DEBIPM - Sum of sensitivities

Viktor Thunell

2022-01-27

Sensitivity of lambda to kappa - comparison of total and decomposed sensitivity

A quick reminder: We do a sensitivity (perturbation) analysis across different temperatures to study which demographic processes, and the size at which they occur, affect how optimal energy allocation changes with warming. We analyze the partial contribution of each of o(T) (size at age 1), f(x,T) (fecundity) and g(y:x,T) (somatic growth) to sensitivity of λ to changes in energy allocation (κ_0) at three different temperatures to understand how size dependent vital rates affects optimum.

Here, I compare the size specific and total sum of the sensitivity contributions from demographic functions calculated in this analysis (point 3 below) with the sensitivity of lambda to a perturbation of κ_0 (point 1) and the sum of the size dependent sensitivity to a perturbation of the kernel matrix (point 2). This corresponds to a comparison of the first, third and last equality in

$$\begin{aligned}\frac{d\lambda}{d\kappa_0} &= \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\frac{d\lambda}{dK_{ij}}\frac{dK_{ij}}{d\kappa_0} \qquad (1)\\ &= \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} v_i w_j \frac{dK_{ij}}{d\kappa_0} \qquad (2)\\ &= \sum_{j=2}^{n+1} v_1 w_j a_j \frac{df_j}{d\kappa_0} + \sum_{i=2}^{n+1} v_i w_1 a_1 \frac{dg_{i1}}{d\kappa_0} + \sum_{i=2}^{n+1}\sum_{j=2}^{n+1} v_i w_j a_j \frac{dg_{ij}}{d\kappa_0}. \qquad (3)\end{aligned}$$

Create v and w for optimal κ_0 at three temperatures

```
#Optimum kappa at three temperatures (from produced results)
OptPars <- as.data.frame(cbind(T = c(287,289,291), kappa=c(0.99,0.95,0.92), Y=c(1,1,1)))

# Create results for v and w. This takes a few minutes
Res_v287 <- c(T=OptPars[1,1], kappa=OptPars[1,2], Y=1,
             wvlambda.projection(K.matrix(OptPars[1,]))$v)
Res_w287 <- c(T=OptPars[1,1], kappa=OptPars[1,2], Y=1,
               wvlambda.projection(K.matrix(OptPars[1,]))$w)

Res_v289 <- c(T=OptPars[2,1], kappa=OptPars[2,2], Y=1,
               wvlambda.projection(K.matrix(OptPars[2,]))$v)
Res_w289 <- c(T=OptPars[2,1], kappa=OptPars[2,2], Y=1,
               wvlambda.projection(K.matrix(OptPars[2,]))$w)

Res_v291 <- c(T=OptPars[3,1], kappa=OptPars[3,2], Y=1,
              wvlambda.projection(K.matrix(OptPars[3,]))$v)
Res_w291 <- c(T=OptPars[3,1], kappa=OptPars[3,2], Y=1,
              wvlambda.projection(K.matrix(OptPars[3,]))$w)

Res_v <- as.data.frame(rbind(Res_v287, Res_v289, Res_v291))
colnames(Res_v)[4:ncol(Res_v)] <- c(0,x)
Res_w <- as.data.frame(rbind(Res_w287, Res_w289, Res_w291))
colnames(Res_w)[4:ncol(Res_w)] <- c(0,x)
```

Functions for numerical differentiation (difference forward differentiation)

i.e. (f(x + h) - f(x)) / h)) of dK/dk_0, dGrowth/dk_0, dFecundityiy/dk_0 and dAge1/k_0.

```
# Kernel level pert.
K_NumDer <- function(h, Pars){
  (K.matrix(Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]] + h,Y = Pars[["Y"]])) -
     K.matrix(Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]],Y = Pars[["Y"]])))/h
}
# Growth function pert.
G_NumDer <- function(h ,m, Pars){
  (DEBgrowthfun(m, Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]] + h,Y = Pars[["Y"]])) -
     DEBgrowthfun(m, Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]],Y = Pars[["Y"]])))/h
}
# Fecundity function pert.
F_NumDer <- function(h ,m, Pars){
  (DEBrepfun(m, Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]] + h,Y = Pars[["Y"]])) -
   DEBrepfun(m, Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]],Y = Pars[["Y"]])))/h
}
# Age 1 size function pert.
A1_NumDer <- function(h ,m, Pars){
  (DEBage1.size(Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]] + h,Y = Pars[["Y"]])) -
     DEBage1.size(Pars = c(T = Pars[["T"]],kappa = Pars[["kappa"]],Y = Pars[["Y"]])))/h
}
```

1 - Calculate $d\lambda/d\kappa_0$ at 287 Kelvin

```
(wvlambda.projection(K.matrix(Pars = c(T = OptPars[1,1],kappa = OptPars[1,2]
                                       + 0.00001, Y = OptPars[1,3])))$lam
 -
 wvlambda.projection(K.matrix(Pars = c(T = OptPars[1,1],kappa = OptPars[1,2], Y = OptPars[1,3])))$lam) /
```

```
## [1] -0.04407772
```

The sensitivity of λ to a perturbation of κ_0 is -0.04407772.

2 - Calculate total size dependent sensitivity at 287 K by perturbing the K.matrix at optimal κ_0 at 287 K

```
res_v <- as.numeric(Res_v %>% filter(T == OptPars[1,1] & kappa==OptPars[1,2]))[4:ncol(Res_v)]
  res_w <- as.numeric(Res_w %>% filter(T == OptPars[1,1] & kappa==OptPars[1,2]))[4:ncol(Res_w)]

Sens_Kmat <- outer(res_v, res_w, "*") * K_NumDer(0.00001, OptPars[1,])

sum(apply(Sens_Kmat,1,sum))
```

```
## [1] -0.04404539
```

The total sum here (-0.04404539) is very close to the sensitivity of lambda to κ_0. I compare the size dependent sensitivity from these calculations to the summed demographic contributions in plot 1, point 3 below.

3 - Contributions to size dependent sensitivity from demographic functions

This is the analysis that go into the manuscript main text. I have done two major changes since the last draft: First I now use the transposed survival vector and numerical perturbation of the growth matrix when calculating Gr_cont below, i.e. how we calculate growth in the K.matrix (compare line 207 in Rscript_DEBIPM_model_20220123). Second, in the calculation for growth and age 1 size I multiply with dx).

```
Res_Sens <- NULL
for (i in 1:nrow(OptPars)){
  res_v <- as.numeric(Res_v %>% filter(T == OptPars[i,1] & kappa==OptPars[i,2]))[4:ncol(Res_v)]
  res_w <- as.numeric(Res_w %>% filter(T == OptPars[i,1] & kappa==OptPars[i,2]))[4:ncol(Res_w)]

  Gr_cont <- bind_cols(Size=x, Temp=OptPars[i,1], kappa=OptPars[i,2],
                       dfun="Gr", outer(res_v[2:(n+1)], res_w[2:(n+1)], "*")
                       * t(DEBsurvfun(x, OptPars[i,]) * t(G_NumDer(h=0.00001, x, OptPars[i,])))*dx)
  Gr_cont["Sens"] <- apply(Gr_cont[5:ncol(Gr_cont)],1,FUN=sum)# sum the distribution y to get sensitivi
  Gr_cont<-Gr_cont[,c(1:4,ncol(Gr_cont))]

  F_cont  <- bind_cols(Size=x, Temp=OptPars[i,1], kappa=OptPars[i,2],
                       dfun="F",  Sens=res_v[1]*res_w[2:(n+1)]
                       * DEBsurvfun(x, OptPars[i,]) * F_NumDer(h=0.00001, x, OptPars[i,]))

  A1_cont <- bind_cols(Size=x, Temp=OptPars[i,1], kappa=OptPars[i,2],
                       dfun="A1", Sens=res_v[2:(n+1)]*res_w[1]
                       * el_surv * A1_NumDer(h=0.00001, x, OptPars[i,])*dx)

  Res_Sens <- rbind(rbind(Gr_cont,F_cont,A1_cont), Res_Sens)
}

# sum the contributions from demographic functions at each size
Sum_cont <-
  Res_Sens %>%
  group_by(Size, Temp) %>%
  summarise(sum.t = sum(Sens, na.rm = TRUE))
```

```
## `summarise()` regrouping output by 'Size' (override with `.groups` argument)
```

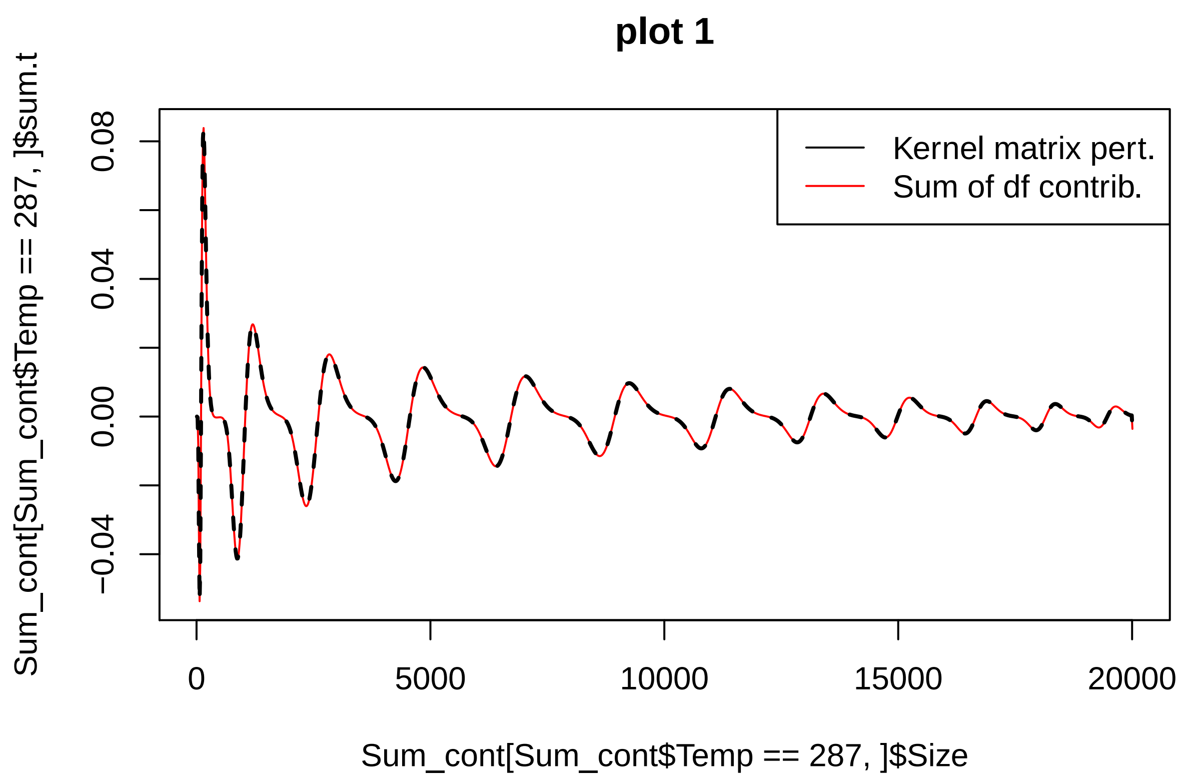
```r
sum(Sum_cont[Sum_cont$Temp==287,]$sum.t) # Same as in both in 1 and 2 ~ -0.04404539
```

```
## [1] -0.04404539
```

The sum of the size dependent sensitivity contributions from demographic functions equals the sum in point 2, i.e. -0.04404539 and thus almost the sum in point 1.

I compare by plotting, the sum of the size dependnet sensitivity contributions from demographic functions and the size dependnet sensitivity from point 2:

```r
# sum of contributions from demographic functions:
plot(Sum_cont[Sum_cont$Temp==287,]$Size,Sum_cont[Sum_cont$Temp==287,]$sum.t, type = "l", col="red", mai
# and the sensitivity from the kernel matrix (here I add the sensitivity contribution from fecundity (f
lines(x,Sens_Kmat[1,2:(n+1)]+apply(Sens_Kmat[2:(n+1),],1,sum), lty=2, lwd=2)
legend("topright", c("Kernel matrix pert.", "Sum of df contrib."), lty = c(1,1), col = c("black", "red"
```



So I get the same results in my comparisons of the demographic function contributions to sensitivity in both total (-0.04407772, -0.04404539, -0.04404539) and size dependent sensitivity (plot 1).

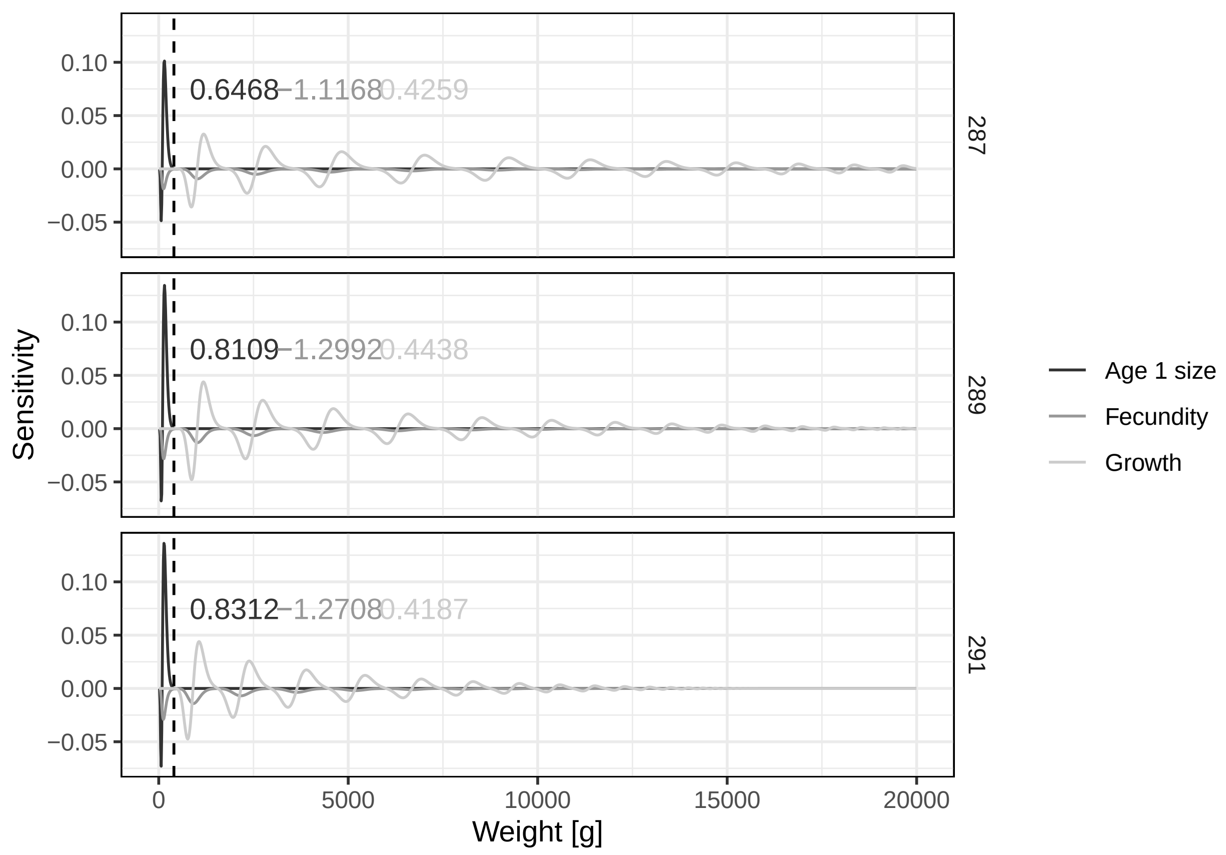
The deviation of the total sensitivity from zero could be due to that our optimum κ_0 is not exact. When I produce the model results, the values in the kappa range differ by 0.01, I guess that the optimum kappa used in the sensitivity analyses may not be precise enough to sum to zero.

4 - Resulting sensitivity contributions (figure 5 main text)

```r
library(tidyverse) # gglot etc.

# sum of size depdenent sensitivities
s_sum <- Res_Sens %>%
  group_by(dfun, Temp) %>%
  summarise(sum.c = sum(Sens, na.rm = TRUE)) #%>%
```

```
## `summarise()` regrouping output by 'dfun' (override with `.groups` argument)
```

```r
s_sum_text <- data.frame(Size = c(2000,2000,2000,4500,4500,4500,7000,7000,7000), Sens = 0.075,
                         dfun = s_sum$dfun, Temp = s_sum$Temp, label = round(s_sum$sum.c,4))

cont.labs <- c("Age 1 size", "Fecundity", "Growth")
names(cont.labs) <- c("A1", "F", "Gr")

Fig_sens <-
Res_Sens %>%
  mutate(dfun=as.factor(dfun)) %>%
  ggplot(.) +
  geom_path(aes(Size, Sens, color = dfun)) +
  facet_grid(Temp~., scales="fixed") +
  xlim(0,20000) +
  geom_vline(xintercept=400, linetype="dashed") +
  ylab("Sensitivity") +
  xlab("Weight [g]") +
  scale_color_grey(labels= cont.labs, name= "",) +
  geom_text(data=s_sum_text, aes(Size, Sens, label = label,
                              colour=dfun), size=4, show.legend = FALSE) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white", colour = "black"),
        strip.background = element_blank(),
        panel.border = element_rect(colour = "black", fill = NA))
Fig_sens
```


0.10
0.05
0.00
−0.05
0.6468
−1.1168
0.4259
287
0.8109
−1.2992
0.4438
289
0.8312
−1.2708
0.4187
291
Sensitivity
0
5000
10000
15000
20000
Weight [g]
Age 1 size
Fecundity
Growth