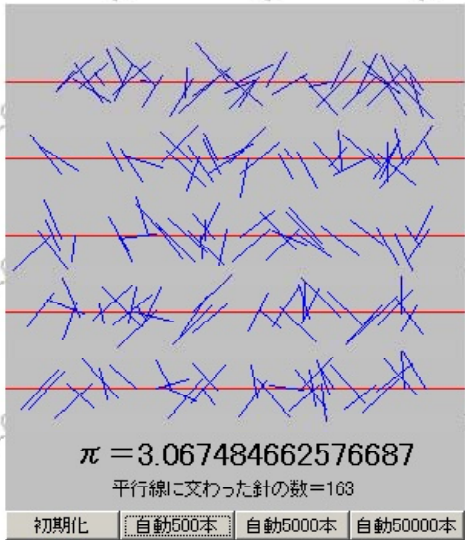
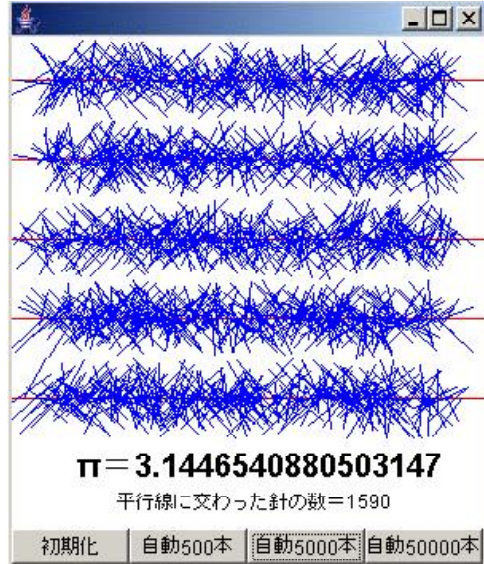


# 【ビュホンの針】

## 1. はじめに



[ J a v a アプレット ]



[ J a v a アプリケーション ]

ビュホン (Buffon 1707-1788) は、針を投げて円周率  $\pi$  を求めることを考えました。

平面上に、幅  $2a$  の間隔で、平行線を無数に引いておきます。

この平面上に長さ  $2b$  の針を落とすと、この針が平行線と交わる確立  $p$  は、

$p = (2b) \div (a\pi) \cdots ①$  となります。ただし、 $b < a$  とします。

平面上にたくさんの針を落としたとき、平行線に交わった針の本数の相対度数を  $r$  とすると、

$r \div p \cdots ②$  となります。

①、②より、 $\pi \div (2b) \div (a r) \cdots ③$  となります。

③より、相対度数  $r$  を実験で求めれば、円周率  $\pi$  の近似値が分かるというわけです。

シミュレーションソフト「ビュホンの針」を使って、円周率  $\pi$  の近似値を求めてみてください。

## 2. J a v a アプレット

### (1) J a v a プログラムリスト

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                                 //
//              「ビュホンの針」                                     //
//              (針をまいて円周率πを求める!!)                       //
//              Copyright (C) K.Niwa 2001.11. 6                       //
//              (J a v a アプレット)                                 //
//                                                                 //
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;
```

```
public class Buffon extends Applet{
```

```
    Button[] myBtn;           //ボタン型として宣言する
    Panel myPanel;           //パネル型として宣言する
    int flag=0;              //変数宣言
    double x;
```

```

int N; //針の本数
int count; //ループカウンタ

//public void init() init メソッド (void は値を返さないメソッドの意) *****
public void init() {
    setBackground(Color.lightGray);
    myBtn=new Button[4]; //ボタンの実体化
    myBtn[0]=new Button("初期化");
    myBtn[1]=new Button("自動 500 本");
    myBtn[2]=new Button("自動 5000 本");
    myBtn[3]=new Button("自動 50000 本");

    myPanel=new Panel(); //パネルの実体化
    myPanel.setLayout(new GridLayout(1,4));
    for (count=0;count<=3;count++) {
        myPanel.add(myBtn[count]); //パネルにボタンを貼り付ける
    }
    setLayout(new BorderLayout());
    add("South",myPanel); //パネルを南に貼り付ける

    //初期化ボタンを押したときのイベント処理
    myBtn[0].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            repaint();
        }
    });

    //自動 500 本ボタンを押したときのイベント処理
    myBtn[1].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=1;
            N=500;
            repaint();
        }
    });

    //自動 5000 本ボタンを押したときのイベント処理
    myBtn[2].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=2;
            N=5000;
            repaint();
        }
    });

    //自動 50000 本ボタンを押したときのイベント処理
    myBtn[3].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=3;
            N=50000;
            repaint();
        }
    });

} //public void init()

//public void paint(Graphics g) //paint メソッド *****
public void paint(Graphics g) { //paint メソッド

    //初期化ボタンを押したとき
    if (flag==0) {
        int i,sum=0,px1,py1,px2,py2,yy; //変数宣言と変数初期化
        g.clearRect(0,0,300,350); //全体のクリア
    }
}

```

```

g.drawString("Copyright (C) K.Niwa 2001.11",80,20);
for (yy=50;yy<=250;yy=yy+50) {
    g.setColor(Color.red);
    g.drawLine(0,yy,300,yy);           //平行線の描写
    g.setColor(Color.black);
}
} //if (flag==0)

//自動ボタンを押したとき
else if (flag==1 || flag==2 || flag==3) {
    double x1,y1,x2,y2,pai,t;           //変数宣言
    int i,sum=0,px1,py1,px2,py2,yy;    //変数宣言と変数初期化
    g.clearRect(0,0,300,350);         //全体のクリア
    for (yy=50;yy<=250;yy=yy+50) {
        g.setColor(Color.red);
        g.drawLine(0,yy,300,yy);     //平行線の描写
        g.setColor(Color.black);
    }

//針のばらまき開始-
g.setColor(Color.blue);

for (i=1;i<=N;i++) {
    for (t=0;t<5000;t++) {           //時間稼ぎ
        }
        x1=25+250*Math.random();
        y1=25+250*Math.random();
        x2=x1+25*Math.cos(2*3.14159263*Math.random());
        y2=y1+25*Math.sin(2*3.14159263*Math.random());
        px1=(int)(x1);
        py1=(int)(y1);
        px2=(int)(x2);
        py2=(int)(y2);
        if (py1>py2) {
            if (py1>50 && py2<50) {
                sum++;
                g.drawLine(px1,py1,px2,py2);
            }
            else if (py1>100 && py2<100) {
                sum++;
                g.drawLine(px1,py1,px2,py2);
            }
            else if (py1>150 && py2<150) {
                sum++;
                g.drawLine(px1,py1,px2,py2);
            }
            else if (py1>200 && py2<200) {
                sum++;
                g.drawLine(px1,py1,px2,py2);
            }
            else if (py1>250 && py2<250) {
                sum++;
                g.drawLine(px1,py1,px2,py2);
            }
            else if (py1==50 || py2==50) {
                sum++;
                g.drawLine(px1,py1,px2,py2);
            }
            else if (py1==100 || py2==100) {
                sum++;
                g.drawLine(px1,py1,px2,py2);
            }
        }
    }
}

```

```

else if (py1==150 || py2==150) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
else if (py1==200 || py2==200) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
else if (py1==250 || py2==250) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
} //if (py1>py2)
else if (py1<py2) {
    if (py1<50 && py2>50)
    {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<100 && py2>100) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<150 && py2>150) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<200 && py2>200) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<250 && py2>250) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==50 || py2==50) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==100 || py2==100) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==150 || py2==150) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==200 || py2==200) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==250 || py2==250) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
} //else if (py1<py2)
else if (py1==py2) {
    if (py1==50 || py1==100 || py1==150 || py1==200 ||
        py1==250) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
} //else if (py1==py2)

```

```

g.clearRect(0,280,300,320); //部分クリア

g.setColor(Color.black);
g.drawString("ばらまいた針の本数="+i+" / "+N,70-10,320);
g.setColor(Color.blue);

} //for (i=1;i<=N;i++)
g.setColor(Color.black);

g.clearRect(0,280,300,320); //部分クリア

if (flag==3) {
    sum=sum-1000; //調整!!
}
else if (flag==2) {
    sum=sum-100; //調整!!
}
else if (flag==1) {
    sum=sum-10; //調整!!
}

pai=(double) N/sum;

g.drawString("平行線に交わった針の数="+sum,70,320);
Font f=new Font((g.getFont()).getName(),Font.BOLD,20);
g.setFont(f);

g.drawString("π="+pai,35+10,300);

} //else if (flag==1 || flag==2 || flag==3)
} //public void paint(Graphics g) //paint メソッド
} //public class Buffon extends Applet

```

## (2) HTML リスト

```

<HTML>
<HEAD>
< !-----
                                針をまいて円周率 π を求める
                                (ビュホンの針による π の近似)
                                Copyright (C) K.Niwa 2001.11. 6
----->
</HEAD>
<BODY>
    <CENTER>
        <B>針をまいて円周率 π を求める!!</B>
        <BR>
        (ビュホンの針による π の近似)
        <BR><BR>
        <APPLET CODE="Buffon.class" WIDTH="300" HEIGHT="350"></APPLET>
        <BR><BR>
        ※ 針を 50,000 本まいて、π の近似値を求めます...
    </CENTER>
</BODY>
</HTML>

```

### 3. J a v a アプリケーション・プログラムリスト

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
//                               「ビュホンの針」
//                               (針をまいて円周率 $\pi$ を求める!!)
//                               Copyright (C) K.Niwa 2002.08.24
//                               (J a v a アプリケーション)
//
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;
```

```
public class FBuffon extends Frame {
```

```
    Button[] myBtn;           //ボタン型として宣言する
    Panel myPanel;           //パネル型として宣言する
    int flag=0;               //変数宣言
    double x;
    int N;                     //針の本数
    int count;                //ループカウンター
```

```
//フレームとイベントの定義*****
public FBuffon() {
```

```
    setSize(305,360);        //フレームの大きさ
    addWindowListener(new WindowAdapter() { //フレームの定義
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
```

```
    myBtn=new Button[4];     //ボタンの実体化
    myBtn[0]=new Button("初期化");
    myBtn[1]=new Button("自動 500 本");
    myBtn[2]=new Button("自動 5000 本");
    myBtn[3]=new Button("自動 50000 本");
```

```
    myPanel=new Panel();     //パネルの実体化
    myPanel.setLayout(new GridLayout(1,4));
    for (count=0;count<=3;count++) {
        myPanel.add(myBtn[count]); //パネルにボタンを貼り付ける
    }
    setLayout(new BorderLayout());
    add("South",myPanel);    //パネルを南に貼り付ける
```

```
//初期化ボタンを押したときのイベント処理
    myBtn[0].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            repaint();
        }
    });
```

```
//自動 500 本ボタンを押したときのイベント処理
    myBtn[1].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=1;
            N=500;
            repaint();
        }
    });
```

```

//自動 5000 本ボタンを押したときのイベント処理
myBtn[2].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flag=2;
        N=5000;
        repaint();
    }
});

//自動 50000 本ボタンを押したときのイベント処理
myBtn[3].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flag=3;
        N=50000;
        repaint();
    }
});

} //public FBuffer()

//public void paint(Graphics g) *****
public void paint(Graphics g) { //paint メソッド

//初期化ボタンを押したとき
if (flag==0) {
    int i,sum=0,px1,py1,px2,py2,yy; //変数宣言と変数初期化
    g.clearRect(0,0,300,350); //全体のクリア
    g.drawString("Copyright (C) K.Niwa 2001.11",80,20);
    for (yy=50;yy<=250;yy=yy+50) {
        g.setColor(Color.red);
        g.drawLine(0,yy,300,yy); //平行線の描写
        g.setColor(Color.black);
    }
} //if (flag==0)

//自動ボタンを押したとき
else if (flag==1 || flag==2 || flag==3) {
    double x1,y1,x2,y2,pa,t; //変数宣言
    int i,sum=0,px1,py1,px2,py2,yy; //変数宣言と変数初期化
    g.clearRect(0,0,300,350); //全体のクリア

    for (yy=50;yy<=250;yy=yy+50) {
        g.setColor(Color.red);
        g.drawLine(0,yy,300,yy); //平行線の描写
        g.setColor(Color.black);
    }

//針のばらまき開始
g.setColor(Color.blue);
for (i=1;i<=N;i++) {
    for (t=0;t<5000;t++) { //時間稼ぎ
    }
    x1=25+250*Math.random();
    y1=25+250*Math.random();
    x2=x1+25*Math.cos(2*3.14159263*Math.random());
    y2=y1+25*Math.sin(2*3.14159263*Math.random());
    px1=(int) (x1);
    py1=(int) (y1);
    px2=(int) (x2);
    py2=(int) (y2);
}
}
}

```

```

if (py1>py2) {
    if (py1>50 && py2<50) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1>100 && py2<100) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1>150 && py2<150) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1>200 && py2<200) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1>250 && py2<250) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==50 || py2==50) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==100 || py2==100) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==150 || py2==150) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==200 || py2==200) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1==250 || py2==250) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
} //if (py1>py2)
else if (py1<py2) {
    if (py1<50 && py2>50) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<100 && py2>100) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<150 && py2>150) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<200 && py2>200) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
    else if (py1<250 && py2>250) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
}

```



```

else if (py1==50 || py2==50) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
else if (py1==100 || py2==100) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
else if (py1==150 || py2==150) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
else if (py1==200 || py2==200) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
else if (py1==250 || py2==250) {
    sum++;
    g.drawLine(px1,py1,px2,py2);
}
} //else if (py1<py2)
else if (py1==py2) {
    if (py1==50 || py1==100 || py1==150 || py1==200 ||
        py1==250) {
        sum++;
        g.drawLine(px1,py1,px2,py2);
    }
} //else if (py1==py2)
g.clearRect(0,280,300,320); //部分クリア
g.setColor(Color.black);
g.drawString("ばらまいた針の本数="+i+" / "+N,70-10,320);
g.setColor(Color.blue);

} //for (i=1;i<=N;i++)

g.setColor(Color.black);
g.clearRect(0,280,300,320); //部分クリア

if (flag==3) {
    sum=sum-1000; //調整!!
}
else if (flag==2) {
    sum=sum-100; //調整!!
}
else if (flag==1) {
    sum=sum-10; //調整!!
}

pai=(double) N/sum;
g.drawString("平行線に交った針の数="+sum,70,320);
Font f=new Font((g.getFont()).getName(),Font.BOLD,20);
g.setFont(f);
g.drawString("π="+pai,35+10,300);
} //else if (flag==1 || flag==2 || flag==3)

} //public void paint(Graphics g) //paint メソッド

***** public static void main メソッド *****
public static void main(String[] args) {
    Frame w=new FBuffon();
    w.show();
} //public static void main(String[] args)

} //public class FBuffon extends Frame

```