

# 汚染物質の拡散における べき乗則

和気正季

羽田野研究室

2001年2月28日

## 概要

岩石中の汚染物質の浸透の様子を調べる実験が行われている。その実験結果は、一般的な輸送のモデルによく使われる移流拡散方程式やランダムウォークでは説明がつかない。これはイオンを吸着する電位の力が吸着部位によって異なり、その場所に吸着される時間がランダムになっているためと考えられる。そこで、吸着時間の分布関数を、平均値を持たないべき乗の関数として考えて、新たなモデルを定義した。そして、べき乗の指数を変化させていくことで、実験値に近いシミュレーション結果を求めた。

# 目次

1	カラム試験	3
2	新しいモデル	5
2.1	吸着のモデル化	5
2.2	吸着時間の平均値とゆらぎ	8
2.3	プログラム	9
3	シミュレーション結果	11
3.1	$\alpha = 2$ の場合と吸着時間一定の場合	11
3.2	$\alpha$ の値を変化させた場合	11
4	まとめ	18
5	謝辞	19
A	吸着のシミュレーションのプログラム	20

# 1 カラム試験

汚染物質の拡散を見る実験として、次のような実験が行われている [1]。図 1 のような直径 2.5 cm、高さ 20 cm の円柱状の入れ物の中に砕いた花崗岩を詰め、そこに水と放射性同位元素を通す。その時の、花崗岩表面におけるイオンの吸着によって、汚染物質の拡散を見ていくものである。

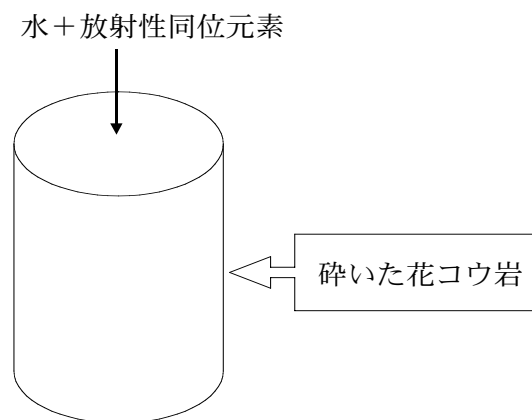


図 1: 汚染物質の拡散の様子を調べるカラム試験の概要

この円柱を 20 個の輪切りにして考え、円柱の入り口からの距離を横軸、輪切りにされた円柱内に含まれる汚染物質の密度を縦軸にとると、拡散の様子を表すグラフが得られる。その結果、図 2 のダイヤ印の値が得られた [1]。

図 2 は、一次元移流拡散方程式

$$\frac{\partial}{\partial t}C(x, t) = D \frac{\partial^2}{\partial x^2}C(x, t) - v \frac{\partial}{\partial x}C(x, t) \quad (1)$$

( $C(x, t)$ : 放射性同位元素の密度  $D$ : 散乱係数  $v$ : 流速) によって求められた理論値との比較を示している [2]。なめらかな曲線と点線で描かれているのが、理論計算によるグラフである。これを見ると、実験値のプロットがピークから長い尾を引いているのに対し、シミュレーションによるグラフはピークからすぐに落ち込んでしまっているのがわかる。

そこで我々は、実験値の様な長い尾を引くグラフをシミュレーションによって得るために、新たなモデルを定義する。このモデルでは、イオンが吸着されながら拡散していく。モデルの新しい点は、イオンの吸着時間が平均値の存在しない形で、ランダムに分布している点である。本研究では、平均値の存在しない場合には、図 2 の実験値のプロットの様に見える長い尾を引く密度分布が得られることをシミュレーションで示すものである。

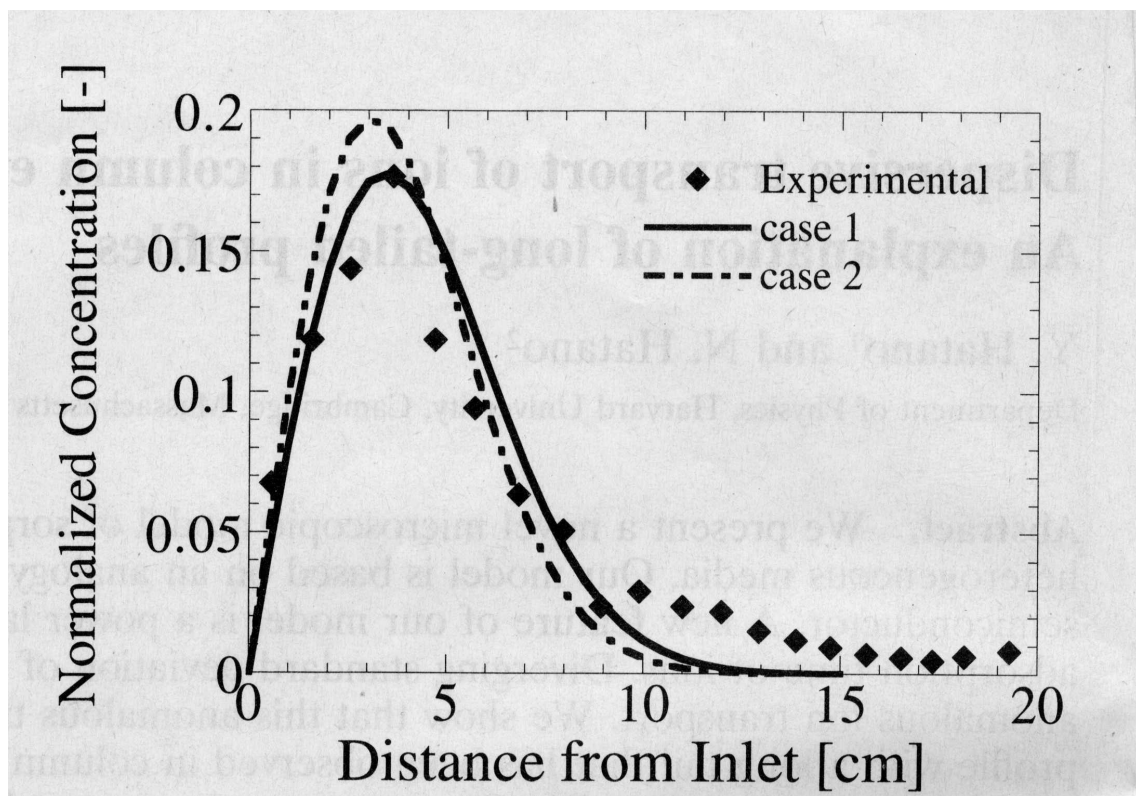


図 2: 実験による近似値と理論値のシミュレーションとの比較 [2]。  
 横軸は円柱の入り口からの距離、縦軸は輪切りにされた円柱内に含まれる汚染物質の密度である。  
 (Dispersive transport of ions *in* column experiment: Y.Hatano and N.Hatano より)

## 2 新しいモデル

### 2.1 吸着のモデル化

円柱内の花崗岩の様子を、図3のようにモデル化する。花崗岩の表面にはたくさんの吸着サイトがあり、イオンは、そのサイトの吸着電位によって捕らえられていると考えられる。花崗岩の表面は様々な形をしているので、それぞれのサイトにおける吸着電位は、自ずと他のサイトとは異なってくる。もし吸着電位が強ければ、そのサイトにおける吸着時間は長くなり、弱ければ短くなる。つまり、吸着時間は吸着電位の違いによってランダムに決定されているということになる。

そこで新しいモデルでは、吸着時間をランダムに与える。ただし、平均値を持たない様な分布をさせたい。というのは、平均値があると長時間の振舞いは、平均吸着時間で説明できてしまうと予想されるからである。そこで、吸着時間の分布関数に、べき乗の関数を導入する。吸着時間を  $\tau$ 、吸着時間を決定する確率を  $\Psi(\tau)$  とし、吸着時間の分布関数を、

$$\Psi(\tau) = C\tau^{-\alpha} \quad (2)$$

とおく。なお、簡単のため  $\tau$  は正の整数とする。ここで、この確率の和が1となるように規格化を行う。

$$\sum_{\tau=1}^{\infty} C\tau^{-\alpha} = 1 \quad (3)$$

式(3)において、 $\sum_{\tau=1}^{\infty} \tau^{-\alpha} = \zeta(\alpha)$  (リーマンのツェータ関数) [3] となるので、規格化定数  $C$  は以下のようなになる。

$$C = \frac{1}{\zeta(\alpha)} \quad (4)$$

このようにして決定した吸着時間の間、サイトにとどまっていたイオンは、熱の作用や水の流れによって再び拡散し、ある距離を進んで他のサイトに吸着される。本来、1つのサイトと次のサイトとの間の距離は、それぞれの場所においてバラバラなのであるが、これをすべて距離一定とすることにより、実験の円柱を1次元の格子としてモデル化する。また時間は離散化して考えるものとする。

こうして考えたモデルの定義をまとめると以下のようなになる。

1. どのイオンも最初は  $X = 0, t = 0$  に位置する。
2. それぞれのイオンは、分布関数  $\Psi(\tau)$  によって吸着時間  $\tau$  を決定し、 $\tau$  の期間サイト  $X$  に吸着される。
3. 吸着時間終了後、イオンは90%の割合で前に進み、10%の割合で後ろへ戻るとする。

4. 移動先のサイトでも同様に、上記の作業を繰り返しながら拡散していく。

こうしてイオンは図4の様な道筋をたどりながら運動する。

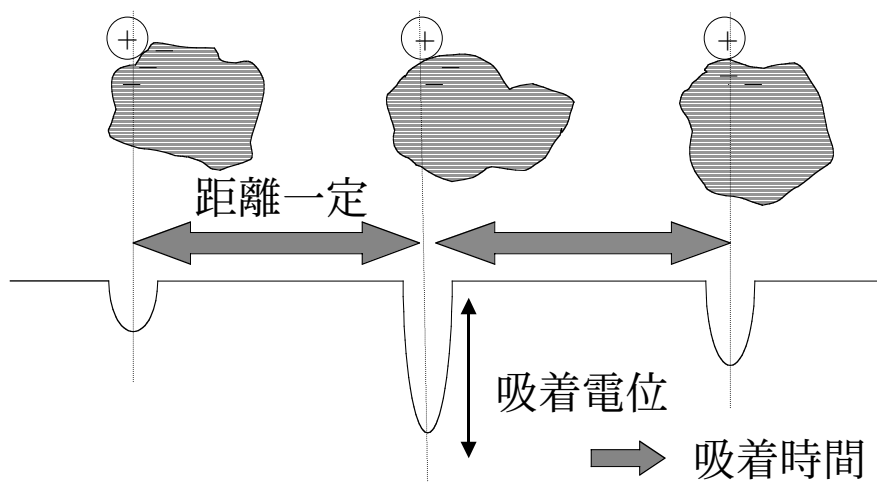


図 3: 円柱内における花崗岩の様子モデル化。

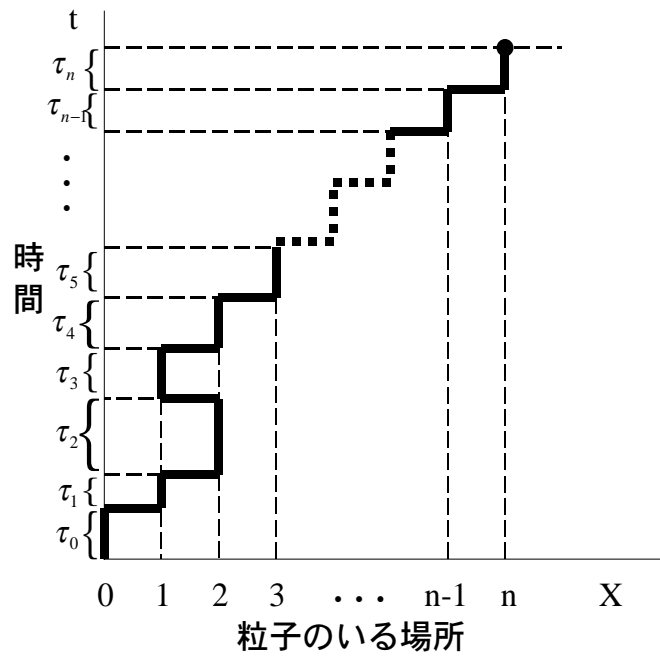


図 4: モデルの定義の概略図。

## 2.2 吸着時間の平均値とゆらぎ

ここでは、吸着時間の分布関数に、べき乗の関数を導入した意義について触れておきたい。重要なことは、指数の違いによる、平均値とゆらぎの振る舞いの変化である。

吸着時間の平均値は以下のように表される。

$$\langle \tau \rangle = \sum_{\tau=1}^{\infty} \tau \Psi(\tau) \quad (5)$$

$$= \frac{1}{\zeta(\alpha)} \sum_{\tau=1}^{\infty} \tau^{1-\alpha} \quad (6)$$

したがって

$$\begin{aligned} \alpha \leq 2 \text{ のとき} & \quad \langle \tau \rangle = \infty \\ \alpha > 2 \text{ のとき} & \quad \langle \tau \rangle = \frac{\zeta(\alpha-1)}{\zeta(\alpha)} \end{aligned} \quad (7)$$

となる。また吸着時間のゆらぎは以下のように表される。

$$\langle \tau^2 \rangle = \sum_{\tau=1}^{\infty} \tau^2 \Psi(\tau) \quad (8)$$

$$= \frac{1}{\zeta(\alpha)} \sum_{\tau=1}^{\infty} \tau^{2-\alpha} \quad (9)$$

したがって

$$\begin{aligned} \alpha \leq 3 \text{ のとき} & \quad \langle \tau^2 \rangle = \infty \\ \alpha > 3 \text{ のとき} & \quad \langle \tau^2 \rangle = \frac{\zeta(\alpha-2)}{\zeta(\alpha)} \end{aligned} \quad (10)$$

である。これによると、吸着時間の平均値は、 $\alpha \leq 2$ の時に発散し、 $\alpha > 2$ の時に有限の値が存在している。また、吸着時間のゆらぎは、 $\alpha \leq 3$ の時に発散し、 $\alpha > 3$ の時に存在している。

このことから、指数 $\alpha$ の違いによって、吸着時間が平均値を持つか持たないか、また、ゆらぎの様子がどのようなかが決まり、イオンの密度分布に影響を与えることが予想される。そこで、吸着時間の分布関数の指数に、様々な値を代入してシミュレーションを試みることにする。



## 2.3 プログラム

次に、前項で定義したモデルによって、上手くシミュレーションが出来るようなプログラムを考えた。使用したプログラムは、20 ページ以降に示す。ここでは、流れ図 (図4) にそってプログラムの概要の簡単な説明を行う。

1. 粒子数の最大値  $N_{particle_{max}}$  を 500、粒子の移動回数  $N_{step_{max}}$  を 50000、粒子の到着地点の最大値  $Length_{max}$  を 1000 と定義する。
2. 指数  $\alpha$  と規格化常数に含まれる  $\zeta(\alpha)$  を代入する。  
 $\alpha$  の値に伴う  $\zeta(\alpha)$  の値は以下ようになる。

$\alpha$	$\zeta(\alpha)$	
1.7	2.054288757	
2.0	1.644934067	(11)
2.5	1.341487257	
3.0	1.202056903	

3. 粒子が後ろへ戻る確率  $P_{backward}$  を 0.1、粒子数  $N_{particle}$  を 500、粒子の総移動時間  $MaxTime$  を 700、粒子の移動した場所をグラフ化する時間間隔  $DeltaTime$  を 20 として、代入する。
4. すべての粒子の  $x = 0$  における出発時刻  $departuretime$  を計算する。  
WaitingTime 関数 (後述) によって  $x = 0$  における待ち時間を計算する。最初の位置では、それが出発時刻となる。
5. すべての粒子の、2 番目以降の場所における出発時刻  $departuretime$  を計算する。  
2 番目以降の場所における出発時刻は、1 つ前の場所における出発時刻に、その場所の待ち時間をたしたものになる。次の場所は、*hopping* 関数 (後述) によって決まる。
6. ヒストグラムの計算をする。  
離散化された時間 20 毎に、粒子のいる場所を横軸、その場所にいる粒子の数を縦軸とするように、計算させた。
7. 待ち時間をきめる *WaitingTime* 関数を作る。  
乱数によって  $P_{wait}$  を選び、 $P_{wait}$  が、 $P_{sum}$  より大きくなるまで、待ち時間  $T_{wait}$  を足し続ける、という方法で待ち時間を決定する。 $\tau = 一定$  とする時は、 $T_{wait} = 一定値$  と書き直した。
8. 次の場所を決定する *hopping* 関数を作る。  
乱数によって  $P_{move}$  を選び、それが  $P_{backward}$  の 0.1 より小さければ場所を 1 つ減らし、大きければ 1 つ増やす、という方法で場所を決定した。

9. 粒子に、次の場所への移動を指示する。  
すべての粒子に対して、離散化している時間が、出発時刻より大きくなったら、次の場所へ移動するようにした。

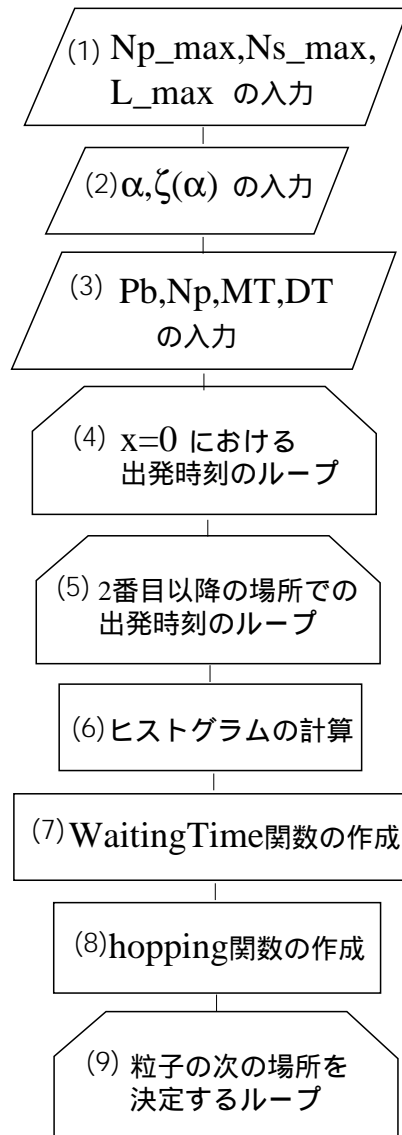


図 5: プログラムの概要の流れ図。

### 3 シミュレーション結果

#### 3.1 $\alpha = 2$ の場合と吸着時間一定の場合

まず、吸着時間の分布関数 (2) におけるべき乗の指数を  $\alpha = 2$  として、シミュレーションさせた。図 6 がその結果を示したグラフである。これを見ると、グラフがある程度、ピークから長い尾を引いている様子がわかる。一方、吸着時間を  $\tau = 2$  で一定とするシミュレーションを行うと、図 7 のようなグラフが得られる。図 6 とこれを比較すると、その振る舞いは明らかに違っていることがわかる。

このことから、吸着時間の平均値、ゆらぎが発散している程、グラフはピークから長い尾を引き、収束している程、吸着時間を一定としたグラフの様な振る舞いをする事が予想される。

#### 3.2 $\alpha$ の値を変化させた場合

そこで、

1. 平均値、ゆらぎ共に発散している場合として、 $\alpha = 1.7$
2. 平均値は存在するが、ゆらぎが発散している場合として、 $\alpha = 2.5$
3. 平均値、ゆらぎ共に存在している場合として、 $\alpha = 3$

におけるシミュレーションを試みた。これらの結果を図 8 から図 10 に示す。

グラフを比較すると、先程の予想の通り、 $\alpha$  が小さい程ピークから長い尾を引くグラフとなり、 $\alpha$  が大きい程、吸着時間を一定としたグラフの様な振る舞いをしていく様子があらわれている。

また、式 (9) において  $\alpha = 3$  における平均値を計算し、その平均値と等しい値に  $\tau$  を固定したシミュレーションを行うと、図 11 のようなグラフが得られる。これを  $\alpha = 3$  におけるシミュレーションの図 10 と比較すると、ほとんど同じ振る舞いをしていくことがわかる。つまり、 $\langle \tau \rangle$  と  $\langle \tau^2 \rangle$  が存在する場合は、吸着時間を一定とした場合で近似できる。逆に、それと異なる振る舞いを得るためには、 $\langle \tau \rangle$  や  $\langle \tau^2 \rangle$  が発散するモデルを使わなければならない。

以上の結果から、指数  $\alpha$  の違いによって、吸着時間が平均値を持つか持たないか、また、ゆらぎの様子がどのようなものであるかが決まり、イオンの密度分布に影響を与えるであろうという予測が、見事に示されたと考えるものである。

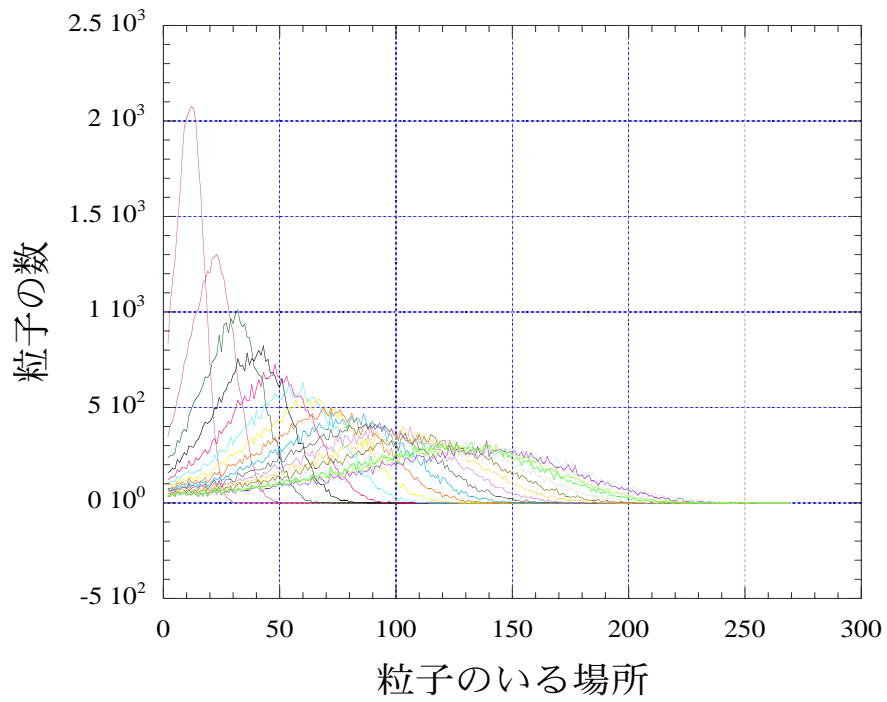


図 6:  $\Psi(\tau) \propto \tau^{-2}$  によって吸着時間を決定した場合の時間 40 ごとの粒子の分布。

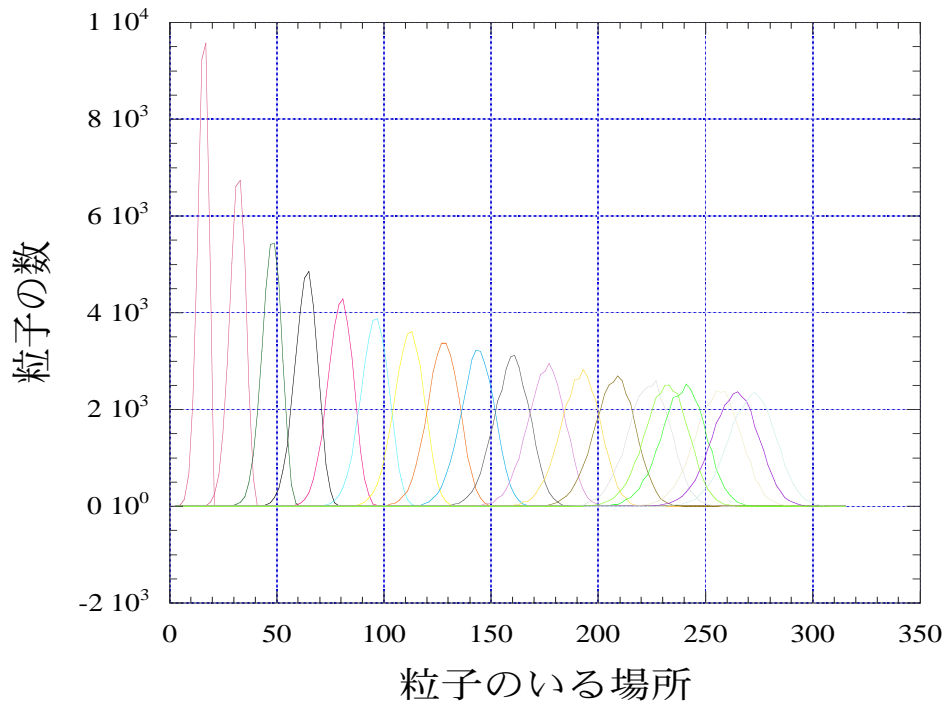


図 7: 吸着時間を  $\tau = 2$  で一定とした場合の時間 40 ごとの粒子の分布。

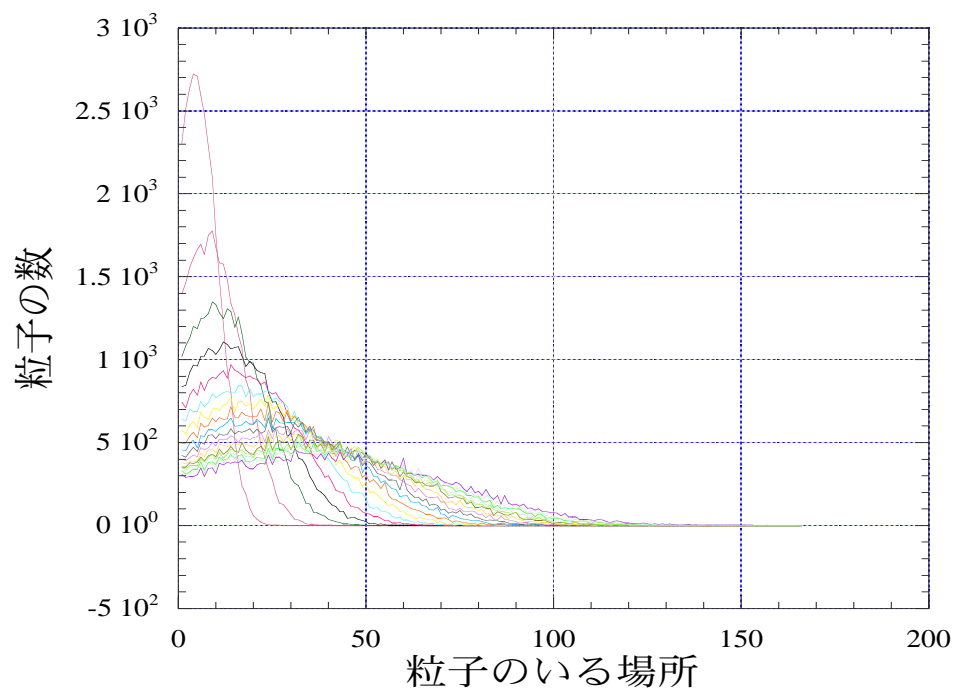


図 8:  $\alpha = 1.7$  とした場合の時間 40 ごとの粒子の分布。

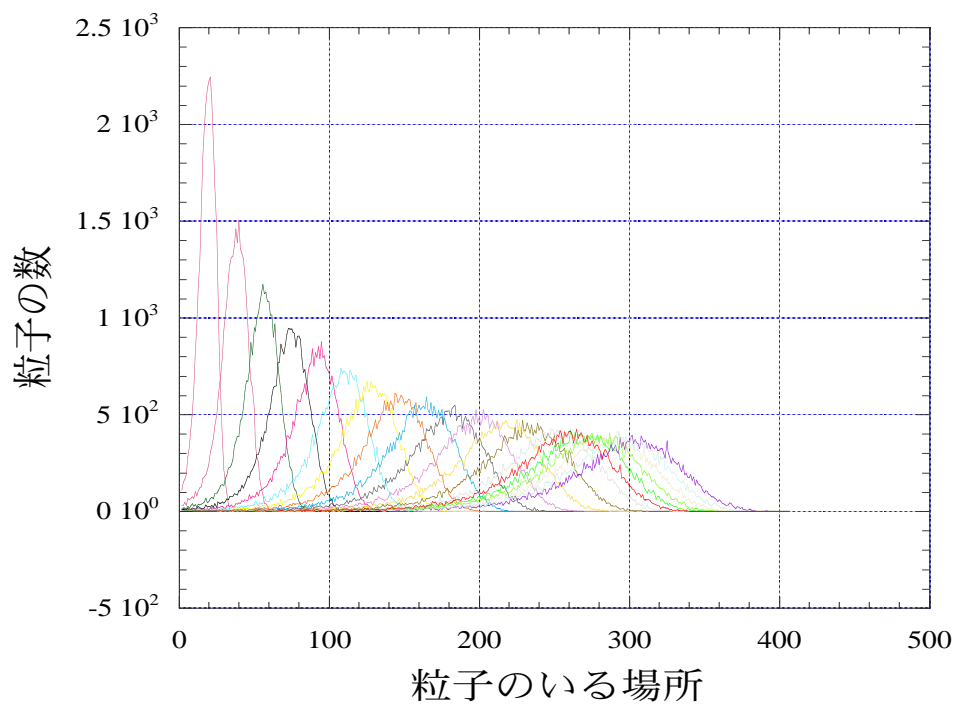


図 9:  $\alpha = 2.5$  とした場合の時間 40 ごとの粒子の分布。

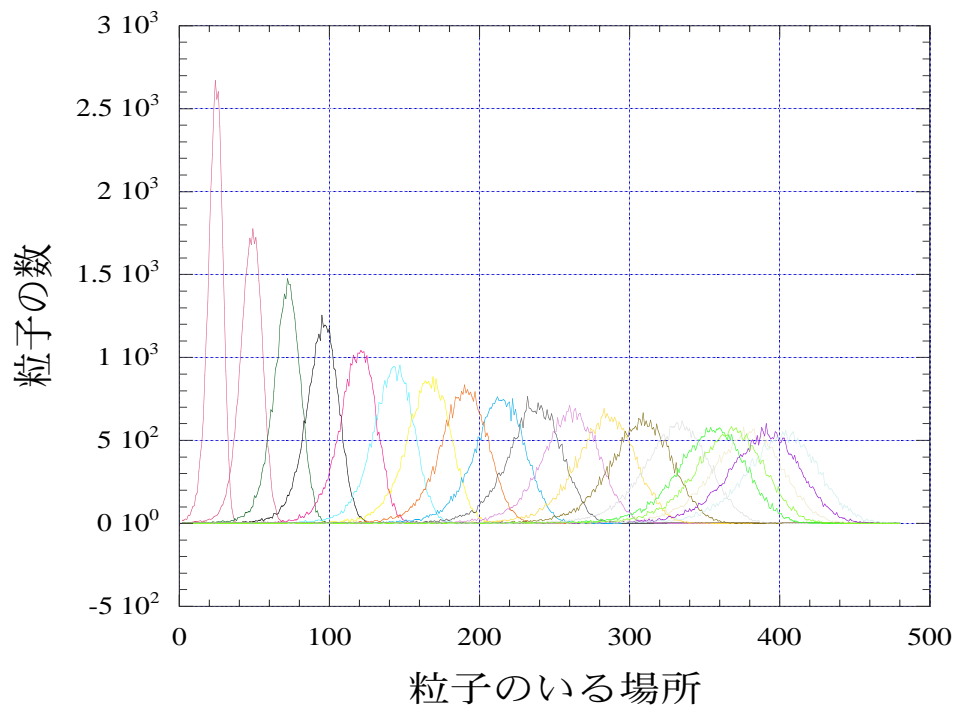


図 10:  $\alpha = 3$  とした場合の時間 40 ごとの粒子の分布。



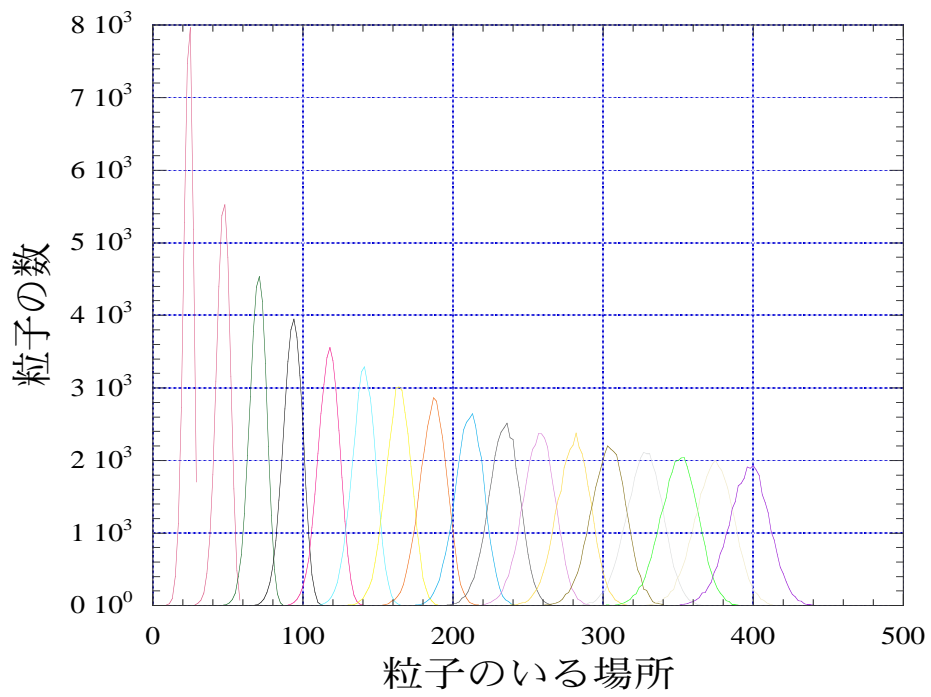


図 11: 吸着時間を  $\alpha = 3$  における平均値と等しい一定値にした場合の時間 40 ごとの粒子の分布。

## 4 まとめ

最後に、実験による実測の近似値のグラフと、 $\alpha = 1.7$ とした場合の、ある時間におけるシミュレーション結果の一つを取り出して比較する。結果のグラフを以下の図 12 に示す。これを見ると、ぴったり一致しているとは言えないまでも、ピークから長い尾を引いている様子は、再現することが出来たと思われる。

今回の結果から、吸着時間の分布関数にべき乗の関数をとることで、カラム試験のデータの振る舞いが再現出来たと考える。なお、理論的に、吸着電位の分布がポアソン分布をする場合に、吸着時間の分布がべき乗則に従う、ということが知られている [2]。今回の結果は、そのような分布を示唆していると思われる。

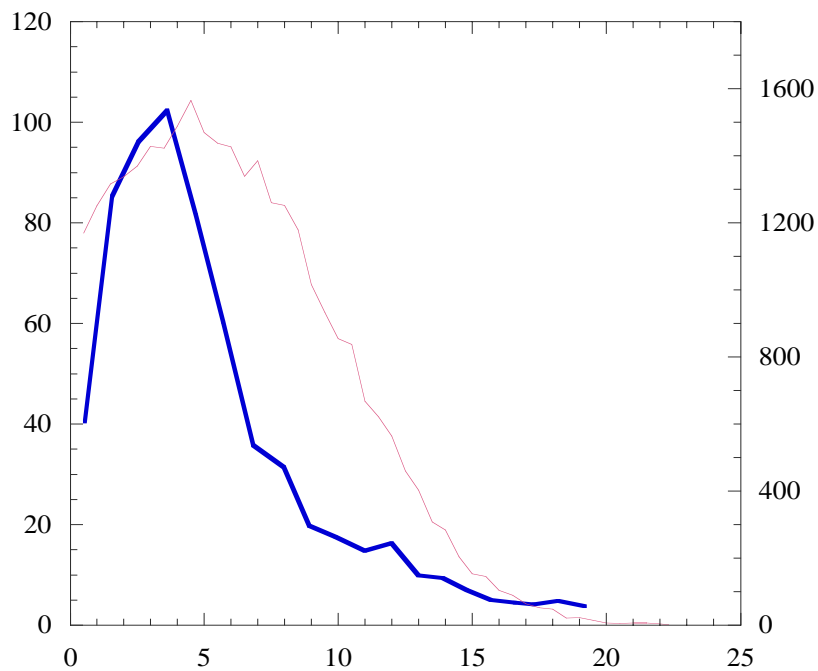


図 12: 実験値 (青線) と  $\alpha = 1.7$  におけるシミュレーション結果 (赤線) との比較。

## 参考文献

- [1] S.Nakayama,T.T.Vandergraaf,and M. Kumata : Experimental study on nuclides migration under deep geological condition,Radioactive Waste Research,1(1994),p.67-76
- [2] Y.Hatano and N.Hatano : Dispersive transport of ions in column experiment,Water Resources Research,34(1998),p.1027-1033
- [3] 森口繁一・宇田川 久・一松信 : 岩波数学公式 II 級数・フーリエ解析 (岩波書店) p.39

## 5 謝辞

本年度卒業研究をするにあたり、私事ではありますが、様々なことがありました。卒業研究の継続を許可して下さった、物理学科の全先生方に、厚く御礼申し上げます。

勉強不足の私に、いつも丁寧に関わりやすく指導して下さった担当教官の羽田野直道先生には、大変感謝致しております。

同研究室の仲間達には、勉学の上でも、それ以外の部分でも、本当に暖かく支えられました。そのおかげで、なんとか1年間頑張ることが出来たと思います。ありがとうございました。

## APPENDIX

### A 吸着のシミュレーションのプログラム

```
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#define _Nparticle_max 500
#define _Nstep_max 50000
#define _Length_max 1000

#define _Alpha 2.0
#define _Zeta_Alpha (1.644934067)

double WaitingTime();
int hopping(int position, double Pbackward);
void histogram(int position[][_Nstep_max+1],
              int departuretime[][_Nstep_max],
              int hist[],
              int Nparticle,
              int Time);

int main(argc,argv)
int argc;
char *argv[];
{

    static int position[_Nparticle_max][_Nstep_max+1],
              departuretime[_Nparticle_max][_Nstep_max],
              hist[_Length_max],totalhist[_Length_max];
    int Nparticle,MaxTime,DeltaTime,Time,Istep,Iparticle,x;
    double Pbackward;
    char filename[80];
    FILE *OUTPUT;
    int nproc,myid;
```

```

MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&nproc);
MPI_Comm_rank(MPI_COMM_WORLD,&myid);

Pbackward=0.1;
Nparticle=500;
MaxTime=700;
DeltaTime=20;
strcpy(filename,"simoutput.dat");

    srand((unsigned)myid);

/* Calculates the departure time from x=0 of each particle */
for(Iparticle=0; Iparticle<Nparticle; Iparticle++)
    {
        departuretime[Iparticle][0]=WaitingTime();
        position[Iparticle][0]=0;
        position[Iparticle][1]=1;
    }

/* Calculates the departure time and the next position of each particle */
for(Istep=1; Istep<_Nstep_max; Istep++)
    {
        for(Iparticle=0; Iparticle<Nparticle; Iparticle++)
        {
            departuretime[Iparticle][Istep]
                =departuretime[Iparticle][Istep-1]+WaitingTime();

            position[Iparticle][Istep+1]
                =hopping(position[Iparticle][Istep],Pbackward);
        }
    }

/* Calculates the histogram */

if(myid==0)
    {
        OUTPUT=fopen(filename,"w");

```

```

    }

for(Time=DeltaTime; Time<MaxTime ;Time+=DeltaTime)
    {

        histogram(position,departurette,hist,Nparticle,Time);

        MPI_Reduce(hist,totalhist,_Length_max,MPI_INT,MPI_SUM,0,MPI_COMM_WORLD);

        if(myid==0)
    {
        for(x=0; x<_Length_max; x++)
    {
        if(x==0||totalhist[x]==0)
            {
                fprintf(OUTPUT,"# %d %d\n",x,totalhist[x]);
            }
        else
            {
                fprintf(OUTPUT,"%d %d\n",x,totalhist[x]);
            }
        }
        fprintf(OUTPUT,"&\n");
    }
        }
        if(myid==0)
            {
                fclose(OUTPUT);
            }

        MPI_Finalize();
    }

double WaitingTime()
{
/*
    Chooses a waiting time of a particle randomly,
    according to the probability
     $P(t)=t^{-\alpha} / \zeta(\alpha)$ 
*/
}

```

```

*/

double Pwait,Psum=0.0;
int Twait=0;

Pwait=(double)rand()/RAND_MAX*_Zeta_Alpha;

do
{
    Twait++;
    Psum+=pow(Twait,-_Alpha);
}
while(Psum<Pwait);

return Twait;
}

int hopping(int position, double Pbackward)
{
/*
    Calculates the position at the next step.
*/

double Pmove;

Pmove=(double)rand()/RAND_MAX;

if(Pmove<Pbackward && position>0)
{
    return position-1;
}
else
{
    return position+1;
}
}

```

```

void histogram(int position[] [_Nstep_max+1],
              int departuretime[] [_Nstep_max],
              int hist[],
              int Nparticle,
              int Time)
{
/*
  Calculates the particle distribution from the movement of each particle
*/

  int x,Iparticle,Istep;

  for(x=0; x<_Length_max; x++)
  {
    hist[x]=0;
  }

  for(Iparticle=0; Iparticle<Nparticle; Iparticle++)
  {
    Istep=0;
    while(departuretime[Iparticle][Istep]<Time)
  {
    Istep++;
    if(Istep==_Nstep_max)
    {
      printf("Exhausted Nstep. Quitting.\n");
      MPI_Finalize();
      exit(1);
    }
  }

    if(position[Iparticle][Istep]>=_Length_max)
  {
    printf("Exhausted Length. Quitting.\n");
    MPI_Finalize();
    exit(1);
  }
}

```



```
        hist[ position[Iparticle][Istep] ]++;  
    }  
}
```