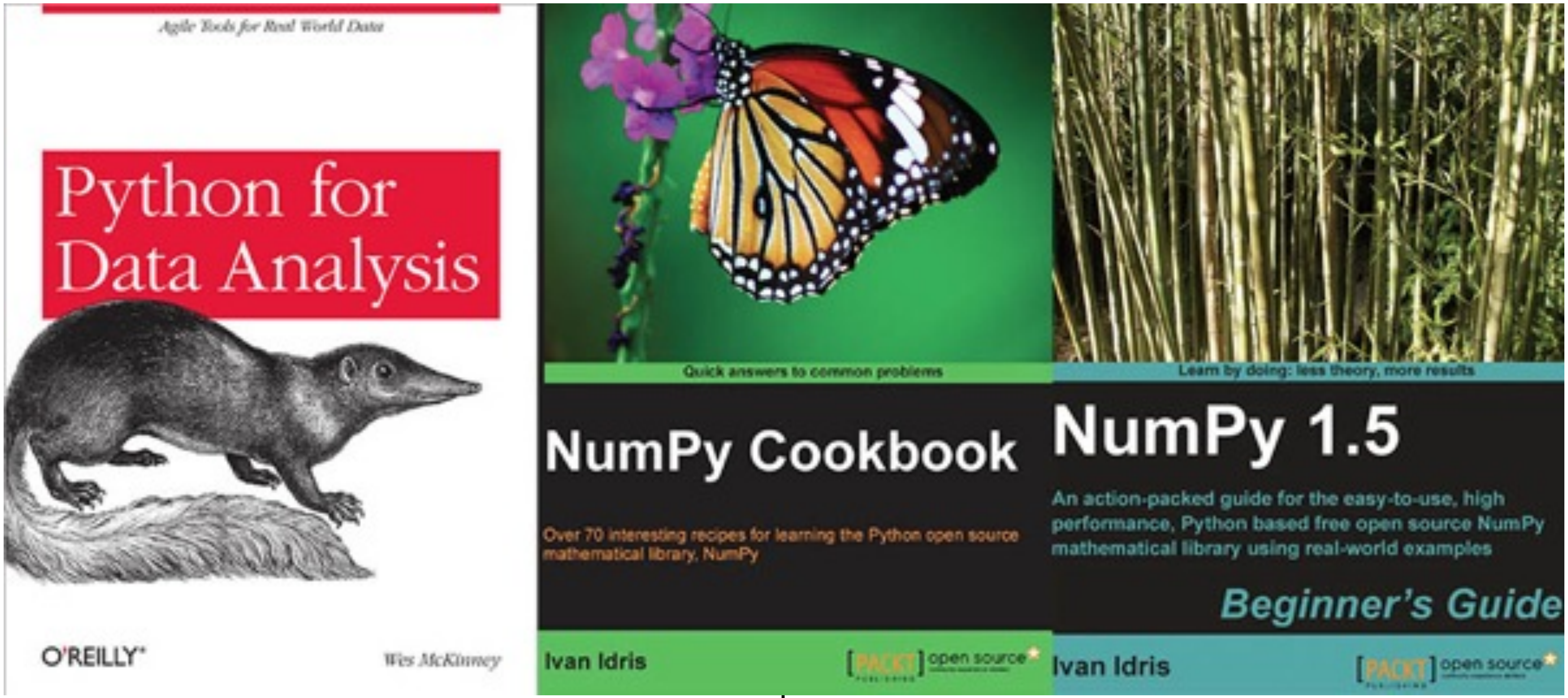
Python

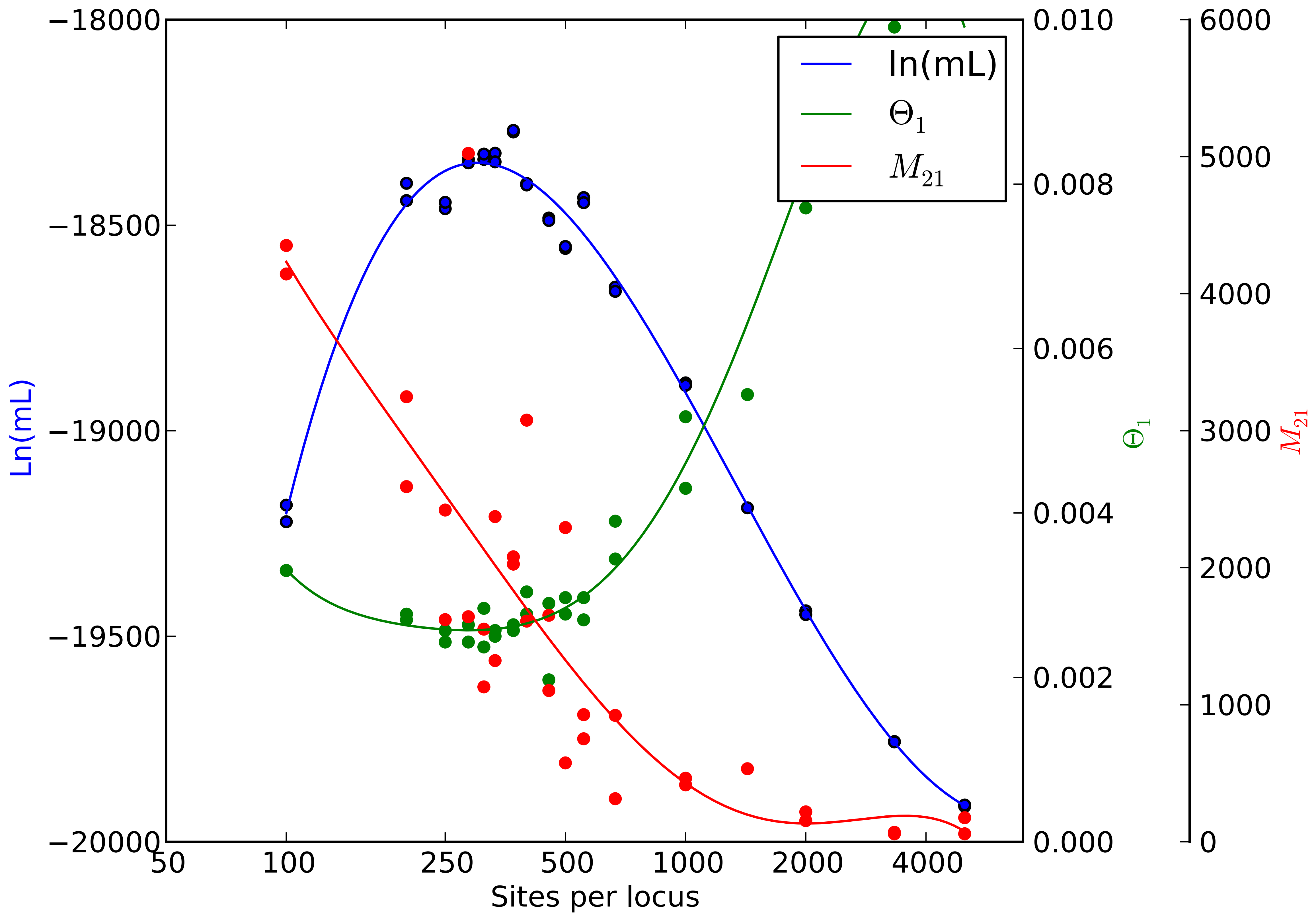
- scientific computing (scipy and numpy)
- interaction with the environment

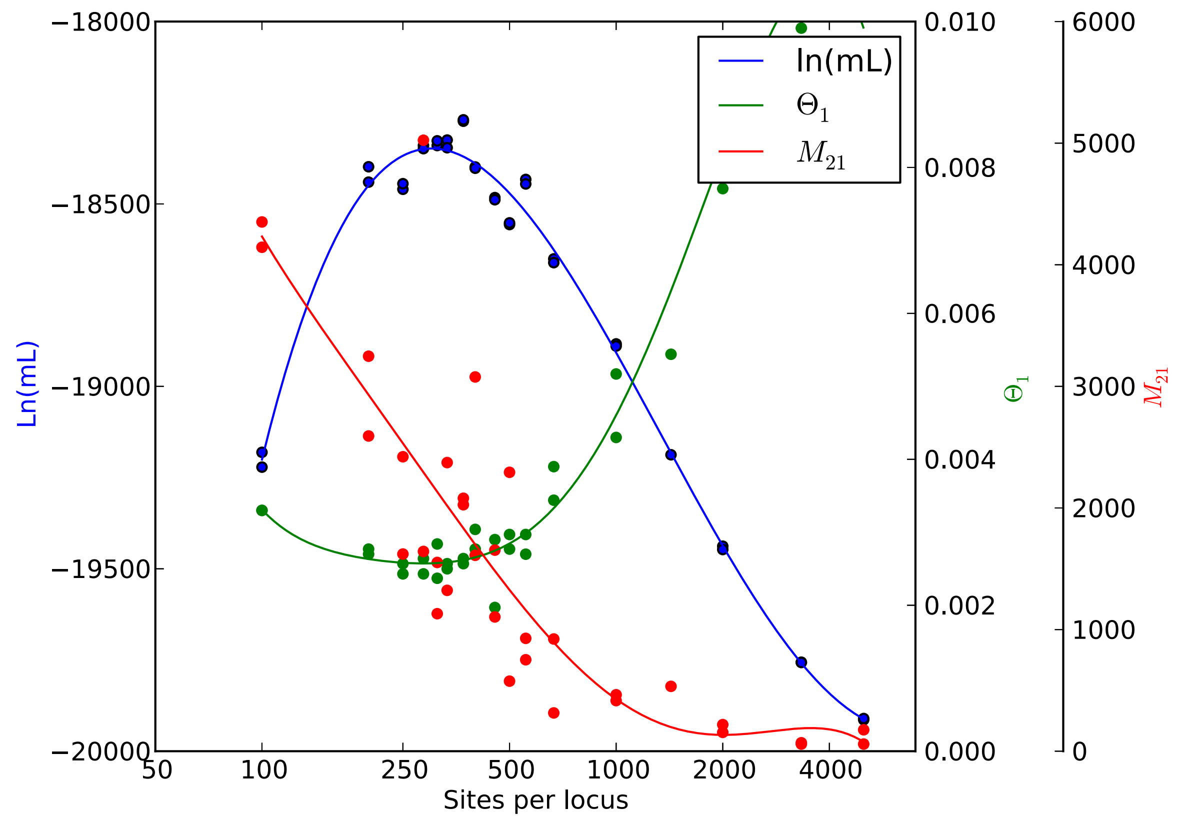

Agile Tools for Real World Data
Python for Data Analysis
O'REILLY
Wes McKinney
Quick answers to common problems
NumPy Cookbook
Over 70 interesting recipes for learning the Python open source mathematical library, NumPy
Ivan Idris
[PACKT] open source
Learn by doing: less theory, more results
NumPy 1.5
An action-packed guide for the easy-to-use, high performance, Python based free open source NumPy mathematical library using real-world examples
Beginner's Guide
Ivan Idris
[PACKT] open source

NumPy/ScpiPy

http://www.numpy.org
http://www.scipy.org/SciPy

http://scikit-image.org


ln(mL)
Θ_1
M_{21}
Ln(mL)
−18000
−18500
−19000
−19500
−20000
0.010
0.008
0.006
0.004
0.002
0.000
6000
5000
4000
3000
2000
1000
0
50
100
250
500
1000
2000
4000
Sites per locus

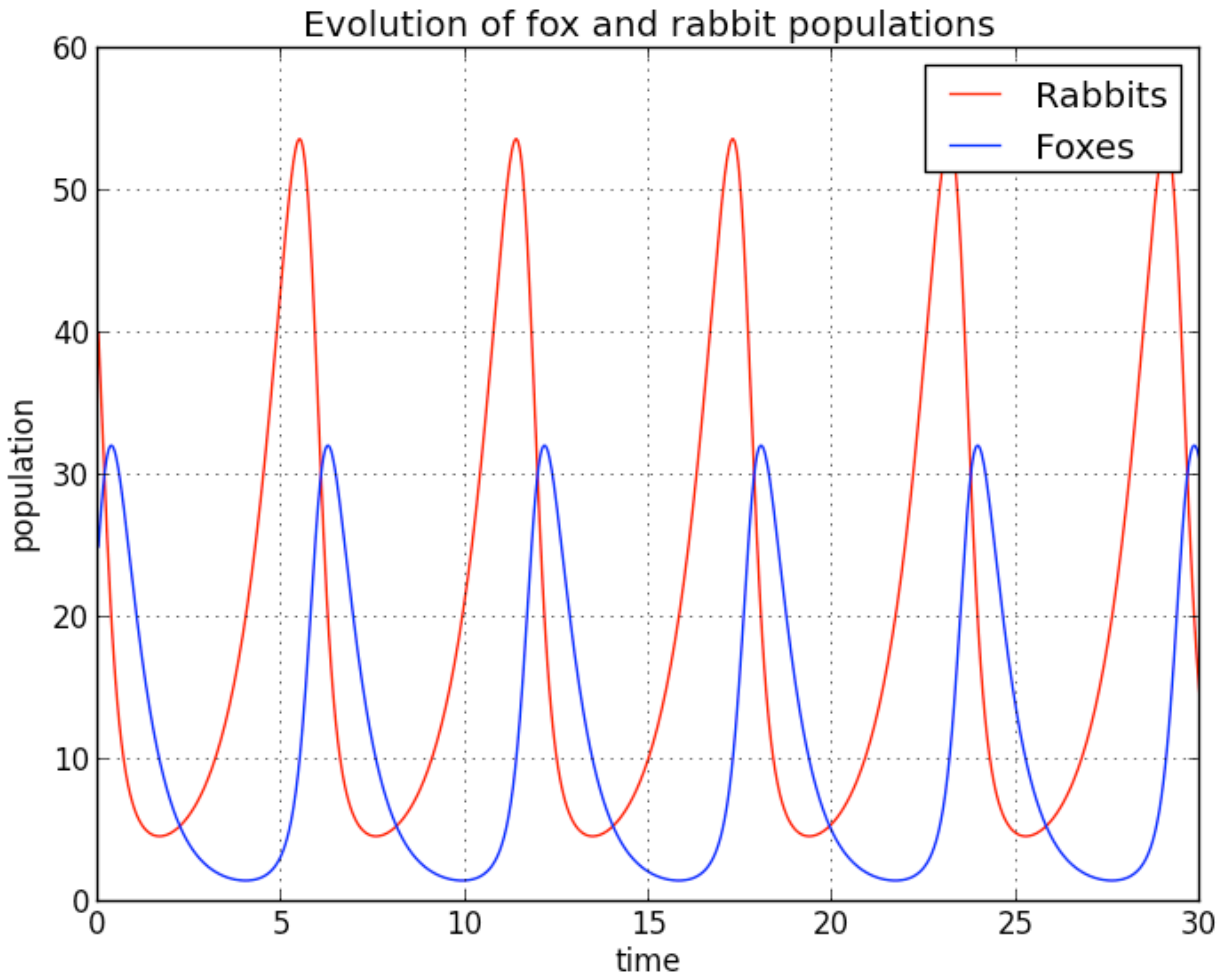


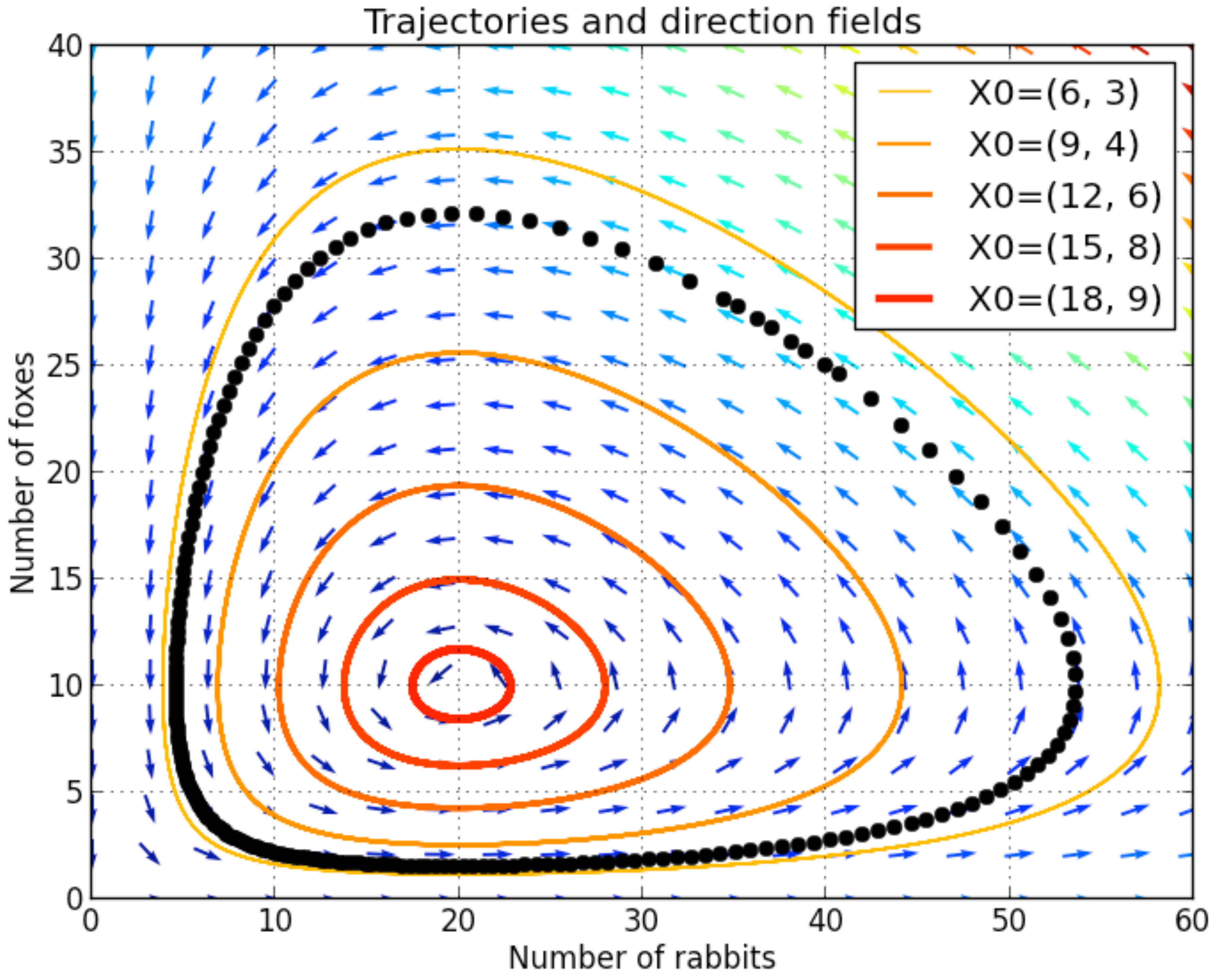
see code in paramplot.py

```
    for line in f:
        myline = line.split()
        m = [float(x) for x in myline]
        data.append(m)
    sorted(data)
    return scipy.array(data)

def coeff(mat,indices,fit):
    yy  = []
    yy2 = []
    ii = indices[:]
    i0 = ii.pop(0)
    x = [np.log(10000.0/(u[i0])) for u in mat]
    x2 = np.arange(min(x), max(x), .01)
    for i in ii:
        y = [u[i] for u in mat]
        coeff = np.polyfit(x, y, fit)
        y2 = np.polyval(coeff, x2)
        yy.append(y)
        yy2.append(y2)
    return x,yy,x2,yy2

ml = read("lml")
th = read("theta1")
m  = read("m21")

rx,ry,xeml,eml = coeff(ml,[0,1],5)
rx1,ry1,xth1,yth1 = coeff(th,[0,1,2,3],5)
rx2,ry2,xm21,ym21 = coeff(m,[0,1,2,3],5)
#print ry1
#print ry2
host = host_subplot(111, axes_class=AA.Axes)
plt.subplots_adjust(right=0.75)
par1 = host.twinx()
```

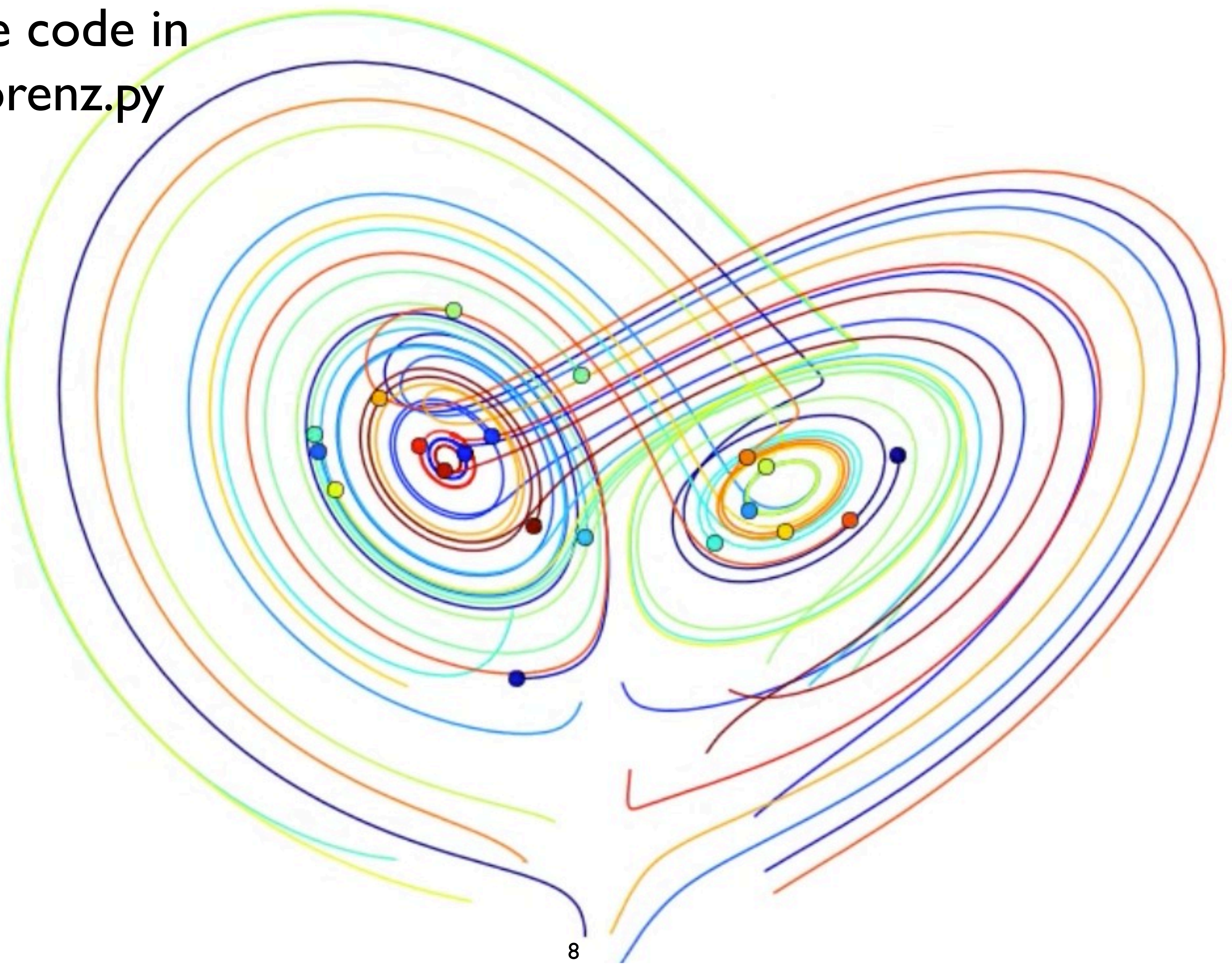
```
#!/usr/bin/env python
#i
# Lotka-Volterra prey-predator model
#
# du/dt =  a*u -   b*u*v
# dv/dt = -c*v + d*b*u*v
#
# u: number of preys (for example, rabbits)
# v: number of predators (for example, foxes)
# a, b, c, d are constant parameters defining the behavior of the population:
# a is the natural growing rate of rabbits, when there's no fox
# b is the natural dying rate of rabbits, due to predation
# c is the natural dying rate of fox, when there's no rabbit
# d is the factor describing how many caught rabbits let create a new fox
#
# We will use X=[u, v] to describe the state of both populations.
#
# http://www.scipy.org/Cookbook/LoktaVolterraTutorial?action=show&redirect=LoktaVolterraTutorial
#
import numpy as np
import pylab as p
from scipy import integrate


# Definition of parameters
#defaults
#a = 1.
#b = 0.1
#c = 1.5
#d = 0.75
a = 1.0
b = 0.1
c = 1.5
d = 0.75
rabbits0 = 40
foxes0=25

def dX_dt(X, t=0):
    """ Return the growth rate of fox and rabbit populations. """
    return np.array([ a*X[0] -   b*X[0]*X[1] ,
                  -c*X[1] + d*b*X[0]*X[1] ])




X_f0 = np.array([     0. ,  0.])
X_f1 = np.array([ c/(d*b), a/b])
np.all(dX_dt(X_f0) == np.zeros(2) ) and np.all(dX_dt(X_f1) == np.zeros(2)) # => True

t = np.linspace(0, 30,  1000)              # time
X0 = np.array([rabbits0,foxes0])                     # initials conditions: 10 rabbits and 5 foxes
X, infodict = integrate.odeint(dX_dt, X0, t, full_output=True)
infodict['message']                     # >>> 'Integration successful.'
```

see code in lotka.py


Evolution of fox and rabbit populations
Rabbits
Foxes
population
time
0
10
20
30
40
50
60
0
5
10
15
20
25
30


Trajectories and direction fields
X0=(6, 3)
X0=(9, 4)
X0=(12, 6)
X0=(15, 8)
X0=(18, 9)
Number of foxes
Number of rabbits
0
5
10
15
20
25
30
35
40
0
10
20
30
40
50
60

see code in
lorenz.py



Object orientation

Python scripts in the shell

https://sites.google.com/site/pythonforscientists/python-vs-matlab

```
nagal:Friday>time python pi.py
3.1408176

real	0m13.107s
user	0m13.079s
sys	0m0.021s
nagal:Friday>time pypy pi.py
3.141696

real	0m1.593s
user	0m1.253s
sys	0m0.053s
```

- Python for your file conversion
- Python for calculations
- Interaction with databases/websites/…/….