

Sandpile model project

Groupe : Oscar STEMMELIN, Cristian CARP et Damien LEBAS

Table des matières

1	Présentation des articles de recherche de Per Bak, Chao Tang, & Kurt Wiesenfeld	2
1.1	Introduction	2
1.2	Le modèle du tas de sable et l'émergence du bruit $1/f$	2
1.3	Décroissance en puissance et lien avec le bruit $1/f$	4
2	La Criticalité Auto-Organisée (SOC)	4
3	Théorie des graphes et géométrie abélienne	5
3.1	Le processus sous l'optique de la théorie des graphes	5
3.2	Le modèle abélien du tas de sable	5
4	Modélisation	7
4.1	Modèle centré	7
4.2	Modèle aléatoire	8
5	Extensions : Les tremblements de Terre et la SOC	9
6	Codes annexes	10

1 Présentation des articles de recherche de Per Bak, Chao Tang, & Kurt Wiesenfeld

1.1 Introduction

Les concepts de **criticalité auto-organisée** (*Self-Organized Criticality*, SOC) et de **bruit rose** (*Flicker noise*) ont été introduits par **Per Bak, Chao Tang** et **Kurt Wiesenfeld** (BTW) dans les deux articles que nous avons étudiés :

- *Self-Organized Criticality : An Explanation of 1/f Noise* (1987)
- *Self-Organized Criticality* (1988)

Ces deux publications montrent que des systèmes physiques, biologiques ou économiques peuvent **s'auto-organiser d'eux mêmes** vers un état critique, sans réglage de paramètres, et ceci grâce à la présence du *Flicker noise*.

Le modèle de référence proposé par BTW est celui du **tas de sable**, une métaphore simple mais puissante de cette dynamique.

1.2 Le modèle du tas de sable et l'émergence du bruit 1/f

Dans ce modèle, on ajoute des grains de sable un à un sur une pile. Lorsque la pente dépasse une valeur critique, un effondrement local ou une *avalanche* (plusieurs effondrements) se produit. Les avalanches peuvent être de toutes tailles : certaines sont locales, d'autres affectent l'ensemble du tas. Ainsi, le système évolue naturellement vers un état où il est "**juste**" **stable**, appelé *état critique auto-organisé*.

Le but de ce modèle est de retrouver la présence du **bruit rose** un type de fluctuation temporelle. En effet, BTW ont fait la prédiction initiale selon laquelle la criticalité auto-organisée se dirige à chaque fois vers un point critique à cause des corrélations spatiales et du bruit rose 1/f.

Ils commencent d'abord avec un modèle en une dimension, cependant celui-ci ne comporte qu'un bruit blanc qui, contrairement au bruit rose, ne peut pas amener à la SOC.

En oppositions, les modèles en 2D et 3D permettent l'observation de la SOC : on retrouve des avalanches de toutes tailles à toutes échelles, une absence d'échelle caractéristique, et un spectre temporel 1/f.

Pour le modèle en 2D, BTW traduisent le processus du tas de sable en un **automate cellulaire** : chaque cellule d'une grille 2D z contient un entier $z(x, y)$ représentant le nombre de grains. Lorsqu'une cellule devient instable $z(x, y) > z_c$ (dans notre cas $z_c = 4$), elle redistribue quatre grains à ses voisins :

$$z(x, y) \rightarrow z(x, y) - 4, \quad (1)$$

$$z(x \pm 1, y) \rightarrow z(x \pm 1, y) + 1, \quad (2)$$

$$z(x, y \pm 1) \rightarrow z(x, y \pm 1) + 1 \quad (3)$$

Le processus se poursuit jusqu'à ce que plus aucune cellule ne dépasse le seuil critique z_c . Chaque séquence d'effondrements constitue une *avalanche* dont :

- s correspond au nombre total d'effondrements
- t correspond au nombre d'itérations avant stabilisation.

Les distributions observées suivent des **lois de puissance** :

$$D(s) \propto s^{-\tau}, \quad D(t) \propto t^{-\alpha} \quad (4)$$

En représentation logarithmique, ces distributions apparaissent comme des droites dont les pentes sont respectivement $-\tau$ et $-\alpha$. Pour un modèle comme le leur en 2D, $\tau \approx 0.98$ et $\alpha \approx 0.43$.

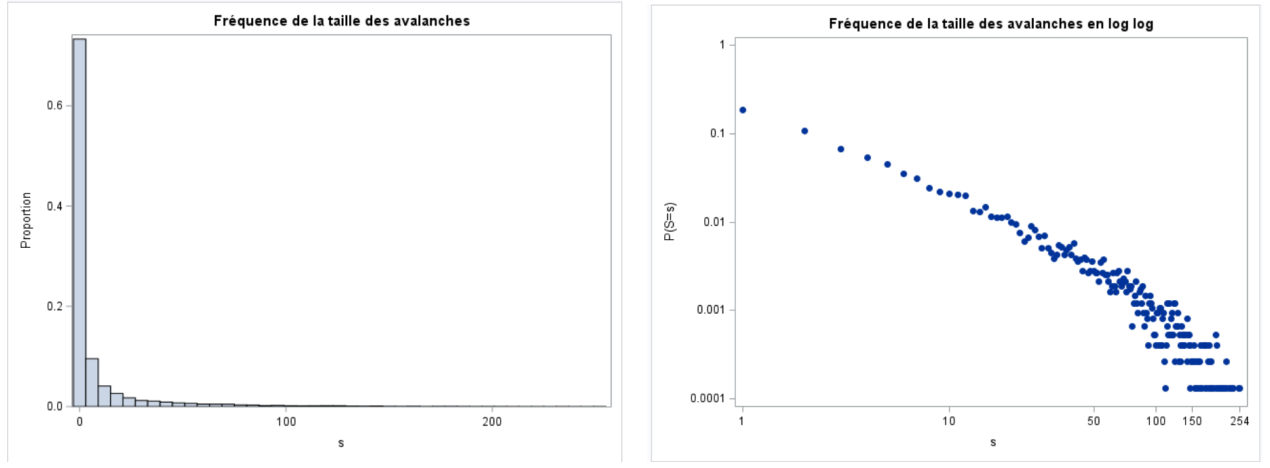


FIGURE 1 – Fréquence de la taille des avalanches sur une matrice 20X20, 20 000 itérations

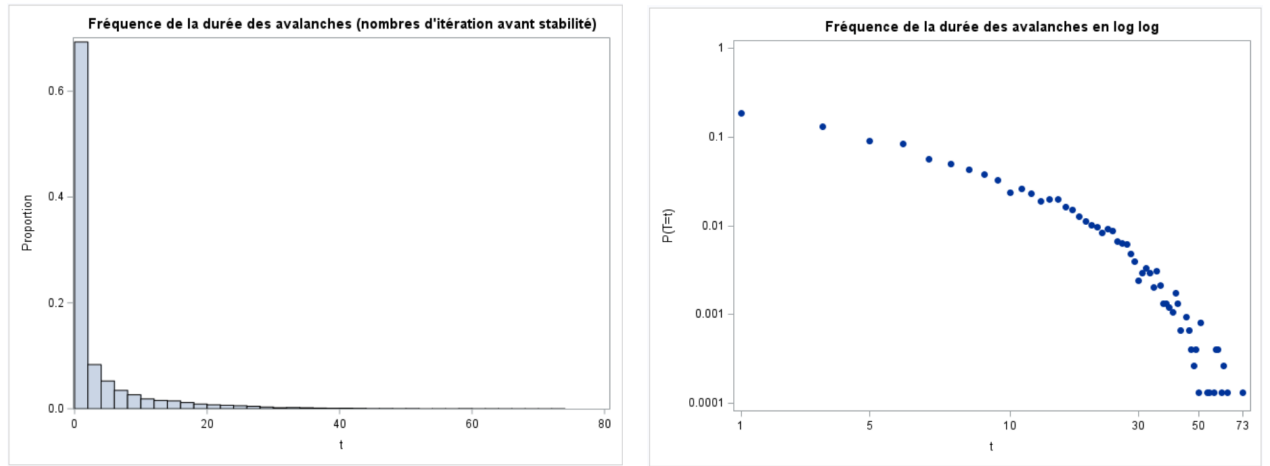


FIGURE 2 – Fréquence de la durée des avalanches sur une matrice 20X20, 20 000 itérations

La distribution en transformation log log se retrouve avec une pente plus accentuée sur la fin de la droite, ceci est dû à la taille limitée de la matrice du tas de sable : les grosses avalanches se retrouvent confrontées aux bords de la matrice.

L'absence d'échelle observée au niveau spatial et temporel a conduit BTW à questionner davantage la durée des avalanches ainsi que leur activité globale sur le long terme.

Une nouvelle unité a été attribuée aux glissements de sable : l'énergie dissipée lors d'un glissement. La quantité $f(x, t)$ représente l'énergie dissipée au point x au moment t . Ainsi, le nombre total de glissements au moment t peut être écrit de la manière suivante :

$$F(t) = \int f(x, t), dx. \quad (5)$$

La taille totale d'une avalanche, représentant l'activité totale d'un événement, est donnée par :

$$s = \int F(t), dt. \quad (6)$$

En se concentrant sur les fluctuations temporelles de $F(t)$, BTW ont cherché sa loi de puissance $S(f)$, représentée de la manière suivante :

$$S(f) = \int \langle F(t_0 + t), F(t_0) \rangle, e^{2\pi i f t}, dt, \quad (7)$$

où $\langle F(t_0 + t)F(t_0) \rangle$ représente l'autocorrélation de $F(t)$.

À la fin des calculs, ils ont retrouvé que le spectre de puissance de $F(t)$ est égal à :

$$S(f) \approx f^{-\beta}. \quad (8)$$

Où :

- $\beta = 0$: $S(f)$ est un bruit blanc ;
- $\beta = 1$: $S(f)$ est un bruit rose (*flicker noise*).

Par conséquent, la prédiction selon laquelle le modèle arrive spontanément dans un état critique, grâce aux corrélations spatiales et au bruit $1/f$, est vérifiée numériquement.

1.3 Décroissance en puissance et lien avec le bruit $1/f$

Un aspect fondamental du modèle de BTW est l'apparition du **spectre** $1/f$ dans l'activité globale du système. Ce comportement ne vient pas d'une décroissance rapide des corrélations temporelles, comme dans les systèmes non critiques, mais au contraire d'une **décroissance lente**, qui s'étend sur le temps.

Dans un système ordinaire, les fluctuations sont indépendantes et l'activité perd rapidement la mémoire des événements précédents. Les corrélations chutent alors **de façon exponentielle**, ce qui conduit à un **bruit blanc**. C'est ce que les auteurs ont pu observer avec le modèle en une dimension.

Dans le modèle en 2D et 3D, la situation est très différente. Les avalanches interagissent sur différentes échelles de temps, ce qui entraîne une décroissance bien plus progressive des corrélations : une **décroissance en puissance**. Le spectre n'est plus "plat" (avec $\beta = 0$), mais suit une loi proche de :

$$S(f) \propto \frac{1}{f} \quad (9)$$

avec $\beta = 1$.

Autrement dit, le **bruit rose** (ou $1/f$ noise) est la manifestation directe du fait que le système ne "décroche" pas rapidement de son passé : il garde une trace des événements sur de longues durées. Cette persistance reflète l'absence d'échelle caractéristique et caractérise la SOC.

Ce comportement se retrouve dans de nombreux phénomènes naturels, séismes, activité cérébrale, turbulence ou marchés financiers.

Lien entre la loi de puissance et le bruit $1/f$

Dans le cadre du modèle du tas de sable, la mesure de la pente des distributions $D(s)$ ou $D(t)$ permet déjà d'observer une caractéristique enécessaire à la criticalité auto-organisée : la **loi de puissance**.

Même si le calcul du spectre de puissance $S(f)$ nécessiterait une transformation de Fourier de la fonction d'autocorrélation $\langle F(t_0 + t)F(t_0) \rangle$, la présence d'une telle loi de puissance dans les distributions $D(s)$ et $D(t)$ constitue déjà une **signature indirecte du bruit** $1/f$. En effet, la superposition d'avalanches de tailles et de durées variées, corrélées sur différentes échelles de temps, conduit à un spectre de fluctuations $S(f) \propto 1/f^\beta$, avec $\beta \approx 1$.

Ainsi, l'estimation de la pente $-\tau \approx -1$ de $D(s)$ ou $-\alpha \approx -0.45$ de $D(t)$ traduit la présence implicite du bruit rose. Dans notre cas nous utiliserons la pente de $D(s)$ pour prouver la présence de ce bruit.

2 La Criticalité Auto-Organisée (SOC)

Le concept de *Criticalité Auto-Organisée* (SOC) vise à expliquer pourquoi certains systèmes physiques ou naturels semblent capables d'évoluer vers un état critique, sans ajustements de paramètres extérieurs. Contrairement aux systèmes traditionnels où un changement d'état dépend d'un paramètre externe, ici, la transition est *endogène*. Le système s'auto-régule.

Dans les transitions de phase classiques, un paramètre externe (par exemple la température) doit être ajusté pour atteindre le point critique. Dans un système auto-organisé critique, la transition de phase est endogène : elle émerge spontanément des interactions internes, en raison des autocorrélations spatiales $\langle F(t_0 + t)F(t_0) \rangle$ entre chaque moment (cf. 1.2).

Le système alterne entre :

- des phases stables, où les perturbations locales sont absorbées sans effet global ;
- des phases d’avalanches, où une petite perturbation (un seul grain) peut déclencher une réorganisation globale.

Pendant les phases stables, le système accumule de l’énergie sous forme de tension interne. Chaque ajout de grain de sable modifie légèrement son équilibre. Une fois le seuil franchi ($z_c = 4$), cette tension est alors relâchée sous forme d’avalanche, provoquant une réorganisation temporaire. Une fois l’avalanche terminée, le système retrouve un état stable, et le cycle recommence.

Cette alternance entre accumulation et libération d’énergie constitue le cœur de la SOC : le système s’auto-régule, puisqu’il maintient lui-même l’équilibre entre stabilité et instabilité.

Ce mécanisme illustre un principe plus général : dans de nombreux systèmes, les petites actions peuvent produire des effets à grande échelle, tant temporels que spatiaux. Des phénomènes naturels tels que les séismes, les taches solaires, le courant des rivières, voire même l’activité neuronale, suivent ces principes dynamiques.

3 Théorie des graphes et géométrie abélienne

3.1 Le processus sous l’optique de la théorie des graphes

Le modèle du tas de sable peut être reformulé en termes de **théorie des graphes**. Chaque cellule est représentée par un **nœud** (ou *vertex*), et les redistributions de grains sont modélisées par des **arêtes** reliant les nœuds voisins.

Ce lien peut être mathématisé en se servant de la **matrice de Laplace discrète**[1], formalisée par Kirchhoff au 19^e siècle. Cette matrice permet de relier la structure d’un graphe avec ses propriétés de diffusion. Pour un graphe à n nœuds :

$$L = D - A \tag{10}$$

où :

- L_{ij} mesure comment un nœud i échange avec un nœud j ,
- D est la matrice des degrés, contenant le nombre de connexions de chaque nœud,
- A est la matrice d’adjacence, telle que $A_{ij} = 1$ s’il existe un lien entre i et j , et 0 sinon.

Cette représentation ouvre la voie à une interprétation *combinatoire* et *topologique* du système, à la fois pour le modèle aléatoire ainsi que pour le modèle déterministe. Elle sera toutefois particulièrement pertinente pour le **modèle abélien**.

3.2 Le modèle abélien du tas de sable

En 1990, le physicien Dhar a remarqué que si le processus est rendu entièrement déterministe (en redistribuant les grains en un même point), le système devient abélien. L’adjectif *abélien* renvoie à la propriété de **commutativité** : l’ordre dans lequel les effondrements se produisent n’a pas d’influence sur la configuration finale. Autrement dit, si deux sites i et j sont instables en même temps, l’ordre du glissement n’a aucune importance.

La commutativité garantit que le système évolue de manière déterministe. Puisque chaque redistribution locale suit les mêmes règles linéaires, les motifs qui apparaissent se répètent à différentes échelles entre la stabilisation et l’effondrement. Le corollaire visuel de cette auto-organisation sont les fractales.

Afin d’illustrer, observons ces tas de sable dont les grains sont distribués dans le centre ou dans 4 points de la matrice :

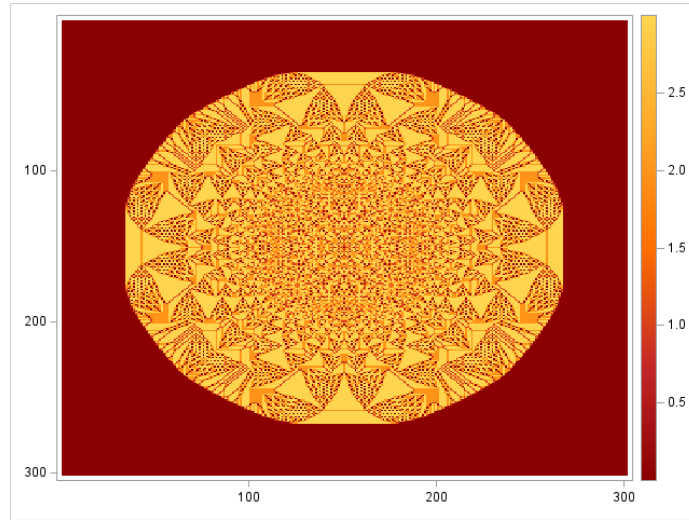


FIGURE 3 – Modèle abélien centré du tas de sable (100 000 itérations)

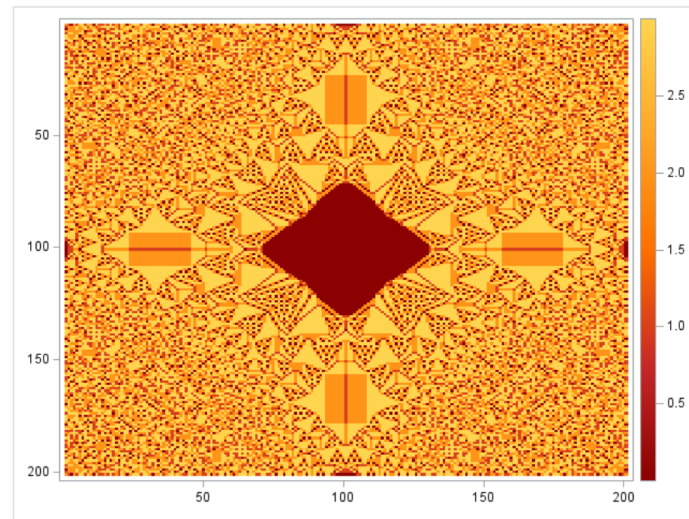


FIGURE 4 – Modèle avec 4 tas de sable aux sommets de la matrice (50 million d'itérations par tas)

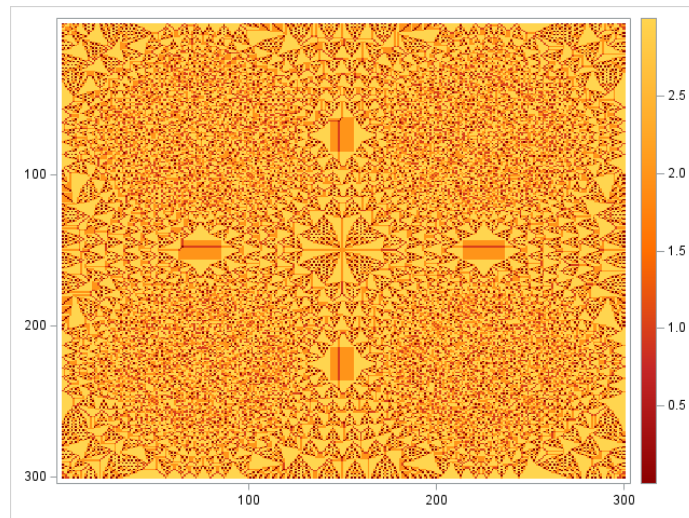


FIGURE 5 – Modèle avec 4 tas de sable au milieu de chaque quart de la matrice (200 000 itérations)

4 Modélisation

4.1 Modèle centré

Listing 1 – Code Sandpile centré

```
proc iml;

    n = 301; *Taille de la matrice;
    sand = j(n, n, 0); *Creation d'une matrice vide;
    center = ceil((n+1) / 2); *Centre de la matrice;
    sand[center,center] = 1000000; *On ajoute du sable au centre;

    *On cree une boucle pour etaler le sable;
    do while(max(sand) >= 4); * jusqu'a que le tas soit stable;
        ind = loc(sand >= 4); *On trouve les cases ou il y a 4 grains ou plus;
        if ncol(ind) > 0 then do;
            sand[ind] = sand[ind] - 4; *La case perd 4 grains;
            /* On redistribut ces grains au cases voisines : */
            gauche = loc(mod(ind, n) ^= 1); *pas sur la premiere colonne;
            droite = loc(mod(ind, n) ^= 0); *pas sur la derniere colonne;
            haut = loc(ind > n); *pas sur la premiere ligne;
            bas = loc(ind <= n*(n-1)); *pas sur la derniere ligne;
            /* redistribution */
            if ncol(gauche) > 0 then sand[ind[gauche] - 1] = sand[ind[gauche] - 1] + 1;
            if ncol(droite) > 0 then sand[ind[droite] + 1] = sand[ind[droite] + 1] + 1;
            if ncol(haut) > 0 then sand[ind[haut] - n] = sand[ind[haut] - n] + 1;
            if ncol(bas) > 0 then sand[ind[bas] + n] = sand[ind[bas] + n] + 1;
        end;
    end;

    title "Représentation du tas de sable apres n iteration";
    call heatmapcont(sand) xvalues=vname yvalues=vname colorramp={cx8B0000 cx62828
        cxFF6F00 cxFFA726 cxFFD54F} displayoutlines=0;
```

Ce programme modélise la diffusion de sable à partir d'un point central du plateau. Le fait de déposer les grains un par un ou de placer directement une grande quantité de sable au centre ne modifie pas la configuration finale.

Le code présenté ici correspond donc au **modèle abélien** du tas de sable de BTW. Il permet de visualiser la structure à caractère **fractale** qui émerge de ce processus **déterministe**.

Cependant, ce modèle ne permet pas d'analyser les propriétés statistiques du système, telles que les **distributions** $D(s)$ et $D(t)$ ainsi que la présence du **bruit rose** ($1/f$) décrits dans les papiers de recherche (bien que ce code soit une bonne base pour créer le modèle aléatoire).

Le code peut paraître assez compliqué surtout au niveau des condition de transfert du grain de la case $[i, j]$ à gauche/droite/haut/bas, ceci est dû à l'utilisation de la fonction **loc** qui nous permet de réduire notre programme à une unique boucle.

4.2 Modèle aléatoire

Listing 2 – Code Sandpile aleatoire

```
proc iml;
  n = 80; *Taille de la matrice;
  z = 200000; *Nombre d'iteration;
  sand = j(n, n, 0); *Creation d'une matrice vide;
  s = j(1, z, 0); *Compte les avalanches;
  t = j(1, z, 0); *Compte le temps des avalanches;

  /*On cree une boucle pour etaler le sable*/
  do i=1 to z;
    x = ceil(n * uniform(1));
    y = ceil(n * uniform(1));
    sand[x,y] = sand[x,y] + 1; *On ajoute un grain sur une case aleatoire;
    do while(max(sand) >= 4); * jusqu'a que le tas soit stable;
      ind = loc(sand >= 4); *On trouve les cases ou il y a 4 grains ou plus;
      if ncol(ind) > 0 then do;
        s[i] = s[i] + ncol(ind); *Compte les avalanches;
        t[i] = t[i] + 1; *Compte le temps des avalanches;
        sand[ind] = sand[ind] - 4; *La case perd 4 grains de sables;
        /* On redistribut ces grains au cases voisines : */
        gauche = loc(mod(ind, n) ^= 1); *pas sur la premiere colonne;
        droite = loc(mod(ind, n) ^= 0); *pas sur la derniere colonne;
        haut = loc(ind > n); *pas sur la premiere ligne;
        bas = loc(ind <= n*(n-1)); *pas sur la derniere ligne;
        /* redistribution */
        if ncol(gauche) > 0 then sand[ind[gauche] - 1] = sand[ind[gauche] - 1] + 1;
        if ncol(droite) > 0 then sand[ind[droite] + 1] = sand[ind[droite] + 1] + 1;
        if ncol(haut) > 0 then sand[ind[haut] - n] = sand[ind[haut] - n] + 1;
        if ncol(bas) > 0 then sand[ind[bas] + n] = sand[ind[bas] + n] + 1;
      end;
    end;
  end;
```

Dans cette seconde approche, les grains de sable ne sont plus déposés au centre du plateau, mais sur une case choisie **aléatoirement** à chaque itération.

Contrairement au modèle centré, où l'évolution est entièrement déterministe, le modèle aléatoire nécessite au minimum deux boucles :

- la première pour déposer **successivement** les grains de sable,
- la seconde pour gérer les effondrements locaux jusqu'à stabilisation du système.

Chaque ajout de grain peut déclencher une avalanche affectant un nombre variable de sites. En enregistrant, pour chaque itération s et t nous pouvons construire les distributions empiriques $D(s)$ et $D(t)$, présentées précédemment avec et sans transformation log log (cf. 1.2).

L'objectif de notre simulation est d'estimer la pente de la distribution en log log de $D(s)$ par une **régression linéaire** sur la première partie de la courbe log $D(s)$ afin de vérifier la validité du modèle. Une pente proche de -1 témoigne d'une **loi de puissance** caractéristique du **bruit rose** ($1/f$).

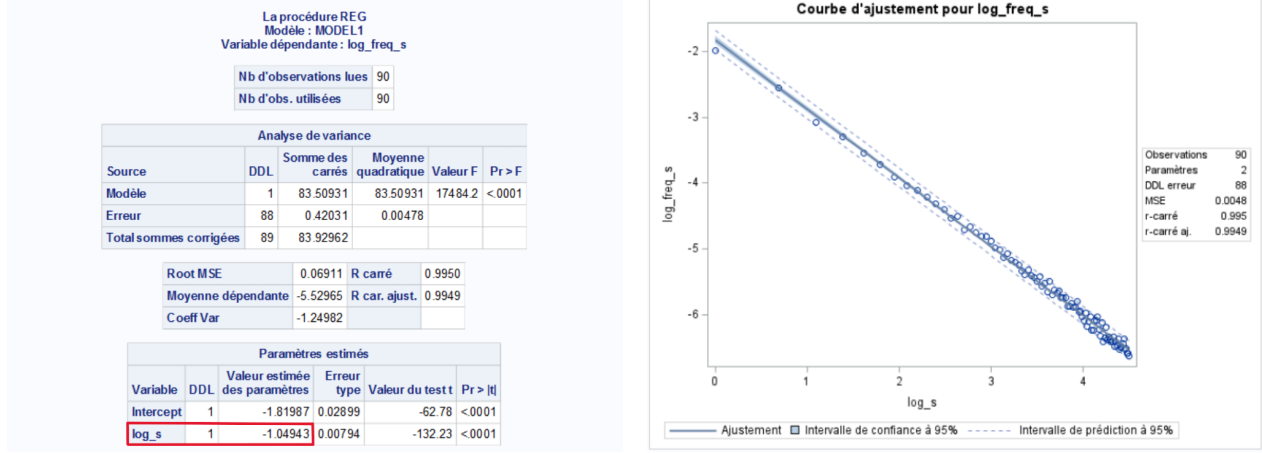


FIGURE 6 – Régression sur la fréquence de la taille des avalanches en log log (max taille = 90)

Sur les figures ci-dessous, nous présentons les résultats de la régression linéaire effectuée sur la courbe $\log(D(s))$. L'analyse a été restreinte aux avalanches de taille inférieure ou égale à 90, ce qui correspond à l'intervalle $\log(s) \in [0, 4.5]$ (puisque $\log(90) \approx 4.5$).

Les avalanches de très grande taille ont été volontairement exclues, car elles sont **biaisées par le caractère discret de la simulation**, notamment en raison de la **taille finie de la matrice** utilisée.

Le coefficient de la pente obtenu est égale à -1.04943 ce qui est relativement proche de 0, nous avons donc bien une **loi en puissance** :

$$D(s) \propto s^{-\tau}, \quad \text{avec } \tau \approx 1.$$

Ce qui traduit bien la présence d'un **bruit rose** $S(f) \propto f^{-\beta}$, avec $\beta \approx 1$, qui est la signature d'un comportement **auto-organisé critique**.

Nous avons donc, grâce à nos observations, validé les prédictions du bruit $1/f$ et de la SOC faites par BTW sur le modèle du tas de sable.

5 Extensions : Les tremblements de Terre et la SOC

Les tremblements de Terre sont un bon exemple de ce qu'on appelle la SOC. Comme dans le modèle du tas de sable de Bak, Tang et Wiesenfeld, le système accumule peu à peu de l'énergie jusqu'à atteindre un point de rupture. Quand ce seuil est dépassé, tout se réorganise brutalement : une avalanche pour le sable, et le séisme pour la Terre.

Les plaques tectoniques se déplacent lentement et exercent une pression continue sur les failles. Cette compression correspond à la phase lente, l'énergie augmente, mais rien ne se passe encore en surface. Puis, dès qu'une faille atteint son seuil de résistance, elle casse d'un coup : c'est la phase rapide, la libération de l'énergie sous la forme de tremblement de terre. Ce va-et-vient entre lente accumulation et relâchement soudain permet à la Terre de rester naturellement proche d'un état critique, sans qu'aucun paramètre externe ne soit ajusté.

Les scientifiques ont aussi remarqué que la fréquence des séismes suit une règle, appelée la loi de Gutenberg-Richter. Cette loi dit que les petits séismes sont très fréquents, alors que les gros sont rares, mais tous ont la même distribution. C'est exactement le comportement typique des systèmes auto-organisés critiques. On retrouve la même logique que dans le modèle du tas de sable, ce qui renforce l'idée que la sismicité terrestre est un phénomène.

Enfin, le modèle d'Olami, Feder et Christensen (1992) adapte ce principe au cas des séismes, chaque portion de faille est représentée comme une cellule qui accumule du stress. Quand elle dépasse son seuil critique, elle transmet une partie de ce stress à ses voisins, ce qui peut déclencher d'autres ruptures. Cela créer une réaction en chaîne comparable à une avalanche dans le tas de sable.

En Conclusion, un tremblement de terre, c'est comme une avalanche dans un tas de sable à échelle de la planète. La terre accumule lentement de l'énergie, puis la relâche sous forme de séisme. Ce fonctionnement, à la fois simple et instable, nous montre comment un système physique peut s'organiser tout seul à la frontière entre stabilité et chaos

6 Codes annexes

Vous trouverez ci dessous les codes complets du Modèle centré et aléatoire : Sandpile Aléatoire
Sandpile Centré

Si les documents ne se téléchargent pas, vous pouvez utiliser les copier coller des codes :

Listing 3 – Code Sandpile aleatoire

```
proc iml;

    n = 80; *Taille de la matrice : 201 pour centr e et 20 pour al atoire;
    z = 200000; *Nombre d'it ration : ' 100 000 pour centr e et 20 000 pour
        al atoire;
    sand = j(n, n, 0); *Cr ation d'une matrice vide;
    s = j(1, z, 0); *Compte les avalanches;
    t = j(1, z, 0); *Compte le temps des avalanches;

    /*On cr e une boucle pour taler le sable*/
    do i=1 to z;
        x = ceil(n * uniform(1));
        y = ceil(n * uniform(1));
        sand[x,y] = sand[x,y] + 1; *On ajoute un grain sur une case al atoire;
        do while(max(sand) >= 4); * jusqu' que le tas soit stable;
            ind = loc(sand >= 4); *On trouve les cases ou il y a 4 grains ou
                plus;
            if ncol(ind) > 0 then do;
                s[i] = s[i] + ncol(ind); *Compte les avalanches;
                t[i] = t[i] + 1; *Compte le temps des avalanches;
                sand[ind] = sand[ind] - 4; *La case perd 4 grains de sables
                ;
                /*On redistribut ces grains au cases voisines*/
                gauche = loc(mod(ind, n) ^= 1); *pas sur la premi re
                    colonne;
                droite = loc(mod(ind, n) ^= 0); *pas sur la
                    derni re colonne;
                haut = loc(ind > n); *pas sur la premi re ligne;
                bas = loc(ind <= n*(n-1)); *pas sur la derni re
                    ligne;
                /* redistribution */
                if ncol(gauche) > 0 then sand[ind[gauche] - 1] =
                    sand[ind[gauche] - 1] + 1;
                if ncol(droite) > 0 then sand[ind[droite] + 1] =
                    sand[ind[droite] + 1] + 1;
                if ncol(haut) > 0 then sand[ind[haut] - n] = sand[
                    ind[haut] - n] + 1;
                if ncol(bas) > 0 then sand[ind[bas] + n] = sand[ind
                    [bas] + n] + 1;
            end;
        end;
    end;

    title "Reprsentation du tas de sable apr s n it ration";
    call heatmapcont(sand) xvalues=vname yvalues=vname colorramp={White
        CXFFEDC7 CXFBE646 CXFBC34D Orange} displayoutlines=0;
    title "Fr quence de la taille des avalanches";
    call histogram(s) scale='PROPORTION';
    title "Fr quence de la dur e des avalanches (nombres d'it ration avant
        stabilit )";
```

```

call histogram(t) scale='PROPORTION';

/*Conversion des donn es de taille pour les graphiques*/
s_nonzero = s[loc(s > 0)];
call tabulate(valeurs, effectif, s_nonzero);
frequence = effectif/sum(effectif);

/*Conversion de la ditribution de taille des avalanches en log log pour la
r gression*/
log_freq_s = log(effectif/sum(effectif));
log_s = log(valeurs);

/*Cr ation d'un data pour exporter en dehors de iml*/
create distlogs var {"log_freq_s" "log_s" "valeurs" "frequence"};
append;
close distlogs;

/*Conversion des donn es du temps pour les graphiques*/
t_nonzero = t[loc(t > 0)];
call tabulate(valeurs, effectif, t_nonzero);
frequence = effectif/sum(effectif);

/*Cr ation d'un data pour exporter en dehors de iml*/
create distlogt var {"frequence" "valeurs"};
append;
close distlogt;

quit;

proc sgplot data=distlogs;
scatter x=valeurs y=frequence / markerattrs = (symbol=circlefilled);
axis type=log label="Taille t";
axis type=log label="P(S=s)";
title "Fr quence de la taille des avalanches (log log)";
run;

proc sgplot data=distlogt;
scatter x=valeurs y=frequence / markerattrs=(symbol=circlefilled);
axis type=log label="Dur e t" minor display=(nolabel);
axis type=log label="P(T=t)" minor display=(nolabel);
title "Distribution de la dur e des avalanches (log-log)";
run;

title "Regression lin aire sur la taille des avalanches en log log avec
max_taille_avalanche = 90";
proc reg data=distlogs(where=(log_s <= 4.5)); *car exp(4.5) = 90;
model log_freq_s = log_s;
run;

```

Listing 4 – Code Sandpile Centr é

```

proc iml;

n = 301; *Taille de la matrice : 201 pour centr e et 20 pour al atoire;
sand = j(n, n, 0); *Cr ation d'une matrice vide;
center = ceil((n+1) / 2); *Centre de la matrice;

/*Centr */
sand[center,center] = 100000;
/*Centres des quart de matrice*/
*sand[center*0.5,center*0.5] = 200000;
*sand[center*1.5,center*1.5] = 200000;
*sand[center*0.5,center*1.5] = 200000;
*sand[center*1.5,center*0.5] = 200000;
/*Sur les 4 sommets*/
*sand[1,1] = 5000000;

```

```

*sand[n,n] = 5000000;
*sand[1,n] = 5000000;
*sand[n,1] = 5000000;

*On cr e une boucle pour taler le sable;
do while(max(sand) >= 4); * jusqu'a que le tas soit stable;
ind = loc(sand >= 4); *On trouve les cases ou il y a 4 grains ou
plus;
    if ncol(ind) > 0 then do;
        sand[ind] = sand[ind] - 4; *La case perd 4 grains;
        /* On redistribut ces grains au cases voisines : */
        gauche = loc(mod(ind, n) ^= 1); *pas sur la premiere
        colonne;
        droite = loc(mod(ind, n) ^= 0); *pas sur la
        derniere colonne;
        haut = loc(ind > n); *pas sur la premi re ligne;
        bas = loc(ind <= n*(n-1)); *pas sur la derniere
        ligne;
        /* redistribution */
        if ncol(gauche) > 0 then sand[ind[gauche] - 1] =
            sand[ind[gauche] - 1] + 1;
        if ncol(droite) > 0 then sand[ind[droite] + 1] =
            sand[ind[droite] + 1] + 1;
        if ncol(haut) > 0 then sand[ind[haut] - n] = sand[
            ind[haut] - n] + 1;
        if ncol(bas) > 0 then sand[ind[bas] + n] = sand[ind
            [bas] + n] + 1;

    end;

end;

title "Reprsentation du tas de sable apres n iteration";
call heatmapcont(sand) xvalues=vname yvalues=vname colorramp={cx8B0000
    cxC62828 cxFF6F00 cxFFA726 cxFFD54F} displayoutlines=0;

```

Références

- [1] *Sandpiles and the Abelian Sandpile Model* : https://doc.sagemath.org/html/en/thematic_tutorials/sandpile.html, section Laplacian.
- [2] *Revue officielle de Olami-Feder-Christensen* : <https://pubmed.ncbi.nlm.nih.gov/10046116/>
- [3] *Gutenberg-Richter Relationship : Magnitude vs. frequency of occurrence* : https://www.eoas.ubc.ca/courses/eosc256/Feb7_2011_GR.pdf
- [4] *The Olami-Feder-Christensen Model on a Small World topology* : <https://indico.ictp.it/event/a05215/session/39/contribution/24/material/0/1.pdf>