

## NYX DE KERRY HAGAN : LE CHAOS COMME SOURCE D'INSPIRATION

Maxence Larrieu  
LISAA, Université Gustave Eiffel  
maxence@larri.eu

### RÉSUMÉ

*nyx* est une pièce pour ordinateur composée en 2017 par la chercheuse et compositrice Kerry Hagan, émanant d'une profonde inspiration pour le chaos. Dans la mythologie grecque, Nyx (la nuit), sœur d'Érèbe (les ténèbres), est née du Chaos, lequel précède la création de la Terre. Cette inspiration est patente au niveau du design sonore, où la compositrice utilise deux fonctions chaotiques, *Gingerbread Man* et *Ikeda*, qui vont être l'unique source sonore de la pièce. Dans cet article, nous apportons une analyse musicale de *nyx*. Nous utilisons pour cela deux matériaux clés : le patch *Pure Data* et le signal audio de la pièce. Dans un premier temps, nous éclairons le patch, lieu où la composition s'est réalisée, et dans un second nous analysons la pièce dans sa forme canonique, entre structure, rendus sonores et musicaux.

### 1. INTRODUCTION

Cet article propose une analyse d'une pièce pour ordinateur, *nyx*, composée par Kerry Hagan en 2017 à partir d'une profonde inspiration pour le chaos. Nous souhaitons introduire cette pièce en trois temps, d'abord en contextualisant cette inspiration fondatrice pour le chaos avec d'autres pièces de la compositrice, ensuite en éclairant trois niveaux de chaos présents dans *nyx*, et enfin en expliquant notre méthode d'analyse.

Kerry Hagan est chercheuse et compositrice basée à l'Université de Limerick (Irlande). Depuis plus de dix ans son travail se situe dans le champ de la création musicale avec ordinateur, avec un catalogue d'une vingtaine d'œuvres comportant des pièces mixtes et des pièces pour ordinateur seul, aussi bien improvisées que fixées sur support. Dans ce catalogue, nous distinguons une série de quatre pièces pour ordinateur qui ont pour point commun d'être composées à partir d'une profonde inspiration pour le bruit : *Morphons & Bions* (2011), *...of pulses and times...* (2012), *s/d* et *nyx* (2017). Cette série a pour origine un changement majeur dans la technique compositionnelle : celui de ne plus utiliser uniquement des *samples* (sons enregistrés), mais d'aller vers la synthèse « pure ». La compositrice contextualise cette série de la façon suivante : « Les sons enregistrés formaient la base du matériau musical de mes précédentes pièces, principalement du fait de leur richesse timbrale et de leur complexité spectrale. Les sons synthétiques sont souvent secs, minces, plastiques et artificiels. Cepen-

dant, les philosophies qui amenaient à composer avec des sons enregistrés se sont petit à petit épuisées. Il devenait nécessaire de changer d'approche. Par conséquent, c'est la synthèse sonore qui est devenue le nouveau défi musical<sup>1</sup>. »

Relevons que cette association entre platitude et sons électroniques est aussi présente vingt ans plus tôt chez Philippe Manoury. Dans *Jupiter* (1987), afin de faire face à la propreté des signaux des modules de synthèse, Manoury et Puckette ont recours d'une part à un ensemble de modules de traitements – *frequency shifter*, *harmoniser*, *reverb*, *noise*, *phaser* –, d'autre part à un *routing* très flexible permettant de les interconnecter à loisir [8]. C'est ainsi que les signaux trop lisses vont pouvoir être bruités et enrichis.

C'est avec cette volonté de se détacher des signaux synthétiques trop lisses que Kerry Hagan va développer une profonde inspiration pour le bruit : « Après réflexion, j'ai décidé que c'était le bruit des sons acoustiques qui les rendaient riches : bruit légers dans les articulations, ou divergences et caractéristiques non prévisible dans le spectre. J'ai donc abordé la synthèse sonore avec le bruit, en l'utilisant comme le fondement de toutes les méthodes de synthèse<sup>2</sup>. »

Nous qualifions cette inspiration de profonde, car il ne s'agit pas simplement d'utiliser le bruit, mais bien plus de changer de mode de conception : considérer le bruit non plus comme quelque chose d'indésirable<sup>3</sup> mais comme la source de la richesse infinie des sons instrumentaux. Comme nous le verrons, ce changement de conception se retrouve à plusieurs niveaux. Pour ce qui est du design sonore par exemple, il ne va plus être

---

<sup>1</sup> « In my previous works, recorded sounds formed the basis of musical material, since recorded acoustic sounds have a richness of timbre and spectral complexity. Synthesized sounds often seem dry, thin, plastic, and artificial. However, the philosophies that engendered working with the recorded sounds exhausted themselves. It became necessary to turn to new approaches. Therefore, taking on synthesis appeared to be the next musical challenge. » [6] Voir également [5].

<sup>2</sup> « After some consideration, I decided that it was the noise of the acoustic sounds that made them richer : slight noises in articulation, or inconsistencies and unpredictable characteristics in the spectra. So, I approached synthesis from the basis of noise, using noise as the foundation of every synthesis method. » [5]

<sup>3</sup> Le bruit, dans la théorie de l'information, est un indésirable qui vient perturber la transmission d'un signal [12]. Cette conception s'applique plus généralement à la technologie : le bruit est ce qui vient altérer la qualité d'un signal originel.

question de bruiteur un signal source, mais d'utiliser le bruit directement dans la synthèse <sup>4</sup>.

*nyx* existe au travers de plusieurs versions. La plus conventionnelle est la version stéréophonique partagée sur le site web de la compositrice <sup>5</sup>, où l'on trouve les deux matériaux clés, le patch *Pure Data*, lieu où la composition s'est réalisée, et un enregistrement stéréophonique. D'autres versions existent en fonction des lieux et des systèmes de diffusion où la pièce a été jouée. Une version pour 124 canaux a ainsi été créée pour la salle du Cube de la *School of Performing Arts* (Virginia Tech, États-Unis).

L'inspiration pour le chaos a été essentielle dans la composition de *nyx*. Nous proposons de distinguer trois niveaux. Le premier est celui du design sonore, où la compositrice va directement utiliser des fonctions mathématiques chaotiques pour produire les signaux de la pièce. Deux fonctions sont utilisées, nommées *Gingerbread Man* et *Ikeda*, qui constituent les uniques producteurs de signaux de toute la pièce.

Le deuxième niveau concerne la mésostructure – une échelle intermédiaire entre la micro- et la macrostructure. Pour construire la pièce, la compositrice a recours à de nombreux algorithmes aléatoires qui vont définir un champ des possibles, au sein duquel un amas d'événements sonores va se déployer. De cette façon, toutes les réalisations de la pièce sont différentes : une suite identifiable d'événements sonores présents dans la version partagée sur le site web pourra ainsi ne plus être présente dans une version réalisée par l'auditeur directement avec le patch. Cependant, du fait 1) du design sonore employé, 2) de la structure de la pièce et 3) de la délimitation limpide du champ des possibles, *nyx* reste identifiable d'une réalisation à l'autre. Ainsi, ce que la compositrice relève pour une autre pièce, *Morphons & Bions*, s'applique tout à fait à *nyx* : « En tant que musique électronique en temps réel, les détails de chaque réalisation changent d'une performance à l'autre. Cependant, la cohérence timbrale et la structure globale de la pièce marquent l'identité de la pièce <sup>6</sup> ».

Le troisième et dernier niveau concerne l'esthétique et la macrostructure de la pièce. En cohérence avec les précédentes fonctions chaotiques, la compositrice trouve son inspiration dans la mythologie grecque d'où est issue le nom de la pièce. En effet, dans la *Théogonie* d'Hésiode, Nyx est la déesse de la Nuit. Avec son frère Érebe, dieux des ténèbres, Nyx est directement née du Chaos, lequel précède l'origine du monde et les divinités grecques. La structure de *nyx* s'assimile aisément à ce processus de création, avec un moment chaotique, sans prise pour l'auditeur, et un autre fait de gestes construits et saillants.

Avant de procéder à l'analyse musicale de la pièce, il nous faut préciser notre méthode d'analyse. Analyser une pièce comme *nyx*, c'est-à-dire une pièce pour ordinateur avec des processus aléatoires, pose une question centrale que nous pouvons formuler ainsi : où se trouve la pièce ? Se trouve-t-elle dans l'écoute *in situ* des importants systèmes de spatialisation ? Dans la version stéréophonique partagée par la compositrice ? Ou bien dans le patch *Pure Data* de cette dernière ?

Il serait tentant d'analyser *nyx* comme une pièce de musique électroacoustique et d'utiliser le fichier audio seul comme objet d'analyse <sup>7</sup>. Mais une telle pratique pose problème pour des pièces contenant des processus aléatoires. En effet, nous avons parfois entendu dans *nyx* des intervalles de sixte et de quarte limpides. Mais à l'écoute d'une autre réalisation, ces derniers étaient simplement absents. Pour rendre compte de cela, notre méthode d'analyse utilise les deux supports clés des pièces d'informatique musicale, à savoir le code et le (ou les) signaux audio de la pièce. *nyx* existe en effet aussi bien dans le code – le patch *Pure Data* support de la composition – que dans le signal audio – support le plus simple permettant l'écoute [7]. Notre analyse se compose ainsi de deux parties : une première où nous rendons compte du patch – que nous préférons appeler artéfact, afin notamment que la pratique qui consiste à lire et comprendre du code ne soit pas réservée au domaine de l'informatique – et une seconde où nous analysons la musique dans sa forme canonique, en éclairant les rendus musicaux et la structure de la pièce. De plus, pour ne pas se focaliser sur une seule réalisation, nous avons effectué l'analyse à partir de trois réalisations : celle partagée sur le site web de la compositrice et deux autres réalisées par nous-mêmes avec le patch. Enfin, le lecteur aura compris que nous analysons ici *nyx* dans sa forme stéréophonique, et non à partir d'une version *ad hoc* réalisée pour des systèmes de spatialisation <sup>8</sup>.

## 2. L'ARTÉFACT

« Ouvrez `_StartHere.pd`, activer le DAC et paramétrez le volume à une valeur non nulle. Cliquez ensuite sur *Start* pour une lecture automatique. *Reset* arrête la lecture et réinitialise les valeurs <sup>9</sup>. » Nous décrivons ci-après l'artéfact avec une certaine continuité, aussi présente dans la pièce, en commençant par les éléments les plus chaotiques.

<sup>4</sup> Par exemple dans *Morphons & Brions* (2011), une technique consiste à moduler un signal sinusoïdal par un oscillateur de bruit blanc [5].

<sup>5</sup> Voir [kerrylhagan.net](http://kerrylhagan.net), accédé le 12/02/2020.

<sup>6</sup> « *As real-time computer music, the details of each realization change from performance to performance. However, the consistent timbres and overall form of the work retain the identity of the piece.* » [5]

<sup>7</sup> Ou bien si *nyx* était pleinement une pièce de musique générative, d'étudier uniquement son code. Voir [1] pour un exemple.

<sup>8</sup> Le patch que nous utilisons a été archivé et partagé publiquement par la compositrice sur un entrepôt de données de la recherche, de sorte que chacun puisse télécharger, partager et citer de façon univoque la même version du patch. Voir [doi.org/10.5281/zenodo.3971610](https://doi.org/10.5281/zenodo.3971610).

<sup>9</sup> « *Open \_StartHere.pd, turn the DAC on by setting the output level to a non-zero value. Then click the Start button for automatic playback. Reset will stop the piece and return to starting values.* » (Extrait du fichier `_readmeFirst.txt` de l'archive du patch.)

## 2.1. Synthèse chaotique

Le matériau premier de la pièce, celui duquel tous les signaux sont produits, vient du module Synthèse chaotique<sup>10</sup>. Ce dernier utilise deux fonctions mathématiques chaotiques, Gingerbread Man et Ikeda, dont les expressions mathématiques sont données par les figures 1 et 2.

$$\begin{aligned} x_{n+1} &= 1 - y_n + |x_n| \\ y_{n+1} &= x_n \end{aligned} \quad \text{Gingerbread Man}$$

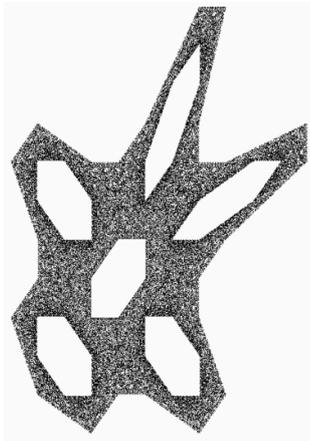
**Figure 1.** Fonction Gingerbread Man.

$$\begin{aligned} x_{n+1} &= 1 + u(x_n \cos t_n - y_n \sin t_n), \\ y_{n+1} &= u(x_n \sin t_n + y_n \cos t_n), \end{aligned} \quad \text{Ikeda}$$

where  $u$  is a parameter and

$$t_n = 0.4 - \frac{6}{1 + x_n^2 + y_n^2}.$$

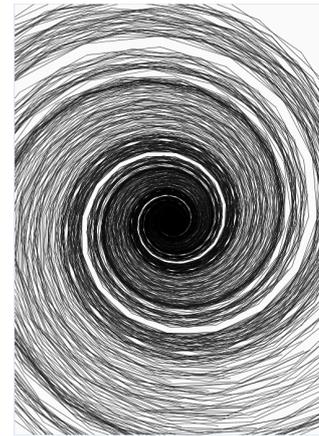
**Figure 2.** Fonction Ikeda.



```
size(500, 700);
background(250);
int margin = 20;
FloatList xn = new FloatList(), yn = new FloatList();
xn.append(-0.1);
yn.append(0.1);

int iterations = 100000;
for (int i=1 ; i<iterations ; i++) {
    float xnext = 1 - yn.get(i-1) + abs(xn.get(i-1));
    float ynext = xn.get(i-1);
    xn.append(xnext);
    yn.append(ynext);
}
for (int i=0; i< iterations; i++) {
    float x = map(xn.get(i), xn.min(), xn.max(), margin, width - margin);
    float y = map(yn.get(i), yn.max(), yn.min(), margin, height - margin);
    point(x, y);
}
```

**Figure 3.** La fonction Gingerbread Man et son implémentation dans Processing.



```
size(500, 700);
background(250);
strokeWeight(0.7);

int iterations = 1000;
int p = 200;
float tn;
float u = 0.94;
float[] xn = new float[iterations];
float[] yn = new float[iterations];
int margin = 100;

translate(width/2, height/2);
for (int k=0; k<p; k++) {
    xn[0] = random(width-margin);
    yn[0] = random(height-margin);
    float c = 30;

    for (int i=1; i<iterations; i++) {
        tn = 0.4 - (6 / (1 + pow(xn[i-1], 2) + pow(yn[i-1], 2)));
        xn[i] = 1 + u * (xn[i-1] * cos(tn) - yn[i-1] * sin(tn));
        yn[i] = u * (xn[i-1] * sin(tn) + yn[i-1] * cos(tn));
        if (i%10 == 0) c-- 10;
        stroke(c);
        line(xn[i-1], yn[i-1], xn[i], yn[i]);
    }
}
```

**Figure 4.** La fonction Ikeda et son implémentation dans Processing.

Comme ces fonctions possèdent deux dimensions, il est d'usage d'accompagner leur description de représentations graphiques. Afin de mieux les appréhender, nous les avons implémentées dans le langage de programmation Processing. La figure 3 représente la fonction Gingerbread Man. Elle a été obtenue avec 100 000 itérations, où chaque résultat devient un point de coordonnées  $(x_n, y_n)$ . Comme nous pouvons le voir, le « bonhomme en pain d'épices » apparaît si l'on effectue une rotation de  $135^\circ$  dans le sens des aiguilles d'une montre.

La figure 4 représente la fonction Ikeda. Elle a été obtenue en répétant 200 fois une itération de 1000 pas, et en reliant chaque point successif,  $(x_n, y_n)$  à  $(x_{n+1}, y_{n+1})$ . La fonction Ikeda ressemble à une spirale, avec cette relation pour le paramètre  $u$  : plus sa valeur est élevée plus la spirale est enroulée.

<sup>10</sup> Nous ajoutons une majuscule aux termes désignant les modules de la composition de façon à les différencier des techniques communes à l'informatique musicale, comme la synthèse formantique.

2.1.1. Comment cela sonne-t-il ?

L'utilisation des fonctions chaotiques<sup>11</sup> est plutôt originale puisque la compositrice, afin de produire les signaux, va directement utiliser les résultats des fonctions comme valeurs d'échantillons. Les implémentations fonctionnent à la fréquence d'échantillonnage de 44,1 kHz. Ceci est réalisé grâce à deux objets *Pure Data*, *block~* et *metro*<sup>12</sup> (figure 5).

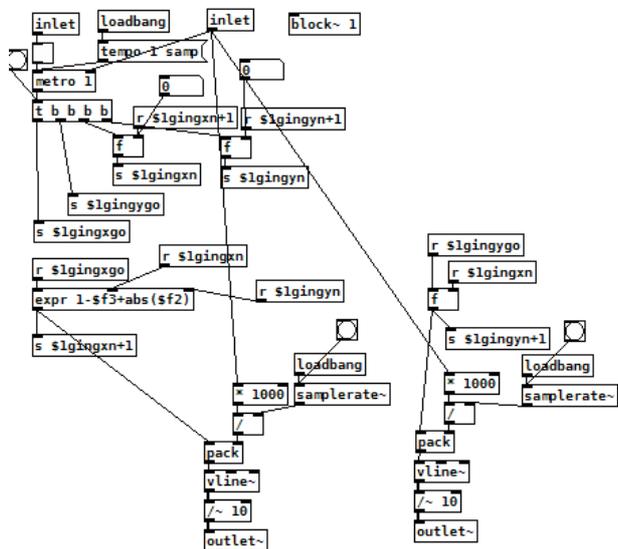


Figure 5. La fonction *Gingerbread Man* implémentée dans *Pure Data*.

Si les rendus graphiques renvoient à des formes, les résultats sonores renvoient fondamentalement à du bruit, à quelque chose d'insaisissable. Comme son succédané bruit blanc, le signal produit ne possède aucun composant perceptuel comme une sinusoïde, un formant, une organisation fréquentielle ou un pattern temporel. Dans le sonagramme de la fonction *GingerBreadMan* (figure 6), nous pouvons observer un certain pattern, mais celui-ci reste imperceptible à l'écoute : ce qui est perçu est un signal très aigu – la ligne vers 9,5 kHz – composé d'une myriade de mouvements infinitésimaux.

Précisons que ce signal a pu être décrit en l'extrayant du patch, pour servir l'analyse, mais qu'il n'est jamais diffusé tel quel dans *nyx*. En effet, pour atteindre une richesse musicale la compositrice a implémenté trois fonctions *GingerBreadMan* et deux fonctions *Ikeda*, chacune étant paramétrée différemment. Dès lors, comme ces fonctions possèdent deux dimensions, le module Synthèse chaotique produit pas moins de dix signaux. Cette masse chaotique est l'unique source sonore de la pièce, les dix signaux qui la composent étant traités, découpés et retraités afin de créer différents événements musicaux.

<sup>11</sup> Voir [11, 13, 14] pour d'autres exemples de synthèse sonore utilisant le chaos.

<sup>12</sup> Le métronome de *Pure Data* a ceci de particulier qu'il peut être paramétré en valeur d'échantillon, représentant la période des impulsions – comme en témoigne le message `tempo 1 samp` dans le patch.

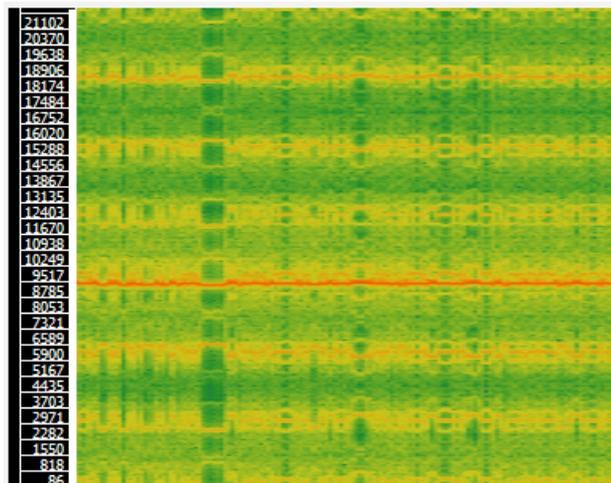


Figure 6. Sonagramme du signal produit par la fonction *GingerBreadMan*. Durée : 2 s. Axe vertical : Hz. Toutes les analyses spectrales sont réalisées avec une fenêtre de Hann (durée : 1024 samples ; threshold : -43 dB).

2.1.2. Contrôle

Nous terminons la description de ce module en précisant l'ensemble des contrôles, c'est-à-dire les points de contacts où la compositrice exerce une action<sup>13</sup>.

L'implémentation des fonctions appelle d'abord une série de paramètres de bas niveau, dans le sens où ils impactent peu l'écoute. Il s'agit des variables  $x_0$  et  $y_0$  et du paramètre  $u$ . Ils sont néanmoins importants car c'est avec ceux-ci que différents signaux peuvent être produits. Il y a ensuite deux contrôles de plus haut niveau, sur lesquels la compositrice intervient. Le premier concerne la fréquence de calcul des fonctions. Comme nous l'avons vu, elles sont initialement paramétrées pour fonctionner à la fréquence d'échantillonnage, mais la compositrice va les modifier de sorte à produire d'importants glissandos de l'ordre de 10 secondes, qui vont être les gestes structurants de la pièce. Il y a ainsi une variable pour chaque implémentation, nommée *lgingrate*, *2gingrate*, etc., qui impacte la fréquence de calcul de ces fonctions. Le deuxième contrôle utilisé impacte l'amplitude : tous les signaux sont attachés à un paramètre d'amplitude, qui va permettre de découper des événements dans cette masse chaotique.

2.2. Spatialisateur

Nous abordons ensuite le module *Spat*, car il se trouve très proche de l'inspiration pour le chaos. La spatialisation relève en effet d'un désordre organisé<sup>14</sup>, et non

<sup>13</sup> Avec Marie-Élisabeth Duchez [2], nous comprenons que ces points de contacts où s'exercent les actions compositionnelles sont inévitablement en lien avec le matériau musical.

<sup>14</sup> L'œuvre de Iannis Xenakis a beaucoup inspiré la compositrice : « Pour Xenakis, les méthodes sérielles utilisées par ses contemporains lui paraissaient hasardeuses. Il affirmait que la logique interne des systèmes sériels n'était pas audible, qu'elle aurait aussi bien pu être produite par le hasard. Xenakis a donc substitué aux systèmes sériels des processus aléatoires, tout en contrôlant le niveau de désordre. » [6] (« *To Xenakis the serial methods used by his contemporaries appeared*

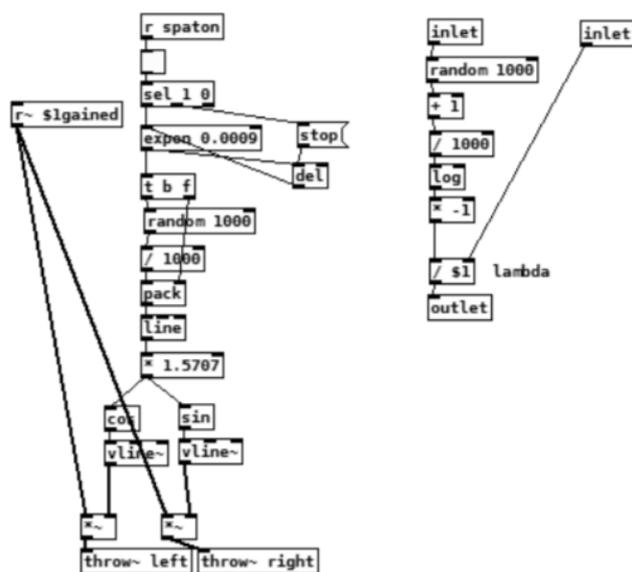
du fin placement d'un son dans l'espace stéréophonique. Le Spat fonctionne ainsi sur toute la durée de la pièce, et les positionnements des signaux – ou plus exactement leurs déplacements – sont régis par un algorithme autonome. Ce dernier génère une paire de valeurs ( $x$ ,  $y$ ), avec  $x$  représentant la position dans l'espace stéréophonique et  $y$  la durée pour l'atteindre. Le Spat applique indépendamment ces déplacements sur les dix signaux de la Synthèse chaotique et les dix autres d'un dernier module que nous verrons par la suite.

```
run randomDur() and spat() functions, wait for
duration and start them again, etc.
```

```
randomDur() function is
d = random(0.001, 1) #step of 0.001
duration = -log(d) / 0.0009
```

```
spat() function is
x = random (0.001, 0.999) #step of 0.001
smoothx = line(x, duration)
```

```
leftChannel = cos( smoothx * PI/2)
rightChannel = sin( smoothx * PI/2)
```



**Figure 7.** L'algorithme du Spat : pseudo code (en haut) ; patch général (en bas à gauche) ; abstraction *expog* (en bas à droite).

La figure 7 propose une description de cet algorithme, entre pseudo-code et élément du patch. La fonction exponentielle permet à la compositrice d'avoir un contrôle sur la durée moyenne des résultats [3], qui est ici de l'ordre de 770 millisecondes ( $0,69315/0,0009$ ). Les déplacements spatiaux se font donc à l'échelle de la note, mais ils ne seront pas nettement perceptibles puisque le module applique vingt déplacements en parallèle.

### 2.3. Routing

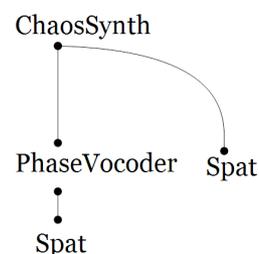
Nous pouvons à présent traiter du *routing*, des connexions entre modules. Un module n'a cependant

---

*to be random. He claimed that the inner logic of serial systems was not audible, that they might as well have been random. So, Xenakis proposed using randomness instead, while controlling the disorder through random processes.* »)

pas encore été abordé : le PhaseVocoder, qui permet de modifier la durée et la hauteur d'un signal entrant. Le *routing* réalisé s'inspire aussi du bruit puisqu'il va permettre de démultiplier les signaux produits par la Synthèse chaotique.

Le *routing* se décompose en un ensemble de dix canaux, lesquels sont formés d'une part d'une connexion de la Synthèse chaotique vers le Spat, d'autre part d'une autre connexion de la Synthèse chaotique vers le PhaseVocoder, lequel est lui-même relié au Spat (figure 8).



**Figure 8.** Le *routing* à l'échelle d'un canal.

Nous retrouvons ainsi un principe d'accumulation de signaux entre :

- 5 fonctions chaotiques ;
- 10 signaux produits par la Synthèse chaotique ;
- 20 signaux diffusés en parallèle.

La composition se fait donc avec pas moins de vingt signaux, chacun diffusés par le Spat. Nous comprenons pourquoi le contrôle de l'espace ne relève pas du pointillisme, mais davantage de l'organisation d'un ensemble désordonné de vingt déplacements simultanés.

### 2.4. PhaseVocoder

Le PhaseVocoder permet de modifier indépendamment la durée (*time stretching*) et la hauteur du signal entrant. L'implémentation est celle réalisée par Miller Puckette, décrite dans son célèbre livre *The Theory and Technique of Electronic Music* [10], et donné en exemple dans la bibliothèque de Pure Data<sup>15</sup>. Comme ce module correspond à l'exemple donné par M. Puckette<sup>16</sup> nous renvoyons le lecteur à son célèbre livre pour une explication technique et nous en décrivons ci-après l'usage qui a été fait par la compositrice.

Le PhaseVocoder est utilisé pour densifier et colorer la masse chaotique originelle. Cela se fait par l'intermédiaire de trois contrôles. Le premier, de bas niveau, est la vitesse de lecture des signaux – les variables *speed1*. Les signaux vont être modifiés une seule fois en divisant par 5 leurs vitesses de lecture. Les myriades de mouvements infinitésimaux qui les composaient vont ainsi être ralenties, ce qui va densifier la masse chaotique. Le deuxième contrôle concerne l'amplitude. Comme précédemment, sur chacun des dix signaux produits la compositrice ajoute un contrôle d'amplitude afin de tailler des événements. Le dernier contrôle concerne la hauteur. Afin d'ajouter du relief, la compositrice va transposer les dix signaux entrants.

<sup>15</sup> Le patch est nommé *I07.phase.vocoder.pd*.

<sup>16</sup> L'abstraction de Kerry Hagan s'appelle *millerpvocabstract*.

Comme ceux-ci sont déjà extrêmement aigus, toutes les transpositions vont se faire dans le grave, entre une transposition nulle et une maximale jusqu'à plus de six octaves.

## 2.5. Algorithmes sur l'amplitude

Afin d'extraire des événements sonores, la compositrice a recours à deux algorithmes d'amplitude similaires placés sur les modules. Ces algorithmes possèdent un comportement balistique simple et cyclique : deux valeurs sont générées au hasard, la première est une destination et la seconde une durée pour l'atteindre. Une fois la cible atteinte, les valeurs sont régénérées, et ainsi de suite.

Au niveau des échelles de valeur, les amplitudes sont générées au hasard dans une échelle de décibels comprise entre 46 et 104<sup>17</sup>, et les durées dans une échelle entre 1 et 5 secondes. Ajoutons enfin que chaque module intègre cinq algorithmes d'amplitude – un par fonction chaotique – avec leur propre échelle de décibels, ce qui autorise une plus grande variété (figure 9).

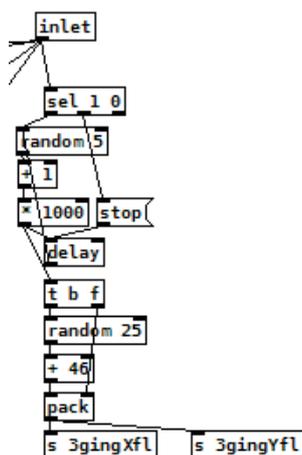


Figure 9. Algorithme d'amplitude.

## 2.6. Algorithmes sur les hauteurs

Les deux derniers algorithmes impactent la hauteur. Leur application et leur fonctionnement sont similaires aux précédents : ils sont appliqués aux contrôles de hauteur des modules de Synthèse chaotique et du PhaseVocoder, et leur fonctionnement reproduit le même comportement balistique.

Ces deux algorithmes sont conçus de sorte à pouvoir fonctionner en série. Le premier a pour rôle d'appliquer des glissandos sur les nombreux signaux de la masse chaotique et le second de doubler ces derniers avec des transpositions dans le grave, via le PhaseVocoder.

Le premier algorithme – activé par la variable *randrate* – est directement appliqué sur la fréquence de calcul des fonctions mathématiques, ce qui aura pour effet de baisser la hauteur perçue des signaux produits. Les résultats du premier algorithme correspondent ainsi

<sup>17</sup> Avec un pas de un, où 100 représente le gain unitaire.

aux fréquences des calculs, exprimées directement en nombre d'échantillons par période. Ce nombre est tiré au hasard dans l'intervalle [1, 100] avec un pas de 1. L'algorithme passe d'une valeur à l'autre linéairement, avec des durées de l'ordre de la phrase, entre 10 et 29 secondes.

Le deuxième algorithme – activé par la variable *detune* – impacte la valeur de transposition du PhaseVocoder, exprimé en cents. Elles sont tirées aléatoirement dans l'intervalle [-8000, 0]. La deuxième valeur générée représente la période temporelle avant l'attribution d'une nouvelle valeur de transposition. Les durées, quant à elles, ont la même échelle que précédemment, comprise entre 10 et 29 secondes.

## 2.7. Synoptique

Nous pouvons à présent apporter une vue synoptique de l'artéfact (figure 10). Nous proposons de la réaliser entre deux domaines connus en informatique musicale<sup>18</sup>, celui du signal – le *routing* – et celui du signe – les contrôles des modules sur lesquels la compositrice agit.

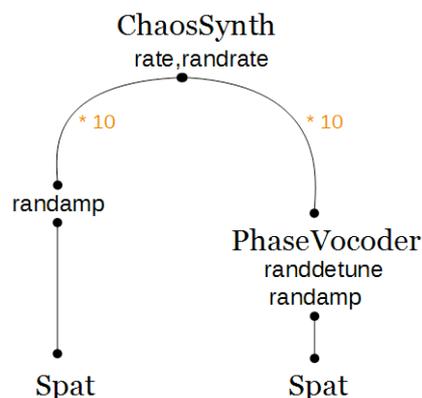


Figure 10. Vue synoptique de l'artéfact.

Le matériau de la pièce est maintenant éclairci. Nous pouvons différencier trois niveaux : en premier un substrat, une masse chaotique brute et bruitée dans laquelle la compositrice peut activer des groupes de glissandos ; en deuxième des événements sonores taillés depuis cette masse chaotique simplement à l'aide de l'amplitude ; en dernier une nouvelle extraction plus saillante, puisqu'elle se fait avec le PhaseVocoder qui peut agir sur trois dimensions – la durée, l'amplitude et la hauteur.

## 2.8. Structure de contrôle

Enfin nous pouvons éclairer une dernière fois l'artéfact avec sa structure de contrôle, c'est-à-dire en organisant les contrôles et modules de la composition (figure 11). L'organisation des modules procède d'une logique : plus un module est situé en haut plus il est important pour la pièce, c'est-à-dire qu'il agrège d'importantes significations musicales. En bas se trouve

<sup>18</sup> Voir par exemple le cadre conceptuel de Malloch et coll. sur les « instruments de musique numérique » (*Digital Musical Instruments*) [9].

donc le Spat, qui reste autonome sur la durée de la pièce, au milieu le PhaseVocoder et au plus haut la Synthèse chaotique, source originelle des matériaux de la pièce.

```

    ChaosSynth( amp, rate, randrate)
    PhaseVocoder( speed, amp, detune)
    Spat()
    
```

**Figure 11.** La structure de contrôle de l'artéfact.

### 3. ANALYSE MUSICALE

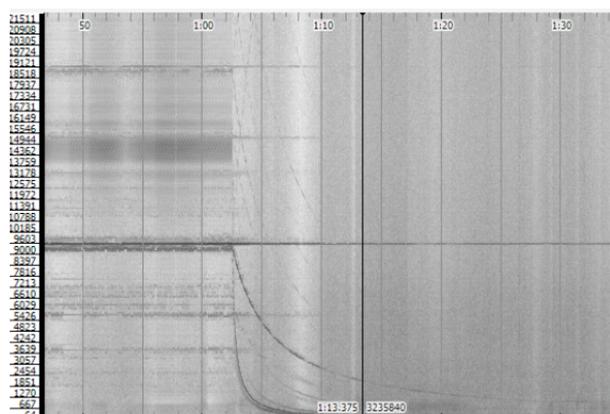
Si *nyx* est réalisée à partir d'une profonde inspiration pour le chaos la pièce n'en possède pas moins une structure. En effet, le long de ses 8 minutes 15 secondes, l'auditeur relèvera aisément des sections délimitées par deux gestes saillants – des glissandos qui balaient la totalité du spectre. Ces gestes sont réalisés en modifiant directement les fréquences de calcul de la Synthèse chaotique, de sorte à créer des gestes ascendants ou descendants. Le premier apparaît à 1', le second vers la fin, à 7'30". Tous deux encapsulent donc une section de plus 6 minutes, la section principale. La structure est donc de forme tripartite, avec une ouverture, un développement et une fermeture.

#### 3.1. Ouverture

L'ouverture est constituée par l'exposition ou, plus exactement, la confrontation de la masse chaotique à l'auditeur. Les dix signaux de la Synthèse chaotique sont diffusés au travers du Spat, brutalement, sans aucune modification durant une minute. De prime abord, il est possible d'interpréter cette ouverture comme une confrontation au chaos. La masse chaotique apparaît brutalement, sans fondu d'entrée, ce qui peut engendrer chez l'auditeur une certaine désorientation : ce bruit est en effet inhabituel et inouï pour la plupart des auditeurs. Cependant, comme cette ouverture dure un certain temps, il serait plus juste de la voir comme une invitation au bruit et au chaos. En prolongeant à l'identique ce chaos sur une durée d'une minute, la compositrice invite l'auditeur à découvrir et à considérer ce chaos, c'est-à-dire à dépasser finalement la signification usuelle du bruit comme indésirable, pour la remplacer par quelque chose à découvrir et à écouter.

L'ouverture se termine avec le premier geste saillant, un balayement descendant d'environ 30 secondes (figure 12). Il est réalisé en diminuant toutes les fréquences de calcul des fonctions chaotiques en passant d'une période de 1 à 50 échantillons, avec des temps différents de sorte à augmenter le trait.

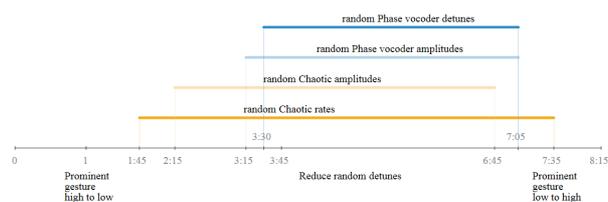
Ce geste s'inscrit en contraste de la masse chaotique diffusée durant les 60 précédentes secondes. Avec celui-ci nous comprenons que *nyx* n'est pas exclusivement fait de bruits, mais que la pièce intègre des gestes plus communs et construits. Finalement, ce glissando place l'auditeur dans une vive situation d'expectative, comme si quelque chose allait commencer.



**Figure 12.** Sonagramme du premier geste saillant.

#### 3.2. Développement

C'est la section la plus importante, celle durant laquelle se déroulent tous les matériaux de la pièce. De surcroît, elle se démarque des autres car elle contient, du fait des algorithmes, les éléments indéterminés. Avant d'aborder le rendu sonore et musical de cette section il est donc important de se focaliser sur les écritures, en éclairant l'état des algorithmes. La figure 13 montre que cette section principale est le siège des algorithmes ; tous vont en effet être actifs.



**Figure 13.** États des algorithmes sur la durée de la pièce.

Le premier à être activé concerne la Synthèse chaotique : il va engendrer de longs glissandos dans la masse chaotique. Puis c'est l'algorithme d'amplitude du même module qui va tailler des événements sonores dans cette masse. D'une façon similaire, la compositrice va ensuite activer les algorithmes du PhaseVocoder, en premier celui qui concerne l'amplitude et en second celui sur la hauteur, qui va transposer les précédents événements dans le grave. Nous remarquons ainsi, d'une façon similaire à ce que nous avons relevé dans l'artéfact<sup>19</sup>, un procédé d'accumulation au niveau des algorithmes (figure 14).

```

    randomeChaoticRates
    + randomChaoticAmp
    + randomPvAmp
    + randomPvDetune
    
```

**Figure 14.** Procédé d'accumulation des algorithmes.

Même si cette section est fortement construite avec des algorithmes aléatoires elle conserve de fortes similitudes d'une réalisation à l'autre, c'est-à-dire que l'indétermination provoquée par les algorithmes reste

<sup>19</sup> Voir à ce sujet la section 2.3.

contrôlée : ils dessinent un champ des possibles dans un espace fini, et c'est au sein de ce champ des possibles que nous pouvons analyser le rendu sonore et musical.

La section principale commence avec un ensemble de mouvements spectraux continus (1'45"). Ils sont produits avec l'algorithme *random rates* de la Synthèse chaotique, qui va dessiner un ensemble de mouvements de l'ordre de la phrase, entre 10 et 30 secondes. Ces mouvements rappellent le premier geste saillant, mais de façon nettement moins marquée. L'auditeur découvre une nouvelle direction dans la pièce, plus construite, que l'on peut positionner entre la masse chaotique et le geste saillant. Cette direction s'enrichit à 2' avec le PhaseVocoder. Il ajoute progressivement une nouvelle couche de bruit, profonde, provoquée par les étirements (*time stretching*).

Un nouveau moment apparaît vers 3'35" qui se caractérise par une zone de calme nettement marquée. L'énergie spectrale est par conséquent moins forte, avec de plus l'apparition de courts événements sonores de différentes hauteurs. Ceci est provoqué par le PhaseVocoder, avec l'activation successive de ses deux algorithmes : le premier va ciseler de nouveaux événements sonores et le second les transposer dans le grave. Cette zone de plus faible énergie apparaît nettement dans le sonagramme (figure 15, à environ 3'40").

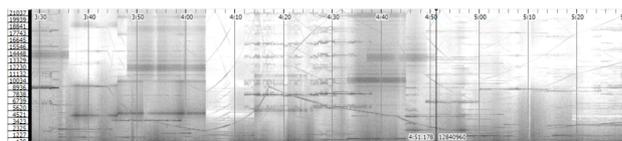


Figure 15. Sonagramme du début du développement.

À ce moment de la pièce, tous les matériaux ont été exposés. Comme il n'y a plus d'actions compositionnelles sur plusieurs minutes, nous sommes tentés de dire que la compositrice laisse parler les algorithmes. Grâce au PhaseVocoder, des sons de courtes durées vont apparaître et disparaître, laissant parfois place à des zones de faibles énergies. Des sons de différentes hauteurs se juxtaposent, laissant apparaître des intervalles limpides et familiers. C'est dans cette période que se trouve précisément le champ des possibles dessiné par les algorithmes. Les événements qui apparaissent s'inscrivent dans des espaces délimités en durée, en amplitude et en hauteurs.

De surcroît, la compositrice a ajouté une direction à ce champ des possibles, en le faisant légèrement évoluer durant les 3 minutes. Cela est fait avec l'algorithme *random detune* du PhaseVocoder, par un procédé que nous n'avons pas encore abordé : les valeurs de transposition du PhaseVocoder, tirées au hasard, vont progressivement être réduites jusqu'à devenir nulles. Au début, ces valeurs peuvent descendre jusqu'à -8000 cents puis quelques secondes après -7000 et ainsi de suite jusqu'à une transposition nulle (6'45"). Avec ce procédé, la compositrice ajoute donc une ligne de tension, une direction, qui vient fortement contraster avec l'indétermination.

### 3.3. Fermeture

La pièce se termine avec le second geste saillant (7'35"), qui balaie l'intégralité du spectre, mais cette fois-ci de façon ascendante (figure 16). Comme le premier, ce geste est effectué en modifiant directement les fréquences de calcul des fonctions chaotiques : l'algorithme qui les déterminait au hasard (*random rates*) est éteint, puis les fonctions chaotiques retournent progressivement à la fréquence d'échantillonnage, ce qui va provoquer l'important glissando.

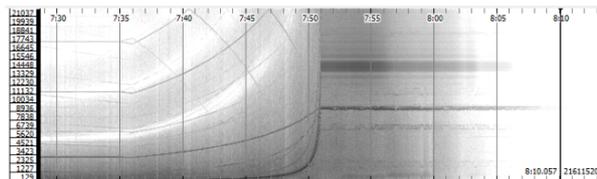


Figure 16. Sonagramme de la fermeture.

Ce geste rappelle bien évidemment le premier, comme son reflet, et place de nouveau l'auditeur dans une situation d'attente, comme l'annonce d'un événement nouveau. Ce geste amène la fin de la pièce d'une manière très fluide et agréable. Le balayement atteint le suraigu puis laisse place à la masse chaotique qui avait ouvert la pièce, mais ici elle ne dure pas : elle s'éteint dans un long fondu de sortie. À l'opposé de l'ouverture, le chaos n'est plus utilisé ici comme un élément brut et violent, mais pour finaliser un geste musical à part entière, à l'instar d'une construction.

### 3.4. Structure

À l'issue de notre analyse nous pouvons apporter une nouvelle lumière sur la structure de la pièce. Après quelques écoutes, l'auditeur pourra constater des points de contrastes, des éléments qui s'inscrivent par oppositions et que l'on traverse durant l'écoute de *nyx*. Nous proposons de voir la pièce comme un jeu de tensions entre deux pôles. Le premier est celui du chaos, avec cette masse chaotique, et le second celui du geste, élément construit et porteur de forme, avec les deux gestes saillants comme archétypes.

Afin d'éclairer la structure de *nyx*, nous pouvons placer ces pôles de part et d'autre d'un axe, qui représenterait l'ensemble des possibilités sonores et musicales de la pièce (figure 17). Sur cet axe, nous ajoutons ensuite les algorithmes en fonction de leur proximité avec les pôles : ceux d'amplitude sont plus proches du chaos, ceux de hauteur – qui vont jusqu'à produire des lignes de tensions – proches du geste.

Nous pouvons alors décrire les trois sections en fonction de leur parcours par rapport à cet axe, comme le propose le tableau 1. Nous voyons que l'ouverture est la section la plus abrupte, car elle passe d'un pôle à l'autre sans transition. Le développement est la section qui profite le plus de ces oppositions ; c'est celle qui exploite les richesses de contraste entre les deux pôles. La fermeture, quant à elle, se confond avec le pôle du geste ; c'est en effet le geste le plus fluide de la composition.

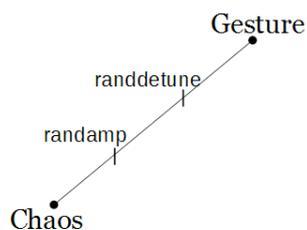


Figure 17. L'axe chaos-geste pour éclairer *nyx*.

Ouverture	c, g
Développement	c → g
Fermeture	g

Tableau 1. La structure de *nyx* selon l'axe chaos-geste (c pour chaos et g pour geste).

Il y a enfin une dernière façon d'éclairer la structure, selon l'esthétique et le sens du Chaos dans la mythologie grecque. Dans la narration mythologique le Chaos précède la création de la Terre et celle des Dieux, dont Nyx : « Donc, au commencement, fut Chaos, et puis la Terre au vaste sein<sup>20</sup> ». Le Chaos et la Terre s'inscrivent donc en opposition : le premier est un vide béant, une faille abyssale, tandis que la Terre apporte une fondation. Il est alors aisé de retrouver cette narration dans *nyx* : l'ouverture est une métaphore du Chaos, un vide où l'auditeur est désarmé, sans prise ; le développement s'assimile au commencement de la création de la Terre, avec l'émergence de formes – les légers glissandos par exemple ; puis la fermeture renvoie à l'aboutissement de la création de la Terre, avec un geste musical aboutit, où le Chaos est maîtrisé en disparaissant progressivement.

#### 4. CONCLUSION

Nous avons analysé *nyx*, une pièce pour ordinateur avec des processus aléatoires composée par Kerry Hagan en 2017.

Dans la première partie nous avons décrit le patch, que nous préférons appeler l'artéfact, de sorte à diriger sa description vers le musical et non le technologique. Ses différents composants ont été décrits selon deux grands domaines de l'informatique musicale, celui du signal – les modules – et celui du signe – les algorithmes. L'artéfact de *nyx* est composé d'un module de Synthèse chaotique, qui produit le matériau originel de la pièce, d'un Spat, que nous avons réduit pour cet article à un espace stéréophonique, et d'un PhaseVocoder, qui est utilisé pour extraire des événements de la masse chaotique. Les éléments appartenant au domaine du signe, plus proches de la compositrice, ont ensuite été décrits : les algorithmes d'amplitude intégrés aux modules de Synthèse chaotique et du PhaseVocoder, et finalement les algorithmes sur les hauteurs, intégrés aussi à ces deux modules. Nous avons terminé la description de l'artéfact en éclairant sa structure de contrôle, qui organise les différents modules et contrôles selon leur impor-

tance dans la pièce : en premier se trouve la Synthèse chaotique, source sonore de la pièce, ensuite le PhaseVocoder, utilisé pour colorer des éléments, et enfin le Spat.

Dans la seconde partie nous avons analysé la pièce dans sa forme canonique. *nyx* possède une structure limpide de forme tripartite. Chaque section a été analysée en décrivant les événements sonores, les différentes phases musicales, et en se reportant à la technique éclairée dans la première partie. Nous avons terminé l'analyse en éclairant la structure de *nyx* selon l'opposition entre deux pôles, le chaos et le geste. L'inscription temporelle de *nyx* peut se comprendre comme différents chemins tracés entre ces derniers, de sorte à exploiter les richesses de contrastes qui se dessinent entre une masse chaotique débordante et des gestes saillants couvrant l'intégralité du spectre.

Nous souhaitons conclure en éclairant différents segments de chaos présents dans *nyx*. Le premier relève du design sonore : il s'agit de l'utilisation originale de fonctions mathématiques chaotiques afin de produire des signaux audio, et surtout de les utiliser comme fondements de la pièce. Le deuxième relève de la mésostructure : il concerne l'utilisation d'algorithmes aléatoires afin de délimiter un champ des possibles au sein duquel des événements sonores prendront place. Enfin, le dernier niveau relève de la macrostructure : il s'agit d'évoquer, au travers de la structure de la pièce, la narration mythologique présente entre Chaos et création. Ces trois niveaux peuvent être compris comme trois segments d'un ensemble disparate que représenterait le Chaos. Ils nous amènent à voir *nyx* comme une pièce d'informatique musicale aussi surprenante que captivante.

#### 5. RÉFÉRENCES

- [1] Collins, N. « The Analysis of Generative Music Programs », *Organised Sound* 13/3 (2008), p. 237-248.
- [2] Duchez, M.-E. « L'évolution scientifique de la notion de matériau musical », *Le timbre, métaphore pour la composition*, Barrière, J.-B. (dir.). Christian Bourgeois, Paris, 1991, p. 47-81.
- [3] Hagan, K. « Computer Music for Absolute Beginners : Real-time Sound Synthesis, Manipulation and Composition », Université de Limerick, Irlande, 2018. [Support pédagogique.]
- [4] Hagan, K. « The Intersection of “Live” and “Real-time” », *Organised Sound* 21/2 (2016), p. 138-146.
- [5] Hagan, K. « The Noise of *Morphons & Bions* », International Computer Music Conference, Perth, Australie, 2013. [hdl.handle.net/10344/3512, accédé le 19/10/2020.]
- [6] Hagan, K. « Aesthetic and Philosophical Implications of Noise in *Morphons & Bions* », Music and Invention Workshop, Ewing, USA,

<sup>20</sup> Extrait de la Théogonie d'Hésiode.

2012. [[hdl.handle.net/10344/3509](https://hdl.handle.net/10344/3509), accédé le 10/10/2020.]

- [7] Larrieu, M. « A Consideration of the Code of Computer Music as Writing, and Some Thinking on Analytical Theories », *Organised Sound* 24/3 (2019), p. 319-328. [[hal-02386818](#)]
- [8] Larrieu, M. « Outils informatiques et analyse musicale : représenter le routing de *Jupiter* », Journées d'Informatique Musicale, Amiens, 2018. [[hal-01791378](#)]
- [9] Malloch, J., Birnbaum, D., Sinyor, E., Wanderley, M. « Towards a New Conceptual Framework for Digital Musical Instruments », Proceeding of the the 9th International Conference on Digital Audio Effects, Montréal, Canada, 2006.
- [10] Puckette, M. *The Theory and Technique of Electronic Music*. World Scientific Publishing, Singapore, 2006. [[msp.ucsd.edu/techniques.htm](http://msp.ucsd.edu/techniques.htm), accédé le 12/02/2020.]
- [11] Rodet, X. « Sound and Music from Chua's circuit », *Chua's Circuit : A Paradigm For Chaos*, Madan, R. N. (dir.). World Scientific, [Singapour], 1993, p. 434-446.
- [12] Shannon, C. E. « A Mathematical Theory of Communication », *Bell System Technical Journal* 27/3 (1948), p. 379-423.
- [13] Slater, D. « Chaotic Sound Synthesis », *Computer Music Journal* 22/2 (1998), p. 12-19.
- [14] Truax, B. « Chaotic Non-Linear Systems and Digital Synthesis : An Exploratory Study », International Computer Music Conference, Glasgow, Royaume-Uni, 1990.

*Texte édité par Nathalie Hérold*