The Lotka-Volterra predator-prey system of two differential equations is justly famous. Given a prey species y_1, e.g. rabbits, and a predator species y_2, e.g. foxes, as functions of time, x say, then a plausible simple system is the following.

$$\frac{dy_1}{dx} = p_1 y_1 - p_2 y_1 y_2$$
$$\frac{dy_2}{dx} = -p_3 y_2 + p_4 y_1 y_2$$

Starting in a situation where the population of foxes is at a low ebb then the rabbit population will grow almost exponentially until they provide ample food for the foxes to multiply, and so on.

It is assumed that the parameters p_1, p_2, p_3, and p_4 and also the populations y_1 and y_2 are nonnegative so there are clearly the two stable states

$$y_1 = 0$$
$$y_2 = 0, \text{ and}$$
$$y_1 = p_3/p_4$$
$$y_2 = p_1/p_2$$

of which only the second is of interest.

As the system is autonomous, i.e., with no independent variable on the right hand sides then, apart for the meaningful singular stable state referred to, it is possible to eliminate x and consider the derivative

$$\frac{dy_1}{dy_2} = \frac{p_1 y_1 - p_2 y_1 y_2}{-p_3 y_2 + p_4 y_1 y_2}.$$

In other words, given an initial point $y_1(0) = \alpha$ and $y_2(0) = \beta$ in y_1, y_2 phase space, then the shape of the plot of $f(x, y_1, y_2 | \alpha, \beta) = 0$ is determined uniquely as a pseudo-elliptical type of orbit around the singularity at the center $(p_3/p_4, p_1/p_2)$.

This provides two ways to study the qualitative behavior of this system of equations.

1. **A vector field**
   A grid of points is selected and at each point an arrow is drawn to indicate the direction of dy_2/dy_1. Such a vector field is easy to create, and can be further enhanced by coloring the arrows to indicate direction, as well as altering the size of the stem of the arrows in proportion to the size of the derivatives.

2. **A collection of orbits**
   A set of initial conditions is chosen then the orbits are plotted and subsequently collected together to form a composite graph. This can be more tedious to do, but perhaps the result is easier to understand in terms of the way that the populations fluctuate depending on the initial conditions.

Vector field representation of the phase plane

The next graph is the default phase portrait diagram using colour to emphasize changes in direction while the following one demonstrates the option to increase arrow length in proportion to the absolute size of the derivative.

Lotka-Volterra Phase Portrait
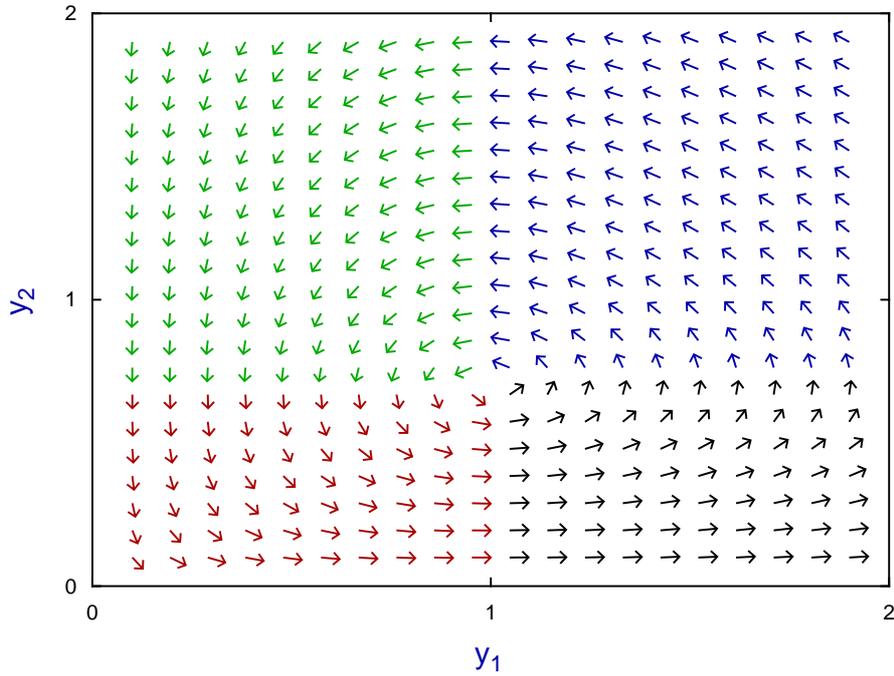
Arrow Length Proportional To Size
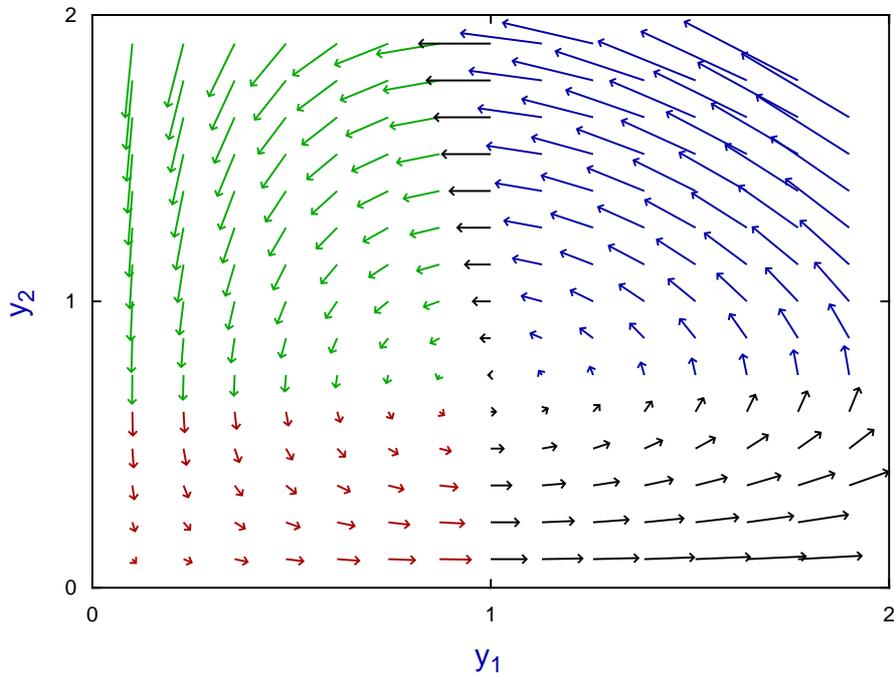
Note that the base of the arrow is at coordinates y_1, y_2 but, because the tips of the arrows will exceed the range set by these coordinates it is easiest to use ranges that will allow for sensible labels for the axes. In this case the range selected for the portrait was $0.1 \leq y_1 \leq 1.9$ and $0.1 \leq y_2 \leq 1.9$ resulting in a plot with integer labels. Also the density of arrows must be considered when plotting arrows with sizes proportional to the length of the derivative vectors as well as the proportionality factor which, in this case, was 0.1.

Plotting orbits

Once a model has been selected and parameters defined then clearly the trajectory will depend on the initial conditions. In the case of the Lotka-Volterra equations there is a point of stable equilibrium, and the equations are autonomous so the orbits will surround this critical point as shown in the next plot.

Lotka-Volterra Orbit
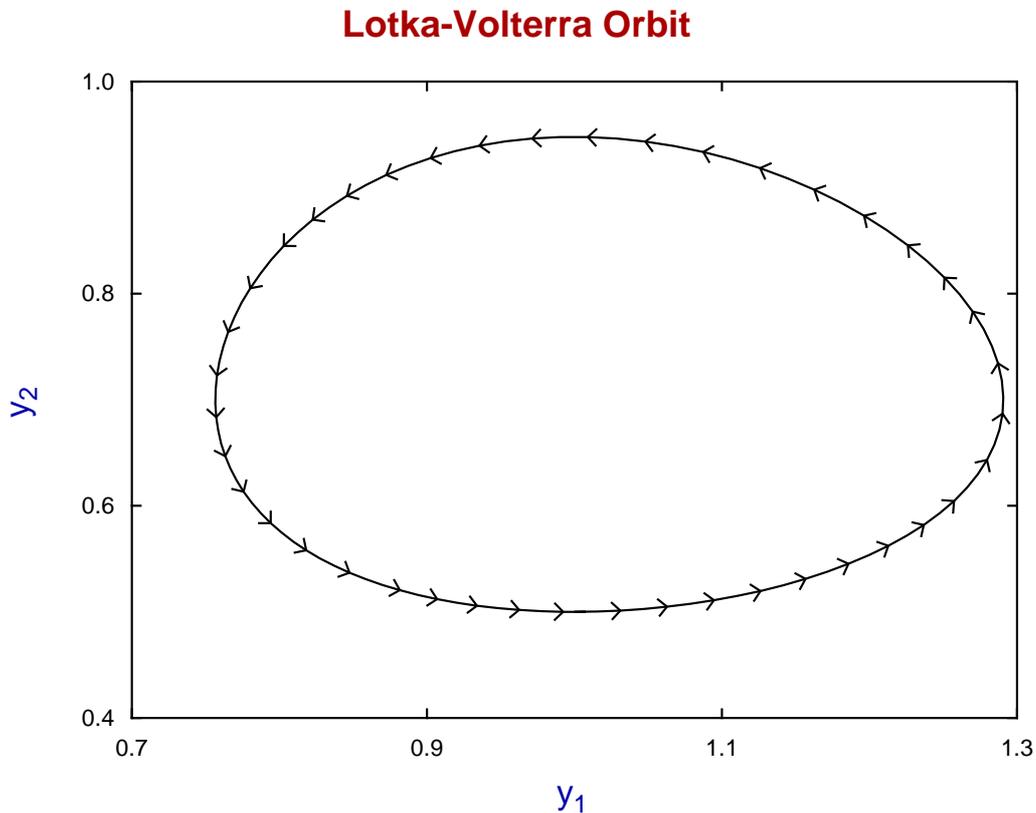
In order to compose a graph with several orbits as an alternative to the simple vector-field phase portrait the following actions are required.

1. Fix the parameter values that do not represent initial conditions. In this case these would be p_1, p_2, p_3, and p_4.

2. Alter the initial conditions, that is parameters p_5 and p_6.

3. When a satisfactory orbit has been constructed where a complete circuit of the equilibrium point has been completed the orbit can be archived.

4. These orbit files will be `f$orbits.001, f$orbits.002`, etc. created in the folder `\ProgramData\Simfit\user\res`

5. These can be retrieved as required to create a composite graph.

To illustrate this a set of orbits was created by maintaining $y_1(0) = 1.0$ then varying $y_2(0)$ over the range 0.25, 0.5, 1.0, 1.25 archiving the orbits each time then using the facility to import stored orbits. The resulting collection of archived orbits is displayed in the next figure.

Lotka-Volterra Collected Orbits
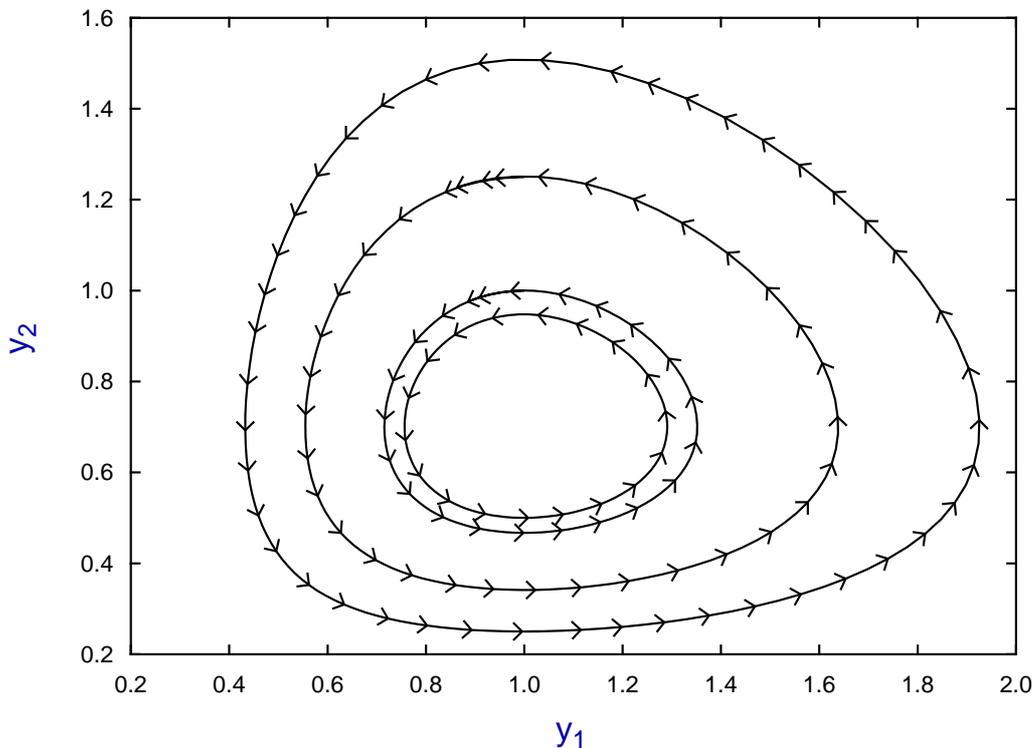
Curve fitting

In order to fit data sets using systems of differential equations the following steps are required.

1. Data sets must be prepared in one of two formats, that is
   Column 1 = $x(i)$, Column 2 = $y_j(i)$, or
   Column 1 = $x(i)$, Column 2 = $y_j(i)$, Column 3 = $s_j(i), i = 1, 2, \ldots, n_j$ for $j = 1, 2$

2. Each data set must have $x(i)$ in nondecreasing order for $i = 1, 2, \ldots, n_j$ and standard errors $s_j(i)$ must be accurate estimates if weighting is to be used.

3. Ideally the collected data file should be referenced by a library file if possible where it is understood that if data sets are missing for a component this will be indicated by a percentage sign %.

4. Starting estimates and limits should be appended to the first data file.

Here, for example is the default library file `deqsol.tfl` for the SIMF_IT program **deqsol**.

```
deqsol (library file) to fit Lotka-Volterra eqns. (2-eqns/menu item 2)
lv1.tf1
lv2.tf1
```

Note that, in general, the two filenames supplied in such a library file must be fully qualified filenames. This exception is because SIMF_IT recognizes the two files `lv1.tf1` and `lv2.tf1` as default test files.

The next two graphs illustrate first overlaying the starting curve on the data, and then the best-fit curves resulting from fitting the system of differential equations, followed by the parameter estimates, then goodness of fit analysis.

4

Lotka-Volterra Model Before Fitting
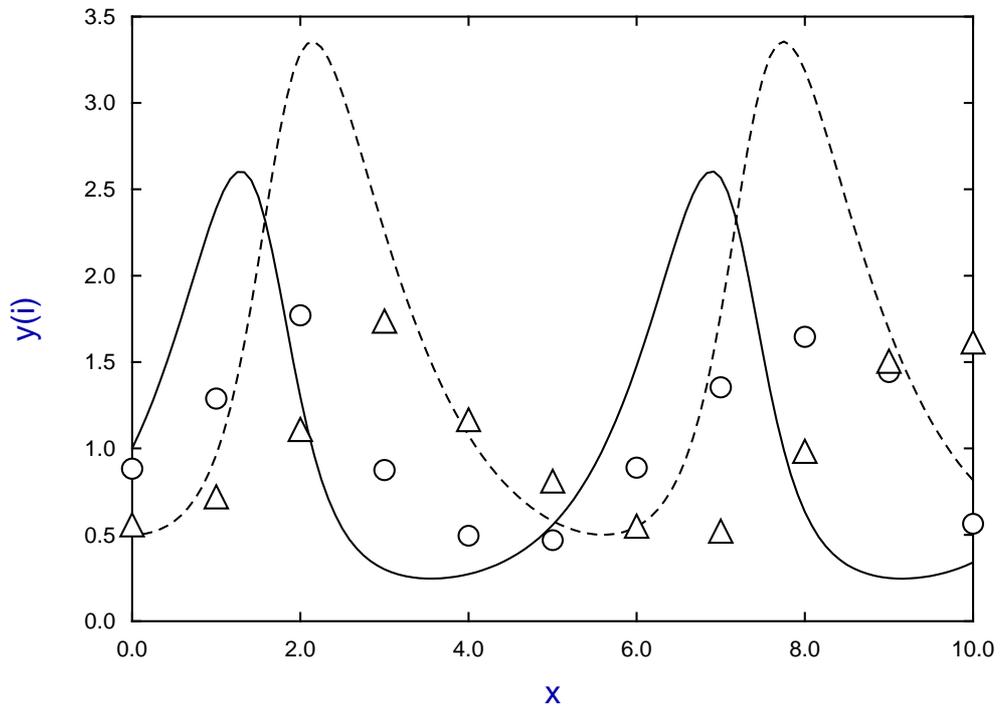
Lotka-Volterra Model After Fitting
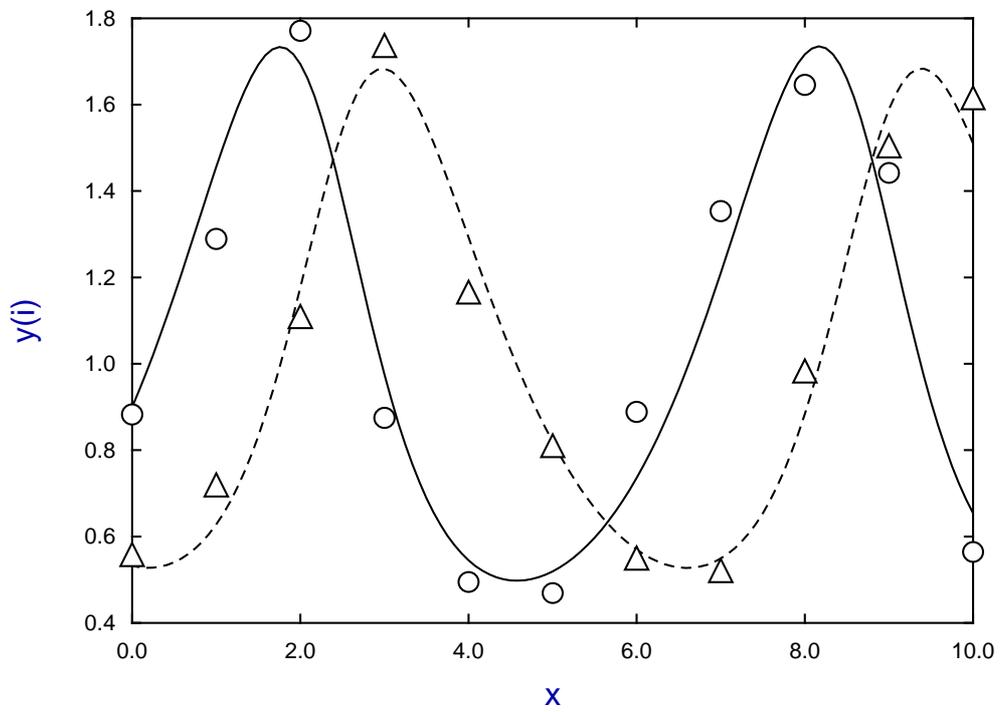
Best-fit parameters for Lotka-Volterra model

Number	Low-Limit	High-Limit	Parameter	Std. Error	Lower95%cl	Upper95%cl	p
1	0.0	2.0	1.08447	0.129892	0.809116	1.35983	0.0000
2	0.0	2.0	1.08901	0.123310	0.827608	1.35042	0.0000
3	0.0	2.0	0.93975	0.115146	0.695649	1.18384	0.0000
4	0.0	2.0	0.94854	0.108451	0.718635	1.17845	0.0000
5	0.0	2.0	0.89979	0.055012	0.783168	1.01641	0.0000
6	0.0	2.0	0.53189	0.043215	0.440273	0.62350	0.0000

Analysis of residuals: SSQ	0.18589	
Average % coefficient of variation	10.33%	
R^2, i.e. correlation coefficient$(theory, data)^2$	0.9547	
Largest Absolute relative residual	18.81%	
Smallest Absolute relative residual	1.10%	
Average Absolute relative residual	8.11%	
Absolute relative residual in range 0.1-0.2	40.91%	
Absolute relative residual in range 0.2-0.4	0.00%	
Absolute relative residual in range 0.4-0.8	0.00%	
Absolute relative residual > 0.8	0.00%	
Number of negative residuals (m)	13	
Number of positive residuals (n)	9	
Number of runs observed (r)	14	
$P(\text{runs} \leq r : \text{given } m \text{ and } n)$	0.9031	
5% lower tail point	7	
1% lower tail point	6	
$P(\text{runs} \leq r : \text{given } m \text{ plus } n)$	0.9054	
$P(\text{signs} \leq \text{least number observed})$	0.5235	
Durbin-Watson test statistic	2.6043	>2.5, -ve serial correlation?
Shapiro-Wilks W statistic	0.9515	
Significance level of W	0.3385	
Akaike AIC statistic	-93.020	
Schwarz SC statistic	-86.474	
Verdict on goodness of fit	very good	

User-defined models

Model files have three mandatory sections separated by the escape character % followed an optional section which is very important when constructing models for differential equations, summarized as follows.

%

Title section

Arbitrary text used to describe the equations but must not exceed 24 lines

%

2) Summary section

The number of equations, type of equations, and number of parameters

%

3) Equation section

The equations and Jacobian if required

%

4) Optional additional information

The SimFIT default test file deqmod2_e.tf2 for the Lotka-Volterra scheme is displayed next.

```
%
Example of a user supplied pair of differential equations
file: deqmod2_e.tf2 (with appended parameter limits data)
model: Lotka-Volterra predator-prey equations
differential equations: f(1) = dy(1)/dx
                             = p(1)*y(1) - p(2)*y(1)*y(2)
                        f(2) = dy(2)/dx
                             = -p(3)*y(2) + p(4)*y(1)*y(2)
              jacobian: j(1) = df(1)/dy(1)
                             = p(1) - p(2)*y(2)
                        j(2) = df(2)/dy(1)
                             = p(4)*y(2)
                        j(3) = df(1)/dy(2)
                             = -p(2)*y(1)
                        j(4) = df(2)/dy(2)
                             = -p(3) + p(4)*y(1)
     initial condition: y0(1) = p(5), y0(2) = p(6)
 Note: the last parameters must be y0(i) in differential equations
%
 2 equations
 differential equation
 6 parameters
%
begin{expression}
f(1) = p(1)y(1) - p(2)y(1)y(2)
f(2) = -p(3)y(2) + p(4)y(1)y(2)
end{expression}
%
begin{expression}
j(1) = p(1) - p(2)y(2)
j(2) = p(4)y(2)
j(3) = -p(2)y(1)
j(4) = -p(3) + p(4)y(1)
end{expression}
%
 The coding for the model is now finished but, optionally, parameter starting
 values, curve fitting limits and the range of integration can be appended.
 If these are not supplied program DEQSOL will request an initialisation file.
 ==============================================================================

 begin{limits}
 0  1.0  3
 0  1.0  3
 0  1.0  3
 0  1.0  3
 0  1.0  3
 0  0.5  3
 end{limits}

 begin{range}
 121
 0
 10
 end{range}
```