

卒業論文

地震の発生頻度とエネルギーの  
尺度不変性

木内大輔

青山学院大学・理工学部・物理学科  
羽田野研究室

2002年度

# 地震の発生頻度とエネルギーの 尺度不変性

木内 大輔

羽田野研究室

平成15年1月20日

## 概要

地震には特徴的な長さを持たない様々な尺度不変性というものがある。地震の発生頻度とエネルギーの関係にも尺度不変性がある。具体的にはエネルギー  $E$  の地震の頻度が  $E^{-2b/d}$  乗に比例する (ここで  $d$  は次元である)。この尺度不変性をモデルで再現するのが本研究の目的である。シミュレーションには、地震をおこす原因となるプレートとマンツルの動きを、バネと箱にみたてたスティックスリップモデルを用いた。

# 目次

1	はじめに	3
2	モデルの説明	4
2.1	スティックスリップモデルとは？	4
2.2	二次元のスティックスリップモデル	5
3	シミュレーションの流れと結果	9
3.1	シミュレーションの流れ	9
3.2	シミュレーション結果	10
3.2.1	地震の時間間隔	10
3.2.2	地震の広がり	12
3.2.3	地震の発生頻度とエネルギー	14
3.2.4	一次元モデルと二次元モデルの違い	17
4	まとめ	20

## 1 はじめに

日本周辺では地震が頻繁に発生する。地震はランダムに発生していると思っている人がいるであろうが、地震の規模とその発生頻度の間にはある経験則が見出されている。その経験則はグーテンベルグリヒター則といい、以下のような関係式である：

$$\log N = a - bM \quad (1)$$

ここで、 $M$  は地震のマグニチュード、 $N(M)$  はマグニチュードが  $M$  以上の地震の発生回数、 $a, b$  は定数である。 $b$  は多くの場合 1 に近いことがわかっている。一般に大地震の発生する地域では  $b$  値は大きく、小地震の発生する地域では小さい。

一方で、一回の地震で放出されるエネルギー  $E$  とマグニチュード  $M$  は

$$M = \frac{2}{d} \log E \quad (2)$$

のように表される。ここで  $d$  は次元である。したがって式 (1) は

$$N(E) \propto E^{-2b/d} \quad (3)$$

と変換することができる。ここで  $N(E)$  はエネルギー  $E$  以上の地震の発生回数である。つまり地震の発生頻度はエネルギーに関してべき乗則になる。

べき乗則には指数関数のような特徴的なスケールがない。どのようなエネルギースケールで見ても同じ指数のべき乗則に見える。これを尺度不変性と言う。地震をモデル化して、上の尺度不変な関係式 3 を再現するのが本研究の目的である。

第 2 節で本研究で用いたスティックスリップモデルを説明する。第 3 節でシミュレーションの流れと、その結果としてエネルギーの尺度不変性が再現できたことを報告する。第 4 節で本研究のまとめを行う。

## 2 モデルの説明

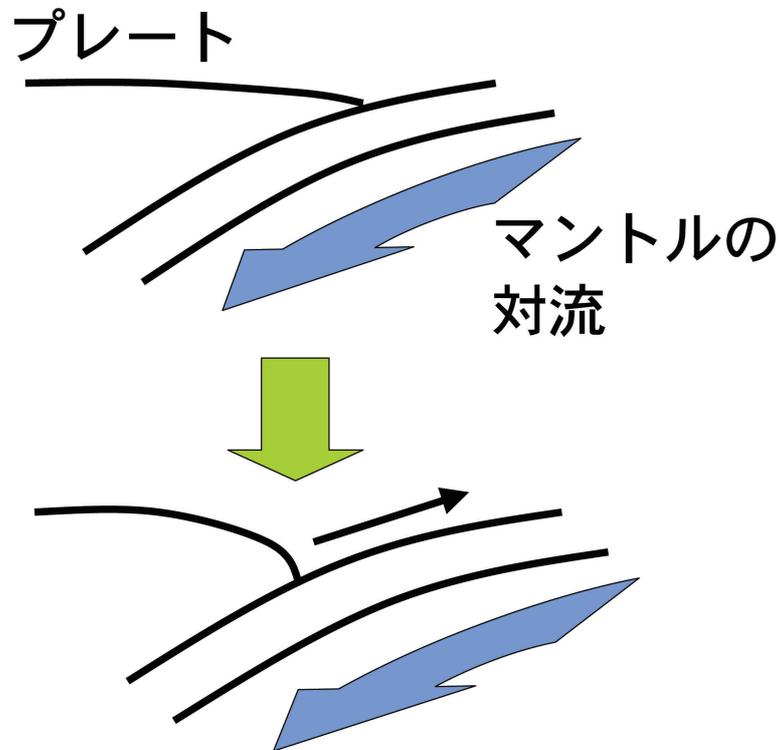


図 1: マントルとプレートによる地震発生の原理。

大きな地震はマンタル対流が引き起こしていることはよく知られている。マンタルの対流によって年間数センチほど動く地殻が、大陸側の動かない地殻と、あるところで擦れる。そこで溜った歪みのエネルギーが一気に開放されるのが地震である (図 1)。地殻は岩石から構成されており、弾性的な性質を持つ。以上のような地震の性質をモデル化したものが以下で述べるスティックスリップモデルである。

### 2.1 スティックスリップモデルとは？

スティックスリップモデルとは、摩擦のある箱と天井が、板状のバネによってつながれたモデルである [1]。箱と床は摩擦のある接触をしている (図 2)。床がゆっくり左に移動すると、天井からのバネは箱に右のほうに戻そうとする力を及ぼす。しかし箱と床との間には摩擦があるため、バ

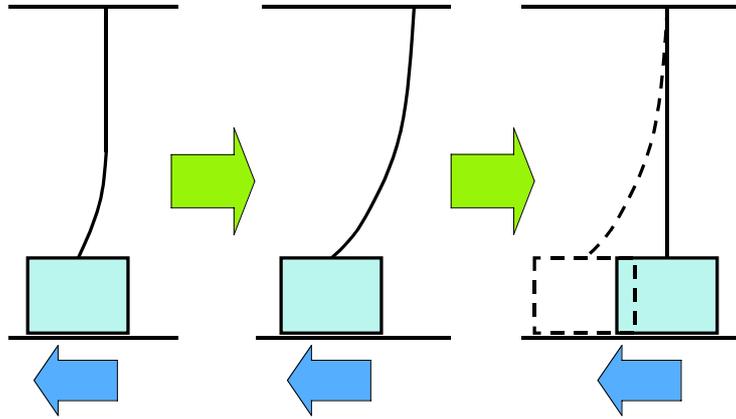


図 2: スティックスリップモデルの一連の動き。

ネの力が最大静摩擦力を越えないうちは、箱は床にひっついたまま動く（スティック）。箱の変位が大きくなり、バネの力が最大静摩擦力を越えると箱は右に動き出す。この時箱の質量が大変小さく、また床と箱の間の動摩擦も静摩擦に比べずっと小さいならば、箱はほとんど瞬間的にバネの力がゼロになる位置にまで移動する（スリップ）。このような運動をする箱をいくつもつなげたのが、次に説明する二次元のスティックスリップモデルである。

## 2.2 二次元のスティックスリップモデル

本研究で用いた二次元のスティックスリップモデルとは図3のようなものである [2]。ブロック  $B_{i,j}$  は前後のブロックとコイルバネ  $K_{i,j-1}$  と  $K_{i,j}$  で、左右のブロックと板バネ  $L_{i-1,j}$  と  $L_{i,j}$  で、天井とは板バネ  $M_{i,j}$  でつながれている。また各ブロックは床と摩擦接触している。床は矢印の方向に一定速度で進んでいるとする。床の進行方向を  $y$  軸、それと直角方向を  $x$  軸とする。 $y$  方向にはコイルバネ  $K$  で、 $x$  軸方向には板バネ  $L$  でブロック同士はつながれている。よってブロックの運動方向は  $y$  軸方向のみである。そこで  $y_{i,j}$  を  $M_{i,j}$  の支点の真下からの変位とする。初期状態はすべてのブロックに対して  $y_{i,j} = 0$  とする。

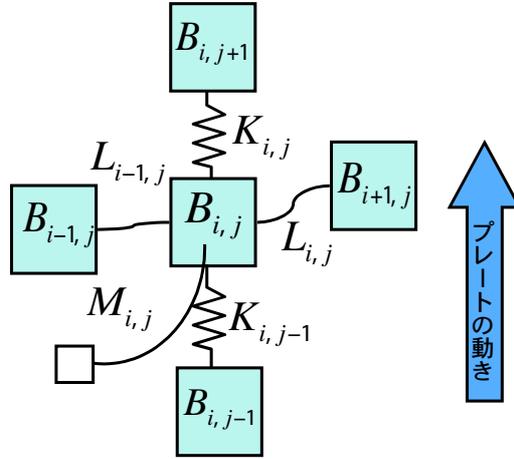


図 3: 二次元のスティックスリップモデル。

(i) ブロック  $B_{i,j}$  がうける力  $F_{i,j}$

ブロック  $B_{i,j}$  が周りの 5 個のバネ (バネ定数  $K_{i,j}$ 、 $K_{i,j-1}$ 、 $L_{i,j}$ 、 $L_{i-1,j}$ 、 $M_{i,j}$ ) から  $y$  軸の負方向にうける力は

$$F_{i,j} = K_{i,j-1}(y_{i,j} - y_{i,j-1}) - K_{i,j}(y_{i,j+1} - y_{i,j}) + L_{i,j}(y_{i,j} - y_{i+1,j}) - L_{i-1,j}(y_{i-1,j} - y_{i,j}) + M_{i,j}y_{i,j} \quad (4)$$

$$= (K_{i,j} + K_{i,j-1} + L_{i,j} + L_{i-1,j} + M_{i,j})y_{i,j} - (K_{i,j}y_{i,j+1} + K_{i,j-1}y_{i,j-1} + L_{i,j}y_{i+1,j} + L_{i-1,j}y_{i-1,j}) \quad (5)$$

である。ここで、

$$(K_{i,j} + K_{i,j-1} + L_{i,j} + L_{i-1,j} + M_{i,j})y_{i,j} = \phi_{i,j} \quad (6)$$

$$(K_{i,j}y_{i,j+1} + K_{i,j-1}y_{i,j-1} + L_{i,j}y_{i+1,j} + L_{i-1,j}y_{i-1,j}) = \psi_{i,j} \quad (7)$$

とおくと、式 (5) は

$$F_{i,j} = \phi_{i,j}y_{i,j} - \psi_{i,j} \quad (8)$$

と表せる。

力  $F_{i,j}$  が最大静摩擦力  $R_{i,j}$  を超えたとき滑りが起こる。また 1 個のブロックが滑りを起こしたとき、この影響で周りのブロックも滑りを起こす

可能性が生じる。こうして一連のブロックの滑りが引き起こされる。この現象を地震の伝搬とする。ここでは、滑った複数のブロックの塊全体を1つの地震とみなす。

(ii) ブロック  $B_{i,j}$  が滑りを起こしたときの変位  $y_{i,j}$  の変化

力が  $F_{i,j} > R_{i,j}$  となりブロック  $B_{i,j}$  が滑りを起こしたときの変位  $y_{i,j}$  の変化について説明する。 $y_{i,j}$  の滑り後の値を  $y_{i,j}^*$  とする。 $y_{i,j}^*$  は  $F_{i,j} = 0$  とするようなものでなくてはならない。よって式 (8) は

$$\phi_{i,j} y_{i,j}^* - \psi_{i,j} = 0 \quad (9)$$

となる。すなわち

$$y_{i,j}^* = \frac{\psi_{i,j}}{\phi_{i,j}} \quad (10)$$

である。一方で式 (8) より

$$y_{i,j} = \frac{\psi_{i,j}}{\phi_{i,j}} - \frac{F_{i,j}}{\phi_{i,j}} \quad (11)$$

である。この式と式 (10) の差をとると

$$y_{i,j} - y_{i,j}^* = \frac{F_{i,j}}{\phi_{i,j}} \quad (12)$$

となる。したがって  $B_{i,j}$  が滑りを起こしたときの  $y_{i,j}$  の変化量を  $S_{i,j}$  とすると

$$S_{i,j} = \frac{F_{i,j}}{\phi_{i,j}} \quad (13)$$

と書ける。

(iii) ブロック  $B_{i,j}$  が滑りを起こしたときに放出されるエネルギー  $E_{i,j}$

ブロック  $B_{i,j}$  の滑りを起こしたことによりエネルギーを放出するのは、バネ  $K_{i,j}$ 、 $K_{i,j-1}$ 、 $L_{i,j}$ 、 $L_{i-1,j}$ 、 $M_{i,j}$  の5個である。以下にこれらのバネのエネルギーの変化量を示す。

バネ	放出するエネルギー
$K_{i,j}$	$\frac{K_{i,j} S_{i,j}}{2} [(y_{i,j} + y_{i,j}^*) - 2y_{i,j+1}]$
$K_{i,j-1}$	$\frac{K_{i,j-1} S_{i,j}}{2} [(y_{i,j} + y_{i,j}^*) - 2y_{i,j-1}]$
$L_{i,j}$	$\frac{L_{i,j} S_{i,j}}{2} [(y_{i,j} + y_{i,j}^*) - 2y_{i+1,j}]$
$L_{i-1,j}$	$\frac{L_{i-1,j} S_{i,j}}{2} [(y_{i,j} + y_{i,j}^*) - 2y_{i-1,j}]$
$M_{i,j}$	$\frac{M_{i,j} S_{i,j}}{2} (y_{i,j} + y_{i,j}^*)$

これらのバネの放出するエネルギーの和  $E_{i,j}$  は

$$E_{i,j} = \frac{S_{i,j}}{2} [\phi_{i,j}(y_{i,j} + y_{i,j}^*) - 2\psi_{i,j}] \quad (14)$$

である。式 (9) を用いると

$$E_{i,j} = \frac{S_{i,j}}{2} (\phi_{i,j}y_{i,j} - \psi_{i,j}) = \frac{S_{i,j}F_{i,j}}{2} = \frac{F_{i,j}^2}{2\phi_{i,j}} \quad (15)$$

となる。つまり、ブロック  $B_{i,j}$  が滑りを起こしたときに発生するエネルギーは、滑りを起こす直前の  $B_{i,j}$  が受ける力の自乗を  $2\phi_{i,j}$  で割ったものであることがわかる。

今回用いたパラメーターの大きさは以下の通りである:

$$K_{i,j} = 5.0 \times 10^{23} \times r_{K,i,j} \quad (16)$$

$$L_{i,j} = 5.0 \times 10^{23} \times r_{L,i,j} \quad (17)$$

$$M_{i,j} = 5.0 \times 10^{23} \times r_{M,i,j} \quad (18)$$

$$R_{i,j} = 1.0 \times 10^{25} \times r_{R,i,j} \quad (19)$$

ここで  $r_{K,i,j}$  などはそれぞれのパラメーターにゆらぎを与えるための、ある一定の幅  $\Delta$  を持った乱数である。なお、ゆらぎの幅  $\Delta$  は  $r_{K,i,j}$  などに共通とする。

### 3 シミュレーションの流れと結果

この節では、具体的なシミュレーションの手順と、得られた結果を述べる。

#### 3.1 シミュレーションの流れ

##### 1. ゆらぎの決定

バネ定数や最大摩擦係数等にゆらぎとしてパラメーターに幅  $\Delta$  の乱数を与える。一旦決めた値はシミュレーションを通して固定する。

##### 2. ブロックの配置

$x$  方向に 100 個、 $y$  方向に 40 個のブロックを等間隔に配置する。このスティックスリップモデルはプレートの歪みをモデル化したもので、横長の系にした。

##### 3. ブロックの滑り

毎ステップ一定の速さ (10cm) で床を動かし、全ブロックについて  $F_{i,j}$  を計算する。滑り条件  $F_{i,j} > R_{i,j}$  になっているブロックが無いか確認する。滑り条件  $F_{i,j} > R_{i,j}$  に達しているブロックが 1 個でもあれば、そのブロックの変位  $y_{i,j}$  を  $y_{i,j}^*$  に変え、放出エネルギー  $E_{i,j}$  を式 (15) から計算。次のステップには移らず、もう一度全ブロックについての  $F_{i,j}$  を計算し、周りのブロックが滑りを起こすか調べる。この操作を繰り返し、最終的に滑りを起こすブロックの数がゼロになれば、滑りを起こしたブロックの塊が放出したエネルギーの総和を計算する。そして次のステップに移る。

## 3.2 シミュレーション結果

### 3.2.1 地震の時間間隔

パラメーターのゆらぎ幅  $\Delta = 0.05$  のときシミュレーションにおいて、各ステップで滑ったブロックの総数の時間変化を図4に示す。ある一定の周期(およそ400~500ステップ)で滑りを起こしたブロックの数が増減している。各周期において本震(滑りを起こしたブロックの数が最も多いとき)、余震(本震発生後の多くのブロックが滑るとき)、平穏期(滑りを起こしたブロック数が少ないかゼロであることが続いている期間)があることが見て分かる。よってスティックスリップモデルは時間に対する地震の発生を再現できている。

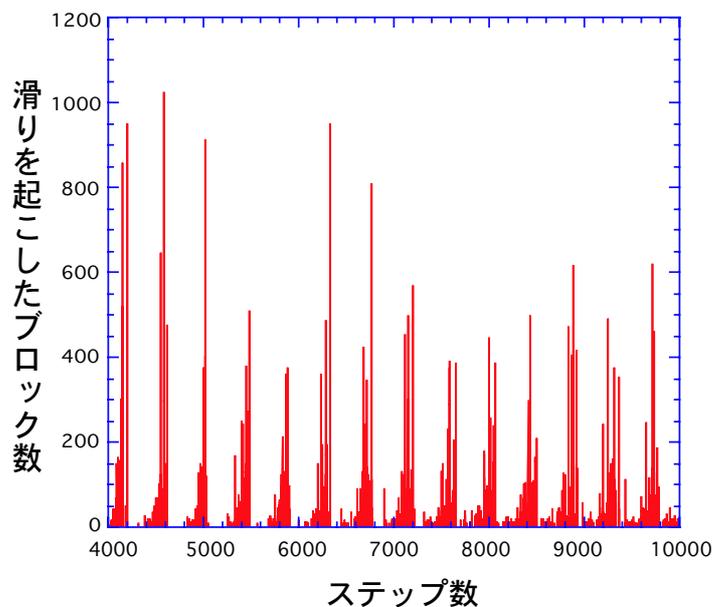


図4: パラメーターの揺らぎ幅  $\Delta = 0.05$  を与えたときの、滑りを起こしたブロックの総数の時間変化。

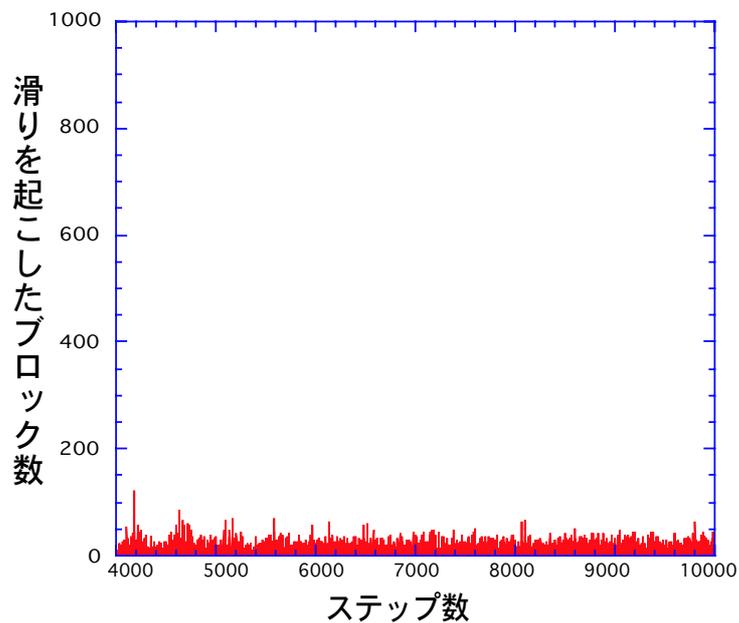


図 5: パラメーターのゆらぎ幅  $\Delta = 0.12$  を与えたときの、滑りを起こしたブロックの総数の時間変化。

一方、図 5 は図 4 より  $\Delta$  を大きくしたものである。 $\Delta$  を大きくしていくと本震、余震、平穏期をはっきりと見ることができなくなる。ブロックの連鎖反応は起きず、常に小規模の地震が発生している。大地震の発生しない地域では、このように多数の小規模な地震が発生している。そのため大地震に発展しない。つまり、 $\Delta$  を大きくすると大地震の発生しない地域の地震を再現していると言える。

### 3.2.2 地震の広がり

本研究では、滑りをおこしたブロックの広がり全体を一つの地震とする。図6は本震時のシミュレーションのスナップショットである。ブロックの連鎖反応が起きており、三つの地震が発生していることが分かる。このシミュレーションでは、境界条件を用いているので右上の塊と右下の塊は同じ塊である。図7は同じシミュレーションの平穩期の地震の広がりである。図6のように多数のブロックの連鎖反応は起こっていない。

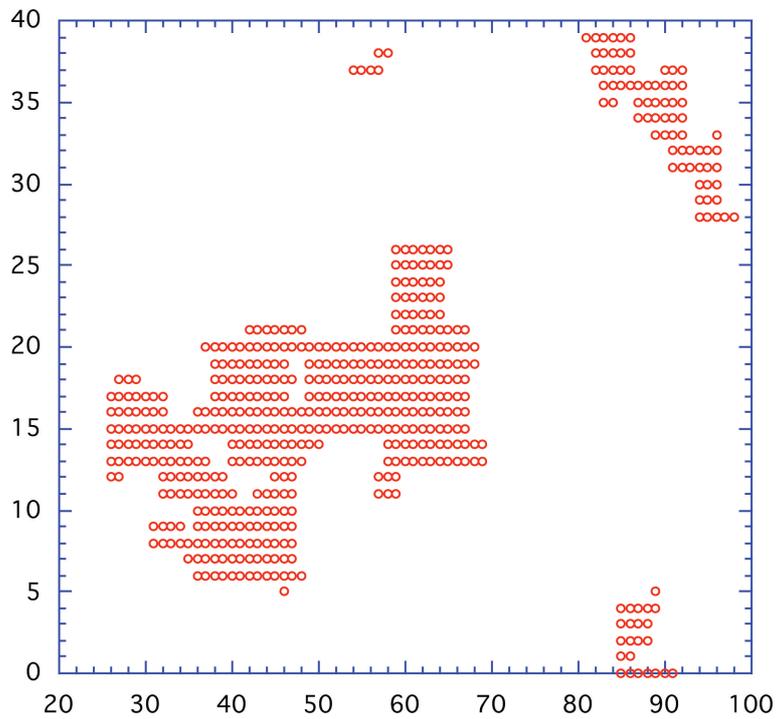


図6: 本震の地震の広がり (ブロックの連鎖反応)。 $\Delta = 0.05$  のとき。

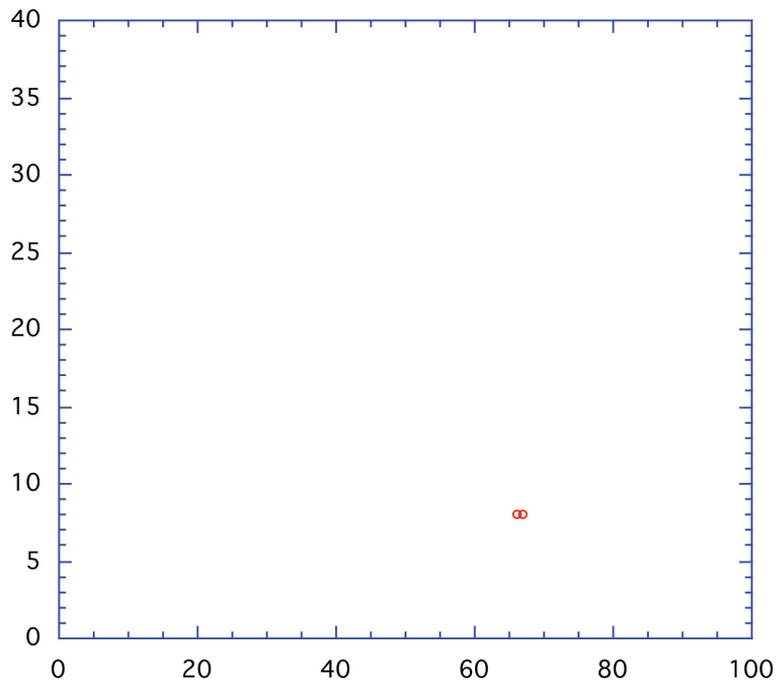


図 7: 平穩期の地震の広がり (ブロックの連鎖反応)。 $\Delta = 0.05$  のとき。

パラメーターにゆらぎを与えることによって、一つのブロックが滑りをおこしたとき、図 6 のように周りの多数のブロックが同じような周期で動いており連鎖的に滑ることもあるが、図 7 のように周りのブロックに連鎖反応が見られないこともある。この  $\Delta$  の大きさが地震の発生頻度とエネルギーの尺度不変性に関わってくる。そのことを次の小節で説明する。

### 3.2.3 地震の発生頻度とエネルギー

図8は  $\Delta = 0.09$  での各ステップにおける地震の発生エネルギー  $E$  と、それ以上の地震の発生回数  $N(E)$  の両対数プロットである。直線は傾き  $-0.87$  の線であるので、指数  $0.87$  の地震の発生頻度とエネルギーの尺度不変性を再現できたと言える。

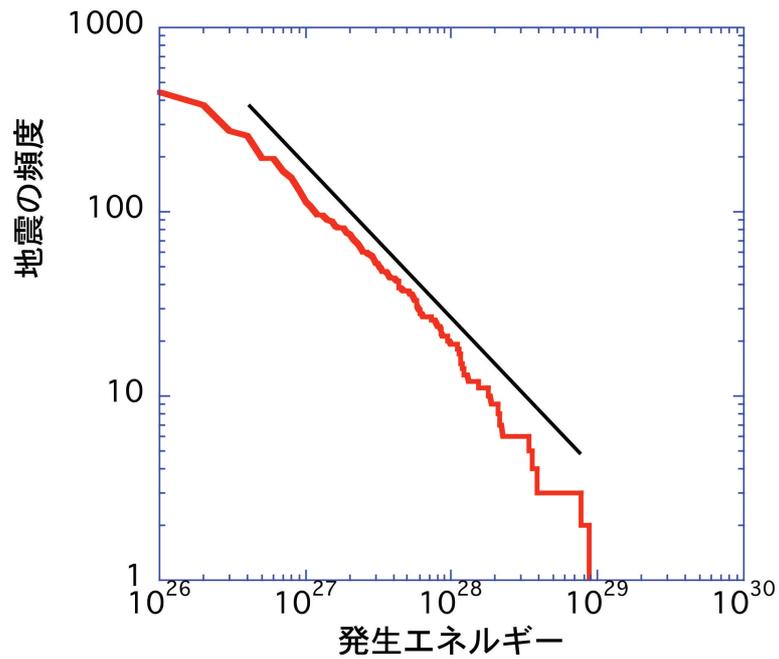


図 8:  $\Delta = 0.09$  のときの尺度不変性。

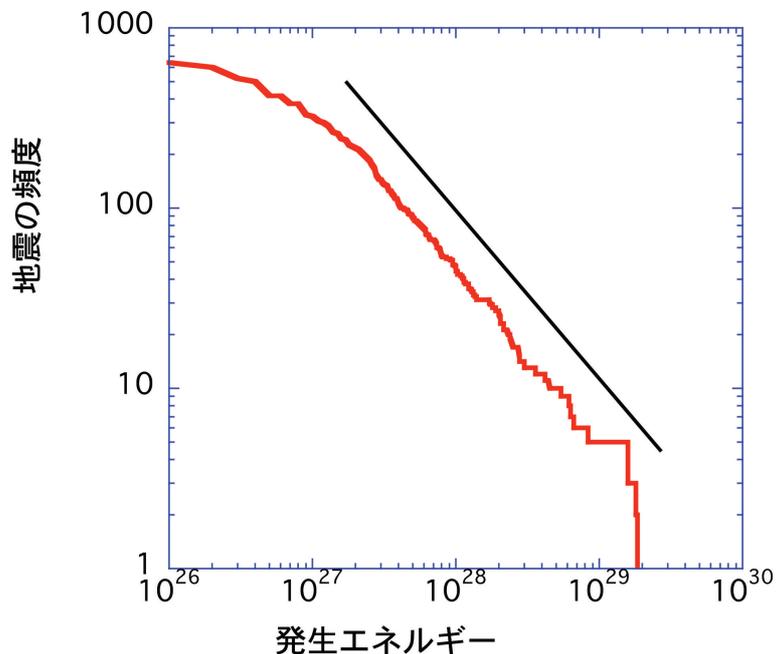


図 9:  $\Delta = 0.05$  のときの尺度不変性。

図 9 は  $\Delta = 0.05$  のときのシミュレーション結果である。直線は傾き  $-1.00$  の線であるので、 $\Delta = 0.05$  のときも地震の発生頻度とエネルギーの尺度不変性を再現できたと言える。

再現することのできた  $b$  値の範囲とパラメーターにあたえたゆらぎの幅を図 10 に示した。

図 10 より、パラメーターにあたえるゆらぎの幅を大きくすると、 $b$  値 (直線の傾き) は小さくなっていくことが分かる。しかし  $b$  値の変化はゆらぎの幅に比例しているわけではない。ゆらぎの幅を小さくしていくと  $b$  値の変化量は小さくなり、 $b$  値は 1 付近に収束をしている。またゆらぎの幅を 0.03、0.04、0.05 と変化させたとき  $b$  値は大きくなっている。

このような現象がおきるのは以下のようなことが原因であると思われる。ゆらぎの幅の小さいときは、それぞれのブロックが同じような周期で動くのでブロックの連鎖反応が起きやすい。したがって、大きなエネルギーを放出する地震が発生することがしばしばある。しかしゆらぎの幅を大きくするにつれてブロックの周期はばらつき大きなエネルギーを放出する地震が発生しにくくなる。よってゆらぎの幅が小さくなると  $b$  値は大きくなる。

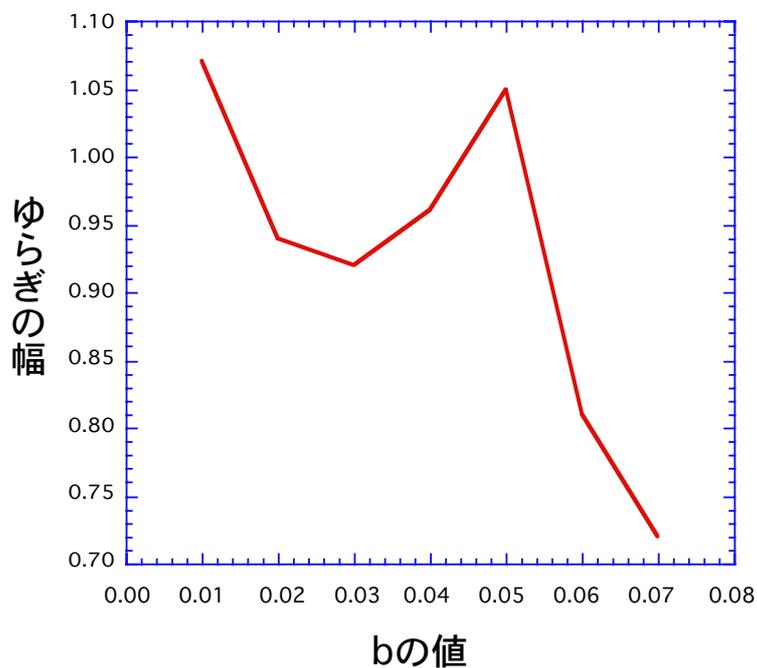


図 10: ゆらぎの幅と  $b$  値の関係。

またゆらぎの幅は乱数を用いてあたえているので、ゆらぎの幅を大きくすると発生する乱数により、連鎖反応に対する不確定要素が大きくなる。よって同じゆらぎの幅でも、発生する乱数を変えればより  $b$  値が違ってくる。このことがゆらぎ幅を大きくさせても  $b$  値が小さくならない、という現象を引き起こしている。

### 3.2.4 一次元モデルと二次元モデルの違い

本研究では、一次元のスティックスリップモデルについてもシミュレーションを行った。一次元モデルでの地震の発生頻度とエネルギーの関係は図11のようになった。図11より、一次元モデルにおいては、地震の頻

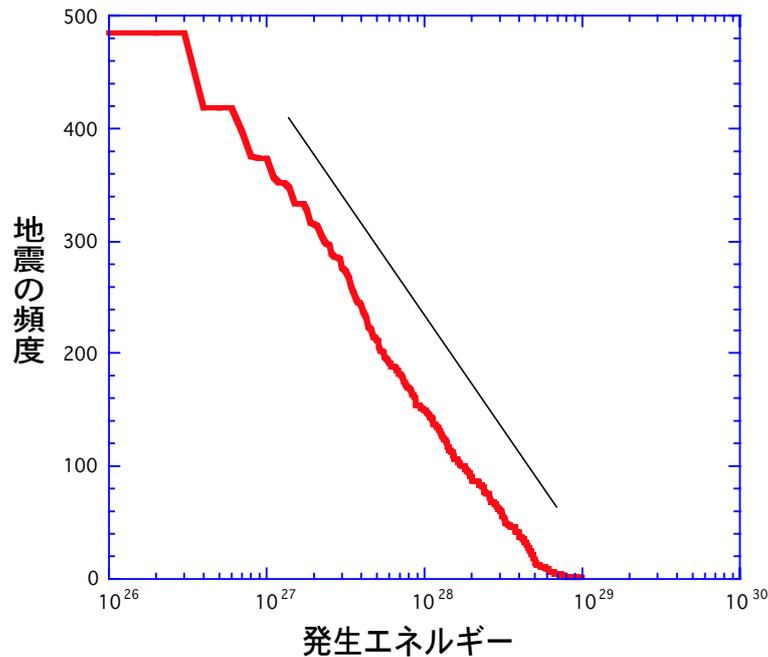


図 11: パラメーター (バネ定数と最大静摩擦力) のゆらぎ幅  $\Delta = 0.05$  のときの  $N(E)$ 。縦軸は線形、横軸は対数である。

度と発生エネルギーの関係が指数関数的になっていることが分かる。つまり特徴的なスケールが存在する。

一次元のモデルでは図12のように、あるブロックが滑りを起こさないとその先のブロックは決して連鎖的に滑らない。よって滑りが隣に連鎖する割合を平均的に  $P$  とすると、 $n$  個先のブロックが連鎖的に滑る確率は  $P^n = e^{-n|\log P|}$  となる。つまり  $1/|\log P|$  が特徴的スケールになる。これにより、尺度不変性を確認できなかったと考えられる。

一方、図8、図9より二次元モデルでは尺度不変性が再現できたことが分かる。三次元空間において発生する地震を二次元化したモデルにおいても、正しく尺度不変性を表すことがなぜできたのか。二次元空間は三次元空間ほどの複雑性をもたないが、地震の広がり方の特徴を再現可能

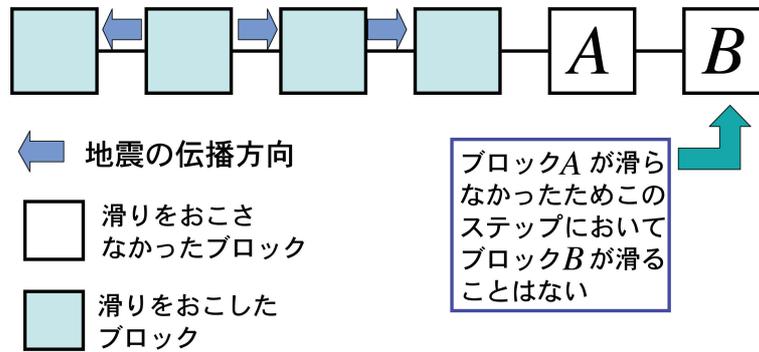


図 12: 一次元モデルにおける地震伝搬の一例。

だったためと考えられる。二次元のスティックスリップモデルでは、仮に動かないブロックがあったとしても、図 13 のように回り込んでその先のブロックにまで連鎖反応を及ぼすことがある。このことが地震の空間的

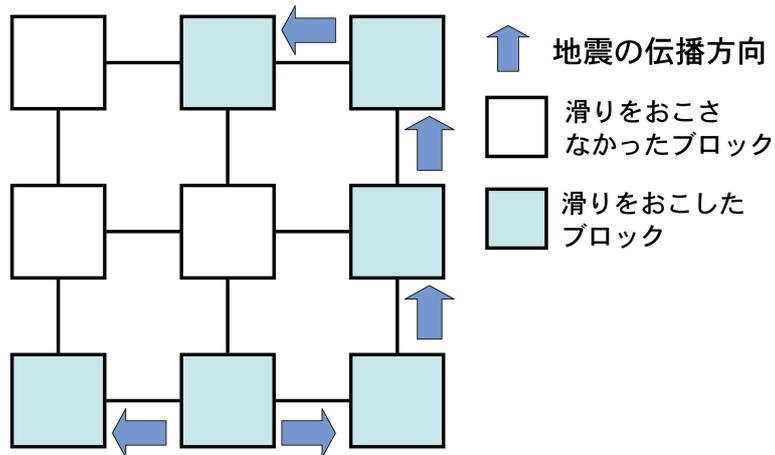


図 13: 二次元モデルにおける地震伝搬の一例。

な尺度不変性となり、地震の発生頻度とエネルギーの尺度不変性にも関係していると思われる。

## 4 まとめ

本研究では地震の発生の原因となる、プレートの動きとマンツルの対流をモデル化したスティックスリップモデルを用いて地震の発生頻度と尺度不変性の関係式 (3) を再現することを目的とした。

シミュレーション結果より、二次元モデルで地震の発生頻度とエネルギーの尺度不変性を再現できたことがわかる。前節より、再現することのできた  $b$  値の範囲は 0.72 ~ 1.07 である。一方、観測データからは、 $b$  値は全世界規模で約 0.8 ~ 1.20 とされている。よってスティックスリップモデルは、平均的な規模の地震の尺度不変性を高い精度で再現することができるモデルであることが分かった。

しかし日本のように地震が多発する地域の  $b$  値 (およそ 1.10 ~ 1.50) までは十分に表せることができなかった。そのような  $b$  値を表すためには今回のモデルでは取り入れなかった力学的要素が必要なかもしれない。またモデルを三次元化してシミュレーションを行えばより高い精度地震の発生頻度とエネルギーの尺度不変性の関係を再現できると期待される。

## 参考文献

- [1] 高安秀樹 『ネットワークと引き込み相転移』数理科学 289 (1987)  
30-35
- [2] 大塚道男 地震、第2号 第24巻 (1971) 13-25

## 二次元のスティックスリップモデルのプログラムリスト

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

#define SIMPLE_SPRNG      /*Simple interface.*/
#include "sprng.h"        /*SPRNG header file.*/
#define SEED 985456376
#define _L 10000          /*Max of rand number.*/

#define _N 100            /*The number of blocks.*/
#define _M 40             /*The number of blocks.*/
#define _S 4000           /*The number of steps.*/
#define _T 20
#define _U 3000           /*OUTPUT3,4*/
#define _H 500            /*OUTPUT3*/
#define _F 0.1
#define _A 180/1000
#define _B 0.91
#define _C 180/1000
#define _D 0.91
#define _E 3711           /*OUTPUT2(cluster)*/

main()
{
    FILE *OUTPUT1;
    FILE *OUTPUT2;
    FILE *OUTPUT3;
    FILE *OUTPUT4;
    FILE *OUTPUT5;

    double b[_N][_M], l[_N][_M], k[_N][_M], m[_N][_M], r[_N][_M];
    double y[_N][_M], f[_N][_M], c[_N][_M], e[_N][_M], p[_N][_M];
```

```

double o[_N][_M];
double a,u,s,h,ransuu1,ransuu2,ransuu3,ransuu4,ransuu5;
double energy1[_S],energy2[_S][_T],katamari[_S][_T];
int n,ban,hantei,kaisuu,kaisuu1,kaisuu2,kaisuu3,step;
int i,j,ti,tj,hindo,drawn,slip[_S],hassei[_N][_M];
int bangou[_N][_M];

OUTPUT1=fopen("jis1.dat","w");
OUTPUT2=fopen("jis2.dat","w");
OUTPUT3=fopen("jis3.dat","w");
OUTPUT4=fopen("jis4.dat","w");
OUTPUT5=fopen("jis5.dat","w");

init_sprng(DEFAULT_RNG_TYPE,SEED,SPRNG_DEFAULT);
/*Initialization*/
print_sprng();

ban=0;
kaisuu=0;
kaisuu1=0;
kaisuu2=0;
kaisuu3=0;
hantei=0;
hindo=0;
h=2.0*pow(10,6);

for(i=0; i<=_S-1; i++){
    for(j=0; j<=_T-1; j++){
        energy2[i][j]=0;
        katamari[i][j]=0;
    }
}

for(i=0; i<=_N-1; i++){
    for(j=0; j<=_M-1; j++){

```

```

k[i][j]=0;
l[i][j]=0;
m[i][j]=0;
r[i][j]=0;
f[i][j]=0; /*Power concerning a block.*/
e[i][j]=0; /*Emission energy.*/
b[i][j]=0; /*Distance from a starting point.*/
y[i][j]=0; /*Move distance of a block.*/
o[i][j]=0; /*The generating place of a earthquake.*/
c[i][j]=0;
hassei[i][j]=0;
bangou[i][j]=0;
}
}

/*Determination of parameter.*/

for(i=0; i<=_N-1; i++){
  for(j=0; j<=_M-1; j++){
    ransuu1=(int)(sprng()*(float)_L);
    /*System size.*/
    ransuu2=(int)(sprng()*(float)_L);
    /*System size.*/
    ransuu3=(int)(sprng()*(float)_L);
    /*System size.*/
    ransuu4=(int)(sprng()*(float)_L);
    /*System size.*/
    k[i][j]=5.0*pow(10,22)*(ransuu1/((float)_L)*_A+_B);
    /*Spring constant.*/
    l[i][j]=5.0*pow(10,22)*(ransuu2/((float)_L)*_A+_B);
    /*Spring constant.*/
    m[i][j]=5.0*pow(10,22)*(ransuu3/((float)_L)*_A+_B);
    /*Spring constant.*/
    r[i][j]=1.0*pow(10,25)*(ransuu4/((float)_L)*_C+_D);
    /*Friction power.*/

```

```

    }
}

/*Arrangement of blocks.*/

for(i=0; i<=_N-1; i++){
    for(j=0; j<=_M-1; j++){
        b[i][j]=j*h;
    }
}

/*A motion of mantle.*/

for(step=0; step<=_S-1; step++){
    kaisu=0;
    energy1[step]=0;
    slip[step]=0;

    for(j=0; j<=_M-1; j++){
        for(i=0; i<=_N-1; i++){
            b[i][j]=b[i][j]+_F;
            y[i][j]=b[i][j]-j*h;
        }
    }

/*Power in which each block receives.*/

    do{
        kaisu=kaisu+1;
        hantei=0;

        for(j=0; j<=_M-1; j++){
for(i=0; i<=_N-1; i++){

```

```

    if(j==0){
        if(i==0){
            f[i][j]=(k[i][_M-1]*(y[i][j]-y[i][_M-1]))-k[i]
[j]*(y[i][j+1]-y[i][j])+l[i][j]*(y[i][j]-y[i+1][j])-1[_N-1][
j]*(y[_N-1][j]-y[i][j])+m[i][j]*y[i][j]);
        }
        else if(i==_N-1){
            f[i][j]=(k[i][_M-1]*(y[i][j]-y[i][_M-1]))-k[i]
[j]*(y[i][j+1]-y[i][j])+l[i][j]*(y[i][j]-y[0][j])-1[i-1][j]*
(y[i-1][j]-y[i][j])+m[i][j]*y[i][j]);
        }
        else{
            f[i][j]=(k[i][_M-1]*(y[i][j]-y[i][_M-1]))-k[i]
[j]*(y[i][j+1]-y[i][j])+l[i][j]*(y[i][j]-y[i+1][j])-1[i-1][j]
*(y[i-1][j]-y[i][j])+m[i][j]*y[i][j]);
        }
    }

    else if(j==_M-1){
        if(i==0){
            f[i][j]=(k[i][j-1]*(y[i][j]-y[i][j-1]))-k[i][j]
*(y[i][0]-y[i][j])+l[i][j]*(y[i][j]-y[i+1][j])-1[_N-1][j]*(
y[_N-1][j]-y[i][j])+m[i][j]*y[i][j]);
        }
        else if(i==_N-1){
            f[i][j]=(k[i][j-1]*(y[i][j]-y[i][j-1]))-k[i][j]
*(y[i][0]-y[i][j])+l[i][j]*(y[i][j]-y[0][j])-1[i-1][j]*(y[i
-1][j]-y[i][j])+m[i][j]*y[i][j]);
        }
        else{
            f[i][j]=(k[i][j-1]*(y[i][j]-y[i][j-1]))-k[i][j]
*(y[i][0]-y[i][j])+l[i][j]*(y[i][j]-y[i+1][j])-1[i-1][j]*(y
[i-1][j]-y[i][j])+m[i][j]*y[i][j]);
        }
    }
}

```

```

    else if(i==0 && j!=0 && j!=_M-1){
        f[i][j]=(k[i][j-1]*(y[i][j]-y[i][j-1])-k[i][j]*
(y[i][j+1]-y[i][j]))+l[i][j]*(y[i][j]-y[i+1][j])-l[_N-1][j]*(
y[_N-1][j]-y[i][j]))+m[i][j]*y[i][j]);
    }

    else if(i==_N-1 && j!=0 && j!=_M-1){
        f[i][j]=(k[i][j-1]*(y[i][j]-y[i][j-1])-k[i][j]*(y
[i][j+1]-y[i][j]))+l[i][j]*(y[i][j]-y[0][j])-l[i-1][j]*(y[i-1
][j]-y[i][j]))+m[i][j]*y[i][j]);
    }

    else{
        f[i][j]=(k[i][j-1]*(y[i][j]-y[i][j-1])-k[i][j]*(y
[i][j+1]-y[i][j]))+l[i][j]*(y[i][j]-y[i+1][j])-l[i-1][j]*(y[i-1
][j]-y[i][j]))+m[i][j]*y[i][j]);
    }

    c[i][j]=f[i][j];

}
}

/*The block which carries out a chain.*/

for(j=0; j<=_M-1; j++){
for(i=0; i<=_N-1; i++){

    tj=j;
    ti=i;

    if(f[ti][tj]>r[ti][tj]){

        if(tj==0){

```

```

        if(ti==0){
            s=f[ti][tj]/(k[ti][tj]+k[_N-1][_N-1]+l[ti][
tj]+l[_N-1][tj]+m[ti][tj]);
            e[ti][tj]=s*f[ti][tj]/2;
            b[ti][tj]=b[ti][tj]-s;
            y[ti][tj]=b[ti][tj]-tj*h;
            hassei[ti][tj]=kaisuu;
            if(kaisuu==1){
                o[ti][tj]=o[ti][tj]+1;
                printf("STEP=%d HASSEIBASYO=(%d,%d)\n",st
ep,ti,tj);
            }
            else{
                p[ti][tj]=p[ti][tj]+1;
                printf("STEP=%d RENSABASYO=(%d,%d)\n",ste
p,ti,tj);
            }
            c[ti][tj]=(k[ti][_M-1]*(y[ti][tj]-y[ti][_M-
1])-k[ti][tj]*(y[ti][tj]-y[ti][tj+1])+l[ti][tj]*(y[ti][tj]-y
[ti+1][tj])-l[_N-1][tj]*(y[_N-1][tj]-y[ti][tj])+m[ti][tj]*y[
ti][tj]);
        }
        else{
            s=f[ti][tj]/(k[ti][tj]+k[ti][_M-1]+l[ti][tj
]+l[ti-1][tj]+m[ti][tj]);
            e[ti][tj]=s*f[ti][tj]/2;
            b[ti][tj]=b[ti][tj]-s;
            y[ti][tj]=b[ti][tj]-tj*h;
            hassei[ti][tj]=kaisuu;
            if(kaisuu==1){
                o[ti][tj]=o[ti][tj]+1;
                printf("STEP=%d HASSEIBASYO=(%d,%d)\n",st
ep,ti,tj);
            }
            else{

```

```

    p[ti][tj]=p[ti][tj]+1;
    printf("STEP=%d RENSABASYO=(%d,%d)\n",ste
p,ti,tj);
}
if(ti==_N-1){
    c[ti][tj]=(k[ti][_M-1]*(y[ti][tj]-y[ti][_
M-1])-k[ti][tj]*(y[ti][tj+1]-y[ti][tj])+l[ti][tj]*(y[ti][tj]
-y[0][tj])-l[ti-1][tj]*(y[ti-1][tj]-y[ti][tj])+m[ti][tj]*y[t
i][tj]);
}
else{
    c[ti][tj]=(k[ti][_M-1]*(y[ti][tj]-y[ti][_
M-1])-k[ti][tj]*(y[ti][tj+1]-y[ti][tj])+l[ti][tj]*(y[ti][tj]
-y[ti+1][tj])-l[ti-1][tj]*(y[ti-1][tj]-y[ti][tj])+m[ti][tj]*
y[ti][tj]);
}
}

else if(tj==_M-1){
    if(ti==0){
        s=f[ti][tj]/(k[ti][tj]+k[ti][_M-1]+l[ti][tj
]+l[ti-1][tj]+m[ti][tj]);
        e[ti][tj]=s*f[ti][tj]/2;
        b[ti][tj]=b[ti][tj]-s;
        y[ti][tj]=b[ti][tj]-tj*h;
        hassei[ti][tj]=kaisuu;
        if(kaisuu==1){
            o[ti][tj]=o[ti][tj]+1;
            printf("STEP=%d HASSEIBASYO=(%d,%d)\n",st
ep,ti,tj);
        }
        else{
            p[ti][tj]=p[ti][tj]+1;
            printf("STEP=%d RENSABASYO=(%d,%d)\n",ste

```

```

p,ti,tj);
}
c[ti][tj]=(k[ti][tj-1]*(y[ti][tj]-y[ti][tj-
1])-k[ti][tj]*(y[ti][tj+1]-y[ti][tj])+l[ti][tj]*(y[ti][tj]-y
[ti+1][tj])-l[_N-1][tj]*(y[_N-1][tj]-y[ti][tj])+m[ti][tj]*y[
ti][tj]);
}
else{
s=f[ti][tj]/(k[ti][tj]+k[ti][tj-1]+l[ti][tj
]+l[ti-1][tj]+m[ti][tj]);
e[ti][tj]=s*f[ti][tj]/2;
b[ti][tj]=b[ti][tj]-s;
y[ti][tj]=b[ti][tj]-tj*h;
hassei[ti][tj]=kaisuu;
if(kaisuu==1){
o[ti][tj]=o[ti][tj]+1;
printf("STEP=%d HASSEIBASYO=(%d,%d)\n",st
ep,ti,tj);
}
else{
p[ti][tj]=p[ti][tj]+1;
printf("STEP=%d RENSABASYO=(%d,%d)\n",ste
p,ti,tj);
}
if(ti==_N-1){
c[ti][tj]=(k[ti][tj-1]*(y[ti][tj]-y[ti][t
j-1])-k[ti][tj]*(y[ti][0]-y[ti][tj])+l[ti][tj]*(y[ti][tj]-y[
0][tj])-l[ti-1][tj]*(y[ti-1][tj]-y[ti][tj])+m[ti][tj]*y[ti][
tj]);
}
else{
c[ti][tj]=(k[ti][tj-1]*(y[ti][tj]-y[ti][t
j-1])-k[ti][tj]*(y[ti][0]-y[ti][tj])+l[ti][tj]*(y[ti][tj]-y[
ti+1][tj])-l[ti-1][tj]*(y[ti-1][tj]-y[ti][tj])+m[ti][tj]*y[ti][
ti][tj]);
}
}

```

```

}
    }
}

else if(ti==0 && tj!=0 && tj!=_M-1){
    s=f[ti][tj]/(k[ti][tj]+k[ti][tj-1]+l[ti][tj]+
l[_N-1][tj]+m[ti][tj]);
    e[ti][tj]=s*f[ti][tj]/2;
    b[ti][tj]=b[ti][tj]-s;
    y[ti][tj]=b[ti][tj]-tj*h;
    hassei[ti][tj]=kaisuu;
    if(kaisuu==1){
o[ti][tj]=o[ti][tj]+1;
printf("STEP=%d HASSEIBASYO=(%d,%d)\n",step
,ti,tj);
    }
    else{
p[ti][tj]=p[ti][tj]+1;
printf("STEP=%d RENSABASYO=(%d,%d)\n",step,
ti,tj);
    }
    c[ti][tj]=(k[ti][tj-1]*(y[ti][tj]-y[ti][tj-1]
)-k[ti][tj]*(y[ti][tj+1]-y[ti][tj])+l[ti][tj]*(y[ti][tj]-y[t
i+1][tj])-l[_N-1][tj]*(y[_N-1][tj]-y[ti][tj])+m[ti][tj]*y[ti
][tj]);
    }

else{
    s=f[ti][tj]/(k[ti][tj]+k[ti][tj-1]+l[ti][tj]+
l[ti-1][tj]+m[ti][tj]);
    e[ti][tj]=s*f[ti][tj]/2;
    b[ti][tj]=b[ti][tj]-s;
    y[ti][tj]=b[ti][tj]-tj*h;
    hassei[ti][tj]=kaisuu;
    if(kaisuu==1){

```

```

o[ti][tj]=o[ti][tj]+1;
printf("STEP=%d HASSEIBASYO=(%d,%d)\n",step
,ti,tj);
}
else{
p[ti][tj]=p[ti][tj]+1;
printf("STEP=%d RENSABASYO=(%d,%d)\n",step,
ti,tj);
}
if(ti==_N-1 && tj!=0 && tj!=_M-1){
c[ti][tj]=(k[ti][tj-1]*(y[ti][tj]-y[ti][tj
-1])-k[ti][tj]*(y[ti][tj+1]-y[ti][tj])+l[ti][tj]*(y[ti][tj]
-y[0][tj])-l[ti-1][tj]*(y[ti-1][tj]-y[ti][tj])+m[ti][tj]
*y[ti][tj));
}
else{
c[ti][tj]=(k[ti][tj-1]*(y[ti][tj]-y[ti][tj
-1])-k[ti][tj]*(y[ti][tj+1]-y[ti][tj])+l[ti][tj]*(y[ti][tj]
-y[ti+1][tj])-l[ti-1][tj]*(y[ti-1][tj]-y[ti][tj])+m[ti][tj]
*y[ti][tj));
}
}
hantei=hantei+1;

if(kaisuu==1){
ban=ban+1;
bangou[ti][tj]=ban;
}

else{
if(j==0){
if(i==0){
if(bangou[ti+1][tj]!=0){
bangou[ti][tj]=bangou[ti+1][tj];
}
}
}
}

```

```

else if(bangou[_N-1][tj]!=0){
    bangou[ti][tj]=bangou[_N-1][tj];
}
else if(bangou[ti][tj+1]!=0){
    bangou[ti][tj]=bangou[ti][tj+1];
}
else{
    bangou[ti][tj]=bangou[ti][_M-1];
}
}
else if(i==_N-1){
    if(bangou[0][tj]!=0){
        bangou[ti][tj]=bangou[0][tj];
    }
    else if(bangou[ti-1][tj]!=0){
        bangou[ti][tj]=bangou[ti-1][tj];
    }
    else if(bangou[ti][tj+1]!=0){
        bangou[ti][tj]=bangou[ti][tj+1];
    }
    else{
        bangou[ti][tj]=bangou[ti][_M-1];
    }
}
else{
    if(bangou[ti+1][tj]!=0){
        bangou[ti][tj]=bangou[ti+1][tj];
    }
    else if(bangou[ti-1][tj]!=0){
        bangou[ti][tj]=bangou[ti-1][tj];
    }
    else if(bangou[ti][tj+1]!=0){
        bangou[ti][tj]=bangou[ti][tj+1];
    }
    else{

```

```

        bangou[ti][tj]=bangou[ti][_M-1];
    }
}

        }

        else if(j==_M-1){
if(i==0){
    if(bangou[ti+1][tj]!=0){
        bangou[ti][tj]=bangou[ti+1][tj];
    }
    else if(bangou[_N-1][tj]!=0){
        bangou[ti][tj]=bangou[_N-1][tj];
    }
    else if(bangou[ti][0]!=0){
        bangou[ti][tj]=bangou[ti][0];
    }
    else{
        bangou[ti][tj]=bangou[ti][tj-1];
    }
}
else if(i==_N-1){
    if(bangou[0][tj]!=0){
        bangou[ti][tj]=bangou[0][tj];
    }
    else if(bangou[ti-1][tj]!=0){
        bangou[ti][tj]=bangou[ti-1][tj];
    }
    else if(bangou[ti][0]!=0){
        bangou[ti][tj]=bangou[ti][0];
    }
    else{
        bangou[ti][tj]=bangou[ti][tj-1];
    }
}
else{

```

```

    if(bangou[ti+1][tj]!=0){
        bangou[ti][tj]=bangou[ti+1][tj];
    }
    else if(bangou[ti-1][tj]!=0){
        bangou[ti][tj]=bangou[ti-1][tj];
    }
    else if(bangou[ti][0]!=0){
        bangou[ti][tj]=bangou[ti][0];
    }
    else{
        bangou[ti][tj]=bangou[ti][tj-1];
    }
}

        else if(i==0 && j!=0 && j!=_M-1){
if(bangou[ti+1][tj]!=0){
    bangou[ti][tj]=bangou[ti+1][tj];
}
else if(bangou[_N-1][tj]!=0){
    bangou[ti][tj]=bangou[_N-1][tj];
}
else if(bangou[ti][tj+1]!=0){
    bangou[ti][tj]=bangou[ti][tj+1];
}
else{
    bangou[ti][tj]=bangou[ti][tj-1];
}

        else if(i==_N-1 && j!=0 && j!=_M-1){
if(bangou[0][tj]!=0){
    bangou[ti][tj]=bangou[0][tj];
}
else if(bangou[ti-1][tj]!=0){

```

```

    bangou[ti][tj]=bangou[ti-1][tj];
}
else if(bangou[ti][tj+1]!=0){
    bangou[ti][tj]=bangou[ti][tj+1];
}
else{
    bangou[ti][tj]=bangou[ti][tj-1];
}
    }

    else{
if(bangou[ti+1][tj]!=0){
    bangou[ti][tj]=bangou[ti+1][tj];
}
else if(bangou[ti-1][tj]!=0){
    bangou[ti][tj]=bangou[ti-1][tj];
}
else if(bangou[ti][tj+1]!=0){
    bangou[ti][tj]=bangou[ti][tj+1];
}
else{
    bangou[ti][tj]=bangou[ti][tj-1];
}
    }
    }
    ransuu5=(int)(sprng()*(float)_L);
/*System size*/
    r[ti][tj]=1.0*pow(10,25)*(ransuu5/((float)_L)*
_C+_D); /*Friction power*/
    }
    f[ti][tj]=c[ti][tj];

}
    }
}while(hantei!=0);

```

```

        for(j=0; j<=_M-1; j++){
            for(i=0; i<=_N-1; i++){
hindo=hindo+o[i][j];
                drawn=drawn+p[i][j];
                slip[step]=slip[step]+o[i][j]+p[i][j];
energy1[step]=energy1[step]+e[i][j];
            }
        }

        if(hindo!=0){
            printf("STEP=%d,HINDO=%d,DRAWN=%d,ENERGY=%f\n\n",step
,hindo,drawn,energy1[step]);
        }

        fprintf(OUTPUT1,"%d %d\n",step,drawn);
        fprintf(OUTPUT5,"%d %f\n",step,energy1[step]);

        for(j=0; j<=_M-1; j++){
            for(i=0; i<=_N-1; i++){
if(step==_E){
                if(hassei[i][j]!=0 && bangou[i][j]!=0){
                    fprintf(OUTPUT2,"%d %d %d\n",i,j,hassei[i][j]);
                }
            }
        }

        for(n=1; n<=kaisuu; n++){
            for(j=0; j<=_M-1; j++){
for(i=0; i<=_N-1; i++){
                if(bangou[i][j]==n){
                    energy2[step][n]=energy2[step][n]+e[i][j];
                    katamari[step][n]=katamari[step][n]+1;
                }
            }
        }
    }

```

```

}
    }
}

    for(j=0; j<=_M-1; j++){
        for(i=0; i<=_N-1; i++){
o[i][j]=0;
p[i][j]=0;
        e[i][j]=0;
hassei[i][j]=0;
bangou[i][j]=0;
        }
    }

    ban=0;
    drawn=0;
    hindo=0;

}

    for(a=1; a<=_H; a++){
        for(step=_U; step<=_S-1; step++){
            for(n=1; n<=_T-1; n++){
if(katamari[step][n]>=a){
                kaisuu2=kaisuu2+1;
}
            }
        }
        if(kaisuu2!=0){
            fprintf(OUTPUT3,"%f %d\n",a,kaisuu2);
        }
        kaisuu2=0;
    }

    for(u=1; u<=10000; u++){

```

```

    a=u*pow(10,26);

    for(step=_U; step<=_S-1; step++){
        for(n=1; n<=_T-1; n++){
if(energy2[step][n]>a){
    kaisu1=kaisu1+1;
}
        }
    }
    if(kaisu1!=0){
        fprintf(OUTPUT4,"%f %d\n",a,kaisu1);
    }
    kaisu1=0;
}

fclose(OUTPUT1);
fclose(OUTPUT2);
fclose(OUTPUT3);
fclose(OUTPUT4);
fclose(OUTPUT5);

}

```

## 謝辞

本研究を行うにあたって、羽田野先生をはじめ研究室の先輩方には大変お世話になりました。羽田野先生や先輩方からの御指導、アドバイスのおかげで研究をここまで進めることができました。また卒研生とは互いの研究内容を話合うなど、良い刺激を受けながら研究に取り組むことができました。本当にありがとうございました。