The standard SIR epidemic differential equation model considers the interaction between susceptible, infected, and resistant individuals as a development of a Lotka-Volterra type of interaction. This system of equations can be extended in many ways to include censoring by birth, death, immigration, or emigration as well as including additional covariates and other complicating factors such as seasonal effects and vaccination.

For instance, a simple extension to include incomplete resistance after infection can be included by similar mass action reasoning but, in order to preserve a constant conservation equation, the simple addition of a third parameter $p_3$ can model births adding to the susceptible sub–population and loss by death from the resistant sub–population according to the following scheme.

For $y_1(x)$ susceptible, $y_2(x)$ infected, and $y_3(x)$ resistant individuals in the population as functions of time $x$ we would then have

$$f_1 = \frac{dy_1}{dx} = -p_1 y_1 y_2 + p_3$$
$$f_2 = \frac{dy_2}{dx} = p_1 y_1 y_2 - p_2 y_2$$
$$f_3 = \frac{dy_3}{dx} = p_2 y_2 - p_3$$

where $p_1$, $p_2$ and $p_3$ are positive parameters, and the additional parameters $p_4 = y_1(0), p_5 = y_2(0)$ and $p_6 = y_3(0)$ are the initial conditions. Here the number of susceptible individuals declines as a result of contact between themselves and those already infected, the number of infected individuals increases as a result of such contacts but declines resulting from the development of resistance, while resistance increases in proportion to the number of those infected. Note that the overall population $Y(x) = y_1(x) + y_2(x) + y_3(x)$ remains static as will be obvious from the derivative of the conservation equation

$$\frac{dY}{dx} = \frac{dy_1}{dx} + \frac{dy_2}{dx} + \frac{dy_2}{dx}$$
$$= 0$$

while the Jacobian matrix required for stiff systems is defined as follows.

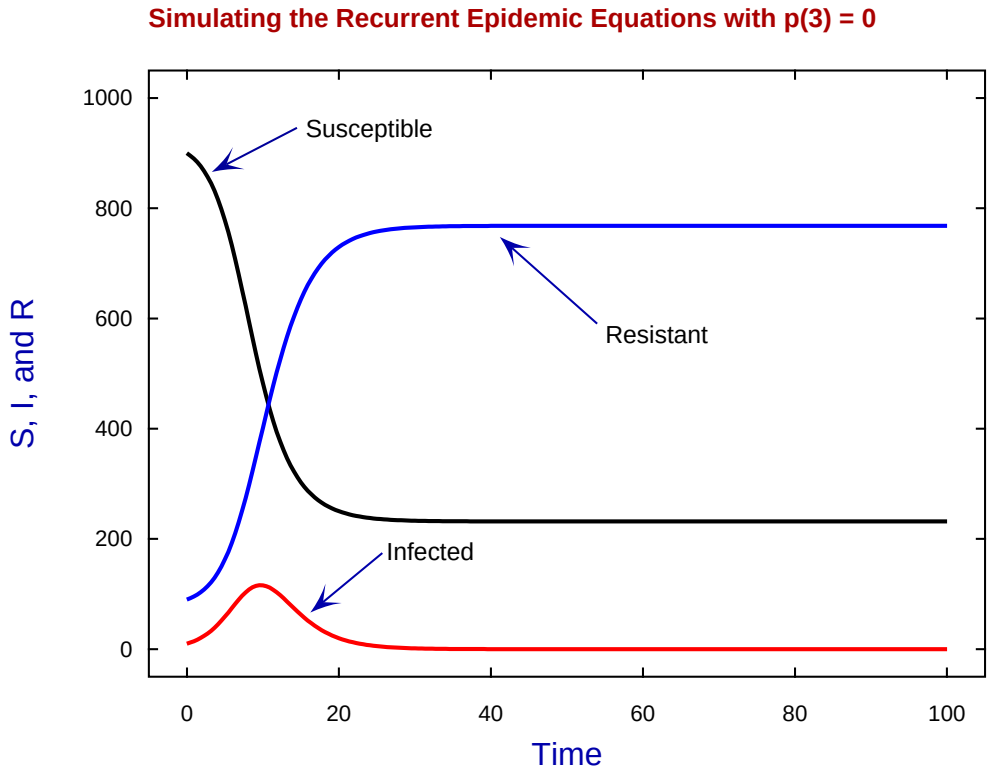
$$\frac{\partial f_i}{\partial y_j} = \begin{pmatrix} -p_1 y_2 & -p_1 y_1 & 0 \\ p_1 y_2 & p_1 y_1 - p_2 & 0 \\ 0 & p_2 & 0 \end{pmatrix}$$

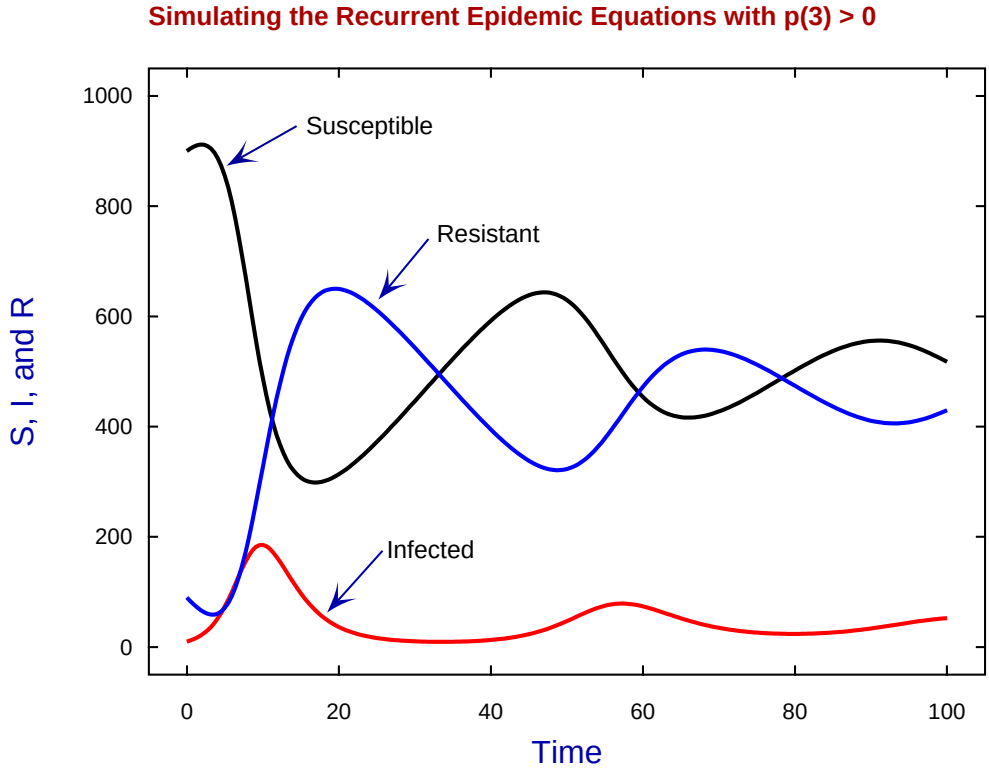
When $p_3 > 0$ it can be shown that if $p_1 p_3 / p_2^2 < 4$ there is light damping to an equilibrium points with the system converging as follows

$$y_1 \approx \frac{p_2}{p_1}$$
$$y_2 \approx \frac{p_3}{p_2}$$

at large time values, i.e., the endemic state of partial herd immunity.

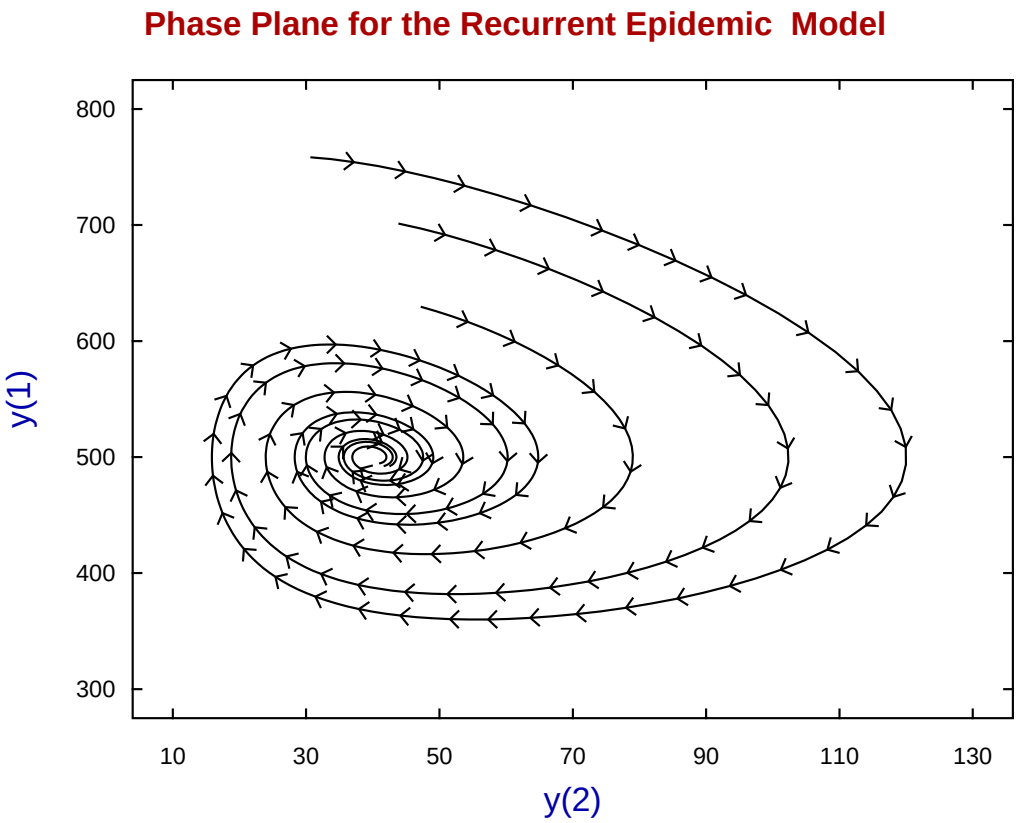
The next two figures illustrate the non–oscillating behaviour when $p_3 = 0$ compared to the damped oscillations when $p_3 > 0$ and so establishing this set of equations being a model for a recurrent epidemic, i.e. where acquired resistance does not preclude subsequent re–infection.




These figures were generated using SIMFIT program **deqsol** with the built–in defaults and initial conditions for $p(i)$ where $i = 1, 6$.

A convenient way to display this oscillatory property of the recurrent epidemic model is to plot the phase plane as shown next as it is more convincing when visualized such a plot than staring at mere algebra.



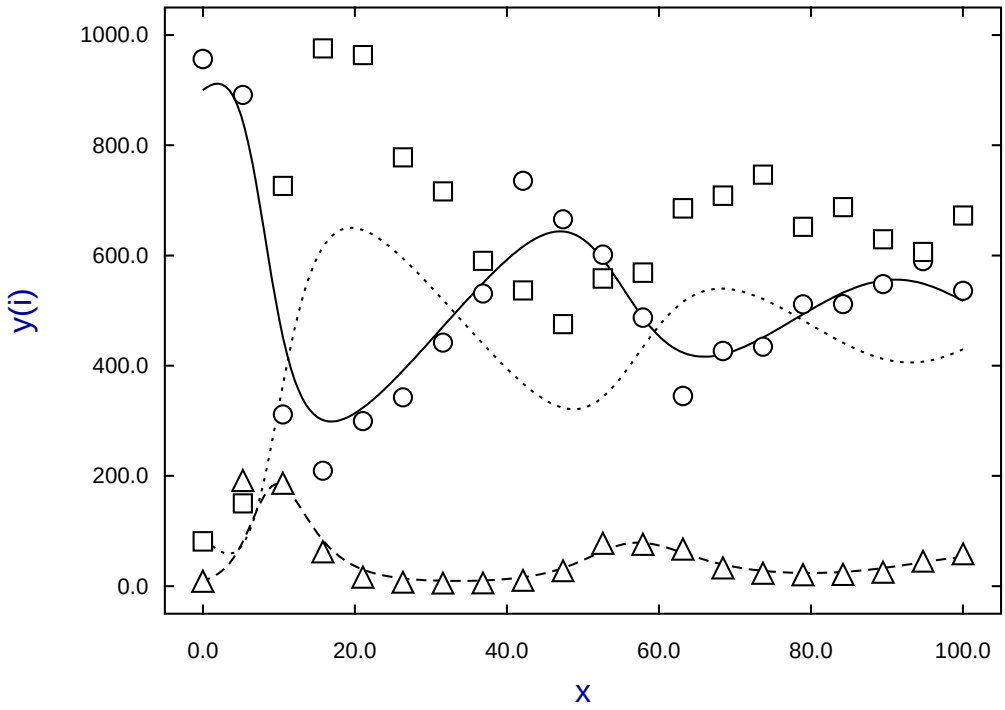

Such phase plane diagrams are easily constructed using the SimFIT program **deqsol** using the following steps.

1. Fix parameters $p_i, i = 1, 6$ and also the starting and ending points for the integration and number of points required then integrate.
2. Plot the orbit then choose the option to store the orbit.
3. Repeat the process keeping the parameters $p_i, i = 1, 3$ fixed but varying the initial conditions $p_i, i = 4, 6$ then storing the orbits for each choice of initial conditions.
4. Choose the option to plot the archived orbits that will have been temporarily stored then select the archived orbits required to form a composite phase plane diagram like the above figure.
5. Proceed to sculpture the graph then save

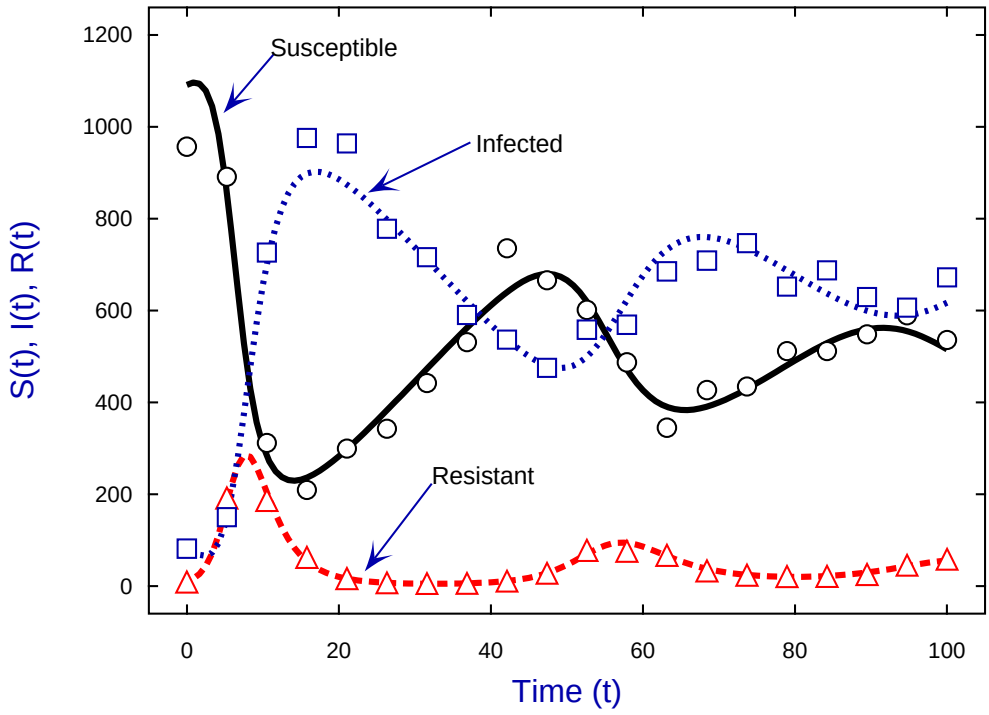
Now we consider fitting the recurrent epidemic data. Here the next two figures illustrate that before fitting was attempted the starting estimates gave profiles far away from the data. However, after constrained nonlinear optimization, a good fit was eventually located.

Actually, after several abortive attempts from numerous starting estimates for the six parameters made no progress, the curve fitting had to make extensive use of the SimFIT procedures to vary the starting estimates randomly over a range specified by the upper and lower parameter limits until eventually convergence was achieved. It must be emphasized that this use of several random starts using a uniform or normal distribution to sequentially perturb the starting estimates within the limits of constraint is frequently required when the final solution is far removed from the starting estimates and the process of optimization is stalled so that the WSSQ does not change appreciably until the perturbation generates a sensible descent vector.

## Recurrent Epidemic Data before Fitting




## Fitting The Recurrent Epidemic Differential Equations To Data



The next table lists the best fit parameters and confidence limits and it will be clear from the $p$ values that the parameters are well determined.

| Number | Parameter | Std.Error | Lower95%cl | Upper95%cl | p |
|---|---|---|---|---|---|
| 1 | 0.00101 | 0.00012 | 0.00099 | 0.00103 | 0.0000 |
| 2 | 0.49581 | 0.00615 | 0.48348 | 0.50815 | 0.0000 |
| 3 | 19.7314 | 0.17848 | 19.3736 | 20.0893 | 0.0000 |
| 4 | 1089.28 | 6.91486 | 1075.42 | 1103.15 | 0.0000 |
| 5 | 9.85414 | 0.19370 | 9.46580 | 10.2425 | 0.0000 |
| 6 | 90.7113 | 1.08120 | 88.5436 | 92.8790 | 0.0000 |

Some technical details follow giving more information to assist users who want to develop their own equations for simulating and fitting.

1. To state the obvious: the model to be fitted must be appropriate, the data must be extensive and reasonably accurate, the starting estimates and parameter limits must be sensible, and attention must be paid to any weighting that may be required. The copious goodness of fit tables, residuals analysis, and advice output by SIMFIT must be read and appreciated because simulating and fitting nonlinear models is extremely difficult. Usually several runs using randomised starting estimates will be required.

2. SIMFIT uses the Open Source programs DVODE to simulate the differential equations using the BDF method by default and the constrained nonlinear optimisation quasi Newton routine LBFGSB. These are bundled as part of the SIMFIT package which uses a built–in reverse communication procedure during the optimisation in an attempt to maintain the parameters of order unity at the start each iteration.

3. For those who have a license to access the numerical algorithms group routines (NAG), SIMFIT has an interface to the NAG library that can be used instead of the built in SIMFIT routines. This is extremely valuable as being able to switch at will between different simulation and particularly optimisation codes can often improve the success of data fitting.

4. The SIMFIT built–in interface to the NAG library supports the following tried and tested routines.

   D02CJF, D02EJF for solving systems of nonlinear differential equations and E04KZF, E04YJF, E04UEF, E04UFF for constrained nonlinear optimisation

   However there are documents available on the website showing how SIMFIT can be programmed to call any NAG library routine.

5. It must be emphasised that to simulate and fit differential equations users do need to configure the methods used by specifying values to be used to to control step length and convergence criteria.

6. The SIMFIT test library file `recurrent.TFL` contains the data described in this document and the model required to simulate and fit the recurrent epidemic is built into program **deqsol**. However it should be pointed out that if a method to integrate stiff systems is required instead of the Runge–Kutta or Adams methods a user–defined Jacobian can be supplied if possible or a Jacobian can be estimated. Note that an incorrect explicit user–supplied Jacobean is much worse than one estimated.

7. SIMFIT users can write their own equations for simulation and fitting by just using a text editor. This involves a method whereby the equations can be written using standard mathematical expressions because there is a built–in routine to transform them into reverse Polish. To facilitate the models being used in iterative procedures SIMFIT scans the ASCII text file just once then creates a temporary internal stack so the program does not have to re–read the model file again.

8. The file `deqmod3_e.tf3` used to simulate and fit the recurrent epidemic equations as described in this document is listed next.

```
%
Example of a user supplied set of 3 differential equations
file: deqmod3_e.tf1
model: coupled equations for a recurrent epidemic
differential equations: f(1) = dy(1)/dx = -p(1)y(1)y(2) + p(3)
                        f(2) = dy(2)/dx =  p(1)y(1)y(2) - p(2)y(2)
                        f(3) = dy(3)/dx =  p(2)y(2) - p(3)
y(1) = Susceptible, y(2) = Infected, y(3) = Resistant
 jacobian: j(1) = df(1)/dy(1) = -p(1)y(2)
           j(2) = df(2)/dy(1) =  p(1)y(2)
           j(3) = df(3)/dy(1) =  0
           j(4) = df(1)/dy(2) = -p(1)y(1)
           j(5) = df(2)/dy(2) =  p(1)y(1) - p(2)
           j(6) = df(3)/dy(2) =  p(2)
           j(7) = df(1)/dy(3) =  0
           j(8) = df(2)/dy(3) =  0
           j(9) = df(3)/dy(3) =  0
   initial condition: y0(1) = p(3), y0(2) = p(4), y0(3) = p(5)
Note: the last parameters must be y0(i) in differential equations
%
3 equations
differential equation
6 parameters
%
begin{expression}
A = p(1)y(1)y(2)
B = p(2)y(2)
f(1) = -A + p(3)
f(2) = A - B
f(3) = B - p(3)
end{expression}
%
begin{expression}
C = p(1)y(2)
D = p(1)y(1)
j(1) = -C
j(2) = C
j(3) = 0
j(4) = -D
j(5) = D - p(2)
j(6) = p(2)
j(7) = 0
j(8) = 0
j(9) = 0
end{expression}
%
begin{limits}
  0.0      0.001     1.0
  0.0      0.5       2.0
  0.0     30.0     100.0
400.0    900.0    1400.0
  0.0     10.0      50.0
  0.0     90.0     150.0
end{limits}
begin{range}
121
0
200
end{range}
```