, a time frame of several weeks to a month (i.e., a 20- to 30-day CBOD) is probably a valid period in order to oxidize most

² The nomenclature FNU stands for Filtration, Nitrification inhibition and Ultimate

of the readily degradable organic carbon. We base this assumption on the fact that bottle rates for sewage-derived organic carbon are on the order of 0.05 to 0.3/d (Chapra 1997). As in Figure 19, such rates suggest that much of the readily oxidizable CBOD will be exerted in about 20 to 30 days.

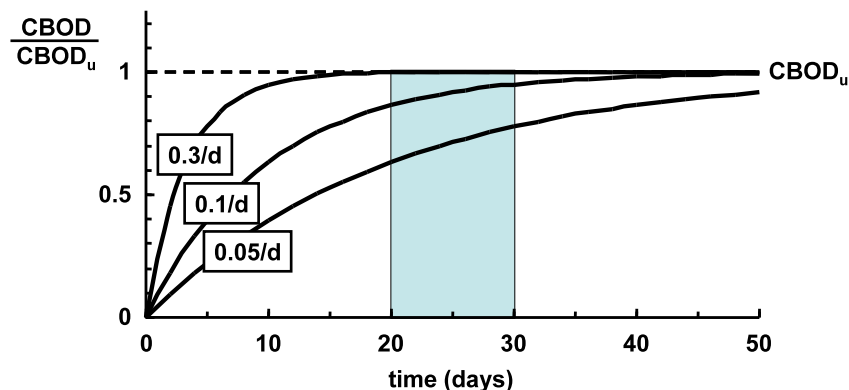


Figure 19 Progression of CBOD test for various levels of the bottle decomposition rate.

In addition, we believe that practitioners should consider conducting long-term CBOD tests at 30°C rather than at the commonly employed temperature of 20°C. The choice of 20°C originated from the fact that the average daily temperature of most receiving waters and wastewater treatment plants in the temperate zone in summer is approximately 20°C.

If the CBOD measurement is intended to be used for regulation or to assess treatment plant performance, it makes sense to standardize the test at a particular temperature. And for such purposes, 20°C is as reasonable a choice as any. However, if the intent is to measure an ultimate CBOD, anything that speeds up the process while not jeopardizing the measurement's integrity would seem beneficial.

The saprophytic bacteria that break down nonliving organic carbon in natural waters and sewage thrive best at temperatures from 20°C to 40°C. Thus, a temperature of 30°C is not high enough that the bacterial assemblage would shift to thermophilic organisms that are atypical of natural waters and sewage. The benefit should be higher oxidation rates which would result in shorter analysis times for CBOD measurements. Assuming that a $Q_{10} \approx 2$ is a valid approximation for bacterial decomposition, a 20-day BOD at 30°C should be equivalent to a 30-day BOD at 20°C.

5.5 Constituent Reactions

The mathematical relationships that describe the individual reactions and concentrations of the model state variables (Table 5) are presented in the following paragraphs.

5.5.1 Conservative Substance (s)

By definition, conservative substances are not subject to reactions:

$$S_s = 0 \quad (76)$$

5.5.2 Phytoplankton (a_p)

Phytoplankton increase due to photosynthesis. They are lost via respiration, death, and settling

$$S_{ap} = \text{PhytoPhoto} - \text{PhytoResp} - \text{PhytoDeath} - \text{PhytoSettl} \quad (77)$$

5.5.2.1 Photosynthesis

Phytoplankton photosynthesis is computed as

$$\text{PhytoPhoto} = \mu_p a_p \quad (78)$$

where μ_p = phytoplankton photosynthesis rate [/d] is a function of temperature, nutrients, and light,

$$\mu_p = k_{gp}(T) \phi_{Np} \phi_{Lp} \quad (79)$$

where $k_{gp}(T)$ = the maximum photosynthesis rate at temperature T [/d], ϕ_{Np} = phytoplankton nutrient attenuation factor [dimensionless number between 0 and 1], and ϕ_{Lp} = the phytoplankton light attenuation coefficient [dimensionless number between 0 and 1].

Nutrient Limitation. The nutrient limitation due to inorganic carbon is represented by a Michaelis-Menten. In contrast, for nitrogen and phosphorus, the photosynthesis rate depends on intracellular nutrient levels using a formulation originally developed by Droop (1974). The minimum value is then employed to compute the nutrient attenuation factor,

$$\phi_{Np} = \min \left[1 - \frac{q_{0Np}}{q_{Np}}, 1 - \frac{q_{0Pp}}{q_{Pp}}, \frac{[\text{H}_2\text{CO}_3^*] + [\text{HCO}_3^-]}{k_{sCp} + [\text{H}_2\text{CO}_3^*] + [\text{HCO}_3^-]} \right] \quad (80)$$

where q_{Np} and q_{Pp} = the phytoplankton cell quotas of nitrogen [mgN mgA⁻¹] and phosphorus [mgP mgA⁻¹], respectively, q_{0Np} and q_{0Pp} = the minimum phytoplankton cell quotas of nitrogen [mgN mgA⁻¹] and phosphorus [mgP mgA⁻¹], respectively, k_{sCp} = inorganic carbon half-saturation constant for phytoplankton [mole/L], $[\text{H}_2\text{CO}_3^*]$ = dissolved carbon dioxide concentration [mole/L], and $[\text{HCO}_3^-]$ = bicarbonate concentration [mole/L]. The minimum cell quotas are the levels of intracellular nutrient at which growth ceases. Note that the nutrient limitation terms cannot be negative. That is, if $q < q_0$, the limitation term is set to 0.

The cell quotas represent the ratios of the intracellular nutrient to the phytoplankton biomass,

$$q_{Np} = \frac{IN_p}{a_p} \quad (81)$$

$$q_{Pp} = \frac{IP_p}{a_p} \quad (82)$$

where IN_p = phytoplankton intracellular nitrogen concentration [$\mu\text{gN/L}$] and IP_p = phytoplankton intracellular phosphorus concentration [$\mu\text{gP/L}$].

Light Limitation. It is assumed that light attenuation through the water follows the Beer-Lambert law,

$$PAR(z) = PAR(0)e^{-k_e z} \quad (83)$$

where $PAR(z)$ = photosynthetically available radiation (PAR) at depth z below the water surface [ly/d]³, and k_e = the light extinction coefficient [m^{-1}]. The PAR at the water surface is assumed to be a fixed fraction of the solar radiation (Szeicz 1984, Baker and Frouin 1987):

$$PAR(0) = 0.47 I(0)$$

The extinction coefficient is related to model variables by

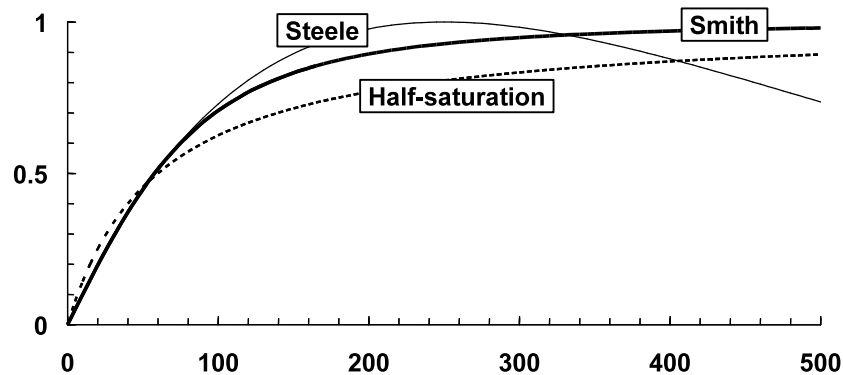
$$k_e = k_{eb} + \alpha_i m_i + \alpha_o m_o + \alpha_p a_p + \alpha_{pn} a_p^{2/3} \quad (84)$$

where k_{eb} = the background coefficient accounting for extinction due to water and color [m^{-1}], α_i , α_o , α_p , and α_{pn} , are constants accounting for the impacts of inorganic suspended solids [L/mgD/m], particulate organic matter [L/mgD/m], and chlorophyll [$\text{L}/\mu\text{gA/m}$ and $(\text{L}/\mu\text{gA})^{2/3}/\text{m}$], respectively. Suggested values for these coefficients are listed in Table 6.

Table 6 Suggested values for light extinction coefficients

Symbol	Value	Reference
α_i	0.052	Di Toro (1978)
α_o	0.174	Di Toro (1978)
α_p	0.0088	Riley (1956)
α_{pn}	0.054	Riley (1956)

Three models are used to characterize the impact of light on phytoplankton photosynthesis (Figure 20):



³ ly/d = langley per day. A langley is equal to a calorie per square centimeter. Note that a ly/d is related to the $\mu\text{E/m}^2/\text{d}$ by the following approximation: $1 \mu\text{E/m}^2/\text{s} \approx 0.45 \text{ Langley/day}$ (LIC-OR, Lincoln, NE).

Figure 20 The three models used for phytoplankton and bottom algae photosynthetic light dependence. The plot shows growth attenuation versus PAR intensity [ly/d].

Half-Saturation (Michaelis-Menten) Light Model (Baly 1935):

$$F_{Lp} = \frac{I(z)}{K_{Lp} + I(z)} \quad (85)$$

where F_{Lp} = phytoplankton growth attenuation due to light and K_{Lp} = the phytoplankton light parameter. In the case of the half-saturation model, the light parameter is a half-saturation coefficient [ly/d]. This function can be combined with the Beer-Lambert law and integrated over water depth, H [m], to yield the phytoplankton light attenuation coefficient

$$\phi_{Lp} = \frac{1}{k_e H} \ln \left(\frac{K_{Lp} + I(0)}{K_{Lp} + I(0)e^{-k_e H}} \right) \quad (86)$$

Smith's Function (Smith 1936):

$$F_{Lp} = \frac{I(z)}{\sqrt{K_{Lp}^2 + I(z)^2}} \quad (87)$$

where K_{Lp} = the Smith parameter for phytoplankton [ly/d]; that is, the PAR at which growth is 70.7% of the maximum. This function can be combined with the Beer-Lambert law and integrated over water depth to yield

$$\phi_{Lp} = \frac{1}{k_e H} \ln \left(\frac{I(0)/K_{Lp} + \sqrt{1 + (I(0)/K_{Lp})^2}}{(I(0)/K_{Lp})e^{-k_e H} + \sqrt{1 + ((I(0)/K_{Lp})e^{-k_e H})^2}} \right) \quad (88)$$

Steele's Equation (Steele 1962):

$$F_{Lp} = \frac{I(z)}{K_{Lp}} e^{1 - \frac{I(z)}{K_{Lp}}} \quad (89)$$

where K_{Lp} = the PAR at which phytoplankton growth is optimal [ly/d]. This function can be combined with the Beer-Lambert law and integrated over water depth to yield

$$\phi_{Lp} = \frac{2.718282}{k_e H} \left(e^{-\frac{I(0)}{K_{Lp}} e^{-k_e H}} - e^{-\frac{I(0)}{K_{Lp}}} \right) \quad (90)$$

5.5.2.2 Losses

Respiration. Phytoplankton respiration is represented as a first-order rate that is attenuated at low oxygen concentration,

$$\text{PhytoResp} = F_{\text{exp}} k_{rp}(T) a_p \quad (91)$$

where $k_{rp}(T)$ = temperature-dependent phytoplankton respiration/excretion rate [/d] and F_{exp} = attenuation due to low oxygen [dimensionless]. Oxygen attenuation is modeled by Eqs. (127) to (129) with the oxygen dependency represented by the parameter K_{sop} .

Death. Phytoplankton death is represented as a first-order rate,

$$\text{PhytoDeath} = k_{dp}(T) a_p \quad (92)$$

where $k_{dp}(T)$ = temperature-dependent phytoplankton death rate [/d].

Settling. Phytoplankton settling is represented as

$$\text{PhytoSettl} = \frac{v_a}{H} a_p \quad (93)$$

where v_a = phytoplankton settling velocity [m/d].

5.5.3 Phytoplankton Internal Nitrogen (IN_b)

The change in intracellular nitrogen in phytoplankton cells is calculated from

$$S_{pN} = \text{PhytoUpN} - q_{Np} \text{PhytoDeath} - \text{PhytoExN} \quad (94)$$

where PhytoUpN = the uptake rate of nitrogen by phytoplankton ($\mu\text{gN/L/d}$), PhytoDeath = phytoplankton death ($\mu\text{gN/L/d}$), and PhytoExN = the phytoplankton excretion of nitrogen ($\mu\text{gN/L/d}$), which is computed as

$$\text{PhytoExN} = q_{Np} k_{ep}(T) a_p \quad (95)$$

where $k_{ep}(T)$ = the temperature-dependent phytoplankton excretion rate [/d].

The N uptake rate depends on both external and intracellular nutrients as in (Rhee 1973),

$$\text{PhytoUpN} = \rho_{mNp} \frac{n_a + n_n}{k_{sNp} + n_a + n_n} \frac{K_{qNp}}{K_{qNp} + (q_{Np} - q_{0Np})} a_p \quad (96)$$

where ρ_{mNp} = the maximum uptake rate for nitrogen [mgN/mgA/d], k_{sNp} = half-saturation constant for external nitrogen [$\mu\text{gN/L}$] and K_{qNp} = half-saturation constant for intracellular nitrogen [mgN mgA^{-1}].

5.5.4 Phytoplankton Internal Phosphorus (IP_b)

The change in intracellular phosphorus in phytoplankton cells is calculated from

$$S_{pP} = \text{PhytoUpP} - q_{pP} \text{PhytoDeath} - \text{PhytoExP} \quad (97)$$

where PhytoUpP = the uptake rate of phosphorus by phytoplankton ($\mu\text{gP/L/d}$), PhytoDeath = phytoplankton death ($\mu\text{gP/L/d}$), and PhytoExP = the phytoplankton excretion of phosphorus ($\mu\text{gP/L/d}$), which is computed as

$$\text{PhytoExP} = q_{pP} k_{ep}(T) a_p \quad (98)$$

where $k_{ep}(T)$ = the temperature-dependent phytoplankton excretion rate [$/\text{d}$].

The P uptake rate depends on both external and intracellular nutrients as in (Rhee 1973),

$$\text{PhytoUpP} = \rho_{mPp} \frac{p_i}{k_{sPp} + p_i} \frac{K_{qPp}}{K_{qPp} + (q_{pP} - q_{0Pp})} a_p \quad (99)$$

where ρ_{mPp} = the maximum uptake rate for phosphorus [mgP/mgA/d], k_{sPp} = half-saturation constant for external phosphorus [$\mu\text{gP/L}$] and K_{qPp} = half-saturation constant for intracellular phosphorus [mgP mgA^{-1}].

5.5.5 Bottom algae (a_b)

Bottom algae increase due to photosynthesis. They are lost via respiration and death.

$$S_{ab} = \text{BotAlgPhoto} - \text{BotAlgResp} - \text{BotAlgDeath} \quad (100)$$

5.5.5.1 Photosynthesis

Two representations can be used to model bottom algae photosynthesis. The first is based on a temperature-corrected zero-order rate attenuated by nutrient and light limitation (McIntyre 1973, Rutherford et al. 1999),

$$\text{BotAlgPhoto} = C_{gb}(T) \phi_{Nb} \phi_{Lb} \quad (101)$$

where $C_{gb}(T)$ = the zero-order temperature-dependent maximum photosynthesis rate [$\text{mgA}/(\text{m}^2 \text{d})$], ϕ_{Nb} = bottom algae nutrient attenuation factor [dimensionless number between 0 and 1], and ϕ_{Lb} = the bottom algae light attenuation coefficient [dimensionless number between 0 and 1].

The second uses a first-order model,

$$\text{BotAlgPhoto} = C_{gb}(T) \phi_{Nb} \phi_{Lb} \phi_{Sb} a_b \quad (102)$$

where, for this case, $C_{gb}(T)$ = the first-order temperature-dependent maximum photosynthesis rate [d^{-1}], and ϕ_{Sb} = bottom algae space limitation attenuation factor.

Temperature Effect. As for the first-order rates, an Arrhenius model is employed to quantify the effect of temperature on bottom algae photosynthesis,

$$C_{gb}(T) = C_{gb}(20)\theta^{T-20} \quad (103)$$

Nutrient Limitation. The effect of nutrient limitation on bottom plant photosynthesis is modeled in the same way as for the phytoplankton. That is, a Droop (1974) formulation is used for nitrogen and phosphorus limitation whereas a Michaelis-Menten equation is employed for inorganic carbon,

$$\phi_{Nb} = \min \left[1 - \frac{q_{0Nb}}{q_{Nb}}, 1 - \frac{q_{0Pb}}{q_{Pb}}, \frac{[H_2CO_3^*] + [HCO_3^-]}{k_{sCb} + [H_2CO_3^*] + [HCO_3^-]} \right] \quad (104)$$

where q_{Nb} and q_{Pb} = the bottom algae cell quotas of nitrogen [mgN mgA⁻¹] and phosphorus [mgP mgA⁻¹], respectively, q_{0Nb} and q_{0Pb} = the bottom algae minimum cell quotas of nitrogen [mgN mgA⁻¹] and phosphorus [mgP mgA⁻¹], respectively, and k_{sCb} = the bottom algae inorganic carbon half-saturation constant [mole/L]. As was the case for phytoplankton, the nutrient limitation terms cannot be negative.

The cell quotas represent the ratios of the intracellular nutrient to the bottom plant's biomass,

$$q_{Nb} = \frac{IN_b}{a_b} \quad (105)$$

$$q_{Pb} = \frac{IP_b}{a_b} \quad (106)$$

where IN_b = intracellular nitrogen concentration [mgN/m²] and IP_b = intracellular phosphorus concentration [mgP/m²].

Light Limitation. In contrast to the phytoplankton, light limitation at any time is determined by the amount of PAR reaching the bottom of the water column. This quantity is computed with the Beer-Lambert law (recall Eq. 83) evaluated at the bottom of the river,

$$I(H) = I(0)e^{-k_e H} \quad (107)$$

As with the phytoplankton, three models (Eqs. 85, 87 and 89) are used to characterize the impact of light on bottom algae photosynthesis. Substituting Eq. (107) into these models yields the following formulas for the bottom algae light attenuation coefficient,

Half-Saturation Light Model (Baly 1935):

$$\phi_{Lb} = \frac{I(0)e^{-k_e H}}{K_{Lb} + I(0)e^{-k_e H}} \quad (108)$$

Smith's Function: (Smith 1936)

$$\phi_{Lp} = \frac{I(0)e^{-k_e H}}{\sqrt{K_{Lb}^2 + (I(0)e^{-k_e H})^2}} \quad (109)$$

Steele's Equation (Steele 1962):

$$\phi_{Lb} = \frac{I(0)e^{-k_e H}}{K_{Lb}} e^{1 + \frac{I(0)e^{-k_e H}}{K_{Lb}}} \quad (110)$$

where K_{Lb} = the appropriate bottom algae light parameter for each light model.

Space Limitation. If a first-order growth model is used, a term must be included to impose a space limitation on the bottom algae. A logistic model is used for this purpose as in

$$\phi_{Sb} = 1 - \frac{a_b}{a_{b,\max}}$$

where $a_{b,\max}$ = the carrying capacity [mgA/m²].

5.5.5.2 Losses

Respiration. Bottom algae respiration is represented as a first-order rate that is attenuated at low oxygen concentration,

$$\text{BotAlgResp} = F_{oxb} k_{rb}(T) a_b \quad (111)$$

where $k_{rb}(T)$ = temperature-dependent bottom algae respiration rate [/d] and F_{oxb} = attenuation due to low oxygen [dimensionless]. Oxygen attenuation is modeled by Eqs. (127) to (129) with the oxygen dependency represented by the parameter K_{sob} .

Death. Bottom algae death is represented as a first-order rate,

$$\text{BotAlgDeath} = k_{db}(T) a_b \quad (112)$$

where $k_{db}(T)$ = the temperature-dependent bottom algae death rate [/d].

5.5.6 Bottom Algae Internal Nitrogen (I/N_b)

The change in intracellular nitrogen in bottom algal cells is calculated from

$$S_{bN} = \text{BotAlgUpN} - q_{Nb} \text{BotAlgDeath} - \text{BotAlgExN} \quad (113)$$

where BotAlgUpN = the uptake rate of nitrogen by bottom algae (mgN/m²/d), BotAlgDeath = bottom algae death (mgA/m²/d), and BotAlgExN = the bottom algae excretion of nitrogen (mgN/m²/d), which is computed as

$$\text{BotAlgExN} = q_{Nb} k_{eb}(T) a_b \quad (114)$$

where $k_{eb}(T)$ = the temperature-dependent bottom algae excretion rate [/d].

The N uptake rate depends on both external and intracellular nutrients as in (Rhee 1973),

$$\text{BotAlgUpN} = \rho_{mNb} \frac{n_a + n_n}{k_{sNb} + n_a + n_n} \frac{K_{qNb}}{K_{qNb} + (q_{Nb} - q_{0Nb})} a_b \quad (115)$$

where ρ_{mNb} = the maximum uptake rate for nitrogen [mgN/mgA/d], k_{sNb} = half-saturation constant for external nitrogen [$\mu\text{gN/L}$] and K_{qNb} = half-saturation constant for intracellular nitrogen [mgN mgA⁻¹].

5.5.7 Bottom Algae Internal Phosphorus (IP_b)

The change in intracellular phosphorus in bottom algal cells is calculated from

$$S_{bP} = \text{BotAlgUpP} - q_{Pb} \text{BotAlgDeath} - \text{BotAlgExP} \quad (116)$$

where BotAlgUpP = the uptake rate of phosphorus by bottom algae (mgP/m²/d), BotAlgDeath = bottom algae death (mgA/m²/d), and BotAlgExP = the bottom algae excretion of phosphorus (mgP/m²/d), which is computed as

$$\text{BotAlgExP} = q_{Pb} k_{eb}(T) a_b \quad (117)$$

where $k_{eb}(T)$ = the temperature-dependent bottom algae excretion rate [/d].

The P uptake rate depends on both external and intracellular nutrients as in (Rhee 1973),

$$\text{BotAlgUpP} = \rho_{mPb} \frac{p_i}{k_{sPb} + p_i} \frac{K_{qPb}}{K_{qPb} + (q_{Pb} - q_{0Pb})} a_b \quad (118)$$

where ρ_{mPb} = the maximum uptake rate for phosphorus [mgP/mgA/d], k_{sPb} = half-saturation constant for external phosphorus [$\mu\text{gP/L}$] and K_{qPb} = half-saturation constant for intracellular phosphorus [mgP mgA⁻¹].

5.5.8 Detritus (m_o)

Detritus or particulate organic matter (POM) increases due to plant death. It is lost via dissolution and settling

$$S_{mo} = r_{da} \text{PhytoDeath} + r_{da} \frac{\text{BotAlgDeath}}{H} - \text{DetrDiss} - \text{DetrSettl} \quad (119)$$

where

$$\text{DetrDiss} = k_{dt}(T) m_o \quad (120)$$

where $k_{dt}(T)$ = the temperature-dependent detritus dissolution rate [/d] and

$$\text{DetrSettl} = \frac{v_{dt}}{H} m_o \quad (121)$$

where v_{dt} = detritus settling velocity [m/d].

5.5.9 Slowly Reacting CBOD (c_s)

Slowly reacting CBOD increases due to detritus dissolution. It is lost via hydrolysis and oxidation,

$$S_{cs} = (1 - F_f) r_{od} \text{ DetrDiss} - \text{SlowCHydr} - \text{SlowCOxid} \quad (122)$$

where F_f = the fraction of detrital dissolution that goes to fast reacting CBOD [dimensionless], and

$$\text{SlowCHydr} = k_{hc}(T) c_s \quad (123)$$

where $k_{hc}(T)$ = the temperature-dependent slow CBOD hydrolysis rate [/d], and

$$\text{SlowCOxid} = F_{oxc} k_{dcs}(T) c_s \quad (124)$$

where $k_{dcs}(T)$ = the temperature-dependent slow CBOD oxidation rate [/d] and F_{oxc} = attenuation due to low oxygen [dimensionless].

5.5.10 Fast Reacting CBOD (c_f)

Fast reacting CBOD is gained via the dissolution of detritus and the hydrolysis of slowly-reacting CBOD. It is lost via oxidation and denitrification.

$$S_{cf} = F_f r_{od} \text{ DetrDiss} + \text{SlowCHydr} - \text{FastCOxid} - r_{ondn} \text{ Denitr} \quad (125)$$

where

$$\text{FastCOxid} = F_{oxc} k_{dc}(T) c_f \quad (126)$$

where $k_{dc}(T)$ = the temperature-dependent fast CBOD oxidation rate [/d] and F_{oxc} = attenuation due to low oxygen [dimensionless]. The parameter r_{ondn} is the ratio of oxygen equivalents lost per nitrate nitrogen that is denitrified (Eq. 67). The term Denitr is the rate of denitrification [$\mu\text{gN/L/d}$]. It will be defined in Sec. 5.5.15 below.

Three formulations are used to represent the oxygen attenuation:

Half-Saturation:

$$F_{oxrp} = \frac{o}{K_{soef} + o} \quad (127)$$

where K_{soef} = half-saturation constant for the effect of oxygen on fast CBOD oxidation [mgO_2/L].

Exponential:

$$F_{oxrp} = (1 - e^{-K_{socf}o}) \quad (128)$$

where K_{socf} = exponential coefficient for the effect of oxygen on fast CBOD oxidation [L/mgO₂].

Second-Order Half Saturation:

$$F_{oxrp} = \frac{o^2}{K_{socf} + o^2} \quad (129)$$

where K_{socf} = half-saturation constant for second-order effect of oxygen on fast CBOD oxidation [mgO₂²/L²].

5.5.11 Organic Nitrogen (n_o)

Organic nitrogen increases due to plant death. It is lost via hydrolysis and settling.

$$S_{no} = f_{onp} q_{Np} \text{PhytoDeath} + f_{onb} q_{Nb} \frac{\text{BotAlgDeath}}{H} - \text{ONHydr} - \text{ONSettl} \quad (130)$$

where f_{onp} = the fraction of the phytoplankton internal nitrogen that is in organic form which is calculated as

$$\begin{aligned} f_{onp} &= \frac{r_{na}}{q_{Np}} & \text{if } q_{Np} > r_{na} \\ f_{onp} &= 1 & \text{if } q_{Np} \leq r_{na} \end{aligned} \quad (131)$$

The fraction of the bottom algae internal nitrogen that is in organic form, f_{onb} , is calculated in a similar fashion.

The rate of organic nitrogen hydrolysis is computed as

$$\text{ONHydr} = k_{hn}(T)n_o \quad (132)$$

where $k_{hn}(T)$ = the temperature-dependent organic nitrogen hydrolysis rate [/d]. Organic nitrogen settling is determined as

$$\text{ONSettl} = \frac{v_{on}}{H} n_o \quad (133)$$

where v_{on} = organic nitrogen settling velocity [m/d].

5.5.12 Ammonia Nitrogen (n_a)

Ammonia nitrogen increases due to organic nitrogen hydrolysis and plant death and excretion. It is lost via nitrification and plant photosynthesis:

$$\begin{aligned}
S_{na} = & \text{ONHydr} + (1 - f_{onp})q_{Np} \text{PhytoDeath} + (1 - f_{onb})q_{Nb} \frac{\text{BotAlgDeath}}{H} \\
& + \text{PhytoExN} + \frac{\text{BotAlgExN}}{H} - \text{Nitrif} - P_{ap} \text{PhytoUpN} - P_{ab} \frac{\text{BotAlgUpN}}{H} - \text{NH3GasLoss}
\end{aligned} \tag{134}$$

The ammonia nitrification rate is computed as

$$\text{Nitrif} = F_{oxna} k_n(T) n_a \tag{135}$$

where $k_n(T)$ = the temperature-dependent nitrification rate for ammonia nitrogen [1/d] and F_{oxna} = attenuation due to low oxygen [dimensionless]. Oxygen attenuation is modeled by Eqs. (127) to (129) with the oxygen dependency represented by the parameter K_{sona} .

The coefficients P_{ap} and P_{ab} are the preferences for ammonium as a nitrogen source for phytoplankton and bottom algae, respectively,

$$P_{ap} = \frac{n_a n_n}{(k_{hnxp} + n_a)(k_{hnxp} + n_n)} + \frac{n_a k_{hnxp}}{(n_a + n_n)(k_{hnxp} + n_n)} \tag{136}$$

$$P_{ab} = \frac{n_a n_n}{(k_{hnxb} + n_a)(k_{hnxb} + n_n)} + \frac{n_a k_{hnxb}}{(n_a + n_n)(k_{hnxb} + n_n)} \tag{137}$$

where k_{hnxp} = preference coefficient of phytoplankton for ammonium [mgN/m³] and k_{hnxb} = preference coefficient of bottom algae for ammonium [mgN/m³].

5.5.13 Unionized Ammonia

The model simulates total ammonia. In water, the total ammonia consists of two forms: ammonium ion, NH_4^+ , and unionized ammonia, NH_3 . At normal pH (6 to 8), most of the total ammonia will be in the ionic form. However at high pH, unionized ammonia predominates. The amount of unionized ammonia can be computed as

$$n_{au} = F_u n_a \tag{138}$$

where n_{au} = the concentration of unionized ammonia [μgN/L], and F_u = the fraction of the total ammonia that is in unionized form,

$$F_u = \frac{K_a}{10^{-\text{pH}} + K_a} \tag{139}$$

where K_a = the equilibrium coefficient for the ammonia dissociation reaction, which is related to temperature by

$$\text{p}K_a = 0.09018 + \frac{2729.92}{T_a} \tag{140}$$

where T_a is absolute temperature [K] and $pK_a = -\log_{10}(K_a)$. Note that the fraction of the total ammonia that is in ionized form, F_i , can be computed as $1 - F_u$ or

$$F_i = \frac{10^{-pH}}{10^{-pH} + K_a} \quad (141)$$

5.5.14 Ammonia Gas Transfer

The loss of ammonia via gas transfer is computed as

$$\text{NH3GasLoss} = \frac{v_{nh3}(T)}{H} (n_{aus}(T) - n_{au})$$

where $v_{nh3}(T)$ = the temperature-dependent ammonia gas-transfer coefficient [m/d], and $n_{as}(T)$ = the saturation concentration of ammonia [$\mu\text{gN/L}$] at temperature, T .

The transfer coefficient is calculated as

$$v_{nh3} = K_l \frac{H_e}{H_e + RT_a \left(\frac{K_l}{K_g} \right)}$$

where v_v = the mass-transfer coefficient (m/d), K_l and K_g = liquid and gas film exchange coefficients [m/d], respectively, R = the universal gas constant ($= 8.206 \times 10^{-5} \text{ atm m}^3/(\text{K mole})$), T_a = absolute temperature [K], and H_e = Henry's constant ($\text{atm m}^3/\text{mole}$).

The saturation concentration is calculated as

$$n_{aus}(T) = \frac{p_{nh3}}{H_e} \times CF$$

where p_{nh3} = the partial pressure of ammonia in the atmosphere (atm), and CF is a conversion factor ($\mu\text{gN/L}$ per $\text{moleNH}_3/\text{m}^3$). The partial pressure of ammonia ranges from 1-10 ppb in rural and moderately polluted areas to 10-100 ppb in heavily polluted areas (Holland 1978, Finlayson-Pitts and Pitts 1986). We will assume that a value of 2 ppb, which corresponds to $2 \times 10^{-9} \text{ atm}$, represents a typical value. The conversion factor is

$$CF = \frac{\text{m}^3}{1000\text{L}} \times \frac{14\text{gN}}{\text{moleNH}_3} \times \frac{10^6 \mu\text{gN}}{\text{gN}} = 14 \times 10^3 \frac{\mu\text{gN/L}}{\text{moleNH}_3 / \text{m}^3}$$

The liquid-film coefficient can be related to the oxygen reaeration rate by (Mills et al. 1982),

$$K_l = k_a H \left[\frac{32}{17} \right]^{0.25} = 1.171 k_a H$$

The gas-film coefficient is computed by

$$K_g = K_{g,H_2O} \left[\frac{18}{17} \right]^{0.25}$$

K_{g,H_2O} = a mass-transfer velocity for water vapor [m/d], which can be related to wind speed by (Schwarzenbach et al. 1993)

$$K_{g,H_2O} = (0.2U_{w,10} + 0.3) \times \frac{m}{100cm} \times \frac{86,400s}{d}$$

where $U_{w,10}$ = wind speed at a height of 10 m (m/s). Combining the above equations gives

$$K_g = 175.287U_{w,10} + 262.9305$$

The Henry's constant for ammonia at 20°C is 1.3678×10^{-5} atm m³/mole (Kavanaugh and Trussell 1980). The value at temperatures other than 20°C can be computed with

$$H_e(T) = H_e(20)1.052^{T-20}$$

5.5.15 Nitrate Nitrogen (n_n)

Nitrate nitrogen increases due to nitrification of ammonia. It is lost via denitrification and plant uptake:

$$S_{ni} = \text{Nitrif} - \text{Denitr} - (1 - P_{ab}) \frac{\text{BotAlgUptakeN}}{H} \quad (142)$$

The denitrification rate is computed as

$$\text{Denitr} = (1 - F_{oxdn})k_{dn}(T)n_n \quad (143)$$

where $k_{dn}(T)$ = the temperature-dependent denitrification rate of nitrate nitrogen [1/d] and F_{oxdn} = effect of low oxygen on denitrification [dimensionless] as modeled by Eqs. (127) to (129) with the oxygen dependency represented by the parameter K_{sodn} .

5.5.16 Organic Phosphorus (p_o)

Organic phosphorus increases due to plant death and excretion. It is lost via hydrolysis and settling.

$$S_{po} = f_{opp}q_{pp} \frac{\text{PhytoDeath}}{H} + f_{opb}q_{pb} \frac{\text{BotAlgDeath}}{H} - \text{OPHydr} - \text{OPSettl} \quad (144)$$

where f_{opp} = the fraction of the phytoplankton internal phosphorus that is in organic form which is calculated as

$$\begin{aligned}
f_{opp} &= \frac{r_{pa}}{q_{pp}} & \text{if } q_{pp} > r_{pa} \\
f_{opp} &= 1 & \text{if } q_{pp} \leq r_{pa}
\end{aligned}
\tag{145}$$

The fraction of the bottom algae internal phosphorus that is in organic form, f_{opb} , is calculated in a similar fashion.

The rate of organic phosphorus hydrolysis is computed as

$$\text{OPHydr} = k_{hp}(T) p_o \tag{146}$$

where $k_{hp}(T)$ = the temperature-dependent organic phosphorus hydrolysis rate [/d]. Organic phosphorus settling is determined as

$$\text{OPSettl} = \frac{v_{op}}{H} p_o \tag{147}$$

where v_{op} = organic phosphorus settling velocity [m/d].

5.5.17 Inorganic Phosphorus (p_i)

Inorganic phosphorus increases due to organic phosphorus hydrolysis and plant excretion. It is lost via plant uptake. In addition, a settling loss is included for cases in which inorganic phosphorus is lost due to sorption onto settleable particulate matter such as iron oxyhydroxides:

$$\begin{aligned}
S_{pi} = & \text{OPHydr} + (1 - f_{opp}) q_{pp} \text{PhytoDeath} + (1 - f_{opb}) q_{pb} \frac{\text{BotAlgDeath}}{H} \\
& + \text{PhytoExp} + \frac{\text{BotAlgExp}}{H} - \text{PhytoUpP} - \frac{\text{BotAlgUpP}}{H} - \text{IPSettl}
\end{aligned}
\tag{148}$$

where

$$\text{IPSettl} = \frac{v_{ip}}{H} p_i \tag{149}$$

where v_{ip} = inorganic phosphorus settling velocity [m/d].

5.5.18 Inorganic Suspended Solids (m_i)

Inorganic suspended solids are lost via settling,

$$S_{mi} = - \text{InorgSettl}$$

where

$$\text{InorgSettl} = \frac{v_i}{H} m_i \tag{150}$$

where v_i = inorganic suspended solids settling velocity [m/d].

5.5.19 Dissolved Oxygen (o)

Dissolved oxygen increases due to plant photosynthesis. It is lost via fast CBOD oxidation, nitrification and plant respiration. Depending on whether the water is undersaturated or oversaturated it is gained or lost via reaeration,

$$S_o = r_{oa} \text{PhytoPhoto} + r_{oa} \frac{\text{BotAlgPhoto}}{H} - r_{oc} \text{FastCOxid} - r_{on} \text{NH4Nitr} - r_{oa} \text{PhytoResp} - r_{oa} \frac{\text{BotAlgResp}}{H} + \text{OxReaer} \quad (151)$$

where

$$\text{OxReaer} = k_a(T)(o_s(T, \text{elev}) - o) \quad (152)$$

where $k_a(T)$ = the temperature-dependent oxygen reaeration coefficient [/d], $o_s(T, \text{elev})$ = the saturation concentration of oxygen [mgO₂/L] at temperature, T , and elevation above sea level, elev .

5.5.19.1 Oxygen Saturation

The following equation is used to represent the dependence of oxygen saturation on temperature (APHA 1995)

$$\ln o_s(T, 0) = -139.34411 + \frac{1.575701 \times 10^5}{T_a} - \frac{6.642308 \times 10^7}{T_a^2} + \frac{1.243800 \times 10^{10}}{T_a^3} - \frac{8.621949 \times 10^{11}}{T_a^4} \quad (153)$$

where $o_s(T, 0)$ = the saturation concentration of dissolved oxygen in freshwater at 1 atm [mgO₂/L] and T_a = absolute temperature [K] where $T_a = T + 273.15$.

The effect of elevation is accounted for by

$$o_s(T, \text{elev}) = e^{\ln o_s(T, 0)}(1 - 0.0001148 \text{elev}) \quad (154)$$

5.5.19.2 Reaeration Formulas

The reaeration coefficient (at 20 °C) can be prescribed on the **Reach Worksheet**. If reaeration is not prescribed (that is, it is blank or zero for a particular reach), it is computed as a function of the river's hydraulics and (optionally) wind velocity,

$$k_a(20) = k_{ah}(20) + \frac{K_{Lw}(20)}{H} \quad (155)$$

where $k_{ah}(20)$ = the reaeration rate at 20 °C computed based on the river's hydraulic characteristics [d^{-1}], $K_L(20)$ = the reaeration mass-transfer coefficient based on wind velocity [m/d], and H = mean depth [m].

Hydraulic-based Formulas:

O'Connor-Dobbins (O'Connor and Dobbins 1958):

$$k_{ah}(20) = 3.93 \frac{U^{0.5}}{H^{1.5}} \quad (156)$$

where U = mean water velocity [m/s] and H = mean water depth [m].

Churchill (Churchill et al. 1962):

$$k_{ah}(20) = 5.026 \frac{U}{H^{1.67}} \quad (157)$$

Owens-Gibbs (Owens et al. 1964):

$$k_{ah}(20) = 5.32 \frac{U^{0.67}}{H^{1.85}} \quad (158)$$

Tsivoglou-Neal (Tsivoglou and Neal 1976):

Low flow, $Q = 0.0283$ to 0.4247 cms (1 to 15 cfs):

$$k_{ah}(20) = 31,183 US \quad (159)$$

High flow, $Q = 0.4247$ to 84.938 cms (15 to 3000 cfs):

$$k_{ah}(20) = 15,308 US \quad (160)$$

where S = channel slope [m/m].

Thackston-Dawson (Thackston and Dawson 2001):

$$k_{ah}(20) = 2.16(1 + 9F^{0.25}) \frac{U_*}{H} \quad (161)$$

where U_* = shear velocity [m/s], and F = the Froude number [dimensionless]. The shear velocity and the Froude number are defined as

$$U_* = \sqrt{gR_h S} \quad (162)$$

and

$$F = \frac{U}{\sqrt{gH_d}} \quad (163)$$

where g = gravitational acceleration ($= 9.81 \text{ m/s}^2$), R_h = hydraulic radius [m], S = channel slope [m/m], and H_d = the hydraulic depth [m]. The hydraulic depth is defined as

$$H_d = \frac{A_c}{B_t} \quad (164)$$

where B_t = the top width of the channel [m].

USGS (Pool-riffle) (Melching and Flores 1999):

Low flow, $Q < 0.556 \text{ cms}$ ($< 19.64 \text{ cfs}$):

$$k_{ah}(20) = 517(US)^{0.524} Q^{-0.242} \quad (165)$$

High flow, $Q > 0.556 \text{ cms}$ ($> 19.64 \text{ cfs}$):

$$k_{ah}(20) = 596(US)^{0.528} Q^{-0.136} \quad (166)$$

where Q = flow (cms).

USGS (Channel-control) (Melching and Flores 1999):

Low flow, $Q < 0.556 \text{ cms}$ ($< 19.64 \text{ cfs}$):

$$k_{ah}(20) = 88(US)^{0.313} H^{-0.353} \quad (167)$$

High flow, $Q > 0.556 \text{ cms}$ ($> 19.64 \text{ cfs}$):

$$k_{ah}(20) = 142(US)^{0.333} H^{-0.66} B_t^{-0.243} \quad (168)$$

Internal (Covar 1976):

Reaeration can also be internally calculated based on the following scheme patterned after a plot developed by Covar (1976) (Figure 21):

- ☐ If $H < 0.61 \text{ m}$, use the Owens-Gibbs formula
- ☐ If $H > 0.61 \text{ m}$ and $H > 3.45U^{2.5}$, use the O'Connor-Dobbins formula
- ☐ Otherwise, use the Churchill formula

This is referred to as option **Internal** on the **Rates Worksheet** of Q2K. Note that if no option is specified, the Internal option is the default.

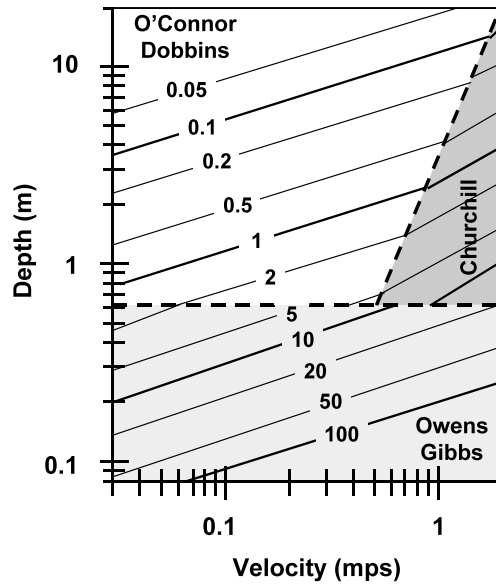


Figure 21 Reaeration rate (K_d) versus depth and velocity (Covar 1976).

Wind-based Formulas:

Three options are available to incorporate wind effects into the reaeration rate: (1) it can be omitted, (2) the Banks-Herrera formula and (3) the Wanninkhof formula.

Banks-Herrera formula (Banks 1975, Banks and Herrera 1977):

$$K_{lw} = 0.728U_{w,10}^{0.5} - 0.317U_{w,10} + 0.0372U_{w,10}^2 \quad (169)$$

where $U_{w,10}$ = wind speed measured 10 meters above the water surface [m s^{-1}]

Wanninkhof formula (Wanninkhof 1991):

$$K_{lw} = 0.0986U_{w,10}^{1.64} \quad (170)$$

Note that the model uses Eq. (48) to correct the wind velocity entered on the **Meteorology Worksheet** (7 meters above the surface) so that it is scaled to the 10-m height.

5.5.19.3 Effect of Control Structures: Oxygen

Oxygen transfer in streams is influenced by the presence of control structures such as weirs, dams, locks, and waterfalls (Figure 22). Butts and Evans (1983) have reviewed efforts to characterize this transfer and have suggested the following formula,

$$r_d = 1 + 0.38a_d b_d H_d (1 - 0.11H_d)(1 + 0.046T) \quad (171)$$

where r_d = the ratio of the deficit above and below the dam, H_d = the difference in water elevation [m] as calculated with Eq. (7), T = water temperature ($^{\circ}\text{C}$) and a_d and b_d are coefficients that correct for water-quality and dam-type. Values of a_d and b_d are summarized in Table 7. If no

values are specified, Q2K uses the following default values for these coefficients: $a_d = 1.25$ and $b_d = 0.9$.

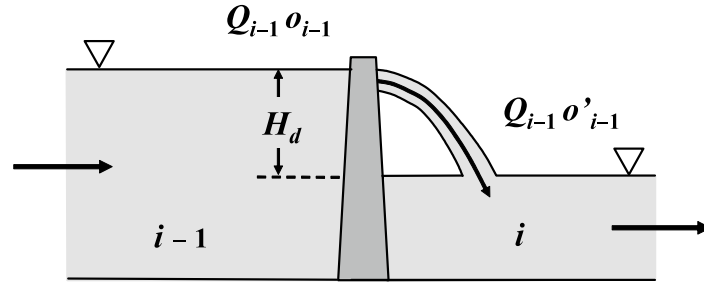


Figure 22 Water flowing over a river control structure.

Table 7 Coefficient values used to predict the effect of dams on stream reaeration.

(a) Water-quality coefficient	
Polluted state	a_d
Gross	0.65
Moderate	1.0
Slight	1.6
Clean	1.8
(b) Dam-type coefficient	
Dam type	b_d
Flat broad-crested regular step	0.70
Flat broad-crested irregular step	0.80
Flat broad-crested vertical face	0.60
Flat broad-crested straight-slope face	0.75
Flat broad-crested curved face	0.45
Round broad-crested curved face	0.75
Sharp-crested straight-slope face	1.00
Sharp-crested vertical face	0.80
Sluice gates	0.05

The oxygen mass balance for the element below the structure is written as

$$\frac{do_i}{dt} = \frac{Q_{i-1}}{V_i} o'_{i-1} - \frac{Q_i}{V_i} o_i - \frac{Q_{ab,i}}{V_i} o_i + \frac{E'_i}{V_i} (o_{i+1} - o_i) + \frac{W_{o,i}}{V_i} + S_{o,i} \quad (172)$$

where o'_{i-1} = the oxygen concentration entering the element [mgO_2/L], where

$$o'_{i-1} = o_{s,i-1} - \frac{o_{s,i-1} - o_{i-1}}{r_d} \quad (173)$$

5.5.20 Pathogen (X)

Pathogens are subject to death and settling,

$$S_X = -\text{PathDeath} - \text{PathSettl} \quad (174)$$

5.5.20.1 Death

Pathogen death is due to natural die-off and light (Chapra 1997). The death of pathogens in the absence of light is modeled as a first-order temperature-dependent decay and the death rate due to light is based on the Beer-Lambert law,

$$\text{PathDeath} = k_{dX}(T)X + \alpha_{path} \frac{I(0)/24}{k_e H} (1 - e^{-k_e H})X \quad (175)$$

where $k_{dX}(T)$ = temperature-dependent pathogen die-off rate [1/d] and α_{path} = a light efficiency factor [dimensionless].

5.5.20.2 Settling

Pathogen settling is represented as

$$\text{PathSettl} = \frac{v_X}{H} x \quad (176)$$

where v_X = pathogen settling velocity [m/d].

5.5.21 pH

The following equilibrium, mass balance and electroneutrality equations define a freshwater dominated by inorganic carbon (Stumm and Morgan 1996),

$$K_1 = \frac{[\text{HCO}_3^-][\text{H}^+]}{[\text{H}_2\text{CO}_3^*]} \quad (177)$$

$$K_2 = \frac{[\text{CO}_3^{2-}][\text{H}^+]}{[\text{HCO}_3^-]} \quad (178)$$

$$K_w = [\text{H}^+][\text{OH}^-] \quad (179)$$

$$c_T = [\text{H}_2\text{CO}_3^*] + [\text{HCO}_3^-] + [\text{CO}_3^{2-}] \quad (180)$$

$$\text{Alk} = [\text{HCO}_3^-] + 2[\text{CO}_3^{2-}] + [\text{OH}^-] - [\text{H}^+] \quad (181)$$

where K_1 , K_2 and K_w are acidity constants, Alk = alkalinity [eq L⁻¹], H_2CO_3^* = the sum of dissolved carbon dioxide and carbonic acid, HCO_3^- = bicarbonate ion, CO_3^{2-} = carbonate ion, H^+ = hydronium ion, OH^- = hydroxyl ion, and c_T = total inorganic carbon concentration [mole L⁻¹]. The brackets [] designate molar concentrations.

Note that the alkalinity is expressed in units of eq/L for the internal calculations. For input and output, it is expressed as mgCaCO₃/L. The two units are related by

$$Alk(\text{mgCaCO}_3/\text{L}) = 50,000 \times Alk(\text{eq/L}) \quad (182)$$

The equilibrium constants are corrected for temperature by

Harned and Hamer (1933):

$$\text{p}K_w = \frac{4787.3}{T_a} + 7.1321 \log_{10}(T_a) + 0.010365T_a - 22.80 \quad (183)$$

Plummer and Busenberg (1982):

$$\log K_1 = -356.3094 - 0.06091964T_a + 21834.37/T_a + 126.8339 \log T_a - 1,684,915/T_a^2 \quad (184)$$

Plummer and Busenberg (1982):

$$\log K_2 = -107.8871 - 0.03252849T_a + 5151.79/T_a + 38.92561 \log T_a - 563,713.9/T_a^2 \quad (185)$$

The nonlinear system of five simultaneous equations (177 through 181) can be solved numerically for the five unknowns: [H₂CO₃*], [HCO₃⁻], [CO₃²⁻], [OH⁻], and {H⁺}. An efficient solution method can be derived by combining Eqs. (177), (178) and (180) to define the quantities (Stumm and Morgan 1996)

$$\alpha_0 = \frac{[\text{H}^+]^2}{[\text{H}^+]^2 + K_1[\text{H}^+] + K_1K_2} \quad (186)$$

$$\alpha_1 = \frac{K_1[\text{H}^+]}{[\text{H}^+]^2 + K_1[\text{H}^+] + K_1K_2} \quad (187)$$

$$\alpha_2 = \frac{K_1K_2}{[\text{H}^+]^2 + K_1[\text{H}^+] + K_1K_2} \quad (188)$$

where α_0 , α_1 , and α_2 = the fraction of total inorganic carbon in carbon dioxide, bicarbonate, and carbonate, respectively. Equations (179), (187), and (188) can be substituted into Eq. (181) to yield,

$$\text{Alk} = (\alpha_1 + 2\alpha_2)c_T + \frac{K_w}{[\text{H}^+]} - [\text{H}^+] \quad (189)$$

Thus, solving for pH reduces to determining the root, {H⁺}, of

$$f([H^+]) = (\alpha_1 + 2\alpha_2)c_T + \frac{K_w}{[H^+]} - [H^+] - \text{Alk} \quad (190)$$

where pH is then calculated with

$$\text{pH} = -\log_{10}[H^+] \quad (191)$$

The root of Eq. (190) is determined with a numerical method. The user can choose either bisection, Newton-Raphson or Brent's method (Chapra and Canale 2006, Chapra 2007) as specified on the QUAL2K sheet. The Newton-Raphson is the fastest but can sometimes diverge. In contrast, the bisection method is slower, but more reliable. Because it balances speed with reliability, Brent's method is the default.

5.5.22 Total Inorganic Carbon (c_T)

Total inorganic carbon concentration increases due to fast carbon oxidation and plant respiration. It is lost via plant photosynthesis. Depending on whether the water is undersaturated or oversaturated with CO_2 , it is gained or lost via reaeration,

$$S_{cT} = r_{cco} \text{FastCOxid} + r_{cca} \text{PhytoResp} + r_{cca} \frac{\text{BotAlgResp}}{H} - r_{cca} \text{PhytoPhoto} - r_{cca} \frac{\text{BotAlgPhoto}}{H} + \text{CO2Reaer} \quad (192)$$

where

$$\text{CO2Reaer} = k_{ac}(T)([\text{CO}_2]_s - \alpha_0 c_T) \quad (193)$$

where $k_{ac}(T)$ = the temperature-dependent carbon dioxide reaeration coefficient [1/d], and $[\text{CO}_2]_s$ = the saturation concentration of carbon dioxide [mole/L].

The stoichiometric coefficients are computed as⁴

$$r_{cca} = r_{ca} \left(\frac{\text{gC}}{\text{mgA}} \right) \times \frac{\text{moleC}}{12 \text{ gC}} \times \frac{\text{m}^3}{1000 \text{ L}} \quad (194)$$

$$r_{cco} = \frac{1}{r_{oc}} \left(\frac{\text{gC}}{\text{gO}_2} \right) \frac{\text{moleC}}{12 \text{ gC}} \times \frac{\text{m}^3}{1000 \text{ L}} \quad (195)$$

5.5.22.1 Carbon Dioxide Saturation

The CO_2 saturation is computed with Henry's law,

$$[\text{CO}_2]_s = K_H p_{\text{CO}_2} \quad (196)$$

⁴ The conversion, $\text{m}^3 = 1000 \text{ L}$ is included because all mass balances express volume in m^3 , whereas total inorganic carbon is expressed as mole/L.

where K_H = Henry's constant [mole (L atm)⁻¹] and p_{CO_2} = the partial pressure of carbon dioxide in the atmosphere [atm]. Note that the partial pressure is input on the **Rates Worksheet** in units of ppm. The program internally converts ppm to atm using the conversion: 10^{-6} atm/ppm.

The value of K_H can be computed as a function of temperature by (Edmond and Gieskes 1970)

$$pK_H = -\frac{2385.73}{T_a} - 0.0152642T_a + 14.0184 \quad (197)$$

The partial pressure of CO₂ in the atmosphere has been increasing, largely due to the combustion of fossil fuels (Figure 23). Values in 2007 are approximately $10^{-3.416}$ atm (= 383.7 ppm).

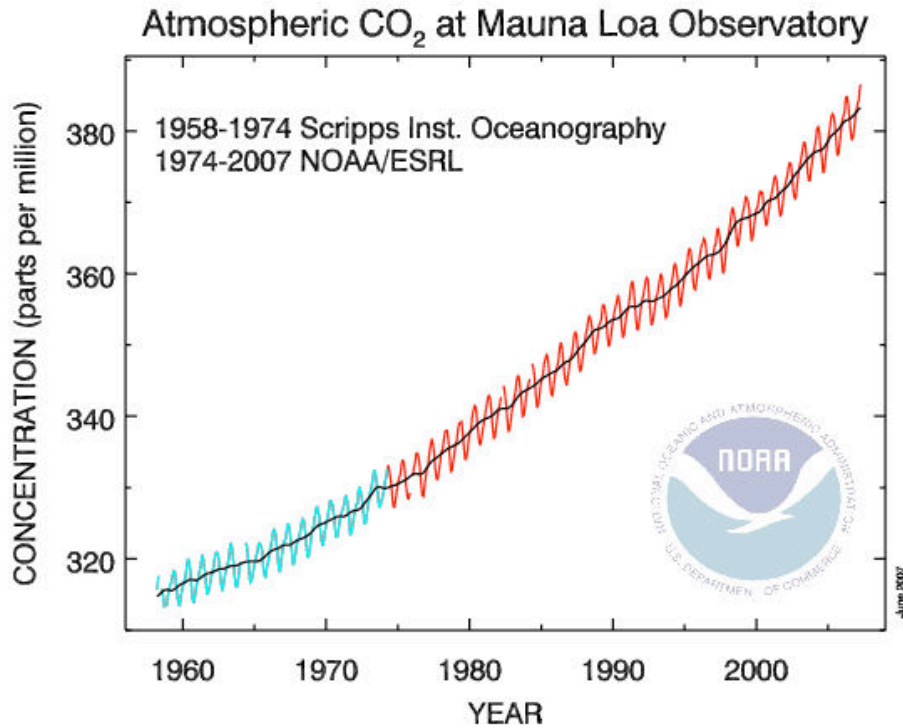


Figure 23 Concentration of carbon dioxide in the atmosphere as recorded at Mauna Loa Observatory, Hawaii.⁵

5.5.22.2 CO₂ Gas Transfer

The CO₂ reaeration coefficient can be computed from the oxygen reaeration rate by

$$k_{ac}(20) = \left(\frac{32}{44}\right)^{0.25} = 0.923 k_a(20) \quad (198)$$

⁵ Dr. Pieter Tans, NOAA/ESRL (www.esrl.noaa.gov/gmd/cgg/trends)

5.5.22.3 Effect of Control Structures: CO₂

As was the case for dissolved oxygen, carbon dioxide gas transfer in streams can be influenced by the presence of control structures. Q2K assumes that carbon dioxide behaves similarly to dissolved oxygen (recall Sec. 5.5.19.3). Thus, the inorganic carbon mass balance for the element immediately downstream of the structure is written as

$$\frac{dc_{T,i}}{dt} = \frac{Q_{i-1}}{V_i} c'_{T,i-1} - \frac{Q_i}{V_i} c_{T,i} - \frac{Q_{ab,i}}{V_i} c_{T,i} + \frac{E'_i}{V_i} (c_{T,i+1} - c_{T,i}) + \frac{W_{cT,i}}{V_i} + S_{cT,i} \quad (199)$$

where $c'_{T,i-1}$ = the concentration of inorganic carbon entering the element [mgO₂/L], where

$$c'_{T,i-1} = (\alpha_1 + \alpha_2) c_{T,i-1} + CO_{2,s,i-1} - \frac{CO_{2,s,i-1} - \alpha_2 c_{T,i-1}}{r_d} \quad (200)$$

where r_d is calculated with Eq. (171).

5.5.23 Alkalinity (Alk)

As summarized in the present model accounts for changes in alkalinity due to several mechanisms:

Table 8 Processes that effect alkalinity.

Process	Utilize	Create	Alkalinity change
Nitrif	NH4	NO3	Decrease
Denitr	NO3		Increase
OPHydr		SRP	Decrease
ONHydr		NH4	Increase
PhytoPhoto	NH4		Decrease
	NO3		Increase
	SRP		Increase
PhytoResp		NH4	Increase
		SRP	Decrease
PhytoUpN	NH4		Decrease
	NO3		Increase
PhytoUpP	SRP		Increase
PhytoExcrN		NH4	Increase
PhytoExcrP		SRP	Decrease
BotAlgUpN	NH4		Decrease
	NO3		Increase
BotAlgUpP	SRP		Increase
BotAlgExcrN		NH4	Increase
BotAlgExcrP		SRP	Decrease

5.5.23.1 Nitrification

According to Eq. (60), nitrification utilizes ammonium and creates nitrate. Hence, because a positive ion is taken up and a negative ion is created, the alkalinity is decreased by two equivalents. The change in alkalinity can be related to the nitrification rate by

$$S_{a,nitr} = - \frac{2 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \text{Nitrif} \left(\frac{\mu\text{gN}}{\text{L d}} \right) \quad (201)$$

5.5.23.2 Denitrification

According to Eq. (61), denitrification utilizes nitrate and creates nitrogen gas. Hence, because a negative ion is taken up and a neutral compound is created, the alkalinity is increased by one equivalent. The change in alkalinity can be related to the denitrification rate by

$$S_{a,denitr} = \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \text{Denitr} \left(\frac{\mu\text{gN}}{\text{L d}} \right) \quad (202)$$

where the r 's are ratios that translate the processes into the corresponding amount of alkalinity. The stoichiometric coefficients are derived from Eqs. (58) through (61) as in

5.5.23.3 Organic P Hydrolysis

Hydrolysis of organic P results in the creation of inorganic phosphate. Depending on the pH, the phosphate will either have 1 (pH \approx 2 to 7) or 2 (pH \approx 7 to 12) negative charges. Hence, because negative ions are being created, the alkalinity is decreased by one or two equivalents, respectively. The change in alkalinity can be related to the P hydrolysis rate by⁶,

$$S_{a,OPh} = -(\alpha_{H_2PO_4} + 2\alpha_{HPO_4} + 3\alpha_{PO_4}) \frac{\text{eq}}{\text{moleP}} \frac{\text{moleP}}{30.974 \text{ gP}} \frac{\text{gP}}{10^6 \mu\text{gP}} \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \text{OPHydr} \left(\frac{\mu\text{gP}}{\text{L d}} \right) \quad (203)$$

where

$$\alpha_{H_2PO_4} = \frac{K_{p1}[\text{H}^+]^2}{[\text{H}^+]^3 + K_{p1}[\text{H}^+]^2 + K_{p1}K_{p2}[\text{H}^+] + K_{p1}K_{p2}K_{p3}} \quad (204)$$

$$\alpha_{HPO_4} = \frac{K_{p1}K_{p2}[\text{H}^+]}{[\text{H}^+]^3 + K_{p1}[\text{H}^+]^2 + K_{p1}K_{p2}[\text{H}^+] + K_{p1}K_{p2}K_{p3}} \quad (205)$$

$$\alpha_{PO_4} = \frac{K_{p1}K_{p2}K_{p3}}{[\text{H}^+]^3 + K_{p1}[\text{H}^+]^2 + K_{p1}K_{p2}[\text{H}^+] + K_{p1}K_{p2}K_{p3}} \quad (206)$$

⁶ Note that although it will almost always be negligible, Eq. (193) PO_4^{3-} - for completeness.

where $K_{p1} = 10^{-2.15}$, $K_{p2} = 10^{-7.2}$, and $K_{p3} = 10^{-12.35}$.

5.5.23.4 Organic N Hydrolysis

Hydrolysis of organic N results in the creation of ammonia. Depending on the pH, the ammonia will either be in the form of ammonium ion with a single positive charge ($\text{pH} < 9$) or neutral ammonia gas ($\text{pH} > 9$). Hence, when the positive ions are created, the alkalinity is increased by one equivalent. The change in alkalinity can be related to the N hydrolysis rate by

$$S_{a,ONh} = F_i \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \text{ONHydr} \left(\frac{\mu\text{gN}}{\text{L d}} \right) \quad (207)$$

5.5.23.5 Phytoplankton Photosynthesis

Phytoplankton photosynthesis takes up nitrogen as either ammonia or nitrate and phosphorus as inorganic phosphate. If ammonia is the primary nitrogen source, this leads to a decrease in alkalinity because the uptake of the positively charged ammonium ions is much greater than the uptake of the negatively charged phosphate ions (recall the stoichiometry as described on p. 34). If nitrate is the primary nitrogen source, this leads to an increase in alkalinity because both nitrate and phosphate are negatively charged.

The following representation relates the change in alkalinity to phytoplankton photosynthesis depending on the nutrient sources as well as their speciation as governed by the pH,

$$\begin{aligned} S_{a,PhytP} = & \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \\ & \left(-r_{na} P_{ap} F_i \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \right. \\ & + r_{na} (1 - P_{ap}) \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \\ & \left. + r_{pa} (\alpha_{H_2PO_4} + 2\alpha_{HPO_4} + 3\alpha_{PO_4}) \frac{1 \text{ eq}}{\text{moleP}} \frac{\text{moleP}}{30.974 \text{ gP}} \frac{\text{gP}}{10^6 \mu\text{gP}} \right) \\ & \times \text{PhytoPhoto} \left(\frac{\mu\text{gA}}{\text{L d}} \right) \end{aligned} \quad (208)$$

5.5.23.6 Phytoplankton Nutrient Uptake

Phytoplankton take up nitrogen as either ammonia or nitrate and phosphorus as inorganic phosphate. The following representation relates the change in alkalinity to phytoplankton uptake rates depending on the nutrient sources as well as their speciation as governed by the pH,

$$\begin{aligned}
S_{a,PU_p} &= \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \\
&\left(\left(-P_{ap}F_i + (1 - P_{ap}) \right) \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \times \frac{\text{PhytoUpN}(\mu\text{gN/L/d})}{H(\text{m})} \right. \\
&\quad \left. + (\alpha_{H_2PO_4} + 2\alpha_{HPO_4} + 3\alpha_{PO_4}) \frac{1 \text{ eq}}{\text{moleP}} \frac{\text{moleP}}{30.974 \text{ gP}} \frac{\text{gP}}{10^6 \mu\text{gP}} \times \frac{\text{PhytoUpP}(\mu\text{gP/L/d})}{H(\text{m})} \right)
\end{aligned} \tag{209}$$

5.5.23.7 Phytoplankton Nutrient Excretion

Phytoplankton excrete ammonia and inorganic phosphate. The following representation relates the change in alkalinity to phytoplankton excretion rates including the effect of pH on the nutrient's speciation,

$$\begin{aligned}
S_{a,PE_x} &= \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \\
&\left(F_i \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \times \frac{\text{PhytoExN}(\mu\text{gN/L/d})}{H(\text{m})} \right. \\
&\quad \left. - (\alpha_{H_2PO_4} + 2\alpha_{HPO_4} + 3\alpha_{PO_4}) \frac{1 \text{ eq}}{\text{moleP}} \frac{\text{moleP}}{30.974 \text{ gP}} \frac{\text{gP}}{10^6 \mu\text{gP}} \times \frac{\text{PhytoExP}(\mu\text{gP/L/d})}{H(\text{m})} \right)
\end{aligned} \tag{210}$$

5.5.23.8 Bottom Algae Nutrient Uptake

Bottom algae take up nitrogen as either ammonia or nitrate and phosphorus as inorganic phosphate. The following representation relates the change in alkalinity to bottom algae uptake rates depending on the nutrient sources as well as their speciation as governed by the pH,

$$\begin{aligned}
S_{a,BAUp} &= \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \\
&\left(\left(-P_{ab}F_i + (1 - P_{ab}) \right) \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \times \frac{\text{BotAlgUpN}(\text{mgN/m}^2/\text{d})}{H(\text{m})} \right. \\
&\quad \left. + (\alpha_{H_2PO_4} + 2\alpha_{HPO_4} + 3\alpha_{PO_4}) \frac{1 \text{ eq}}{\text{moleP}} \frac{\text{moleP}}{30.974 \text{ gP}} \frac{\text{gP}}{10^6 \mu\text{gP}} \times \frac{\text{BotAlgUpP}(\text{mgP/m}^2/\text{d})}{H(\text{m})} \right)
\end{aligned} \tag{211}$$

5.5.23.9 Bottom Algae Nutrient Excretion

Bottom algae excrete ammonia and inorganic phosphate. The following representation relates the change in alkalinity to bottom algae excretion rates including the effect of pH on the nutrient's speciation,

$$\begin{aligned}
S_{a,BAEx} = & \frac{50,000 \text{ mgCaCO}_3}{1 \text{ eq}} \\
& \left(F_i \frac{1 \text{ eq}}{\text{moleN}} \frac{\text{moleN}}{14.007 \text{ gN}} \frac{\text{gN}}{10^6 \mu\text{gN}} \times \frac{\text{BotAlgExN}(\text{mgN/m}^2/\text{d})}{H(\text{m})} \right. \\
& \left. - (\alpha_{H_2PO_4} + 2\alpha_{HPO_4} + 3\alpha_{PO_4}) \frac{1 \text{ eq}}{\text{moleP}} \frac{\text{moleP}}{30.974 \text{ gP}} \frac{\text{gP}}{10^6 \mu\text{gP}} \times \frac{\text{BotAlgExP}(\text{mgP/m}^2/\text{d})}{H(\text{m})} \right)
\end{aligned} \tag{212}$$

5.6 SOD/Nutrient Flux Model

Sediment nutrient fluxes and sediment oxygen demand (SOD) are based on a model developed by Di Toro (Di Toro et al. 1991, Di Toro and Fitzpatrick. 1993, Di Toro 2001). The present version also benefited from James Martin's (Mississippi State University, personal communication) efforts to incorporate the Di Toro approach into EPA's WASP modeling framework.

A schematic of the model is depicted in Figure 24. As can be seen, the approach allows oxygen and nutrient sediment-water fluxes to be computed based on the downward flux of particulate organic matter from the overlying water. The sediments are divided into 2 layers: a thin (≈ 1 mm) surface aerobic layer underlain by a thicker (10 cm) lower anaerobic layer. Organic carbon, nitrogen and phosphorus are delivered to the anaerobic sediments via the settling of particulate organic matter (i.e., phytoplankton and detritus). There they are transformed by mineralization reactions into dissolved methane, ammonium and inorganic phosphorus. These constituents are then transported to the aerobic layer where some of the methane and ammonium are oxidized. The flux of oxygen from the water required for these oxidations is the sediment oxygen demand. The following sections provide details on how the model computes this SOD along with the sediment-water fluxes of carbon, nitrogen and phosphorus that are also generated in the process.

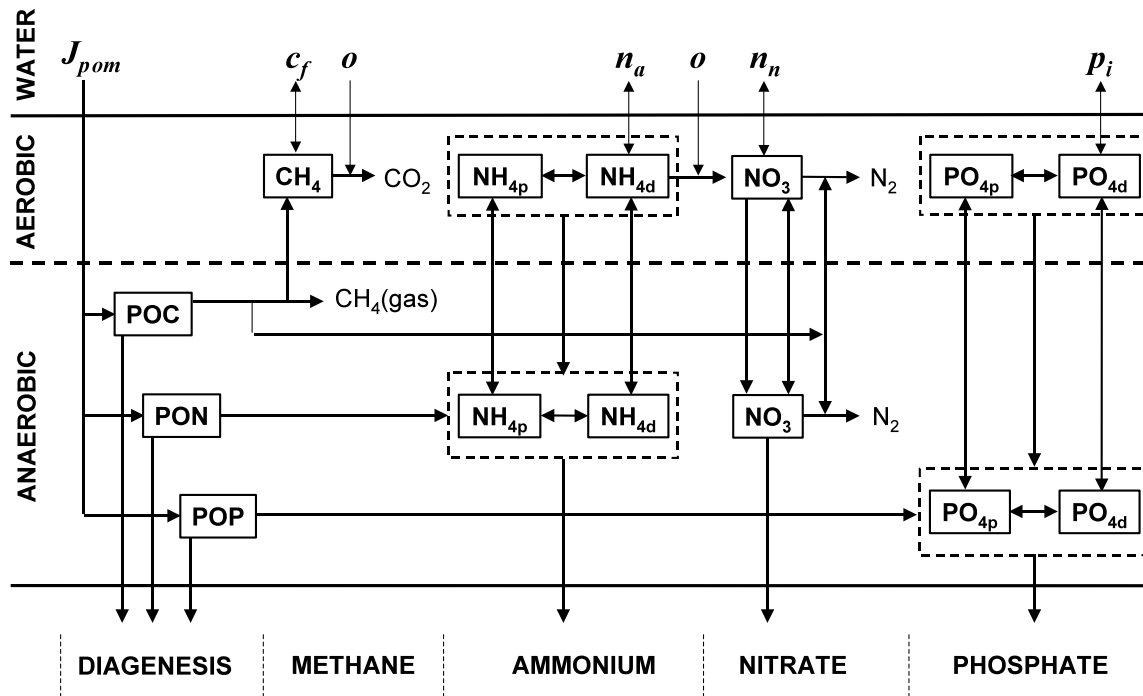


Figure 24 Schematic of SOD-nutrient flux model of the sediments.

5.6.1 Diagenesis

As summarized in Figure 25, the first step in the computation involves determining how much of the downward flux of particulate organic matter (POM) is converted into soluble reactive forms in the anaerobic sediments. This process is referred to as diagenesis. First the total downward flux

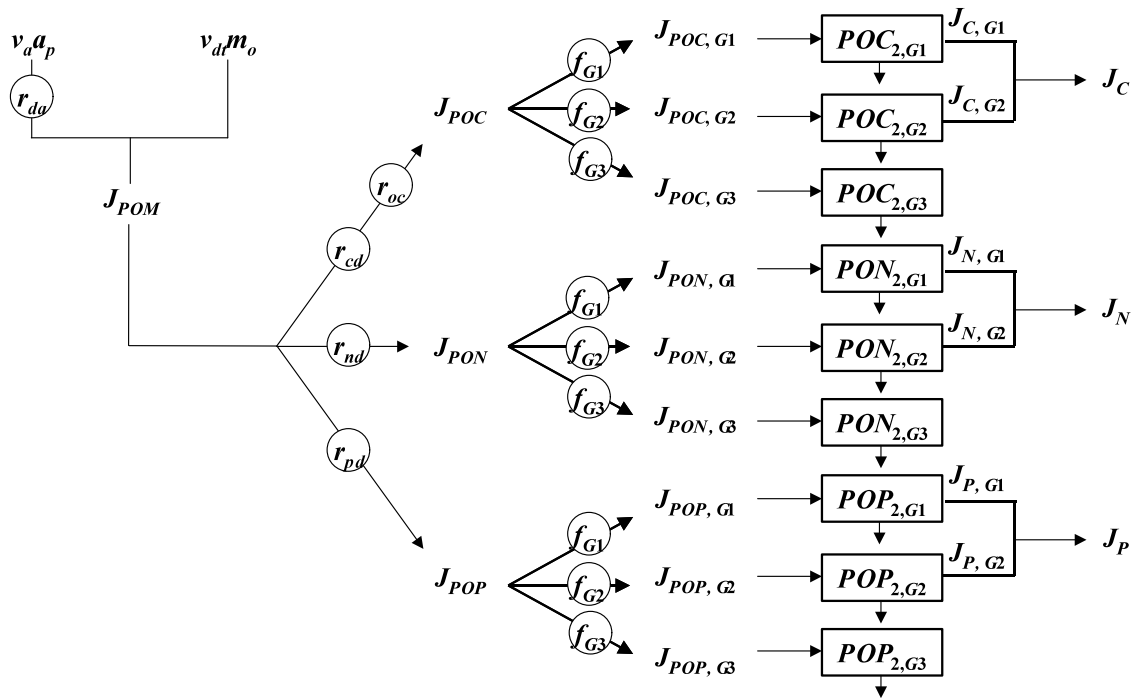
of carbon, nitrogen and phosphorus are computed as the sums of the fluxes of settling phytoplankton and organic matter delivered to the sediments from the water column

$$J_{POC} = r_{ca} v_a a_p + r_{cd} v_{dt} m_o$$

$$J_{PON} = v_a q_{Np} a_p + v_{on} n_o \quad (213)$$

$$J_{POP} = v_a q_{Pp} a_p + v_{op} p_o$$

where J_{POC} = the downward flux of POC [$\text{gC m}^{-2} \text{d}^{-1}$], r_{ca} = the ratio of carbon to chlorophyll a [gC/mgA], v_a = phytoplankton settling velocity [m/d], a_p = phytoplankton concentration [mgA/m^3], r_{cd} = the ratio of carbon to dry weight [gC/gD], v_{dt} = detritus settling velocity [m/d], m_o = detritus concentration [gD/m^3], J_{PON} = the downward flux of PON [$\text{mgN m}^{-2} \text{d}^{-1}$], q_{Np} = the phytoplankton nitrogen cell quota [mgN mgA^{-1}], v_{on} = organic nitrogen settling velocity [m/d], n_o = organic nitrogen concentration [mgN m^{-3}], J_{POP} = the downward flux of POP [$\text{mgP m}^{-2} \text{d}^{-1}$], q_{Pp} = the phytoplankton phosphorus cell quota [mgP mgA^{-1}], v_{op} = organic phosphorus settling velocity [m/d], and p_o = organic phosphorus concentration [mgP m^{-3}].



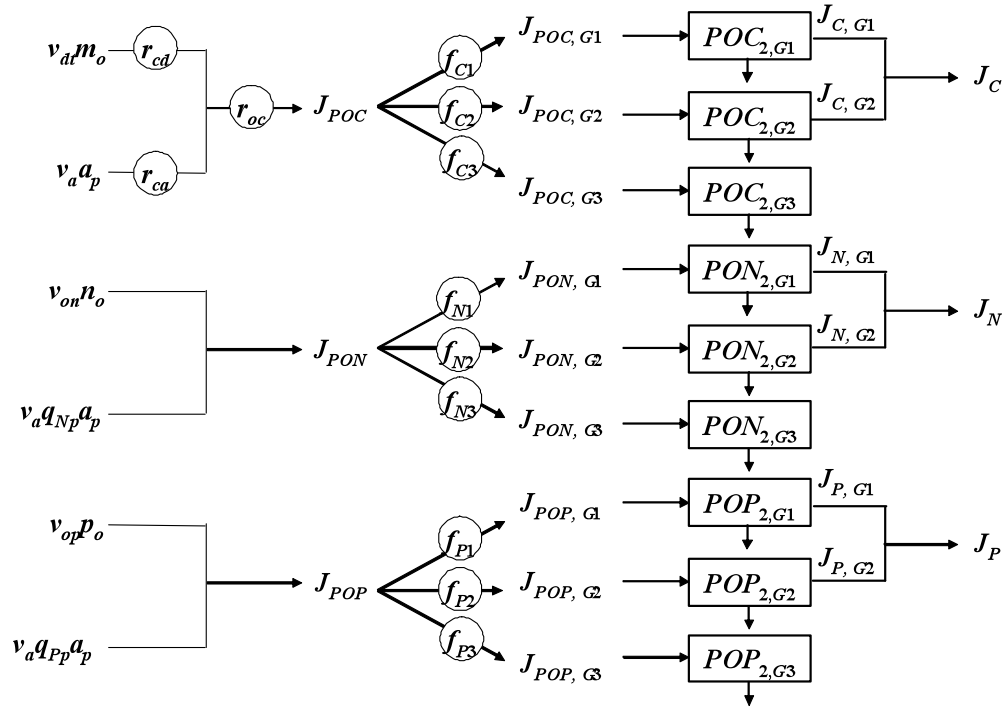


Figure 25 Representation of how settling particulate matter is transformed into fluxes of dissolved carbon (J_C), nitrogen (J_N) and phosphorus (J_P) in the anaerobic sediments.

Note that for convenience, we express the particulate organic carbon (POC) as oxygen equivalents using the stoichiometric coefficient r_{oc} . Each of the nutrient fluxes is further broken down into three reactive fractions: labile (G1), slowly reacting (G2) and non-reacting (G3).

These fluxes are then entered into mass balances to compute the concentration of each fraction in the anaerobic layer. For example, for labile POC, a mass balance is written as

$$H_2 \frac{dPOC_{2,G1}}{dt} = J_{POC,G1} - k_{POC,G1} \theta_{POC,G1}^{T-20} H_2 POC_{2,G1} - w_2 POC_{2,G1} \quad (214)$$

where H_2 = the thickness of the anaerobic layer [m], $POC_{2,G1}$ = the concentration of the labile fraction of POC in the anaerobic layer [gO_2/m^3], $J_{POC,G1}$ = the flux of labile POC delivered to the anaerobic layer [$\text{gO}_2/\text{m}^2/\text{d}$], $k_{POC,G1}$ = the mineralization rate of labile POC [d^{-1}], $\theta_{POC,G1}$ = temperature correction factor for labile POC mineralization [dimensionless], and w_2 = the burial velocity [m/d]. At steady state, Eq. 197 can be solved for

$$POC_{2,G1} = \frac{J_{POC,G1}}{k_{POC,G1} \theta_{POC,G1}^{T-20} H_2 + v_b} \quad (215)$$

The flux of labile dissolved carbon, $J_{C,G1}$ [$\text{gO}_2/\text{m}^2/\text{d}$], can then be computed as

$$J_{C,G1} = k_{POC,G1} \theta_{POC,G1}^{T-20} H_2 POC_{2,G1} \quad (216)$$

In a similar fashion, a mass balance can be written and solved for the slowly reacting dissolved organic carbon. This result is then added to Eq. 199 to arrive at the total flux of dissolved carbon generated in the anaerobic sediments.

$$J_C = J_{C,G1} + J_{C,G2} \quad (217)$$

Similar equations are developed to compute the diagenesis fluxes of nitrogen, J_N [gN/m²/d], and phosphorus J_P [gP/m²/d].

5.6.2 Ammonium

Based on the mechanisms depicted in Figure 24, mass balances can be written for total ammonium in the aerobic layer and the anaerobic layers,

$$\begin{aligned} H_1 \frac{dNH_{4,1}}{dt} = & \omega_{12} (f_{pa2} NH_{4,2} - f_{pal} NH_{4,1}) + K_{L12} (f_{da2} NH_{4,2} - f_{dal} NH_{4,1}) - w_2 NH_{4,1} \\ & + s \left(\frac{n_a}{1000} - f_{dal} NH_{4,1} \right) - \frac{\kappa_{NH4,1}^2}{s} \theta_{NH4}^{T-20} \frac{K_{NH4}}{K_{NH4} + NH_{4,1}} \frac{o}{2K_{NH4,O2} + o} f_{dal} NH_{4,1} \end{aligned} \quad (218)$$

$$\begin{aligned} H_2 \frac{dNH_{4,2}}{dt} = & J_N + \omega_{12} (f_{pal} NH_{4,1} - f_{pa2} NH_{4,2}) + K_{L12} (f_{dal} NH_{4,1} - f_{da2} NH_{4,2}) \\ & + w_2 (NH_{4,1} - NH_{4,2}) \end{aligned} \quad (219)$$

where H_1 = the thickness of the aerobic layer [m], $NH_{4,1}$ and $NH_{4,2}$ = the concentration of total ammonium in the aerobic layer and the anaerobic layers, respectively [gN/m³], n_a = the ammonium concentration in the overlying water [mgN/m³], $\kappa_{NH4,1}$ = the reaction velocity for nitrification in the aerobic sediments [m/d], θ_{NH4} = temperature correction factor for nitrification [dimensionless], K_{NH4} = ammonium half-saturation constant [gN/m³], o = the dissolved oxygen concentration in the overlying water [gO₂/m³], and $K_{NH4,O2}$ = oxygen half-saturation constant [mgO₂/L], and J_N = the diagenesis flux of ammonium [gN/m²/d].

The fraction of ammonium in dissolved (f_{dai}) and particulate (f_{pai}) form are computed as

$$f_{dai} = \frac{1}{1 + m_i \pi_{ai}} \quad (220)$$

$$f_{pai} = 1 - f_{dai} \quad (221)$$

where m_i = the solids concentration in layer i [gD/m³], and π_{ai} = the partition coefficient for ammonium in layer i [m³/gD].

The mass transfer coefficient for particle mixing due to bioturbation between the layers, ω_{12} [m/d], is computed as

$$\omega_{12} = \frac{D_p \theta_{Dp}^{T-20}}{H_2} \frac{POC_{2,G1} / r_{oc}}{POC_R} \frac{o}{K_{M,Dp} + o} \quad (222)$$

where D_p = diffusion coefficient for bioturbation [m^2/d], θ_{Dp} = temperature coefficient [dimensionless], POC_R = reference G1 concentration for bioturbation [gC/m^3] and $K_{M,Dp}$ = oxygen half-saturation constant for bioturbation [gO_2/m^3].

The mass transfer coefficient for pore water diffusion between the layers, K_{L12} [m/d], is computed as,

$$K_{L12} = \frac{D_d \theta_{Dd}^{T-20}}{H_2 / 2} \quad (223)$$

where D_d = pore water diffusion coefficient [m^2/d], and θ_{Dd} = temperature coefficient [dimensionless].

The mass transfer coefficient between the water and the aerobic sediments, s [m/d], is computed as

$$s = \frac{SOD}{o} \quad (224)$$

where SOD = the sediment oxygen demand [$gO_2/m^2/d$].

At steady state, Eqs. 201 and 202 are two simultaneous nonlinear algebraic equations. The equations can be linearized by assuming that the $NH_{4,1}$ term in the Monod term for nitrification is constant. The simultaneous linear equations can then be solved for $NH_{4,1}$ and $NH_{4,2}$. The flux of ammonium to the overlying water can then be computed as

$$J_{NH4} = s \left(f_{dal} NH_{4,1} - \frac{n_a}{1000} \right) \quad (225)$$

5.6.3 Nitrate

Mass balances for nitrate can be written for the aerobic and anaerobic layers as

$$\begin{aligned} H_1 \frac{dNO_{3,1}}{dt} = & K_{L12} (NO_{3,2} - NO_{3,1}) - w_2 NO_{3,1} + s \left(\frac{n_n}{1000} - NO_{3,1} \right) \\ & + \frac{\kappa_{NH4,1}^2}{s} \theta_{NH4}^{T-20} \frac{K_{NH4}}{K_{NH4} + NH_{4,1}} \frac{o}{2K_{NH4,O2} + o} f_{dal} NH_{4,1} - \frac{\kappa_{NO3,1}^2}{s} \theta_{NO3}^{T-20} NO_{3,1} \end{aligned} \quad (226)$$

$$H_2 \frac{dNO_{3,2}}{dt} = J_N + K_{L12} (NO_{3,1} - NO_{3,2}) + w_2 (NO_{3,1} - NO_{3,2}) - \kappa_{NO3,2} \theta_{NO3}^{T-20} NO_{3,2} \quad (227)$$

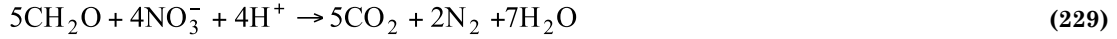
where $NO_{3,1}$ and $NO_{3,2}$ = the concentration of nitrate in the aerobic layer and the anaerobic layers, respectively [gN/m^3], n_n = the nitrate concentration in the overlying water [mgN/m^3], $\kappa_{NO3,1}$ and

$\kappa_{NO3,2}$ = the reaction velocities for denitrification in the aerobic and anaerobic sediments, respectively [m/d], and θ_{NO3} = temperature correction factor for denitrification [dimensionless].

In the same fashion as for Eqs. 201 and 202, Eqs. 209 and 210 can be linearized and solved for $NO_{3,1}$ and $NO_{3,2}$. The flux of nitrate to the overlying water can then be computed as

$$J_{NO3} = s \left(NO_{3,1} - \frac{n_n}{1000} \right) \quad (228)$$

Denitrification requires a carbon source as represented by the following chemical equation,



The carbon requirement (expressed in oxygen equivalents per nitrogen) can therefore be computed as

$$r_{ondn} = 2.67 \frac{gO_2}{gC} \frac{5 \text{ moleC} \times 12 \text{ gC/moleC}}{4 \text{ moleN} \times 14 \text{ gN/moleN}} \times \frac{1 \text{ gN}}{1000 \text{ mgN}} = 0.00286 \frac{gO_2}{mgN} \quad (230)$$

Therefore, the oxygen equivalents consumed during denitrification, $J_{O2,dn}$ [gO₂/m²/d], can be computed as

$$J_{O2,dn} = 1000 \frac{mgN}{gN} \times r_{ondn} \left(\frac{\kappa_{NO3,1}^2}{s} \theta_{NO3}^{T-20} NO_{3,1} + \kappa_{NO3,2} \theta_{NO3}^{T-20} NO_{3,2} \right) \quad (231)$$

5.6.4 Methane

The dissolved carbon generated by diagenesis is converted to methane in the anaerobic sediments. Because methane is relatively insoluble, its saturation can be exceeded and methane gas produced. As a consequence, rather than write a mass balance for methane in the anaerobic layer, an analytical model developed by Di Toro et al. (1990) is used to determine the steady-state flux of dissolved methane corrected for gas loss delivered to the aerobic sediments.

First, the carbon diagenesis flux is corrected for the oxygen equivalents consumed during denitrification,

$$J_{CH4,T} = J_C - J_{O2,dn} \quad (232)$$

where $J_{CH4,T}$ = the carbon diagenesis flux corrected for denitrification [gO₂/m²/d]. In other words, this is the total anaerobic methane production flux expressed in oxygen equivalents.

If $J_{CH4,T}$ is sufficiently large ($\geq 2K_{L12}C_s$), methane gas will form. In such cases, the flux can be corrected for the gas loss,

$$J_{CH4,d} = \sqrt{2K_{L12}C_s J_{CH4,T}} \quad (233)$$

where $J_{CH_4,d}$ = the flux of dissolved methane (expressed in oxygen equivalents) that is generated in the anaerobic sediments and delivered to the aerobic sediments [$\text{gO}_2/\text{m}^2/\text{d}$], C_s = the saturation concentration of methane expressed in oxygen equivalents [mgO_2/L]. If $J_{CH_4,T} < 2K_{L12}C_s$, then no gas forms and

$$J_{CH_4,d} = J_{CH_4,T} \quad (234)$$

The methane saturation concentration is computed as

$$C_s = 100 \left(1 + \frac{H}{10} \right) 1.024^{20-T} \quad (235)$$

where H = water depth [m] and T = water temperature [$^{\circ}\text{C}$].

A methane mass balance can then be written for the aerobic layer as

$$H_1 \frac{dCH_{4,1}}{dt} = J_{CH_4,d} + s(c_f - CH_{4,1}) - \frac{\kappa_{CH_4,1}^2}{s} \theta_{CH_4}^{T-20} CH_{4,1} \quad (236)$$

where $CH_{4,1}$ = methane concentration in the aerobic layer [gO_2/m^3], c_f = fast CBOD in the overlying water [gO_2/m^3], $\kappa_{CH_4,1}$ = the reaction velocity for methane oxidation in the aerobic sediments [m/d], and θ_{CH_4} = temperature correction factor [dimensionless]. At steady, state, this balance can be solved for

$$CH_{4,1} = \frac{J_{CH_4,d} + sc_f}{s + \frac{\kappa_{CH_4,1}^2}{s} \theta_{CH_4}^{T-20}} \quad (237)$$

The flux of methane to the overlying water, J_{CH_4} [$\text{gO}_2/\text{m}^2/\text{d}$], can then be computed as

$$J_{CH_4} = s(CH_{4,1} - c_f) \quad (238)$$

5.6.5 SOD

The SOD [$\text{gO}_2/\text{m}^2/\text{d}$] is equal to the sum of the oxygen consumed in methane oxidation and nitrification,

$$SOD = CSOD + NSOD \quad (239)$$

where $CSOD$ = the amount of oxygen demand generated by methane oxidation [$\text{gO}_2/\text{m}^2/\text{d}$] and $NSOD$ = the amount of oxygen demand generated by nitrification [$\text{gO}_2/\text{m}^2/\text{d}$]. These are computed as

$$CSOD = \frac{\kappa_{CH_4,1}^2}{s} \theta_{CH_4}^{T-20} CH_{4,1} \quad (240)$$

$$NSOD = r_{on} \frac{\kappa_{NH4,1}^2}{s} \theta_{NH4}^{T-20} \frac{K_{NH4}}{K_{NH4} + NH_{4,1}} \frac{o}{2K_{NH4,O2} + o} f_{dal} NH_{4,1} \quad (241)$$

where r_{on} = the ratio of oxygen to nitrogen consumed during nitrification [= 4.57 gO₂/gN].

5.6.6 Inorganic Phosphorus

Mass balances can be written total inorganic phosphorus in the aerobic layer and the anaerobic layers as

$$H_1 \frac{dPO_{4,1}}{dt} = \omega_{12} (f_{pp2} PO_{4,2} - f_{pp1} PO_{4,1}) + K_{L12} (f_{dp2} PO_{4,2} - f_{dp1} PO_{4,1}) - w_2 PO_{4,1} + s \left(\frac{p_i}{1000} - f_{dal} PO_{4,1} \right) \quad (242)$$

$$H_2 \frac{dPO_{4,2}}{dt} = J_P + \omega_{12} (f_{pp1} PO_{4,1} - f_{pp2} PO_{4,2}) + K_{L12} (f_{dp1} PO_{4,1} - f_{dp2} PO_{4,2}) + w_2 (PO_{4,1} - PO_{4,2}) \quad (243)$$

where $PO_{4,1}$ and $PO_{4,2}$ = the concentration of total inorganic phosphorus in the aerobic layer and the anaerobic layers, respectively [gP/m³], p_i = the inorganic phosphorus in the overlying water [mgP/m³], and J_P = the diagenesis flux of phosphorus [gP/m²/d].

The fraction of phosphorus in dissolved (f_{dpi}) and particulate (f_{ppi}) form are computed as

$$f_{dpi} = \frac{1}{1 + m_i \pi_{pi}} \quad (244)$$

$$f_{ppi} = 1 - f_{dpi} \quad (245)$$

where π_{pi} = the partition coefficient for inorganic phosphorus in layer i [m³/gD].

The partition coefficient in the anaerobic layer is set to an input value. For the aerobic layer, if the oxygen concentration in the overlying water column exceeds a critical concentration, O_{crit} [gO₂/m³], then the partition coefficient is increased to represent the sorption of phosphorus onto iron oxyhydroxides as in

$$\pi_{p1} = \pi_{p2} (\Delta \pi_{PO4,1}) \quad (246)$$

where $\Delta \pi_{PO4,1}$ is a factor that increases the aerobic layer partition coefficient relative to the anaerobic coefficient.

If the oxygen concentration falls below O_{crit} then the partition coefficient is decreased smoothly until it reaches the anaerobic value at zero oxygen,

$$\pi_{p1} = \pi_{p2} (\Delta \pi_{PO4,1})^{1/o_{crit}} \quad (247)$$

Equations 225 and 226 can be solved for $PO_{4,1}$ and $PO_{4,2}$. The flux of phosphorus to the overlying water can then be computed as

$$J_{PO4} = s \left(PO_{4,1} - \frac{P_i}{1000} \right) \quad (248)$$

5.6.7 Solution Scheme

Although the foregoing sequence of equations can be solved, a single computation will not yield a correct result because of the interdependence of the equations. For example, the surface mass transfer coefficient s depends on SOD. The SOD in turn depends on the ammonium and methane concentrations which themselves are computed via mass balances that depend on s . Hence, an iterative technique must be used. The procedure used in QUAL2K is

1. Determine the diagenesis fluxes: J_C , J_N and J_P .
2. Start with an initial estimate of SOD,

$$SOD_{init} = J_C + r_{on}' J_N \quad (249)$$

where r_{on}' = the ratio of oxygen to nitrogen consumed for total conversion of ammonium to nitrogen gas via nitrification/denitrification [= 1.714 gO₂/gN]. This ratio accounts for the carbon utilized for denitrification.

3. Compute s using

$$s = \frac{SOD_{init}}{o} \quad (250)$$

4. Solve for ammonium, nitrate and methane, and compute the $CSOD$ and $NSOD$.
5. Make a revised estimate of SOD using the following weighted average

$$SOD = \frac{SOD_{init} + CSOD + NSOD}{2} \quad (251)$$

6. Check convergence by calculating an approximate relative error

$$\varepsilon_a = \left| \frac{SOD - SOD_{init}}{SOD} \right| \times 100\% \quad (252)$$

7. If ε_a is greater than a prespecified stopping criterion ε_s then set $SOD_{init} = SOD$ and return to step 2.
8. If convergence is adequate ($\varepsilon_a \leq \varepsilon_s$), then compute the inorganic phosphorus concentrations.

9. Compute the ammonium, nitrate, methane and phosphate fluxes.

5.6.8 Supplementary Fluxes

Because of the presence of organic matter deposited prior to the summer steady-state period (e.g., during spring runoff), it is possible that the downward flux of particulate organic matter is insufficient to generate the observed SOD. In such cases, a supplementary SOD can be prescribed,

$$SOD_t = SOD + SOD_s \quad (253)$$

where SOD_t = the total sediment oxygen demand [$\text{gO}_2/\text{m}^2/\text{d}$], and SOD_s = the supplementary SOD [$\text{gO}_2/\text{m}^2/\text{d}$]. In addition, prescribed ammonia and methane fluxes can be used to supplement the computed fluxes.

REFERENCES

- Adams, E.E., D. J. Cosler, and K.R. Helfrich. 1987. "Analysis of evaporation data from heated ponds", cs-5171, research project 2385-1, Electric Power Research Institute, Palo Alto, California 94304. April.
- Andrews and Rodvey, 1980. Heat exchange between water and tidal flats. D.G.M 24(2). (in German).
- APHA, 1995. Standard methods for the examination of water and wastewater, 19th Edn. American Public Health Association, American Water Works Association and Water Environment Federation: Washington, D.C.
- Asaeda, T., and T. Van Bon, Modelling the effects of macrophytes on algal blooming in eutrophic shallow lakes. *Ecol. Model.*, 104: 261-287, 1997.
- Baker, K.S. and Frouin, R. 1987. Relation between photosynthetically available radiation and total insolation at the ocean surface under clear skies. *Limnol. Oceanogr.* 1370-1377.
- Baly, E.C.C. 1935. The Kinetics of Photosynthesis. *Proc. Royal Soc. London Ser. B*, 117:218-239.
- Banks, R. B. 1975. "Some Features of Wind Action on Shallow Lakes." *J. Environ Engr. Div. ASCE*. 101(E5): 813-827.
- Banks, R. B. and Herrera, F. F. 1977. "Effect of Wind and Rain on Surface Reaeration." *J. Environ Engr. Div. ASCE*. 103(E3): 489-504.
- Barnwell, T.O., Brown, L.C., and Mareck, W. 1989). Application of Expert Systems Technology in Water Quality Modeling. *Water Sci. Tech.* 21(8-9):1045-1056.
- Bejan, A. 1993. *Heat Transfer*. Wiley, New York, NY.
- Bowie, G.L., Mills, W.B., Porcella, D.B., Campbell, C.L., Pagenkopf, J.R., Rupp, G.L., Johnson, K.M., Chan, P.W.H., Gherini, S.A. and Chamberlin, C.E. 1985. Rates, Constants, and Kinetic Formulations in Surface Water Quality Modeling. U.S. Envir. Prot. Agency, ORD, Athens, GA, ERL, EPA/600/3-85/040.
- Brady, D.K., Graves, W.L., and Geyer, J.C. 1969. Surface Heat Exchange at Power Plant Cooling Lakes, Cooling Water Discharge Project Report, No. 5, Edison Electric Inst. Pub. No. 69-901, New York, NY.
- Bras, R.L. 1990. Hydrology. Addison-Wesley, Reading, MA.
- Brown, L.C., and Barnwell, T.O. 1987. The Enhanced Stream Water Quality Models QUAL2E and QUAL2E-UNCAS, EPA/600/3-87-007, U.S. Environmental Protection Agency, Athens, GA, 189 pp.
- Brunt, D. 1932. Notes on radiation in the atmosphere: I. *Quart J Royal Meteorol Soc* 58:389-420.
- Brutsaert, W. 1982. Evaporation into the atmosphere: theory, history, and applications. D.Reidel Publishing Co., Hingham MA, 299 p.
- Butts, T. A. and Evans, R. L. 1983. Effects of Channel Dams on Dissolved Oxygen Concentrations in Northeastern Illinois Streams, Circular 132, State of Illinois, Dept. of Reg. and Educ., Illinois Water Survey, Urbana, IL.
- Carslaw, H.S. and Jaeger, J.C. 1959. Conduction of Heat in Solids, Oxford Press, Oxford, UK, 510 pp.
- Cengel, Y.A. 1998 *Heat Transfer: A Practical Approach*. New York, McGraw-Hill.
- Chapra and Canale 2006. *Numerical Methods for Engineers*, 5th Ed. New York, McGraw-Hill.
- Chapra, S.C. 1997. *Surface water quality modeling*. New York, McGraw-Hill.
- Chapra, S.C. 2007. Applied Numerical Methods with MATLAB for Engineering and Science, 2nd Ed., WCB/McGraw-Hill, New York, N.Y.
- Chow, V.T., Maidment, D.R., and Mays, L.W. 1988. *Applied Hydrology*. New York, McGraw-Hill, 592 pp.

- Churchill, M.A., Elmore, H.L., and Buckingham, R.A. 1962. The prediction of stream reaeration rates. *J. Sanit. Engrg. Div.*, ASCE, 88(4),1-46.
- Coffaro, G. and A. Sfriso, Simulation model of *Ulva rigida* growth in shallow water of the Lagoon of Venice. *Ecol. Model.*, 102: 55-66, 1997.
- Covar, A. P. 1976. "Selecting the Proper Reaeration Coefficient for Use in Water Quality Models." Presented at the U.S. EPA Conference on Environmental Simulation and Modeling, April 19-22, 1976, Cincinnati, OH.
- Di Toro, D. M. and J. F. Fitzpatrick. 1993. Chesapeake Bay sediment flux model. Tech. Report EL-93-2, U.S. Army Corps of Engineers, Waterways Experiment Station, Vicksburg, Mississippi, 316 pp.
- Di Toro, D.M., Paquin, P.R., Subburamu, K. and Gruber, D.A. 1991. Sediment Oxygen Demand Model: Methane and Ammonia Oxidation. *J. Environ. Eng.*, 116(5):945-986.
- Di Toro, D.M. 1978. Optics of Turbid Estuarine Waters: Approximations and Applications. *Water Res.* 12:1059-1068.
- Di Toro, D.M. 2001. *Sediment Flux Modeling*. Wiley-Interscience, New York, NY.
- Droop, M.R. 1974. The nutrient status of algal cells in continuous culture. *J.Mar.Biol.Assoc. UK* 54:825-855.
- Ecology. 2003. Shade.xls - a tool for estimating shade from riparian vegetation. Washington State Department of Ecology. <http://www.ecy.wa.gov/programs/eap/models/>
- Edinger, J.E., Brady, D.K., and Geyer, J.C. 1974. Heat Exchange and Transport in the Environment. Report No. 14, EPRI Pub. No. EA-74-049-00-3, Electric Power Research Institute, Palo Alto, CA.
- Finlayson-Pitts, B.J. and Pitts, J.N., Jr. 2000. Chemistry of the upper and lower atmosphere : theory, experiments, and applications. Academic Press.
- Finnemore, E.J. and Franzini, J.B. 2002. Fluid Mechanics with Engineering Applications, 10th Ed. New York, McGraw, Hill.
- Geiger, R. 1965. The climate near the ground. Harvard University Press. Cambridge, MA.
- Gordon, N.D, T.A. McMahon, and B.L. Finlayson. 1992. Stream Hydrology, An Introduction for Ecologists. Published by John Wiley and Sons.
- Grigull, U. and Sandner, H. 1984. *Heat Conduction*. Springer-Verlag, New York, NY.
- Hamilton, D.P., and S.G. Schladow, Prediction of water quality in lakes and reservoirs. 1. Model description. *Ecol. Model.*, 96: 91-110, 1997.
- Harbeck, G. E., 1962, A practical field technique for measuring reservoir evaporation utilizing mass-transfer theory. US Geological Survey Professional Paper 272-E, 101-5.
- Harned, H.S., and Hamer, W.J. 1933. "The Ionization Constant of Water." *J. Am., Chem. Soc.* 51:2194.
- Helfrich, K.R., E.E. Adams, A.L. Godbey, and D.R.F. Harleman. 1982. Evaluation of models for predicting evaporative water loss in cooling impoundments. Report CS-2325, Research project 1260-17. Electric Power Research Institute, Palo Alto, CA. March 1982.
- Hellweger, F/L., K.L. Farley, U. Lall, and D.M. DiToro. 2003. Greedy algae reduce arsenate. *Limnol.Oceanogr.* 48(6), 2003, 2275-2288.
- Holland, H.D. 1978. The Chemistry of the Atmosphere and Oceans. Wiley-Interscience, NY, NY.
- Hutchinson, G.E. 1957. *A Treatise on Limnology, Vol. 1, Physics and Chemistry*. Wiley, New York, NY.
- Jobson, H.E. 1977. Bed Conduction Computation for Thermal Models. *J. Hydraul. Div. ASCE.* 103(10):1213-1217.
- Koberg, G.E. 1964. Methods to compute long-wave radiation from the atmosphere and reflected solar radiation from a water surface. US Geological Survey Professional Paper 272-F.
- Kreith, F. and Bohn, M.S. 1986. *Principles of Heat Transfer, 4th Ed.* Harper and Row, New York, NY.

- Laws, E. A. and Chalup, M. S. 1990. A Microalgal Growth Model. *Limnol. Oceanogr.* 35(3):597-608.
- LI-COR, 2003. Radiation Measurement Instruments, LI-COR, Lincoln, NE, 30 pp.
- Likens, G. E., and Johnson, N. M. (1969). "Measurements and analysis of the annual heat budget for sediments of two Wisconsin lakes." *Limnol. Oceanogr.*, 14(1):115-135.
- Mackay, D. and Yeun, A.T.K. 1983. Mass Transfer Coefficient Correlations for Volatilization of Organic Solutes from Water. *Environ. Sci. Technol.* 17:211-233.
- Marciano, J.K. and G.E. Harbeck. 1952. Mass transfer studies in water loss investigation: Lake Hefner studies. Geological Circular 229. U.S. Geological Survey, Washington DC.
- McIntyre, C. D. 1973. Periphyton dynamics in laboratory streams: A simulation model and its implications. *Ecol. Monogr.* 43:399-420.
- Meeus, J. 1999. Astronomical algorithms. Second edition. Willmann-Bell, Inc. Richmond, VA.
- Melching, C.S. and Flores, H.E. 1999. Reaeration Equations from U.S. Geological Survey Database. *J. Environ. Engin.*, 125(5):407-414.
- Mills, A.F. 1992. *Heat Transfer*. Irwin, Homewood, IL.
- Moog, D.B., and Iirka, G.H. 1998. Analysis of reaeration equations using mean multiplicative error. *J. Environ. Engng.*, ASCE, 124(2), 104-110.
- Nakshabandi, G.A. and H. Kohnke. 1965. Thermal conductivity and diffusivity of soils as related to moisture tension and other physical properties. *Agr. Met.* Vol 2.
- O'Connor, D.J., and Dobbins, W.E. 1958. Mechanism of reaeration in natural streams. *Trans. ASCE*, 123, 641-684.
- Owens, M., Edwards, R.W., and Gibbs, J.W. 1964. Some reaeration studies in streams. *Int. J. Air and Water Pollution*, 8, 469-486.
- Plummer, L.N. and Busenberg, E. 1982. The Solubilities of Calcite, Aragonite and Vaterite in CO₂-H₂O Solutions Between 0 and 90 °C, and an Evaluation of the Aqueous Model for the System CaCO₃-CO₂-H₂O. *Geochim. Cosmochim.* 46:1011-1040.
- Raudkivi, A. I. 1979. *Hydrology*. Pergamon, Oxford, England.
- Redfield, A.C., Ketchum, B.H. and Richards, F.A. 1963. The Influence of Organisms on the Composition of Seawater, in *The Sea*, M.N. Hill, ed. Vol. 2, pp. 27-46, Wiley-Interscience, NY.
- Riley, G.A. 1956. Oceanography of Long Island Sound 1952-1954. II. Physical Oceanography, Bull. Bingham Oceanog. Collection 15, pp. 15-16.
- Rosgen, D. 1996. Applied river morphology. Wildland Hydrology publishers. Pagosa Springs, CO.
- Rutherford, J.C., Scarsbrook, M.R. and Broekhuizen, N. 2000. Grazer Control of Stream Algae: Modeling Temperature and Flood Effects. *J. Environ. Eng.* 126(4):331-339.
- Ryan, P.J. and D.R.F. Harleman. 1971. Prediction of the annual cycle of temperature changes in a stratified lake or reservoir. Mathematical model and user's manual. Ralph M. Parsons Laboratory Report No. 137. Massachusetts Institute of Technology. Cambridge, MA.
- Ryan, P.J. and K.D. Stolzenbach. 1972. Engineering aspects of heat disposal from power generation, (D.R.F. Harleman, ed.). R.M. Parson Laboratory for Water Resources and Hydrodynamics, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA
- Schwarzenbach, R.P., Gschwend, P.M., and Imboden, D.M. 1993. Environmental Organic Chemistry. Wiley-Interscience, 681 pp.
- Shanahan, P. 1984. Water temperature modeling: a practical guide. In: Proceedings of stormwater and water quality model users group meeting, April 12-13, 1984. USEPA, EPA-600/9-85-003. (users.rcn.com/shanahan.ma.ultranet/TempModeling.pdf).
- Smith, E.L. 1936. Photosynthesis in Relation to Light and Carbon Dioxide. *Proc. Natl. Acad. Sci.* 22:504-511.

- Steele, J.H. 1962. Environmental Control of Photosynthesis in the Sea. *Limnol. Oceanogr.* 7:137-150.
- Stumm, W. and Morgan, J.J. 1996. *Aquatic Chemistry*, 3rd Ed., New York, Wiley-Interscience, 1022 pp.
- Szeicz, G. 1974. Solar radiation for plant growth. *J. Appl. Ecol.* 11:617-636.
- Thackston, E. L., and Dawson, J.W. III. 2001. Recalibration of a Reaeration Equation. *J. Environ. Engrg.*, 127(4):317-320.
- Thackston, E. L., and Krenkel, P. A. 1969. Reaeration-prediction in natural streams. *J. Sanit. Engrg. Div.*, ASCE, 95(1), 65-94.
- Tsivoglou, E. C., and Neal, L.A. 1976. Tracer Measurement of Reaeration. III. Predicting the Reaeration Capacity of Inland Streams. *Journal of the Water Pollution Control Federation*, 48(12):2669-2689.
- Thomann, R.V. and Mueller, J.A. 1987. *Principles of Surface Water Quality Modeling and Control*. New York, Harper-Collins.
- TVA, 1972. Heat and mass transfer between a water surface and the atmosphere. *Water Resources Research*, Laboratory Report No. 14. Engineering Laboratory, Division of Water Control Planning, Tennessee Valley Authority, Norris TN.
- Wanninkhof, R., Ledwell, I. R., and Crusius, I. 1991. "Gas Transfer Velocities on Lakes Measured with Sulfur Hexafluoride." In *Symposium Volume of the Second International Conference on Gas Transfer at Water Surfaces*, S.C. Wilhelms and I.S. Gulliver, eds., Minneapolis, MN .
- Wood, K.G. 1974. Carbon Dioxide Diffusivity Across the Air-Water Interface. *Archiv für Hydrobiol.* 73(1):57-69.

APPENDIX A: NOMENCLATURE

Symbol	Definition	Units
$A_{acres,i}$	surface area of element i	acres
$[CO_2]_s$	saturation concentration of carbon dioxide	mole/L
$[CO_3^{2-}]$	carbonate ion	mole/L
$[H^+]$	hydronium ion	mole/L
$[H_2CO_3^*]$	sum of dissolved carbon dioxide and carbonic acid	mole/L
$[HCO_3^-]$	bicarbonate ion	mole/L
$[OH^-]$	hydroxyl ion	mole/L
a	velocity rating curve coefficient	dimensionless
a''	mean atmospheric transmission coefficient after scattering and absorption	Dimensionless
a'	mean atmospheric transmission coefficient	Dimensionless
a_1	atmospheric molecular scattering coefficient for radiation transmission	dimensionless
A_a	atmospheric long-wave radiation coefficient	dimensionless
A_b	atmospheric long-wave radiation coefficient	mmHg ^{-0.5} or mb ^{-0.5}
a_b	bottom algae	mgA/m ²
A_c	cross-sectional area	m ²
a_d	coefficient correcting dam reaeration for water quality	dimensionless
Alk	alkalinity	eq L ⁻¹ or mgCaCO ₃ /L
a_p	phytoplankton	mgC/L
a_p	phytoplankton concentration	mgA/m ³
a_t	atmospheric attenuation	dimensionless
a_{tc}	atmospheric transmission coefficient	dimensionless
B	average element width	m
b	velocity rating curve exponent	dimensionless
B_0	bottom width	m
b_d	coefficient correcting dam reaeration for dam type	dimensionless
c_1	Bowen's coefficient	0.47 mmHg ^{°C}
c_f	fast reacting CBOD	mgO ₂ /L
c_f	fast CBOD in the overlying water	gO ₂ /m ³
$C_{gb}(T)$	temperature-dependent maximum photosynthesis rate	mgA/(m ² d)
$CH_{4,1}$	methane concentration in the aerobic sediment layer	gO ₂ /m ³
C_L	fraction of sky covered with clouds	dimensionless
$c_{nps,i,j}$	j th non-point source concentration for element i	°C
C_p	specific heat of water	cal/(g °C)
$c_{ps,i,j}$	j th point source concentration for element i	°C
c_s	slowly reacting CBOD	mgO ₂ /L
C_s	saturation concentration of methane	mgO ₂ /L
$CSOD$	the amount of oxygen demand generated by methane oxidation	gO ₂ /m ² /d
c_T	total inorganic carbon	mole L ⁻¹
$c'_{T,i-1}$	concentration of inorganic carbon entering element below a dam	mgO ₂ /L

d	dust attenuation coefficient	dimensionless
D_d	pore water diffusion coefficient	m^2/d
D_p	diffusion coefficient for bioturbation	m^2/d
E'_i	bulk dispersion coefficient between elements i and $i + 1$	m^3/d
e_{air}	air vapor pressure	mm Hg
$elev$	elevation above sea level	m
E_n	numerical dispersion	m^2/d
$E_{p,i}$	longitudinal dispersion between elements i and $i + 1$	m^2/s
$eqtime$	equation of time: the difference between mean solar time and true solar time when located on the reference longitude of the time zone considered	minutes
e_s	saturation vapor pressure at water surface	mmHg
f	photoperiod	fraction of day
f_{dai}	fraction of ammonium in dissolved form in sediment layer i	dimensionless
f_{dpi}	fraction of inorganic phosphorus in dissolved form in sediment layer i	dimensionless
F_{LP}	phytoplankton growth attenuation due to light	dimensionless
F_{oxc}	attenuation of CBOD oxidation due to low oxygen	dimensionless
F_{oxdn}	enhancement of denitrification at low oxygen concentration	dimensionless
F_{oxna}	attenuation due to low oxygen	dimensionless
F_{oxrb}	attenuation due to low oxygen	dimensionless
F_{oxrp}	attenuation due to low oxygen	dimensionless
f_{pai}	fraction of ammonium in particulate form in sediment layer i	dimensionless
f_{ppi}	fraction of inorganic phosphorus in particulate form in sediment layer i	dimensionless
F_u	fraction unionized ammonia	
g	acceleration due to gravity	$= 9.81 \text{ m/s}^2$
gX	mass of element X	g
H	water depth	m
H	water depth	m
H_d	drop in water elevation for a dam	m
H_i	thickness of sediment layer i	m
H_{sed}	sediment thickness	cm
$I(0)$	solar radiation at water surface	$\text{cal}/\text{cm}^2/\text{d}$
I_0	extraterrestrial radiation	$\text{cal}/\text{cm}^2/\text{d}$
J_{an}	net atmospheric longwave radiation flux	$\text{cal}/(\text{cm}^2 \text{ d})$
J_{br}	longwave back radiation flux from water	$\text{cal}/(\text{cm}^2 \text{ d})$
J_c	conduction flux	$\text{cal}/(\text{cm}^2 \text{ d})$
$J_{C,G1}$	flux of labile dissolved carbon	$\text{gO}_2/\text{m}^2/\text{d}$
J_{CH4}	flux of methane from sediments to the overlying water	$\text{gO}_2/\text{m}^2/\text{d}$
$J_{CH4,d}$	flux of dissolved methane generated in anaerobic sediments corrected for methane gas formation	$\text{gO}_2/\text{m}^2/\text{d}$
$J_{CH4,T}$	carbon diagenesis flux corrected for denitrification	$\text{gO}_2/\text{m}^2/\text{d}$
J_e	evaporation flux	$\text{cal}/(\text{cm}^2 \text{ d})$
J_a	air-water heat flux	$\text{cal}/(\text{cm}^2 \text{ d})$

J_N	nitrogen diagenesis flux	gN/m ² /d
$J_{O_2,dn}$	oxygen equivalents consumed during denitrification	gO ₂ /m ² /d
J_P	phosphorus diagenesis flux	gP/m ² /d
$J_{POC,G1}$	the flux of labile POC delivered to the anaerobic sediment layer	gO ₂ /m ² /d
J_{POM}	the downward flux of POM	gD m ⁻² d ⁻¹
J_{si}	sediment-water heat flux	cal/(cm ² d)
J_{sn}	net solar shortwave radiation flux	cal/(cm ² d)
$k(T)$	temperature dependent first-order reaction rate	/d
K_1	acidity constant for dissociation of carbonic acid	
K_2	acidity constant for dissociation of bicarbonate	
K_a	equilibrium coefficient for ammonium dissociation	
$k_a(T)$	temperature-dependent oxygen reaeration coefficient	/d
$k_{ac}(T)$	temperature-dependent carbon dioxide reaeration coefficient	/d
$k_{db}(T)$	temperature-dependent bottom algae death rate	/d
$k_{dc}(T)$	temperature-dependent fast CBOD oxidation rate	/d
$k_{dn}(T)$	temperature-dependent denitrification rate	/d
$k_{dp}(T)$	temperature-dependent phytoplankton death rate	/d
$k_{dt}(T)$	temperature-dependent detritus dissolution rate	/d
$k_{dX}(T)$	temperature-dependent pathogen die-off rate	/d
k_e	light extinction coefficient	/m ¹
k_{eb}	a background coefficient accounting for extinction due to water and color	/m
$k_{gp}(T)$	maximum photosynthesis rate at temperature T	/d
K_H	Henry's constant	mole/(L atm)
$k_{hc}(T)$	temperature-dependent slow CBOD hydrolysis rate	/d
$k_{hn}(T)$	temperature-dependent organic nitrogen hydrolysis rate	/d
k_{hnxb}	preference coefficient of bottom algae for ammonium	mgN/m ³
k_{hnxp}	preference coefficient of phytoplankton for ammonium	mgN/m ³
$k_{hp}(T)$	temperature-dependent organic phosphorus hydrolysis rate	/d
K_{L12}	pore water diffusion mass transfer coefficient	m/d
K_{Lb}	bottom algae light parameter	
K_{Lp}	phytoplankton light parameter	ly/d
$K_{M,Dp}$	oxygen half-saturation constant for bioturbation	gO ₂ /m ³
$k_{na}(T)$	temperature-dependent nitrification rate for ammonia nitrogen	/d
K_{NH_4}	ammonium half-saturation constant	gN/m ³
K_{NH_4,O_2}	oxygen half-saturation constant	mgO ₂ /L
$k_{POC,G1}$	the mineralization rate of labile POC	d ⁻¹
$k_{rb}(T)$	temperature-dependent bottom algae respiration rate	/d
$k_{rp}(T)$	temperature-dependent phytoplankton respiration/excretion rate	/d
k_{sNb}	nitrogen half-saturation constant for bottom algae	μgN/L
k_{sNp}	nitrogen half-saturation constant for phytoplankton	μgN/L
K_{soct}	parameter for oxygen dependency of fast CBOD oxidation	

K_{sodn}	parameter for oxygen dependency of denitrification	
K_{sona}	parameter for oxygen dependency of nitrification	
k_{sPb}	phosphorus half-saturation constant for bottom algae	$\mu\text{gP/L}$
k_{sPp}	phosphorus half-saturation constant for phytoplankton	$\mu\text{gP/L}$
K_w	acidity constant for dissociation of water	
L_{at}	latitude	radians
L_{lm}	longitude of local meridian	degrees
$localTime$	local standard time	minutes
L_{sm}	longitude of standard meridian	degrees
m	optical air mass	dimensionless
mgY	mass of element Y	mg
m_i	inorganic suspended solids	mgD/L
m_i	the solids concentration in sediment layer i	gD/m^3
m_o	detritus	mgD/L
m_o	detritus concentration	gD/m^3
n	Manning roughness coefficient	
n_a	the ammonium concentration in the overlying water	mgN/m^3
n_{au}	unionized ammonia nitrogen	mgN/m^3
n_{fac}	atmospheric turbidity factor	dimensionless
$NH_{4,i}$	the concentration of total ammonium in sediment layer i	gN/m^3
n_n	nitrate concentration in the overlying water	mgN/m^3
n_o	organic nitrogen	mgN/m^3
$NO_{3,i}$	nitrate concentration in layer i	gN/m^3
$npai$	total number of non-point withdrawals outflows from element i	dimensionless
$npsi$	total number of non-point sources inflows to element i	dimensionless
$NSOD$	the amount of oxygen demand generated by nitrification	$\text{gO}_2/\text{m}^2/\text{d}$
o	dissolved oxygen	mgO_2/L
o	the dissolved oxygen concentration in the overlying water	gO_2/m^3
o'_{i-1}	oxygen concentration entering an element below a dam	mgO_2/L
o_{crit}	critical oxygen concentration for sediment phosphorus sorption	gO_2/m^3
$o_s(T, \text{elev})$	saturation concentration of oxygen at temperature, T , and elevation above sea level, elev	mgO_2/L
P	wetted perimeter	m
P_{ab}	preference for ammonium as a nitrogen source for bottom algae	dimensionless
pai	total number of point withdrawals from element i	dimensionless
P_{ap}	preference for ammonium as a nitrogen source for phytoplankton	dimensionless
$PAR(z)$	photosynthetically available radiation (PAR) at depth z below water surface	ly/d
p_{atm}	atmospheric pressure	mm Hg
p_{CO_2}	atmospheric partial pressure of carbon dioxide	atm
p_i	inorganic phosphorus	$\mu\text{gP/L}$
p_i	the inorganic phosphorus in the overlying water	mgP/m^3
p_o	organic phosphorus	$\mu\text{gP/L}$

$PO_{4,i}$	the concentration of total inorganic phosphorus in sediment layer i	gP/m^3
$POC_{2,G1}$	the concentration of the labile fraction of POC in the anaerobic sediment layer	gO_2/m^3
POC_R	reference G1 concentration for bioturbation	gC/m^3
psi	total number of point sources to element i	dimensionless
p_{wc}	mean daily atmospheric precipitable water content	
Q	flow	m^3/s or m^3/d
$Q_{out,i}$	total outflow from element due to point and nonpoint withdrawals	m^3/d
Q_i	outflow from element i into element $i + 1$	m^3/d
$Q_{in,i}$	total inflow into element from point and nonpoint sources	m^3/d
$Q_{npa,i,j}$	j th non-point withdrawal outflow from element i	m^3/d
$Q_{nps,i,j}$	j th non-point source inflow to element i	m^3/d
$Q_{pa,i,j}$	j th point withdrawal outflow from element i	m^3/d
$Q_{ps,i,j}$	j th point source inflow to element i	m^3/d
r	normalized radius of earth's orbit (i.e., ratio of actual earth-sun distance to mean earth-sun distance)	dimensionless
r_{cndn}	ratio of oxygen equivalents lost per nitrate nitrogen that is denitrified	gO_2/gN
r_d	ratio of deficit above and below dam	dimensionless
r_{da}	the ratio of dry weight to chlorophyll a	gD/mgA
R_L	longwave reflection coefficient	dimensionless
r_{oc}	ratio of oxygen consumed per organic carbon oxidized to carbon dioxide	gO_2/gC
r_{oca}	ratio of oxygen generated per organic carbon produced by photosynthesis when nitrate is taken up	gO_2/gC
r_{ocn}	ratio of oxygen generated per organic carbon produced by photosynthesis when ammonia is taken up	gO_2/gC
r_{on}	the ratio of oxygen to nitrogen consumed during nitrification	$= 4.57 \text{ gO}_2/\text{gN}$
r_{on}'	ratio of oxygen consumed per ammonia nitrogen converted by nitrification/denitrification to nitrogen gas	gO_2/gN
r_{on}''	the ratio of oxygen to nitrogen consumed for total conversion of ammonium to nitrogen gas via nitrification/denitrification	$= 1.714 \text{ gO}_2/\text{gN}$
R_s	albedo or reflectivity (fraction of solar radiation that is reflected)	dimensionless
S	channel slope	dimensionless
s	chloride	mgCl/L
s	mass transfer coefficient between the water and the aerobic sediments	m/d
S_0	bottom slope	m/m
$S_{b,i}$	sources and sinks of constituent due to reactions for bottom algae	$\text{mgA/m}^2/\text{d}$
S_i	sources and sinks of constituent due to reactions and mass transfer mechanisms for water constituents	$\text{g/m}^3/\text{d}$ or $\text{mg/m}^3/\text{d}$
SOD	the sediment oxygen demand	$\text{gO}_2/\text{m}^2/\text{d}$

SOD_s	the supplemental sediment oxygen demand	$\text{gO}_2/\text{m}^2/\text{d}$
SOD_t	total sediment oxygen demand = $SOD + SOD_s$	$\text{gO}_2/\text{m}^2/\text{d}$
s_s	channel side slope	m/m
t	time	d
T	water temperature	$^{\circ}\text{C}$
T	water temperature	$^{\circ}\text{C}$
$T_{w,f}$	water temperature	$^{\circ}\text{F}$
T_a	absolute temperature	K
T_{air}	air temperature	$^{\circ}\text{C}$
$T_{air,f}$	air temperature	$^{\circ}\text{F}$
T_d	dew-point temperature	$^{\circ}\text{C}$
T_i	temperature in element i	$^{\circ}\text{C}$
$timezone$	time zone indicates local time zone in relation to Greenwich Mean Time (GMT) (negative in western hemisphere)	hours
$T_{nps,i,j}$	j th non-point source temperature for element i	$^{\circ}\text{C}$
$T_{ps,i,j}$	j th point source temperature for element i	$^{\circ}\text{C}$
$trueSolarTime$	time determined from actual position of the sun in the sky	minutes
$T_{s,i}$	temperature of bottom sediment	$^{\circ}\text{C}$
t_{sr}	time of sunrise	Hr
t_{ss}	time of sunset	Hr
T_{std}	standard time	Hr
$t_{t,i}$	travel time from headwater to end of element i	D
U	mean velocity	m/s
U_i^*	shear velocity	m/s
U_w	wind speed	m/s
$U_{w,mph}$	wind speed	mph
$U_{w,z}$	wind speed at height z_w above water surface	m/s
v_a	phytoplankton settling velocity	m/d
v_a	phytoplankton settling velocity	m/d
v_{dt}	detritus settling velocity	m/d
v_{dt}	detritus settling velocity	m/d
v_i	inorganic suspended solids settling velocity	m/d
V_i	volume of i^{th} element	m^3
v_p	pathogen settling velocity	m/d
v_{pc}	non-living particulate organic carbon settling velocity	m/d
W_0	solar constant	$2851 \text{ cal}/\text{cm}^2/\text{d}$
w_2	the burial velocity	m/d
$W_{h,i}$	net heat load from point and non-point sources into element i	cal/d
W_i	external loading of constituent to element i	g/d or mg/d
X	pathogen	cfu/100 mL
z_w	height above water for wind speed measurements	m

Greek:

Symbol	Definition	Units
α	depth rating curve coefficient	dimensionless
α	sun's altitude	radians
α_d	sun's altitude	degrees
α_s	sediment thermal diffusivity	cm ² /s
α_0	fraction of total inorganic carbon in carbon dioxide	dimensionless
α_1	fraction of total inorganic carbon in bicarbonate	dimensionless
α_2	fraction of total inorganic carbon in carbonate	dimensionless
α_i	effect of inorganic suspended solids on light attenuation	L/mgD/m
α_o	effect of particulate organic matter on light attenuation	L/mgD/m
α_p	linear effect of chlorophyll on light attenuation	L/ μ gA/m
α_{pn}	non-linear effect of chlorophyll on light attenuation	(L/ μ gA) ^{2/3} /m
α_{path}	pathogen light efficiency factor	dimensionless
β	depth rating curve exponent	dimensionless
δ	solar declination	radians
$\Delta\pi_{PO4,1}$	factor that increases the aerobic sediment layer partition coefficient relative to the anaerobic coefficient	dimensionless
$\Delta\theta_v$	virtual temperature difference between the water and air	°F
Δt_s	difference between standard and local civic time	hr
Δx_i	length of i th element	m
ε	emissivity of water	dimensionless
ε_{clear}	emissivity of longwave radiation from the sky with no clouds	0-1
ε_{sky}	emissivity of longwave radiation from the sky with clouds	0-1
ε_a	estimated error	%
ϕ_{Lb}	bottom algae light attenuation	0-1
ϕ_{Lp}	phytoplankton light attenuation	0-1
ϕ_{Nb}	bottom algae nutrient attenuation factor	0-1
ϕ_{Np}	phytoplankton nutrient attenuation factor	0-1
μ_p	phytoplankton photosynthesis rate	/d
ρ	density of water	g/cm ³
σ	Stefan-Boltzmann constant	11.7 $\times 10^{-8}$ cal/(cm ² d K ⁴)
τ	local hour angle of sun	radians
τ_i	residence time of i^{th} element	d
θ	temperature coefficient for zero and first-order reactions	dimensionless
θ_{am}	elevation adjusted optical air mass	
ω_{12}	the bioturbation mass transfer coefficient between the sediment layers	m/d
π_{ai}	the partition coefficient for ammonium in sediment layer i	m ³ /gD
π_{pi}	the partition coefficient for inorganic phosphorus in sediment layer i	m ³ /gD
θ_{CH4}	temperature correction factor for sediment methane oxidation	dimensionless

θ_{Dd}	temperature coefficient for porewater diffusion	dimensionless
θ_{Dp}	temperature coefficient for bioturbation diffusion	dimensionless
θ_{NH4}	temperature correction factor for sediment nitrification	dimensionless
θ_{NO3}	sediment denitrification temperature correction factor	dimensionless
$\theta_{POC,G1}$	temperature correction factor for labile POC mineralization	dimensionless
$\kappa_{CH4,1}$	the reaction velocity for methane oxidation in the aerobic sediments	m/d
$\kappa_{NH4,1}$	the reaction velocity for nitrification in the aerobic sediments	m/d
$\kappa_{NO3,i}$	denitrification reaction velocity sediment layer i	m/d

APPENDIX B: SOLAR POSITION, SUNRISE, AND SUNSET CALCULATIONS

The sunrise/sunset and solar position functions are a VBA translation of NOAA's sunrise/sunset calculator and NOAA's solar position calculator at the following web pages:

- <http://www.srrb.noaa.gov/highlights/sunrise/sunrise.html>
- <http://www.srrb.noaa.gov/highlights/sunrise/azel.html>

The calculations in the NOAA Sunrise/Sunset and Solar Position Calculators are based on equations from *Astronomical Algorithms*, by Jean Meeus. The sunrise and sunset results have been verified by NOAA to be accurate to within a minute for locations between $\pm 72^\circ$ latitude, and within 10 minutes outside of those latitudes.

Five main functions are included for use from Excel worksheets or VBA programs:

- sunrise (lat, lon, year, month, day, timezone, dlstime) calculates the local time of sunrise for a location and date
- solarnoon (lat, lon, year, month, day, timezone, dlstime) calculates the local time of solar noon for a location and date (the time when the sun crosses the meridian)
- sunset (lat, lon, year, month, day, timezone, dlstime) calculates the local time of sunset for a location and date
- solarazimuth (lat, lon, year, month, day, hour, minute, second, timezone, dlstime) calculates the solar azimuth for a location, date, and time (degrees clockwise from north to the point on the horizon directly below the sun)
- solarelevation (lat, lon, year, month, day, hour, minute, second, timezone, dlstime) calculates the solar elevation for a location, date, and time (degrees vertically from horizon to the sun)

A subroutine is also provided that calculates solar azimuth (az), solar elevation (el):

- solarposition (lat, lon, year, month, day, hour, minute, second, timezone, dlstime, az, el, earthRadiusVector)

The sign convention for the main functions and subroutine is:

- positive latitude decimal degrees for northern hemisphere
- negative longitude degrees for western hemisphere
- negative time zone hours for western hemisphere

The Excel/VBA functions and subroutines for solar position, sunrise, and sunset times are as follows:

```
Option Explicit

Function radToDeg(angleRad)
'// Convert radian angle to degrees

    radToDeg = (180# * angleRad / Application.WorksheetFunction.Pi())
End Function

Function degToRad(angleDeg)
'// Convert degree angle to radians

    degToRad = (Application.WorksheetFunction.Pi() * angleDeg / 180#)
End Function

Function calcJD(year, month, day)

'*****
'* Name:      calcJD
'* Type:      Function
'* Purpose:   Julian day from calendar day
'* Arguments:
'*   year : 4 digit year
```

```

'* month: January = 1
'* day : 1 - 31
'* Return value:
'* The Julian day corresponding to the date
'* Note:
'* Number is returned for start of day. Fractional days should be
'* added later.
'*****/

Dim A As Double, B As Double, jd As Double

    If (month <= 2) Then
        year = year - 1
        month = month + 12
    End If

    A = Application.WorksheetFunction.Floor(year / 100, 1)
    B = 2 - A + Application.WorksheetFunction.Floor(A / 4, 1)

    jd = Application.WorksheetFunction.Floor(365.25 * (year + 4716), 1) + _
        Application.WorksheetFunction.Floor(30.6001 * (month + 1), 1) + day + B - 1524.5
    calcJD = jd

'gp put the year and month back where they belong
    If month = 13 Then
        month = 1
        year = year + 1
    End If
    If month = 14 Then
        month = 2
        year = year + 1
    End If

End Function

Function calcTimeJulianCent(jd)

'*****/
'* Name:    calcTimeJulianCent
'* Type:    Function
'* Purpose: convert Julian Day to centuries since J2000.0.
'* Arguments:
'*   jd : the Julian Day to convert
'* Return value:
'*   the T value corresponding to the Julian Day
'*****/

Dim t As Double

    t = (jd - 2451545#) / 36525#
    calcTimeJulianCent = t

End Function

Function calcJDFromJulianCent(t)

'*****/
'* Name:    calcJDFromJulianCent
'* Type:    Function
'* Purpose: convert centuries since J2000.0 to Julian Day.
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   the Julian Day corresponding to the t value
'*****/

Dim jd As Double

    jd = t * 36525# + 2451545#
    calcJDFromJulianCent = jd

End Function

Function calcGeomMeanLongSun(t)

'*****/
'* Name:    calGeomMeanLongSun
'* Type:    Function
'* Purpose: calculate the Geometric Mean Longitude of the Sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0

```



```

'* Return value:
'* the Geometric Mean Longitude of the Sun in degrees
'*****

Dim l0 As Double

    l0 = 280.46646 + t * (36000.76983 + 0.0003032 * t)
    Do
        If (l0 <= 360) And (l0 >= 0) Then Exit Do
        If l0 > 360 Then l0 = l0 - 360
        If l0 < 0 Then l0 = l0 + 360
    Loop

    calcGeomMeanLongSun = l0

End Function

Function calcGeomMeanAnomalySun(t)

'*****
'* Name:      calcGeomAnomalySun
'* Type:      Function
'* Purpose: calculate the Geometric Mean Anomaly of the Sun
'* Arguments:
'* t : number of Julian centuries since J2000.0
'* Return value:
'* the Geometric Mean Anomaly of the Sun in degrees
'*****

Dim m As Double

    m = 357.52911 + t * (35999.05029 - 0.0001537 * t)
    calcGeomMeanAnomalySun = m

End Function

Function calcEccentricityEarthOrbit(t)

'*****
'* Name:      calcEccentricityEarthOrbit
'* Type:      Function
'* Purpose: calculate the eccentricity of earth's orbit
'* Arguments:
'* t : number of Julian centuries since J2000.0
'* Return value:
'* the unitless eccentricity
'*****

Dim e As Double

    e = 0.016708634 - t * (0.000042037 + 0.0000001267 * t)
    calcEccentricityEarthOrbit = e

End Function

Function calcSunEqOfCenter(t)

'*****
'* Name:      calcSunEqOfCenter
'* Type:      Function
'* Purpose: calculate the equation of center for the sun
'* Arguments:
'* t : number of Julian centuries since J2000.0
'* Return value:
'* in degrees
'*****

Dim m As Double, mrad As Double, sinm As Double, sin2m As Double, sin3m As Double
Dim c As Double

    m = calcGeomMeanAnomalySun(t)

    mrad = degToRad(m)
    sinm = Sin(mrad)
    sin2m = Sin(mrad + mrad)
    sin3m = Sin(mrad + mrad + mrad)

    c = sinm * (1.914602 - t * (0.004817 + 0.000014 * t))
        + sin2m * (0.019993 - 0.000101 * t) + sin3m * 0.000289

    calcSunEqOfCenter = c

```

```

End Function

Function calcSunTrueLong(t)

'*****/
'* Name:    calcSunTrueLong
'* Type:    Function
'* Purpose: calculate the true longitude of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's true longitude in degrees
'*****/

Dim l0 As Double, c As Double, O As Double

    l0 = calcGeomMeanLongSun(t)
    c = calcSunEqOfCenter(t)

    O = l0 + c
    calcSunTrueLong = O

End Function

Function calcSunTrueAnomaly(t)

'*****/
'* Name:    calcSunTrueAnomaly (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the true anomaly of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's true anomaly in degrees
'*****/

Dim m As Double, c As Double, v As Double

    m = calcGeomMeanAnomalySun(t)
    c = calcSunEqOfCenter(t)

    v = m + c
    calcSunTrueAnomaly = v

End Function

Function calcSunRadVector(t)

'*****/
'* Name:    calcSunRadVector (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the distance to the sun in AU
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun radius vector in AUs
'*****/

Dim v As Double, e As Double, r As Double

    v = calcSunTrueAnomaly(t)
    e = calcEccentricityEarthOrbit(t)

    r = (1.000001018 * (1 - e * e)) / (1 + e * Cos(degToRad(v)))
    calcSunRadVector = r

End Function

Function calcSunApparentLong(t)

'*****/
'* Name:    calcSunApparentLong (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the apparent longitude of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's apparent longitude in degrees
'*****/

```

```

Dim O As Double, omega As Double, lambda As Double

    O = calcSunTrueLong(t)

    omega = 125.04 - 1934.136 * t
    lambda = O - 0.00569 - 0.00478 * Sin(degToRad(omega))
    calcSunApparentLong = lambda

End Function

Function calcMeanObliquityOfEcliptic(t)

'*****/
'* Name:    calcMeanObliquityOfEcliptic
'* Type:    Function
'* Purpose: calculate the mean obliquity of the ecliptic
'* Arguments:
'*    t : number of Julian centuries since J2000.0
'* Return value:
'*    mean obliquity in degrees
'*****/

Dim seconds As Double, e0 As Double

    seconds = 21.448 - t * (46.815 + t * (0.00059 - t * (0.001813)))
    e0 = 23# + (26# + (seconds / 60#)) / 60#
    calcMeanObliquityOfEcliptic = e0

End Function

Function calcObliquityCorrection(t)

'*****/
'* Name:    calcObliquityCorrection
'* Type:    Function
'* Purpose: calculate the corrected obliquity of the ecliptic
'* Arguments:
'*    t : number of Julian centuries since J2000.0
'* Return value:
'*    corrected obliquity in degrees
'*****/

Dim e0 As Double, omega As Double, e As Double

    e0 = calcMeanObliquityOfEcliptic(t)

    omega = 125.04 - 1934.136 * t
    e = e0 + 0.00256 * Cos(degToRad(omega))
    calcObliquityCorrection = e

End Function

Function calcSunRtAscension(t)

'*****/
'* Name:    calcSunRtAscension (not used by sunrise, solarnoon, sunset)
'* Type:    Function
'* Purpose: calculate the right ascension of the sun
'* Arguments:
'*    t : number of Julian centuries since J2000.0
'* Return value:
'*    sun's right ascension in degrees
'*****/

Dim e As Double, lambda As Double, tananum As Double, tanadenom As Double
Dim alpha As Double

    e = calcObliquityCorrection(t)
    lambda = calcSunApparentLong(t)

    tananum = (Cos(degToRad(e)) * Sin(degToRad(lambda)))
    tanadenom = (Cos(degToRad(lambda)))

'original NOAA code using javascript Math.Atan2(y,x) convention:
'    var alpha = radToDeg(Math.atan2(tananum, tanadenom));
'    alpha = radToDeg(Application.WorksheetFunction.Atan2(tananum, tanadenom))

'translated using Excel VBA Application.WorksheetFunction.Atan2(x,y) convention:
    alpha = radToDeg(Application.WorksheetFunction.Atan2(tanadenom, tananum))

    calcSunRtAscension = alpha

```

```

End Function

Function calcSunDeclination(t)

'*****/

'* Name:      calcSunDeclination
'* Type:      Function
'* Purpose: calculate the declination of the sun
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   sun's declination in degrees
'*****/

Dim e As Double, lambda As Double, sint As Double, theta As Double

    e = calcObliquityCorrection(t)
    lambda = calcSunApparentLong(t)

    sint = Sin(degToRad(e)) * Sin(degToRad(lambda))
    theta = radToDeg(Application.WorksheetFunction.Asin(sint))
    calcSunDeclination = theta

End Function

Function calcEquationOfTime(t)

'*****/

'* Name:      calcEquationOfTime
'* Type:      Function
'* Purpose: calculate the difference between true solar time and mean
'*           solar time
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'* Return value:
'*   equation of time in minutes of time
'*****/

Dim epsilon As Double, l0 As Double, e As Double, m As Double
Dim y As Double, sin2l0 As Double, sinm As Double
Dim cos2l0 As Double, sin4l0 As Double, sin2m As Double, Etime As Double

    epsilon = calcObliquityCorrection(t)
    l0 = calcGeomMeanLongSun(t)
    e = calcEccentricityEarthOrbit(t)
    m = calcGeomMeanAnomalySun(t)

    y = Tan(degToRad(epsilon)) / 2#
    y = y ^ 2

    sin2l0 = Sin(2# * degToRad(l0))
    sinm = Sin(degToRad(m))
    cos2l0 = Cos(2# * degToRad(l0))
    sin4l0 = Sin(4# * degToRad(l0))
    sin2m = Sin(2# * degToRad(m))

    Etime = y * sin2l0 - 2# * e * sinm + 4# * e * y * sinm * cos2l0 _
            - 0.5 * y * y * sin4l0 - 1.25 * e * e * sin2m

    calcEquationOfTime = radToDeg(Etime) * 4#

End Function

Function calcHourAngleSunrise(lat, SolarDec)

'*****/

'* Name:      calcHourAngleSunrise
'* Type:      Function
'* Purpose: calculate the hour angle of the sun at sunrise for the
'*           latitude
'* Arguments:
'*   lat : latitude of observer in degrees
'*   solarDec : declination angle of sun in degrees
'* Return value:
'*   hour angle of sunrise in radians
'*****/

Dim latrad As Double, sdRad As Double, HAarg As Double, HA As Double

```

```

        latrad = degToRad(lat)
        sdRad = degToRad(SolarDec)

        HAarg = (Cos(degToRad(90.833)) / (Cos(latrad) * Cos(sdRad)) - Tan(latrad) * Tan(sdRad))

        HA = (Application.WorksheetFunction.Acos(Cos(degToRad(90.833)) -
            / (Cos(latrad) * Cos(sdRad)) - Tan(latrad) * Tan(sdRad))) -

        calcHourAngleSunrise = HA

End Function

Function calcHourAngleSunset(lat, SolarDec)

'*****/
'* Name:      calcHourAngleSunset
'* Type:      Function
'* Purpose: calculate the hour angle of the sun at sunset for the
'*           latitude
'* Arguments:
'*   lat : latitude of observer in degrees
'*   solarDec : declination angle of sun in degrees
'* Return value:
'*   hour angle of sunset in radians
'*****/

Dim latrad As Double, sdRad As Double, HAarg As Double, HA As Double

        latrad = degToRad(lat)
        sdRad = degToRad(SolarDec)

        HAarg = (Cos(degToRad(90.833)) / (Cos(latrad) * Cos(sdRad)) - Tan(latrad) * Tan(sdRad))

        HA = (Application.WorksheetFunction.Acos(Cos(degToRad(90.833)) -
            / (Cos(latrad) * Cos(sdRad)) - Tan(latrad) * Tan(sdRad))) -

        calcHourAngleSunset = -HA

End Function

Function calcSunriseUTC(jd, Latitude, longitude)

'*****/
'* Name:      calcSunriseUTC
'* Type:      Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of sunrise
'*           for the given day at the given location on earth
'* Arguments:
'*   JD : julian day
'*   latitude : latitude of observer in degrees
'*   longitude : longitude of observer in degrees
'* Return value:
'*   time in minutes from zero Z
'*****/

Dim t As Double, eqtime As Double, SolarDec As Double, hourangle As Double
Dim delta As Double, timeDiff As Double, timeUTC As Double
Dim newt As Double

        t = calcTimeJulianCent(jd)

        ' // *** First pass to approximate sunrise

        eqtime = calcEquationOfTime(t)
        SolarDec = calcSunDeclination(t)
        hourangle = calcHourAngleSunrise(Latitude, SolarDec)

        delta = longitude - radToDeg(hourangle)
        timeDiff = 4 * delta
        ' in minutes of time
        timeUTC = 720 + timeDiff - eqtime
        ' in minutes

        ' *** Second pass includes fractional jday in gamma calc

        newt = calcTimeJulianCent(calcJDFFromJulianCent(t) + timeUTC / 1440#)
        eqtime = calcEquationOfTime(newt)
        SolarDec = calcSunDeclination(newt)
        hourangle = calcHourAngleSunrise(Latitude, SolarDec)
        delta = longitude - radToDeg(hourangle)
        timeDiff = 4 * delta

```

```

        timeUTC = 720 + timeDiff - eqtime
' in minutes

        calcSunriseUTC = timeUTC

End Function

Function calcSolNoonUTC(t, longitude)

'*****/
'* Name:    calcSolNoonUTC
'* Type:    Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of solar
'*          noon for the given day at the given location on earth
'* Arguments:
'*   t : number of Julian centuries since J2000.0
'*   longitude : longitude of observer in degrees
'* Return value:
'*   time in minutes from zero Z
'*****/

Dim newt As Double, eqtime As Double, solarNoonDec As Double, solNoonUTC As Double

    newt = calcTimeJulianCent(calcJDFFromJulianCent(t) + 0.5 + longitude / 360#)

    eqtime = calcEquationOfTime(newt)
    solarNoonDec = calcSunDeclination(newt)
    solNoonUTC = 720 + (longitude * 4) - eqtime

    calcSolNoonUTC = solNoonUTC

End Function

Function calcSunsetUTC(jd, Latitude, longitude)

'*****/
'* Name:    calcSunsetUTC
'* Type:    Function
'* Purpose: calculate the Universal Coordinated Time (UTC) of sunset
'*          for the given day at the given location on earth
'* Arguments:
'*   JD : julian day
'*   latitude : latitude of observer in degrees
'*   longitude : longitude of observer in degrees
'* Return value:
'*   time in minutes from zero Z
'*****/

Dim t As Double, eqtime As Double, SolarDec As Double, hourangle As Double
Dim delta As Double, timeDiff As Double, timeUTC As Double
Dim newt As Double

    t = calcTimeJulianCent(jd)

'    // First calculates sunrise and approx length of day

    eqtime = calcEquationOfTime(t)
    SolarDec = calcSunDeclination(t)
    hourangle = calcHourAngleSunset(Latitude, SolarDec)

    delta = longitude - radToDeg(hourangle)
    timeDiff = 4 * delta
    timeUTC = 720 + timeDiff - eqtime

'    // first pass used to include fractional day in gamma calc

    newt = calcTimeJulianCent(calcJDFFromJulianCent(t) + timeUTC / 1440#)
    eqtime = calcEquationOfTime(newt)
    SolarDec = calcSunDeclination(newt)
    hourangle = calcHourAngleSunset(Latitude, SolarDec)

    delta = longitude - radToDeg(hourangle)
    timeDiff = 4 * delta
    timeUTC = 720 + timeDiff - eqtime
'    // in minutes

    calcSunsetUTC = timeUTC

End Function

Function sunrise(lat, lon, year, month, day, timezone, dlstime)

```

```

*****/
'* Name:    sunrise
'* Type:    Main Function called by spreadsheet
'* Purpose: calculate time of sunrise for the entered date
            and location.
'* For latitudes greater than 72 degrees N and S, calculations are
            accurate to within 10 minutes. For latitudes less than +/- 72°
            accuracy is approximately one minute.
'* Arguments:
'   latitude = latitude (decimal degrees)
'   longitude = longitude (decimal degrees)
'   NOTE: longitude is negative for western hemisphere for input cells
'         in the spreadsheet for calls to the functions named
'         sunrise, solarnoon, and sunset. Those functions convert the
'         longitude to positive for the western hemisphere for calls to
'         other functions using the original sign convention
'         from the NOAA javascript code.
'   year = year
'   month = month
'   day = day
'   timezone = time zone hours relative to GMT/UTC (hours)
'   dlstime = daylight savings time (0 = no, 1 = yes) (hours)
'* Return value:
'*   sunrise time in local time (days)
*****/

Dim longitude As Double, Latitude As Double, jd As Double
Dim riseTimeGMT As Double, riseTimeLST As Double

' change sign convention for longitude from negative to positive in western hemisphere
  longitude = lon * -1
  Latitude = lat
  If (Latitude > 89.8) Then Latitude = 89.8
  If (Latitude < -89.8) Then Latitude = -89.8

  jd = calcJD(year, month, day)

'      // Calculate sunrise for this date
  riseTimeGMT = calcSunriseUTC(jd, Latitude, longitude)

'      // adjust for time zone and daylight savings time in minutes
  riseTimeLST = riseTimeGMT + (60 * timezone) + (dlstime * 60)

'      // convert to days
  sunrise = riseTimeLST / 1440

End Function

Function solarnoon(lat, lon, year, month, day, timezone, dlstime)

*****/
'* Name:    solarnoon
'* Type:    Main Function called by spreadsheet
'* Purpose: calculate the Universal Coordinated Time (UTC) of solar
            noon for the given day at the given location on earth
'* Arguments:
'   year
'   month
'   day
'* longitude : longitude of observer in degrees
'   NOTE: longitude is negative for western hemisphere for input cells
'         in the spreadsheet for calls to the functions named
'         sunrise, solarnoon, and sunset. Those functions convert the
'         longitude to positive for the western hemisphere for calls to
'         other functions using the original sign convention
'         from the NOAA javascript code.
'* Return value:
'*   time of solar noon in local time days
*****/

Dim longitude As Double, Latitude As Double, jd As Double
Dim t As Double, newt As Double, eqtime As Double
Dim solarNoonDec As Double, solNoonUTC As Double

' change sign convention for longitude from negative to positive in western hemisphere
  longitude = lon * -1
  Latitude = lat
  If (Latitude > 89.8) Then Latitude = 89.8
  If (Latitude < -89.8) Then Latitude = -89.8

```

```

jd = calcJD(year, month, day)
t = calcTimeJulianCent(jd)

newt = calcTimeJulianCent(calcJDFromJulianCent(t) + 0.5 + longitude / 360#)

eqtime = calcEquationOfTime(newt)
solarNoonDec = calcSunDeclination(newt)
solNoonUTC = 720 + (longitude * 4) - eqtime

'      // adjust for time zone and daylight savings time in minutes
solarnoon = solNoonUTC + (60 * timezone) + (dlstime * 60)

'      // convert to days
solarnoon = solarnoon / 1440

End Function

Function sunset(lat, lon, year, month, day, timezone, dlstime)

'*****/
'* Name:      sunset
'* Type:      Main Function called by spreadsheet
'* Purpose:   calculate time of sunrise and sunset for the entered date
'*            and location.
'* For latitudes greater than 72 degrees N and S, calculations are
'* accurate to within 10 minutes. For latitudes less than +/- 72°
'* accuracy is approximately one minute.
'* Arguments:
'   latitude = latitude (decimal degrees)
'   longitude = longitude (decimal degrees)
'   NOTE: longitude is negative for western hemisphere for input cells
'         in the spreadsheet for calls to the functions named
'         sunrise, solarnoon, and sunset. Those functions convert the
'         longitude to positive for the western hemisphere for calls to
'         other functions using the original sign convention
'         from the NOAA javascript code.
'   year = year
'   month = month
'   day = day
'   timezone = time zone hours relative to GMT/UTC (hours)
'   dlstime = daylight savings time (0 = no, 1 = yes) (hours)
'* Return value:
'*   sunset time in local time (days)
'*****/

Dim longitude As Double, Latitude As Double, jd As Double
Dim setTimeGMT As Double, setTimeLST As Double

' change sign convention for longitude from negative to positive in western hemisphere
longitude = lon * -1
Latitude = lat
If (Latitude > 89.8) Then Latitude = 89.8
If (Latitude < -89.8) Then Latitude = -89.8

jd = calcJD(year, month, day)

'      // Calculate sunset for this date
setTimeGMT = calcSunsetUTC(jd, Latitude, longitude)

'      // adjust for time zone and daylight savings time in minutes
setTimeLST = setTimeGMT + (60 * timezone) + (dlstime * 60)

'      // convert to days
sunset = setTimeLST / 1440

End Function

Function solarazimuth(lat, lon, year, month, day, _
    hours, minutes, seconds, timezone, dlstime)

'*****/
'* Name:      solarazimuth
'* Type:      Main Function
'* Purpose:   calculate solar azimuth (deg from north) for the entered
'*            date, time and location. Returns -999999 if darker than twilight
'*
'* Arguments:
'*   latitude, longitude, year, month, day, hour, minute, second,
'*   timezone, daylightsavingtime
'* Return value:
'*   solar azimuth in degrees from north

```



```

'*
'* Note: solarelevation and solarazimuth functions are identical
'*       and could be converted to a VBA subroutine that would return
'*       both values.
'*
'*****/

Dim longitude As Double, Latitude As Double
Dim Zone As Double, daySavings As Double
Dim hh As Double, mm As Double, ss As Double, timenow As Double
Dim jd As Double, t As Double, r As Double
Dim alpha As Double, theta As Double, Etime As Double, eqtime As Double
Dim SolarDec As Double, earthRadVec As Double, solarTimeFix As Double
Dim trueSolarTime As Double, hourangle As Double, harad As Double
Dim csz As Double, zenith As Double, azDenom As Double, azRad As Double
Dim azimuth As Double, exoatmElevation As Double
Dim step1 As Double, step2 As Double, step3 As Double
Dim refractionCorrection As Double, Te As Double, solarzen As Double

    longitude = lon * -1
    Latitude = lat
    If (Latitude > 89.8) Then Latitude = 89.8
    If (Latitude < -89.8) Then Latitude = -89.8

    Zone = timezone * -1
    daySavings = dlstime * 60
    hh = hours - (daySavings / 60)
    mm = minutes
    ss = seconds

'//    timenow is GMT time for calculation in hours since 0Z
    timenow = hh + mm / 60 + ss / 3600 + Zone

    jd = calcJD(year, month, day)
    t = calcTimeJulianCent(jd + timenow / 24#)
    r = calcSunRadVector(t)
    alpha = calcSunRtAscension(t)
    theta = calcSunDeclination(t)
    Etime = calcEquationOfTime(t)

    eqtime = Etime
    SolarDec = theta '//    in degrees
    earthRadVec = r

    solarTimeFix = eqtime - 4# * longitude + 60# * Zone
    trueSolarTime = hh * 60# + mm + ss / 60# + solarTimeFix
    '//    in minutes

    Do While (trueSolarTime > 1440)
        trueSolarTime = trueSolarTime - 1440
    Loop

    hourangle = trueSolarTime / 4# - 180#
    '//    Thanks to Louis Schwarzmayr for the next line:
    If (hourangle < -180) Then hourangle = hourangle + 360#

    harad = degToRad(hourangle)

    csz = Sin(degToRad(Latitude)) * _
        Sin(degToRad(SolarDec)) + _
        Cos(degToRad(Latitude)) * _
        Cos(degToRad(SolarDec)) * Cos(harad)

    If (csz > 1#) Then
        csz = 1#
    ElseIf (csz < -1#) Then
        csz = -1#
    End If

    zenith = radToDeg(Application.WorksheetFunction.Acos(csz))

    azDenom = (Cos(degToRad(Latitude)) * Sin(degToRad(zenith)))

    If (Abs(azDenom) > 0.001) Then
        azRad = ((Sin(degToRad(Latitude)) * _
            Cos(degToRad(zenith))) -
            Sin(degToRad(SolarDec))) / azDenom
        If (Abs(azRad) > 1#) Then
            If (azRad < 0) Then
                azRad = -1#
            Else
                azRad = 1#
            End If
        End If
    End If

```

```

        azRad = 1#
    End If
End If

azimuth = 180# - radToDeg(Application.WorksheetFunction.Acos(azRad))

If (hourangle > 0#) Then
    azimuth = -azimuth
End If
Else
    If (Latitude > 0#) Then
        azimuth = 180#
    Else
        azimuth = 0#
    End If
End If
If (azimuth < 0#) Then
    azimuth = azimuth + 360#
End If

exoatmElevation = 90# - zenith

If (exoatmElevation > 85#) Then
    refractionCorrection = 0#
Else
    Te = Tan(degToRad(exoatmElevation))
    If (exoatmElevation > 5#) Then
        refractionCorrection = 58.1 / Te - 0.07 / (Te * Te * Te) + _
            0.000086 / (Te * Te * Te * Te * Te)
    ElseIf (exoatmElevation > -0.575) Then
        step1 = (-12.79 + exoatmElevation * 0.711)
        step2 = (103.4 + exoatmElevation * (step1))
        step3 = (-518.2 + exoatmElevation * (step2))
        refractionCorrection = 1735# + exoatmElevation * (step3)
    Else
        refractionCorrection = -20.774 / Te
    End If
    refractionCorrection = refractionCorrection / 3600#
End If

solarzen = zenith - refractionCorrection
solarazimuth = azimuth

End Function

Function solarelevation(lat, lon, year, month, day, _
    hours, minutes, seconds, timezone, dlstime)

'*****/
'* Name:      solarazimuth
'* Type:      Main Function
'* Purpose:   calculate solar azimuth (deg from north) for the entered
'*           date, time and location. Returns -999999 if darker than twilight
'*
'* Arguments:
'*   latitude, longitude, year, month, day, hour, minute, second,
'*   timezone, daylightsavingtime
'* Return value:
'*   solar azimuth in degrees from north
'*
'* Note: solarelevation and solarazimuth functions are identical
'*       and could be converted to a VBA subroutine that would return
'*       both values.
'*
'*****/

Dim longitude As Double, Latitude As Double
Dim Zone As Double, daySavings As Double
Dim hh As Double, mm As Double, ss As Double, timenow As Double
Dim jd As Double, t As Double, r As Double
Dim alpha As Double, theta As Double, Etime As Double, eqtime As Double
Dim SolarDec As Double, earthRadVec As Double, solarTimeFix As Double
Dim trueSolarTime As Double, hourangle As Double, harad As Double
Dim csz As Double, zenith As Double, azDenom As Double, azRad As Double
Dim azimuth As Double, exoatmElevation As Double
Dim step1 As Double, step2 As Double, step3 As Double
Dim refractionCorrection As Double, Te As Double, solarzen As Double

    longitude = lon * -1
    Latitude = lat
    If (Latitude > 89.8) Then Latitude = 89.8

```

```

If (Latitude < -89.8) Then Latitude = -89.8

Zone = timezone * -1
daySavings = dlstime * 60
hh = hours - (daySavings / 60)
mm = minutes
ss = seconds

'//    timenow is GMT time for calculation in hours since 0Z
      timenow = hh + mm / 60 + ss / 3600 + Zone

      jd = calcJD(year, month, day)
      t = calcTimeJulianCent(jd + timenow / 24#)
      r = calcSunRadVector(t)
      alpha = calcSunRtAscension(t)
      theta = calcSunDeclination(t)
      Etime = calcEquationOfTime(t)

      eqtime = Etime
      SolarDec = theta '//    in degrees
      earthRadVec = r

      solarTimeFix = eqtime - 4# * longitude + 60# * Zone
      trueSolarTime = hh * 60# + mm + ss / 60# + solarTimeFix
      '//    in minutes

      Do While (trueSolarTime > 1440)
        trueSolarTime = trueSolarTime - 1440
      Loop

      hourangle = trueSolarTime / 4# - 180#
      '//    Thanks to Louis Schwarzmayr for the next line:
      If (hourangle < -180) Then hourangle = hourangle + 360#

      harad = degToRad(hourangle)

      csz = Sin(degToRad(Latitude)) * _
           Sin(degToRad(SolarDec)) + _
           Cos(degToRad(Latitude)) * _
           Cos(degToRad(SolarDec)) * Cos(harad)

      If (csz > 1#) Then
        csz = 1#
      ElseIf (csz < -1#) Then
        csz = -1#
      End If

      zenith = radToDeg(Application.WorksheetFunction.Acos(csz))

      azDenom = (Cos(degToRad(Latitude)) * Sin(degToRad(zenith)))

      If (Abs(azDenom) > 0.001) Then
        azRad = ((Sin(degToRad(Latitude)) * _
                  Cos(degToRad(zenith))) - _
                  Sin(degToRad(SolarDec))) / azDenom
        If (Abs(azRad) > 1#) Then
          If (azRad < 0) Then
            azRad = -1#
          Else
            azRad = 1#
          End If
        End If

        azimuth = 180# - radToDeg(Application.WorksheetFunction.Acos(azRad))

        If (hourangle > 0#) Then
          azimuth = -azimuth
        End If
      Else
        If (Latitude > 0#) Then
          azimuth = 180#
        Else
          azimuth = 0#
        End If
      End If

      If (azimuth < 0#) Then
        azimuth = azimuth + 360#
      End If

      exoatmElevation = 90# - zenith

```

```

    If (exoatmElevation > 85#) Then
        refractionCorrection = 0#
    Else
        Te = Tan(degToRad(exoatmElevation))
        If (exoatmElevation > 5#) Then
            refractionCorrection = 58.1 / Te - 0.07 / (Te * Te * Te) + _
                0.000086 / (Te * Te * Te * Te * Te)
        ElseIf (exoatmElevation > -0.575) Then
            step1 = (-12.79 + exoatmElevation * 0.711)
            step2 = (103.4 + exoatmElevation * (step1))
            step3 = (-518.2 + exoatmElevation * (step2))
            refractionCorrection = 1735# + exoatmElevation * (step3)
        Else
            refractionCorrection = -20.774 / Te
        End If
        refractionCorrection = refractionCorrection / 3600#
    End If

    solarzen = zenith - refractionCorrection
    solarelevation = 90# - solarzen

End Function

Sub solarposition(lat, lon, year, month, day, _
    hours, minutes, seconds, timezone, dlstime, _
    solarazimuth, solarelevation, earthRadVec)

'*****/
'* Name:      solarposition
'* Type:      Subroutine
'* Purpose:   calculate solar azimuth (deg from north)
'*           and elevation (deg from horizon) for the entered
'*           date, time and location.
'*
'* Arguments:
'*   latitude, longitude, year, month, day, hour, minute, second,
'*   timezone, daylightsavingtime
'* Return value:
'*   solar azimuth in degrees from north
'*   solar elevation in degrees from horizon
'*   earth radius vector (distance to the sun in AU)
'*
'* Note: solarelevation and solarazimuth functions are identical
'*       and could converted to a VBA subroutine that would return
'*       both values.
'*
'*****/

Dim longitude As Double, Latitude As Double
Dim Zone As Double, daySavings As Double
Dim hh As Double, mm As Double, ss As Double, timenow As Double
Dim jd As Double, t As Double, r As Double
Dim alpha As Double, theta As Double, Etime As Double, eqtime As Double
'Dim SolarDec As Double, earthRadVec As Double, solarTimeFix As Double
Dim SolarDec As Double, solarTimeFix As Double
Dim trueSolarTime As Double, hourangle As Double, harad As Double
Dim csz As Double, zenith As Double, azDenom As Double, azRad As Double
Dim azimuth As Double, exoatmElevation As Double
Dim step1 As Double, step2 As Double, step3 As Double
Dim refractionCorrection As Double, Te As Double, solarzen As Double

    longitude = lon * -1
    Latitude = lat
    If (Latitude > 89.8) Then Latitude = 89.8
    If (Latitude < -89.8) Then Latitude = -89.8

    Zone = timezone * -1
    daySavings = dlstime * 60
    hh = hours - (daySavings / 60)
    mm = minutes
    ss = seconds

'//   timenow is GMT time for calculation in hours since 0Z
    timenow = hh + mm / 60 + ss / 3600 + Zone

    jd = calcJD(year, month, day)
    t = calcTimeJulianCent(jd + timenow / 24#)
    r = calcSunRadVector(t)
    alpha = calcSunRtAscension(t)
    theta = calcSunDeclination(t)
    Etime = calcEquationOfTime(t)

```

```

eqtime = Etime
SolarDec = theta '//'      in degrees
earthRadVec = r

solarTimeFix = eqtime - 4# * longitude + 60# * Zone
trueSolarTime = hh * 60# + mm + ss / 60# + solarTimeFix
'//      in minutes

Do While (trueSolarTime > 1440)
    trueSolarTime = trueSolarTime - 1440
Loop

hourangle = trueSolarTime / 4# - 180#
'//      Thanks to Louis Schwarzmayr for the next line:
If (hourangle < -180) Then hourangle = hourangle + 360#

harad = degToRad(hourangle)

csz = Sin(degToRad(Latitude)) * _
      Sin(degToRad(SolarDec)) + _
      Cos(degToRad(Latitude)) * _
      Cos(degToRad(SolarDec)) * Cos(harad)

If (csz > 1#) Then
    csz = 1#
ElseIf (csz < -1#) Then
    csz = -1#
End If

zenith = radToDeg(Application.WorksheetFunction.Acos(csz))

azDenom = (Cos(degToRad(Latitude)) * Sin(degToRad(zenith)))

If (Abs(azDenom) > 0.001) Then
    azRad = ((Sin(degToRad(Latitude)) * _
              Cos(degToRad(zenith))) - _
              Sin(degToRad(SolarDec))) / azDenom
    If (Abs(azRad) > 1#) Then
        If (azRad < 0) Then
            azRad = -1#
        Else
            azRad = 1#
        End If
    End If
    azimuth = 180# - radToDeg(Application.WorksheetFunction.Acos(azRad))

    If (hourangle > 0#) Then
        azimuth = -azimuth
    End If
Else
    If (Latitude > 0#) Then
        azimuth = 180#
    Else
        azimuth = 0#
    End If
End If
If (azimuth < 0#) Then
    azimuth = azimuth + 360#
End If

exoatmElevation = 90# - zenith

If (exoatmElevation > 85#) Then
    refractionCorrection = 0#
Else
    Te = Tan(degToRad(exoatmElevation))
    If (exoatmElevation > 5#) Then
        refractionCorrection = 58.1 / Te - 0.07 / (Te * Te * Te) + _
            0.000086 / (Te * Te * Te * Te * Te)
    ElseIf (exoatmElevation > -0.575) Then
        step1 = (-12.79 + exoatmElevation * 0.711)
        step2 = (103.4 + exoatmElevation * (step1))
        step3 = (-518.2 + exoatmElevation * (step2))
        refractionCorrection = 1735# + exoatmElevation * (step3)
    Else
        refractionCorrection = -20.774 / Te
    End If
    refractionCorrection = refractionCorrection / 3600#
End If

```

```
solarzen = zenith - refractionCorrection  
solarazimuth = azimuth  
solarelevation = 90# - solarzen
```

```
End Sub
```

APPENDIX C: SEDIMENT-WATER HEAT EXCHANGE

Although the omission of sediment-water heat exchange is usually justified for deeper systems, it can have a significant impact on the heat balance for shallower streams. Consequently, sediment-water heat exchange is included in QUAL2K.

A major impediment to its inclusion is that incorporating sediment heat transfer often carries a heavy computational burden. This is because the sediments are usually represented as a vertically segmented distributed system. Thus, inclusion of the mechanism results in the addition of numerous sediment segments for each overlying water element.

In the present appendix, I derive a computationally-efficient lumped approach that yields comparable results to the distributed methods.

The conduction equation is typically used to simulate the vertical temperature distribution in a distributed sediment (Figure 26a)

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (254)$$

This model can be subjected to the following boundary conditions:

$$T(0, t) = \bar{T} + T_a \cos[\omega(t - \phi)]$$

$$T(\infty, t) = \bar{T}$$

where T = sediment temperature [$^{\circ}\text{C}$], t = time [s], α = sediment thermal diffusivity [$\text{m}^2 \text{s}^{-1}$], and z = depth into the sediments [m], where $z = 0$ at the sediment-water interface and z increases downward, \bar{T} = mean temperature of overlying water [$^{\circ}\text{C}$], T_a = amplitude of temperature of overlying water [$^{\circ}\text{C}$], ω = frequency [s^{-1}] = $2\pi/T_p$, T_p = period [s], and ϕ = phase lag [s]. The first boundary condition specifies a sinusoidal Dirichlet boundary condition at the sediment-water interface. The second specifies a constant temperature at infinite depth. Note that the mean of the surface sinusoid and the lower fixed temperature are identical.

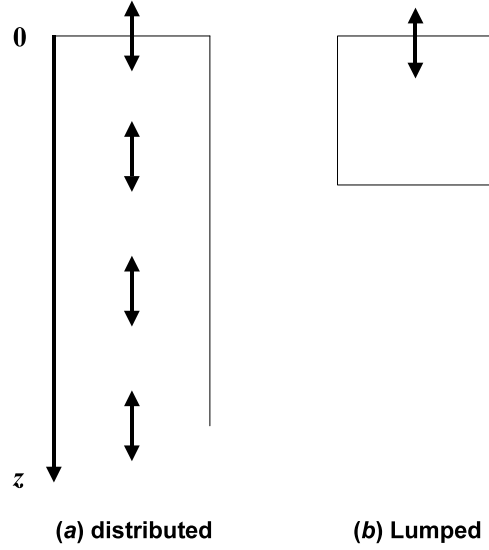


Figure 26. Alternate representations of sediments: (a) distributed and (b) lumped.

Applying these boundary conditions, Eq. (230) can be solved for (Carslaw and Jaeger 1959)

$$T(z, t) = \bar{T} + T_a e^{-\omega' z} \cos[\omega(t - \phi) - \omega' z] \quad (255)$$

where ω' [m^{-1}] is defined as

$$\omega' = \sqrt{\frac{\omega}{2\alpha}} \quad (256)$$

The heat flux at the sediment water interface can then be determined by substituting the derivative of Eq. (231) into Fourier's law and evaluating the result at the sediment-water interface ($z = 0$) to yield

$$J(0, t) = \rho C_p \sqrt{\omega \alpha} T_a \cos[\omega(t - \phi) + \pi / 4] \quad (257)$$

where $J(0, t) = \text{flux } [\text{W}/\text{m}^2]$.

An alternative approach can be developed using a first-order lumped model (Figure 26b),

$$H_s \rho_s C_{ps} \frac{dT_s}{dt} = \frac{\alpha_s \rho_s C_{ps}}{H_s / 2} [\bar{T} + T_a \cos[\omega(t - \phi)] - T_s]$$

where H_{sed} = the thickness of the sediment layer [m], ρ_s = sediment density [kg/m^3], and C_{ps} = sediment specific heat [$\text{joule } (\text{kg } ^\circ\text{C})^{-1}$]. Collecting terms gives,

$$\frac{dT}{dt} + k_h T = k_h \bar{T} + k_h T_a \cos[\omega(t - \phi)]$$

where

$$k_h = \frac{2\alpha_s}{H_{sed}^2}$$

After initial transient have died out, this solution to this equation is

$$T = \bar{T} + \frac{k_h}{\sqrt{k_h^2 + \omega^2}} T_a \cos[\omega(t - \phi) - \tan^{-1}(\omega / k_h)] \quad (258)$$

which can be used to determine the flux as

$$J = \frac{2\alpha}{H_{sed}} \rho C_p T_a \left[\cos[\omega(t - \phi)] - \frac{k_h}{\sqrt{k_h^2 + \omega^2}} \cos[\omega(t - \phi) - \tan^{-1}(\omega / k_h)] \right] \quad (259)$$

It can be shown that Eqs. (231) and (234) yield identical results if the depth of the single layer is set at

$$H_{sed} = \frac{1}{\omega'} \quad (260)$$

Water quality models typically consider annual, weekly and diel variations. Using $\alpha = 0.0035$ cm²/s (Hutchinson 1957), the single-layer depth that would capture these frequencies can be calculated as 2.2 m, 30 cm and 12 cm, respectively.

Because QUAL2K resolves diel variations, a value on the order of 12 cm should be selected for the sediment thickness. We have chosen of value of 10 cm as being an adequate first estimate because of the uncertainties of the river sediment thermal properties (Table 4).