**Ryan’s check process of ZooMSS code on Monday 11 January 2020.**

I’m going to work backwards, from “A\_iter” on line 111 of “fZooMSS\_Run.R”, then checking the calculation of the components of A\_iter, then the components of those components etc until we get back to the base inputs, to make sure the code is doing what it should be. I will run the code for 1 time step only, and I’ll only be focussing on group 2 (ciliates) and size-class -9 on the log10 scale (index of this size class in the weight vector is 18).

1. I downloaded ZooMSS from the github repo “ZoopModelSizeSpectra”.
2. I set everything up by running “setup\_RUN\_NAME.R” up to line 30. I’m using all the default settings from the code in the github repo, so I’ll be using cell 1 environmental data for this check, and will not be changing any parameters.
3. I enter “fZooMSS\_model.R” and manually run the code up to line 14, so I’ve set up param and model lists.
4. I enter “fZooMSS\_Run.R” and manually run the code up to line 111 (the calculation of A\_iter). Along the way, I manually set itime = 1 and skip line 65 (which initialises the for loop along time)
5. A\_iter[2,18] = 0.184373
6. A\_iter[2,18] = dt/dx \* gg[2,17] = 0.1\*1.84373 = 0.184373
7. gg[2,17] = ingested\_phyto[2,17] + cs[2,17] = 1.843728 + 2.319341e-0.9 = 1.84373.

**Now, I will check ingested\_phyto[2,17]. I’ll leave cs[2,17] at this stage, but it would follow almost the exact same steps.**

ingested\_phyto[2,17] = 1.843728  
\* I need to go back to “fZooMSS\_Setup.R” now to check the components of model$ingested\_phyto

Line 263 of “fZooMSS\_Setup.R” gives us

model$ingested\_phyto =<-model$temp\_eff\*(rowSums(sweep(model$phyto\_growthkernel, 3, model$nPP, "\*"), dims = 2))

and

(rowSums(sweep(model$phyto\_growthkernel, 3, model$nPP, "\*"), dims = 2))[2,17] = 13.65397

and also

(rowSums(sweep(model$phyto\_growthkernel, 3, model$nPP, "\*"), dims = 2))[2,17] = model$ingested\_phyto[2,17] / model$temp\_eff[2,17] = 1.843728/0.1350324 = 13.65397

Manually working out the rowsum and sweep answer for [2,17]:

sum(model$phyto\_growthkernel[2,17,]\*model$nPP) = 13.65397

Let’s check one size class of model$nPP, to make sure it is what it should be. I will fill in phyto\_int and phyto\_slope from the original inputs, and w\_phyto[5] from the w\_phyto vector:

model$nPP[5] = 1.25874e12 (I’ve rounded here)

model$nPP[5] <- 10^(param$phyto\_int)\*(param$w\_phyto[5]^(param$phyto\_slope)) = 10^(-2.924586)\*7.943282e-15^-1.065569 = 1.25874e12

So model$nPP looks okay. Let’s do something similar with model$phyto\_growthkernel, by focussing on model$phyto\_growthkernel[2,17,30]

model$phyto\_growthkernel[2,17,30] = 1.303599e-09

Remember, the growthkernel rows are predator size classes, the columns are prey size classes. Line 245 of fZooMSS.R gives us

model$phyto\_growthkernel <- sweep(sweep(model$phyto\_growthkernel, c(1,2), phyto\_theta, "\*"), 1, assim\_phyto, "\*")

phyto\_theta is a vector of 1s for group 2 (this matrix tells you whether or not a group and size class have access to phytoplankton, a relic of dvm but works fine), so the inner sweep does nothing to group 2 size classes. From line 76, assim\_phyto is

assim\_phyto <- (param$Groups$GrossGEscale) \* param$cc\_phyto = 2.5\*0.1 = 0.25 for all groups. So,

model$phyto\_growthkernel[2,17,30] = model$phyto\_growthkernel[2,17,30]\*0.25

= 5.2143959e-0.9\*0.25 = 1.303599e-09

Now, the model$phyto\_growthkernel on the RHS of the equation above is calculated earlier on line 215. From here on we’ll be referring to the line 215 phyto\_growthkernel.

From line 215, ignoring the for loop on line 152 and manually setting i=2

model$phyto\_growthkernel[2,17,30] = matrix(SearchVol[2,], nrow = param$ngrid, ncol = param$ngridPP)[17,30] \*

sp\_phyto\_predkernel[17,30] \* gg\_log\_t\_phyto[17,30] \* sm\_phyto[17,30] = 5.2143959e-0.9

Now to check each component, starting with matrix(SearchVol[2,], nrow = param$ngrid, ncol = param$ngridPP)[17,30]

matrix(SearchVol[2,], nrow = param$ngrid, ncol = param$ngridPP)[17,30] = SearchVol[2,17] = 3.3587677e-05 (param$Groups$SearchCoef[2])\*(param$w[17]^(param$Groups$SearchExp[2])) = 640\*param$w[17]^0.8 = 3.3587677e-05

Search volume is okay. What about sp\_phyto\_predkernel[17,30]? This is the feeding kernel given by the lognormal function:

Notionally, this function gives the probability a phytoplankton from size class w\_phyto[30] will be consumed by a predator from group 2 of size w[17]. From line 180-182:

sp\_phyto\_predkernel[17,30] = exp(-0.5\*(log((beta\_mat\_phyto[17,30]\*phyto\_prey\_weight\_matrix[17,30])/

phyto\_pred\_weight\_matrix[17,30])/param$Groups$FeedWidth[2])^2)/

sqrt(2\*pi\*param$Groups$FeedWidth[2]^2) = 0.84781478

Now, let’s see if we can get the same number with all the constituent parts. From line 176:

beta\_mat\_phyto[17,30] = (exp(0.02\*log(D.z[17])^2 - param$Groups$PPMRscale[2] + 1.832))^3

D.z[17] = 2\*(3\*param$w[17]\*1e12/(4\*pi))^(1/3) # convert body mass g to ESD (um)

= 11.490364

beta\_mat\_phyto[17,30] = (exp(0.02\*log(11.490364)^2 – 0.04 + 1.832))^3 = 309.09785, which agrees with the code’s output too

phyto\_prey\_weight\_matrix is a matrix of phytoplankton body sizes, where the rows are predator body sizes and the columns are prey body sizes, so phyto\_prey\_weight\_matrix[17,30] should equal param$w\_phyto[30], which it does (2.5118864e-12).

Similarly phyto\_pred\_weight\_matrix is a matrix of predator body sizes, where the rows are predator body sizes and the columns are prey body sizes, so phyto\_pred\_weight\_matrix[17,30] should equal param$w[17], which it does (7.9432823e-10).

Putting this all together:

sp\_phyto\_predkernel[17,30] = exp(-0.5\*(log((309.09785\*2.5118864e-12)/

7.9432823e-10)/0.47)^2)/

sqrt(2\*pi\*0.47^2) = 0.84781478

So, sp\_phyto\_predkernel is okay, what about gg\_log\_t\_phyto[17,30]?

gg\_log\_t\_phyto (line 99) is the bit of the growth equation that covers substituting weight in natural space w, with log10 weight space w = 10x. Rows are predator sizes, columns are prey sizes. Each element [i,j] of gg\_log\_t\_phyto should be (prey\_weight[j]/pred\_weight[i])/log(10)

gg\_log\_t\_phyto[17,30] = 0.0013733597

Doing this manually: (w\_phyto[30]/w[17])/ln(10) = 0.0013733597

So, gg\_log\_t\_phyto is okay. Last step is checking sm\_phyto, then we can put all the bits together again and see if we get line 215 model$phyto\_growthkernel.

sm\_phyto is the matrix of values to solve the integrals for growth, diffusion and mortality using simpson’s rule. Rows are predator sizes, columns are prey sizes. For each predator size class, we’re integrating along prey sizes to get the total amount of food each predator size class consumes. So, each row will be the same. The first element and final element of each row should be dx/3 = 0.03333… (which they are). All even elements excluding the final element (if applicable) should be dx/3\*4 = 0.13333… (which they are). All odd elements excluding the first and final element (if applicable) should be dx/3\*2 = 0.0666…. (which they are). So, sm\_phyto[17,30] is an even number for the column (prey) index, so it should be equal to 0.13333…, which it is.

Returning to model$phyto\_growthkernel[2,17,30] on line 215, which equalled 5.2143959e-0.9

model$phyto\_growthkernel[2,17,30] = matrix(SearchVol[2,], nrow = param$ngrid, ncol = param$ngridPP)[17,30] \*

sp\_phyto\_predkernel[17,30] \* gg\_log\_t\_phyto[17,30] \* sm\_phyto[17,30] = 3.3587677e-05\*0.84781478\*0.0013733597\*0.13333… = **5.2142654e-09**.

You should be able to do something very similar with the dynamic component of growth, which is only a tiny bit of the total growth rate of this group at this size class, so I’ve not done that here.