P609 Final Project

Spring-Block Model of Earthquakes

Wencao Yang

Introduction:

The first simulations of earthquakes were done by Burridge and Knopoff in 1967. The model consists of a one-dimensional array of blocks on a substracte (Figure 1). Each block is connected to its nearest neighbors by springs with spring constant k_c , which represent the linear elastic response of the system to compressional deformations. Each block is also connected by a spring with k_L to a fixed loader plate.

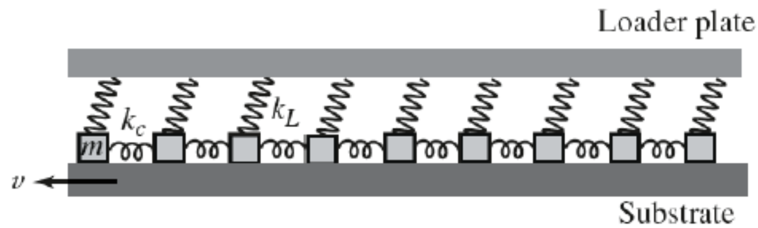


Figure.1

Schematic of the Burridge-Knopoff model. Blocks with mass m attached to their nearest neighbors by springs with spring constant k_c. They are also attached to a fixed loader plate with spring constant k_L. The substrate moves with speed v to left.

The equation of motion of the Burridge-Knopoff model can be written as

$$m\ddot{x}_j = k_c(x_{j+1} - 2x_j + x_{j-1}) - k_L x_j - F(v + \dot{x}_j) \, , \tag{1}$$

And then introduce dimensionless variables, which we take to be $u_j = (k_L / F_0)x_j$, $\omega_L^2 = k_L / m$ and $\tau = \omega_L t$

$$\ddot{u}_j = l^2(u_{j+1} - 2u_j + u_{j-1}) - u_j - \phi(2\alpha\nu + 2\alpha\dot{u}_j) \, , \tag{2}$$

Where $\phi(\omega) = F(\omega)/F_0$, the stiffness parameter $l = \sqrt{k_c / k_L}$, $\nu = \upsilon k_L / \omega_L F_0$ and $2\alpha = \omega_L F_0 / k_L \upsilon$; the dot denotes differentiation with respect to τ . The equation of motion can be solved using Euler-Richardson algorithm with $\Delta\tau = 10^{-3}$

And the velocity of a block is set to zero if at any time the speed of the block relative to the substrate is less than a parameter υ_0, its speed is decreasing , and the force due to the springs is less than F_0. Otherwise, the friction force is given by

$$\phi(\omega) = \frac{1-\sigma}{1+\dfrac{\omega}{1-\sigma}}(\omega > 0) \tag{3}$$

Where the parameter σ represents the drop of the friction force at the onset of the slip. If a block is stuck, the calculation of the static friction force is a bit more involved. If the total force on a block due to the springs is to the right, then the static friction force up to a maximum value of F_0. However, if the total spring force is to the left, the static friction is chosen so that the acceleration of the block is zero. Typical values of the parameter are $F_0 = 1$, $l = 10$, $\sigma = 0.01$, $\alpha = 2.5$ and $\upsilon_0 = 10^{-5}$.

Initially we set $u_j = 0$ and assign small random displacements to all the blocks. The blocks will then move according to (2). For simplicity we set the substrate velocity v = 0, and when all the blocks become stuck, we move all the blocks to the left by an equal amount such that the total force due to the springs on one block equals unity (F_0). This procedure will then cause one block to move or slip. As this block moves, other neighboring blocks may move leading to an earthquake. Eventually, all the blocks will again become stuck. The main quantities of interest are P(S), the distribution of the number of blocks that have moved during an earthquake, and P(M), the distribution of the net displacement of the blocks during an earthquake, where

$$M = \sum_i \Delta u_j \tag{4}$$

The sum over i in (14.20) is over the blocks involved in an earthquake and Δu_i is the net displacement of the blocks during the earthquake. Do P(s) and P(M) exhibit scaling consistent with Gutenberg–Richter?

The movement of the blocks represents the slip of the two surfaces of a fault past one another during an earthquake. The stick-slip behavior of this model is similar to that of a real earthquake fault.

Simulation:

Plug parameters into equation (1), we get

$$\ddot{u}_j = 100(u_{j+1} - 2u_j + u_{j-1}) - u_j - \phi(5\dot{u}_j) \tag{5}$$

Substituting equation (3)

$$\ddot{u}_j = 100u_{j+1} - 201u_j + 100u_{j-1} - \frac{0.99}{1+\frac{5\dot{u}_j}{0.99}}(\dot{u}_j > 0) \qquad (6)$$

And here we define + to the right, - to the left, following is the procedure:

1). Initialize u_j and $v_j(\dot{u}_j)$

2). Solve equation (6), set the time step $\Delta\tau = 10^{-3}$. About the static friction, if the total force on a block due to the springs is to the right, then the static friction force up to a maximum value of F_0. If the total spring force is to the left, the static friction is chosen so that the acceleration of the block is zero (we don't use periodic boundary condition here)

3). Update velocity according to the following rule:

$v_j = 0$ if at any time

a) the speed of the block relative to the substrate is less than a parameter $v_0 = 10^{-5}$

b) its speed is decreasing

c) the force due to the springs is less than $F_0 = 1$

4). Judge if all the blocks stuck, if not, repeat 3). Else, move all the blocks to the left by an equal amount such that the total force due to the springs on one block equals unity (F_0). (pick this block randomly) Then repeat 3).

Note every time the blocks stuck, we will add one to the number of earthquake events.

5). Run a long time and stop, for each earthquake, calculate the number of moving blocks and total displacement (S and M), then calculate P(S) and P(M), and plot them finally. Verify if they satisfy the Gutenberg-Richter relationship.

Code: #python

```
import random as rd
from math import *
import matplotlib.pyplot as plt
import numpy as np


def initialVelocity():
    v=[]
    for i in range(n):
```

```
      v.append(0)
    return v

def initialU():
    u=[]
    for i in range(n):
        u.append((rd.random())/m)#m to make the displacement
small
    return u

def spring(u,i):
    f=0
    if i==0:
        f=100*u[i+1]-201*u[i]
    elif i==n-1:
        f=100*u[i-1]-201*u[i]
    elif 0<i<n-1:
        f=100*u[i+1]-201*u[i]+100*u[i-1]
    return f

def ifDecreasing(v,v2,i):
    if v[i]<v2[i]:
        return 1#decreasing
    else:
        return 0

def updateVelocity(u,v,v2):
    for i in range(n):
        if v[i]<v0:
            v[i]=0
        if spring(u,i)<1:
            v[i]=0
        if ifDecreasing(v,v2,i):
            v[i]=0

def ifAllStuck(v):
    if v.count(0)==len(v):
      return 1#stuck
    else:
      return 0

def draw(M):
    a=np.histogram(M,bins=1000)[0]
    b=np.histogram(M,bins=1000)[1]
    c=[]
    d=[]
    e=[]
    f=[]
```

```
    sum_a=sum(a)
    for i in range(len(a)):
        d.append((a[i])/float(sum_a))
    for i in range(len(b)-1):
        c.append((b[i]+b[i+1])/2.0)
    for i in range(len(d)):
        if d[i]>threshold:
            e.append(log(d[i]))
            f.append(log(c[i]))
    plt.plot(f,e,marker='o',color='r')

n=400
m=5
dt=0.001
v0=0.00001
t=100000
alpha=2.5
threshold=0.00001
S=[]
M=[]
v=initialVelocity()
u=initialU()
u2=list(u)#make a copy of u

for i in range(t):
    v2=list(v)
    for j in range(n):
        if v[j]!=0:
            a=spring(u,j)-0.99/(1+(2*alpha/0.99)*v[j])
            vmid=v[j]+a*dt
            amid=spring(u,j)-0.99/(1+2*alpha*vmid/0.99)
            v[j]=v[j]+amid*dt
            if v[j]>v2[j]:
                u[j]=u[j]+vmid*dt
        elif spring(u,j)-1>0:
            a=spring(u,j)-1
            vmid=v[j]+a*dt
            amid=spring(u,j)-0.99/(1+5*vmid/0.99)
            v[j]=v[j]+amid*dt
            if v[j]>v2[j]:
                u[j]=u[j]+vmid*dt
    updateVelocity(u,v,v2)
    if ifAllStuck(v):
        a=list(map(lambda x: x[0]-x[1], zip(u, u2)))#find the
moved blocks
        moved=n-a.count(0)
        #for i in range(n):
        #    a[i]=abs(a[i])
```

```
        if moved!=0:
            S.append(moved)
            M.append(sum(a))
        magic=rd.randint(0,n-1)#pick a random block to make F=F0
        offset=(1-spring(u,magic))/200
        for i in range(n):
            u[i]=u[i]-offset
        u2=list(u)#keep the old_version u to be compared next
time

fig = plt.figure()
plt.subplot(211)
draw(S)
plt.xlabel('log(S)')
plt.ylabel('log(P(S))')
plt.subplot(212)
draw(M)
plt.xlabel('log(M)')
plt.ylabel('log(P(M))')
plt.show()
```

Result:

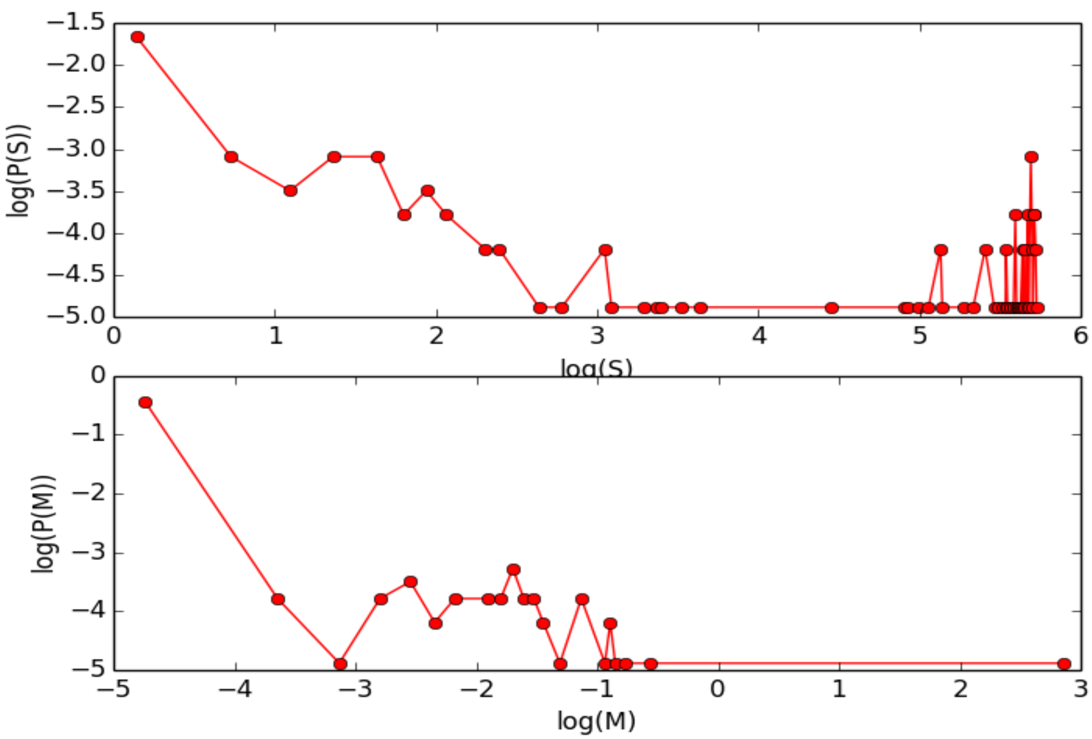


Figure.2

Gutenberg–Richter scaling rule: $P(M) = M^{-b}$, which gives $\log(P(M)) = -b\log(M) + A$.

From Figure.1 we can find that the first half part basically obey the $\log(P(M)) = -b\log(M) + A$

Estimation of b is approximate 2.2 which is slightly bigger than the value 2.02 given by Ref[1]

So the most import conclusion of my simulation is:

$$P(M) \propto M^{-2.2}$$

Discussion :

whole process is like Ref[2], where different α and other parameters also be discussed. And I just check different α here

α =1

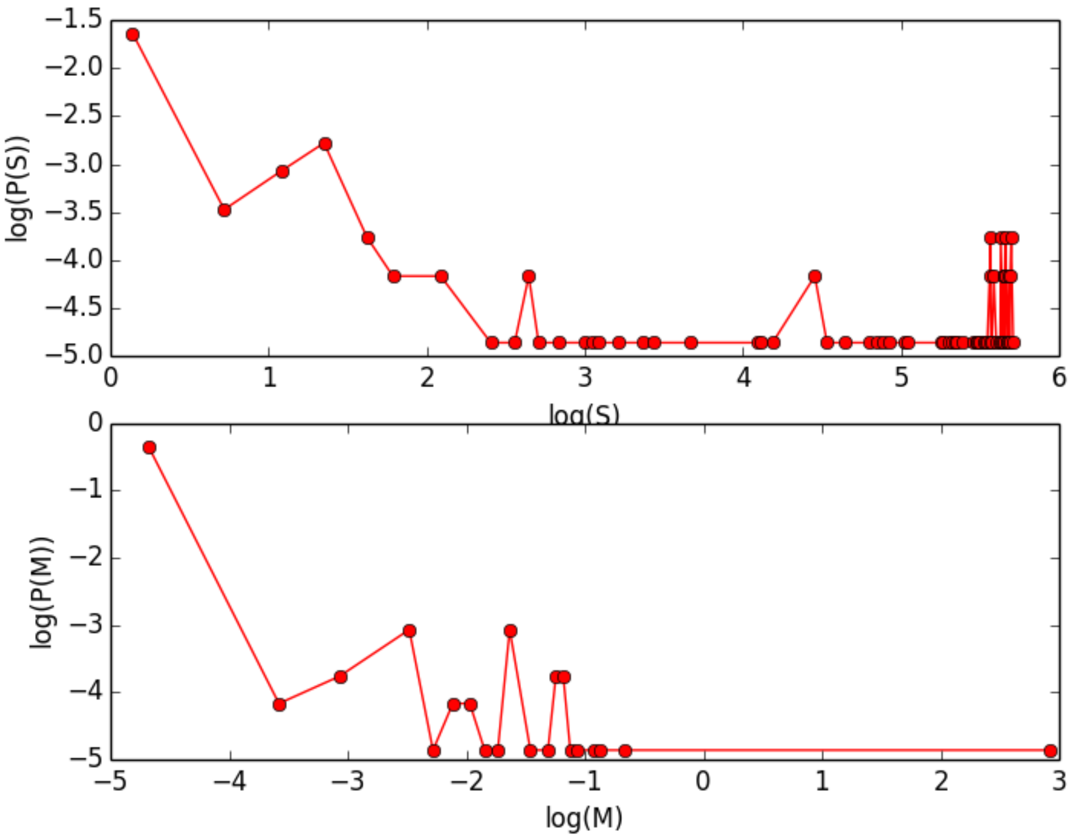


Figure.3

α =5

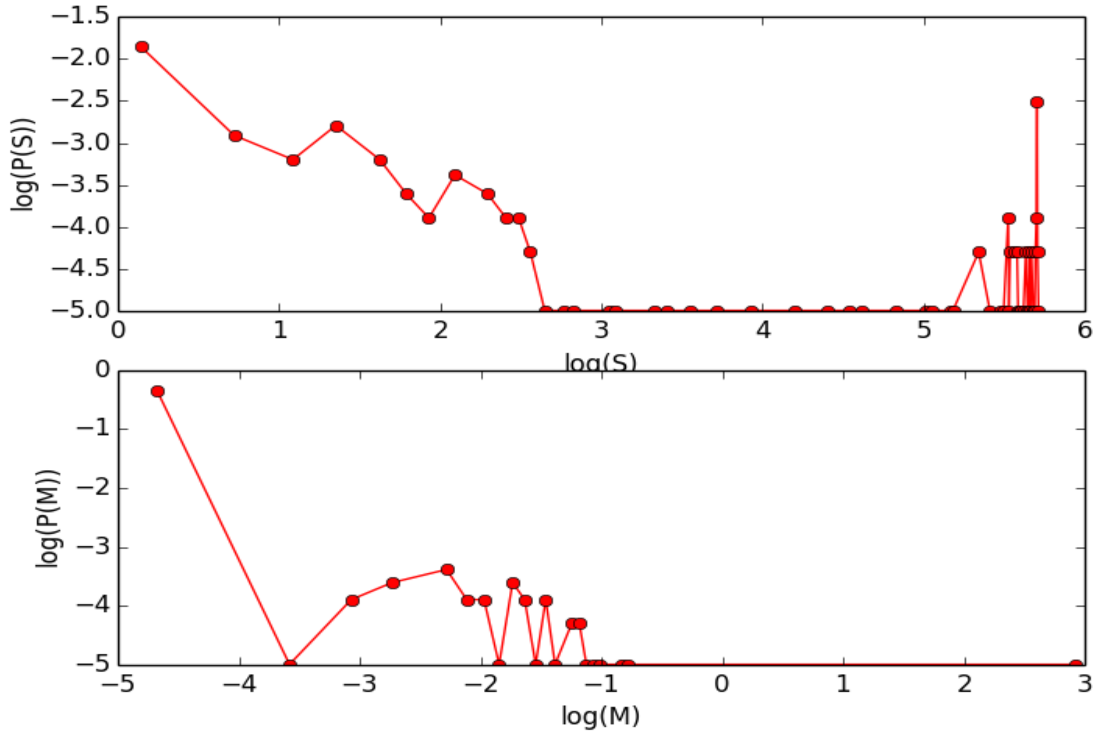
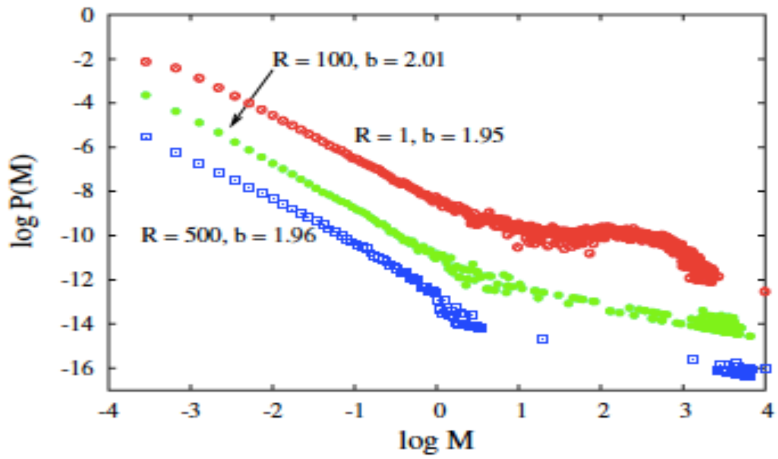


Figure.4

Ref[1] gives an improvement of this model where one block is connected to R neighbors (different directions) with spring constant kc/R, R = 1 corresponds to the usual Burridge-Knopoff model. From where we can find figure.5 gives the compare of different R

And we can find the left half part (log(M)<-3) of our figure is similar with figure.5, however the rest part is not. I think it's because the simulation is sensitive to many parameter, i.e. initial value, number of blocks, simulation time, etc



(a) $\alpha = 2.5$.

Figure.5[1]

Some intrinsic property of Burridge and Knopoff model can be found in Ref[3]

Ref[4] gives the relation with Self-Organized Criticality of cellular automata earthquake model

Reference:

[1] Junchao Xia, Harvey Gould, W. Klein, and J. B. Rundle , Phys. Rev. Letter 95, 248501 (2005)
[2] J. M. Carlson and J. S. Langer, Phys. Rev. A 40, 6470(1989).
[3] J. M. Carlson, J. S. Langer, B. E. Shaw, and C. Tang, Phys.Rev. A 44, 884 (1991).
[4]Z. Olami, H. J. S. Feder, and K. Christensen, Phys. Rev. Lett. 68, 1244 (1992).