

Calculating Transient Dynamics

Load the popdemo package

```
library(popdemo)
```

Data

We will use a matrix for the desert tortoise *Gopherus agassizii*, with medium fecundity, which was published in Doak et al. (1994) Ecol. Appl., 4, 446-460. There are 8 stages are based on age and size (carapace length in mm):

Yearling (age 0-1)

Juvenile 1 (<60 mm)

Juvenile 2 (90-99mm)

Immature 1 (100-139mm)

Immature 2 (140-179mm)

Subadult (180-207mm)

Adult 1 (208-239mm)

Adult 2 (>240mm).

Load in the data

```
data(Tort)
```

Exercise 1: Transient dynamics of specific population structures

We can calculate amplified and/or attenuated transient indices for specified population vectors. All indices standardise the matrix (removing effects of asymptotic dynamics) and the population vector (removing the effects of initial population size). This means transient indices are comparable both within and between models, no matter their dimension, rate of asymptotic growth, or overall population size.

A population that amplifies:

Create the population structure vector

```
Tortamp<-Matlab2R("[1;1;2;3;5;8;13;21]")
```

Calculate REACTIVITY

```
( amp1<-reac(Tort, vector=Tortamp) )
```

```
## [1] 2.631922
```

This is the population size in the first timestep of the projection (one year), relative to a population with stable growth. The nonstable population grows 2.6 times faster than its stable growth rate in the first timestep.

Calculate MAXIMUM AMPLIFICATION

```
( amp2<-maxamp(Tort, vector=Tortamp) )
```

```
## [1] 5.110659
```

This is the largest population size that is achieved, relative to a population with stable growth. The nonstable population is at most 5.1 times as big as a population that grows with a stable growth rate.

Calculate POPULATION INERTIA

```
( amp3<-inertia(Tort, vector=Tortamp) )
```

```
## [1] 4.144233
```

This is the ratio of the size of population in the long-term, relative to a population with stable growth. After the transient period, the population settles to a size 4.14 times as large as a population that grows with stable rate.

Indices of amplification are always >1 , which means that log-transformed indices of amplification are always >0 .

A population that attenuates:

Create the population structure vector

```
Tortatt<-Matlab2R("[21;13;8;5;3;2;1;1]")
```

Calculate FIRST-TIMESTEP ATTENUATION

```
( att1<-reac(Tort, vector=Tortatt) )
```

```
## [1] 0.9152956
```

This is the complement to reactivity, describing relative population size in the first timestep. In this time, the population declines to a size 0.92 times smaller than a population growing at a stable rate.

Calculate MAXIMUM ATTENUATION

```
( att2<-maxatt(Tort, vector=Tortatt) )
```

```
## [1] 0.8376878
```

This is the smallest population size that is achieved, relative to a population with stable growth. The population is never smaller than 0.84 times the size of a population growing at a stable rate.

Calculate POPULATION INERTIA

```
( att3<-inertia(Tort, vector=Tortatt) )
```

```
## [1] 0.9122843
```

After the transient period, the population settles to a size 0.91 times smaller than a population that grows with stable rate.

Indices of attenuation are always <1 but >0 . This means that log-transformed indices of attenuation are always <0 .

A specific population vector will always EITHER amplify OR attenuate in the first-timestep, and EITHER amplify OR attenuate asymptotically. But, it's possible that across the whole population projection there will be both amplification and attenuation, so a projection may have both a maximum amplification and a maximum attenuation.

There are other options available when calculating transient dynamics. The functions calculate indices standardised by long-term growth and initial population size, but it might be important to know the actual population size including these effects.

Calculate first-timestep population sizes for the two population structures

```
( amp1.2<-reac(Tort, vector=Tortamp, return.N=TRUE) )
```

```
## $reac
## [1] 2.631922
##
## $N
## [1] 136.163
```

```
( att1.2<-reac(Tort, vector=Tortatt, return.N=TRUE) )
```

```
## $reac
## [1] 0.9152956
##
## $N
## [1] 47.353
```

Calculate inertia for the two population structures and the population size at $t=50$

```
( amp3.2<-inertia(Tort, vector=Tortamp, return.N=TRUE, t=50) )
```

```
## $inertia
## [1] 4.144233
##
## $N
## [1] 26.26972
```

```
( att3.2<-inertia(Tort, vector=Tortatt, return.N=TRUE, t=50) )
```

```
## $inertia
## [1] 0.9122843
##
## $N
## [1] 5.782845
```

For maximum amplification and attenuation, it may also be important to know the point in time that the maximum amplification or attenuation is reached. Remember that this is measured relative to long-term growth, so it is not necessarily the point in time of largest or smallest population size (in fact the largest and smallest population sizes just depend on the length of the projection for increasing and declining populations respectively)

Calculate time of maximum amplification and attenuation for the two population structures

```
( att2.2<-maxatt(Tort, vector=Tortatt, return.N=TRUE, return.t=TRUE) )
```

```
## $maxatt
## [1] 0.8376878
##
## $N
## [1] 38.11056
##
## $t
## [1] 4
```

```
( att2.2<-maxatt(Tort, vector=Tortatt, return.N=TRUE, return.t=TRUE) )
```

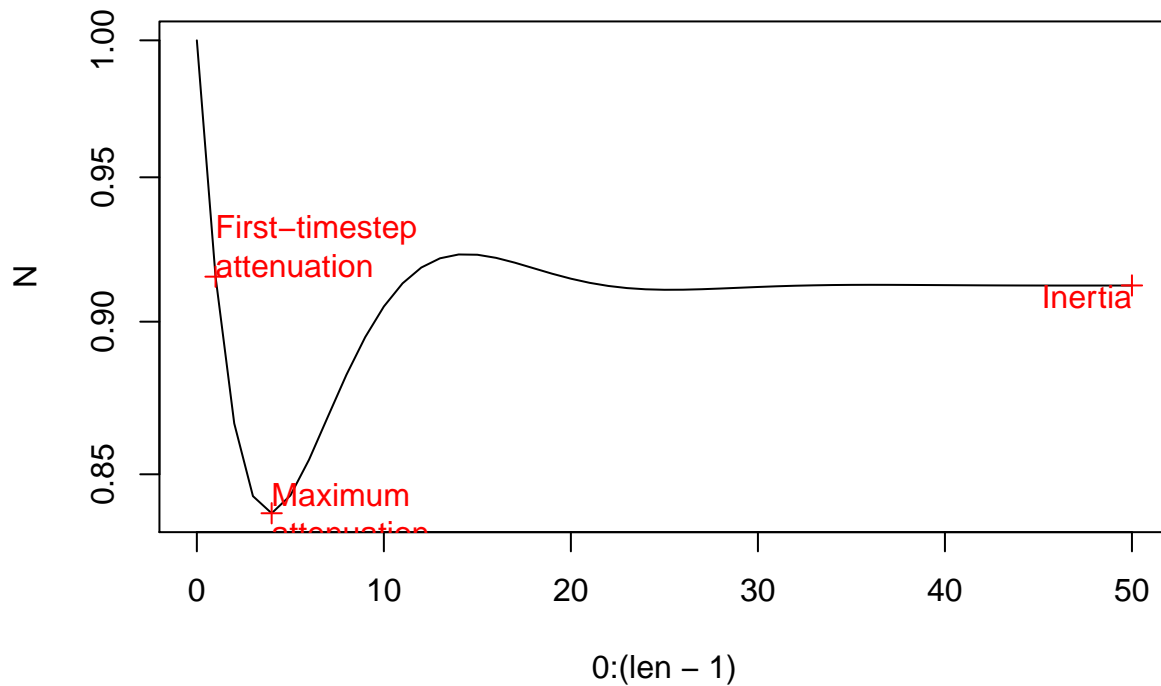
```
## $maxatt
## [1] 0.8376878
##
## $N
## [1] 38.11056
##
## $t
## [1] 4
```

popdemo has some useful tools for visualising projection of population dynamics. We can visualise both standardised and non-standardised transient dynamics on a population projection plot.

STANDARDISED DYNAMICS:

```
stdpr<-project(Tort, vector=Tortatt, time=50, standard.A=TRUE, standard.vec=TRUE)
plot(stdpr, log="y", main="Standardised transient dynamics")
points(1, att1, col="red", pch=3)
text(1, att1, "First-timestep\nattenuation", col="red", adj=c(0,0))
points(att2.2$t, att2.2$maxatt, col="red", pch=3)
text(att2.2$t, att2.2$maxatt, "Maximum\nattenuation", col="red", adj=c(0,0.5))
points(50, att3, col="red", pch=3)
text(50, att3, "Inertia", col="red", adj=c(1,1))
```

Standardised transient dynamics

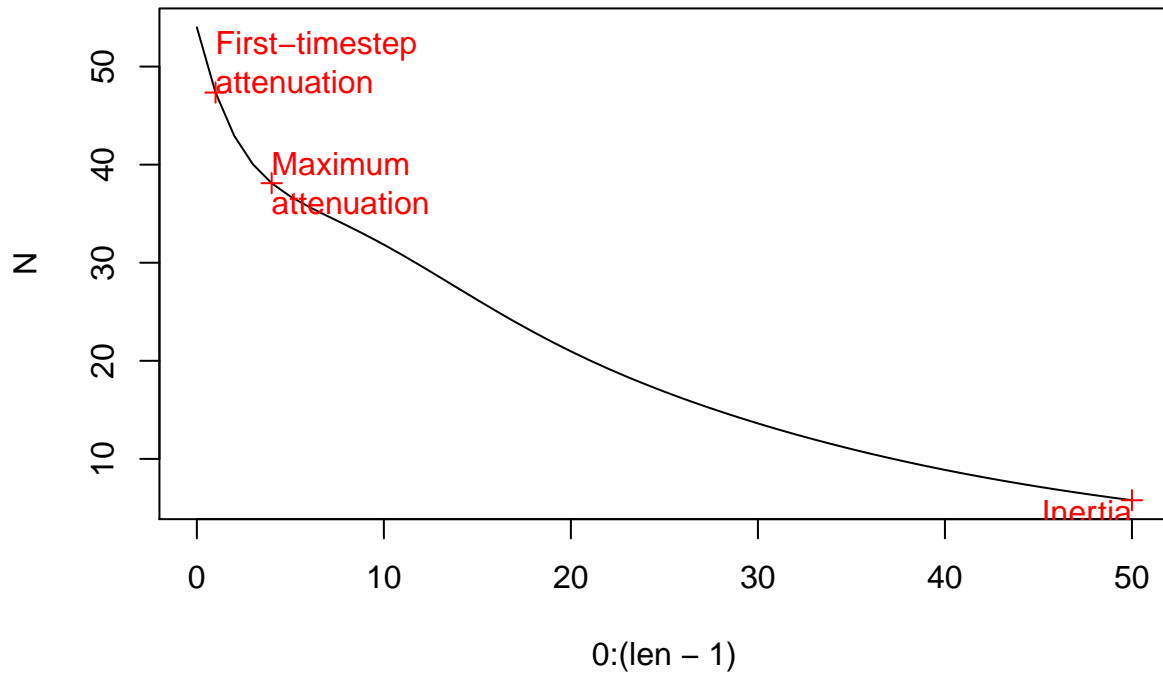


Note that the y-axis is on a log scale. Attenuation dynamics are bounded between 0 and 1 whilst amplification dynamics are bounded between 1 and infinity. An amplification of 2 means the population is twice as big as if it was stable, whilst an attenuation of 0.5 means the population is half as big as if it was stable. When visualising, it's good to give these two processes equal weight by visualising on a log-scale. When doing comparative analyses of transient dynamics between populations and/or species, it's very important to log-transform indices so that attenuation and amplification are given equal “weight” in the analysis. Log-transformed attenuation is bounded between negative infinity and 0, and log-transformed amplification is bounded between 0 and positive infinity.

NON-STANDARDISED DYNAMICS:

```
nstdpr<-project(Tort, vector=Tortatt, time=50)
plot(nstdpr, main="Non-standardised population dynamics")
points(1, att1.2$N, col="red", pch=3)
text(1, att1.2$N, "First-timestep\nattenuation", col="red", adj=c(0,0))
points(att2.2$t, att2.2$N, col="red", pch=3)
text(att2.2$t, att2.2$N, "Maximum\nattenuation", col="red", adj=c(0,0.5))
points(50, att3.2$N, col="red", pch=3)
text(50, att3.2$N, "Inertia", col="red", adj=c(1,1))
```

Non-standardised population dynamics



This shows the projected population size over time. Notice how the maximum attenuation is not the smallest population size, as it is measured relative to the long-term decline in the population.

Exercise 2: Bounds on transient dynamics

Bounds on transient dynamics represent the absolute largest and the absolute smallest population sizes relative to stable growth that transient dynamics can achieve. These result from 'stage-biased' population projections, which are population structures which have all individuals in one stage class. In maths these are called 'standard basis' vectors: column vectors consisting of zeroes except for a 1 in one of the elements. For example, a set of stage-biased vectors for a 3 by 3 matrix is $c(1,0,0)$; $c(0,1,0)$; $c(0,0,1)$.

Transient bounds can be useful to use to inform on the potential range of transient dynamics that a population could exhibit: they are best- and worst-case scenarios depending on what the population management goals are. They can still be used when the structure of the population is unknown.

Every matrix has an upper and a lower bound on reactivity/first-timestep attenuation, an upper bound on maximum amplification, a lower bound on maximum attenuation and an upper and a lower bound on population inertia. To calculate bounds on transient dynamics, just don't pass a population structure to the functions.

Calculate the bounds on reactivity and first-timestep attenuation:

```
( reac<-reac(Tort,bound="upper") )
```

```
## [1] 3.580154
```

```
( firststep<-reac(Tort,bound="lower") )
```

```
## [1] 0.7473442
```

In the first timestep after any disturbance to population structure, the population cannot be more than 3.5 times, or less than 0.75 times its size if it were growing at stable rate.

Calculate the bounds on maximum amplification and maximum attenuation:

```
( mxamp<-maxamp(Tort, return.t=TRUE) )
```

```
## $maxamp  
## [1] 6.829891  
##  
## $t  
## [1] 5
```

```
( mxatt<-maxatt(Tort, return.t=TRUE) )
```

```
## $maxatt  
## [1] 0.12284  
##  
## $t  
## [1] 9
```

If the environment stays constant after a disturbance, then the population should not ever become more than 6.8 or less than 0.12 times its size if it were growing at stable rate.

Calculate the upper and lower bounds on inertia:

```
( upinertia<-inertia(Tort, bound="upper") )
```

```
## [1] 5.123709
```

```
( lowinertia<-inertia(Tort, bound="lower") )
```

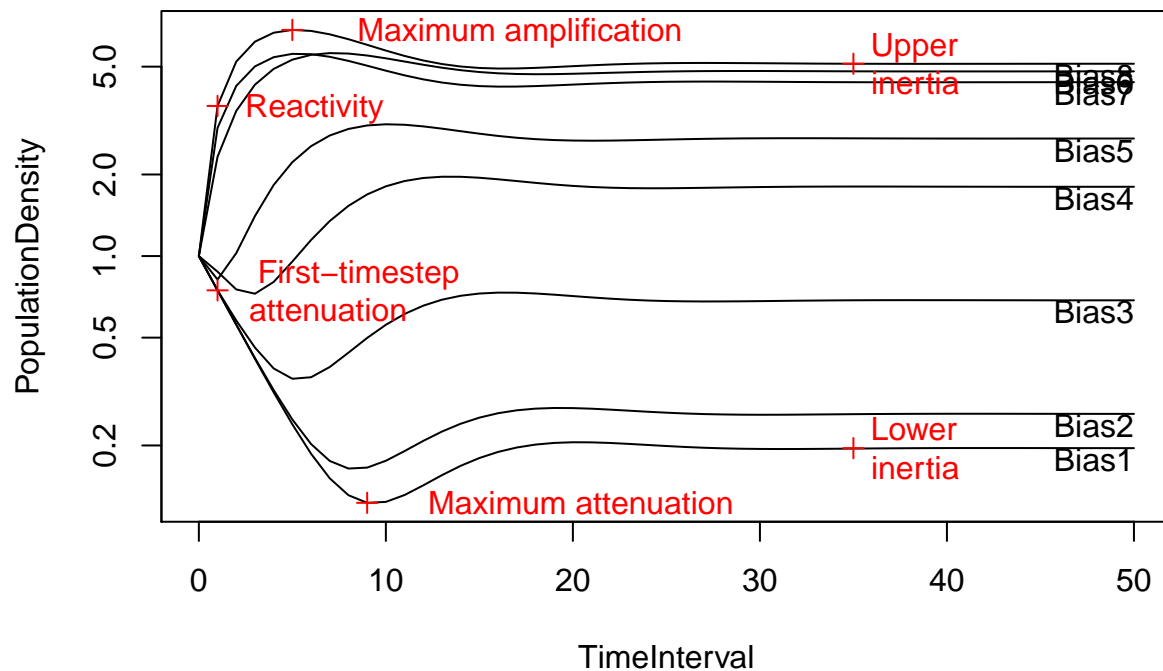
```
## [1] 0.1954968
```

If the environment stays constant following a disturbance, then the population will settle to a size between 0.2 and 5.1 times its size if it were growing at stable rate.

The same argument is available to return the realised (non-standardised) population size using .

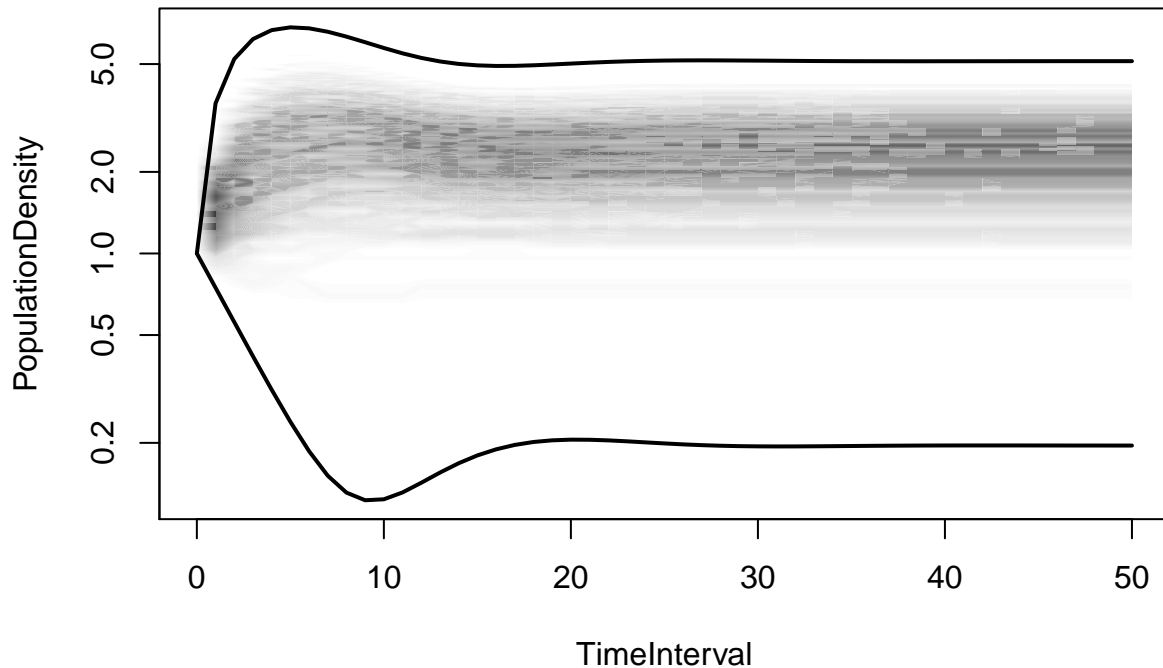
Let's Visualise these on a population projection. Again, if we don't specify the vector then the bounds will be automatically calculated.

```
SBpr<-project(Tort, time=50, standard.A=TRUE)
plot(SBpr, log="y")
points(c(1,1), c(reac,firststep), col="red", pch=3)
text(c(1,1), c(reac,firststep), col="red", c("Reactivity","First-timestep\nattenuation"),
     adj=c(-0.2,0.5))
points(c(mxamp$t,mxatt$t), c(mxamp$maxamp,mxatt$maxatt), pch=3, col="red")
text(c(mxamp$t,mxatt$t), c(mxamp$maxamp,mxatt$maxatt),
     c("Maximum amplification","Maximum attenuation"), col="red", adj=c(-0.2,0.5))
points(c(35,35), c(upinertia,lowinertia), col="red", pch=3)
text(c(35,35), c(upinertia,lowinertia), col="red", c("Upper\ninertia","Lower\ninertia"),
     adj=c(-0.2,0.5))
```



All the possible population densities lie within the bounds on transient dynamics. We can visualise this by drawing random population structures from a dirichlet distribution, projecting them, and then plotting a heatmap of the probability densities of population size over time, within the bounds. We do this by specifying in the project function. We will draw from a uniform dirichlet distribution by specifying .

```
diripr<-project(Tort, time=50, standard.A=TRUE, vector="diri", draws=1000, alpha.draws="unif")
plot(diripr, plottype="contour", log="y")
```

Exercise 3: COM(P)ADRE tasks

You can download the COMPADRE and COMADRE databases [here](#). Save the data somewhere on your computer and load it up using `load(your/file/path/here/compadre.Rdata)`. Then, pick a matrix to work with. For example:

```
M<-comadre$mat[comadre$metadata$SpeciesAccepted=="Ursus_maritimus"][[1]]$matA
```

This chooses us a matrix for the polar bear *Ursus maritimus*.

Now try and calculate the transient dynamics of your chosen matrix.

1. Plot the stage-biased transient dynamics of the matrix.
2. Calculate the bounds on reactivity, maximal amplification, maximal attenuation, and inertia.
3. Create a specific population structure for that population.
4. Calculate the specific reactivity, maximal amplification (and/or attenuation), and inertia for your chosen population.
5. Calculate the time of maximal amplification and/or attenuation.
6. Which stage bias achieves the bound on maximal amplification and/or attenuation?
7. What is the population size, including the effects of initial population size and long-term dynamics, at the first timestep of projection?
8. Create a second, different population structure

9. How are the transient dynamics of this population structure different? Does one amplify or attenuate more than the other? Compare the two population structures: why are the transients different? How does the population structure cause the transients to be different? Is the immediate reproductive value of one population more than the other?
 10. Did you encounter any errors, warnings or peculiarities when calculating the transients? Why? How can you solve these problems?
 11. Plot a heatmap of probable population densities for this matrix under a uniform dirichlet draw of 1000 population structures.
 12. Rinse and repeat steps 1-11 with other matrices. Compare the transient dynamics of different species, populations, and of the same population at different points in time. Note how diverse the dynamics can be!
-

Extras

For a review of transient indices, see Stott et al. 2011 *Ecol. Lett.*, 14, 959-970.