Appendix J. GEMSS Code Review

Final: December 19, 2011

GEMSS Code Review

Module Equations and Variables

Robert B. Ambrose, Jr., P.E.

Table of Contents

Task 1.1 provides a review of Fortran code implemented in GEMSS modules WQCBM, GenAlgae (aka, GAM), and WQADD. This review was conducted during June and early July 2010. Questions and findings were relayed to Greg Pelletier, Washington Ecology and Venkat Kolluru by email throughout the review. Priority was given to WQCBM and GenAlgae, which are being implemented in a set of South Puget Sound simulations, and the organic matter module in WQADD. A review of the final code conducted in May 2011 verified that all of the recommended changes were properly implemented in WQCBM and GenAlgae. A final review on December 19, 2011 verified that all the recommended changes in the WQADD Organic Matter module were properly implemented. As of this date, all issues affecting present applications of the model by Washington Ecology are resolved. Note that errors remain to be resolved in the pH-TIOC module. Finally, there is still an issue to be resolved in the Bottom Algae module concerning the transfor of ON and OP from WQADD to WQE5M and to WQDPM.

Approach

For each model state variable, locate and review its implementation in code. Check for accuracy in translating appropriate equations to algorithms. Verify each variable used in the algorithm, tracing its declaration, point of origin, and transfer through code. Document findings for each equation, including questions or recommendations for the developers. Several algorithms were checked against equivalent sections of code in WASP, and some were checked against equations documented in QUAL2K. Each equation in the scientific algrithms was examined and the units for each term were documented and checked for consistency. The overall logic implementing each set of equations was checked.

Summary of Findings

First, the three codes were well-organized, providing little difficulty in overall navigation and understanding. Like most codes, these would benefit from more internal documentation, particularly providing the definition and units of input data (which are not documented externally). WQCBM seems to be the code with more testing and history of use. Only three real coding errors were discovered, though a couple more suggested changes should be considered in the long term. Some algorithms and coding options have evolved over time, and this model would benefit from "cleaning up" to make parallel logic with other similar sections. GenAlgae seems to be a newer, more specialized code, and is well written. A few errors were found, but this model should be in very good shape now. WQADD seems to be the newest, least-tested code, with a number of errors. This code will require more thorough verification testing after the known errors are corrected.

Five coding errors and suggested revisions were reported for WQCBM Version 4:

1. Algal growth temperature correction multiplier (term WQCBM3D_TL for diatoms and dinoflagellates): remove growth rate constant from this term, so that it is a dimensionless multiplier

2. DO from algal photosynthesis (term p06a for diatoms and dinoflagellates): implement the theoretically-based method in WASP
3. CBODslow from POC hydrolysis (term p16c): add aoc to expression to correct units
4. Algal light limitation units conversion (terms rins_d and rins-f): change 4.1868 [J/cal-I.T.] to 4.184 [J/cal-thermochemical]
5. Dinoflagellate light and temperature migration: change the first condition during daylight from WQCBM3D_ri(k) > rins_f .and. WQCBM3D_ri(k+1) <= rins_f to WQCBM3D_ri(k) > rins_f .and. WQCBM3D_ri(k+1) > rins_f.

The first three suggested coding revisions were reviewed by Greg Pelletier and implemented by Venkat Kolluru in WQCBM Version 5. The revised code was reviewed and found accurate. The fourth suggested coding revision was reported later. This units conversion was removed from WQCBM Version 5 with a note that units conversions are handled now in subroutine WQMReadsubs.f90. The fifth correction was implemented in WQCBM Version 5. All five revisions were verified in WQCMB version 5 received on 5/25/11.

Six coding errors and suggested revisions were reported for GenAlgae Version 3:

1. Salinity toxicity units (term GAM3D_stf): adjust units from [1/sec] to [1/day] before use in growth equation
2. Algal growth limit (term AlgaeGrwthNetLimit): revise code to calculate AlgaeGrwthNetLimit when stMethod=2
3. Algal growth temperature correction (term GAM3D_ft): revise code to use options for either theta correction or Topt correction, not both
4. DO from algal photosynthesis (term DOFromAlg): implement the theoretically-based method in WASP
5. Particulate organic matter from zooplankton grazing (terms POCFFromAlg, POCSFromAlg, POCRFromAlg, PONFromAlg, POPFromAlg): remove algal concentration from grazing term
6. Refractory dissolved organic matter from algal death (variable RDOM): remove algal death term from RDOM derivative

The first five suggested coding revisions were reviewed by Greg Pelletier and implemented by Venkat Kolluru in GenAlgae Version 4. The revised code was reviewed and found accurate. The sixth suggested coding revision was reported later, and the correction was implemented in GenAlgae Version 4, received on 5/25/11. All revisions were verified in version 4 received on 5/25/11.

WQADD contains separate sections (or modules) for organic matter, pH-TIOC, and bottom algae. Coding errors and suggested revisions for the WQADD Version 2 Organic Matter module include:

1. Labile Dissolved Organic Matter (term L2RDOMTransfer): multiply rate constant by c(n,k,I_LDOM), not c(n,k,I_LPOM)
2. Labile Dissolved Organic Matter (term h(n,k,I_LDOM)): multiply terms in derivative by volume
3. Refractory Dissolved Organic Matter (term h(n,k,I_RDOM)): multiply terms in derivative by volume

4. Labile Particulate Organic Matter (term LPOMDissolution): multiply term by LPOM concentration, c(n,k,I_LPOM)
5. Labile Particulate Organic Matter (term h(n,k,I_LPOM)): multiply terms in derivative by volume
6. Refractory Particulate Organic Matter (term L2RPOMTransfer): multiply rate constant by c(n,k,I_LPOM), not c(n,k,I_RPOM)
7. Refractory Particulate Organic Matter (term RPOMDissolution): multiply term by RPOM concentration, c(n,k,I_RPOM)
8. Refractory Particulate Organic Matter (term h(n,k,I_RPOM)): multiply terms in derivative by volume
9. Inorganic carbon from organic matter (term h(n,k,I_TIOC)): RPOMDecay is used twice; the first instance should be LPOMDecay
10. Phosphate from organic matter (term h(n,k,I_PO4)): RPOMDecay is used twice; the first instance should be LPOMDecay
11. Settling units (terms vs_LPOM and vs_RPOM): units of [1/sec] work here, but this is labelled "settling velocity," which implies units of L/T and the need to divide by cell depth

These revisions were checked in WQADDConstituents Version 2 received on 5/25/11. Revision 6 was checked in the final version received December 14, 2011. Corrections to all of these 11 flagged errors have been verified.

The following errors were flagged in the pH-TIOC module:

1. CO2Sat using pCO2SetMethod=2 (or 0): the term kacT [m/sec] was used instead of a term such as KHCO2 [moleC/L-atm]
2. In the TIOC derivative, the bottom algae term should be multiplied by CellArea rather than CellVolume
3. The TIOC derivative is missing a CBOD oxidation term
4. The alkalinity derivative is missing phytoplankton PO_4 photosynthesis and respiration terms

These revisions were checked in WQADDConstituents Version 2 received on December 14, 2011. Errors 2 and 3 were addressed, but error 1 and omission 4 remain. In addition, the TIOC derivative presently includes mixed units. As a result, this version of the pH-TIOC code is not ready for use.

The following errors, warnings, and suggestions were flagged in the bottom algae module:

1. DO from bottom algae has an error using factor 138/107 in the second term: the term should be multiplied by 1.5*anc*12/14 (0.289 for anc of 9/40), or the first term should be multiplied by Pab
2. Bottom light for periphyton is calculated using a single light extinction coefficient for layer “k” over the water column depth; I recommend using calculated bottom light from WQCBM, which is RBotPAR(n,k) for k=k0(i,j)
3. Bottom algae use the same constants as WQCBM phytoplankton (diatoms and dinoflagellates) to cycle algal carbon and nutrients; I recommend using coefficients specific to bottom algae

4. In the Bottom Algae code for Case(WQSSTERMS_USING_WQCBM), the particulate OP derivative is equated to particulate organic nitrogen from bottom algae (PONFromBotAlg); that term should use POPFromBotAlg
5. In the Bottom Algae code the dissolved oxygen derivative is multiplied by the 1.0e-3 [g/mg], creating a units error; the term 1.0e-3 should be removed
6. In the Bottom Algae code for Case(WQSSTERMS_USING_WQDPM), the ON and OP derivatives use incorrect index terms "acc(I_ON_D)" and "(h(n,k,I_ON_D);" these terms should use indeces "I_ON" and "I_OP" rather than "I_OP_D."
7. In the Bottom Algae code for Case(WQSSTERMS_USING_WQE5M) and Case(WQSSTERMS_USING_WQDPMM), the ON derivative includes the term "DONFromBotAlg," but not "PONFromBotAlg." If state variable I_ON in these models represents total nonliving organic nitrogen, then the derivative should also include "PONFromBotAlg." The same comment applies to the OP derivative in these models.

These revisions were checked in WQADDConstituents Version 2 received on 5/25/11. Errors 1, 2, 4, and 5 have been corrected. Observation 3 appears to have been made in error, and so this revision was not necessary. In Version 2 received on December 14, 2011, potential error 6 has been corrected, but potential error 7 has not been addressed. This potential error does not affect the use of WQADD with WQCBM.

These findings are described in more detail in the sections below.

WQCBM

WQCBM implements 21 state variables:

1. Ammonia Nitrogen, I_NH3 [g N/m^3]
2. Nitrate Nitrogen, I_N03 [g N/m^3]
3. Inorganic Phosphorus, I_PO4 [g P/m^3]
4. Phytoplankton Carbon, I_Phyt [g C/m^3]
5. Carbonaceous BOD, I_CBOD [g O2/m^3]
6. Dissolved Oxygen, I_DO [g O2/m^3]
7. Organic Nitrogen, I_ON [g N/m^3]
8. Organic Phosphorus, I_OP [g P/m^3]
9. Dissolved Carbonaceous BOD, I_CBOD_D [g O2/m^3]
10. Particulate Carbonaceous BOD, I_CBOD_P [g O2/m^3]
11. Dissolved Organic Nitrogen, I_ON_D [g N/m^3]
12. Particulate Organic Nitrogen, I_ON_P [g N/m^3]
13. Dissolved Organic Phosphorus, I_OP_D [g P/m^3]
14. Particulate Organic Phosphorus, I_OP_P [g P/m^3]
15. Fast Reating Dissolved Carbonaceous BOD, I_CBOD_F [g O2/m^3]
16. Slow Reating Dissolved Carbonaceous BOD, I_CBOD_S [g O2/m^3]
17. Fast Reacting Particulate Organic Carbon, I_OC_P_F [g C/m^3]
18. Slow Reacting Particulate Organic Carbon, I_OC_P_S [g C/m^3]
19. Refractory Particulate Organic Carbon, I_OC_P_R [g C/m^3]

20. Diatoms algae group, I_DAP [g C/m^3]
21. Dinoflagellates algae group, I_DFP [g C/m^3]

Ammonia Nitrogen

No errors were found in the ammonia nitrogen section of code. In the long run, this code would benefit from cleaning up the logical structures to be consistent with similar code in the dinoflagellate section and the phosphate section.

Some legacy code dealing with ammonia inhibition contains mass balance errors. This code is effectively bypassed by setting option_fnh3 = 0. It would be better if this code is simply commented out. This old section of code recalculates NH3 uptake by phytoplankton "using inhibition:"

```
If(option_fnh3.eq.1) Then                  ! use inhibition
   p01c =   - gp_d*anc*c(n,k,I_DAP)        !(c) algae growth - Diat
   If(xri_f.eq.0.0) Then                     !(c) algae growth - Dino
      p01c =   p01c - dr_f*anc*c(n,k,I_DFP) !    dark respiration
   Else
      p01c =   p01c - gp_f*anc*c(n,k,I_DFP)
   End If
End If
```

This takes up only NH3 (the ammonia preference term, pnh3, is omitted). But the N-limitation term used to calculate gp_d includes NO3, and NO3 is still being taken up using (1-pnh3). There seems to be a mass balance error if option_fnh3 is set to 1. It would be better coded with IF-THEN-ELSE.

Final review – while this section of code was not changed, an earlier section of code assures that option_nh3 is always 0, bypassing the mass balance problem:

```
option_fnh3 = 0
If(option_fnh3.eq.1) Then
      kinhib = 0.056
      fnh3 = 1./(1.+c(n,k,I_NH3)/kinhib)
End If
```

Nitrate Nitrogen

No errors were found in this section of code. In the long run, this code would benefit from separating dark respiration from growth (using a new term, say p02d), as is done in the phosphorus section of code. Dinoflagellate growth is added to term p02b in a day-night conditional expression. In the phosphorus code, the growth term is located outside of the day-night conditional expression. This shouldn't matter because at night the growth term is 0. It would be better if the equivalent code were used in the ammonia, nitrate, and phosphate sections.

Inorganic Phosphorus

No errors were found in this section of code.

Carbonaceous BOD

One error was found in this section of code. The hydrolysis rate constants for fast and slow particulate organic matter (kdp9f and kdp9s) should be multiplied by the oxygen to carbon ratio, aoc:

```
p16c =   kpd9f*thpd9f**temp20*c(n,k,I_OC_P_F) + &      !(c) Hydrolysis
         kpd9s*thpd9s**temp20*c(n,k,I_OC_P_S)
```

Final review – This bug has been fixed in WQCBM Version 5:

```
!Ambrose Bug Fix
p16c =   aoc * kpd9f*thpd9f**temp20*c(n,k,I_OC_P_F) + &
         aoc * kpd9s*thpd9s**temp20*c(n,k,I_OC_P_S)
```

Dissolved Oxygen

No errors were found in this section of code. Nevertheless, the expession for photosynthetic DO production is unconventional. The code correctly implements equation 2 in the supporting document, but at the upper end of the photosynthetic quotient (PQ) range for growth using NH_4 and NO_3 (i.e., 0.5+/-0.3 and 0.1+/-0.1 in equation 2, versus 0.8 and 0.2 in code).

I recommend using the more conventional WASP equation for growth:

```
dC6/dt = GP1*PHYT * (PNH3*(32/12) + (1-PNH3) * 32*(1/12 + 1.5 *
NCRB*1/14)
```

where NCRB is the phytoplankton N:C ratio by weight [gN/gC] (anc in WQCBM). For typical anc values of 0.18 and 0.25, the WASP equation gives PQ values of 1.23 and 1.32, both within the experimental range of 1.2 to 1.8 quoted in the WQCBM manual. To get a PQ value of 1.5, anc would have to be 0.39, and to get PQ to the upper end of the range, 1.8, anc would have to be 0.62. It seems to me that an anc value of 0.4 is at the upper end of reasonable stoichiometry. Therefore, I question the use of 0.2 and 0.8 in the WQCMB model instead of 0.1 and 0.5, the midpoint of the empirical range cited in the documentation.

The WQCBM approach forces the model to satisfy experimental data by hard-wiring code, and by removing the dependence of PQ on algal stoichiometry (specifically, anc). The WASP approach has a theoretical basis that falls within the range of experimental results, albeit near the lower end for reasonable input values of anc. I understand the use of empirical relationships in models, but prefer more mechanistic approaches when available.

For the long-term, a slightly more efficient code construct is recommended for reaeration Case 2, to avoid switching units of K2 back and forth (see in-line comments in Appendix).

Final review – The WASP approach was introduced as an option in WQCBM Version 5, and examined for accuracy.

```
!Ambrose Bug Fix
If(UseLOTTMethod == 1) Then
   !
   !use the original LOTT study method
   p06a = gp_d*(32./12. + 0.8*32./12.*(1. - pnh3_d) & !(a) Photo- Diat
            + 0.2*32./12.*(pnh3_d))*c(n,k,I_DAP)
   p06a = p06a + &
          gp_f*(32./12. + 0.8*32./12.*(1. - pnh3_f) & !(a) Photo- Dinos
                        + 0.2*32./12.*(pnh3_f))*c(n,k,I_DFP)
Else
   !
   !use the WASP EUTRO method adapted from WASP DISSOXYG.FOR
  !note: c(n,k,I_DAP) and c(n,k,I_DFP) is phyto biomass in mgC/L=gC/m^3
   !growth of diatoms (I_DAP) using CO2 and NH3
   p06a = pnh3_d  * gp_d * c(n,k,I_DAP) * 32. / 12.
   !growth of diatoms (I_DAP) using CO2 and NO3 (2NO3 = 2NH3 + 3O2)
   p06a = p06a + (1. - pnh3_d) * gp_d * c(n,k,I_DAP) * 32. * (1./12.+
1.5 * anc / 14.)
      !growth of dinoflagellates (I_DFP) using CO2 and NH3
      p06a = p06a +       pnh3_f  * gp_f * c(n,k,I_DFP) * 32. / 12.
      !growth of dinoflagellates (I_DFP) using CO2 and NO3 (2NO3 = 2NH3
      + 3O2)
      p06a = p06a + (1. - pnh3_f) * gp_f * c(n,k,I_DFP) * 32. * (1./12.
+ 1.5 * anc / 14.)
                           !
!! here is the equivalent code from WASP DISSOXYG.FOR:
!C        Growth of phytoplankton using CO2 and NH3
!             SR19PA = SR19PA + PNH3G1(i)*GP1(i)*PHYT(i)*32./12.
!C        Growth of phytoplankton using CO2 and NO3 (2NO3 = 2NH3 + 3O2)
!             SR19PB = SR19PB + (1. - PNH3G1(i))*GP1(i)*PHYT(i)
!     .        *32.*(1./12. + 1.5*NCRB(i)/14.)
End If
```

Dissolved Organic Nitrogen

No errors were found in this section of code.

Particulate Organic Nitrogen

No errors were found in this section of code.

Dissolved Organic Phosphorus

No errors were found in this section of code.

Particulate Organic Phosphorus

No errors were found in this section of code.

Particulate Organic Carbon

No errors were found in this section of code.

The terms fd9f, fd9s, and fd9r are the unitless fractions of POC assigned to fast, slow, and refractory POC. The code here brings these variables in, but does not assure that their sum is 1.0. This is probably handled in the main model input code that is not part of this review. The same comments apply to fg9f, fg9s, and fg9r for grazing.

Light and Temperature Limitation

No significant errors were found in this section of code, but two small changes are recommended.

First, there is an error in the fourth decimal place in converting the saturating light intensity (rins_d and rins_f) from Ly/day to W/m^2. Note that [Ly] = [thermochemical cal/cm^2], and there are 4.1840 J/thermochemical cal. The code used the conversion factor 4.1868 [J/cal-I.T.].

```
!rins_d = rins_d * 4.1868 * 10000.0 / 86400.0  !convert from cal/cm^2/d
to W/m^2
```

Note that the code is commented out here, but used elsewhere to make the conversion.

Second, the temperature limitation terms (used for output, I believe) include the growth rate constant, and are actually the temperature-adjusted rate constants:

```
    WQCBM3D_TL(n,k,1) = k1c_d*th1c_d**temp20
    WQCBM3D_TL(n,k,2) = k1c_f*th1c_f**temp20
```

I recommend omitting klc_d and klc_f from these expressions.

Note that the dinoflagellate light and temperature velocities were checked for obvious errors, but were not compared to documented equations. This verification should be done.

Final review – The terms klc_d and klc_f were omitted in WQCBM Version 5.

```
!Ambrose Bug Fix
WQCBM3D_TL(n,k,1) = th1c_d**temp20
!temperature limitation factor for diatoms in cell n layer k
!
!Ambrose Bug Fix
WQCBM3D_TL(n,k,2) = th1c_f**temp20
```

The terms converting saturating light intensity (rins_d and rins_f) from Ly/day to W/m^2 were commented out, noting that they are handled now in subroutine WQMReadsubs.f90.

```
!VSK: 06-24-2010 Commented since unit conversions are done in
WQMReadsubs.f90
!Greg originally asked to use this conversions here.
!We then decided to add multiple units for light
!But this was not updated
!rins_d = rins_d * 4.1868 * 10000.0 / 86400.0    !convert from
cal/cm^2/d to W/m^2
!rins_f = rins_f * 4.1868 * 10000.0 / 86400.0    !convert from
cal/cm^2/d to W/m^2
```

Diatoms

No errors were found in this section of code. Nevertheless, two points are worth highlighting. First, grazing option 1 (constant grazing rate) is not temperature-corrected in WQCBM, but is temperature-corrected in GenAlgae. Second, the growth rate nitrogen limitation uses total ammonia-N instead of dissolved NH_4^+-N. This model does not include partitioning of ammonia, and so the total concentration should be the same as the dissolved concentration. The model also does not include speciation of ammonia, although with the pH calculation from WQADD, the information is present to include this calculation.

Although the growth rate calculation is correct:

```
  gp_d    = k1c_d*th1c_d**temp20*xri_d*dmin1((c(n,k,I_NH3) +
c(n,k,I_NO3))        &
          /(kmn_d + c(n,k,I_NH3) + c(n,k,I_NO3)),(plimit**acc(I_PO4)))
```

I'd prefer to see the temperature factor and the nitrogen limitation factor calculated separately before use in the growth rate:

```
   nlimit_d   = (c(n,k,I_NH3) + c(n,k,I_NO3)) /(kmn_d + c(n,k,I_NH3) +
c(n,k,I_NO3))
   plimit_d   = plimit**acc(I_PO4)
   xnutlim_d  = dmin1(nlimit_d, plimit_d)
   xtemp_d    = th1c_d**temp20
   gp_d       =  k1c_d * xtemp_d * xri_d * xnutlim_d
```

Final review – The constant grazing rate is now temperature-corrected:

```
!Ambrose Bug Fix
Case(ConstantGrazing)
      ggraze_d = kgmicro_d*thkt_d**temp20 + kgmacro_d*thkt_d**temp20
```

Dinoflagellates

One error was found in this section of code. The logic for settling and diel vertical migration as a function of temperature and light is complicated, but an examination of the units and logic revealed an inconsistency in logic. The conditional expression in the first two cases for daytime migration are the same:

```
If (WQCBM3D_ri(k) > rins_f .and. WQCBM3D_ri(k+1) <= rins_f) Then
!'                                          should be >
    p42f =  expression includes source from above, not below
    So there is (or should be) too much light in this segment and too
much light below.

Else If (WQCBM3D_ri(k) > rins_f .and. WQCBM3D_ri(k+1) <= rins_f) Then
    p42f =  expression includes source from above and below
    So there is too much light in this segment and too little light
below.

Else If (WQCBM3D_ri(k) <= rins_f .and. WQCBM3D_ri(k-1) > rins_f) Then
    p42f =  expression includes source from above and below
    So there is too little light in this segment and too much light
above.


Else If (WQCBM3D_ri(k) <= rins_f .and. WQCBM3D_ri(k-1) <= rins_f) then
    p42f =  expression includes source from below, not above
    So there is too little light in this segment and too little light
above.

End If
```

The comments about the diatom code (grazing option 1, ammonia limitation, and growth rate calculation) apply also to the dinoflagellate code.

Final review – The logic for settling and diel vertical migration has been corrected as recommended:

```
!Ambrose Bug Fix
If (WQCBM3D_ri(k) > rins_f .and. WQCBM3D_ri(k+1) > rins_f) Then
      p42f =  (-WQCBM3D_vlight(n,k)    - WQCBM3D_VTemp(n,k)   -
vgrav)*c(n,k,  I_DFP)/d1 + &
                ( WQCBM3D_vlight(n,k-1)  + WQCBM3D_VTemp(n,k-1) +
vgrav)*c(n,k-1,I_DFP)*SetTop/dzk(k-1)
```

The constant grazing rate is now temperature-corrected:

```
!Ambrose Bug Fix
Case(ConstantGrazing)
      ggraze_f = kgmicro_f*thkt_f**temp20 + kgmacro_f*thkt_f**temp20
```

Sediment Fluxes

No errors were found in this section of code. This subroutine checks out if input fluxes are in g/m^2-day and if SOD fluxes are referenced to 11 C (not 20 C).

GenAlgae

GenAlgae implements ngamcs phytoplankton types as state variables. It also calculates the effect of these phytopankton processes on state variables in WQCBM and WQADD.

Depth-Related Solar Radiation

Errors were found in this section of code. Since it is not being used now, a thorough review and resolution of the issues was not completed.

To calculate light attenuation, it looks like overall depth to layer mid-point is used along with some kind of depth-averaged Chl concentration from SumPhyt/cchl(id), where SumPhyt is depth-summed phytC per unit area at and above the current layer. Newer code in WQCBM accomplish this in a much more transparent and correct manner.

Temperature Limit

This section calculates the temperature correction factor GAM3D_ft using the optimum temperature formulation. This code was correct, but used incorrectly in the algal growth code where the optimum temperature limit was multiplied by the theta temperature correction formulation. I recommend modifying this section to calculate GAM3D_ft using the theta approach if input GAM3D_Tht_k1c is greater than 1, and otherwise calculate GAM3D_ft using the present Topt code. Suggested code details are provided in the Phytoplankton Production section below and in the Appendix.

Final review **–** The suggested code is incorporated into GenAlgae Version 4.

Light Limit

No errors were found in this section of code. Three light limitation expressions are provided – Half Saturation, Smith, and Steele. The code was verified against equations presented in the QUAL2K documentation.

Saline Limit

One error was found in this section of code, which calculates salt toxicity to growth, GAM3D_ftox, as either a unitless multiplication factor (Case 1) or as a reduction in the growth rate constant (Case 2). There was a units mismatch between GAM3D_ftox calculated in Case 2 (1/sec) and its use in the phytoplankton growth equation (1/day):

```
!dimensional 1/sec
Case(2)
      GAM3D_ftox(id) = GAM3D_stf(id,RegionNum)*0.5*(1.0 + dtanh(stox-
GAM3D_khst(id,RegionNum)))
```

Final review – This mismatch was corrected with an update the units of GAM3D_stf, an input constant:

```
!Ambrose Bug Fix:
GAM3D_stf(i,j) = RateConversion(GAM3D_stf(i,j), Units)
```

Nutrient Limit

No errors were found in this section of code. This section calculates nitrogen and phosphorus limitation factors based on standard Michaelis-Menton expressions. If the input half-saturation constant GAM3D_khn or GAM3D_khp is 0 then that nutrient factor is set to 1 (a good feature). The overall nutrient limitation factor GAM3D_fn is the minimum of the nitrogen and phosphorus limitation factors.

Phytoplankton Production

Three errors were found in this section of code; they have been reported and fixed.

Coding error – There was a units mismatch for salinity toxicity Case 2 (see Saline Limit section above).

Final review – As reported in the Saline Limit section above, this units mismatch was corrected with an update the units of GAM3D_stf, an input constant:

```
!Ambrose Bug Fix:
GAM3D_stf(i,j) = RateConversion(GAM3D_stf(i,j), Units)
```

Coding error – The second error was that AlgaeGrwthNetLimit was not calculated when salinity toxicity option 2 was chosen (see "Else If" section of code below):

```
If(GAM3D_stMethod(id,RegionNum) == 1) Then
      If(UseLOTTMethod == 1) Then
         AlgaeGrwthNetLimit =
      (dmin1(GAM3D_fn(id),GAM3D_fi(id),GAM3D_ft(id),GAM3D_ftox(id)))
      Else
         AlgaeGrwthNetLimit =
         GAM3D_fn(id)*GAM3D_fi(id)*GAM3D_ft(id)*GAM3D_ftox(id)
      End if
      GAM3D_gp(id)   =  GAM3D_gp(id)*AlgaeGrwthNetLimit
Else If(GAM3D_stMethod(id,RegionNum) == 2) Then
      GAM3D_gp(id)   =
      GAM3D_gp(id)*GAM3D_fn(id)*GAM3D_fi(id)*GAM3D_ft(id) -
      GAM3D_ftox(id)
End If
```

Final review – This error was corrected by adding the calculation of AlgaeGrwthNetLimit to the salinity toxicity option 2 section of code:

```
Else If(GAM3D_stMethod(id,RegionNum) == 2) Then
  AlgaeGrwthNetLimit = GAM3D_fn(id)*GAM3D_fi(id)*GAM3D_ft(id)
```

```
  GAM3D_gp(id) = GAM3D_k1c(id,RegionNum)*AlgaeGrwthNetLimit -
GAM3D_ftox(id)
End If
```

Coding error – The third error was using both the theta temperature correction factor and the Topt temperature correction factor in the phytoplankton growth equation. I recommend coding the growth temperature correction term GAM3D_ft to use either theta correction or Topt correction, not both. Here is recommended coding for three sections of code.

First section (beginning on line 615):

```
                  !
!Temperature effects
GAM3D_ft = 1.0
If(GAM3D_UseTempLimit(RegionNum) == 1) Then
   Do id = 1, ngamcs
      IF (GAM3D_Tht_k1c(id,RegionNum) .gt. 1.0) THEN
         GAM3D_ft(id) = GAM3D_Tht_k1c(id,RegionNum)**Temp20
      ELSE
         If (c(n,k,I_Temp) < GAM3D_tm(id,RegionNum)) Then
            GAM3D_ft(id) = exp(-GAM3D_ktg1(id,RegionNum)*(c(n,k,I_Temp)
- GAM3D_tm(id,RegionNum))**2)
         Else
            GAM3D_ft(id) = exp(-
GAM3D_ktg2(id,RegionNum)*(GAM3D_tm(id,RegionNum) - c(n,k,I_Temp))**2)
         End If
      END
  End Do
End If
!
```

Second section (beginning on line 647):

```
!Production
Do id = 1, ngamcs
   !GAM3D_gp(id)    =
GAM3D_k1c(id,RegionNum)*GAM3D_Tht_k1c(id,RegionNum)**Temp20  !(remove
this line)
   If(GAM3D_stMethod(id,RegionNum) == 1) Then
      If(UseLOTTMethod == 1) Then
         AlgaeGrwthNetLimit =
(dmin1(GAM3D_fn(id),GAM3D_fi(id),GAM3D_ft(id),GAM3D_ftox(id)))
      Else
         AlgaeGrwthNetLimit =
GAM3D_fn(id)*GAM3D_fi(id)*GAM3D_ft(id)*GAM3D_ftox(id)
      End if
      GAM3D_gp(id)    =  GAM3D_k1c(id,RegionNum)*AlgaeGrwthNetLimit
   Else If(GAM3D_stMethod(id,RegionNum) == 2) Then
      AlgaeGrwthNetLimit = GAM3D_fn(id)*GAM3D_fi(id)*GAM3D_ft(id)
      GAM3D_gp(id)    =  GAM3D_k1c(id,RegionNum)*AlgaeGrwthNetLimit -
GAM3D_ftox(id)
   End If
```

Third section (beginning on line 677):

```
   !Temperature limitation and effect factor for GAM in cell n layer k
```

```
   GAM_3D_T_lim(m1(i,j),k,id) = GAM3D_ft(id)
```

Final review – This error fixed by adopting the recommended coding:
First section (beginning on line 604):

```
!Temperature effects
GAM3D_ft = 1.0
If(GAM3D_UseTempLimit(RegionNum) == 1) Then
      Do id = 1, ngamcs
            If (GAM3D_Tht_k1c(id,RegionNum) .gt. 1.0) Then
            GAM3D_ft(id) = GAM3D_Tht_k1c(id,RegionNum)**Temp20
            Else
                  If (c(n,k,I_Temp) < GAM3D_tm(id,RegionNum)) Then
                        GAM3D_ft(id) = exp(-
GAM3D_ktg1(id,RegionNum)*(c(n,k,I_Temp) - GAM3D_tm(id,RegionNum))**2)
                  Else
                        GAM3D_ft(id) = exp(-
GAM3D_ktg2(id,RegionNum)*(GAM3D_tm(id,RegionNum) - c(n,k,I_Temp))**2)
                  End If
            End If
      End Do
End If
```

Second section (beginning on line 639):

```
!Production
Do id = 1, ngamcs
      !
      !Ambrose Bug Fix
      If(GAM3D_stMethod(id,RegionNum) == 1) Then
            If(UseLOTTMethod == 1) Then
```

Third section (beginning on line 672):

```
!Temperature limitation and effect factor for GAM in cell n layer k
!Ambrose Bug Fix
GAM_3D_T_lim(m1(i,j),k,id) = GAM3D_ft(id)
```

Water Quality Effects from Phytoplankton

Two errors were found in this section of code, and one recommendation is offered.

Coding error – The first error was multiplying the grazing term GAM3D_ggraze by algal concentration in the calculation of five mass balance terms – POCFFromAlg, POCSFromAlg, POCRFromAlg, PONFromAlg, POPFromAlg. The variable GAM3D_ggraze is already in units of [gC/m^3-day]. Algal concentration, c(n,k,I_GAM), should be removed from these equations in GenAlgae Version 4:

```
POCFFromAlg  =  GAM3D_dd(id)*fd9f*(1 -
GAM3D_foc(id,RegionNum))*c(n,k,I_GAM(id))  + &
                GAM3D_ggraze(id)*fg9f*c(n,k,I_GAM(id))

h(n,k,I_OC_P_F) = h(n,k,I_OC_P_F) + POCFFromAlg*CellVolume
```

```
POCSFromAlg  =  GAM3D_dd(id)*fd9s*(1 -
GAM3D_foc(id,RegionNum))*c(n,k,I_GAM(id))   + &
                    GAM3D_ggraze(id)*fg9s*c(n,k,I_GAM(id))

h(n,k,I_OC_P_S) = h(n,k,I_OC_P_S) + POCSFromAlg*CellVolume

POCRFromAlg =   GAM3D_dd(id)*fd9r*(1 -
GAM3D_foc(id,RegionNum))*c(n,k,I_GAM(id))   + &
                  GAM3D_ggraze(id)*fg9r*c(n,k,I_GAM(id))

h(n,k,I_OC_P_R) = h(n,k,I_OC_P_R) + POCRFromAlg*CellVolume
!End SP 05/16/2008

…

!SP 05/16/2008 Adding source to ON_P and OP_P from algae death and
excretion
PONFromAlg   =  GAM3D_dd(id)*anc*(1.0 -
GAM3D_fon(id,RegionNum))*c(n,k,I_GAM(id)) + &
                       GAM3D_ggraze(id)*anc*c(n,k,I_GAM(id))

h(n,k,I_ON_P) = h(n,k,I_ON_P) + PONFromAlg*CellVolume

POPFromAlg    = GAM3D_dd(id)*apc*(1 -
GAM3D_fop(id,RegionNum))*c(n,k,I_GAM(id)) + &
                       GAM3D_ggraze(id)*apc*c(n,k,I_GAM(id))

h(n,k,I_OP_P) =  h(n,k,I_OP_P) + POPFromAlg*CellVolume
!End SP 05/16/2008 Adding source to ON_P and OP_P from algae death and
excretion
```

Final review – This error was corrected by removing algal concentration, c(n,k,I_GAM), from the grazing term in the equations for POCFFromAlg, POCSFromAlg, POCRFromAlg, PONFromAlg, and POPFromAlg in GenAlgae Version 4:

```
!Ambrose Bug Fix
POCFFromAlg  =  GAM3D_dd(id)*fd9f*(1 -
GAM3D_foc(id,RegionNum))*c(n,k,I_GAM(id))  + &
                 GAM3D_ggraze(id)*fg9f

h(n,k,I_OC_P_F) = h(n,k,I_OC_P_F) + POCFFromAlg*CellVolume

POCSFromAlg  =  GAM3D_dd(id)*fd9s*(1 -
GAM3D_foc(id,RegionNum))*c(n,k,I_GAM(id))   + &
                    GAM3D_ggraze(id)*fg9s

h(n,k,I_OC_P_S) = h(n,k,I_OC_P_S) + POCSFromAlg*CellVolume

POCRFromAlg =   GAM3D_dd(id)*fd9r*(1 -
GAM3D_foc(id,RegionNum))*c(n,k,I_GAM(id))   + &
                  GAM3D_ggraze(id)*fg9r

h(n,k,I_OC_P_R) = h(n,k,I_OC_P_R) + POCRFromAlg*CellVolume
```

…

```
!Ambrose Bug Fix
PONFromAlg   =  GAM3D_dd(id)*anc*(1.0 -
GAM3D_fon(id,RegionNum))*c(n,k,I_GAM(id)) + &
                        GAM3D_ggraze(id)*anc
!
![gN/m3-day]  =       [1/day] * [gN/gC] * [] * [gC/m3]
!               + [gC/m3/day] * [gN/gC] * [] * [gC/m3]

h(n,k,I_ON_P) = h(n,k,I_ON_P) + PONFromAlg*CellVolume
![gN/day]    =    [gN/day]   +  [gN/m3-day]  *  [m3]
!
!Ambrose Bug Fix
POPFromAlg    = GAM3D_dd(id)*apc*(1 -
GAM3D_fop(id,RegionNum))*c(n,k,I_GAM(id)) + &
                        GAM3D_ggraze(id)*apc
!
![gP/m3-day]  =      [1/day] * [gP/gC] * [] * [gC/m3]
!              + [gC/m3/day] * [gP/gC] * [] * [gC/m3]

h(n,k,I_OP_P) =  h(n,k,I_OP_P) + POPFromAlg*CellVolume
![gP/day]     =     [gP/day]   [gP/m3-day]  *  [m3]
```

Coding error – The second error was allocating a fraction of algal death (MalgDeath) to refractory dissolved organic matter (I_RDOM equation in red below). The variable "Pam" is the fraction of algal death converted into POM, and is allocated to labile particulate organic matter (I_LPOM). including refractory dissolved organic matter from algal death (variable RDOM). The h(n,k,I_RDOM) equation in red should be removed.

```
h(n,k,I_LDOM) = h(n,k,I_LDOM) + ((1.0 - Pam)*MAlgDeath + MAlgExrec) *
CellVolume
h(n,k,I_RDOM) = h(n,k,I_RDOM) + (       Pam)*MAlgDeath*CellVolume
h(n,k,I_LPOM) = h(n,k,I_LPOM) + (       Pam)*MAlgDeath*CellVolume
```

Final review – The second error was corrected by commenting out the h(n,k,I_RDOM) equation in GenAlgae Version 4:

```
!Ambrose bug fix
!h(n,k,I_RDOM) = h(n,k,I_RDOM) + (       Pam)*MAlgDeath*CellVolume
h(n,k,I_LPOM) = h(n,k,I_LPOM) + (       Pam)*MAlgDeath*CellVolume
!
!gm/sec
```

Recommendation – In addition to the error corrections, another change is recommended. The expession for photosynthetic DO production is unconventional (see discussion in Dissolved Oxygen section of WQCBM above). I recommend using the more conventional WASP equation for growth. This recommendation was completed as an option in GenAlgae Version 4:

Final review – The recommended expession for photosynthetic DO production was added as an option in GenAlgae Version 4:

```
!Ambrose Bug Fix
If(UseLOTTMethod == 1) Then
!
   !use the original LOTT study method
   DOFromAlg    =   GAM3D_gp(id)*(32./12. + &
                    0.8*32./12.*(1. -  GAM3D_pnh3(id)) + &
                    0.2*32./12.*(     GAM3D_pnh3(id)))*c(n,k,I_GAM(id))
Else
   !use the WASP EUTRO method adapted from WASP DISSOXYG.FOR
   !note: phyto biomass is in mgC/L = gC/m^3
   !growth of phytoplankton using CO2 and NH3
   DOFromAlg = GAM3D_pnh3(id)*GAM3D_gp(id) * c(n,k,I_GAM(id)) * 32./12.
   !growth of phytoplankton using CO2 and NO3 (2NO3 = 2NH3 + 3O2)
   DOFromAlg = DOFromAlg + (1. - GAM3D_pnh3(id)) * GAM3D_gp(id) *
c(n,k,I_GAM(id)) * 32. * (1./12. + 1.5 * anc / 14.)
End If
```

Zooplankton Grazing

The zooplankton grazing code has no errors and is in good shape. Three options are implemented – constant grazing, linear grazing, and density dependent grazing. Note that constant grazing includes temperature correction, unlike the equivalent code in WQCBM.

WQADD

WQADD implements an additional set of state variables to WQCBM simulations:

1. Labile Dissolved Organic Matter, I_LDOM [g /m^3]
2. Refractory Dissolved Organic Matter, I_RDOM [g /m^3]
3. Labile Particulate Organic Matter, I_LPOM [g /m^3]
4. Refractory Particulate Organic Matter, I_RPOM [g /m^3]
5. Total suspended solids, I_SSS [gD/m^3]
6. Total Inorganic Carbon, I_TIOC [g C/m^3]
7. Total Alkalinity, I_ALKL [g CaCO3/m^3]
8. pH, I_PH [g /m^3]
9. Conductivity, I_COND [g /m^3]
10. Bottom Algae, I_MPHYT [g D/m^2]
11. Bottom Algal Nitrogen, INb [mg N/m^2]
12. Bottom Algal Phosphorus IPb [mg P/m^2]

The time units in WQADD appear to be sec, and most derivatives are in [g/sec].

Total Suspended Solids

No errors were found in this section of code. Settling velocity is presumably calculated in function ComputeSettlingVelcityWQDPM, but this function is not found in WQADD, WQCBM, or GenAlgae. It must be in GEMSS main code. The term

WQADD3DA_cnss(I_SSS-ncWQADDSt+1,WQADD3DRegionStatus(i,j)+1) is a switch used in the TSS derivative, but the terms in the argument are unclear to me.

Organic Matter

Several errors were found in this section of code.

Coding error – Four derivatives need to be multiplied by CellVolume, including the derivatives for Labile Dissolved Organic Matter, Refractory Dissolved Organic Matter, Labile Particulate Organic Matter, and Refractory Particulate Organic Matter:

```
h(n,k,I_LDOM)  = h(n,k,I_LDOM) + (PhytoDeath*(1.0 - Pam) +
PhytoExcretion - &
                  LDOMDecay - L2RDOMTransfer)

h(n,k,I_RDOM)  = h(n,k,I_RDOM) + (L2RDOMTransfer - RDOMDecay)

h(n,k,I_LPOM)  = h(n,k,I_LPOM) + (PhytoDeath*Pam - LPOMDecay -
L2RPOMTransfer - LPOMSettling -  &
          LPOMDissolution)

h(n,k,I_RPOM)  = h(n,k,I_RPOM) + (L2RPOMTransfer - RPOMDecay -
RPOMSettling - &
          RPOMDissolution)
```

Final review – The error in these four derivatives has been corrected:

```
!Labile Dissolved Organic Matter
If(acc(I_LDOM) == 1 .and. WQADD3DA_cnss(I_LDOM-ncWQADDSt+1,rdd) == 1)
Then
      LDOMDecay      = K_LDOM*Tht_K_LDOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_LDOM)
      L2RDOMTransfer = K_L2RDOMTR*c(n,k,I_LDOM)
      h(n,k,I_LDOM)  = h(n,k,I_LDOM) + (PhytoDeath*(1.0 - Pam) +
PhytoExcretion - &
                         LDOMDecay - L2RDOMTransfer)*CellVolume
End If
!
!Refractory Dissolved Organic Matter
If(acc(I_RDOM) == 1 .and. WQADD3DA_cnss(I_RDOM-ncWQADDSt+1,rdd) == 1)
Then
      RDOMDecay      = K_RDOM*Tht_K_RDOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_RDOM)
      h(n,k,I_RDOM)  = h(n,k,I_RDOM) + (L2RDOMTransfer -
RDOMDecay)*CellVolume
End If
!
!Labile Particulate Organic Matter
If(acc(I_LPOM) == 1 .and. WQADD3DA_cnss(I_LPOM-ncWQADDSt+1,rdd) == 1)
Then
      LPOMDecay      = K_LPOM*Tht_K_LPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_LPOM)
      L2RPOMTransfer = K_L2RPOMTR*c(n,k,I_LPOM)
```

```
      LPOMSettling   = vs_LPOM*c(n,k,I_LPOM)/d1 - SetTop*vs_LPOM*c(n,k-
1,I_LPOM)/dzk(k-1)
      LPOMDissolution= kds_LPOM*Tht_kds_LPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_LPOM)

      h(n,k,I_LPOM)  = h(n,k,I_LPOM) + (PhytoDeath*Pam - LPOMDecay -
L2RPOMTransfer - LPOMSettling -  &
                                LPOMDissolution)*CellVolume
End If
!
!Refractory Particulate Organic Matter
If(acc(I_RPOM) == 1 .and. WQADD3DA_cnss(I_RPOM-ncWQADDSt+1,rdd) == 1)
Then
      RPOMDecay      = K_RPOM*Tht_K_RPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_RPOM)
      L2RPOMTransfer = K_L2RPOMTR*c(n,k,I_RPOM)
      RPOMSettling   = vs_RPOM*c(n,k,I_RPOM)/d1 - SetTop*vs_RPOM*c(n,k-
1,I_RPOM)/dzk(k-1)
      RPOMDissolution= kds_RPOM*Tht_kds_RPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_RPOM)
      h(n,k,I_RPOM)  = h(n,k,I_RPOM) + (L2RPOMTransfer - RPOMDecay -
RPOMSettling - &
                                RPOMDissolution)*CellVolume
End If
```

Coding Error - Labile Dissolved Organic Matter section: in the L2RDOMTransfer equation, K_L2RDOMTR should be multiplied by c(n,k,I_LDOM), not c(n,k,I_LPOM):

```
L2RDOMTransfer = K_L2RDOMTR*c(n,k,I_LPOM)
```

Final review – This is corrected :

```
L2RDOMTransfer = K_L2RDOMTR*c(n,k,I_LDOM)
```

Coding Error - Labile Particulate Organic Matter section: the LPOMDissolution equation should be multiplied by LPOM concentration, c(n,k,I_LPOM):

```
LPOMDissolution= kds_LPOM*Tht_kds_LPOM**(c(n,k,I_Temp) - 20.0)
```

Final review – This is corrected:

```
LPOMDissolution= kds_LPOM*Tht_kds_LPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_LPOM)
```

Coding Error - Refractory Particulate Organic Matter section: in the L2RPOMTransfer equation, K_L2RPOMTR should be multiplied by c(n,k,I_LPOM), not c(n,k,I_RPOM):

```
L2RPOMTransfer = K_L2RPOMTR*c(n,k,I_RPOM)
```

Final review – This is corrected:

```
!Labile Particulate Organic Matter
```

```
If(acc(I_LPOM) == 1 .and. WQADD3DA_cnss(I_LPOM-ncWQADDSt+1,rdd) == 1)
Then
      LPOMDecay      = K_LPOM*Tht_K_LPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_LPOM)
      !GUI Change
      L2RPOMTransfer = K_L2RPOMTR*Tht_K_L2RPOMTR**(c(n,k,I_Temp) -
20.0)*c(n,k,I_LPOM)
      LPOMSettling   = vs_LPOM*c(n,k,I_LPOM)/d1 - SetTop*vs_LPOM*c(n,k-
1,I_LPOM)/dzk(k-1)
      LPOMDissolution= kds_LPOM*Tht_kds_LPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_LPOM)



      h(n,k,I_LPOM)  = h(n,k,I_LPOM) +
      (PhytoDeath*Pam*PhytoDandEinWQADD/r_OMC - LPOMDecay -
      L2RPOMTransfer - LPOMSettling -  &
      LPOMDissolution)*CellVolume
End If
```

Coding Error - Refractory Particulate Organic Matter section: the RPOMDissolution equation should be multiplied by RPOM concentration, c(n,k,I_RPOM):

```
RPOMDissolution= kds_RPOM*Tht_kds_RPOM**(c(n,k,I_Temp) - 20.0)
```

Final review – This is corrected:

```
RPOMDissolution= kds_RPOM*Tht_kds_RPOM**(c(n,k,I_Temp) -
20.0)*c(n,k,I_RPOM)
```

Coding Error - OC, PO4 and NH3 sources and sinks from OM section: in the equation for h(n,k,I_TIOC), RPOMDecay is used twice; the first instance should be LPOMDecay:

```
h(n,k,I_TIOC) = (h(n,k,I_TIOC) +  (LDOMDecay*r_OMC + RDOMDecay*r_OMC &
                               +  RPOMDecay*r_OMC + RPOMDecay*r_OMC &
       !'                         ^^^^^^^^^
```

Final review – This is corrected:

```
h(n,k,I_TIOC) = (h(n,k,I_TIOC) +  (LDOMDecay*r_OMC + RDOMDecay*r_OMC &
                               +  LPOMDecay*r_OMC + RPOMDecay*r_OMC &
                                                               +
```

Coding Error - OC, PO4 and NH3 sources and sinks from OM section: in the equation for h(n,k,I_PO4), RPOMDecay is used twice; the first instance should be LPOMDecay:

```
If(acc(I_PO4) == 1) h(n,k,I_PO4)  =  (h(n,k,I_PO4)  +
acc(I_PO4)*(LDOMDecay*r_OMP + RDOMDecay*r_OMP &
                     +  RPOMDecay*r_OMP + RPOMDecay*r_OMP)*CellVolume)
      !'                 ^^^^^^^^^
```

Final review – This is corrected:

```
If(acc(I_PO4) == 1) h(n,k,I_PO4)  =  (h(n,k,I_PO4)  +
acc(I_PO4)*(LDOMDecay*r_OMP + RDOMDecay*r_OMP &
                      +  LPOMDecay*r_OMP + RPOMDecay*r_OMP)*CellVolume)
```

Possible Error - Question about settling units for LPOM and RPOM: for terms vs_LPOM and vs_RPOM, the settling units are [1/sec]. These terms, however, are labelled "settling velocity," which implies units of L/T and the need to divide by cell depth. This needs to be checked.

```
  LPOMSettling   = vs_LPOM*c(n,k,I_LPOM)
  RPOMSettling   = vs_RPOM*c(n,k,I_RPOM)
!'[g/m3-sec]     = [1/sec]   * [g/m3]
```

Final review – This is corrected:

```
LPOMSettling   = vs_LPOM*c(n,k,I_LPOM)/d1 - SetTop*vs_LPOM*c(n,k-
1,I_LPOM)/dzk(k-1)

RPOMSettling   = vs_RPOM*c(n,k,I_RPOM)/d1 - SetTop*vs_RPOM*c(n,k-
1,I_RPOM)/dzk(k-1)
```

pH-TIOC-Alkalinity

Two coding errors and two formulation omissions were found in this section of code. In addition, there are some suggestions and warnings.

Suggestions – Consider using more precision in molecular weight values, such as [gC/mol], 50.450 [gCaCO3/eq], 14.007 [gN/mol], and 30.974 [gP/mol].

The calculation of CO3 is corrrect, but not straightforward:

```
        !Carbonate concentration
        CO3 = TIOC/(1.0 + ph_hh**2.0/(ph_k1*ph_k2) + ph_hh/ph_k2)
     !'[molC/L]
```

For ease of verification, I recommend calculating and using an “alp2” term in the same manner as “alp0” and “alp1” were used for CO2 and HCO3:

```
!'     alp2 = ph_k1*ph_k2 / (ph_hh**2.0 + ph_k1*ph_hh + ph_k1*ph_k2)
!' or
!'     alp2 = 1 - alp0 - alp1
!'      CO3 = alp2*TIOC
```

For computational efficiency, I recommend calculating the denominator for use in alp0, alp1, and alp2, such as:

```
!' DENOM = (ph_hh**2.0 + ph_k1*ph_hh + ph_k1*ph_k2)
```

```
!' alp2 = ph_k1*ph_k2 / DENOM
```

Errors and warnings – In the calculation of TIOC reaeration, k2T is used but not located in WQADD. This must be the temperature-corrected oxygen reaeration velocity in [m/sec], and kacT would be the CO2 reaeration velocity in [m/sec]. Without documentation, it is not clear what the term "ACF" is or what pCO2SetMethod=1 is implementing. There is an error in calculating CO2Sat using pCO2SetMethod=2 (or 0). It appears that the term kacT was used instead of a term such as KHCO2 [moleC/L-atm].

```
!At the water surface
If(k == kt) Then
    kacT = (32.0/44.0)**0.25*k2T
 !'[m/sec] = []     * [m/sec]
    ACF  = (1.0 - (el(kt) - z(n     ))/1000.0/44.3)**5.25
    If(pCO2SetMethod == 1) Then
            CO2Sat = 0.286*exp(-0.0314*c(n,k,I_Temp))*ACF
    Else
            CO2Sat = kacT*pCO2
 !'       [moleC/L]=[m/sec]*[atm]
    End If
    h(n,kt,I_TIOC) = h(n,kt,I_TIOC) + kacT*CellArea*(CO2Sat - CO2)
 !' [molC-m3/L-sec]= [molC-m3/L-sec]+ [m/sec] * [m2] * ( [molC/L] )
End If
```

Final review – This error has not been corrected as of 12/19/11:

```
!At the water surface
If(k == kt) Then
      kacT = (32.0/44.0)**0.25*k2T
      ACF  = (1.0 - (el(kt) - z(n   ))/1000.0/44.3)**5.25
!
!Is pCO2 still in units of ppm here? Note that ph_kh is Henryís
constant calculated at line 208

      If(pCO2SetMethod == 1) Then
            CO2Sat = 0.286*exp(-0.0314*c(n,k,I_Temp))*ACF
      Else
            CO2Sat = kacT*pCO2/ 1000000.0
      End If
      h(n,kt,I_TIOC) = h(n,kt,I_TIOC) !+ kacT*CellArea*(CO2Sat - CO2)
End If
```

First, there appears to be confusion about the units of pCO2, with the comment line questioning whether they might be [ppm]. The units of CO2Sat must be in [moles C/L] to match the units of CO2, which is calculated from the variable TIOC in units of [moles C/L]. Note also that the calculation of CO2Sat still involves the product of pCO2 [moles C/L] and kacT [m/sec]. It appears that the term kacT was used instead of the Henry's Law constant, which the comment indicates is ph_kh. Henry's Law constant must be expressed in units that convert atmospheric pCO2 to aqueous CO2Sat. Note also that when CO2Sat and CO2 are in units of [mole C/L], the TIOC derivative will be in mixed units of [molC-m^3/L-sec].

Warning – The TIOC derivative, h(n,k,I_TIOC), has units of [molC-m3/L-sec] rather than a more conventional [gC/sec] or [molC/sec]. Is the derivative ever converted to [g/sec], or is it used directly in these mixed units? I don't have the solver, and so cannot check whether this mixed-units derivative is being used properly.

Final review – This warning has not been fully addressed as of 12/19/11. At present, it appears that the derivative adds mixed units. The reaeration term (see above) has units of [molC-m^3/L-sec], but other terms added below have more conventional units of [gC/sec]. If the variables CO2Sat and CO2 in the reaeration term are converted from [mole C/L] to [gC/m^3], then this issue will be resolved.

Warning – Note that the units of PhytoResp and PhytoPhoto must be [mgA/m3-sec] for the derivative to work out to these mixed units. In WQCBM, PhytoResp and PhytoPhoto are in units of [gC/m3-sec], not [mgA/m^3-sec].

```
          rcca      = gC/mgA/12.0d+00/1000.0d+00
!'[molC-m3/mgA-L] = [gC]/[mgA]/[gC/molC] / [L/m3]

   h(n,k,I_TIOC) [=] rcca * PhytoResp * CellVolume)
!'[molC-m3/L-sec] = [molC-m3/mgA-L] * [mgA/m3-sec] * [m3]
```

Are PhytoResp and PhytoPhoto ever converted to [mgA/m^3-sec] units?

Final review – This warning has been addressed as of 12/19/11 by removing the term rcca.

```
!Sources and sinks due to Phytoplankton
    !h(n,k,I_TIOC) = acc(I_TIOC)*(h(n,k,I_TIOC) + (rcca*PhytoResp -
rcca*PhytoPhoto)*CellVolume)
    h(n,k,I_TIOC) = acc(I_TIOC)*(h(n,k,I_TIOC) + (PhytoResp -
PhytoPhoto)*CellVolume)
```

Note that the units of this term are now [gC/sec].

Coding Error – In the TIOC derivative, the bottom algae term should be multiplied by CellArea rather than CellVolume:

```
h(n,k,I_TIOC) = acc(I_TIOC)*(h(n,k,I_TIOC) + WQADD3DA_cnss(I_SSS-
ncWQADDSt+1,rdd)*(rccd*BotAlgResp - rccd*BotAlgPhoto)*CellVolume)
```

Final review – This error has not been corrected as of 5/28/11. But note the different units for the TIOC derivative, which work out to be [mol C-m^3/sec] (highlighted units comments below are my additions):

```
      rccd        = gC/gD/12.0d+00/1000.0d+00
! [molC-m3/gD-L] = [gC]/[gD]/[gC/molC] / [L/m3]

!Sources and sinks due to bottom algae
!SP 05/27/2008
!Bob Ambrose: 06-14-2011: CellVolume changed to CellArea
```

```
      h(n,k,I_TIOC) = acc(I_TIOC)*(h(n,k,I_TIOC) + WQADD3DA_cnss(I_SSS-
ncWQADDSt+1,rdd)*(rccd*BotAlgResp - rccd*BotAlgPhoto)*CellArea)

!'[molC-m3/sec] = [molC-m3/gD-L] [gD/m2-sec] [m2]
```

Formulation omissions – The TIOC derivative is missing a CBOD oxidation term.

The alkalinity derivative is missing phytoplankton PO_4 photosynthesis and respiration terms.

Final review – The CBOD omission in the TIOC derivative has been addressed as of 12/14/11. I cannot verify where the new term, PhytoDandEinWQADD, is set. This appears to be a switch for using either CBOD (when the term is set to 0) or LDOM (when the term is set to 1). Note that the units for the TIOC derivative here work out to [gC/sec].

```
!SP 08/31/2011 Skip OM decay if CBOD constituents are active
If(acc(I_TIOC) == 1 .and. WQADD3DA_cnss(I_TIOC-ncWQADDSt+1,rdd) ==
1)Then
      h(n,k,I_TIOC) = h(n,k,I_TIOC) + (CBODDecay*r_CBODC*(1-
PhytoDandEinWQADD))*CellVolume
      h(n,k,I_TIOC) = h(n,k,I_TIOC) + (LDOMDecay*r_OMC +
RDOMDecay*r_OMC + LPOMDecay*r_OMC +
RPOMDecay*r_OMC)*PhytoDandEinWQADD*CellVolume

End If
```

The phytoplankton PO4 photosynthesis and respiration terms omission in the alkalinity derivative have not been addressed as of 12/14/11.

Bottom Algae

One coding errors was found in this section of code. There are some suggestions and warnings.

Suggestion – Solar radiation is converted from Ly/day to W/m^2 using a conversion factor of 4.183076 J/cal-thermochemical. The conversion factor quoted in "onlineconversion.com" gives the conversion factor of 4.184 J/cal-thermochemical. The conversion factor here should be consistent with what is used in the rest of the code.

```
  !Light Limitation
SolRadInLPerDay = PAR*SolRad/((4.183076*100*100)/86400.0) !Cal/cm^2-day
```

Warning and suggestion – Bottom light is calculated using surface solar radiation, a light attenuation factor, and depth. The light attenuation coefficient, ke, is for layer "k", and is function of phytoplankton concentration, which varies with layer. A single value of ke, then, will not accurately account for light attenuation through the depth (unless the bottom algae occur at the bottom of the surface layer). I recommend using calculated bottom light from WQCBM, which is RBotPAR(n,k) for k=k0(i,j).

```
!Attenuation coeffcient from WQM models
ke = LEC(n,k)

BotLight = SolRadInLPerDay*exp(-ke*Depth)
```

Final review – This suggestion has been adopted as an option if the WQCMB model is being used:

```
!Light Limitation
If(WaterQuality) Then
      BotLight = RBotPAR(n,k0(i,j))
Else
      SolRadInLPerDay = PAR*SolRad/((4.183076*100*100)/86400.0)
      BotLight = SolRadInLPerDay*exp(-ke*Depth)
End If
```

Coding error – DO from bottom algae has an error in the second term. In the equation below, the first term represents the production of DO from photosynthesis using *both* NH_4 and NO_3 (thus it is not multiplied by Pab). The seccond term should represent evolution of DO from the reduction of NO_3 to NH_4 prior to photosynthesis. Using the factor 138/107, however, the second term presently accounts for the production of DO from the reduction of NO_3 and the subsequent production of DO with photosynthesis. The equation as written double counts the DO evolution from photosynthesis using NO_3. This could be fixed by multiplying the first term by "Pab" or by adapting the WASP code, provided below.

```
!gO2/s
DOFromBotAlg = (rod*BotAlgPhoto + rod*BotAlgPhoto*(1-Pab) * 138/107)
*CellArea
```

The WASP code is:

```
!Growth of phytoplankton using CO2 and NH3
 SRPeriA = RBotAlgPhoto * ROC/ADC
!            gD/m3-day     gO2/gD

!Evolution of oxygen with the internal reduction of NO3 to NH4
 SRPeriB = RBotAlgPhoto * (ANC/ADC)*(1.-PrefAMF) * (1.5*32./14.)
!            gD/m3-day     gN/gD     gNO3N/gN      gO2/gNO3N
!note that (1 + 1.5 * ANC * 12/14) = 1.289 = 138/107 if ANC = 9/40
```

The term SRPeriB in the WASP code can be manipulated to give the following expression for the second term in the WQADD equation:

```
 SRPeriB = rod * BotAlgPhoto * (1-Pab) * (1.5*ANC*12/14)
         = rod * BotAlgPhoto * (1-Pab) * 0.289
 !  0.289 is obtained if ANC = 9/40
```

Final review – This error has been corrected:

```
!Dissolved Oxygen Source - Equation 136
```

```
!mgO2/m^2-sec -> mgO2/sec
!gO2/s
DOFromBotAlg = (rod*BotAlgPhoto*Pab + rod * BotAlgPhoto * (1 - Pab) *
138 / 107)*CellArea
```

Observation – Bottom algae use the same constants as WQCBM phytoplankton (diatoms and dinoflagellates) to cycle algal carbon to dissolved organic carbon (foc) and the fraction of this internal loading going to fast CBOD (fd5). Also, it uses the same constants for the N and P stoichiometry. GenAlgae uses separate values for each phytoplankton simulated. It would be better to use values specific for benthic algae.

Final review – The first observation was itself erroneous. Although the variable names are the same as for phytoplankton in WQCBM, a subroutine assigns WQADD values to these variables:

```
Entry WQADD3DSSTerms

      aoc = aoc1

      !SP 05/16/2008
      foc = foc_WQADD;  fd5 = fd5_WQADD;  fd9f = fd9f_WQADD;
      fd9s = fd9s_WQADD;
      fd9r = fd9r_WQADD;  fon = fon_WQADD;  fop = fop_WQADD
      !End SP 05/16/2008
            !
```

Coding error – In the section of code for Case(WQSSTERMS_USING_WQCBM), the particulate OP derivative is equated to particulate organic nitrogen from bottom algae (PONFromBotAlg); that term should use POPFromBotAlg.

```
h(n,k,I_OP_P)    = acc(I_OP_P)*h(n,k,I_OP_P) + PONFromBotAlg *1.0e-03
!' [gP/sec]        =    []       *  [gP/sec]  + [mgN/sec] * [gN/mgN]
!'                              **** ERROR: should be POPFromBotAlg ****
```

Final review – This error has been corrected:

```
Case(WQSSTERMS_USING_WQCBM)
  h(n,k,I_ON_D)= acc(I_ON_D)*(h(n,k,I_ON_D) + DONFromBotAlg*1.0e-03)
  h(n,k,I_OP_D)= acc(I_OP_D)*(h(n,k,I_OP_D) + DOPFromBotAlg*1.0e-03)
      !SP 05/16/2008
  h(n,k,I_CBOD_F) = acc(I_CBOD_F)*h(n,k,I_CBOD_F)  + FCBODFromBotAlg
  h(n,k,I_CBOD_S) = acc(I_CBOD_S)*h(n,k,I_CBOD_S)  + SCBODFromBotAlg
  h(n,k,I_OC_P_F) = acc(I_OC_P_F)*h(n,k,I_OC_P_F)  + POCFFromBotAlg
  h(n,k,I_OC_P_S) = acc(I_OC_P_S)*h(n,k,I_OC_P_S)  + POCSFromBotAlg
  h(n,k,I_OC_P_R) = acc(I_OC_P_R)*h(n,k,I_OC_P_R)  + POCRFromBotAlg
  h(n,k,I_ON_P)   = acc(I_ON_P)*h(n,k,I_ON_P)  + PONFromBotAlg *1.0e-03
  h(n,k,I_OP_P)   = acc(I_OP_P)*h(n,k,I_OP_P)  + POPFromBotAlg *1.0e-03
…
```

Coding error – In three sections of code – Case(WQSSTERMS_USING_WQE5M), Case(WQSSTERMS_USING_WQDPM), and Case(WQSSTERMS_USING_WQCBM) – the dissolved oxygen derivative in [gO_2/sec] is multiplied by the product of (DOFromBotAlg – DOToBotAlg) in [gO_2/sec] and 1.0e-3 [g/mg]. The term 1.0e-3 should be removed.

```
h(n,k,I_DO) = acc(I_DO)*(h(n,k,I_DO)+(DOFromBotAlg-DOToBotAlg)*1.0e-03)
!' [gO2/sec] =    []   *  [gO2/sec] + (     [gO2/sec]    ) * [gO2/mgO2]
!'                              **** ERROR: should omit 1.0e-03 ****
```

Final review – This error has been corrected:

```
Case(WQSSTERMS_USING_WQE5M)
  …
  h(n,k,I_DO) = acc(I_DO)*(h(n,k,I_DO) + (DOFromBotAlg - DOToBotAlg))
  …
!
!WQDPM
Case(WQSSTERMS_USING_WQDPM)
  …
  h(n,k,I_DO) = acc(I_DO)*(h(n,k,I_DO) + (DOFromBotAlg - DOToBotAlg))
  …
!
!WQCBM
  …
  h(n,k,I_DO) = acc(I_DO)*(h(n,k,I_DO) + (DOFromBotAlg - DOToBotAlg))
```

Coding errors – In the section of code for Case(WQSSTERMS_USING_WQDPM), the ON derivative uses terms "acc(I_ON_D)" and "(h(n,k,I_ON_D)." These terms should use index "I_ON" instead of "I_ON_D." The same error occurs in the OP derivative, which should use terms with index "I_OP" rather than "I_OP_D."

```
h(n,k,I_ON)  = acc(I_ON_D)*(h(n,k,I_ON_D)     + DONFromBotAlg*1.0e-03)
!'                  ^^^^^^          ^^^^^^
!'     Should be    I_ON            I_ON            Probable ERROR
!' [gN/sec]  =     []     * [gN/sec]           + [mgN/sec] * [gN/mgN]
!'                 Shouldnt this term include PONFromBotAlg also?

h(n,k,I_OP)  = acc(I_OP_D)*(h(n,k,I_OP_D)     + DOPFromBotAlg*1.0e-03)
!'                  ^^^^^^          ^^^^^^
!'     Should be    I_OP            I_OP            Probable ERROR
!' [gP/sec]  =     []     * [gP/sec]           + [mgP/sec] * [gP/mgP]
!'                 Shouldnt this term include PONFromBotAlg also?
```

Final review – This possible error has been addressed by changing the index consitently to I_ON_D. This is correct if WQDPM separates the organic nitrogen into dissolved (I_ON_D) and particulate variables rather than using a lumped (I_ON) variable.

```
!WQDPM
Case(WQSSTERMS_USING_WQDPM)
   h(n,k,I_ON_D)  = acc(I_ON_D)*(h(n,k,I_ON_D) + DONFromBotAlg*1.0e-03)
   h(n,k,I_OP_D)  = acc(I_OP_D)*(h(n,k,I_OP_D) + DOPFromBotAlg*1.0e-03)
```

Possible formulation errors – In two sections of code – Case(WQSSTERMS_USING_WQE5M) and Case(WQSSTERMS_USING_WQDPMM) – the ON derivative includes the term "DONFromBotAlg," but not "PONFromBotAlg." If state variable I_ON in these models represents total nonliving organic nitrogen, then the derivative should also include "PONFromBotAlg." The same comment applies to the OP derivative in these models.

```
Case(WQSSTERMS_USING_WQE5M)
    h(n,k,I_ON)  = acc(I_ON)*(h(n,k,I_ON)   + DONFromBotAlg*1.0e-03)

    h(n,k,I_OP)  = acc(I_OP)*(h(n,k,I_OP)   + DOPFromBotAlg*1.0e-03)


Case(WQSSTERMS_USING_WQDPM)
    h(n,k,I_ON)  = acc(I_ON_D)*(h(n,k,I_ON_D) + DONFromBotAlg*1.0e-03)


    h(n,k,I_OP)  = acc(I_OP_D)*(h(n,k,I_OP_D) + DOPFromBotAlg*1.0e-03)
```

Final review – This possible error has not been corrected, and so the tranfers of ON and OP from periphyton to the ON and OP pools are inconsistent among these models. In WQE5M, the total ON derivative will not receive particulate ON or OP from bottom algae. In WQDPM, there are no transfers of particulate ON or OP from bottom algae to particulate ON and OP derivatives.

```
!WQE5M
Case(WQSSTERMS_USING_WQE5M)
      h(n,k,I_ON)  = acc(I_ON)*(h(n,k,I_ON)   + DONFromBotAlg*1.0e-03)
      h(n,k,I_OP)  = acc(I_OP)*(h(n,k,I_OP)   + DOPFromBotAlg*1.0e-03)
      …

!
!WQDPM
Case(WQSSTERMS_USING_WQDPM)
      h(n,k,I_ON_D) = acc(I_ON_D)*(h(n,k,I_ON) + DONFromBotAlg*1.0e-03)
      h(n,k,I_OP_D) = acc(I_OP_D)*(h(n,k,I_OP) + DOPFromBotAlg*1.0e-03)
      …

!
!WQCBM
Case(WQSSTERMS_USING_WQCBM)
      h(n,k,I_ON_D) = acc(I_ON_D)*(h(n,k,I_ON_D)+DONFromBotAlg*1.0e-03)
      h(n,k,I_OP_D) = acc(I_OP_D)*(h(n,k,I_OP_D)+DOPFromBotAlg*1.0e-03)
      …

      h(n,k,I_ON_P) = acc(I_ON_P)*h(n,k,I_ON_P) +PONFromBotAlg *1.0e-03
      h(n,k,I_OP_P) = acc(I_OP_P)*h(n,k,I_OP_P) +POPFromBotAlg *1.0e-03
      …
```

Appendix J.2. GEMSS code corrections by Ecology

Ecology conducted a review of the Fortran code for the WQCBM, GAM, and WQADD water quality modules in GEMSS prior to the review that was presented in Appendix K.1 by Bob Ambrose. All of the coding revisions conducted by Ecology were also reviewed by Bob Ambrose in his review. This appendix presents the coding errors that were discovered by Ecology and corrected by ERM.

Changes made after Ecology's 2008 draft report

Provision for optional use of legacy methods

In order to preserve some of the original methods in the model code for legacy applications, and also provide a corrected coding option for this project and future Ecology projects, a new variable was introduced in the WQCBM and GAM modules called UseLOTTMethod. If UseLOTTMethod = 1, then the original method would be used for a particular process. If UseLOTTMethod = 0 then the corrected method would be used. For this project and all future Ecology projects, we set UseLOTTMethod = 0 to use the corrected methods. Therefore, sections of the code that would be executed if UseLOTTMethod = 1 are ignored during model execution for Ecology's applications and can be ignored by external reviewers for the purpose of reviewing the model code that is used for Ecology's project. Ecology does not recommend use of the legacy methods, due to numerous significant errors.

Dinoflagellate dark uptake of nutrients and conservation of mass of N and P

The original code represented the respiration term (dr_f) for the dinoflagellate variable (I_DFP) as a nutrient source term during periods of photosynthesis, and nutrient sink term at night for the nutrient variables (ammonia (I_NH3) and inorganic P (I_PO4)). This process provided dinoflagellates with the ability to take up nutrients at night ("dark uptake" of nutrients). The main problem with this formulation was that the DFP variable (and all other phytoplankton variables in WQCBM and GAM) is assumed to have constant stoichiometric ratios of C:N:P that are specified as model input parameters.

The assumption of constant stoichiometry requires that any nutrient source or sink term associated with the kinetics of DFP must be associated with a corresponding proportional increase or decrease in the biomass of DFP, otherwise the total mass of nutrients in the system would not be conserved. For example, if respiration of DFP at night reduces the concentration of ammonia and phosphate, but there is not a corresponding increase in the N and P content of DFP biomass, then the dark uptake of nutrients represents a leak of the total mass of nutrients from the system which would violate conservation of mass. This problem was compounded by the treatment of respiration as a loss term for biomass of DFP (and the proportional loss of stored N and P contained in that biomass) at the same time that it was treated as a loss term for the nutrient variables with dark uptake. Both of these processes contributed to a leak of N and P from the system because the dark uptake of N and P did not go into a corresponding increase in

the biomass of N and P, and the simultaneous loss of biomass of N and P did not go into the pools of N and P in the nutrient variables.

The first attempt to fix this problem involved changing the respiration term for DFP to a growth term for DFP at night. In other words, during the daytime the respiration of DFP is a source term for nutrients and a corresponding proportional sink term for the biomass of DFP, but at night the respiration of DFP is a sink term for nutrients and a corresponding proportional source term for the biomass of DFP, as follows (lines 607-611 in WQCBM).

```
If(xri_f.gt.0.0) Then
       p42b = - dr_f*c(n,k,I_DFP)    !daytime respiration is a loss term for Dinos
Else
       p42b = dr_f*c(n,k,I_DFP)      !dark respiration is a growth term for Dinos
EndIf
```

Where xri_f is the light limitation factor for growth of DFP (varies between 0 and 1, with 0 representing complete cessation of growth due to lack of light at night, and 1 representing no inhibition of growth due to light.

This change had the advantage of solving the problem of conservation of mass of N and P. However it had the disadvantage of unrealistic representation of the actual metabolism of dinoflagellates, and it also does not account for the source of C that would be assimilated into the biomass of DFP at night. Therefore this modification was abandoned.

The best fix for the problem of conservation of mass with dark uptake of nutrients was to remove this mechanism from the model and always treat respiration as a loss term for the biomass of DFP. In other words, respiration is always treated as a source term for nutrients and a corresponding proportional loss term for the biomass of DFP to maintain conservation of mass. This was accomplished in the Fortran code in WQCBM by changing the conditional in line 607 from “gt” to “ge” as follows, such that the second condition in the If statement would never occur, because the term xri_f is never less than zero:

```
If(xri_f.ge.0.0) Then
       p42b = - dr_f*c(n,k,I_DFP)    !daytime respiration is a loss term for Dinos
Else
       p42b = dr_f*c(n,k,I_DFP)      !dark respiration is a growth term for Dinos
EndIf
```

The same change was applied to the derivatives for nutrients to always treat respiration as a source term in lines 524-528 and lines 577-581, where xri_f is never less than zero.

Light extinction and PAR

The original LOTT method was discovered to contain significant errors related to light limitation calculation:

- total chlorophyll a was not correctly summed across phytoplankton groups
- light extinction was not correctly calculated from total chlorophyll a

- the integrated average light intensity in each layer was not correctly calculated
- total solar radiation was assumed to represent photosynthetically active radiation
- incorrect unit conversions were used to convert solar radiation from W/m^2 to langleys/day
- the formula for the Steele function for light limitation was incorrect.

The corrected calculations for each grid cell layer are documented in lines 384-432 and 460-486 of WQCBM.f90

The revised equation for the light extinction (Ke, m^-1) in the WQCBM module is the following formula:

```
Ke = Ke_a + Ke_b * Chla ^ Ke_c
```

Where Chla is the total phytoplankton chlorophyll a concentration (ug/L) in the water column summed across all phytoplankton groups used in the WQCBM and GAM modules, and Ke_a, Ke_b, and Ke_c are empirical model input parameters. The parameter values Ke_a=0.336, Ke_b=0.0365, and Ke_c=0.64 were used as model inputs in this application estimated based on regression of observed extinction and phytoplankton chlorophyll a data from South Puget Sound.

Earlier model changes through Ecology's 2008 draft report

From 2003 through 2007, Ecology contracted with ERM to perform the following tasks:

- Conversion of the 1997 LOTT model to the latest version of GEMSS.
- Verification of goodness-of-fit relative to the 1997 calibration.
- Capitol Lake setup in GEMSS, linkage with Budd Inlet, and testing.
- Incorporation of a more recent data set (2000-2001 Miller Brewing Company study of Capitol Lake).
- Conduct a workshop on the GEMSS application for Ecology staff.
- Calibration assistance and technical support.

The following additional tasks were performed by ERM:

1. Updated the existing Budd Inlet model setup to include the combined Budd Inlet and Capitol Lake grid connected by the outlet structure at the dam.
2. Calibrated the flow exchange of the combined Capitol Lake/Budd Inlet GEMSS model application to recent data collected during 2003-2004 by Ecology and Thurston County. The simulation period was May 18 – September 30, 2004.
3. Calibrated temperature, DO, and other water quality constituents of the combined Capitol Lake/Budd Inlet GEMSS model application to recent data collected during 2003-2004 by Ecology and Thurston County. The simulation period was May 18 – September 30, 2004.
4. Confirmed flow exchange, temperature, DO, and other water quality constituents of the combined Capitol Lake/Budd Inlet model application with recent data collected during 2000-2001 by CH2M Hill (2001) and Thurston County. The simulation period was April 25 – June 13, 2001.
5. Performed model scenario runs and comparisons for a total of eight scenarios. Four scenarios included the dam, and four scenarios assumed that the dam was not present and that Capitol Lake functioned as an estuary.
6. Documented and delivered of the work products of bullets 1 through 5. A memo report was provided by ERM.

The project was scoped originally to utilize the calibrated and accepted model of Budd Inlet from the Budd Inlet Scientific Study, given the significant resources invested in this model calibration and extensive review by multiple entities. Because that model considered Capitol Lake as a boundary condition, the model domain was extended south to encompass Capitol Lake. However, no recalibration of Budd Inlet was planned.

In 2006, ERM provided the latest calibrated model of Budd Inlet to Ecology along with a comparison to the previous model outputs (Kolluru, 2006). ERM also provided a combined calibrated model of Budd Inlet/Capitol Lake in January 2008.

In the course of review of the calibrated Budd Inlet/Capitol Lake model, the source codes for the Water Quality Carbon-Based Module (WQCBM) were reviewed. The following issues with WQCBM were noted and later corrected by ERM:

1. The increase in zooplankton biomass due to grazing was not included in any of the mass balance equations for particulate organic phosphorus, nitrogen, and carbon. This inclusion was necessary for mass balance to be correct.
2. The particulate organic carbon was expressed in units of oxygen rather than carbon.
3. Grazing of diatoms was incorrectly included twice, while the grazing of dinoflagellates was included only once. Correcting this resulted in an increase in the diatom population and a decrease in dinoflagellates.

With the model code errors corrected, the parameters calibrated in the Budd Inlet Scientific Study were re-evaluated by Ecology to determine if calibration to the 1996-1997 data was still acceptable. Because some differences were noted, Ecology re-calibrated the Budd Inlet region of the model for the 2008 draft report. The re-calibrated model at that stage was used as the tool for evaluating the model scenarios that were presented in the 2008 draft report (Roberts et al. 2008).