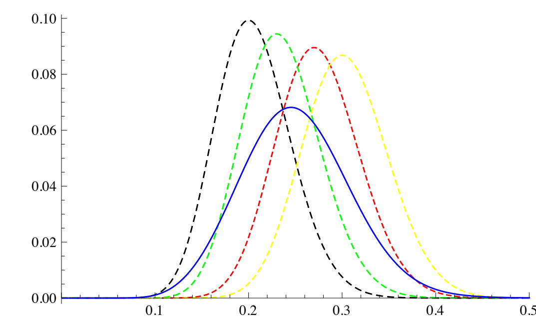
# Predator Prey modeling

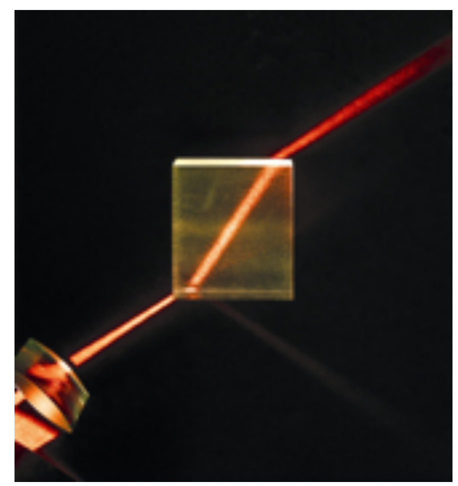
## Solving ordinary differential equations

# Projects

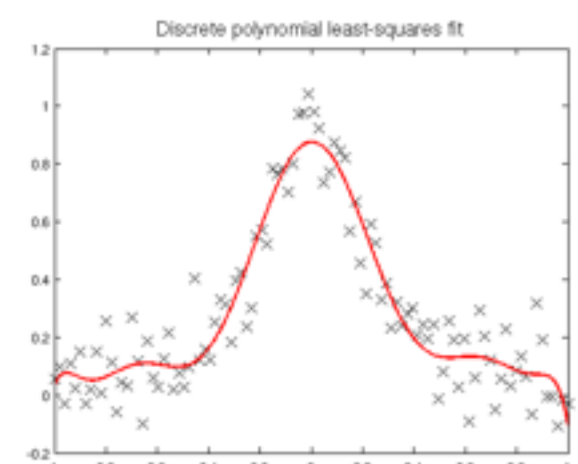
Are dice manipulated or not




Refraction

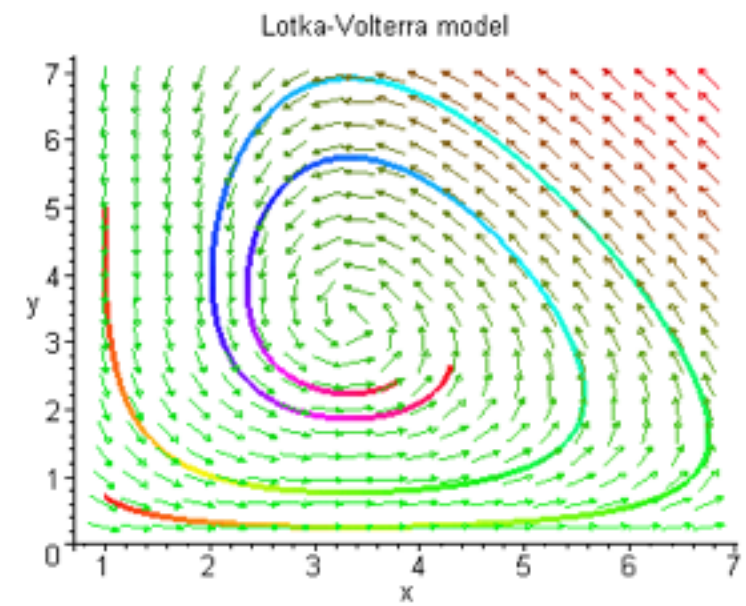


Least-Squares regression/interpolation



Showing trends

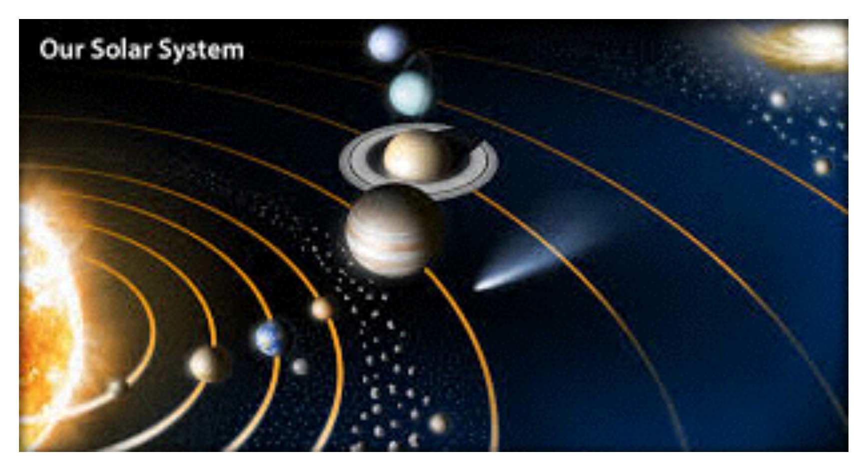

Analyze in detail Lotka-Volterra equation (predator-prey) system




Develop a file converter for some analysis

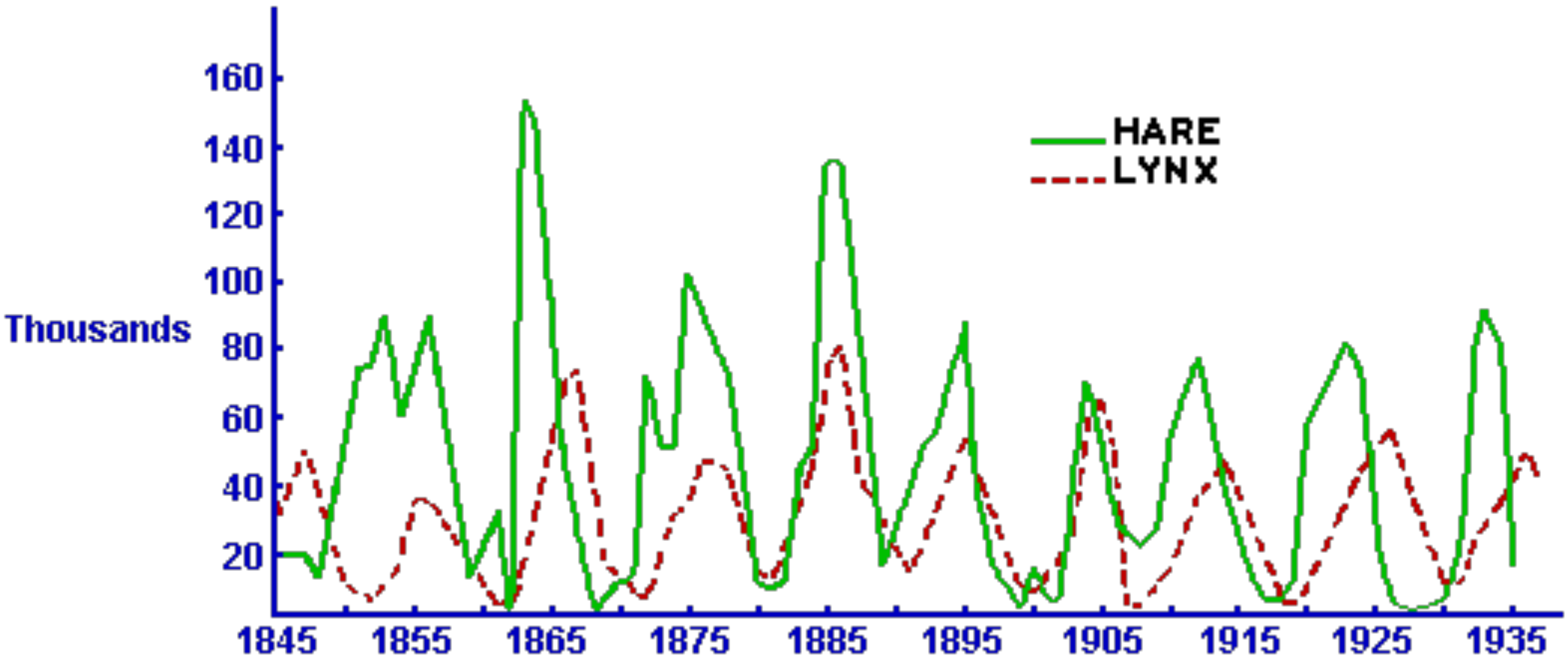
A → B → C →

Calculate the position of planets on day X



# Predator Prey modeling

## Solving ordinary differential equations


HARE
LYNX
Thousands
160
140
120
100
80
60
40
20
1845
1855
1865
1875
1885
1895
1905
1915
1925
1935

# Predator-Prey model

$$\frac{dr}{dt} = \alpha r - \beta r f$$
$$\frac{df}{dt} = \delta r f - \gamma f$$

$r$ = number of rabbits

$f$ = number of foxes

$\alpha$ = birth rate of rabbits

$\beta$ = death rate of rabbits

$\delta$ = birth rate of foxes

$\gamma$ = death rate of foxes

In a small time interval rabbits are born and rabbits die because of foxes (there is no other cause of death for rabbits. In the same time interval foxes are born, the more rabbits are available the more foxes can be fed and maintained, foxes die of unknown causes.

$$\frac{dr}{dt} = \alpha r - \beta r f$$

$$\frac{df}{dt} = \delta r f - \gamma f$$

$$\frac{dr}{dt} = \alpha r - \beta r f$$

$$\frac{df}{dt} = \delta r f - \gamma f$$

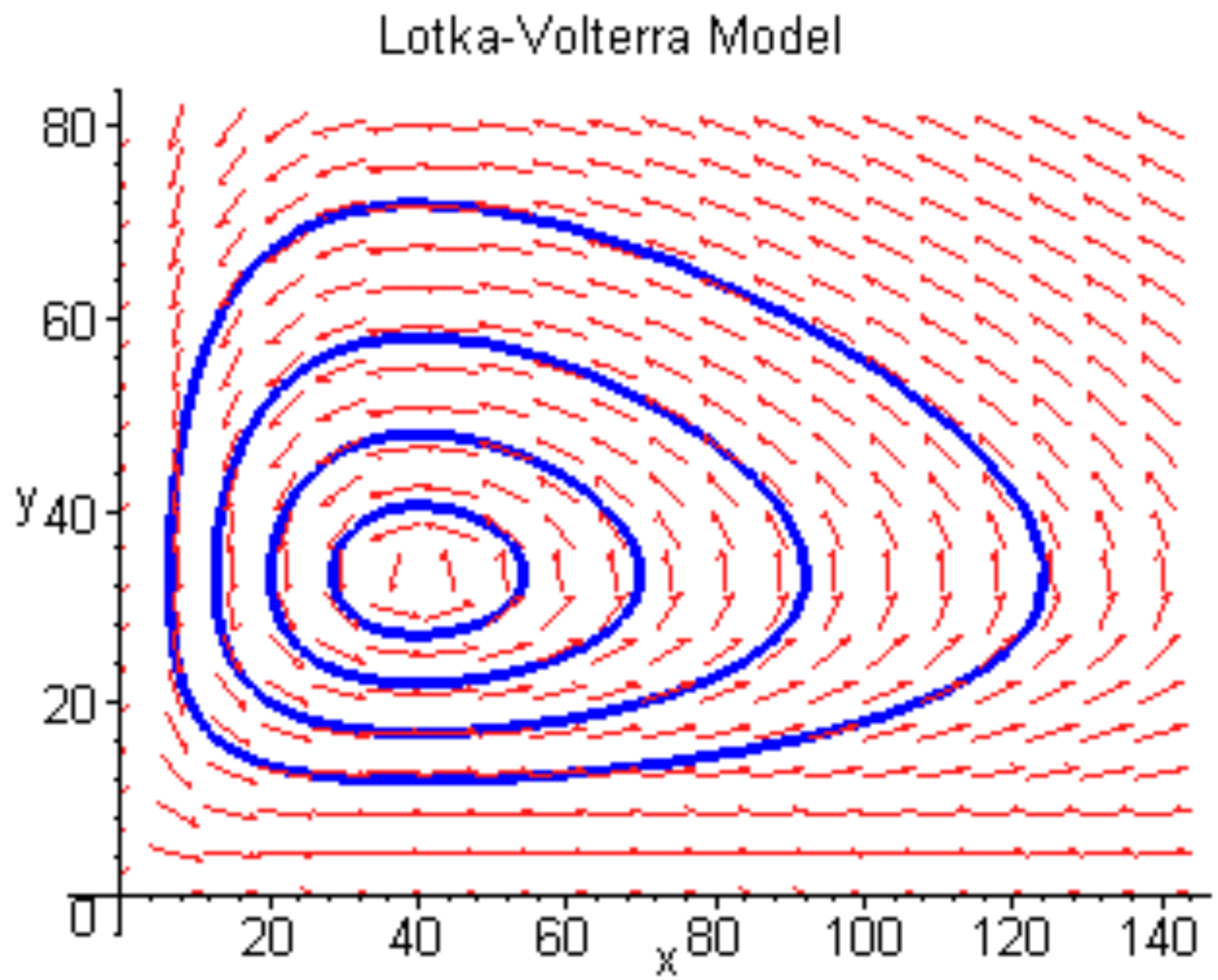
$$0 = \alpha r = \beta r f$$

$$0 = \delta r f - \gamma f$$

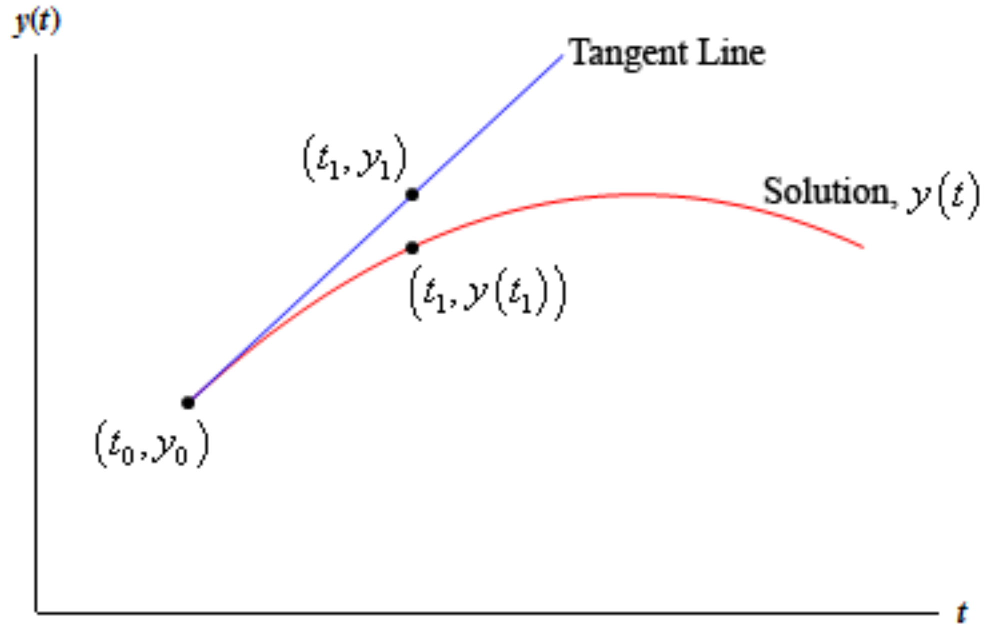
$$\frac{dr}{dt} = \alpha r - \beta r f$$
$$\frac{df}{dt} = \delta r f - \gamma f$$

$$0 = \alpha r + \beta r f$$
$$0 = \delta r f - \gamma f$$

$$r = \frac{\gamma}{\delta}$$
$$f = \frac{\alpha}{\beta}$$

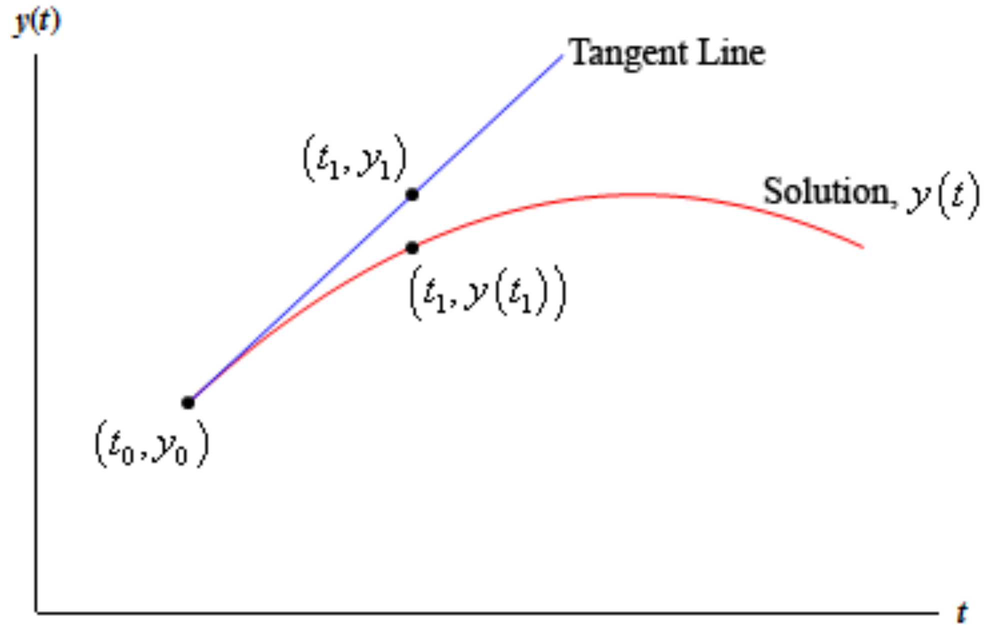


# Numerical Solution of an ODE




$$y = y_0 + f(t_0, y_0)(t - t_0)$$

# Euler method




$$y_1 = y_0 + y'(t_0)\Delta$$

# Euler method to solve for Lotka Voltera

$$x_1 = x_0 + x'(t_0)\Delta$$
$$y_1 = y_0 + y'(t_0)\Delta$$

# Euler method to solve for Lotka Voltera

$$x_1 = x_0 + x'(t_0)\Delta$$
$$y_1 = y_0 + y'(t_0)\Delta$$

```
// rabbits gets eaten at rate beta and reproduce at rate alpha
//
double prey(double alpha, double beta, double r, double f)
{
  return alpha*r - beta*r*f;
}

// foxes eat rabbits and produce more foxes at rate delta
// and die at rate alpha
double predator(double delta, double gamma, double r, double f)
{
  return delta * r * f - gamma * f;
}


//
// xk = xk-1 + (axk-1 - bxk-1yk-1) Delta-t,
// yk = yk-1 + (-cyk-1 + pxk-1yk-1) Delta-t.
//
void euler(double & r0, double & f0, double deltat, vector<double> parameters)
{
  double alpha = parameters[0];
  double beta = parameters[1];
  double delta = parameters[2];
  double gamma = parameters[3];
  double r;
  double f;
  r = prey(alpha, beta, r0, f0) * deltat;
  f = predator(delta, gamma, r0, f0) * deltat;
  r0 += r;
  f0 += f;
}
```