

Interpolations

Denis Lorrain

Professeur d'électroacoustique et informatique musicales
SONVS, Département de composition, CNSMD de Lyon

denis.lorrain@cnsmd-lyon.fr
<http://www.cnsmd-lyon.fr>
<http://denislorrain.org>

Résumé : La notion d'interpolation, très générale et puissante, se prête à de nombreuses réalisations. Un processus d'interpolation a été expérimenté dans deux contextes différents par des étudiants en composition, non spécialistes de l'informatique musicale. Il s'agit d'interpolations spectrales dans un cas, et rythmiques dans l'autre. Les procédés et leur contexte de mise en œuvre sont décrits. Des résultats sont présentés.

Le concept d'interpolation est à la fois très général et puissant. C'est une idée assez classique, voire académique, consistant à faire varier une entité par étapes, intermédiaires et graduelles, entre deux états déterminés. C'est donc essentiellement un processus borné par un état initial A et un état terminal B : $[A \dots B]$.

On en connaît d'innombrables exemples. Le *Boléro* de Maurice Ravel¹ est ainsi une interpolation dynamique et orchestrale. D'une manière plus subtile, par exemple dans *Ur*, Magnus Lindberg utilise des glissandi de cordes pour effectuer des transitions interpolées entre agrégats harmoniques². Dans les mêmes années, lors de collaborations avec l'IRCAM, ce compositeur s'est explicitement préoccupé d'algorithmes d'interpolation harmonique³. Jason Eckardt, un jeune compositeur américain, se réfère aussi à ce concept en tant que fondement de sa démarche compositionnelle :

« La forme dans mon travail est définie par des processus, tandis que les frontières formelles sont délimitées par les points de début et de fin de ceux-ci. Commencant avec un matériau musical de base, le processus le transforme, changeant son état. La forme que prend le matériau au début et à la fin du processus n'est finalement pas aussi importante que son

¹ Paris : Durand, 1929.

² Helsinki : Wilhelm Hansen, 1987.

³ Algorithmes développés originellement en *PatchWork*, un environnement de programmation graphique basé sur Common LISP, initié par Mikael Laurson en 1985 (cf. <http://www2.siba.fi/soundingscore/MikaelsHomePage/MikaelsHomepage.html>) et développé à l'IRCAM antérieurement à *OpenMusic* (cf. <http://www.ircam.fr>). Cf. LINDBERG, Magnus, Séminaire donné à SONVS, Lyon : CNSMD, 18 mai 1990.

évolution progressive à l'intérieur du processus même. Par conséquent, le matériau est défini par son mouvement à travers un présent qui ne cesse de changer, dans un état de devenir qui donne à la musique un sens du mouvement, de l'instabilité et du changement.⁴ »

Il serait spécieux de prétendre tout ramener au concept d'interpolation, mais il n'en demeure pas moins qu'il se retrouve, aussi bien au niveau de l'individu — [naissance ... mort] — qu'à celui d'une cosmogonie mythique ou métaphysique — [d'où venons-nous ? ... où allons-nous ?] — dans toute démarche ou prise de conscience biographique, historique ou cosmologique — [big bang ... ?]⁵. Parmi une infinité d'exemples, la *Capella degli Scrovegni* de Giotto (Figure 1) en est une réalisation frappante. C'est un parcours graduel, théologique et iconographique, à travers une période de l'histoire sainte, depuis les prémices de l'Annonciation jusqu'à la Résurrection⁶.



Figure 1 — Padoue : *Capella degli Scrovegni*, fresques de Giotto [1267-1317], réalisées entre 1303 et 1305.
Cf. <http://www.capelladeglisrovegni.it>

Comme on le constate en admirant la fresque magistrale de Giotto, à partir du moment où le terme, la fin, d'un parcours est connu et anticipé, que cela soit d'une évolution, d'une histoire, d'une transformation quelconque, apparaît une notion de devenir, une tension vers ce terme. En ce sens, même une forme *ABA*⁷ peut-être rapprochée d'une interpolation. Sans

⁴ ECKARDT, Jason, "Devenir", Deborah Lopatin (trad. de l'anglais), *L'étincelle*, n° 1, IRCAM / Centre Georges Pompidou, 2007, p. 14. La suite du texte de J. Eckardt nuance ce propos, quant à la mise en œuvre musicale du procédé. Je reviendrai ci-dessous sur cette mise en perspective plus réaliste.

⁵ *Big crunch... ou expansion infinie ?*

⁶ On trouve de multiples fresques et tableaux analogues, mais moins connus. Parmi d'autres, je citerais *Das Leben Christi* (tableau comportant 35 scènes, réalisé à Cologne vers 1410-1420, Berlin, *Gemäldegalerie* n° cat. 1224), les *Strophen des Akathistos-Hymnos* (panneau de 24 icônes, réalisé en Crète fin XVII^{ème} siècle, exposition *Der Glanz des Himmels, Griechische Ikonen der Sammlung Velimezis*, Berlin, Pergamon Museum, 2007). Relevant du même principe, mais sans référence religieuse, la série de tableaux de Edvard Munch *Lebensfries* couvre symboliquement le parcours d'une vie, de la naissance à la mort (série réalisée à l'origine pour le théâtre Max Reinhardt, Berlin, *Neue Nationalgalerie*).

⁷ Qu'il s'agisse d'une forme sonate ou d'une simple *aria*, etc.

doute, sa partie centrale *B* peut être vue davantage comme une excursion qu'une interpolation, mais, son terme étant connu, et attendu, le processus étant borné par la réexposition de la partie *A*, on peut l'interpréter comme un cheminement graduel vers ce terme.

A contrario, certains procédés, même graduels, ne sont pas des interpolation. Par exemple en musique, ceux du thème et variations, ou du développement continu chez Debussy et Wagner. Il s'agit alors de processus ouverts, essentiellement infinis⁸, ou du moins dont le terme n'est pas prévisible lors de la première audition.

Par extension, ce que l'on nomme aujourd'hui *morphing*⁹ relève bien de l'interpolation. Cette idée fut appliquée très tôt au timbre dans le domaine de l'informatique musicale. En effet, lorsque des timbres sont décomposés en ingrédients distincts et relevant de modèles compatibles, il est relativement aisé de transformer graduellement les paramètres de l'un en ceux d'un autre. De telles opérations d'hybridations et de transitions de timbres peuvent être réalisées dans le cadre de différentes techniques : synthèse pure, comme ci-dessous, synthèse croisée¹⁰, modèles de résonances¹¹. En ce domaine, la pièce *Ritual Melodies*, de Jonathan Harvey¹², réalisée entre 1984 et 1990 par Jan Vandenheede¹³, constitue un aboutissement remarquable.

Le principe d'une transformation graduelle peut aussi être porté à un niveau quelconque. Par exemple, aux niveaux symbolique et sémantique, les étonnantes interpolations stylistiques de David Cope¹⁴, où l'on peut entendre un pastiche de concerto pour piano de Mozart se transformer graduellement en pastiche de gamelan balinaï, avant de revenir à Mozart¹⁵, formant ainsi un *ABA* à partir d'un aller-retour interpolé.

Dans son acception usuelle, l'interpolation réalise ainsi l'idée d'un parcours unidimensionnel — ou, du moins, linéaire — et graduel. Les exemples qui seront développés ci-dessous sont de cette nature. En théorie, cependant, rien n'empêche de multiplier son principe en des cheminements multidimensionnels, et de s'affranchir de sa graduation systématique.

Une œuvre, ou une série d'œuvres, peut ainsi se proposer d'explorer un champ, un domaine de possibles. J'en vois un exemple dans les vues de la façade de la cathédrale de Rouen peintes par Monet¹⁶ (Figure 2), dans lesquelles le peintre explore — échantillonne —

⁸ L'expression anglaise *open-ended* traduit mieux cela, sans faire référence à la notion délicate d'infini.

⁹ Terme d'origine info-graphique (dite *2D* ou *3D*) : effet montrant une figure ou forme se transformant graduellement en une autre.

¹⁰ Procédé d'analyse et resynthèse. Cf. par ex. l'application *AudioSculpt* de l'IRCAM (<http://www.ircam.fr/logiciels-forum.html>), ainsi que son moteur de calcul *SuperVP* (*ibid.* et <http://forumnet.ircam.fr/349.html>) et manuels afférents (*AS2-Users_Manual-FR.pdf*, *ibid.*, section 23.2).

¹¹ Cf. BAISNEE, P.-F., POTARD, Y. et BARRIÈRE, J.-B., "Méthodologie de synthèse du timbre : l'exemple des modèles de résonance", *Le timbre, métaphore de la composition*, Paris : IRCAM / Bourgois, 1991, pp. 135-163.

¹² Œuvre acousmatique, pour bande seule : CD Bridge Records BCD 9031, 1991.

¹³ VANDENHEEDE, Jan, "Jonathan Harvey's *Ritual Melodies*", *Interface*, vol. 21, 1992, pp. 149-183.

¹⁴ De l'université de Californie à Santa-Barbara : COPE, David, *Experiments in Musical Intelligence*, A-R Editions (*The Computer Music and Digital Audio Series*, vol. 12), 1996.

¹⁵ Cf. HOFSTADTER, Douglas, conférence *La mécanisation de la composition de la musique*, Palaiseau : École Polytechnique, 30 janvier 2002.

¹⁶ Il traita aussi de la même manière un sujet plus modeste : des meules de foin.



[http://parspassion.canalblog.com/albums/mon_musee_d_orsay_pascal/photos/...]

« Durant les années 1880 et 1890, Monet peignait une série de peintures de la cathédrale de Rouen, à partir de différents points de vue et à différentes heures du jour. Vingt vues de la cathédrale ont été exposées à la Galerie Durand-Ruel en 1895. Il fit également une série de peintures de meules de foin.»

[http://fr.wikipedia.org/wiki/Claude_Monet]

Figure 2 — Quatre des vues de la façade de la cathédrale de Rouen, peintes par Claude Monet. Il en réalisa une vingtaine entre 1880 et 1890. Cf. http://fr.wikipedia.org/wiki/Claude_Monet

un domaine bidimensionnel *heure du jour* x *saison*. D'autre part, les travaux de Dmitri Tymoczko¹⁷ montrent que des progressions harmoniques d'œuvres du répertoire¹⁸, voire même, plus généralement, toute la musique tonale, peuvent être représentées comme des parcours exploratoires, souvent graduels et contigus, de zones assez restreintes d'espace géométriques appropriés. Dans ma production personnelle, j'ai assez fréquemment mis en œuvre des procédés de progression graduelle et d'accumulation, relevant du parcours linéaire interpolatoire¹⁹. Par ailleurs, dans un article, j'ai tenté d'organiser l'exploration et l'exploitation d'un domaine multidimensionnel de possibles à partir de la notion de distance, ou contraste²⁰. Par transposition du *Traveling Salesman Problem*²¹ (Figure 3), j'ai ainsi pu construire des œuvres autour de la notion de parcours cyclique de contraste maximal²².

¹⁷ De l'université Princeton : cf. <http://music.princeton.edu/~dmitri>. Cf. aussi ANDREATTA, Moreno, "La géométrie d'un prélude", *Pour la science*, 349, pp. 96-97, 2006, ainsi que LEMONICK, Michael D., "The Geometry of Music", *Time*, Feb. 5 2007, p. 47.

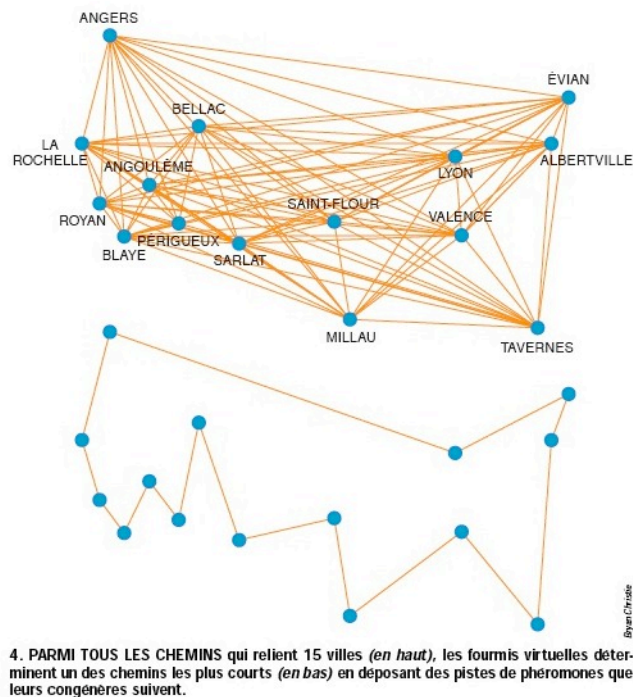
¹⁸ Le *Prélude en Mi mineur Op. 28 n°4* de Frédéric Chopin est ainsi analysé dans l'article de M. Andreatta précité.

¹⁹ Par exemple Arc, P-A, *The Other Shape* (CD *Shadow Box*, Centrediscs, CMC-CD3288, Robert Leroux, percussions), œuvres instrumentales, vocales ou mixtes, dont les partitions sont disponibles au Centre de musique canadienne (<http://www.centremusique.ca/home.cfm>) ; ... *black it stood as night* (CD Wergo, WER 2022-50, *Computer Music Currents*, vol. 2), œuvre acousmatique pour bande seule.

²⁰ "Le contraste bien mesuré", *Interface*, vol. 20, n° 3-4, Lisse : Swets & Zeitlinger, 1991, pp. 165-174.

²¹ Problème du VRP (voyageur représentant placier) devant réaliser un circuit de longueur minimale dans le graphe complet des villes de son territoire. C'est un cas d'école en théorie des graphes.

²² *Contra Mortem, Le talon d'Achille*, partitions disponibles au Centre de musique canadienne (cf. *supra*). Cf. aussi LORRAIN, Denis, *Essais de formalisation en composition musicale*, thèse de doctorat de troisième cycle, UER d'Esthétique et sciences de l'art, Université de Paris I, 1982, ainsi que l'article précité "Le contraste bien mesuré".



Traveling Salesman Problem

in [BONABEAU, E. & THÉRAULAZ, G., 2000: "L'intelligence en essaim", *Pour la science* : 271 : 66sq.]

Figure 3 — Exemple du *Traveling Salesman Problem* dans une région centre-ouest de la France. On y voit en haut le graphe complet des villes du territoire, et en bas un circuit de longueur minimale passant une et une seule fois par chaque ville. Figure extraite de BONABEAU, E. et THÉRAULAZ, G., "L'intelligence en essaim", *Pour la science*, 271, pp. 66 sq.

Dans de tels cas, la composition musicale peut être fondée sur la prééminence de la construction, de l'architecture, de la forme, davantage que sur le détail fin du matériau musical lui-même. Tout comme dans les œuvres stochastiques libres de Xenakis²³, la forme globale d'un passage, ses caractéristiques architecturales et statistiques, prédominent éventuellement sur le détail fin des événements sonores chargés de le constituer. Mais la citation de J. Eckardt ci-dessus montre qu'il n'en est pas forcément ainsi, et que le détail du parcours évolutif du matériau musical peut, au contraire, importer davantage que les termes bornant sa transformation graduelle.

Dans la suite de cet article, je vais détailler deux applications de la notion d'interpolation dans des travaux réalisés avec des étudiants compositeurs, non spécialisés en informatique musicale²⁴. Il s'agit dans le premier cas de transitions de timbres de synthèse, et dans le second d'interpolations rythmiques. Ces réalisations ne sont pas particulièrement innovantes en elles-mêmes, mais elles permettront de dégager ultérieurement quelques conclusions pédagogiques quant à l'intégration de l'informatique musicale par de jeunes compositeurs sans préparation particulière.

²³ XENAKIS, Iannis, "Musiques formelles", *La Revue musicale*, n° spécial 253-254, 1963. Nombreuses rééditions ultérieures, dont *Formalized Music*, Bloomington : Indiana University Press.

²⁴ Dans le cadre de la partie travaux pratiques de mon cours *Initiation à la programmation, à la synthèse et au traitement du signal* (iPST), destiné aux étudiants de l'option *Composition instrumentale et vocale*, CNSMD de Lyon, en 2004 et 2006 respectivement.

*
* *

Pour ce qui concerne leur aspect interpolatoire, constituant le cœur du présent exposé, les exemples qui vont suivre reposent sur le langage de programmation Common LISP²⁵. Dans le cadre réduit d'un cours de deux semestres pour étudiants non spécialisés en informatique musicale, la connaissance d'un langage de programmation est évidemment réduite au niveau d'une initiation. Cependant, après l'acquisition de quelques bases essentielles²⁶, il m'est possible de puiser, sans trop entrer dans leurs détails techniques, dans un corpus très large d'exemples appartenant à un cours plus complet destiné à des étudiants spécialisés²⁷. C'est ainsi que la fonction LISP `interpolist` sera au cœur des exemples suivants.

Cette fonction réalise des interpolations terme à terme, en un nombre d'étapes donné, entre deux listes d'égales longueurs de nombres, l'une de départ, l'autre d'arrivée. Voici un premier exemple d'évaluation, où l'on voit les listes de départ (1 5 10 -20) et d'arrivée (5 1 -10 60), de quatre termes chacune, ainsi que 5, le nombre d'étapes demandé, puis le résultat obtenu²⁸ sous forme d'une liste de cinq sous-listes constituant autant d'étapes, sous-listes de départ et d'arrivée incluses :

```
? (interpolist '(1 5 10 -20)
              '(5 1 -10 60)
              5
      )
(( 1 5 10 -20)
 ( 2 4 5 0)
 ( 3 3 0 20)
 ( 4 2 -5 40)
 ( 5 1 -10 60))
```

Comme on le voit, les interpolations sont linéaires par défaut, mais peuvent être optionnellement exponentielles ou logarithmiques. Dans ces deux derniers cas, les termes des listes de départ et d'arrivée ne doivent pas comprendre de zéro, et être de mêmes signes terme à terme. Voici un exemple exponentiel, cette fois à six étapes :

```
? (interpolist '(1 5 10 20)
              '(5 1 100 60)
              6
              :expon 1)
((1.0 5.0 10.0 20.0 )
 (1.3797 3.6239 15.8489 24.9146 )
 (1.9037 2.6265 25.1189 31.0369 )
 (2.6265 1.9037 39.8107 38.6636 )
 (3.6239 1.3797 63.0957 48.1645 )
 (5.0 1.0 100.0 60.0 ))
```

²⁵ Cf. STEELE, Guy L. Jr., *Common LISP, The Language (2nd Edition)*, Digital Press, 1991. Plus spécifiquement, la version utilisée est *Macintosh Common Lisp*, éditée par Digitool, cf. <http://www.digitool.com>.

²⁶ La simplicité syntaxique de LISP, ainsi que la facilité relative avec laquelle ce langage donne accès à des manipulations symboliques assez intéressantes, constituent en l'occurrence des avantages certains par rapport à d'autres langages de programmation.

²⁷ Cours de quatre semestres *Introduction à la recherche et à la création en informatique musicale* (RCI), pour les étudiants de l'option *Composition électroacoustique et informatique*.

²⁸ Par souci de simplicité, je passe sous silence les fonctions assurant la mise en page présentée ici.

Exemple logarithmique de quatre étapes, avec les mêmes listes de départ et d'arrivée :

```
? (interpolist '(1 5 10 20)
      '(5 1 100 60)
      4 :expon -3)
((1.0      5.0      10.0     20.0    )
 (4.2258   4.871    91.0811  47.6923 )
 (4.871    4.2258   99.1892  56.9231 )
 (5.0      1.0      100.0    60.0    ))
```

La Figure 4 détaille les fonctions mathématiques mises en œuvre. On voit que le

expon = nil (par défaut) ⇒ linéaire :

$$a_n = a_{n-1} + \left[\frac{a_{ntp} - a_0}{ntp} \right]$$

expon ≠ nil ⇒ exponentielle ⇒ $a_0 \times a_{ntp} > 0$

i.e. $a_0 \neq 0 \neq a_{ntp}$, ainsi que a_0 et a_{ntp} de mêmes signes

expon = coefficient $c \neq 0$ quelconque ⇒ exponentielle « variée » :

$$a_n = a_0 + \left\{ amb \left[1 - e^{\left(\frac{n \times baz}{ntp} \right)} \right] \right\}$$

expon = $c = 1$ ⇒ exponentielle « ordinaire », la formule ci-dessus équivalant alors à :

$$a_n = a_{n-1} \left(\sqrt[ntp]{\frac{a_{ntp}}{a_0}} \right)$$

expon = $c = 0 + \varepsilon$ ⇒ ≈ linéaire

où a_0 = l'un des éléments de **listA** (liste de départ)

a_{ntp} = l'élément correspondant de **listB** (liste d'arrivée)

ntp = **netap** - 1 = nombre d'étapes d'interpolation (a_0 et a_{ntp} inclus) moins 1

c = **expon** ≠ 0

$1 \leq n \leq ntp$

$$amb = \frac{a_{ntp} - a_0}{1 - e^{baz}}$$

$amb = 0$ d'office lorsque $a_0 = a_{ntp}$ afin d'éviter de calculer

$baz = 0 \Rightarrow amb = (0/0) \Rightarrow$ erreur LISP. Dans ce cas $a_n = a_0$

$$baz = c \times \ln \left(\frac{a_{ntp}}{a_0} \right)$$

Figure 4 — Fonctions mathématiques réalisées par l'algorithme `interpolist`, permettant de passer d'un mode d'interpolation linéaire à un mode exponentiel ou logarithmique, ce dernier autorisant des courbures variées en fonction du coefficient c précisé lors de l'évaluation LISP par le mot-clef `:expon`

coefficient c y permet de varier la courbure exponentielle ou logarithmique de manière continue. La Figure 5 en montre quatre exemples.

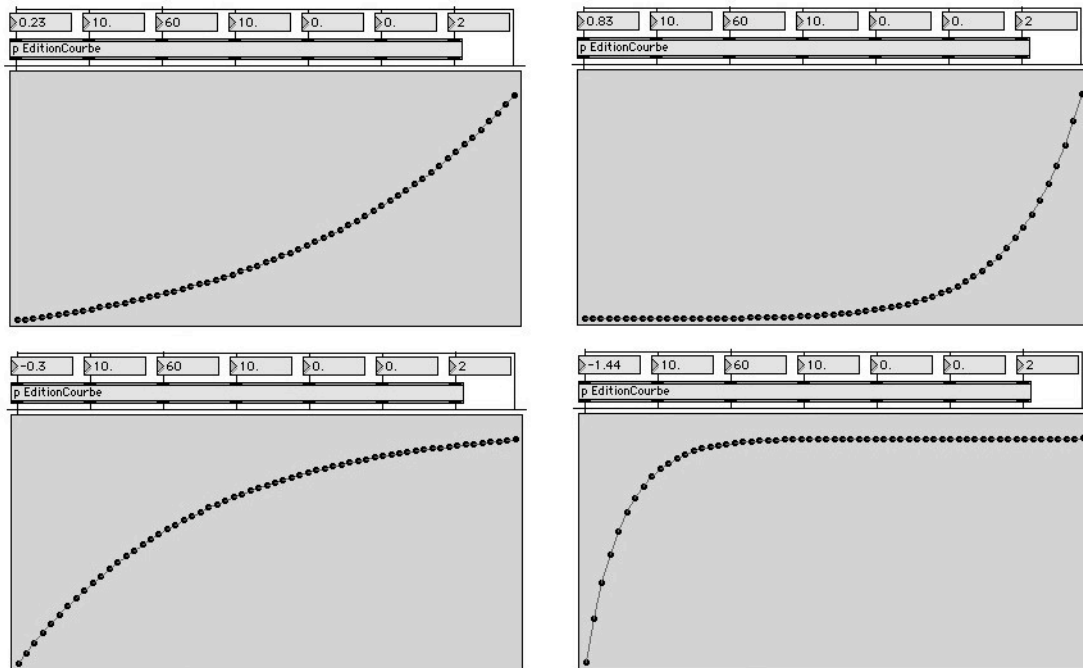


Figure 5 — Cette figure reprend des vues partielles d’une réalisation du même algorithme d’interpolation par un étudiant du CNSMD de Lyon, dans le cadre d’un autre projet (Jérôme Bertholon, 2006). Il s’agit de l’environnement *Max* (cf. <http://www.cycling74.com>). Le coefficient c de la Figure 4 y est représenté par l’objet boîte à nombres (*number box*) le plus à gauche. On voit des valeurs positives de c pour les courbures exponentielles (0.23 et 0.83), et négatives pour les logarithmiques (-0.3 et -1.44). Une partie cachée du *patcher Max* représenté intercepte la valeur zéro de c , et la remplace par un ε produisant un effet quasi-linéaire

La fonction *interpolist* n’a sans doute rien d’extraordinaire en soi, mais elle est un outil assez puissant pour se prêter à de nombreuses applications.

*
* *

La première réalisation recourt à l'interpolation en tant que moyen d'assurer des transitions timbrales (spectrales) dans le cadre de la synthèse additive. Dans un premier temps, nous avons utilisé comme base d'expérimentation les timbres de cloches analysés et catalogués de longue date par Jean-Claude Risset²⁹.

À cette fin, il faut d'abord réaliser la synthèse des cloches elles-mêmes. Nous avons donc traduit les données originelles de Risset sous forme d'instances d'une structure LISP :

```
; CLOCHES (INSTANCES) de c1210 à c1650
(setq c1210 (make-clochesJCR
  :nom "clochesJCR 1210"
  :nbrcomp 11
  :ampglob 1400
  :freqglob 226
  :instr 3
  :freqs '(224 225 368 369.7 476 680 800 1094 1200 1504 1628)
  :amps '(150 100 150 270 400 250 220 200 200 150 200)
  :durees '(20 18.7 13 11 6.5 7 5 4 3 2 1.5)))
(setq c1250 (make-clochesJCR
  :nom "clochesJCR 1250"
  :nbrcomp 7
  :ampglob 450
  :freqglob 394
  :instr 3
  :freqs '(240 277 385 605 100 670 812)
  :amps '(150 125 150 100 50 50 50)
  :durees '(10 9.5 8.5 5 7 5.5 4)))
; ... suivent dix autres instances (modèles) de cloches...
```

Pour mettre ces données en œuvre, nous avons rédigé la fonction LISP `cloche` permettant de développer une seule évaluation faisant référence à une instance — à un modèle — de cloche, en autant d'énoncés *i Csound*³⁰ que la cloche spécifiée comporte de partiels. Cette fonction LISP intègre la possibilité de transposer chaque instance de cloche, aussi bien en fréquence qu'en amplitude. Les énoncés *i Csound* résultant sont destinés à un instrument n° 10 très simple, appliquant à une sinusoïde une enveloppe à attaque très brève et chute exponentielle. Voici l'instrument *Csound* :

```
instr 10
  kenv   oscilli   0,p5,p3, 10
  ason   oscili    kenv,p4, 1
        out       ason
endin
```

et le début d'un fichier `.sco`³¹ associé, où l'on voit les définitions des tables `f1` (sinusoïde) et `f10` (enveloppe), suivies de l'ensemble de énoncés *i* correspondant aux onze partiels de la

²⁹ RISSSET, Jean-Claude, *An Introductory Catalogue of Computer Synthesized Sounds*, Murray Hill : Bell Laboratories, 1969. Ce catalogue a été récemment réédité dans *The Historical CD of Digital Sound Synthesis*, CD Wergo WER 2033-2 282 033-2 (coffret 1 CD avec livret 260 p., Coll. *Computer Music Currents* n° 13). Pour les caractéristiques complètes des timbres de cloche en particulier, cf. LORRAIN, Denis, 1980, *Analyse de la bande magnétique de l'œuvre de Jean-Claude Risset* Inharmonique, Paris : Centre Georges Pompidou, 1980 (Rapport IRCAM n° 26).

³⁰ Cf. <http://www.csounds.com> (et <http://www.csounds.com/matt/MacCsound> pour la version utilisée). Il s'agit d'un environnement logiciel de synthèse et de traitement sonores, conçu originellement par Barry Vercoe dans les années 1980 : VERCOE, Barry, *Csound: A Manual for the Audio Processing System and Supporting Programs with Tutorials*, Cambridge : MIT Media Lab, 1986. Je ne donnerai ici que très peu de détails sur *Csound* proprement dit. Les énoncés *i* commandent l'action d'un *instrument*. Ce dernier est un algorithme de synthèse et/ou de traitement. Chaque partiel est déclenché par un énoncé *i* spécifiant sa fréquence, son amplitude et sa durée.

³¹ `.sco` comme *score* (partition) : il s'agit du fichier contenant les « parties » destinées aux « instruments » du fichier `.orc` (comme *orchestra*).

cloche n° 1210 (cf. première instance de structure dans l'exemple ci-dessus), dont les fréquences sont en colonne 4 (champ *p4* en terminologie *Csound*) et les amplitudes³² en colonne 5 (champ *p5*) :

```

f1 0 4096 10 1
f10 0 4097 5          0.001      4
                        1          3887
                        0.01      205
                        0.001
i10 0.0 20.0 224.0 750.0 ; 0 cloches clochesJCR 1210
i10 0.0 18.7 225.0 500.0 ; 1 cloches clochesJCR 1210
i10 0.0 13.0 368.0 750.0 ; 2 cloches clochesJCR 1210
i10 0.0 11.0 369.7 1350.0 ; 3 cloches clochesJCR 1210
i10 0.0 6.5 476.0 2000.0 ; 4 cloches clochesJCR 1210
i10 0.0 7.0 680.0 1250.0 ; 5 cloches clochesJCR 1210
i10 0.0 5.0 800.0 1100.0 ; 6 cloches clochesJCR 1210
i10 0.0 4.0 1094.0 1000.0 ; 7 cloches clochesJCR 1210
i10 0.0 3.0 1200.0 1000.0 ; 8 cloches clochesJCR 1210
i10 0.0 2.0 1504.0 750.0 ; 9 cloches clochesJCR 1210
i10 0.0 1.5 1628.0 1000.0 ; 10 cloches clochesJCR 1210

```

Les onze énoncés *i* précédents résultent d'une seule évaluation LISP de la fonction `cloches` :
`(cloches 10 0.0 c1210 nil 0 :fichier fic :ajouter t)`

On réalise ainsi additivement, avec de six à onze partiels par instance, la synthèse de sons assez réalistes, dont l'efficacité n'est plus à démontrer. Jusqu'à présent, nous n'avons que recréé fidèlement, dans notre environnement LISP et *Csound*, l'implémentation originelle de Jean-Claude Risset, de manière à constituer la base de tout ce qui suit.

Les premières interpolations spectrales réalisées ensuite n'ont d'ailleurs pas recours à la fonction `interpolist` décrite ci-dessus, qui n'interviendra que dans un dernier temps. Ces premières interpolations procèdent par glissandi, réalisés en *Csound*, entre fréquences de deux instances de cloches.

Afin de réaliser ces transitions par glissandi de partiels, nous avons opté pour la synthèse de trames tenues — à la manière d'accords — dont les fréquences glissent graduellement à partir de celles d'une instance de départ, vers celles d'une instance d'arrivée. Une transition graduelle des amplitudes est réalisée de la même manière. La Figure 6 illustre une telle transition entre les instances 1210 et 1400. La fonction LISP `accordgliss`, faisant référence

³² Comme on le remarquera, dans le cadre de ce projet, nous avons systématiquement multiplié par cinq les amplitudes des partiels de toutes les instances.

Glissando CSound, entre deux spectres de même nombre de partiels

Fréquences (Hz)
Amplitudes (linéaires)

<i>Partiel</i>	<i>Cloche n° 1210</i>		<i>Cloche n° 1400</i>
0	224 150	→	464 500
1	225 100	→	463.2 300
2	368 150	→	605 300
3	369.7 270	→	607 100
4	476 400	→	775 200
5	680 250	→	933 400
6	800 220	→	943 200
7	1094 200	→	1080 200
8	1200 200	→	1240 100
9	1504 150	→	1560 150
10	1628 200	→	1720 200

Figure 6 — Fréquences et amplitudes des instances de cloches 1210 et 1400, entre lesquelles des glissandi sont effectués pour assurer un premier type de transition

à deux instances de départ et d'arrivée, a été créée pour ce faire. Voici un exemple d'évaluation d'accordgliss, où l'on voit des références aux instances 1210 et 1400 :

```
(accordgliss 12 0.0 c1210 nil c1400 nil -10 0 20 :fichier fic
:ajouter t)
```

Le résultat de cette évaluation est reproduit dans le fichier .sco suivant :

```

i12 0.0 20.0 224.0 464.0 237.17082 2500.0 ; 0 ag 1210->1400
i12 0.0 20.0 225.0 463.2 158.11388 1500.0 ; 1 ag 1210->1400
i12 0.0 20.0 368.0 605.0 237.17082 1500.0 ; 2 ag 1210->1400
i12 0.0 20.0 369.7 607.0 426.90748 500.0 ; 3 ag 1210->1400
i12 0.0 20.0 476.0 775.0 632.45553 1000.0 ; 4 ag 1210->1400
i12 0.0 20.0 680.0 933.0 395.28471 2000.0 ; 5 ag 1210->1400
i12 0.0 20.0 800.0 943.0 347.85054 1000.0 ; 6 ag 1210->1400
i12 0.0 20.0 1094.0 1080.0 316.22777 1000.0 ; 7 ag 1210->1400
i12 0.0 20.0 1200.0 1240.0 316.22777 500.0 ; 8 ag 1210->1400
i12 0.0 20.0 1504.0 1560.0 237.17082 750.0 ; 9 ag 1210->1400
i12 0.0 20.0 1628.0 1720.0 316.22777 1000.0 ;10 ag 1210->1400

```

On retrouve dans les champs 4 et 5 ($p4$ et $p5$) des énoncés i les fréquences respectives des partiels des instances 1210 et 1400. De même leurs amplitudes³³ dans les champs 6 et 7. Ces énoncés sont destinés à l'instrument n° 12 :

```

instr 12
  iagliss      =      log(p5/p4)
  iadynam      =      log(p7/p6)
  kenv         oscilli 0,p6,p3, 5
  kgliss       oscilli p3*0.1,iagliss,p3*0.75, 20
  kdynam       oscilli p3*0.1,iadynam,p3*0.75, 20
  ason         oscili   kenv*exp(kdynam),exp(kgliss)*p4,1
  out         out      ason
endin

```

Il est ensuite aisé d'ajouter les cloches proprement dites au début et à la fin de la trame interpolatoire ainsi obtenue³⁴, comme l'illustre la Figure 7.

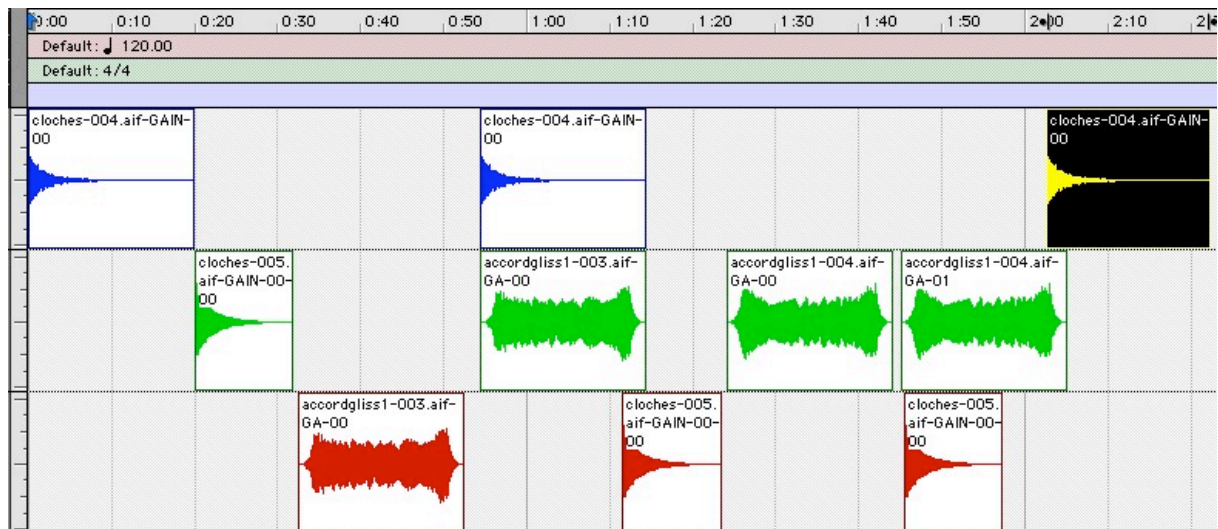


Figure 7 — Interpolations entre des instances de cloches, réalisée par des trames tenues de glissandi. On distingue les profils caractéristiques des cloches (attaques brèves et chutes exponentielles) et des trames interpolatoires tenues

Ce type d'interpolation par glissandi peut-être réalisé directement lorsque les instances de cloches de départ et d'arrivée possèdent le même nombre de partiels. Afin de permettre de telles interpolations même lorsque les instances n'ont pas le même nombre de partiels, il est nécessaire d'intercaler un pré-traitement des listes de fréquences. La fonction LISP

³³ Dans cet exemple, outre la multiplication systématique de ses valeurs d'amplitudes par cinq, l'instance 1210 a subi une atténuation de -10 dB (cf. septième argument de l'évaluation `accordgliss` ci-dessus), par souci d'équilibre dynamique avec l'instance 1400.

³⁴ Cet ajout peut se faire directement en *Csound*, ou avec une application de montage et mixage. Les Figures 7 et 10 reproduisent des fenêtres *Pro Tools* (<http://digidesign.com>) sous Mac OS 9.

appariement a été conçue à cet effet. Pour deux instances données, de nombres différents de partiels, elle conserve des glissandi, comme précédemment, entre partiels de fréquences assez voisines, mais ajoute en outre des composantes supplémentaires, sans glissandi, sur les fréquences des partiels restant. Selon qu'elles appartiennent à l'instance de départ ou d'arrivée, les fréquences de ces composantes supplémentaires disparaîtront ou apparaîtront graduellement, par contrôle de leur amplitude. Compte tenu de ces entrées ou sorties graduelles de composantes de fréquences fixes, de quelques autres partiels en glissandi, parfois eux-mêmes convergeant ou divergeant vers ou depuis des fréquences communes, la fonction appariement crée des données pour un nombre de composantes qui est toujours supérieur au nombre de partiels de l'instance la plus fournie. La Figure 8 illustre comment appariement crée treize composantes pour rendre possible une trame interpolatoire entre les

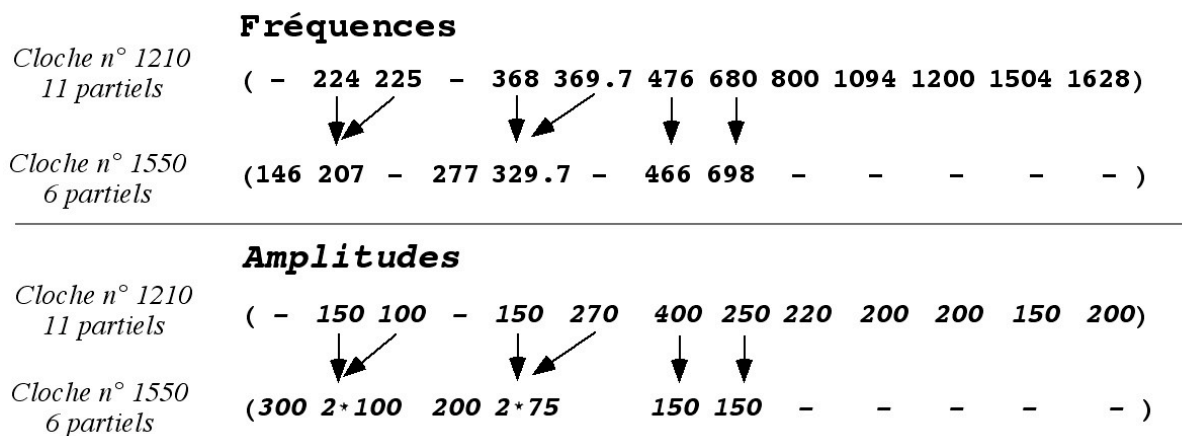


Figure 8 — Résultat de l'évaluation de la fonction appariement pour deux instances de cloches de nombres différents de partiels — respectivement 11 et 6. Certains partiels glisseront pendant de la transition, lorsque les fréquences concernées sont assez voisines. Les partiels restant effectueront des entrées ou sorties graduelles, vers ou à partir de leur amplitude dans chaque instance. Dans le cas illustré, on réalisera la transition en synthétisant un total de 13 composantes

instances 1210 (11 partiels) et 1550 (6 partiels). La Figure 9 résume succinctement le même cas.

Glissando CSound, entre deux spectres de nombre différent de partiels, après *appariement*

Fréquences (Hz)
Amplitudes (linéaires)

<i>Partiel</i>	<i>Cloche n° 1210</i>		<i>Cloche n° 1550</i>	<i>Partiel</i>
	146 0	→	146 300	0
0	224 150	→	207 100	1
1	225 100	→	207 100	
	277 0	→	277 200	2
2	368 150	→	329.7 75	3
3	369.7 270	→	329.7 75	
4	476 400	→	466 150	4
5	680 250	→	698 150	5
6	800 220	→	800 0	
7	1094 200	→	1094 0	
8	1200 200	→	1200 0	
9	1504 150	→	1504 0	
10	1628 200	→	1628 0	

Figure 9 — Tableau résumant la même interpolation qu'à la Figure 8 : 13 composantes réaliseront la transition interpolatoire entre les instances 1210 (11 partiels) et 1550 (6 partiels). Certaines composantes glisseront, d'autres seront de fréquence fixe, mais entreront ou sortiront graduellement au cours de la transition

Après cette opération d'appariement, on se trouve en mesure, comme précédemment, de faire réaliser des trames interpolatoires par l'instrument *Csound* n° 12. Voici une évaluation de la fonction `accordgliss1`, faisant elle-même intervenir appariement :

```
(accordgliss1 12 0 c1210 nil ;structD freqtranspD
                c1250 nil ;structA freqtranspA
                0 ;amptranspD
                6 ;amptranspA
                20 ;durée
                :fichier fic :ajouter t)
```

On y demande la réalisation d'une trame évoluant en 20 secondes entre le spectre de l'instance 1210 (11 partiels) et celui de la 1250 (7 partiels, amplifiés de 6 dB). Cet appel résulte en quatorze composantes correspondant à autant d'énoncés *i* pour l'instrument n° 12 :

```
i12 0.0 20.0 224.0 240.00000 750.00000 748.22337 ; 0 acg11
i12 0.0 20.0 225.0 240.00000 500.00000 748.22337 ; 1 acg11
i12 0.0 20.0 368.0 385.00000 750.00000 748.22337 ; 2 acg11
i12 0.0 20.0 369.7 385.00000 1350.00000 748.22337 ; 3 acg11
i12 0.0 20.0 476.0 476.00000 2000.00000 0.00315 ; 4 acg11
i12 0.0 20.0 680.0 605.00000 625.00000 997.63116 ; 5 acg11
i12 0.0 20.0 680.0 670.00000 625.00000 498.81558 ; 6 acg11
i12 0.0 20.0 800.0 812.00000 1100.00000 498.81558 ; 7 acg11
i12 0.0 20.0 1094.0 1094.00000 1000.00000 0.00315 ; 8 acg11
i12 0.0 20.0 1200.0 1200.00000 1000.00000 0.00315 ; 9 acg11
i12 0.0 20.0 1504.0 1504.00000 750.00000 0.00315 ; 10 acg11
i12 0.0 20.0 1628.0 1628.00000 1000.00000 0.00315 ; 11 acg11
i12 0.0 20.0 277.0 277.00000 0.00158 1247.03895 ; 12 acg11
i12 0.0 20.0 100.0 100.00000 0.00158 498.81558 ; 13 acg11
```

Les fréquences de départ et d'arrivée des composantes sont dans les champs 4 et 5, leurs amplitudes en 6 et 7.

La fonction `interpolist` présentée ci-dessus n'intervient donc pas dans ces interpolations incluant des glissandi. Mais elle entre maintenant en scène pour la réalisation d'un autre type d'interpolation, par synthèse d'une succession de spectres statiques, sans glissandi, entre les instances de cloches de départ et d'arrivée.

Dans ce cas, avec ou sans intervention de la fonction `appariement`, les listes de fréquences des composantes de départ et d'arrivée — plus nombreuses que les partiels proprement dits, comme il vient d'être montré — sont présentées à `interpolist`, qui en déduit les fréquences d'un certain nombre de spectres intermédiaires. De même pour les amplitudes. La fonction LISP `accords` a été rédigée pour réaliser cet ensemble d'opérations. Voici un exemple d'évaluation :

```
(accords 11 '(0 3 6 9 12 15 18) c1250 1
                                c1210 1
                                13 0.001 ;amptr=dB
                                21 ;dur
                                :expon -2.5 ;expon -> amp
                                :fichier fic :ajouter t)
```

On y voit, après une liste (0 3 6 9 12 15 18) des sept dates de début d'autant de spectres successifs, des références aux instances 1250 (7 partiels) et 1210 (11 partiels), etc. Les énoncés *i* suivants en sont le résultat :

```
i11 0.0 4.5 240.00000 1675.06347 ; 0 accord-bis
i11 0.0 4.5 240.00000 1675.06347 ; 1 accord-bis
i11 0.0 4.5 385.00000 1675.06347 ; 2 accord-bis
i11 0.0 4.5 385.00000 1675.06347 ; 3 accord-bis
i11 0.0 4.5 476.00000 0.00706 ; 4 accord-bis
i11 0.0 4.5 605.00000 2233.41796 ; 5 accord-bis
i11 0.0 4.5 670.00000 1116.70898 ; 6 accord-bis
i11 0.0 4.5 812.00000 1116.70898 ; 7 accord-bis
i11 0.0 4.5 1094.00000 0.00706 ; 8 accord-bis
i11 0.0 4.5 1200.00000 0.00706 ; 9 accord-bis
i11 0.0 4.5 1504.00000 0.00706 ; 10 accord-bis
```

```

i11  0.0 4.5 1628.00000  0.00706 ; 11 accord-bis
i11  0.0 4.5  277.00000 2791.77245 ; 12 accord-bis
i11  0.0 4.5  100.00000 1116.70898 ; 13 accord-bis

i11  3.0 4.5  237.25609 1880.19517 ;  0 accord-bis
i11  3.0 4.5  237.43229 1757.33431 ;  1 accord-bis
i11  3.0 4.5  382.11309 1880.19517 ;  2 accord-bis
i11  3.0 4.5  382.40672 2073.71156 ;  3 accord-bis
i11  3.0 4.5  476.00000   0.07340 ;  4 accord-bis
i11  3.0 4.5  616.89933 2318.04398 ;  5 accord-bis
i11  3.0 4.5  671.65640 1300.95820 ;  6 accord-bis
i11  3.0 4.5  809.98757 1429.49329 ;  7 accord-bis
i11  3.0 4.5 1094.00000   0.06540 ;  8 accord-bis
i11  3.0 4.5 1200.00000   0.06540 ;  9 accord-bis
i11  3.0 4.5 1504.00000   0.06233 ; 10 accord-bis
i11  3.0 4.5 1628.00000   0.06540 ; 11 accord-bis
i11  3.0 4.5  277.00000  325.84385 ; 12 accord-bis
i11  3.0 4.5  100.00000  151.84233 ; 13 accord-bis

i11  6.0 4.5  234.54355 2110.44729 ;  0 accord-bis
i11  6.0 4.5  234.89206 1843.64552 ;  1 accord-bis
i11  6.0 4.5  379.24782 2110.44729 ;  2 accord-bis
;...
i11  9.0 4.5  231.86203 2368.87194 ;  0 accord-bis
i11  9.0 4.5  232.37900 1934.17584 ;  1 accord-bis
i11  9.0 4.5  376.40404 2368.87194 ;  2 accord-bis
;...
i11 12.0 4.5  229.21116 2657.51361 ;  0 accord-bis
;...
i11 15.0 4.5  226.59060 2899.59208 ;  0 accord-bis
i11 15.0 4.5  227.43326 2068.20788 ;  1 accord-bis
i11 15.0 4.5  370.78029 2899.59208 ;  2 accord-bis
i11 15.0 4.5  372.20711 4732.20982 ;  3 accord-bis
i11 15.0 4.5  476.00000  834.47850 ;  4 accord-bis
i11 15.0 4.5  666.88353 2613.21809 ;  5 accord-bis
i11 15.0 4.5  678.32303 2328.11265 ;  6 accord-bis
i11 15.0 4.5  801.98761 3729.04720 ;  7 accord-bis
i11 15.0 4.5 1094.00000  468.33522 ;  8 accord-bis
i11 15.0 4.5 1200.00000  468.33522 ;  9 accord-bis
i11 15.0 4.5 1504.00000  368.50315 ; 10 accord-bis
i11 15.0 4.5 1628.00000  468.33522 ; 11 accord-bis
i11 15.0 4.5  277.00000   0.05875 ; 12 accord-bis
i11 15.0 4.5  100.00000   0.05043 ; 13 accord-bis

i11 18.0 3.0  224.00000  750.08635 ;  0 accord-bis
i11 18.0 3.0  225.00000  500.05757 ;  1 accord-bis
i11 18.0 3.0  368.00000  750.08635 ;  2 accord-bis
i11 18.0 3.0  369.70000 1350.15543 ;  3 accord-bis
i11 18.0 3.0  476.00000 2000.23027 ;  4 accord-bis
i11 18.0 3.0  680.00000  625.07196 ;  5 accord-bis
i11 18.0 3.0  680.00000  625.07196 ;  6 accord-bis
i11 18.0 3.0  800.00000 1100.12665 ;  7 accord-bis
i11 18.0 3.0 1094.00000 1000.11514 ;  8 accord-bis
i11 18.0 3.0 1200.00000 1000.11514 ;  9 accord-bis
i11 18.0 3.0 1504.00000  750.08635 ; 10 accord-bis
i11 18.0 3.0 1628.00000 1000.11514 ; 11 accord-bis
i11 18.0 3.0  277.00000   0.00158 ; 12 accord-bis
i11 18.0 3.0  100.00000   0.00158 ; 13 accord-bis

```


La mise en page de cet extrait abrégé de fichier .sco met en évidence une succession de sept spectres tenus de quatorze composantes, attaqués respectivement aux temps 0, 3, 6, 9, 12, 15 et 18 (champ 2), chacun durant 4.5 secondes (sauf 3 secs pour le dernier). Les fréquences des composantes sont dans le champ 4, et leurs amplitudes dans le champ 5. Les fréquences et amplitudes du premier et du dernier spectre correspondent respectivement aux valeurs des instances demandées, 1250 et 1210, compte tenu de l'appariement réalisé préalablement, ainsi que de certaines adaptations des amplitudes qu'il serait superflu de détailler ici. Les fréquences et amplitudes des spectres intermédiaires ont été calculées par *interpolist*.

L'extrait précédent de fichiers .sco a le mérite d'être clair et démonstratif. En réalité, des mises en œuvre moins « carrées » ont été réalisées, dont les temps d'attaque des différentes composantes de chacun des spectres successifs, ainsi que leurs durées individuelles étaient affectés par un coefficient de variation aléatoire, afin de produire un enchaînement plus flou des spectres statiques intermédiaires interpolés. Après ajout des instances proprement dites des cloches de départ et d'arrivée, on arrive à une interpolation spectrale par paliers dont l'allure globale est illustrée Figure 10.

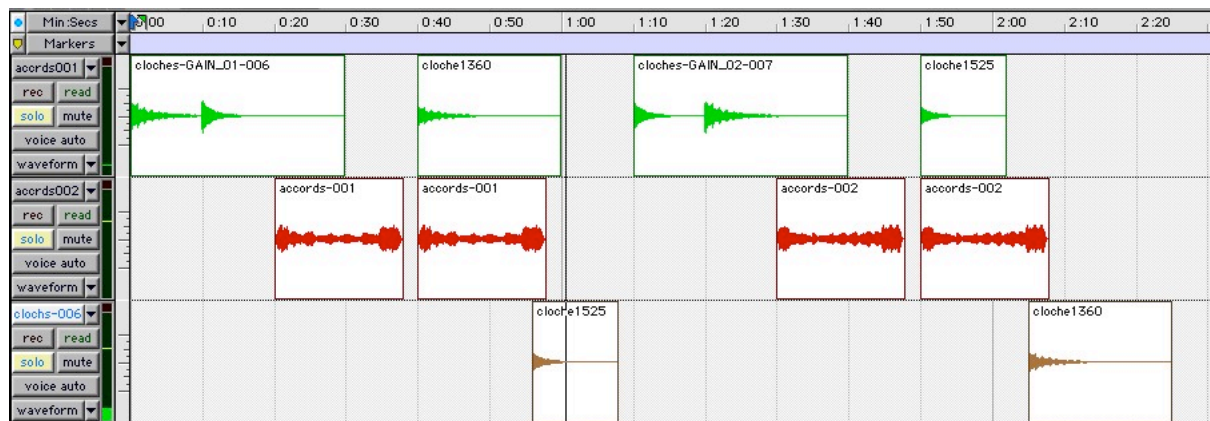


Figure 10 — Montage *Pro Tools* d'instances de cloches (pistes supérieure et inférieure) avec des transitions par des spectres interpolés de composantes tenues calculées par *interpolist* (piste centrale)

La morphologie intrinsèquement asymétrique des sonorités de cloches (attaque - résonance) crée inévitablement des difficultés de raccord lors des transitions interpolées. D'abord, la coexistence de la résonance de la première instance de cloche avec le début de l'interpolation, surtout lors de glissandi, résulte en battements et ambiguïtés. Enfin, l'attaque de la seconde instance rompt brutalement la continuité dynamique entretenue par l'interpolation. On pourrait pallier ces phénomènes par quelques artifices. Par exemple, il serait possible de retarder l'entrée de l'interpolation, d'introduire une à une ses composantes à partir de la plus grave, et d'appliquer simultanément une dynamique croissante afin de préparer l'attaque de la seconde instance... En résumé, cela consisterait à appliquer à l'interpolation une morphologie de son de cloche rétrogradé. Cependant, dans le cadre initial de cette réalisation, les cloches de Jean-Claude Risset ne constituaient que le terrain de mise au point du procédé générique d'interpolation lui-même. Nous n'avons donc pas spécifiquement adapté le procédé à leur morphologie particulière³⁵.

Mais nous en étions néanmoins arrivés à un modèle assez général d'interpolation spectrale, avec ou sans glissandi, applicable entre tout couple de sonorités dans le cadre de la synthèse additive.

³⁵ On verra de surcroît, en conclusion, que le cadre pédagogique restreint a imposé une réduction considérable des ambitions initiales du projet.

*
* *

La seconde expérience que je vais détailler a recours à l'interpolation en tant que moyen de découverte et d'exploration de phénomènes polyrythmiques complexes.

L'application directe de la fonction `interpolist` à des séquences rythmiques repose sur l'existence en Common LISP d'une arithmétique des nombres rationnels³⁶ et sur la notation rationnelle des durées par rapport à l'unité constituée par la ronde³⁷.

La fonction `interpolist` peut ainsi traiter directement des listes de nombres rationnels représentant des successions de figures de notes, c'est-à-dire de durées mesurées. De telles listes traduisent donc des rythmes. En spécifiant un rythme de départ, un rythme d'arrivée, et un nombre d'étapes, on obtient une transformation graduelle du rythme de départ en celui d'arrivée, à travers une succession de rythmes intermédiaires. Voici par exemple (cf. Figure 11) une évaluation d'`interpolist` avec son résultat :

```
(interpolist '(1/4 1/8 1/8 1/12 1/12 1/12)
             '(1/16 1/16 5/16 1/8 1/16 1/8)
             10)
(( 1/4 1/8 1/8 1/12 1/12 1/12)
 ( 11/48 17/144 7/48 19/216 35/432 19/216)
 ( 5/24 1/9 1/6 5/54 17/216 5/54)
 ( 3/16 5/48 3/16 7/72 11/144 7/72)
 ( 1/6 7/72 5/24 11/108 2/27 11/108)
 ( 7/48 13/144 11/48 23/216 31/432 23/216)
 ( 1/8 1/12 1/4 1/9 5/72 1/9)
 ( 5/48 11/144 13/48 25/216 29/432 25/216)
 ( 1/12 5/72 7/24 13/108 7/108 13/108)
 ( 1/16 1/16 5/16 1/8 1/16 1/8))
```

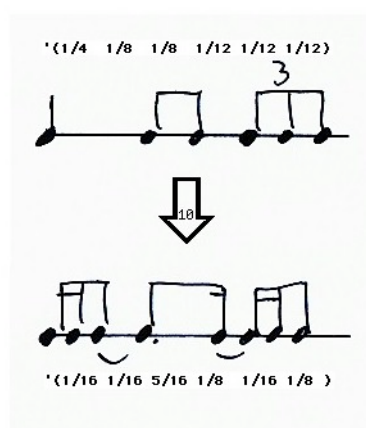


Figure 11 — Deux rythmes donnés, respectivement de départ et d'arrivée, présentés à `interpolist` afin d'obtenir une interpolation en dix étapes

³⁶ Les nombres rationnels sont exprimés comme des rapports de deux entiers (ex. : $-1/2$, $2/3$, $19/16$, $-57/43$ etc.). L'ensemble des rationnels inclut celui des entiers, et est lui même un sous-ensemble des nombres réels. Pour l'arithmétique rationnelle en Common LISP, cf. STEELE, *op. cit.*, pp. 18, 38, 288 sq.

³⁷ Il s'agit simplement de l'application de la terminologie anglo-saxonne moderne désignant les figures de notes en « fractions de ronde ». Par ex. $1/4$ = *quarter note* = noire ; $1/16$ = *sixteenth note* = double croche, etc. Cette notation permet de représenter toute durée mesurée possible. Par ex. $5/4$ = ronde liée à noire ; $6/28 = 3/14$ = croche pointée dans un septolet de triple-croches (7:8), etc.

La contrainte d'égalité de longueurs entre les listes de départ et d'arrivée lors de l'évaluation LISP peut être contournée par l'introduction d'une ou plusieurs durées nulles (zéro) dans la liste la plus courte (cf. Figure 12) :

```
(interpolist '(3/4 0 0 0 0 0)
              '(1/12 1/12 1/12 3/8 1/16 1/16)
              10)
((      3/4      0      0      0      0      0)
 ( 73/108 1/108 1/108 1/24 1/144 1/144)
 ( 65/108 1/54 1/54 1/12 1/72 1/72)
 ( 19/36 1/36 1/36 1/8 1/48 1/48)
 ( 49/108 1/27 1/27 1/6 1/36 1/36)
 ( 41/108 5/108 5/108 5/24 5/144 5/144)
 ( 11/36 1/18 1/18 1/4 1/24 1/24)
 ( 25/108 7/108 7/108 7/24 7/144 7/144)
 ( 17/108 2/27 2/27 1/3 1/18 1/18)
 ( 1/12 1/12 1/12 3/8 1/16 1/16))
```

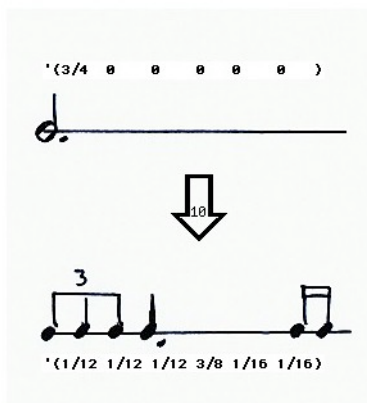


Figure 12 — Un rythme de départ, ne comportant qu'une note, est complété par cinq durées nulles (zéro) dans la liste de départ, afin de comporter autant d'éléments (six notes) que le rythme d'arrivée

En fondant en une seule liste toutes les sous-listes du résultat rendu par *interpolist*, nous pouvons ensuite exécuter l'enchaînement des étapes d'évolution graduelle du rythme au moyen d'un synthétiseur ou instrument MIDI quelconque³⁸.

Ce projet avait l'ambition d'explorer des phénomènes de transition entre ordre et désordre, chaos et déterminisme. Il aurait été possible, en plus du rythme, d'utiliser aussi par exemple le timbre comme champ d'expérimentation, entre bruits et spectres relativement purs. Le cadre du projet imposait cependant de s'en tenir à des limites assez réalistes ; nous avons donc choisi de réaliser concrètement notre expérience dans le domaine d'interpolations rythmiques allant de polyrythmies « désordonnées » à des homorythmies « ordonnées ».

³⁸ Cette étape ne sera pas détaillée ici. Elle exploite diverses fonctions personnelles faisant partie d'un cours plus approfondi (cf. *supra*), et repose sur *CLCE*, une extension MIDI de Macintosh Common LISP (<http://www.digitool.com>) conçue au GRAME (<http://www.grame.fr>). Cf. LETZ, Stéphane, MERLIER, Bernard et ORLAREY, Yann, *CLCE (Common LISP Compositional Environment)*, version 4b beta, Lyon : GRAME, révision septembre 1992.

Cette expérience fait obligatoirement penser au second principe de la thermodynamique, ou loi physique de l'entropie maximale, de James Maxwell et Ludwig Boltzmann, illustrée par l'autoglyphe de la Figure 13. Cette figure, et la loi qu'elle illustre, sont à l'évidence aussi des interpolations. D'après cette loi physique, l'évolution inéluctable de l'univers après

1. Dans les autoglyphes, le dessin rejoint le dessin : ils s'autodécrivent d'une manière indirecte et subtile (voir : <http://www.stanford.edu/~segerman/autoglyphs.html>).

Quant à l'entropie, elle n'échappe pas à la dégradation que la thermodynamique détermine.

[DELAHAYE, J.-P., 2005 : "Ceci n'est pas le titre", *Pour la science* : 327 : 88-89]

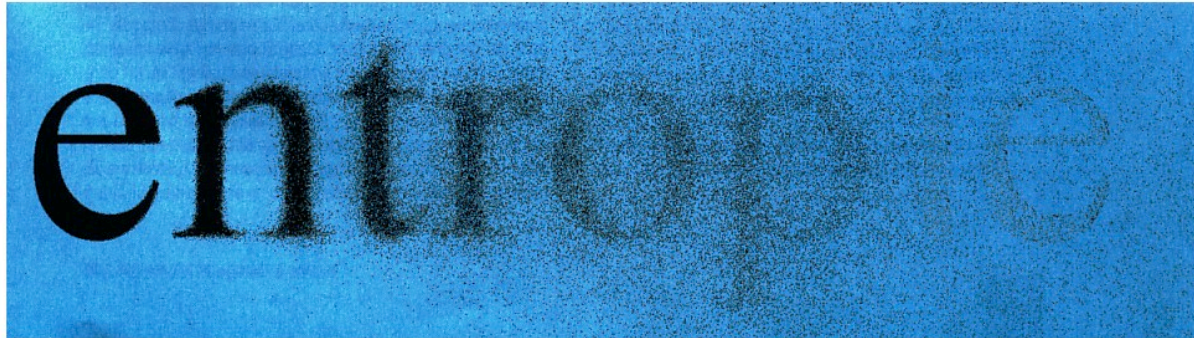


Figure 13 — Illustration du second principe de la thermodynamique, ou loi de l'entropie maximale. Un autoglyphe est une figure qui fait référence de manière imagée et récursive à son propre propos. Figure extraite de DELAHAYE, Jean-Paul, "Ceci n'est pas le titre", *Pour la science*, 327, pp. 88-89

l'explosion du *big bang* devrait l'amener à une répartition « uniformément informe » de matière tiède... Heureusement pour nous, on le sait, ce principe est localement contredit à une grande quantité d'échelles, depuis les galaxies jusqu'à la micro-biologie. Ce sont justement les zones intermédiaires, situées entre la pétrification déterministe et la « soupe » entropique tiède, qui permettent l'émergence de structures complexes³⁹ constituant notre propre essence et celle du monde tel qu'il se présente à nous.

Ces concepts se prêtent à une double interprétation lorsqu'on les transpose à la perception sonore, en posant, d'une part, que le silence infini, ou tout aussi bien un son statique infini⁴⁰, seraient analogues au déterminisme absolu (entropie nulle, à gauche de la Figure 13), et d'autre part que le bruit blanc incarnerait le désordre absolu (entropie maximale, à droite de la figure). On constate premièrement une sorte de transformation cyclique de ce qui est en principe une interpolation rectiligne : pour notre perception, le silence total ou toute sonorité continue, qu'elle soit pure ou bruiteuse⁴¹, sont également sans intérêt, tout aussi lassants l'un que l'autre, tout aussi insatisfaisant, tout aussi suppliciants. Même si la topologie de l'interpolation [*ordre ... désordre*] est, dans ce cas, radicalement transformée par la confusion de ses extrêmes, on y vérifie par contre, dans le monde sonore, le même principe que dans le monde physique : c'est dans la zone intermédiaire que se situent les sonorités « intéressantes » — suffisamment dynamiques et variées, mais pas trop désordonnées, etc. Et ce qui est vrai pour les sonorités elles-mêmes l'est tout autant pour la musique : ne se situe-t-elle pas quelque part entre le silence et le bruit ?

Notre première expérience, peut-être un peu extrême, mais destinée à vérifier le fonctionnement du processus, a consisté en une superposition de neuf couches (ou voix), chacune sur une hauteur d'un cluster de neuf sons, et dotée d'un rythme de départ particulier, convergeant toutes par interpolations vers une synchronisation homorythmique sur un rythme d'arrivée unique. Nous avons nommé ce processus *poly-interpolation rythmique* (Figure 14).

³⁹ D'où la notion même de *complexité*, au sens précis de ce terme dans la science contemporaine. Cf. par ex. *Pour la science*, 314, numéro spécial sur la complexité, 2003.

⁴⁰ Par exemple la sinusoïde théoriquement infinie chère au baron Joseph Fourier !

⁴¹ En dehors de questions de confort, évidemment...

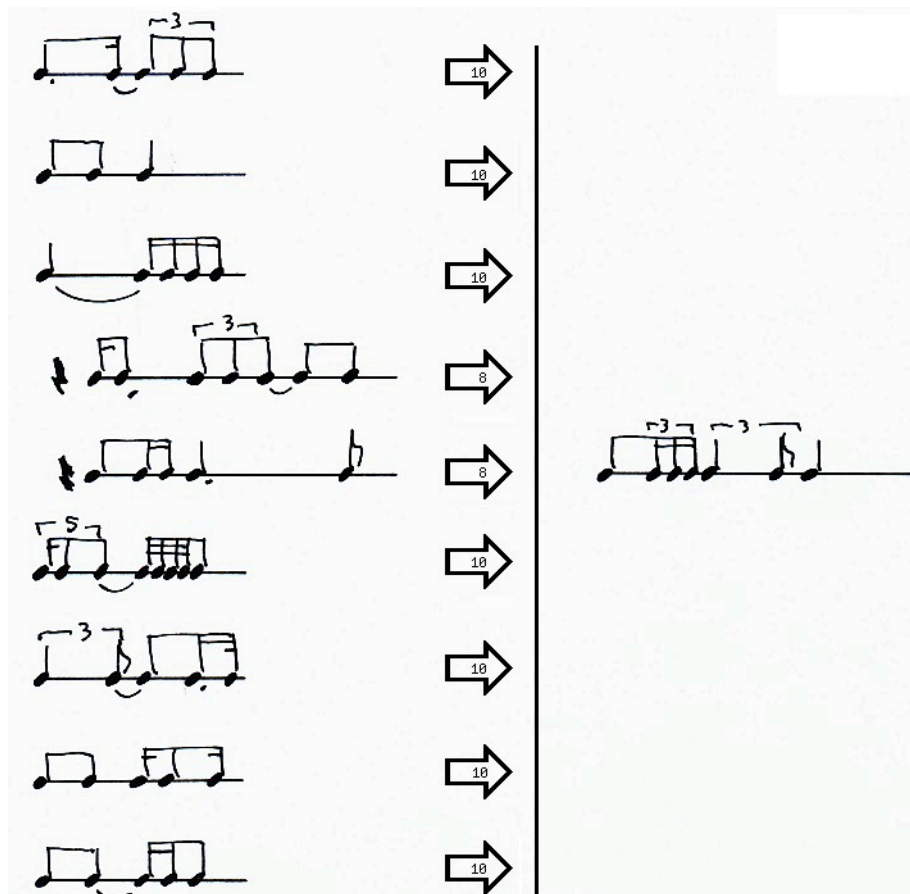


Figure 14 — Poly-interpolation rythmique de neuf couches. L'ensemble des couches superposées avec des rythmes de départ individuels produit initialement une polyrythmie « désordonnée ». Suit une phase où les couches interpolent graduellement, en huit ou dix étapes, vers une synchronisation totale sur un rythme d'arrivée unique

Conformément à ce qui précède, et à la citation de J. Eckardt ci-dessus, cet essai révèle que la part la plus intéressante du processus réside dans ses parties intermédiaires, davantage que dans les états initial et terminal.

On aura remarqué sur la Figure 14 que les rythmes de départ des couches n'étant pas tous de durées égales, la date de départ de chaque couche, ainsi que son nombre d'étapes d'interpolation — états successifs d'évolution de chaque rythme, y compris ses énoncés de départ et d'arrivée — doivent être individualisés de manière à ce que, à l'arrivée, l'ensemble des couches soit homorythmiquement synchrone. Ceci a été fait « manuellement » pour cette première expérience, au moyen d'évaluations individuelles⁴² pour chacune des couches, suivies de leur superposition dans un fichier MIDI :

```

;couche 1
  (setq liste-rythme (...(interpolist r-de1 r-fi1 10)))
  (setq liste-note (make-list (list-length liste-rythme)
                              :initial-element '(4 . 3)))
  (melos2.2 'xx lesco :deltat liste-rythme
             :pitch liste-note)

;couche 2
  (setq liste-rythme (...(interpolist r-de2 r-fi1 10)))
  (setq liste-note (make-list (list-length liste-rythme)
                              :initial-element '(8 . 2)))
  (melos2.2 'xx lesco :deltat liste-rythme
             :pitch liste-note)

```

⁴² Ces évaluations sont quelque peu abrégées ci-dessous.

```

;couche 3
(setq liste-rythme (...(interpolist r-de3 r-fil 10)))
(setq liste-note (make-list (list-length liste-rythme)
                           :initial-element '(9 . 2)))
(melos2.2 'xx lesc0 :deltat liste-rythme
          :pitch liste-note)

;couche 4
(setq liste-rythme (...(interpolist r-de4 r-fil 8)))
(setq liste-note (make-list (list-length liste-rythme)
                           :initial-element '(3 . 3)))
(melos2.2 'xx lesc0 :deltat liste-rythme
          :pitch liste-note
          :decale-t 1/4)

;etc.

```

Dans cet extrait de fichier LISP, on voit comment chaque couche a été préparée individuellement, avec son nombre d'étapes d'interpolation (les troisièmes arguments des évaluations `interpolist` ci-dessus sont 10 ou 8) et un éventuel délai de démarrage (par ex. valeur 1/4 donnée au mot-clef `:decale-t` pour la couche 4).

Nous avons ensuite conçu la fonction LISP `etapes?` destinée à gérer automatiquement ces dernières contraintes. Cette fonction résout les deux questions suivantes : pour chaque couche avec son rythme de départ donné, (1°) quand doit-elle démarrer ? et (2°) combien d'étapes d'interpolation doit-elle effectuer ? afin d'arriver parfaitement synchronisée avec toutes les autres dans une durée globale minimale donnée. L'organigramme d'un algorithme satisfaisant ces contraintes est présenté Figure 15, non pas pour ses détails intrinsèques, mais

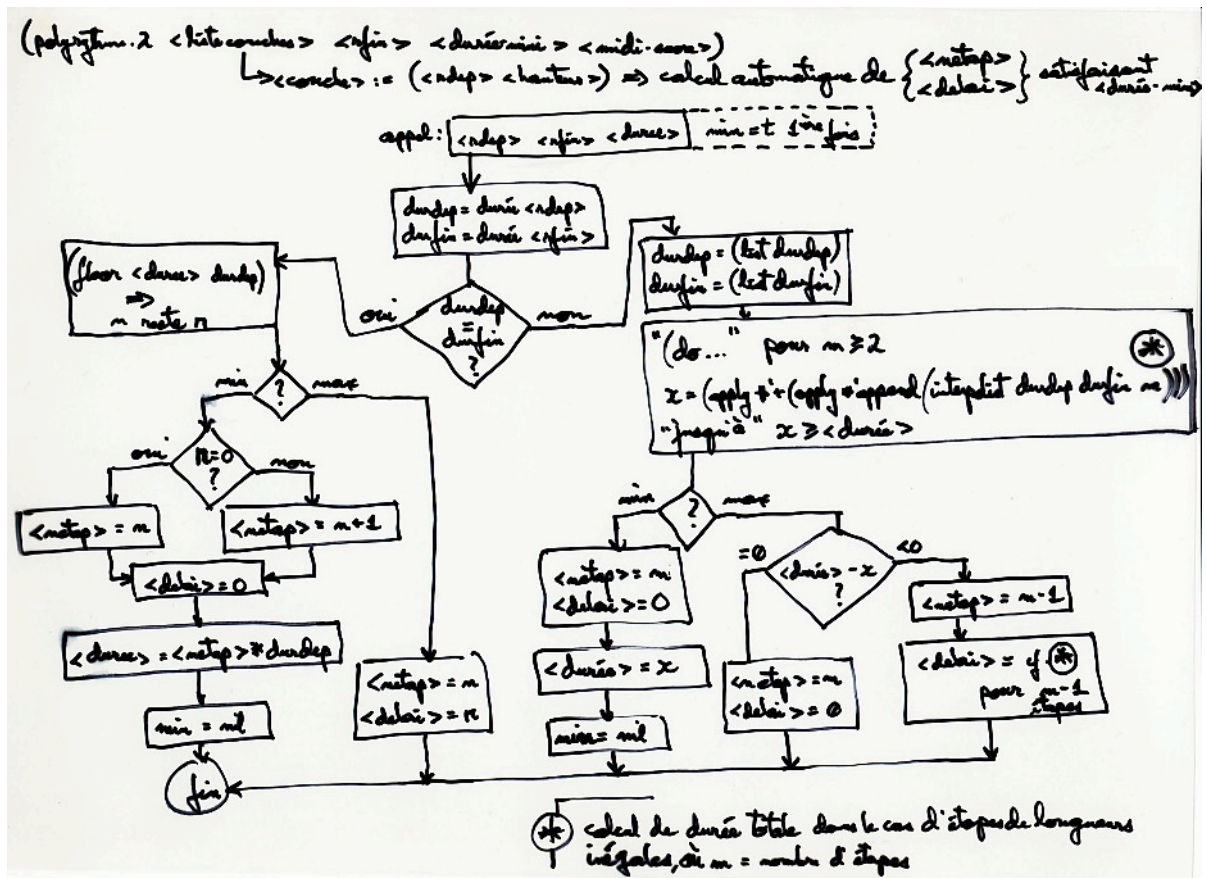


Figure 15 — Organigramme de l'algorithme de la fonction `etapes?` chargée de prévoir la date de démarrage et le nombre d'étapes de chaque couche d'une poly-interpolation rythmique, afin d'assurer la synchronisation finale de l'ensemble des couches

parce que sa relative complexité peut étonner, au regard du cahier des charges qui vient d'être énoncé si brièvement. Il est intéressant de noter que certains problèmes d'apparence assez simple appellent des solutions informatiques assez élaborées. Tout comme le contraire, d'ailleurs : des opérations informatiques simples peuvent parfois accomplir des tâches en apparence sophistiquées. Les complexités informatique et humaine ne se recourent pas nécessairement.

Une fois acquise la fonction `etapes?`, une poly-interpolation rythmique pouvait être commandée par une seule évaluation LISP d'une fonction `polyrythme.2`, elle-même appelant `etapes?`. Voici l'évaluation réalisant la poly-interpolation présentée Figure 16 :

```
(polyrythme.2 '(
  ( (1/6 0 11/96 1/32 1/32 1/32 3/16) (5 . 4) ) ;couche 5
  ( (1/4 0 1/16 19/80 0 1/10 1/10) (4 . 4) ) ;couche 4
  ( (1/8 1/24 1/24 7/24 1/16 0 3/8 ) (9 . 3) ) ;couche 3
  ( (1/8 1/8 1/16 1/16 1/8 0 3/16) (2 . 3) ) ;couche 2
  ( (1/12 1/12 1/12 3/8 3/16 1/16 1/8 ) (0 . 3) ) ;couche 1
  '( 3/16 1/16 5/16 0 3/8 0 0 ) ;<rarr> rythme arrivee
  15 ;<duree_mini> = 15
  'partition) ;<midi-score>
```

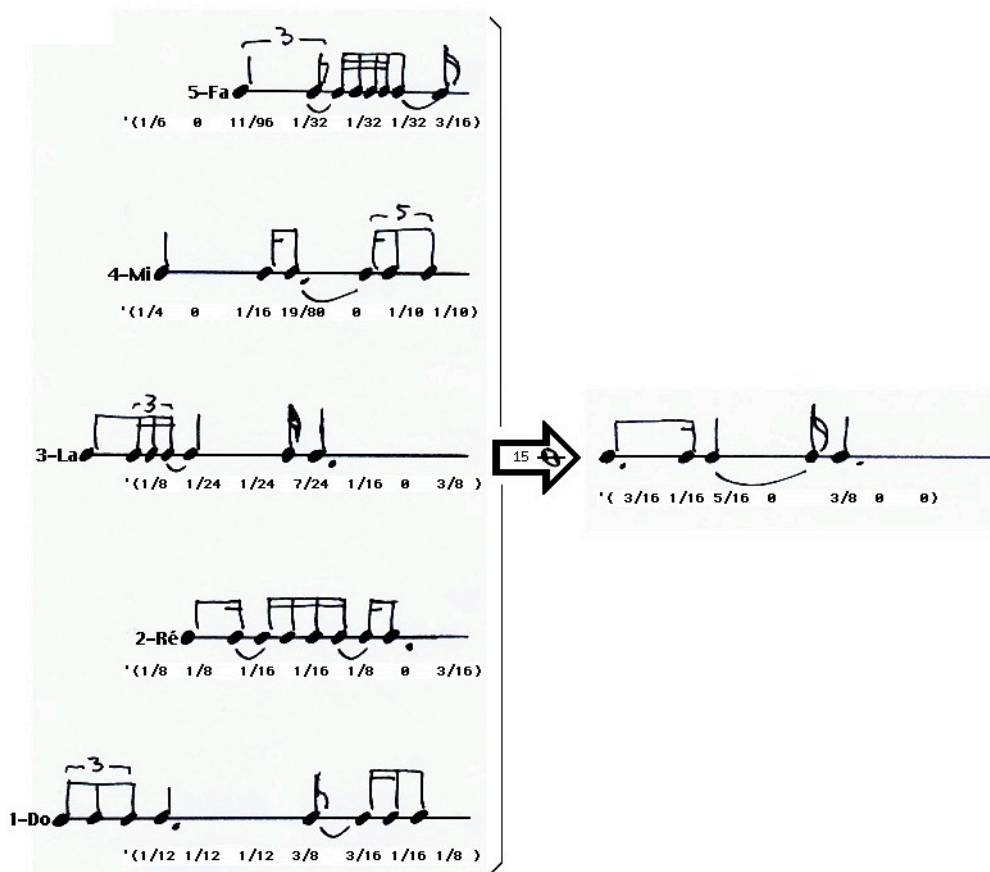


Figure 16 — Poly-interpolation rythmique de cinq couches dotées de rythmes de départ individuels de longueurs différentes. La durée globale demandée étant d'au moins quinze rondes, la fonction `etapes?` calcule automatiquement la date de démarrage et le nombre d'étapes d'interpolation de chacune des couches

On y voit les cinq rythmes de départ et hauteurs individuels — par ex. $(\frac{1}{6} \ 0 \ \frac{11}{96} \ \frac{1}{32} \ \frac{1}{32} \ \frac{1}{32} \ \frac{3}{16})$ et $(5 \ . \ 4)$ respectivement, pour la couche 5 — ainsi que le rythme commun d'arrivée $(\frac{3}{16} \ \frac{1}{16} \ \frac{5}{16} \ 0 \ \frac{3}{8} \ 0 \ 0)$, et une durée globale demandée de 15 unités (rondes).

*
* *

Les expériences que je viens de décrire n'atteignent manifestement pas à un niveau véritablement musical. Elles sont fonctionnelles, mais il leur manque une mise en œuvre réelle et une corrélation avec un contexte musical, conformément à ce qu'énonce J. Eckardt dans la suite de la citation reproduite ci-dessus en introduction :

« Bien que les processus dictent certaines façons dont le matériau sonore peut se comporter, d'autres aspects restent libres, permettant un équilibre entre indépendance locale et cohésion globale. Les processus s'achèvent par l'épuisement des possibilités dans une dimension musicale.⁴³ »

Nos expériences remplissent néanmoins parfaitement leur rôle pédagogique, consistant à mener à terme la réalisation de processus complets et autonomes, définis d'après des objectifs initiés par les étudiants. Mais des conclusions pédagogiques importantes peuvent être tirées de ce que (1°) la définition même de chaque projet ait dû être considérablement réduite par rapport aux ambitions originelles, et que (2°) les limites du cours n'aient ainsi pas permis d'intégrer une véritable mise en œuvre musicale.

Concrètement, les étudiants compositeurs inscrits dans cette option⁴⁴ ne possèdent pas suffisamment d'expérience préalable pour évaluer la complexité de la réalisation technique d'une idée, ni la quantité de temps de développement nécessaire⁴⁵. La naïveté technologique masquant les difficultés des applications de l'informatique est caricaturée dans la question suivante, relevée dans une revue musicale :

« J'aimerais savoir si le matériel informatique existe pour la création d'œuvres musicales par des gens *qui ne connaissent strictement rien à l'informatique ni à la musique* ?⁴⁶ »

On croit relire la promotion aguichante des jeux musicaux de société populaires au XVIII^{ème} siècle : « *INSTRUCTION Pour composer autant de Walzes que l'on veut par le moyen de deux Dez sans avoir la moindre connoissance de la Musique ou de la Composition*⁴⁷ » ! Ironiquement, je présente toujours aux étudiants débutants l'illustration d'une application informatique idéale à laquelle ils ont peut-être rêvé (Figure 17) mais que, malheureusement...

⁴³ *Ibid.*

⁴⁴ Je rappelle qu'ils ne sont pas spécialisés en électroacoustique et informatique musicales, cf. *supra*.

⁴⁵ Toutes options confondues, en incluant donc des étudiants plus spécialisés, j'estime à environ 25% les débutants de première année capables de telles estimations. Une part de mon rôle d'enseignant consiste donc souvent à modérer les ambitions initiales de tels projet De toute façon, la réalité finit toujours par s'imposer...

⁴⁶ Courrier des lecteurs, *La lettre du musicien*, 253, 2^e quinzaine de mai 2001, p. 11. C'est moi qui souligne.

⁴⁷ MOZART, W. A., Paris : Vogt, ca. 1800 (In-folio, 6 p. Paris BN Vm⁸ 605). Edition originale : *Anleitung zum Componiren von Walzern so viele man will vermittelst zweir Würfel ohne etwas von der Musik oder Composition zu verstehen* (sic), Berlin : Simrock, 1793.

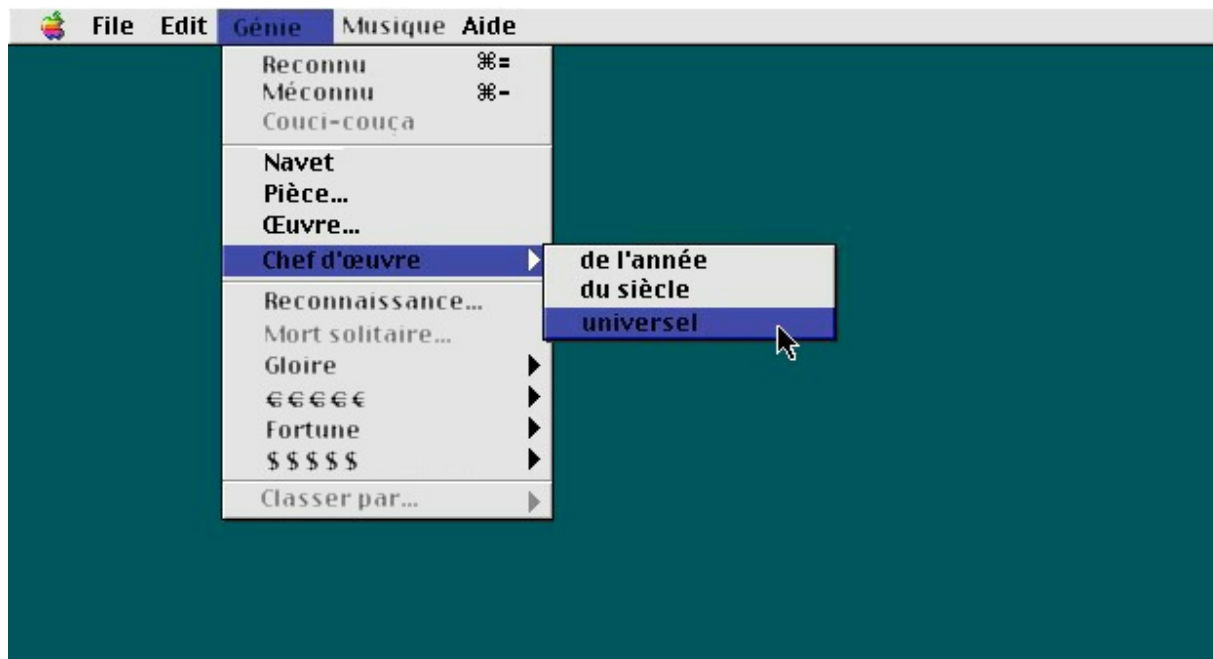


Figure 17 — Menu « Génie » d'une application... aussi idéale que fictive !

il ne dépendra que d'eux de réaliser ! À une telle innocence, on ne peut opposer que la réponse ardue et désabusée de Rimbaud : « Je connais le travail ; et la science est trop lente.⁴⁸ »

Plus profondément, la réalisation de tels projets met en évidence un conflit pédagogique latent entre, d'une part, des objectifs à court terme, et, d'autre part, l'aménagement de perspectives à long terme. Il est essentiel que le parcours pédagogique des étudiants soit mis à profit pour approfondir des sujets fondamentaux, mais cela ne peut se faire qu'au prix d'une certaine distanciation par rapport à des objectifs concrets et immédiats. La seule résolution possible de cette tension est dans le temps, dans la durée :

« Voilà vingt ans que je répète que la maîtrise d'un ordinateur ne s'acquiert pas en quelques jours. C'est comme un instrument de musique : ça ne s'utilise pas, ça se pratique. Et dans la singularité de l'instrument, on va découvrir la singularité de sa propre pratique. La pratique ouvre à l'expérience, alors que la consommation ne produit que le conditionnement.⁴⁹ »

Dans une réflexion sur les implications profondes de l'informatique dans la destinée humaine, et dans les arts en particulier, j'avais eu l'occasion de décrire l'approche pédagogique difficile qui tente d'agir au cœur du compromis entre la satisfaction d'objectifs immédiats et l'édification de bases conceptuelles solides :

« Nous devons [...] prendre garde de ne pas déséquilibrer l'usage que nous ferons de [l'informatique], de ne pas laisser prédominer les dangers qu'elle porte en elle — rentabilisation, déstructuration, saturation et manipulation mentales. Les artistes, et particulièrement les musiciens, sont naturellement parmi les principaux acteurs susceptibles de veiller à un équilibre général entre les voies d'avenir et les inévitables nécroses contenues en germe dans cette avancée.

« L'implication totale des nouvelles technologies, non seulement en musique, non seulement dans le domaine artistique, mais diffusées à travers toute la substance même de la vie contemporaine, impose une redéfinition des notions de réalité, d'art et d'œuvre d'art, de créativité même ; elle laisse notre société dans l'urgence, avec un besoin criant d'artistes compétents, prenant appui sur une authentique *culture technologique* pour relever les défis

⁴⁸ RIMBAUD, Arthur, *Une saison en enfer, L'éclair*.

⁴⁹ STIEGLER, Bernard, "Sauvons le capitalisme", *Technikart*, novembre 2005.

qui se dressent devant elle. L'informatique est une chose beaucoup trop sérieuse pour la laisser entièrement aux informaticiens !

« Il n'existe pas de solution universelle. L'approche tentée à SONVS consiste, sans négliger les aspects concrets d'un apprentissage technique et pratique, à promouvoir l'étude et la mise en oeuvre d'*environnements ouverts*. Nous espérons ainsi doter les étudiants d'une capacité d'adaptation qui leur sera essentielle pendant leur parcours professionnel, et leur permettra d'avoir l'*initiative* de futurs développements technologiques, plutôt que d'attendre passivement des produits commerciaux plus ou moins satisfaisants. Il est nécessaire que les jeunes professionnels posent un regard incisif et créatif sur les technologies, et possèdent les connaissances théoriques ainsi que les structures conceptuelles robustes qui leur permettront d'affronter des situations imprévisibles à l'échelle de leur carrière future. Notre société n'aura rien à faire, dans l'avenir, de ceux qui connaîtront les réponses d'*aujourd'hui*... Ce qu'il faut, c'est contribuer à mettre le pied à l'étrier à ceux qui poseront les bonnes questions de *demain*.⁵⁰ »

⁵⁰ LORRAIN, Denis, *SONVS, X^{ème} anniversaire*, Lyon : CNSMD, 1999.