Using 2019 fecundity rates, waterhemp populations in all crop identity is decreasing in all rotation, fastest in the 3-year rotation (Figures ??).

### conventional   
  
# pick all phases in the 2yr conventional treatment and arrange in backward order, i.e., first subannual matrix in the first crop phase at the right-most (or bottom) position of the list  
all\_periods\_2yr\_conv <- scenario1\_projection\_by\_matrix\_id\_transpose[c("S2\_conv", "C2\_conv")]  
#View(all\_periods\_2yr\_conv)  
  
# remove the phase bounded structure in a chain, making a chain of 12 subannual matrices for the 2-year rotation  
all\_periods\_2yr\_conv\_chain <- unlist(all\_periods\_2yr\_conv, recursive = FALSE)  
  
# make a reference grid, based on a chain starting at the spring\_tillage matrix  
ref\_2yr\_conv <- names(all\_periods\_2yr\_conv\_chain)  
  
# create circular rotation of the first subannual matrix in each chain  
circ\_2yr\_conv <- circ(ref\_2yr\_conv)

## Warning in matrix(1:n, n + 1, n + 1, byrow = T): data length [12] is not a sub-  
## multiple or multiple of the number of rows [13]

full\_circ\_2yr\_conv <- purrr::map(1:12,~all\_periods\_2yr\_conv\_chain[circ\_2yr\_conv[.x,]])  
# name the nested list by the last element in each nested list  
# the last element in each nested list appear in the same order as circ\_2yr\_conv[,12]  
  
names(full\_circ\_2yr\_conv) <- paste0(circ\_2yr\_conv[,12],"\_contribution")   
  
# check if nested lists names matches their last elements' names: YES  
# View( full\_circ\_2yr\_conv) check  
  
# remove the last subannual matrix from the chain in full\_circ\_2yr\_conv  
until\_circ\_2yr\_conv <- full\_circ\_2yr\_conv %>%  
 map(., ~{head(.,11)})  
# until\_circ\_2yr\_conv contains 12 chains of 11 matrices each. In term of contribution to changes in lambda, each chain starts at the subannual matrix following the matrix of interest. The `until` and `though` designation is detailed in Caswell and Trevisan 1994.   
### low  
  
# pick all phases in the 2yr conventional treatment  
all\_periods\_2yr\_low <- scenario1\_projection\_by\_matrix\_id\_transpose[c("S2\_low", "C2\_low")]  
#View(all\_periods\_2yr\_conv)  
  
# rearrange the phases in a chain  
all\_periods\_2yr\_low\_chain <- unlist(all\_periods\_2yr\_low, recursive = FALSE)  
  
# make a reference grid   
ref\_2yr\_low <- names(all\_periods\_2yr\_low\_chain)  
  
circ\_2yr\_low <- circ(ref\_2yr\_low)

## Warning in matrix(1:n, n + 1, n + 1, byrow = T): data length [12] is not a sub-  
## multiple or multiple of the number of rows [13]

full\_circ\_2yr\_low <- purrr::map(1:12,~all\_periods\_2yr\_low\_chain[circ\_2yr\_low[.x,]])  
# name the nested list by the last element in each nested list  
# the last element in each nested list appear in the same order as circ\_2yr\_low[,12]  
  
names(full\_circ\_2yr\_low) <- paste0(circ\_2yr\_low[,12],"\_contribution")   
  
# check if nested lists names matches their last elements' names: YES  
# View( full\_circ\_2yr\_low): each chain's product is identified by the right-most matrix   
  
# remove the last subannual matrix from the chain in full\_circ\_2yr\_low  
until\_circ\_2yr\_low <- full\_circ\_2yr\_low %>%  
 map(., ~{head(.,11)})

In all LTRE procedures presented here, the conventional corn weed management treatment is the reference treatment and low herbicide the treatment of interest. Following @caswellSensitivityAnalysisPeriodic1994’s notions:

The sensitivity of to each element of each sub-annual periodic matrix is calculated with

where,  
 is the periodic projection matrix for sub-annual period h; h = {1,…,6},  
 is the transpose of the matrix product of all the , and  
 is the sensitivity of to each element of (the average annual projection matrix between the reference treatment and the treatment of interest)

The elasticities of to each element of a sub-annual projection matrix is calculated with

where,

is the entry at row i column j of matrix , and other elements as defined in Equation (1).

## In all LTRE procedures, the conventional treatment is the reference matrix and low the treatment of interest  
  
# left matrix multiplication of all chains (as defined by starting point, aka the right-most matrix of the chain) THROUGH the right-most matrix --> matrix A's  
  
# The product of each list in the full\_circ\_2yr\_xx is A(hk),  
# A(hk) is the product of all sub-annual period starting at phase k, period h  
A2\_conv\_contribution <- full\_circ\_2yr\_conv %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
A2\_low\_contribution <- full\_circ\_2yr\_low %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
A2\_contribution <- lst(A2\_conv\_contribution, A2\_low\_contribution)  
  
# calculate the A\* matrix, average of the two products representing conventional and low corn weed management treatments   
  
A2\_avg\_contribution <- lapply(Reduce(\(...)Map('+', ...), A2\_contribution), '/', length(A2\_contribution))  
  
# Matrix S\_A(h)'s: sensitivity of lambda to each element of A(h), i.e. a(ijh), using matrix A\*  
A2\_avg\_sens <- A2\_avg\_contribution %>%  
 map(., ~{sensitivity(.)})  
  
# Matrix D: the product of all sub-annual matrices in backward order, until the B(h) matrix appear.   
# other words: left matrix multiplication of all chains (as defined by starting point, B(h), UNTIL the B(h) appear, the chain that makes up D is one matrix fewer than one that makes S  
  
D2\_conv\_contribution <- until\_circ\_2yr\_conv %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
D2\_low\_contribution <- until\_circ\_2yr\_low %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
D2\_contribution <- lst(D2\_conv\_contribution, D2\_low\_contribution)  
  
### Do I really need D\_avg? ### Yes  
  
D2\_avg\_contribution <- lapply(Reduce(\(...)Map('+', ...), D2\_contribution), '/', length(D2\_contribution))  
  
D2\_avg\_contribution\_transposed <- D2\_avg\_contribution %>%  
 map(., ~{t(.)})  
  
# Matrix S\_B(h): sensitivity of lambda to changes in each element of the subannual matrix B(h)   
S\_B\_contribution\_2yr <- purrr::map2(D2\_avg\_contribution\_transposed, A2\_avg\_sens, `%\*%`)  
  
# remove "conv" in all names, conv was picked up from D2\_conv\_contribution because it was listed first in D2\_contribution.   
  
## pick out the names  
contribution\_2yr\_raw\_names <- names( S\_B\_contribution\_2yr)  
  
## remove "conv" in all names  
contribution\_2yr\_names <- gsub('conv.', '', contribution\_2yr\_raw\_names)  
  
names(S\_B\_contribution\_2yr) <- contribution\_2yr\_names

In the corn phase of the 2-year rotation, the mobility of seeds at the bottom stratum to seedling cohort 2 was the highest contributor to changes in the population growth (); to the seedling cohorts 1, 3 and 4 were immediate contributors to changes in the population growth. The same pattern was observed for the top-stratum seeds in the corn phase of the 2-year rotation, but their contribution to changes in were lower than those of the bottom-stratum seeds.

In the soybean phase of the 2-year rotation, emergence minimally contributed to changes in population growth.

*need a table, as graphs will be too many*

In 2019, as ’s were under 1 for all treatments, fecundity did not contribute substantially to changes.

# lambda atscenario1\_annualized\_lambda\_df\_long$annualized\_lambd  
lambda\_2yr\_conv <-scenario1\_annualized\_lambda\_df\_long$annualized\_lambda[1]  
  
lambda\_2yr\_low <-scenario1\_annualized\_lambda\_df\_long$annualized\_lambda[2]  
# Elasticity contribution left: check if all\_periods\_2yr\_conv\_chain is the appropriate element   
# The right-most matrix of all\_periods\_2yr\_conv\_chain is C2\_spring\_tillage  
# why is the order of "all\_periods\_2yr\_conv\_chain" and "sens\_contribution\_2yr" opposite?  
  
# Elasticity contribution left component: b\_hij\_2/lambda, or \_3, \_4 in the later chunks   
b\_hij\_2\_over\_lambda\_conv <- all\_periods\_2yr\_conv\_chain %>%  
 map(., ~{./lambda\_2yr\_conv})  
  
b\_hij\_2\_over\_lambda\_low <- all\_periods\_2yr\_low\_chain %>%  
 map(., ~{./lambda\_2yr\_low})  
  
# Elasticity contribution right component: S\_B's   
  
# Multiply the left and right components   
E\_B\_contribution\_2yr\_conv <- Map(\(x, y) Reduce(`%\*%`, list(x, y)), b\_hij\_2\_over\_lambda\_conv, S\_B\_contribution\_2yr)  
  
  
E\_B\_contribution\_2yr\_low <- Map(\(x, y) Reduce(`%\*%`, list(x, y)), b\_hij\_2\_over\_lambda\_low, S\_B\_contribution\_2yr)

### conventional   
  
# pick all phases in the 3yr conventional treatment  
all\_periods\_3yr\_conv <- scenario1\_projection\_by\_matrix\_id\_transpose[c("O3\_conv", "S3\_conv", "C3\_conv")]  
#View(all\_periods\_3yr\_conv)  
  
# rearrange the phases in a chain  
all\_periods\_3yr\_conv\_chain <- unlist(all\_periods\_3yr\_conv, recursive = FALSE)  
  
# make a reference grid   
ref\_3yr\_conv <- names(all\_periods\_3yr\_conv\_chain)  
  
circ\_3yr\_conv <- circ(ref\_3yr\_conv)

## Warning in matrix(1:n, n + 1, n + 1, byrow = T): data length [18] is not a sub-  
## multiple or multiple of the number of rows [19]

full\_circ\_3yr\_conv <- purrr::map(1:18,~all\_periods\_3yr\_conv\_chain[circ\_3yr\_conv[.x,]])  
# name the nested list by the last element in each nested list  
# the last element in each nested list appear in the same order as circ\_3yr\_conv[,12]  
  
names(full\_circ\_3yr\_conv) <- paste0(circ\_3yr\_conv[,18],"\_contribution")   
  
# check if nested lists names matches their last elements' names: YES  
# View( full\_circ\_2yr\_conv) check  
  
# remove the last subannual matrix from the chain in full\_circ\_3yr\_conv  
until\_circ\_3yr\_conv <- full\_circ\_3yr\_conv %>%  
 map(., ~{head(.,11)})  
  
### low  
  
# pick all phases in the 3yr low treatment  
all\_periods\_3yr\_low <- scenario1\_projection\_by\_matrix\_id\_transpose[c("O3\_low", "S3\_low", "C3\_low")]  
#View(all\_periods\_2yr\_conv)  
  
# rearrange the phases in a chain  
all\_periods\_3yr\_low\_chain <- unlist(all\_periods\_3yr\_low, recursive = FALSE)  
  
# make a reference grid   
ref\_3yr\_low <- names(all\_periods\_3yr\_low\_chain)  
  
circ\_3yr\_low <- circ(ref\_3yr\_low)

## Warning in matrix(1:n, n + 1, n + 1, byrow = T): data length [18] is not a sub-  
## multiple or multiple of the number of rows [19]

full\_circ\_3yr\_low <- purrr::map(1:18,~all\_periods\_3yr\_low\_chain[circ\_3yr\_low[.x,]])  
# name the nested list by the last element in each nested list  
# the last element in each nested list appear in the same order as circ\_3yr\_low[,12]  
  
names(full\_circ\_3yr\_low) <- paste0(circ\_3yr\_low[,18],"\_contribution")   
  
# check if nested lists names matches their last elements' names: YES  
# View( full\_circ\_3yr\_low)   
  
# remove the last subannual matrix from the chain in full\_circ\_3yr\_low  
until\_circ\_3yr\_low <- full\_circ\_3yr\_low %>%  
 map(., ~{head(.,11)})

## In all LTRE procedures, the conventional treatment is the reference matrix and low the  
  
# left matrix multiplication of all chains (as defined by starting point, aka the right-most matrix of the chain) THROUGH the right-most matrix --> matrix A's  
A3\_conv\_contribution <- full\_circ\_3yr\_conv %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
A3\_low\_contribution <- full\_circ\_3yr\_low %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
A3\_contribution <- lst(A3\_conv\_contribution, A3\_low\_contribution)  
  
A3\_avg\_contribution <- lapply(Reduce(\(...)Map('+', ...), A3\_contribution), '/', length(A3\_contribution))  
  
# Sensitivities with regards to the right-most matrix --> matrix S\_a's  
A3\_avg\_sens <- A3\_avg\_contribution %>%  
 map(., ~{sensitivity(.)})  
  
# left matrix multiplication of all chains (as defined by starting point, aka the right-most matrix of the chain) UNTIL the right-most matrix appear --> matrix D's  
  
D3\_conv\_contribution <- until\_circ\_3yr\_conv %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
D3\_low\_contribution <- until\_circ\_3yr\_low %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
D3\_contribution <- lst(D3\_conv\_contribution, D3\_low\_contribution)  
  
D3\_avg\_contribution <- lapply(Reduce(\(...)Map('+', ...), D3\_contribution), '/', length(D3\_contribution))  
  
S\_B\_contribution\_3yr <- purrr::map2(D3\_avg\_contribution, A3\_avg\_sens, `%\*%`)  
  
# remove "conv" in all names  
contribution\_3yr\_raw\_names <- names(S\_B\_contribution\_3yr )  
  
contribution\_3yr\_names <- gsub('conv.', '', contribution\_3yr\_raw\_names)  
  
names(S\_B\_contribution\_3yr) <- contribution\_3yr\_names

# lambda at scenario1\_rotation\_wise\_lambda   
lambda\_3yr\_conv <- scenario1\_annualized\_lambda\_df\_long$annualized\_lambda[3]  
  
lambda\_3yr\_low <- scenario1\_annualized\_lambda\_df\_long$annualized\_lambda[4]  
  
  
# Elasticity contribution left component: b\_hij\_3/lambda,  
b\_hij\_3\_over\_lambda\_conv <- all\_periods\_3yr\_conv\_chain %>%  
 map(., ~{./lambda\_3yr\_conv})  
  
b\_hij\_3\_over\_lambda\_low <- all\_periods\_3yr\_low\_chain %>%  
 map(., ~{./lambda\_3yr\_low})  
  
# Elasticity contribution right component: S\_B's   
  
# Multiply the left and right components   
E\_B\_contribution\_3yr\_conv <- Map(\(x, y) Reduce(`%\*%`, list(x, y)), b\_hij\_3\_over\_lambda\_conv, S\_B\_contribution\_3yr)  
  
  
E\_B\_contribution\_3yr\_low <- Map(\(x, y) Reduce(`%\*%`, list(x, y)), b\_hij\_3\_over\_lambda\_low, S\_B\_contribution\_3yr)

### conventional   
  
# pick all phases in the 4yr conventional treatment  
all\_periods\_4yr\_conv <- scenario1\_projection\_by\_matrix\_id\_transpose[c("A4\_conv", "O4\_conv", "S4\_conv", "C4\_conv")]  
# View(all\_periods\_4yr\_conv)  
  
# rearrange the phases in a chain  
all\_periods\_4yr\_conv\_chain <- unlist(all\_periods\_4yr\_conv, recursive = FALSE)  
  
# make a reference grid   
ref\_4yr\_conv <- names(all\_periods\_4yr\_conv\_chain)  
  
circ\_4yr\_conv <- circ(ref\_4yr\_conv)

## Warning in matrix(1:n, n + 1, n + 1, byrow = T): data length [24] is not a sub-  
## multiple or multiple of the number of rows [25]

full\_circ\_4yr\_conv <- purrr::map(1:24,~all\_periods\_4yr\_conv\_chain[circ\_4yr\_conv[.x,]])  
# name the nested list by the last element in each nested list  
# the last element in each nested list appear in the same order as circ\_4yr\_conv[,12]  
  
names(full\_circ\_4yr\_conv) <- paste0(circ\_4yr\_conv[,24],"\_contribution")   
  
# check if nested lists names matches their last elements' names: YES  
# View( full\_circ\_4yr\_conv) check  
  
# remove the last subannual matrix from the chain in full\_circ\_3yr\_conv  
until\_circ\_4yr\_conv <- full\_circ\_4yr\_conv %>%  
 map(., ~{head(.,11)})  
  
### low  
  
# pick all phases in the 3yr low treatment  
all\_periods\_4yr\_low <- scenario1\_projection\_by\_matrix\_id\_transpose[c("A4\_low", "O4\_low", "S4\_low", "C4\_low")]  
#View(all\_periods\_4yr\_conv)  
  
# rearrange the phases in a chain  
all\_periods\_4yr\_low\_chain <- unlist(all\_periods\_4yr\_low, recursive = FALSE)  
  
# make a reference grid   
ref\_4yr\_low <- names(all\_periods\_4yr\_low\_chain)  
  
circ\_4yr\_low <- circ(ref\_4yr\_low)

## Warning in matrix(1:n, n + 1, n + 1, byrow = T): data length [24] is not a sub-  
## multiple or multiple of the number of rows [25]

full\_circ\_4yr\_low <- purrr::map(1:24,~all\_periods\_4yr\_low\_chain[circ\_4yr\_low[.x,]])  
# name the nested list by the last element in each nested list  
# the last element in each nested list appear in the same order as circ\_4yr\_low[,12]  
  
names(full\_circ\_4yr\_low) <- paste0(circ\_4yr\_low[,24],"\_contribution")   
  
# check if nested lists names matches their last elements' names: YES  
# View( full\_circ\_4yr\_low)   
  
# remove the last subannual matrix from the chain in full\_circ\_4yr\_low  
until\_circ\_4yr\_low <- full\_circ\_4yr\_low %>%  
 map(., ~{head(.,11)})

## In all LTRE procedures, the conventional treatment is the reference matrix and low the  
  
# left matrix multiplication of all chains (as defined by starting point, aka the right-most matrix of the chain) THROUGH the right-most matrix --> matrix A's  
A4\_conv\_contribution <- full\_circ\_4yr\_conv %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
A4\_low\_contribution <- full\_circ\_4yr\_low %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
A4\_contribution <- lst(A4\_conv\_contribution, A4\_low\_contribution)  
  
A4\_avg\_contribution <- lapply(Reduce(\(...)Map('+', ...), A4\_contribution), '/', length(A4\_contribution))  
  
# Sensitivities with regards to the right-most matrix --> matrix S\_A's  
A4\_avg\_sens <- A4\_avg\_contribution %>%  
 map(., ~{sensitivity(.)})  
  
# left matrix multiplication of all chains (as defined by starting point, aka the right-most matrix of the chain) UNTIL the right-most matrix appear --> matrix D's  
  
D4\_conv\_contribution <- until\_circ\_4yr\_conv %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
D4\_low\_contribution <- until\_circ\_4yr\_low %>%   
 map(., ~{Reduce( "%\*%", .)})   
  
D4\_contribution <- lst(D4\_conv\_contribution, D4\_low\_contribution)  
  
D4\_avg\_contribution <- lapply(Reduce(\(...)Map('+', ...), D4\_contribution), '/', length(D4\_contribution))  
  
D4\_avg\_contribution\_transposed <- D4\_avg\_contribution %>% map(.,~{t(.)})  
  
## contribution of each sub-annual matrix to lambda's sensitivities, matrix S\_B's   
S\_B\_contribution\_4yr <- purrr::map2(D4\_avg\_contribution\_transposed, A4\_avg\_sens, `%\*%`)  
# remove "conv" in all names  
S\_B\_contribution\_4yr\_raw\_names <- names( S\_B\_contribution\_4yr )  
  
S\_B\_contribution\_4yr\_names <- gsub('conv.', '', S\_B\_contribution\_4yr\_raw\_names)  
  
names(S\_B\_contribution\_4yr) <- S\_B\_contribution\_4yr\_names

## calculate elasticity, see p. 121  
  
## NEED NEW NAMES FOR ALL THE LIST  
# lambda at scenario1\_rotation\_wise\_lambda   
lambda\_4yr\_conv <-scenario1\_annualized\_lambda\_df\_long$annualized\_lambda[5]  
  
lambda\_4yr\_low <-scenario1\_annualized\_lambda\_df\_long$annualized\_lambda[6]  
# Elasticity contribution left: check if all\_periods\_4yr\_conv\_chain is the appropriate element   
# The right-most matrix of all\_periods\_4yr\_conv\_chain is C4\_spring\_tillage  
# why is the order of "all\_periods\_4yr\_conv\_chain" and "sens\_contribution\_4yr" opposite?  
  
  
A4\_conv\_elas\_left <- all\_periods\_4yr\_conv\_chain %>%  
 map(., ~{./lambda\_4yr\_conv})  
  
A4\_low\_elas\_left <- all\_periods\_4yr\_low\_chain %>%  
 map(., ~{./lambda\_4yr\_low})  
  
# Elasticity contribution right component: S\_B's sens\_contribution\_4yr  
  
# Elasticity  
elas\_contribution\_4yr\_conv <- Map(\(x, y) Reduce(`%\*%`, list(x, y)), A4\_conv\_elas\_left, S\_B\_contribution\_4yr)

Variance of annualized population growth rate is calculated with Taylor series expansion using the general formula: , in which is the nth-root function used to annualize rotation-wise growth rate. For example, the variance of annualized population growth rate in the 2-year rotation was , where .  
<https://en.wikipedia.org/wiki/Taylor_expansions_for_the_moments_of_functions_of_random_variables> .

The variance of population growth rate is calculated with where is the variance-covariance matrix of each periodic matrix.

# combine all variance into a data frame and calculate annualized lambda's var  
  
scenario1\_annualized\_lambda\_df\_long$Rotation\_var\_lambda <- c(var\_lambda\_conv\_2yr,  
 var\_lambda\_low\_2yr,  
 var\_lambda\_conv\_3yr,  
 var\_lambda\_low\_3yr,  
 var\_lambda\_conv\_4yr,  
 var\_lambda\_low\_4yr)  
  
# add multiplier to the var table  
scenario1\_annualized\_lambda\_df\_long$multiplier <- c(1/4, 1/4, 1/9, 1/9, 1/16, 1/16)  
  
  
# combine lambda, var(lamda), and \*elasticity contribution\* and rename the data frame  
scenario1\_annualized\_lambda\_and\_var <- scenario1\_annualized\_lambda\_df\_long %>%  
 mutate(squared\_annualized\_lambda = annualized\_lambda^2,  
 var\_annualized\_lambda = multiplier \* Rotation\_var\_lambda/squared\_annualized\_lambda) %>%  
 select(.id, Corn\_weed\_management, annualized\_lambda, var\_annualized\_lambda)  
  
View( scenario1\_annualized\_lambda\_and\_var)