Mathematical Techniques in Evolution and Ecology

Numerical and graphical techniques

Based on Chapter 4 in Otto and Day (2007)

Spring Quarter 2015

Simon Aeschbacher
saeschbacher@ucdavis.edu

Outline

Goal

- To describe numerical and graphical techniques for exploring a model's behaviour

Concepts

- Initial conditions
- Variable over time plots
- General solution
- Variable versus variable plots

See OD2007:

- Phase-plane diagrams
- Vector-field plots
- Null clines

Motivation

In the following units, we will explor analytical techniques to study mathematical models. Before doing so, in this unit we learn how we can get a feel for a model's behaviour through **numerical** and **graphical techniques**.

- Basic idea: Specify numerical values for all of the parameters and for the initial values of each variable, and then use the model's equations to predict what happens over time
- Downside: Not well suited for understanding the model in general terms
- Three main types of graphs
  - Variable(s) as a function of time: Illustrate the dynamics for a given set of parameters
  - (Change in) a variable as a function of the variable itself: Illustrate conditions under which a variable grows or shrinks
  - One variable as a function of another variable: Illustrate the effect of interactions

Plots of variables over time

Initial conditions

Initial conditions, also called “starting conditions” are the numerical values of every variable at the initial time point. Often, the qualitative and quantitative behaviour of the dynamics depends on the initial conditions.

Variabe over time plots

Illustrate the behaviour of the variable of interest (vertical axis) as a function of time (horizontal axis).

- For discrete-time models, the recursion equations allow us to plot successive values of the variables, given the initial conditions
- Sometimes, a **general solution** can be found. A general solution is an equation that *gives the value of a variable at any future point in time* as a function of the *parameters*, the *initial conditions*, and the *time.*

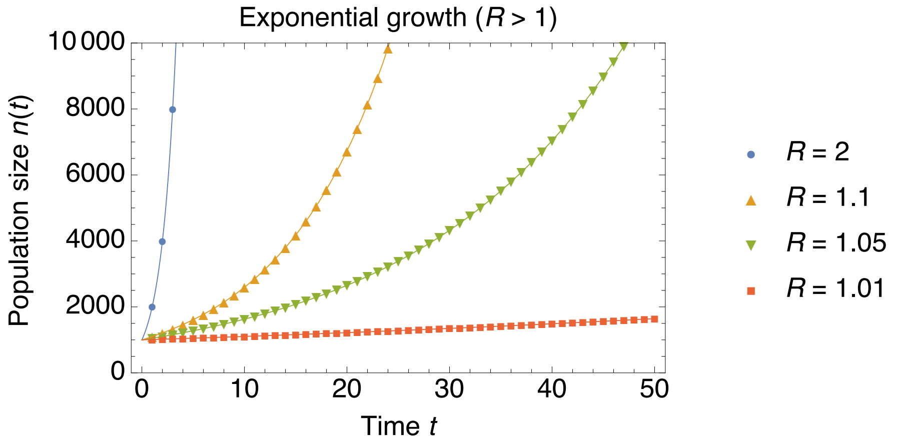
Exercises

- (a) Starting from the *discrete-time* recursion equation for the exponential growth model, $n(t+1) = R\, n(t)$, derive the *general solution*, i.e. express $n(t)$ as a function of R, $n(0)$, and t.
- (b) Tind the explicit solution for the *continuous-time* version, starting from the differential equation $\frac{dn}{dt} = r\, n(t)$, and making an educated guess based on the general solution in discrete time obtained above.

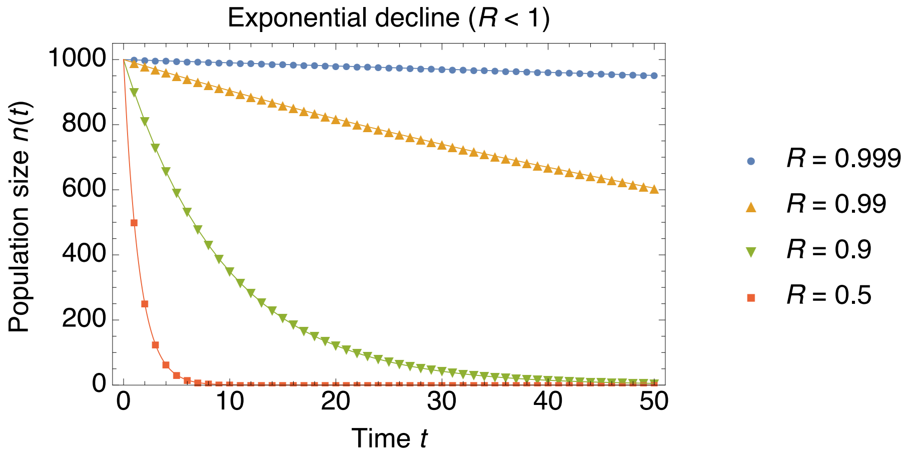
Example: Discrete-time dynamics of the exponential growth model

The dots in the plots below have been produced by iterating the discrete-time recursion equations. The curves were obtained from the explicit solution. For details, see part (a) of the Exercise above.

`expGrowthPlot1`



`expGrowthPlot2`



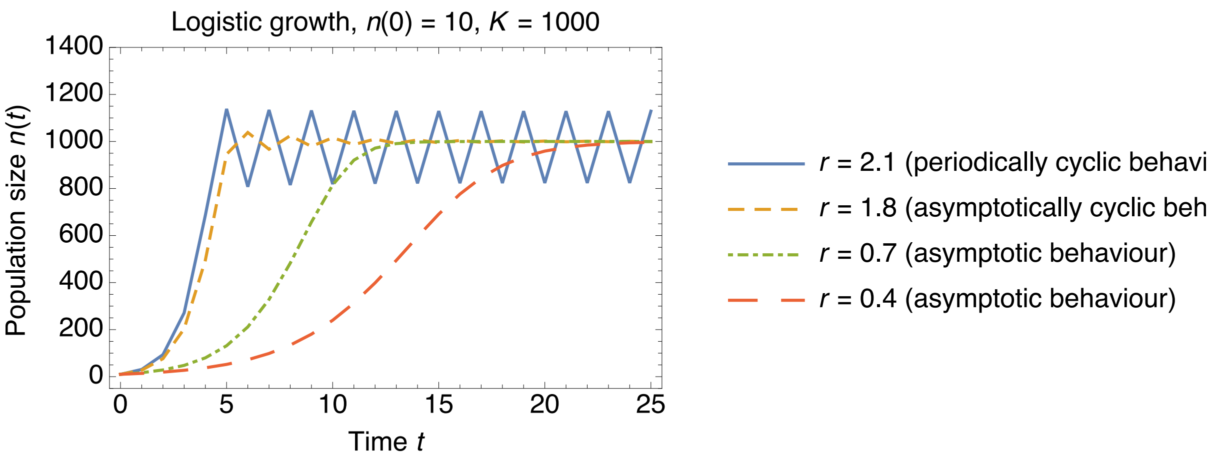
Example: Logistic growth model

For most models, it is not so simple to obtain a general solution as it was for the exponential growth model above. Trying to iterate the recursion equation for the logistic growth model becomes nasty pretty soon:

$$n(t+1) = n(t) + r\,n(t)\left(1 - \frac{n(t)}{K}\right). \tag{1}$$

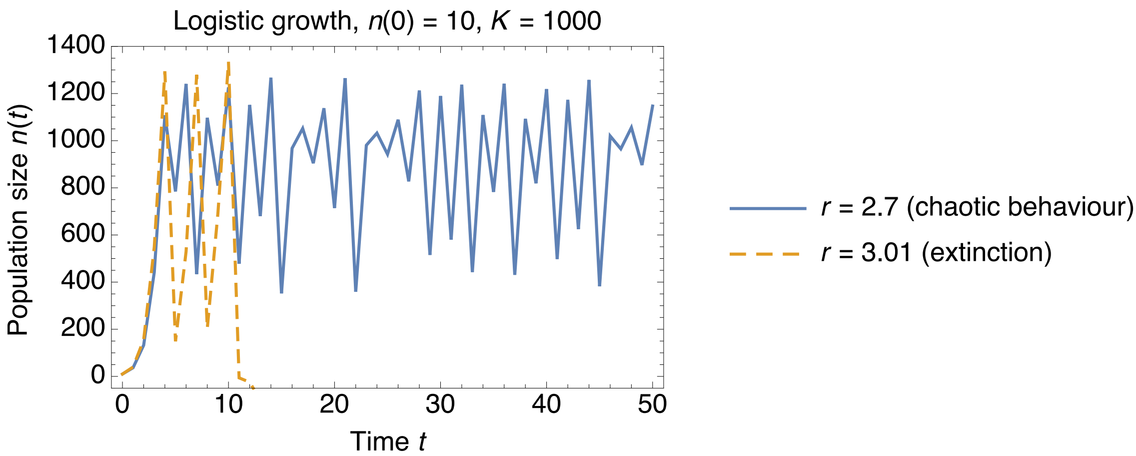
However, we can use a calculator (e.g. *Mathematica*, *Maple*, *Matlab*, *R*) to numerically iterate the recursion.

```
logisticGrowthPlotRec1
```



$$n\,(t+1) = n(t) + \; r\, n(\,t)\left(1 - \frac{n(t)}{K}\right).$$

```
logisticGrowthPlotRec2
```



Chaos

- Superficially, chaotic dynamics (see example above, with $r = 2.7$) appear to be erratic and random, but they are, in fact, entirely *predictable and deterministic*. If we know the values of the parameters and variables, we can predict the population size at any future time simply by iteration of the recursion equation.
- On the other hand, chaotic dynamics imply that the *future state* of the system will be *sensitive to imprecision in the calculations*, e.g. due to rounding errors.
- There is a *continued debate about the importance of chaos* (and periodic cyclies) in biology (e.g. Dennis et al. 2001, *Ecol. Monogr.* 71:277–303; Hastings et al. 1993, *Ecology* 72:896–903; Turchin and Ellner 2000, *Ecology* 81:3099–3116).
- Do we see chaotic behaviour in the continuous-time version, too?

Discretising a differential equation: Euler's Method

In a later unit, we will learn how to find the general solution from the differential equation of the logistic growth model, which we could then plot. Here, we instead analyse the differential equation numerically. This can be a useful technique if a general solution cannot be found.

Let us start from the definition of the differential:

$$\frac{dn}{dt} := \lim_{\Delta t \to 0} \frac{n\,(t + \Delta t) - n(t)}{\Delta t}.$$

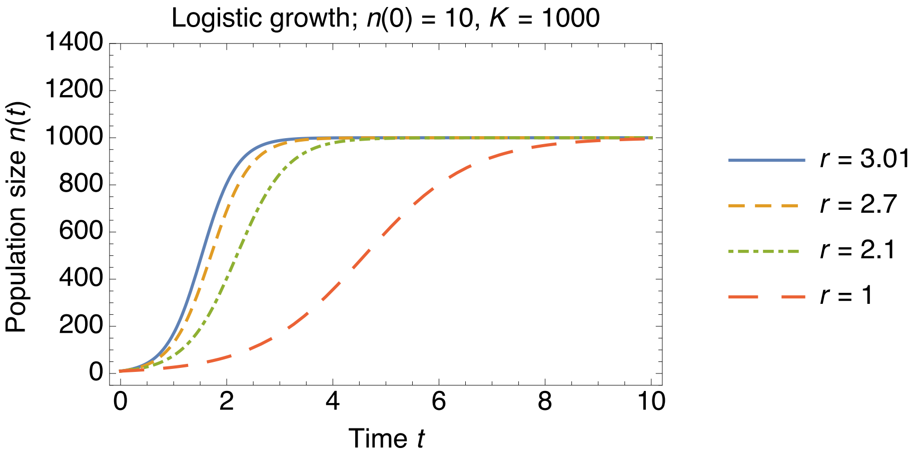
In practice, we cannot let Δt go to infinity in our calculations. However, we can choose a very small Δt and then approximate the differential as

$$\frac{dn}{dt} \approx \left[\frac{n(t + \Delta \mathrm{t}) - n(t)}{\Delta \mathrm{t}}\right]_{\text{for } \Delta t \text{ very small}}. \tag{2}$$

Hence,

$$n(t + \Delta \mathrm{t}) \approx n(t) + \Delta \mathrm{t}\,\frac{dn}{dt} \tag{3}$$

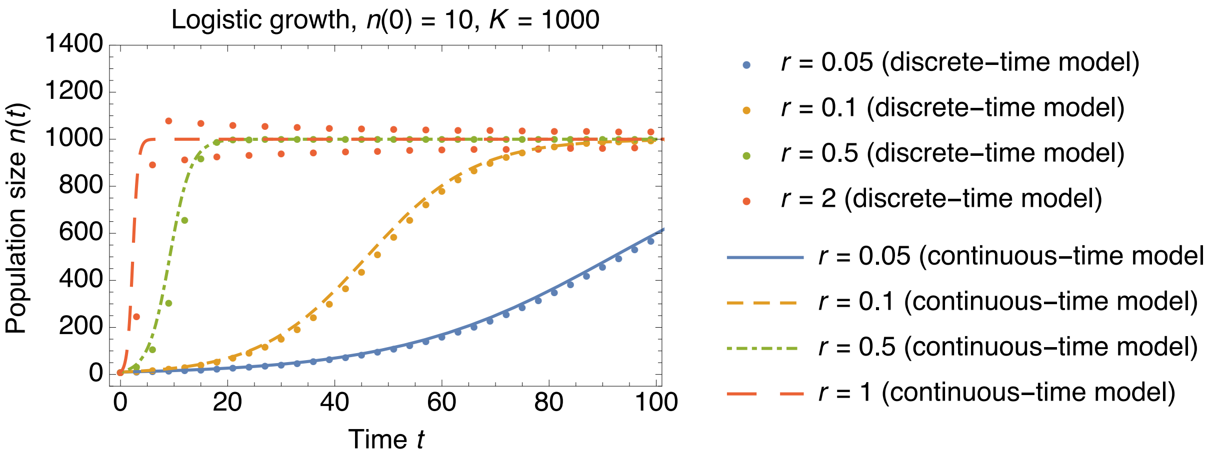
```
logisticGrowthContPlot
```



We see no cyclic or chaotic behaviour in continuos time! Why is this so?

Comparing the discrete-time model (recursion equation) with the discretised continuous-time model, we see that the two agree well as long as r is small:

```
logisticGrowthCompareContDiscPlot
```



Hence, results obtained with the continuous-time approximation are **robust** w.r.t. to the discrete-time version as long as r is not too large.

Example: Models of natural selection

In the logistic growth model the agreement between the discrete-time and continuous-time version was sensitive to the choices of parameter values (and hence to underlying assumptions). In contrast, the haploid and diploid models of natural selection display very similar dynamics in discrete and continuous time (cf. Fig. 4.6 in OD2007).

However, the behaviour of models of selection is *sensitive to the ordering of fitnesses*, i.e. to the relative magnitude of the fitnesses:

- **Overdominance**, or heterozygote advantage

$$W_{AA} < W_{Aa} > W_{aa} \tag{4}$$

- **Directional selection**

$$W_{AA} > W_{Aa} > W_{aa} \tag{5}$$

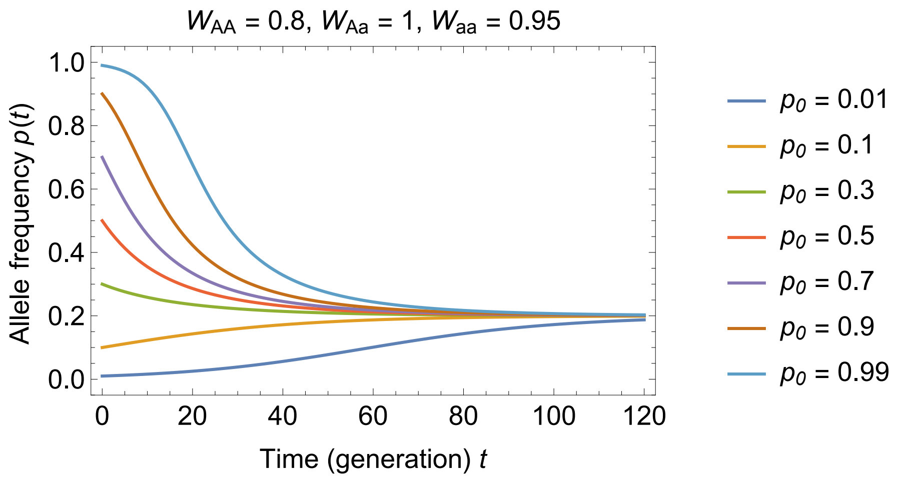
$$W_{AA} < W_{Aa} < W_{aa} \tag{6}$$

- **Underdominance**, or heterozygote disadvantage

$$W_{AA} > W_{Aa} < W_{aa} \tag{7}$$

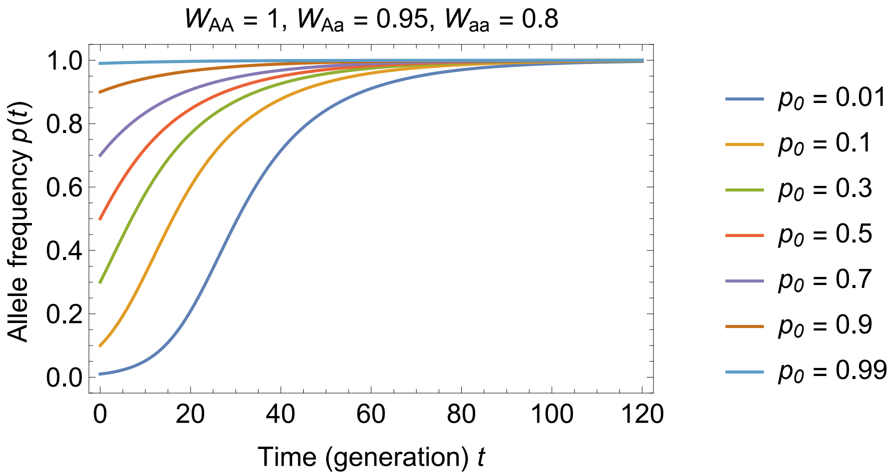
Overdominance (heterozygote advantage)

diplSelOverdomPlot



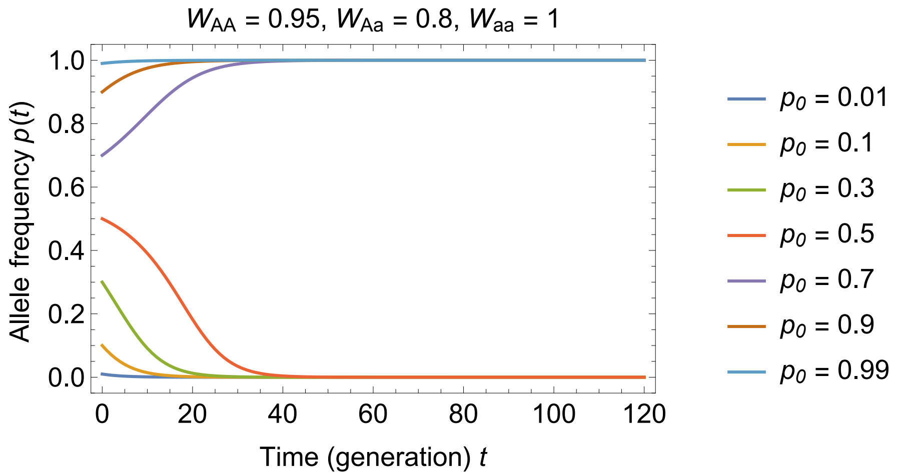
Directional selection in favour of allele *A*

diplSelDirectPlot



Underdominance (heterozygote disadvantage)

diplSelUnderdomPlot



Plots of variables as a function of themselves

Variable-versus-variable plots illustrate the *future state or change* of a variable (vertical axis) *versus* the *current state* of the variable (horizontal axis):

"function of n(t+1) ~ function of n(t)"

Such plots help us understand the dynamics of *one-variable models*, as they illustrate how the direction of change depends on the current state.

Value of a variable at time $t+1$ versus time t

We plot the recursion equation as a function of the variable itself:

$$n(t+1) \sim n(t)$$

Example 1: Haploid model of selection

We plot the recursion equation for $p(t)$,

$$p(t+1) = \frac{W_A\, p(t)}{W_A\, p(t) + W_a\,(1 - p(t))}$$

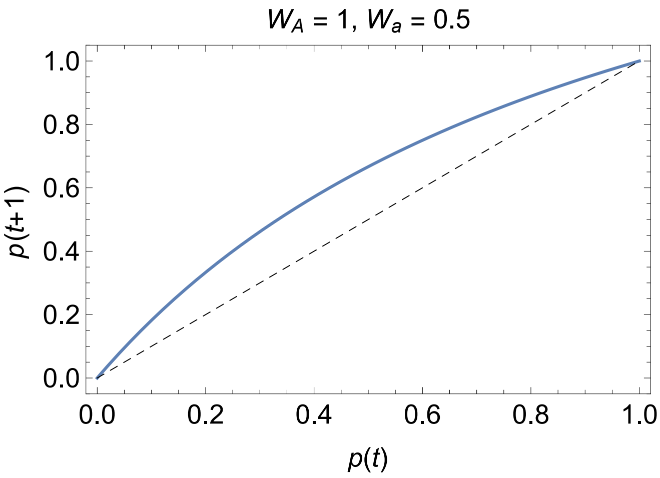
as a function of $p(t)$ itself.

Looking at the following two plots:

- What does the diagonal line represent?
- What does it mean if the curve crosses the diagonal?
- Use the **cobwebbing** technique to determine how the variable changes over time

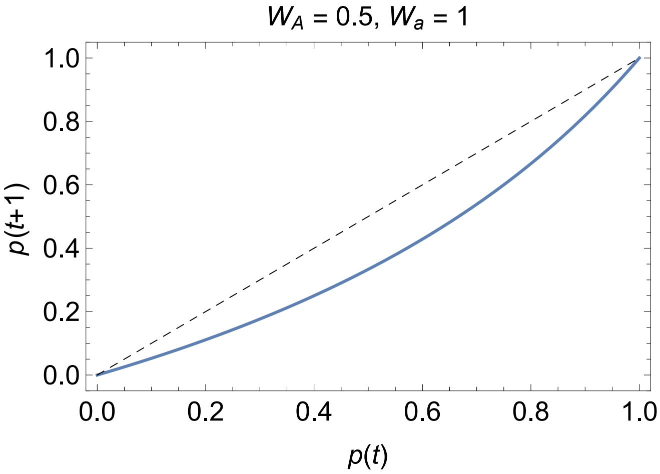
In[194]:=
```
haplSelRecCWPlot[1, 0.5]
```

Out[194]=



In[195]:=
```
haplSelRecCWPlot[0.5, 1]
```

Out[195]=



Example 2: Diploid model of selection

We plot the recursion equation for $p(t)$,

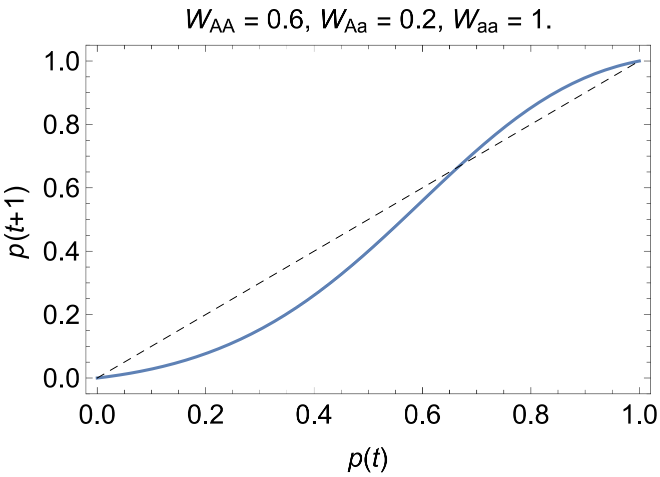
$$p\,(t+1) = \frac{W_{\text{AA}}\,p(t) + W_{\text{Aa}}(1 - p(t))}{W_{\text{AA}}\,p^2(t) + 2\,W_{\text{Aa}}\,p(t)\,(1 - p(t)) + W_{\text{aa}}(1 - p(t))^2}\,p(t)$$

as a function of $p(t)$ itself.

In the following three plots, use the cobwebbing technique to determine the type of the equilibria.

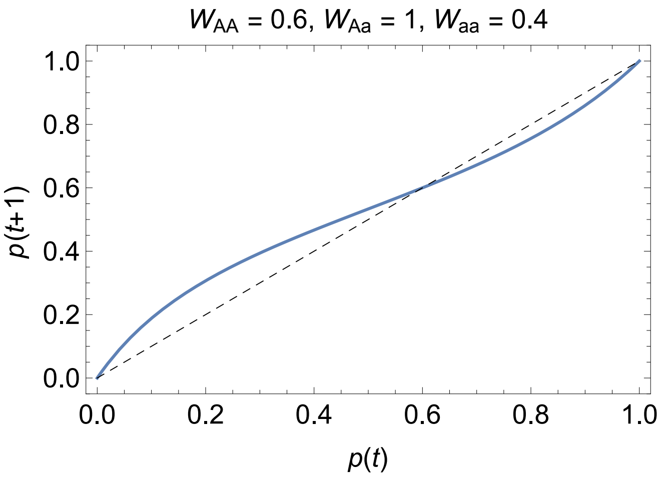
In[196]:= `diplSelRecCWPlot[0.6, 0.2, 1.]`

Out[196]=



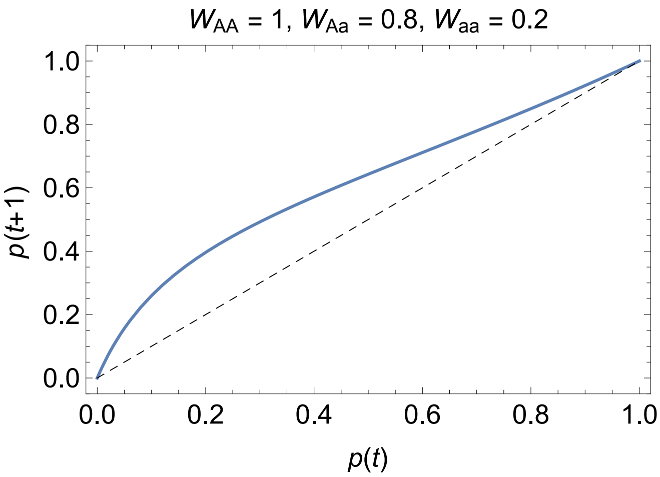
In[197]:=
```
diplSelRecCWPlot[0.6, 1, 0.4]
```

Out[197]=



In[198]:=
```
diplSelRecCWPlot[1, 0.8, 0.2]
```

Out[198]=



Example 3: Logistic growth model

We plot the recursion equation for $n(t)$,

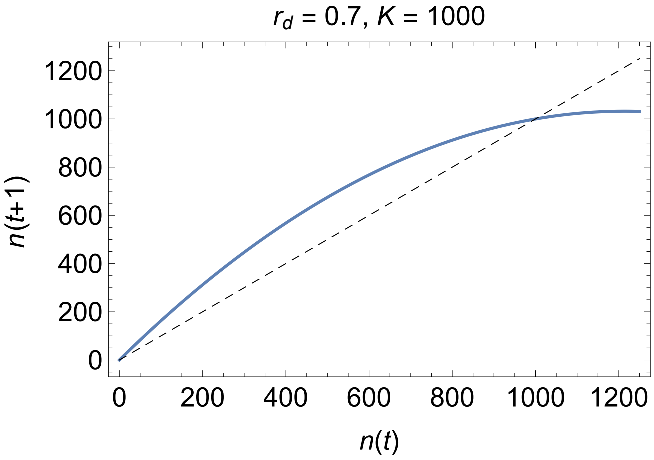
$$n(t+1) = n(t) * \left(1 + r - \frac{r}{K} n(t)\right) = n(t) + r\, n(t) \left(1 - \frac{n(t)}{K}\right)$$

as a function of $n(t)$ itself.

In the following three plots, use the cobwebbing technique to determine the type of the equilibria.

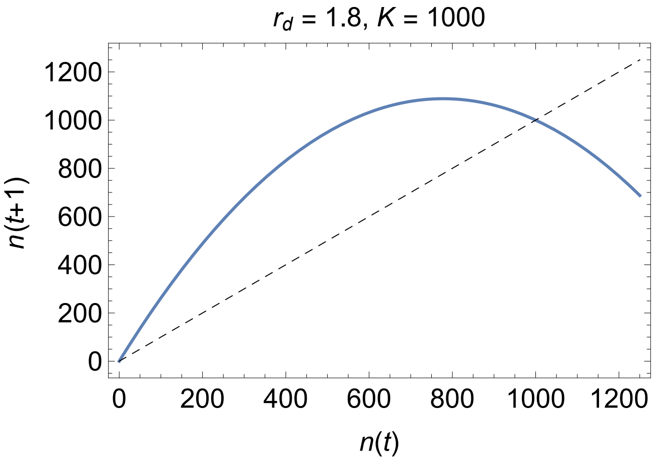
In[199]:= `logisticGrowthCWPlot[0.7, 1000]`

Out[199]=



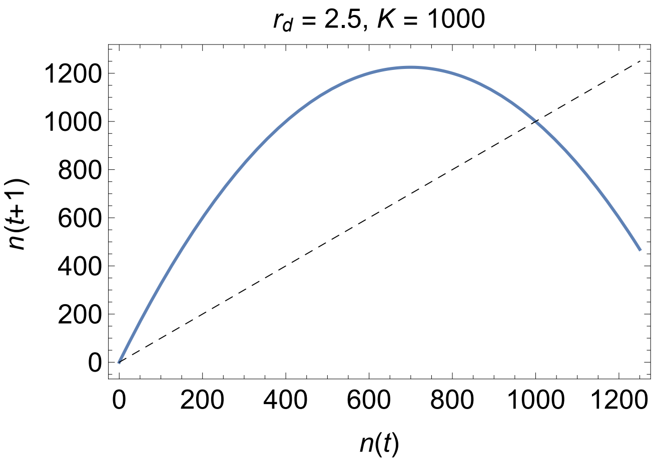
In[200]:= `logisticGrowthCWPlot[1.8, 1000]`

Out[200]=



In[201]:= `logisticGrowthCWPlot[2.5, 1000]`

Out[201]=



The slope of the recursion at the equilibrium

Looking at the various plots and cobwebs above, we see that an equilibrium is

- *locally stable* if the slope of the recursion at the equilibrium is < 1, and
- *unstable* if the slope of the recursion at the equilibrium is > 1.

Change in a variable versus variable at time *t* [Go on here]

We plot the difference or the differential equation as a function of the variable itself:

$$\Delta \mathrm{n}(t) \sim n(t)$$

$$\frac{d n(t)}{d t} \sim n(t)$$

Example: Diploid model of natural selection

Consider the following fitness parametrisation:

$$W_{AA} = 1 + s \qquad W_{Aa} = 1 + h\,s \qquad W_{aa} = 1. \tag{8}$$

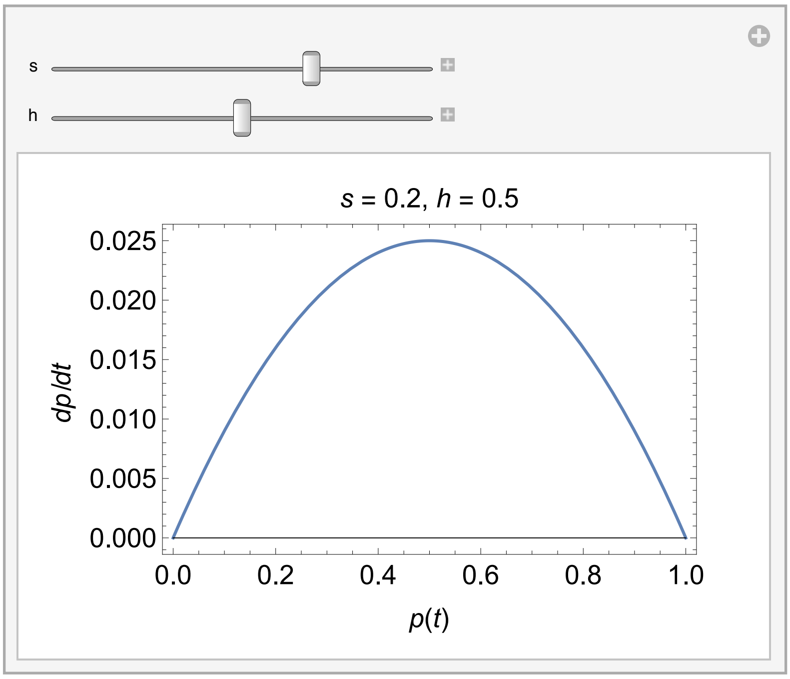
We plot the continuous-time differential equation

$$\frac{d\,p(t)}{d\,t} = s\,p(1-p)\,(p + h\,(1 - 2\,p))$$

as a function of $p(t)$.

In[202]:=
```
Manipulate[diplSelDotPlot[s, h], {{s, 0.2}, -0.5, 0.5}, {{h, 0.5}, -1, 2}]
```

Out[202]=



Recipe 4.1: Identifying equilibria graphically

A system at equilibrium does not change over time.

- In a plot of $n(t+1)$ versus $n(t)$, any point where the recursion equation *crosses the diagonal line* represents an equilibrium, because $n(t+1) = n(t)$ for such a point.
- In a plot of Δn versus $n(t)$, any point where the difference equation *crosses the horizontal axis* represents an equilibrium, because $\Delta n = 0$ for such a point.
- In a plot of dn/dt versus $n(t)$, any point where the differential equation *crosses the horizontal axis* represents an equilibrium, because $dn/dt = 0$ for such a point.

Recipe 4.2: Identifying direction of change graphically

- In a plot of $n(t+1)$ versus $n(t)$, if the recursion equation lies above the diagonal line, the variable increases over time, because $n(t+1) > n(t)$, and vice versa.
- In a plot of Δn versus $n(t)$, if the difference equation lies above the horizontal line, the varuable increases over time, because $\Delta n > 0$, and vice versa.
- In a plot of dn/dt versus $n(t)$, if the differential equation lies above the horizontal line, the variable increases over time, because $dn/dt > 0$, and vice versa.

Multiple variables

With multiple variables, the plotting techniques discussed are of little help, as they would require plotting in more than two dimensions. Otto and Day (2007) discuss several plotting techniques for multiple variables: Phase-plane diagrams, vector-field plots, and null clines. Here we only briefly visit the null clines.

Null clines

Start from a two-dimensional plot with two orthogonal axes, each axis denoting one variable. We call this plot a variable-by-variable plot.

A **null cline** is a *curve on a variable-by-variable plot that indicates when one variable remains constant*. Different variables typically have different null clines.

Null clines are helpful for two-dimensional problems, i.e. for models with two variables. If there are more than two variables, it may sometimes be useful to look at pairs of variables, holding the other variables constant at a certain value.

Using null clines, we can distinguish regions of the variable-by-variable plot in which each variable grows or shrinks.

Points where *null clines intersect* correspond to *equilibria*.

Example: Two-species Lotka–Volterra model of competition

To identify when the two population sizes remain constant, we start from the difference equations (we could just as well start from the differential equations):

$$\Delta n_1 = r_1\, n_1(t)\left(1 - \frac{n_1(t) + \alpha_{12}\, n_2(t)}{K_1}\right)$$

$$\Delta n_2 = r_2\, n_2(t)\left(1 - \frac{n_2(t) + \alpha_{21}\, n_1(t)}{K_2}\right).$$

Recall that α_{ij} is the extent to which species *i* feels competition by species *j* for common resources.

The number of individuals of species 1, $n_1(t)$, stays constant either if $r_1 = 0$, $n_1(t) = 0$, or if $K_1 = n_1(t) + \alpha_{12}\, n_2(t)$. Obviously, the first two cases are of little interest – they correspond to special parameter or variable values. The third case, however, helps us construct the null cline. All we need to do is rearrange the condition such that it has the form "$n_1(t)$ as a function of $n_2(t)$"; we can then plot it on our variable-by-variable plot. Doing this yields

$$n_1(t) = K_1 - \alpha_{12}\, n_2\,(t) \tag{9}$$

Similarly, the null cline for species 2 is given by

$$n_2(t) = K_2 - \alpha_{21}\, n_1\,(t) \tag{10}$$

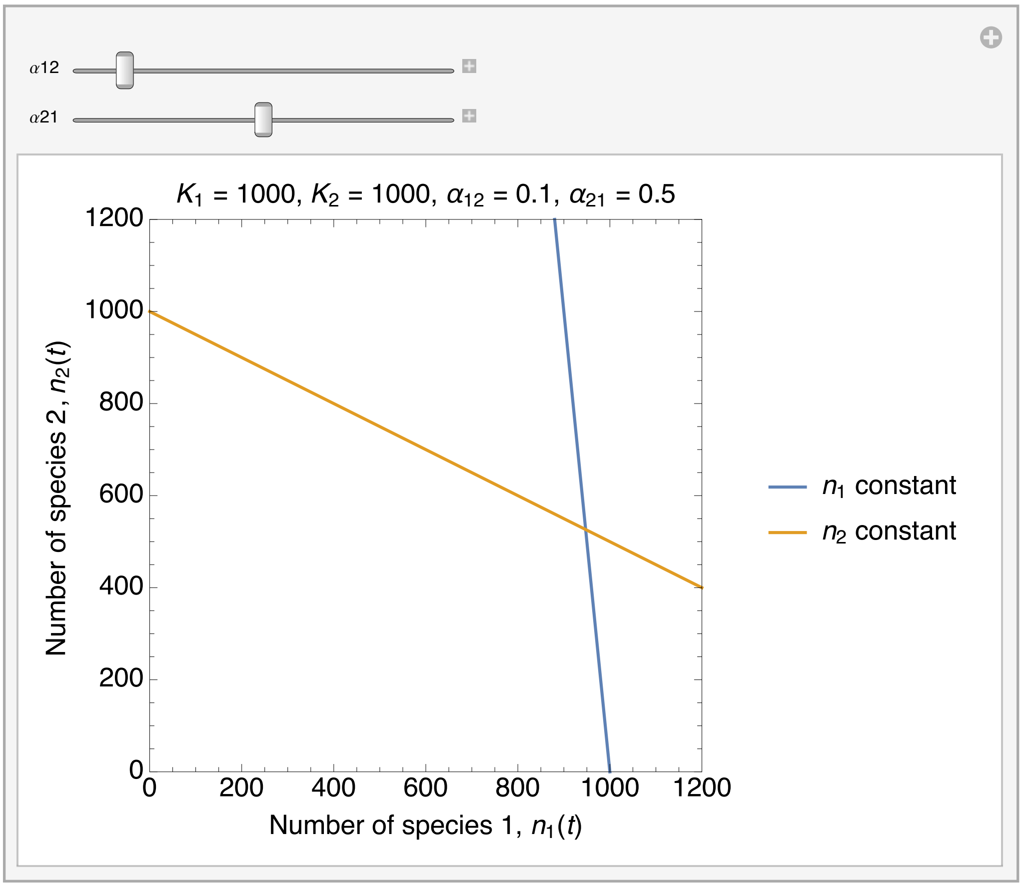
In practice, we want to express the first equation in the form "$n_2(t)$ as a function of $n_1(t)$", so that we can plot the two curves on our variable-by-variable plot.

Exercise

- In the plot below, for each of the four sections of the plane, determine whether n_1 and n_2 are increasing or decreasing.
- What can be said about the point where the two null clines intersect?

In[203]:=
```
LVCompNullClinesPlot
```

Out[203]=



Initialisation cells

```
In[204]:= expGrowthRec[R_, 0] := 1000
          expGrowthRec[R_, t_] := expGrowthRec[R, t] = expGrowthRec[R, t - 1] * R

In[206]:= expGrowthDiscData[R_, tMax_] := Table[{t, expGrowthRec[R, t]}, {t, 1, tMax}]

In[207]:= expGrowthExpl[R_, n0_, t_] := R^t n0

In[208]:= expGrowthPlotRec1 := ListPlot[
            {expGrowthDiscData[2, 50], expGrowthDiscData[1.1, 50],
             expGrowthDiscData[1.05, 50], expGrowthDiscData[1.01, 50]},
            PlotRange → {Full, {0, 10 000}},
            Frame → True,
            FrameLabel →
             {{"Population size n(t)", ""}, {"Time t", "Exponential growth (R > 1)"}},
            LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
            PlotMarkers → {"●", "▲", "▼", "■"},
            PlotLegends → {"R = 2", "R = 1.1", "R = 1.05", "R = 1.01"}
           ]

In[209]:= expGrowthPlotRec2 := ListPlot[
            {expGrowthDiscData[0.999, 50], expGrowthDiscData[0.99, 50],
             expGrowthDiscData[0.9, 50], expGrowthDiscData[0.5, 50]},
            PlotRange → {Full, {0, 1050}},
            Frame → True,
            FrameLabel →
             {{"Population size n(t)", ""}, {"Time t", "Exponential decline (R < 1)"}},
            LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
            PlotMarkers → {"●", "▲", "▼", "■"},
            PlotLegends → {"R = 0.999", "R = 0.99", "R = 0.9", "R = 0.5"}
           ]

In[210]:= expGrowthPlotExpl1 := Plot[
            {expGrowthExpl[2, 1000, t], expGrowthExpl[1.1, 1000, t],
             expGrowthExpl[1.05, 1000, t], expGrowthExpl[1.01, 1000, t]}, {t, 0, 50},
            PlotRange → {Full, {0, 10 000}},
            Frame → True,
            FrameLabel →
             {{"Population size n(t)", ""}, {"Time t", "Exponential decline (R < 1)"}},
            LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
            PlotStyle → Thickness[Small] (*,
            PlotLegends→{"R = 2","R = 1.1","R = 1.05","R = 1.01"}*)
           ]

In[211]:= expGrowthPlotExpl2 := Plot[
            {expGrowthExpl[0.999, 1000, t], expGrowthExpl[0.99, 1000, t],
             expGrowthExpl[0.9, 1000, t], expGrowthExpl[0.5, 1000, t]}, {t, 0, 50},
            PlotRange → {Full, {0, 1050}},
            Frame → True,
            FrameLabel →
             {{"Population size n(t)", ""}, {"Time t", "Exponential decline (R < 1)"}},
            LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
            PlotStyle → Thickness[Small] (*,
            PlotLegends→{"R = 0.999","R = 0.99","R = 0.9","R = 0.5"}*)
           ]

In[212]:= expGrowthPlot1 := Show[{expGrowthPlotRec1, expGrowthPlotExpl1}]
          expGrowthPlot2 := Show[{expGrowthPlotRec2, expGrowthPlotExpl2}]
```

In[214]:=
```
logisticGrowthRec[r_, K_, 0] := 10
logisticGrowthRec[r_, K_, t_] :=
 logisticGrowthRec[r, K, t] = logisticGrowthRec[r, K, t - 1] +
   r logisticGrowthRec[r, K, t - 1] (1 - logisticGrowthRec[r, K, t - 1] / K)
```

In[216]:=
```
logisticGrowthDiscData[r_, K_, tMax_, tStep_] :=
 Table[{t, logisticGrowthRec[r, K, t]}, {t, 0, tMax, tStep}]
```

In[217]:=
```
logisticGrowthPlotRec1 := ListPlot[
   {logisticGrowthDiscData[2.1, 1000, 25, 1],
    logisticGrowthDiscData[1.8, 1000, 25, 1], logisticGrowthDiscData[
     0.7, 1000, 25, 1], logisticGrowthDiscData[0.4, 1000, 25, 1]},
   PlotRange → {Full, {-50, 1400}},
   Joined → True,
   PlotStyle → {{}, Dashing[0.02], DotDashed, Dashing[0.04]},
   Frame → True,
   FrameLabel → {{"Population size n(t)", ""},
     {"Time t", "Logistic growth, n(0) = 10, K = 1000"}},
   LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
   PlotLegends → {"r = 2.1 (periodically cyclic behaviour)",
     "r = 1.8 (asymptotically cyclic behaviour)",
     "r = 0.7 (asymptotic behaviour)", "r = 0.4 (asymptotic behaviour)"}
  ]
```

In[218]:=
```
logisticGrowthPlotRec2 := ListPlot[
   {logisticGrowthDiscData[2.7, 1000, 50, 1],
    logisticGrowthDiscData[3.01, 1000, 50, 1]},
   PlotRange → {Full, {-50, 1400}},
   Joined → True,
   PlotStyle → {{}, Dashing[0.02], DotDashed, Dashing[0.04]},
   Frame → True,
   FrameLabel → {{"Population size n(t)", ""},
     {"Time t", "Logistic growth, n(0) = 10, K = 1000"}},
   LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
   PlotLegends → {"r = 2.7 (chaotic behaviour)", "r = 3.01 (extinction)"}
  ]
```

In[219]:=
```
Clear[logisticGrowthDiscrDot, t, KK, r]
logisticGrowthDiscrDot[r_, KK_, d_, 0] := 10;
logisticGrowthDiscrDot[r_, KK_, d_, t_] := logisticGrowthDiscrDot[r, KK, d, t] =
   logisticGrowthDiscrDot[r, KK, d, t - d] + r logisticGrowthDiscrDot[r, KK, d, t - d]
     (1 - logisticGrowthDiscrDot[r, KK, d, t - d] / KK) d
```

In[222]:=
```
(* To compare with the discrete-time version, we choose Δt = 1. *)
logisticGrothDiscrDotData[r_, K_, tMax_] :=
 Table[{t, logisticGrowthDiscrDot[r, K, 1, t]}, {t, 0, tMax}]
```

In[223]:=
```
logisticGrowthDiscrDotPlot := ListPlot[
   {logisticGrothDiscrDotData[0.05, 1000, 100],
    logisticGrothDiscrDotData[0.1, 1000, 100]},
   PlotRange → {Full, {-50, 1400}},
   PlotStyle → {{}, Dashing[0.02], DotDashed, Dashing[0.04]},
   Frame → True,
   FrameLabel → {{"Population size n(t)", ""},
     {"Time t", "Logistic growth, n(0) = 10, K = 1000"}},
   LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
   PlotLegends → {"r = 0.05 (discretised continuous-time model)",
     "r = 0.1 (discretised continuous-time model)"},
   Joined → True
  ]
```

```
In[224]:= logisticGrowthDiscrPlot := ListPlot[
    {logisticGrowthDiscData[0.05, 1000, 100, 3],
     logisticGrowthDiscData[0.1, 1000, 100, 3], logisticGrowthDiscData[
      0.5, 1000, 100, 3], logisticGrowthDiscData[2., 1000, 100, 3]},
    PlotRange → {Full, {-50, 1400}},
    PlotStyle → {{}, Dashing[0.02], DotDashed, Dashing[0.04]},
    Frame → True,
    FrameLabel → {{"Population size n(t)", ""},
      {"Time t", "Logistic growth, n(0) = 10, K = 1000"}},
    LabelStyle → Directive[FontFamily → "Helvetica", FontSize → 14],
    PlotLegends →
     {"r = 0.05 (discrete-time model)", "r = 0.1 (discrete-time model)",
      "r = 0.5 (discrete-time model)", "r = 2 (discrete-time model)"},
    PlotMarkers → {•}
   ]

In[225]:= Clear[n];
    logisticGrowthDot[r_, KK_] :=
     NDSolve[{n'[t] == r n[t] (1 - n[t] / KK), n[0] == 10}, n, {t, 0, 200}]

In[227]:= logisticGrowthContPlot := Plot[
    {Evaluate[n[t] /. logisticGrowthDot[3.01, 1000]],
     Evaluate[n[t] /. logisticGrowthDot[2.7, 1000]],
     Evaluate[n[t] /. logisticGrowthDot[2.1, 1000]],
     Evaluate[n[t] /. logisticGrowthDot[1, 1000]]}, {t, 0, 10},
    PlotRange → {Full, {-50, 1400}},
    Frame → True,
    FrameLabel → {{"Population size n(t)", ""},
      {"Time t", "Logistic growth; n(0) = 10, K = 1000"}},
    LabelStyle → Directive[FontSize → 14, FontFamily → "Helvetica"],
    PlotStyle → {{}, Dashing[0.02], DotDashed, Dashing[0.04]},
    PlotLegends → {"r = 3.01", "r = 2.7", "r = 2.1", "r = 1"}
   ]

In[228]:= logisticGrowthContPlot2 := Plot[
    {Evaluate[n[t] /. logisticGrowthDot[0.05, 1000]],
     Evaluate[n[t] /. logisticGrowthDot[0.1, 1000]],
     Evaluate[n[t] /. logisticGrowthDot[0.5, 1000]],
     Evaluate[n[t] /. logisticGrowthDot[2, 1000]]}, {t, 0, 200},
    PlotRange → {Full, {-50, 1400}},
    Frame → True,
    FrameLabel → {{"Population size n(t)", ""},
      {"Time t", "Logistic growth; n(0) = 10, K = 1000"}},
    LabelStyle → Directive[FontSize → 14, FontFamily → "Helvetica"],
    PlotStyle → {{}, Dashing[0.02], DotDashed, Dashing[0.04]},
    PlotLegends →
     {"r = 0.05 (continuous-time model)", "r = 0.1 (continuous-time model)",
      "r = 0.5 (continuous-time model)", "r = 1 (continuous-time model)"}
   ]

In[229]:= logisticGrowthCompareContDiscPlot :=
     Show[logisticGrowthDiscrPlot, logisticGrowthContPlot2]

In[230]:= Clear[diplSelRec]
    diplSelRec[WAA_, WAa_, Waa_, 0, p0_] := p0;
    diplSelRec[WAA_, WAa_, Waa_, t_, p0_] := diplSelRec[WAA, WAa, Waa, t, p0] =
      (WAA diplSelRec[WAA, WAa, Waa, t - 1, p0]^2 + WAa diplSelRec[WAA, WAa, Waa, t - 1, p0]
          (1 - diplSelRec[WAA, WAa, Waa, t - 1, p0])) /
       (WAA diplSelRec[WAA, WAa, Waa, t - 1, p0]^2 + 2 WAa
           diplSelRec[WAA, WAa, Waa, t - 1, p0] (1 - diplSelRec[WAA, WAa, Waa, t - 1, p0]) +
         Waa (1 - diplSelRec[WAA, WAa, Waa, t - 1, p0])^2)
```

```
In[233]:= diplSelData[WAA_, WAa_, Waa_, tMax_, p0_] :=
           Table[{t, diplSelRec[WAA, WAa, Waa, t, p0]}, {t, 0, tMax}]

In[234]:= p0List := {0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99}

          myWAA = 0.8;
          myWAa = 1;
          myWaa = 0.95;
          diplSelOverdomPlot = ListPlot[
             MapThread[
              diplSelData[myWAA, myWAa, myWaa, 120, #1] &,
              {p0List}
             ], PlotRange → {Full, {-0.05, 1.05}},
             Frame → True,
             FrameLabel → {{"Allele frequency p(t)", ""},
               {"Time (generation) t", "WAA = " <> ToString[myWAA] <>
                 ", WAa = " <> ToString[myWAa] <> ", Waa = " <> ToString[myWaa]}},
             LabelStyle → Directive[FontSize → 14],
             Joined → True,
             PlotLegends → Table["p0 = " <> ToString[pInit], {pInit, p0List}]
            ];

          myWAA = 1;
          myWAa = 0.95;
          myWaa = 0.8;
          diplSelDirectPlot = ListPlot[
             MapThread[
              diplSelData[myWAA, myWAa, myWaa, 120, #1] &,
              {p0List}
             ], PlotRange → {Full, {-0.05, 1.05}},
             Frame → True,
             FrameLabel → {{"Allele frequency p(t)", ""},
               {"Time (generation) t", "WAA = " <> ToString[myWAA] <>
                 ", WAa = " <> ToString[myWAa] <> ", Waa = " <> ToString[myWaa]}},
             LabelStyle → Directive[FontSize → 14],
             Joined → True,
             PlotLegends → Table["p0 = " <> ToString[pInit], {pInit, p0List}]
            ];

          myWAA = 0.95;
          myWAa = 0.8;
          myWaa = 1;
          diplSelUnderdomPlot = ListPlot[
             MapThread[
              diplSelData[myWAA, myWAa, myWaa, 120, #1] &,
              {p0List}
             ], PlotRange → {Full, {-0.05, 1.05}},
             Frame → True,
             FrameLabel → {{"Allele frequency p(t)", ""},
               {"Time (generation) t", "WAA = " <> ToString[myWAA] <>
                 ", WAa = " <> ToString[myWAa] <> ", Waa = " <> ToString[myWaa]}},
             LabelStyle → Directive[FontSize → 14],
             Joined → True,
             PlotLegends → Table["p0 = " <> ToString[pInit], {pInit, p0List}]
            ];

In[247]:= haplSelRecImpl[WA_, Wa_, p_] := WA p / (WA p + Wa (1 - p))
```

In[248]:=
```
haplSelRecCWPlot[WA_, Wa_] := Plot[
  {haplSelRecImpl[WA, Wa, p], p}, {p, 0, 1},
  PlotRange → {Full, {-0.05, 1.05}},
  Frame → True,
  FrameLabel → {{"p(t+1)", ""},
    {"p(t)", "WA = " <> ToString[WA] <> ", Wa = " <> ToString[Wa]}},
  LabelStyle → Directive[FontSize → 14],
  PlotStyle → {{}, {Dashed, Black, Thickness[Small]}}
 ]
```

In[249]:=
$$\text{diplSelRecImpl[WAA\_, WAa\_, Waa\_, p\_] := }\frac{\text{WAA p + WAa (1 - p)}}{\text{WAA p\^{}2 + 2 WAa p (1 - p) + Waa (1 - p)\^{}2}}\text{ p}$$

In[250]:=
```
diplSelRecCWPlot[WAA_, WAa_, Waa_] := Plot[
  {diplSelRecImpl[WAA, WAa, Waa, p], p}, {p, 0, 1},
  PlotRange → {Full, {-0.05, 1.05}},
  Frame → True,
  FrameLabel → {{"p(t+1)", ""}, {"p(t)", "WAA = " <> ToString[WAA] <>
      ", WAa = " <> ToString[WAa] <> ", Waa = " <> ToString[Waa]}},
  LabelStyle → Directive[FontSize → 14],
  PlotStyle → {{}, {Dashed, Black, Thickness[Small]}}
 ]
```

In[251]:=
$$\text{logisticGrowthImpl[r\_, KK\_, n\_] := n + r n }\left(1-\frac{\text{n}}{\text{KK}}\right)$$

In[252]:=
```
logisticGrowthCWPlot[r_, KK_] := Plot[
  {logisticGrowthImpl[r, KK, n], n}, {n, 0, 1.25 KK},
  PlotRange → {Full, Full},
  Frame → True,
  FrameLabel →
   {{"n(t+1)", ""}, {"n(t)", "rd = " <> ToString[r] <> ", K = " <> ToString[KK]}},
  LabelStyle → Directive[FontSize → 14],
  PlotStyle → {{}, {Dashed, Black, Thickness[Small]}}
 ]
```

$$\text{FullSimplify}\Big[\text{Normal}\Big[\text{Series}\Big[\left(p^2\,\frac{\text{WAA}}{\text{Wbar}}+p\,(1-p)\,\frac{\text{WAa}}{\text{Wbar}}-p\right)\text{ /. }\left\{\text{Wbar}\to p^2\,\text{WAA}+2\,p\,(1-p)\,\text{WAa}+(1-p)^2\,\text{Waa}\right\}\text{ /. }\{\text{WAA}\to 1+s,\ \text{WAa}\to 1+h\,s,\ \text{Waa}\to 1\}\text{ /. }\{s\to\sigma\epsilon\},\ \{\epsilon,0,1\}\Big]\Big]\text{ /. }\{\sigma\to s/\epsilon\}\Big]$$

$$(-1+p)\,p\,(-p+h\,(-1+2\,p))\,s$$

```
% /. {h → 1 / 2} // Simplify
```

$$-\frac{1}{2}\,(-1+p)\,p\,s$$

In[253]:=
```
diplSelDotImpl[s_, h_, p_] := s p (1 - p) (p + h (1 - 2 p))
```

In[254]:=
```
diplSelDotPlot[s_, h_] := Plot[
  {diplSelDotImpl[s, h, p], 0}, {p, 0, 1},
  PlotRange → {Full, Full},
  Frame → True,
  FrameLabel →
   {{"dp/dt", ""}, {"p(t)", "s = " <> ToString[s] <> ", h = " <> ToString[h]}},
  LabelStyle → Directive[FontSize → 14],
  PlotStyle → {{}, {Black, Thickness[Small]}}
 ]
```

In[255]:=
```
LVCompNullClinesPlot := Manipulate[
  Plot[{(K1 - n1) / α12 /. {K1 → 1000, K2 → 1000},
    K2 - α21 n1 /. {K1 → 1000, K2 → 1000}}, {n1, 0, 1200},
   PlotRange → {{0, 1.2 × 1000}, {0, 1.2 × 1000}},
   Frame → True,
   FrameLabel → {{"Number of species 2, n2(t)", ""},
     {"Number of species 1, n1(t)", "K1 = " <> ToString[1000] <> ", K2 = " <>
       ToString[1000] <> ", α12 = " <> ToString[Round[α12, 0.001]] <>
       ", α21 = " <> ToString[Round[α21, 0.001]]}},
   LabelStyle → Directive[FontSize → 14, FontFamily → "Helvetica"],
   AspectRatio → 1,
   PlotLegends → {"n1 constant", "n2 constant"}
  ],
  (*{{K1,1000},10,10000},{{K2,1000},10,10000},*)
  {{α12, 0.1}, 0.0000001, 1}, {{α21, 0.5}, 0.0000001, 1}
 ]
```