

Y Lab 6: Solving differential equations

In this lab, we will investigate the dynamics of one- and two-species competition models in continuous time. To do this,

The basic one-species model with density-dependent growth is given by the logistic equation:

$$dn/dt = r n (1-n/k)$$

The basic two-species model with density-dependent growth and competition is given by the Lotka-Volterra equations:

$$dn_1/dt = r_1 n_1 (1-(n_1+a_{12}*n_2)/k_1)$$

$$dn_2/dt = r_2 n_2 (1-(n_2+a_{21}*n_1)/k_2)$$

In general, we would like to find the solutions of these (and similar) equations in order to be able to describe population sizes as a function of time. *Mathematica* can help us do this.

Y Numerically solving differential equations:

First determine the parameters needed to solve the equation numerically:

r ,
 k ,
 n_0 (the initial population size at time 0)

Mathematica numerically solves differential equations by a series of guesses. It starts with a rough guess of the general closer to dn/dt .

To do this, though, you need to tell *Mathematica* what range of time you want to focus on (because it can't use the above

t_{min} (the first time point that you want to know about),
 t_{max} (the last time point that you want to know about).

To numerically solve a differential equation, we use `NDSolve`:

```
NDSolve[{equations you want to hold true},
        {variables you want to solve for}, {t,tmin,tmax}]
```

In the case of the logistic model, we have two equations we want to hold true:

the differential equation

$$n'[t] == r*n[t]*(1-n[t]/k),$$

and the initial condition

$$n[0] == n_0.$$

We want to solve for one variable: $n[t]$.

[NOTE: The `==` (2 equal signs) is used to denote an equation that we want to be true. We're not setting the right hand

Therefore, the `NDSolve` command we would use is:

```
NDSolve[{n'[t] == r*n[t]*(1-n[t]/k),
        n[0] == n0},
        {n[t]}, {t,tmin,tmax}]
```

Mathematica won't solve this yet, since it doesn't know what the parameters are. An elegant way to go about the problem

```
Clear[logistic,n]
logistic[r_,k_,n0_,tmin_,tmax_] :=
  logistic[r,k,n0,tmin,tmax] =
    NDSolve[{n'[t] == r*n[t]*(1-n[t]/k),
            n[0] == n0},
            {n[t]}, {t,tmin,tmax}]
```

[NOTE: `logistic[r_,k_,n0_,tmin_,tmax_] :=
logistic[r,k,n0,tmin,tmax] =`

This notation ensures that *Mathematica* remembers the numerical solution that it finds.]

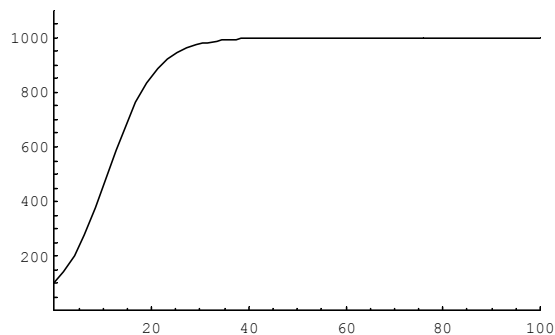
When we call `logistic` with all the parameters, *Mathematica* will return its best numerical guess for the function, $n[t]$,

```
logistic[0.1,1000,100,0,100]
88n@tD® InterpolatingFunction@880., 100.<<, <>D@tD<<
```

We can then plot $n[t]$ by substituting this function for $n[t]$.

For example, with $r=0.2$, $K=1000$, $n_0=100$, $t_{min}=0$, and $t_{max}=100$:

```
Plot[Evaluate[n[t]/.logistic[0.2,1000,100,0,100]],{t,0,100},
PlotRange->{{0,100},{0,1100}}]
```



[NOTE: "`Evaluate[n[t]/.logistic[0.2,1000,100,0,100]]`" tells *Mathematica* to not only assign $n[t]$ the numerical solution list is then plotted. Without the command `Evaluate[]`, *Mathematica* would just keep the numerical solution in `NDSolve`

Question # 1

Plot $n[t]$ as above for the following values of r (try others if you wish):

-0.1, -1, 0.1, 1.

In which cases is $n=K$ approached and in which cases is $n=0$ approached?

Do these results match the local stability analysis you performed in a previous homework?

Question # 2

Write the *Mathematica* code to find the numerical solution to the two-species competition model. Hint: write a function within which are the two differential equations describing the rate of change of species 1 and 2 (see above). Don't forget

```

Clear[competition,n1,n2]
competition[r1_,k1_,a12_,n10_,r2_,k2_,a21_,n20_,tmin_,tmax_] :=
  competition[r1,k1,a12,n10,r2,k2,a21,n20,tmin,tmax] =
    NDSolve[{n1'[t]== r1*n1[t]*(1-(n1[t]+a12*n2[t])/k1),
      n2'[t] == r2*n2[t]*(1-(n2[t]+a21*n1[t])/k2),
      n1[0] == n10, n2[0] == n20},
      {n1[t],n2[t]}, {t,tmin,tmax}]

```

Plot $n1[t]$ and compare this to the graph of the logistic. Use the following parameter values:

- (a) $r1=0.1, k1=1000, n10=100, a12=0, r2=0.1, k2=1000, n20=100, a21=0, tmin=0, tmax=100$.
- (b) $r1=0.1, k1=1000, n10=100, a12=0.2, r2=0.1, k2=1000, n20=100, a21=0, tmin=0, tmax=100$.
- (c) $r1=0.1, k1=1000, n10=100, a12=2.0, r2=0.1, k2=1000, n20=100, a21=0, tmin=0, tmax=100$.
- (d) At least one more parameter set of your choosing.

For example, you can use the plot function:

```

Plot[Evaluate[n1[t] /. competition[0.1,1000,0,100,0.1,1000,0,100,0,100]],{t,
  PlotRange->{{0,100},{0,1100}}]

```

¶ Analytical solutions of differential equations:

Mathematica knows the general analytical solution to some (but by no means all) differential equations.

This is helpful in many cases where solving the differential equation is laborious or difficult.

The procedure is very similar, except we use DSolve rather than NDSolve (the N stands for "numerical").

The main difference is that DSolve solves the differential equation generally, i.e., it gives you a formula for the function

The command we would use to solve the logistic one-species logistic model is:

```

DSolve[{n'[t] == r*n[t]*(1-n[t]/k), n[0]==n0}, {n[t]}, t]

```

Enter the above equation into *Mathematica* and see what it gives you.

DSolve can, in principle, also be used to solve systems of differential equations, such as the Lotka-Volterra equations such systems (including the Lotka-Volterra system) it is mathematically impossible to obtain analytical solutions, and

Try DSolve with some of the differential equations from the homework and see whether you can obtain analytical solutions!

In general:

¶ USE THE COMMANDS NDSOLVE[] AND DSOLVE[] FOR HOMEWORK #6 WHENEVER