Predator - Prey Equations

The Lotka-Volterra Altera predator prey equations are the granddaddy of all models involvement competition between species.

$$y_1' = y_1(1 - \frac{y_2}{\mu_2})$$
$$y_2' = -y_2(1 - \frac{y_1}{\mu_1})$$

They are the foundation of fields like mathematical ecology. Think of the two species as rabbits and foxes or moose an wolves or little fish in big fish. y_1 represents the prey who would live peacefully by themselves if there were no predators. I have chosen the units of time and population so that the coefficients in front of the leading linear terms are one.

$$y_1' = y_1$$
$$y_2' = -y_2$$

But the nonlinear terms are like logistics terms, except with the interaction between the two species. The growth of y_1 is limited by the presence of y_2. So y_1 will grow until the y_2/μ_2 becomes 0. On the other hand, the decay of y_2 becomes 0 when y_1 reaches μ_1.

To complete this specification we need the initial conditions. So we have two values η_1, η_2 which are the initial values of y_1, y_2. Don't worry about the fact that these are continuous variables and that we can have non-integer numbers of individuals. We can have half of a rabbit or a tenth of a moose. These are, after all, models that are idealized versions of what's happening in nature.

The critical point are the derivatives become 0. there is a critical point at the origin. But the interesting one is when these terms become 0.

$$(y_1, y_2) = (\mu_1, \mu_2)$$

We have to look at the Jacobian.

$$J = \begin{bmatrix} 1 - \frac{y_2}{\mu_2} & -\frac{y_1}{\mu_2} \\ \frac{y_2}{\mu_1} & -1 + \frac{y_1}{\mu_1} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\mu_1}{\mu_2} \\ \frac{\mu_2}{\mu_1} & 0 \end{bmatrix}$$

And the eigenvalues of the Jacobian are $\pm i$. And so this critical point is a stable centre with a period 2π.

So these are non-linear equations. We can't express the solution in terms of simple analytic functions. We have to compute them numerically. But we do know this about their behaviour. If the initial conditions are close to the critical point, the solution is periodic. The period is close to 2π. And the orbit is close to an ellipse. On the other hand, if the initial conditions are far from the critical point, then it turns out the solution is still periodic. But the period is greater than 2π and the orbit is far from an ellipse.

Let's bring up MATLAB and compute a solution. We need the parameters.

```
>> mu = [300 200]'
   eta = [400 100]'
```

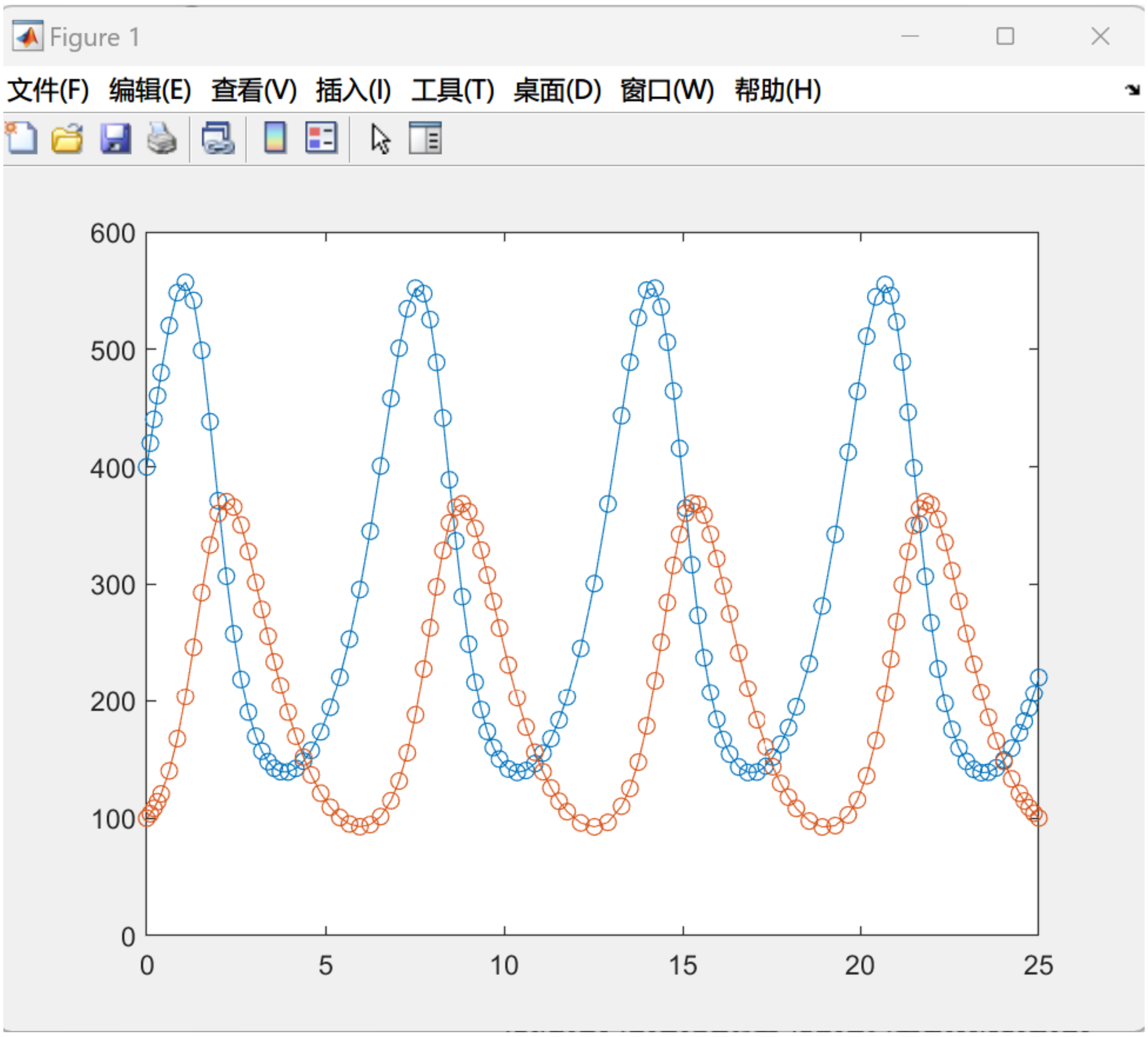
and the function

```
>> F = @(t,y) [(1-y(2)/mu(2))*y(1); -(1-y(1)/mu(1))*y(2)]
```

then choose ODE45 and we will integrate from 0 to 25.

```
ode45(F,[0,25],eta)
```

Here is the solution.



It's periodic. A predator and a prey. And it looks like the period returning back to the initial condition is 100 and 400. And it's returning back. And we have integrated over something more than three periods.

I happen to know that the period is 6.5.

```
>> period = 6.5357
```

And so I want to compute to higher accuracy 10^{-6}.

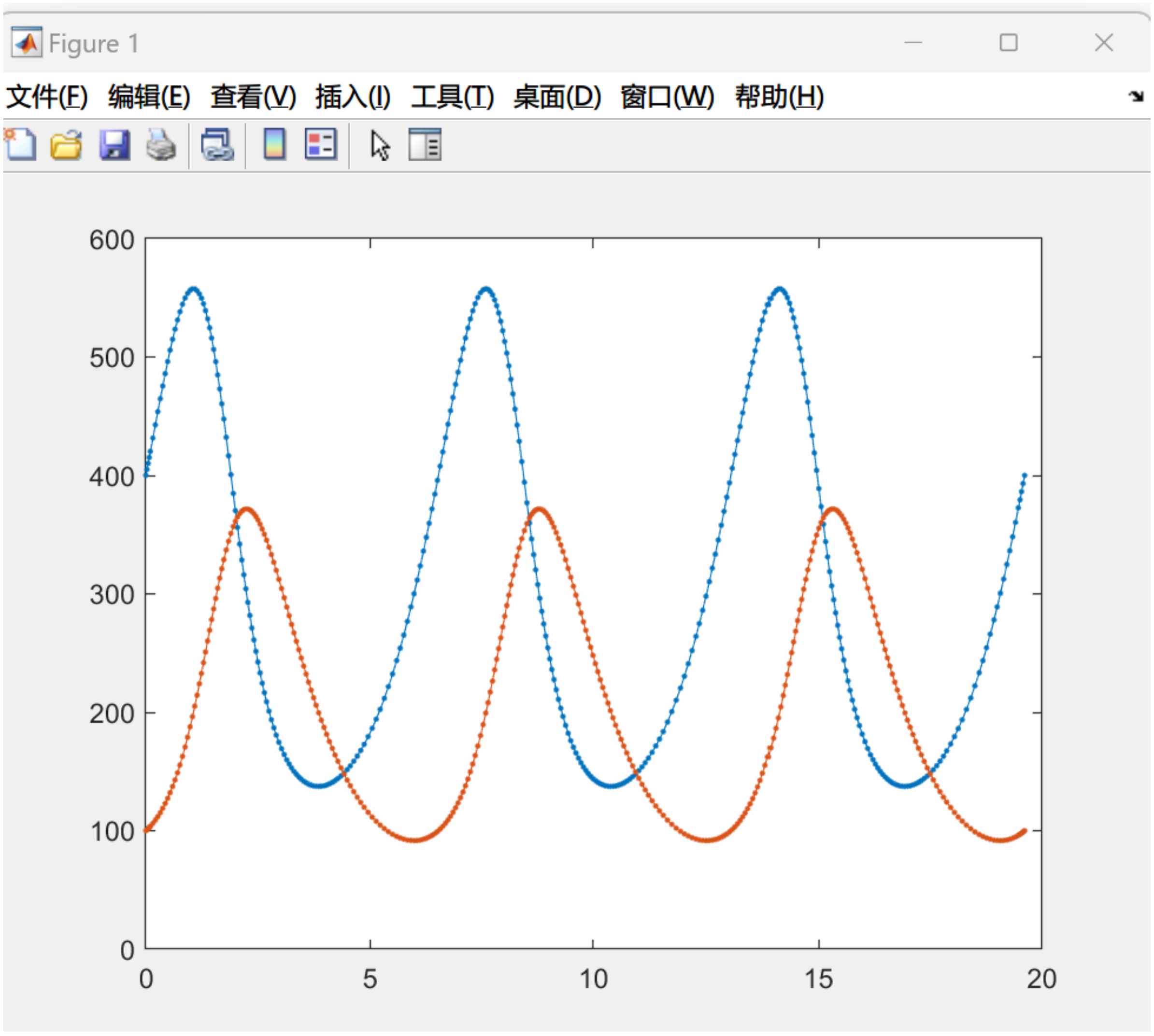
```
>>> opt = odeset('RelTol',1.e-6)
```

And let's capture the solution and integrate over three periods.

```
>> [t,y] = ode45(F,[0,3*period],eta,opt)
```

Let's plot it with the finer dots.

```
>> plot(t,y,'.-')
```



And we can see here, we have gone over three periods and return back to initial values of 100 and 300. And now we can use something that will show off the periodicity of function in MATLAB called Comet.

```
>> axis([100 600 50 400]), hold on, plot(eta(1),eta(2),'o'),
pause(1),comet(y(:,1),y(:,2))
```

Determining the period of a periodic solution is often the important part of a calculation. In the MATLAB ODE Suit, this is done with an event handler.

```
>> opt = odeset('RelTol',1.e-6,'event',@pitstop)
```