日本火山学会2018年秋季大会

溶岩流シミュレーション55LAVAの開発

Development of lava flow simulation software 55LAVA.

○佐伯和人（大阪大学・理学研究科）
○Kazuto Saiki (Graduate School of Science, Osaka Univ.)

55LAVA　・・・・　シンプルで理解し易く、カスタマイズし易い、溶岩流シミュレーションソフト

伊豆大島のように島の任意の場所が火口になる可能性がある火山島は多数存在する。避難経路が限られる火山島では、平時から、住民の手によって溶岩流流下シミュレーションが手軽にできることが、避難行動を決定する上で有効であると考えられる。立体地形模型に液体を流す簡易な実験であっても、溶岩の流下範囲を考察するには大きな効果があると期待される。

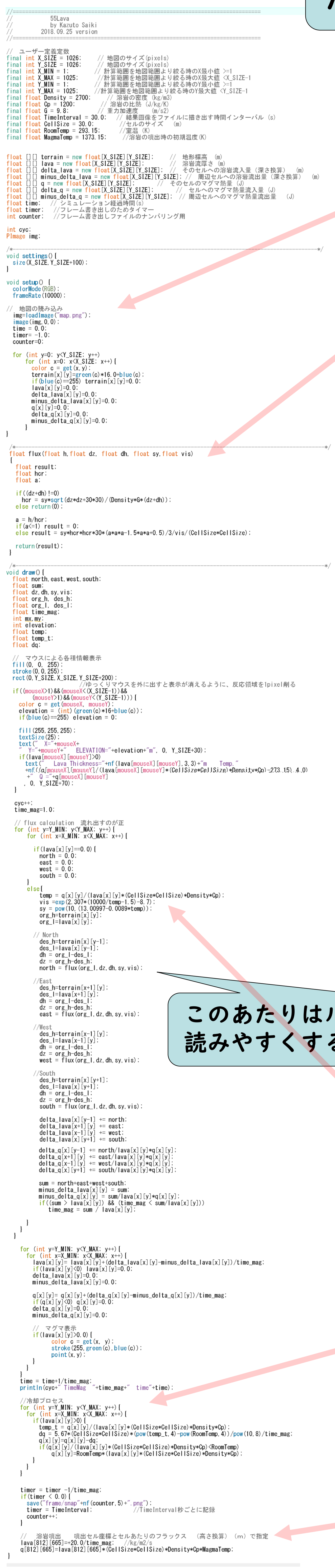
しかし、どこで溶岩が止まるか、隣の谷筋に分岐することはないか、などの疑問に答えることは簡単ではない。溶岩流の流れる方向は地形でほとんどが決まるとは言え、溶岩流の流下速度や到達延長距離は、溶岩の総噴出量や噴出フラックスで決まり、これらの値は噴火前の予測が難しい。さらに、冷却に伴う溶岩流の粘性増大により、時に地形の最大傾斜方向ではない方向に分岐して流下することもある。

高度な溶岩流シミュレーションは存在するが、火口の位置変化や、溶岩流の噴出フラックスの変化等に対応した考察をするためには、様々な初期条件の変化を原理を理解しつつ高校生などが気軽に試すことができるシミュレーションの公開が効果的なのではないかと考えている。その一つの例として開発したのが、今回発表する溶岩流シミュレーションソフト55LAVAである。

55LAVAのアルゴリズム

MAGFLOW(Bilotta et al. (2012)をベースに、容易に改造ができるツールにすることを目的に部分的に変更してProcessing言語で作製。短いソースながらも、データの入出力を視覚的にわかりやすくする工夫もしている。「高校生にも読めるソース」が目標。

Processing言語のソースはたったこれだけ。



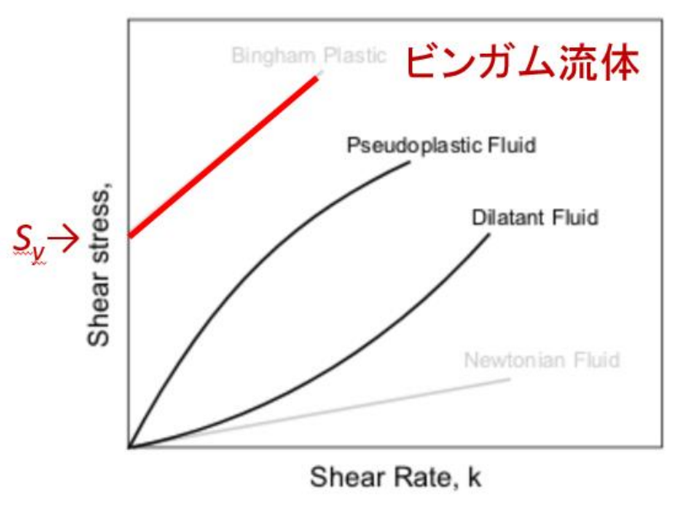
```
//==============================================================
//            55Lava
//            by Kazuto Saiki
//          2018.09.25 version
//==============================================================

//  ユーザー定義定数
final int X_SIZE = 1026;      // 地図のサイズ(pixels)
final int Y_SIZE = 1026;      // 地図のサイズ(pixels)
final int X_MIN = 1;         // 計算範囲を地図範囲より絞る時のX最小値 >=1
final int X_MAX = 1025;       //計算範囲を地図範囲より絞る時のX最大値 <X_SIZE-1
final int Y_MIN = 1;         // 計算範囲を地図範囲より絞る時のY最小値 >=1
final int Y_MAX = 1025;      //計算範囲を地図範囲より絞る時のY最大値 <Y_SIZE-1
final float Density = 2700;       // 溶岩の密度 (kg/m3)
final float Cp = 1200;          // 溶岩の比熱 (J/kg/K)
final float G = 9.8;           // 重力加速度     (m/s2)
final float TimeInterval = 30.0;   // 結果画像をファイルに描き出す時間インターバル (s)
final float CellSize = 30.0;         //セルのサイズ    (m)
final float RoomTemp = 293.15;       //室温 (K)
final float MagmaTemp = 1373.15;      //溶岩の噴出時の初期温度(K)


float [][] terrain = new float[X_SIZE][Y_SIZE];      //  地形標高  (m)
float [][] lava = new float[X_SIZE][Y_SIZE];         //  溶岩流厚さ (m)
float [][] delta_lava = new float[X_SIZE][Y_SIZE];  //  そのセルへの溶岩流入量 (深さ換算)   (m)
float [][] minus_delta_lava = new float[X_SIZE][Y_SIZE]; //  周辺セルへの溶岩流出量 (深さ換算)   (m)
float [][] q = new float[X_SIZE][Y_SIZE];         //  そのセルのマグマ熱量 (J)
float [][] delta_q = new float[X_SIZE][Y_SIZE];       //  セルへのマグマ熱量流入量 (J)
float [][] minus_delta_q = new float[X_SIZE][Y_SIZE];  // 周辺セルへのマグマ熱量流出量   (J)
float time;   // シミュレーション経過時間(s)
float timer;   //フレーム書き出しのためタイマー
int counter;   //フレーム書き出しファイルのナンバリング用

int cyc;
PImage img;

/*--------------------------------------------------------------------------------*/
void settings(){
  size(X_SIZE,Y_SIZE+100);
}

void setup() {
  colorMode(RGB);
  frameRate(10000);

//  地図の読み込み
  img=loadImage("map.png");
  image(img,0,0);
  time = 0.0;
  timer= -1.0;
  counter=0;

  for (int y=0; y<Y_SIZE; y++)
      for (int x=0; x<X_SIZE; x++){
        color c = get(x,y);
        terrain[x][y]=green(c)*16.0+blue(c);
        if(blue(c)==255) terrain[x][y]=0.0;
        lava[x][y]=0.0;
        delta_lava[x][y]=0.0;
        minus_delta_lava[x][y]=0.0;
        q[x][y]=0.0;
        delta_q[x][y]=0.0;
        minus_delta_q[x][y]=0.0;
      }
}

 /*-------------------------------------------------------------------------------*/
 float flux(float h,float dz, float dh, float sy,float vis)
 {
   float result;
   float hcr;
   float a;

   if((dz+dh)!=0)
     hcr = sy*sqrt(dz*dz+30*30)/(Density*G*(dz+dh));
   else return(0);

   a = h/hcr;
   if(a<=1) result = 0;
   else result = sy*hcr*hcr*30*(a*a*a-1.5*a*a+0.5)/3/vis/(CellSize*CellSize);

   return(result);
 }

 /*-------------------------------------------------------------------------------*/
void draw(){
  float north,east,west,south;
  float sum;
  float dz,dh,sy,vis;
  float org_h, des_h;
  float org_l, des_l;
  float time_mag;
  int mx,my;
  int elevation;
  float temp;
  float temp_t;
  float dq;

  //  マウスによる各種情報表示
  fill(0, 0, 255);
  stroke(0,0,255);
  rect(0,Y_SIZE,X_SIZE,Y_SIZE+200);
                        //ゆっくりマウスを外に出すと表示が消えるように、反応領域を1pixel削る
  if((mouseX>1)&&(mouseX<(X_SIZE-1))&&
        (mouseY>1)&&(mouseY<(Y_SIZE-1))){
    color c = get(mouseX, mouseY);
    elevation = (int)(green(c)*16+blue(c));
    if(blue(c)==255) elevation = 0;

    fill(255,255,255);
    textSize(25);
    text("  X="+mouseX+
    "  Y="+mouseY+"    ELEVATION="+elevation+"m", 0, Y_SIZE+30);
    if(lava[mouseX][mouseY]>0)
      text("    Lava Thickness="+nf(lava[mouseX][mouseY],3,3)+"m     Temp."
      +nf((q[mouseX][mouseY]/(lava[mouseX][mouseY]*(CellSize*CellSize)*Density*Cp)-273.15),4,0)
       +"  Q ="+q[mouseX][mouseY]
     , 0, Y_SIZE+70);
  }

  cyc++;
  time_mag=1.0;

  // flux calculation  流れ出すのが正
  for (int y=Y_MIN; y<Y_MAX; y++){
      for (int x=X_MIN; x<X_MAX; x++){

        if(lava[x][y]==0.0){
          north = 0.0;
          east = 0.0;
          west = 0.0;
          south = 0.0;
      }
      else{
          temp = q[x][y]/(lava[x][y]*(CellSize*CellSize)*Density*Cp);
          vis =exp(2.307*(10000/temp-1.5)-8.7);
          sy = pow(10,(13.00997-0.0089*temp));
          org_h=terrain[x][y];
          org_l=lava[x][y];

        // North
          des_h=terrain[x][y-1];
          des_l=lava[x][y-1];
          dh = org_l-des_l;
          dz = org_h-des_h;
          north = flux(org_l,dz,dh,sy,vis);

        //East
          des_h=terrain[x+1][y];
          des_l=lava[x+1][y];
          dh = org_l-des_l;
          dz = org_h-des_h;
          east = flux(org_l,dz,dh,sy,vis);

        //West
          des_h=terrain[x-1][y];
          des_l=lava[x-1][y];
          dh = org_l-des_l;
          dz = org_h-des_h;
          west = flux(org_l,dz,dh,sy,vis);

        //South
          des_h=terrain[x][y+1];
          des_l=lava[x][y+1];
          dh = org_l-des_l;
          dz = org_h-des_h;
          south = flux(org_l,dz,dh,sy,vis);

          delta_lava[x][y-1] += north;
          delta_lava[x+1][y] += east;
          delta_lava[x-1][y] += west;
          delta_lava[x][y+1] += south;

          delta_q[x][y-1] += north/lava[x][y]*q[x][y];
          delta_q[x+1][y] += east/lava[x][y]*q[x][y];
          delta_q[x-1][y] += west/lava[x][y]*q[x][y];
          delta_q[x][y+1] += south/lava[x][y]*q[x][y];

          sum = north+east+west+south;
          minus_delta_lava[x][y] = sum;
          minus_delta_q[x][y] = sum/lava[x][y]*q[x][y];
          if((sum > lava[x][y]) && (time_mag < sum/lava[x][y]))
             time_mag = sum / lava[x][y];

      }
    }
  }

   for (int y=Y_MIN; y<Y_MAX; y++){
     for (int x=X_MIN; x<X_MAX; x++){
       lava[x][y]= lava[x][y]+(delta_lava[x][y]-minus_delta_lava[x][y])/time_mag;
       if(lava[x][y]<0) lava[x][y]=0.0;
       delta_lava[x][y]=0.0;
       minus_delta_lava[x][y]=0.0;

       q[x][y]= q[x][y]+(delta_q[x][y]-minus_delta_q[x][y])/time_mag;
       if(q[x][y]<0) q[x][y]=0.0;
       delta_q[x][y]=0.0;
       minus_delta_q[x][y]=0.0;

       //  マグマ表示
       if(lava[x][y]>0.0){
             color c = get(x, y);
             stroke(255,green(c),blue(c));
             point(x,y);
       }
     }
   }
   time = time+1/time_mag;
   println(cyc+" TimeMag  "+time_mag+"  time"+time);

   //冷却プロセス
   for (int y=Y_MIN; y<Y_MAX; y++){
     for (int x=X_MIN; x<X_MAX; x++){
       if(lava[x][y]>0){
         temp_t = q[x][y]/(lava[x][y]*(CellSize*CellSize)*Density*Cp);
         dq = 5.67*(CellSize*CellSize)*(pow(temp_t,4)-pow(RoomTemp,4))/pow(10,8)/time_mag;
         q[x][y]=q[x][y]-dq;
         if(q[x][y]/(lava[x][y]*(CellSize*CellSize)*Density*Cp)<RoomTemp)
             q[x][y]=RoomTemp*(lava[x][y]*(CellSize*CellSize)*Density*Cp);
       }
     }
   }


   timer = timer -1/time_mag;
   if(timer < 0.0){
     save("frame/snap"+nf(counter,5)+".png");
     timer = TimeInterval;              //TimeInterval秒ごとに記録
     counter++;
   }

   //   溶岩噴出    噴出セル座標とセルあたりのフラックス  (高さ換算) (m) で指定
   lava[812][665]=+20.0/time_mag;   //kg/m2/s
   q[812][665]=lava[812][665]*(CellSize*CellSize)*Density*Cp*MagmaTemp;
}
```

このあたりはループで短くできるが、読みやすくするためにあえて展開

q： 体積フラックス
S_y : 降伏応力
h : 溶岩流の深さ
h_{cr}： 流れるための臨界深さ
Δx： セルサイズ
η： 粘度
$a = h/h_{cr}$　(a>1の時のみ流れる）
T: 温度

【隣のセルへの流れ出しフラックス計算】

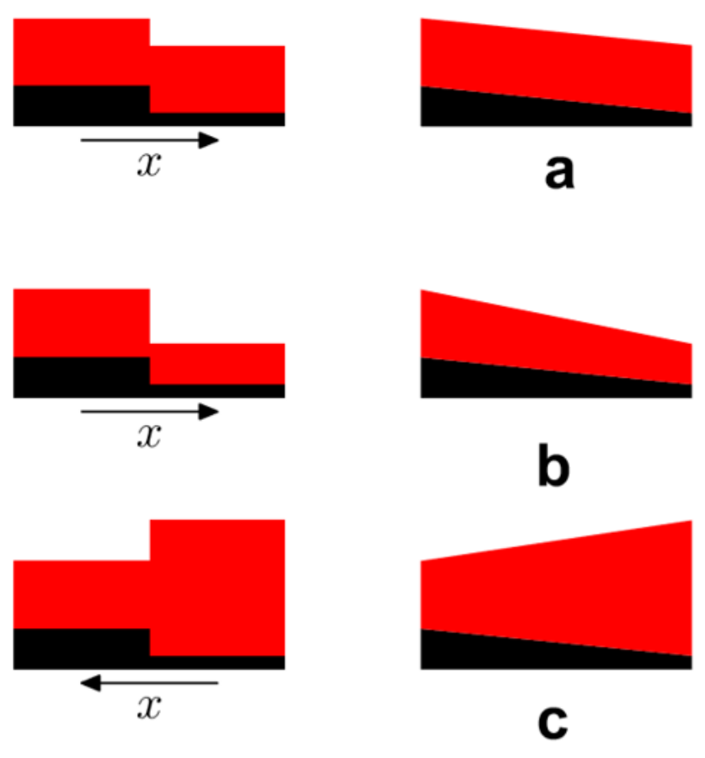
$$q = \frac{S_y h_{cr}^2 \Delta x}{3\eta}\left(a^3 - \frac{3}{2}a^2 + \frac{1}{2}\right).$$



$$\log_{10} S_y = 13.00997 - 0.0089T$$

Ishihara et al. (1990)より

$$h_{cr} = \frac{S_y}{\rho g(\sin\alpha - \partial h/\partial x \cos\alpha)} \simeq S_y \frac{\sqrt{\Delta z^2 + \Delta x^2}}{\rho g(\Delta z + \Delta h)}$$

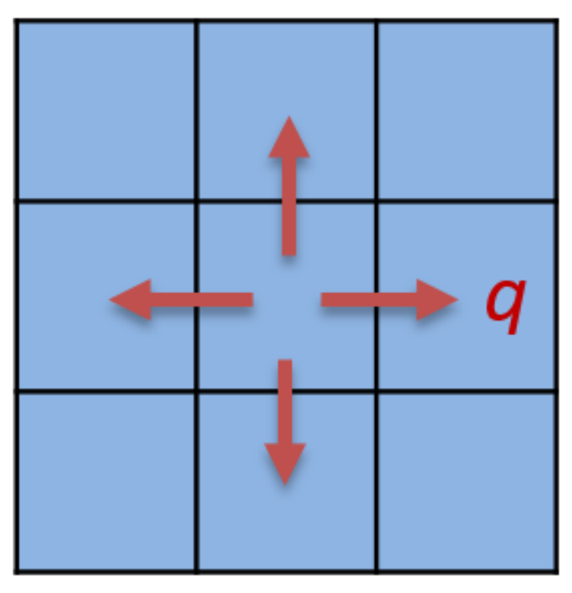


$\alpha > 0, \Delta z = 0.2, \Delta h = 0$

$\alpha > 0, \Delta z = 0.2, \Delta h = 0.2$

$\alpha < 0, \Delta z = -0.2, \Delta h = 0.5$

h_{cr}： 流れるための臨界深さ
ρ： 溶岩の密度
S_y : 降伏応力
h : 溶岩流の深さ
Δx： セルサイズ
Δh: 溶岩の厚さの差
Δz: 標高の差
α： 地形傾斜角度



フラックスは東西南北4方向への流れ出しのみ計算。まず、1秒で流れ出す量を計算しておき、1サイクルの計算の最後で、すべてのセルで流れ出し量がもとある量を越えないように、経過時間を短縮する処理を行う。

【溶岩の粘性計算】

$$\eta = e^{k\left(\frac{1000}{T} - 1.5\right) - 8.7}$$

Shaw(1969)より

【溶岩の放射冷却計算】

$$Q(t + \Delta t) = Q(t) + \boxed{\Delta Q_r} + \boxed{\Delta Q_m}$$

$$\Delta Q_r = \varepsilon\sigma T_s^4 \Delta t$$

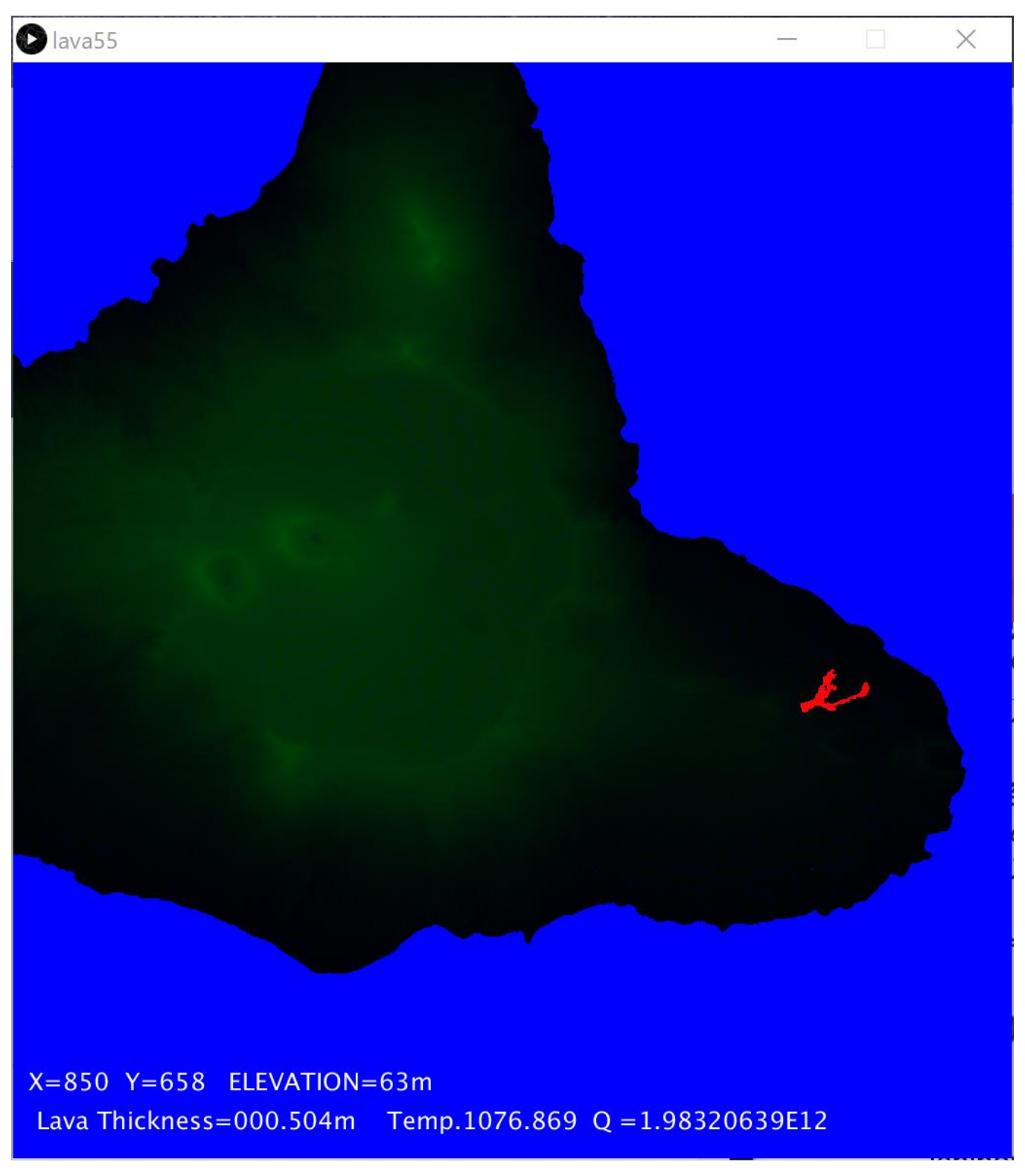
Heat loss due to radiation

Heat loss due to mixture of lavas

Q： 熱量
ε： 溶岩の放射率
t： 時間
T_s： 溶岩の表面温度
σ： ステファンボルツマン定数
η : 溶岩の粘度
k : 溶岩の組成によって決まる係数
T : 溶岩の温度

【溶岩の噴出】
噴出セルの座標に1秒あたりの溶岩上昇高さと熱量を直接加算している。

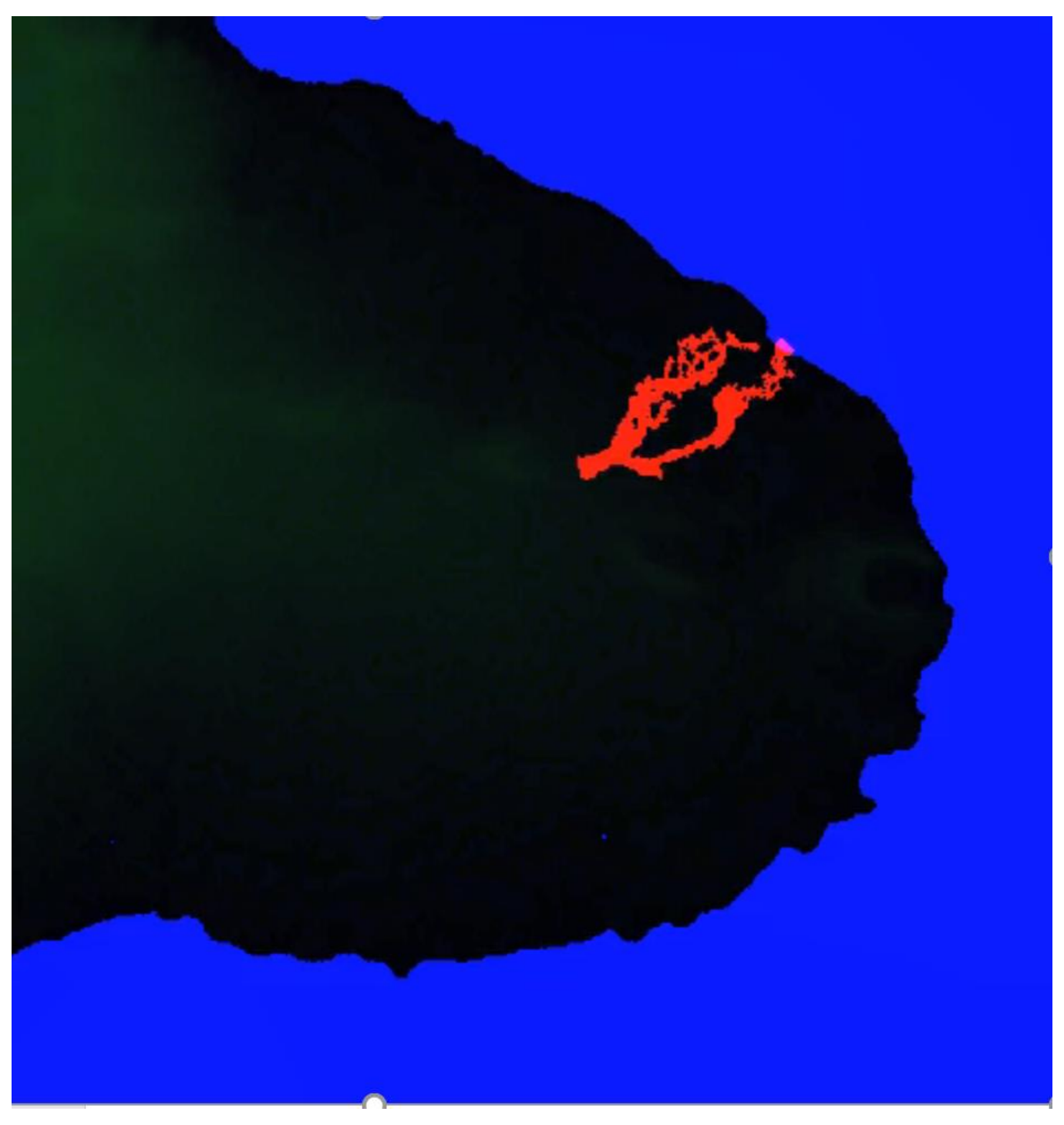
【地図の読み込み】
標高1m刻みのDEMを作り、標高の上位8bitを緑プレーンに、下位4bitを青プレーンにのせた地図を入力地図とする。青プレーンの4bitと、赤プレーンの8bitは丸々余っているので、溶岩を赤プレーンに表示した。

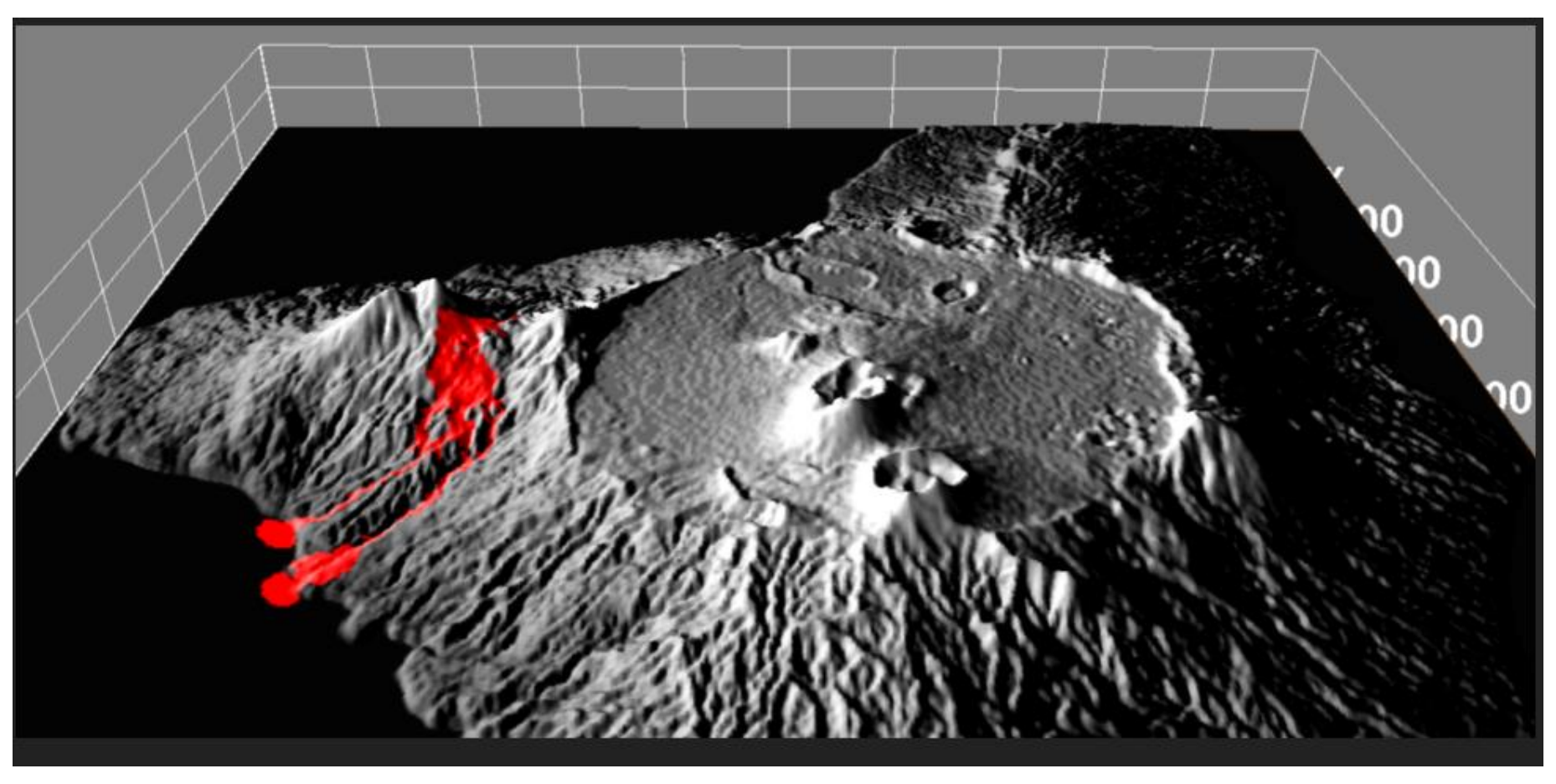


X=850 Y=658 ELEVATION=63m
Lava Thickness=000.504m Temp.1076.869 Q =1.98320639E12

サンプルは、バヌアツ共和国アンブリム島。地形データは、ALOS全球数値地表モデル(©JAXA) から作成。1セルのサイズは30mで1026x1026の地図サイズであるとした。マウスカーソルを地図上に持って行くと、座標と標高が表示される。マグマに持って行くと、マグマの厚さ、温度、熱量が表示される。

【結果】
割れ目噴火で毎秒18000m³、1100℃の溶岩が噴出したという過激な設定の91分後の結果





ImageJを使って、3D地形に結果をラップすることもできる。（左図）

55LAVA公開・配布について

以下のサイトにてソースを公開しております。バグレポートやご要望などお待ちしております。
http://gate.main.jp/saiki_lab/