Patrick’s check process of ZooMizer on Monday 11 January 2021

I too will work backwards from `a` on line 194 of fZooMizer\_run.R.

1. I used ZooMizer from the github branch <https://github.com/patricksykes/ZooMizer/tree/zoomssupdatecompare>
2. In this branch, we’re using cell 1 of the full environmental data (as in Ryan’s test)
3. I enter comparison.RMD and run all chunks
4. This fails on line 413 (which is fine because I want to know about up to line 412)
5. A\_iter[2,18] = 0.184373 but a[2,18] = 1.302383
6. A\_iter[2,18] = param$dt/param$dx \* gg[2,17] = 0.1 \* 1.84373 = 0.184373  
   a[2,18] = e\_growth[2, 17] \* dt / params@dw[18] = 3.372202e-08 \* 0.01 / 2.589254e-10 = 1.302384
7. e\_growth[2,17] = encounter[2,17]-params@metab[2,17] = 3.372202e-08 – 0.
8. encounter[2,17] = params@other\_params$temp\_eff[2,17] \* params@search\_vol[2,17] \* (phi\_prey\_species[2,17] + phi\_prey\_background[2,17]) =0.1350324 \* 3.358768e-05 \* (1.57591e-08+0.007435235). **Search volume agrees with ZooMSS**
9. phi\_prey\_background is roughly the equivalent of ingested\_phyto in ZooMSS. So I’ll focus on that.
10. The exact equivalent of ingested\_phyto would be params@other\_params$temp\_eff[2,17] \* params@search\_vol[2,17] \* phi\_prey\_background[2,17] = 3.372195e-08

phi\_prey\_background[2,17] = params@other\_params$assim\_phyto[2] \* params@species\_params$interaction\_resource[2]\* rowSums(sweep(params@pred\_kernel, 3, params@dw\_full \* params@w\_full \* n\_pp, "\*", check.margin = FALSE), dims = 2)[2,17]  
= 0.25 \* 1 \* 0.02974094 = 0.007435235

params@pred\_kernel[2,17,30] = 0.8478148  
params@dw\_full[30] = 6.503912e-13  
params@w\_full[30] = 2.511886e-12  
n\_pp[30] = 4.196033e+21

* n\_pp[30] = kappa\*params@w\_full[30]^(1-lambda) / params@dw\_full[30] = 0.001189637 \* 2.511886e-12 ^ (1 - 2.065569) / 6.503912e-13 = 2729071679 / 6.503912e-13 = 4.196046e+21

in ZooMSS, nPP[30] is 2729062754 – small rounding error in the 6th s.f. (for the same comparison as Ryan…ZooMizer has n\_pp[5] \* params@dw\_full[5] = 1.258739e+12 **equal to ZooMSS** nPP[5] = 1.258739e+12)

**Now try some different things…**

**Mizer method in ZooMSS**

I inserted this code at the end of the fZooMSS\_Setup.R code before return(model):

model$dw\_phyto <- (10^param$dx - 1) \* param$w\_phyto

model$phi\_prey\_background <- assim\_phyto \* phyto\_theta[,1] \* rowSums(sweep(model$phyto\_growthkernel2, 3, #model$dw\_phyto \* # should remove this to match zoomizer since N/NPP in ZooMSS is absolute, not normalised

model$param$w\_phyto \* model$nPP, "\*", check.margin = FALSE), dims = 2)

model$phyto\_encounter <- model$temp\_eff \* model$phi\_prey\_background #\* SearchVol #SearchVol already included in phyto\_growthkernel2

where:

phyto\_growthkernel2 = array(NA, dim = c(param$ngrps, param$ngrid, param$ngridPP)),  
  
model$phyto\_growthkernel2[i,,] <- matrix(SearchVol[i,], nrow = param$ngrid, ncol = param$ngridPP) \* sp\_phyto\_predkernel

model$phyto\_encounter[2,17] = 3.372195e-08 **Which agrees with the value from ZooMizer. This confirms that the two models are calculating the integrals differently, with the same inputs.**

However, ZooMSS gives model$ingested\_phyto[2,17] = 1.843728

This has to do with the log/absolute weights and is sorted out in the next step:

For ZooMSS: A\_iter[,idx\_iter] <- dt/dx \* (ingested\_phyto + cs)[,idx\_iter-1]  
A\_iter[2,18] = 0.01/0.1 \* (1.843728 + 2.319341e-0.9) = 0.184373

For ZooMizer: a[, idx] <- sweep(-r$e\_growth[, idx - 1, drop = FALSE] \* dt, 2, params@dw[idx], "/")

a[2,18] = -r$e\_growth[2, 17] \* 0.01 / params@dw[18] = 3.372202e-08 \* 0.01 / 2.589254e-10 = 1.302384

**Remove Simpson’s rule**

We can force ZooMSS to use rectangles by replacing sm\_phyto and sm\_dynam with dx. We can do this manually every time those terms appear, or an easier way is replacing line 121 of fZooMSS\_Setup with

sm\_phyto <- matrix(1, nrow = param$ngrid, ncol = param$ngridPP, byrow = TRUE) \* (param$dx)

and line 126 by

sm\_dynam <- matrix(1, nrow = param$ngrid, ncol = param$ngrid, byrow = TRUE) \* (param$dx)

Then A\_iter[2,18] = 0.184373 **so Simpson’s rule is working the same as using Riemann sums**

**Look into the log10 transformation…**

If we write out in full all the parts of the a and A\_iter matrices (for the phyto consumption part, substitute for for the dynamic part), we get (for clarity I’ll use the usual names, not variable names). Note gg\_log\_t\_phyto = .

**ZooMSS:**

**ZooMizer:**

Noting that and the ’s are also the same, we should get

dw <- (10^param$dx - 1) \* param$w # This is a shortcut to w(i+1) – w(i) using the log-spacing of bins  
param$w \* log(10) / dw = 8.892851 **Not the factor of 7 I was expecting…have I missed something?**

In any case, shouldn’t a/A\_iter = 1?

Working it through a little further…

The actual quotient (with Simpson’s method taken out) is 7.063843. Open question is where this comes from/why it doesn’t agree with the theorised value.

Also, it’s the same factor of 7.063843 when looking only at growth due to phyto ingestion and growth due to zooplankton/fish ingestion in isolation.

**Mortality matrices**

As before, Simpson’s rule is turned off so both models are running with Riemann sums, just ZooMSS is doing it in logged-weight space.

Mortality includes both background mortality (senescence and fishing) and predation mortality. Fishing is set to 0 for both models and it’s easy to confirm that this is the case.

Taking a look at getMort(params)/Z, ~~most entries are 74.0563. This is exactly 10/temp\_eff!~~

Looking at senescence mortality: zoomizertest@params@mu\_b/zoomsstest$model$M\_sb, we get all 1s and NaNs (when both are 0). So, senescence mortality is the same for both models. (This is actually set by hand, so no surprise there)

This means the issue is with predation mortality. I found and fixed the missing temp\_eff term in the ZooMizer code, but now most entries are 10. Why?

Note that the predation rate is a vector in ZooMSS and an array (with identical rows for each species) in ZooMizer. Looking at getPredMort(zoomizertest@params)[1,]/M2, most entries are now 10. This now seems to be the main source of inconsistency.

The relevant code in ZooMizer:

new\_PredRate <- function(params, n, n\_pp, n\_other, t, feeding\_level, ...)  
{  
 n\_total\_in\_size\_bins <- sweep(n, 2, params@dw, "\*",  
 check.margin = FALSE)  
 pred\_rate <- sweep(params@pred\_kernel, c(1, 2), (1 - feeding\_level) \*  
params@other\_params$temp\_eff \* params@search\_vol \* n\_total\_in\_size\_bins,  
 "\*", check.margin = FALSE)  
 pred\_rate <- colSums(aperm(pred\_rate, c(2, 1, 3)), dims = 1)  
 return(pred\_rate)  
}

n\_total\_in\_size\_bins (ZooMizer) is the same as N in ZooMSS (as is expected).

The predation kernels are the same (this is done in the comparison.RMD file)

feeding\_level is an array of zeroes so (1-feeding\_level) = 1

params@other\_params$temp\_eff = zoomsstest$model$temp\_eff = 0.1350324 (as an array)

The search volumes are all the same too, except for the largest size class…which due to a rounding error has search volume 0 in Mizer. This is now fixed (by hand) and the result is that getPredMort(zoomizertest@params)[1,]/M2 = 10 for all entries.

From fZooMSS\_Run.R, M2 = dynam\_mortkernel \* N \* temp\_eff.

dynam\_mortkernel is calculated in fZooMSS\_Setup.R.

* model$dynam\_mortkernel=SearchVol \* sp\_dynam\_predkernel \*sm\_dynam

So, M2 = SearchVol \* sp\_dynam\_predkernel \*sm\_dynam \* N \* temp\_eff. This has a factor of sm\_dynam (which for us is just equal to dx=0.1). This entirely explains the discrepancy!

Which is right though?