Le son numérique

au delà du mp3

Charlotte Truchet
LINA, Université de Nantes

Plan du cours

- Introduction
- Représentations du son
- Exemple d'algorithme sur le son : *Karplus-Strong*
- Exemple d'algorithme en musique : *oracle des facteurs - OMax*
- Conclusion

Plan du cours

- Introduction
- Représentations du son
- Exemple d'algorithme sur le son : *Karplus-Strong*
- Exemple d'algorithme en musique : *oracle des facteurs - OMax*
- Conclusion

Exemples

Jingle d'Europe 1.
Glockenspiel
A l'antenne depuis **1956**

Jingle de RTL.
Trompettes et trombonnes. Compositeur : Michel Legrand.
A l'antenne depuis **1964**

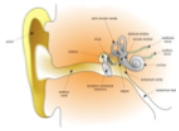
Signalétique SNCF (1994-2004)
IRCAM/SNCF
Design sonore

Signalétique SNCF (actuel)
Identité sonore
Agence *Sixième son* «Renforce l'image humaine, créative, moderne et ambitieuse de la SNCF»
(source : http://www.sixiemeson.com/fr/references/sncf/)

Son

Le son est une relation complexe impliquant

- un objet vibrant (instrument, HP, ...)
- un support de transmission (généralement l'air)



- un récepteur (oreille, micro)

- un interpréteur (cerveau)

Son

Définition

Dans le dictionnaire Larousse en ligne :

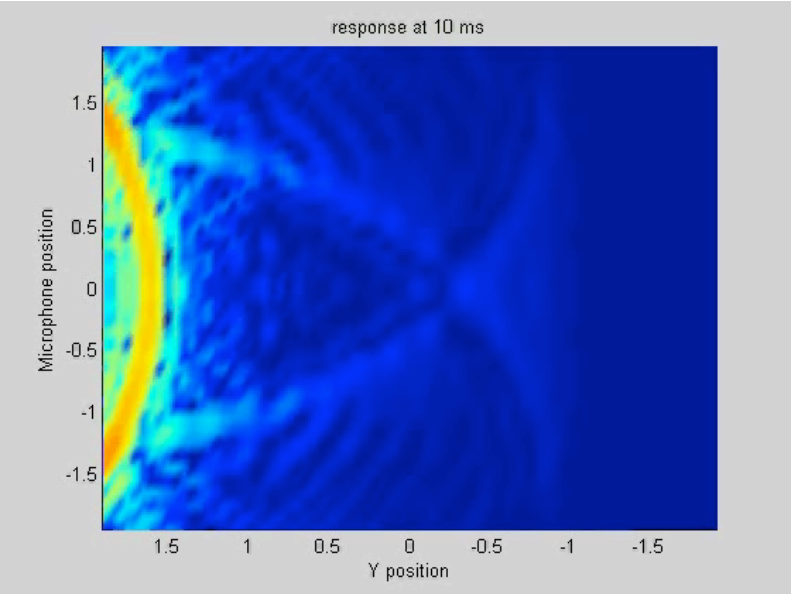
«

1. Sensation auditive engendrée par une onde acoustique.
2. Toute vibration acoustique considérée du point de vue des sensations auditives ainsi créées : *Son strident.*
3. Volume, intensité sonore d'un appareil : *Baisser le son.*
4. Ensemble des techniques d'enregistrement et de reproduction des sons, en particulier au cinéma, à la radio, à la télévision : *Ingénieur du son.* »

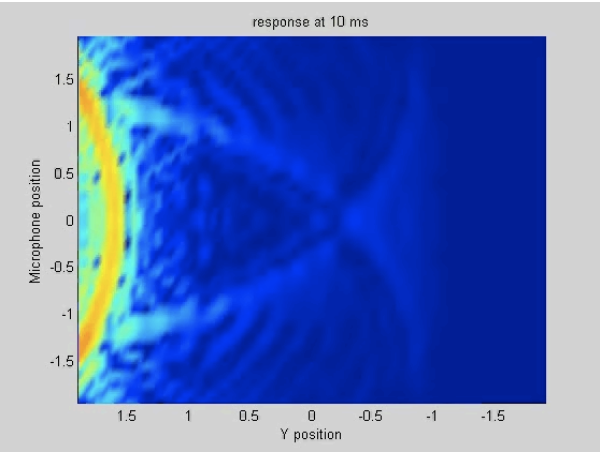
Source : http://www.larousse.fr/dictionnaires/francais/son/73436

Son

Onde produite par une vibration



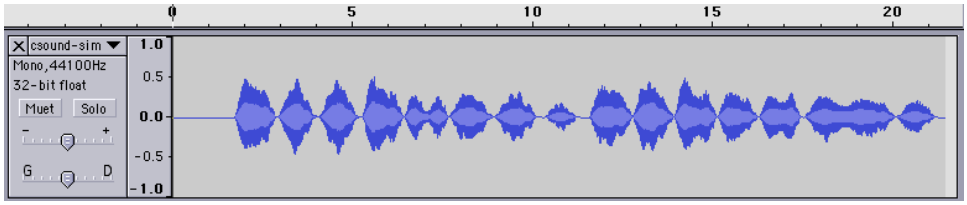
Format numérique



Continu : variation de pression (réel) au cours du temps (réel)

Discret : dans un ensemble fini de valeurs

Stockable sur une machine

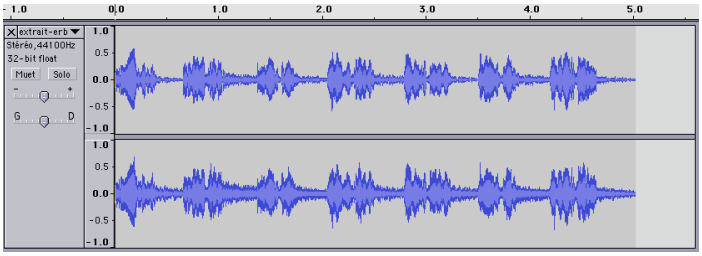


Son numérique

Psychologie / cognition

Mathématiques

Physique



Musique / musicologie

Informatique

Son numérique

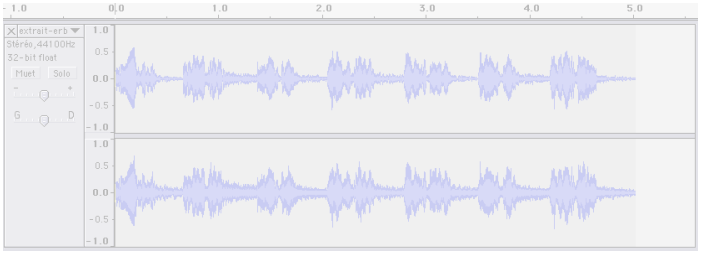
Psychologie / cognition

son = caractéristiques psychoacoustiques

design sonore, conception industrielle...

Mathématiques

Physique



Musique / musicologie

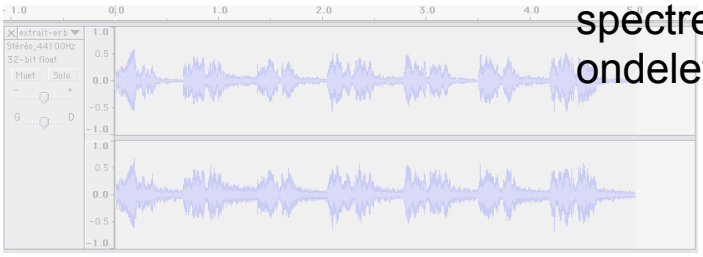
Informatique ?

Son numérique

Psychologie / cognition

Mathématiques
son = objet math
spectre, FFT,
ondelettes...

Physique



Musique / musicologie

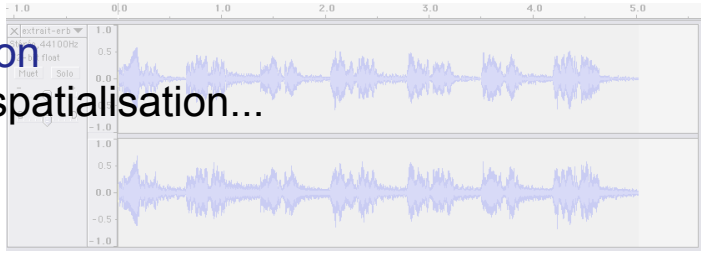
Informatique ?

Son numérique

Psychologie / cognition

Mathématiques

Physique
son = vibration
acoustique, spatialisation...



Musique / musicologie

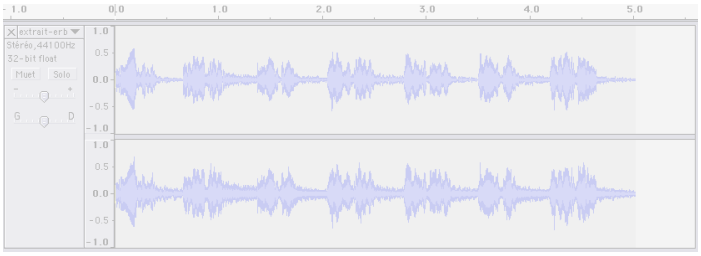
Informatique ?

Son numérique

Psychologie / cognition

Mathématiques

Physique



Musique / musicologie
son = élément musical
création artistique, analyse...

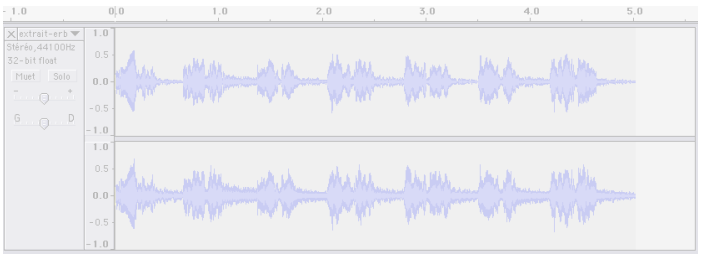
Informatique ?

Son numérique

Psychologie / cognition

Mathématiques

Physique



Musique / musicologie

Informatique ?

Domaines scientifiques

Informatique ?
son = donnée numérisée, calculable

- représentation du son
- calculs / algorithmique
- langages de synthèse, de traitement, de musique...

Souvent en interaction avec les domaines précédemment cités.

Attention !

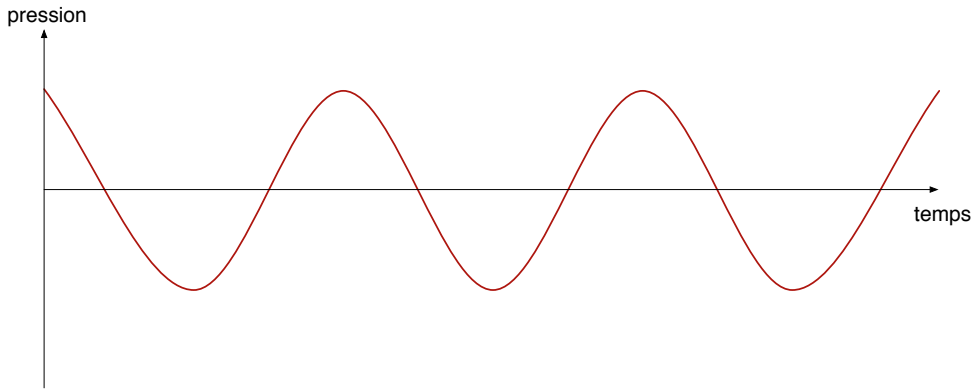
Ce n'est pas parce qu'un son est stocké numériquement que l'on sait quoi en faire

- Sur le son, les caractéristiques haut niveau ne sont pas entièrement maîtrisées
  - ex : hauteur = fréquence fondamentale
  - Illusions sonores [Risset, ~1969]
- En musique, la modélisation est difficile pour des raisons assez similaires au traitement des langues naturelles

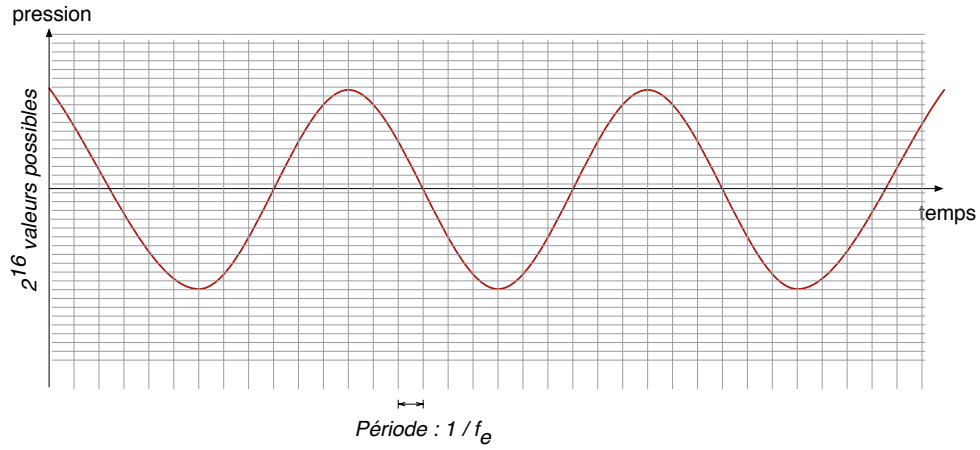
Plan du cours

- Introduction
- Représentations du son
- Exemple d'algorithme sur le son : *Karplus-Strong*
- Exemple d'algorithme en musique : *oracle des facteurs - OMax*
- Conclusion

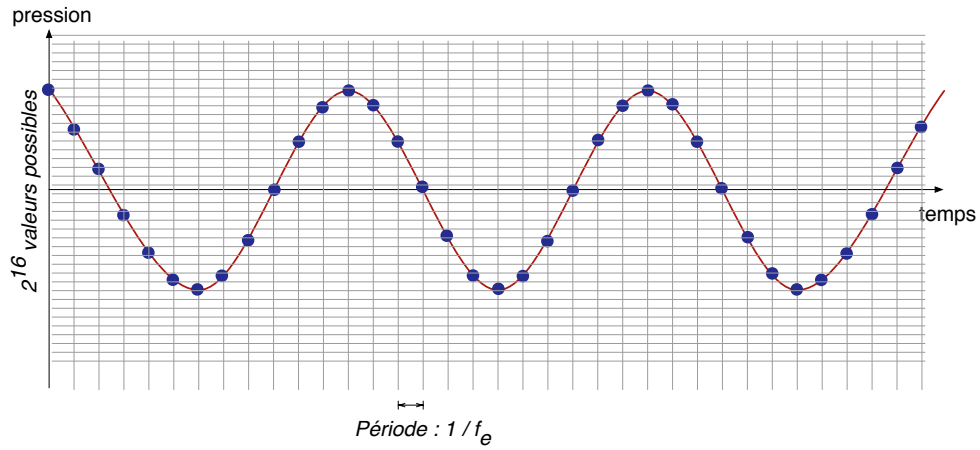
Numérisation



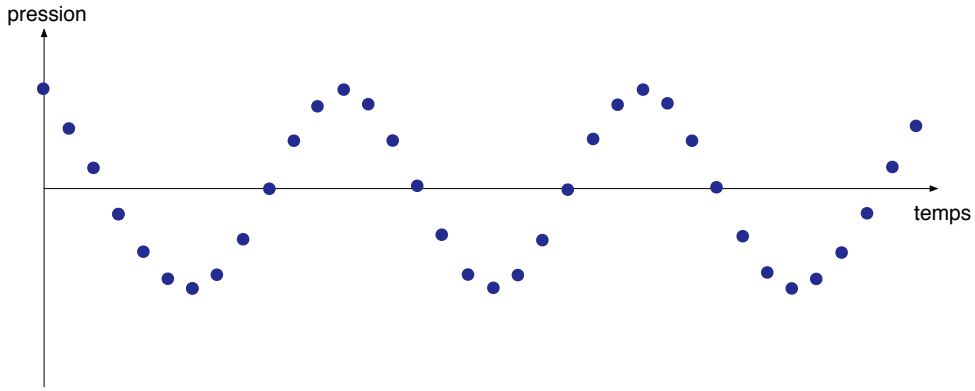
Numérisation



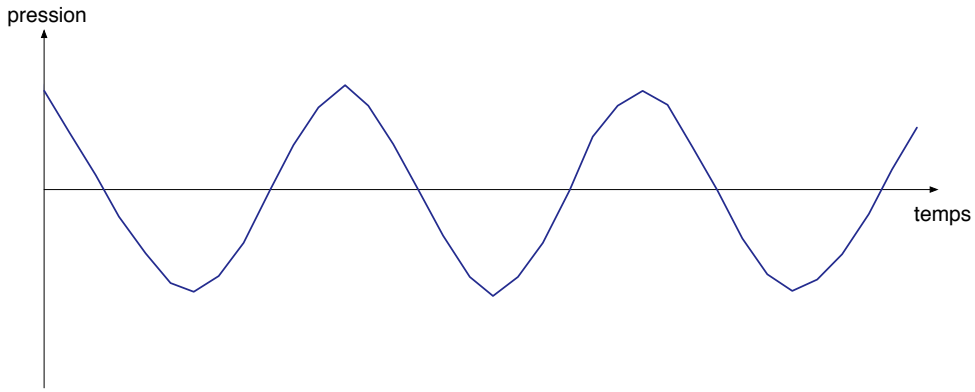
Numérisation



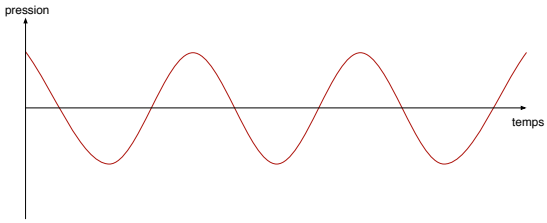
Numérisation

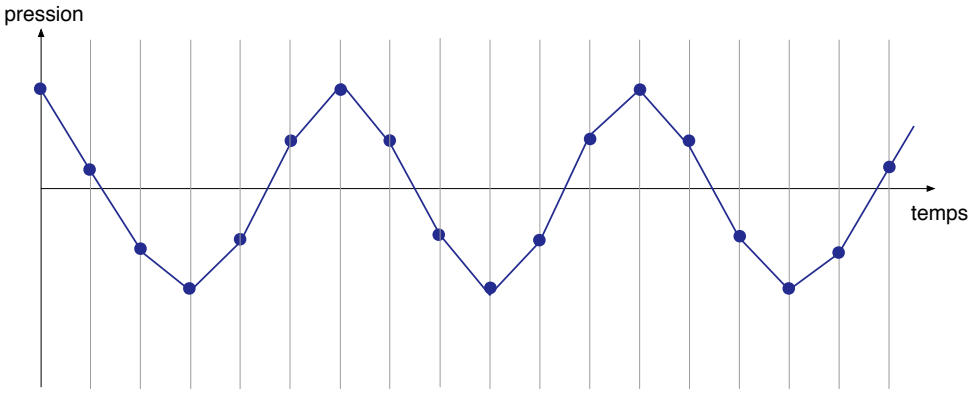


Numérisation



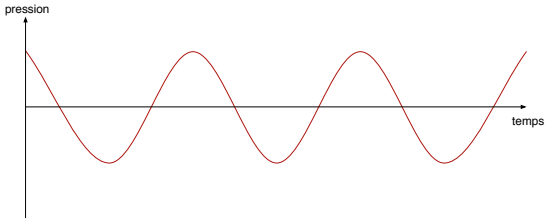
Numérisation

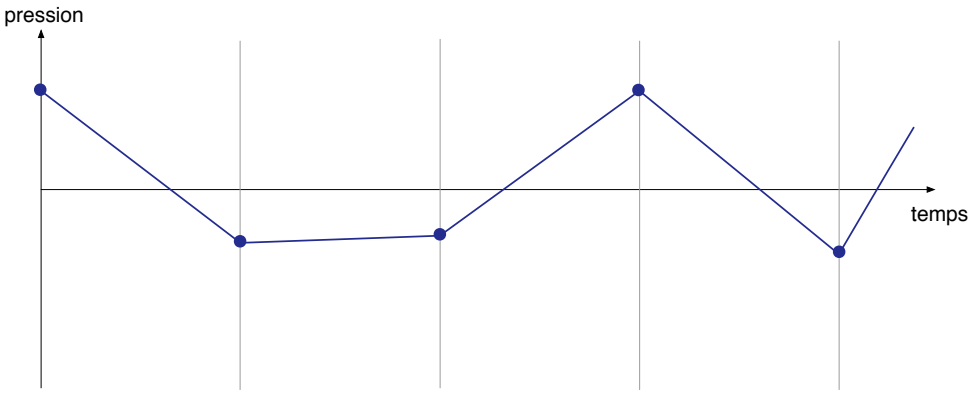




fréquence d'échantillonnage plus petite

Numérisation





fréquence d'échantillonnage vraiment trop petite

Echantillonnage

- Le son est stocké comme une suite de valeurs (échantillons) donnant l'amplitude du signal sonore en fonction du temps
- Temps discret, déterminé par la fréquence d'échantillonnage f_e (> deux fois la plus haute fréquence [Nyquist Shannon])
- Valeurs stockées sur un nombre de bits adapté à la précision souhaitée (de 4 à 24)
- Sur un nombre de canaux souhaité (1, 2, 4, 5...)

Echantillonnage

Fréquence d'échantillonnage
vs occupation mémoire

- Occupation mémoire
  = (durée en secondes) * (nombre de bits) / f_e
- Exemples
  - parole, 11kHz, 8 bits :
    une seconde = 11ko en monophonie
  - musique (qualité CD), 44kHz, 16 bits :
    une seconde = 176ko en stéréophonie

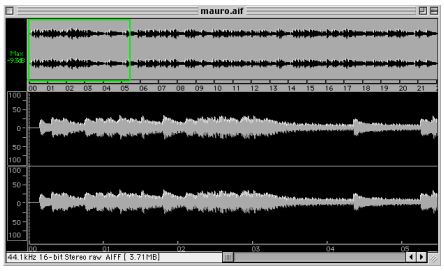
Echantillonnage

Nota bene

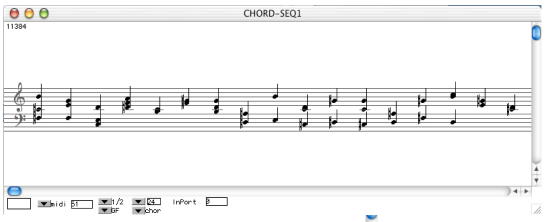
Ne pas confondre :

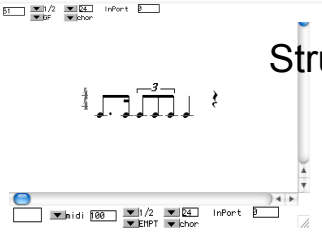
- numérisation
- codage ou format.

Une fois que le son est devenu une donnée numérique, on peut choisir de multiples niveaux de représentation plus ou moins précis et structurés

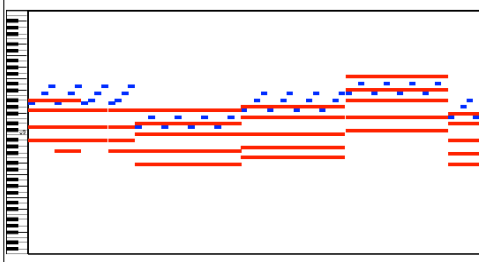


Fichier son



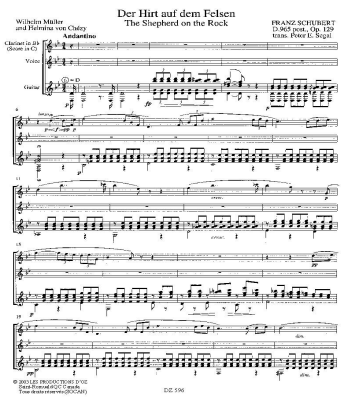


Structures musicales



Fichier MIDI

Partition



Plan du cours

- Introduction
- Représentations du son
- Exemple d'algorithme sur le son : *Karplus-Strong*
- Exemple d'algorithme en musique : *oracle des facteurs - OMax*
- Conclusion

Karplus-Strong

Un grand classique

- Algorithme de synthèse sonore proposé par Strong et Karplus [*Computer Music Journal* 1983] (et inventé un peu par hasard)
- Très simple, économe en mémoire et en calculs
- Bon rendu sonore pour une corde pincée (guitare)
- Alternative aux méthodes de synthèse souvent lourdes

Retour sur les fréquences

- Son pur = sinus

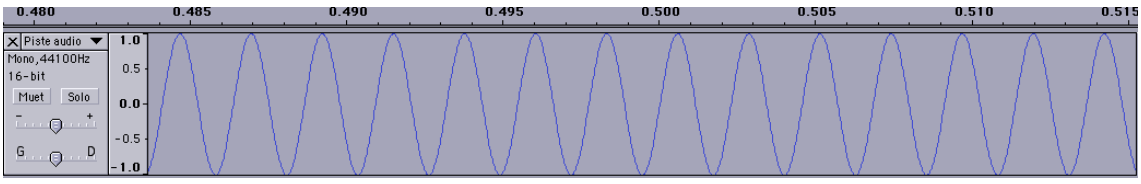
  La fréquence du sinus correspond à la hauteur perçue.

- Son harmonique = somme d'un sinus à une fréquence *f* et d'autres sinus à des fréquences multiples de f)

  La fréquence fondamentale *f* correspond *à peu près* à la hauteur perçue.

Retour sur les fréquences

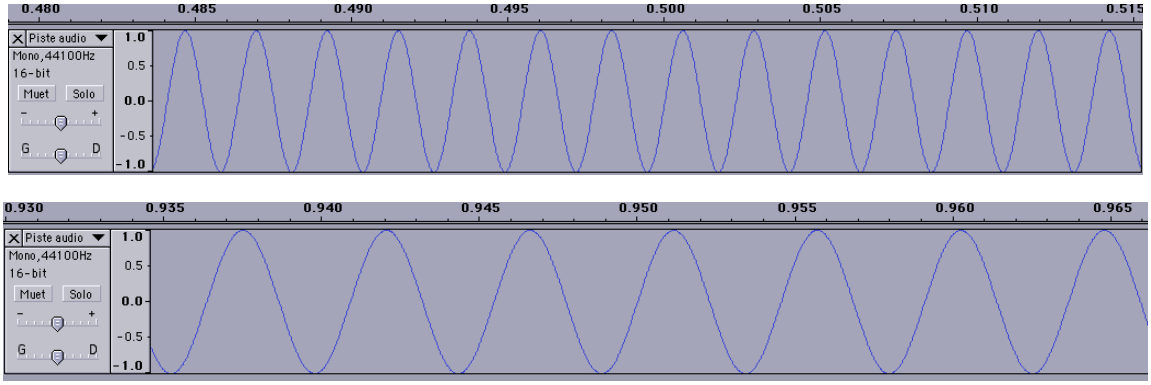
Sinus



Fréquence du sinus : 440 Hz (note *la* du diapason)

Retour sur les fréquences

Sinus



NB : un saut d'une octave correspond à une fréquence multipliée par 2

Retour sur les fréquences

- Son pur = sinus

  La fréquence du sinus correspond à la hauteur perçue.

- Son harmonique = somme d'un sinus à une fréquence *f* et d'autres sinus à des fréquences multiples de f)

  La fréquence fondamentale *f* correspond à la hauteur perçue.

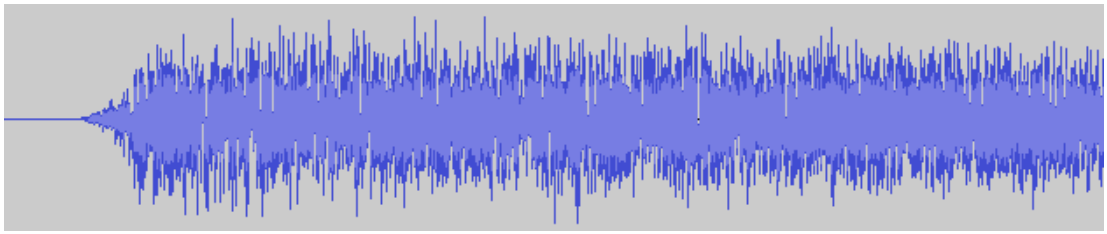
- Son inharmonique = somme de sinus à des fréquences quelconques

  S'il y a une fréquence fondamentale, elle correspond *à peu près* à la hauteur perçue.

Retour sur les fréquences

Extrême inharmonique : bruit blanc

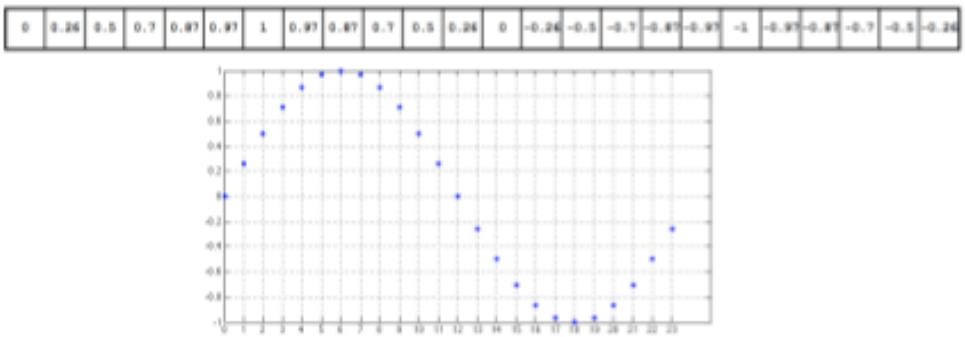
- Son constitué de valeurs aléatoires
- Pas de hauteur perçue



Synthèse par table d'onde

Principe

- Table de valeurs lues en boucle à une fréquence donnée



Karplus-Strong

Principe : corde pincée

- Table d'onde (tableau) évoluant au cours du temps - on imagine que la table représente une corde de guitare
- Au départ, la table est remplie par un son aléatoire (bruit blanc) qui correspond à un *pluck* de la corde
- Si on lit la table répétitivement à une fréquence *f*, on obtient un son de fréquence fondamentale *f*

Karplus-Strong

Principe : corde pincée

- En fait, on joue la table à une fréquence *f*, mais en la modifiant légèrement à chaque fois
- A chaque nouveau cycle, les valeurs de la table sont remplacées par les moyennes des valeurs du cycle précédent : agit comme un filtre
- Un petit calcul montre que ce filtre divise la fréquence par (N+1)/2 : la fréquence entendue est f/((N+1)/2)

Algorithme

Donné : un entier N (période), un autre entier M (répétitions)
Variables
t : tableau de N*M réels;
i,j : entier;
t[0]=0;
Pour i allant de 1 à N faire
t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
Pour i allant de 1 à N faire
t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
Finpour
Finpour

Algorithme : exemple

N=4, M=4

t = | 0 | | | | | | | | | | | | | | | |

t[0]=0;
Pour i allant de 1 à N faire
t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
Pour i allant de 1 à N faire
t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
Finpour
Finpour

Algorithme : exemple

N=4, M=4

t =

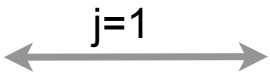
0	0.1	0.3	0.8	0.2											

```
t[0]=0;
Pour i allant de 1 à N faire
        t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
        Pour i allant de 1 à N faire
                t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
        Finpour
Finpour
```

Algorithme : exemple

N=4, M=4



t =

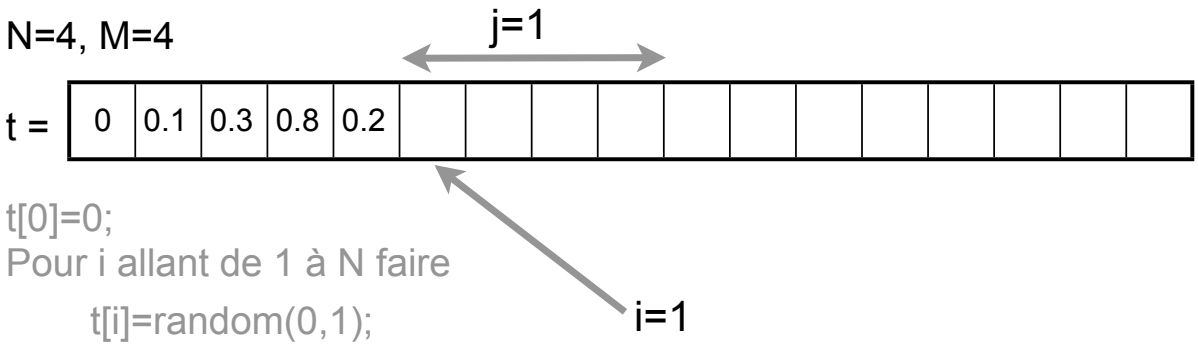
0	0.1	0.3	0.8	0.2											

```
t[0]=0;
Pour i allant de 1 à N faire
        t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
        Pour i allant de 1 à N faire
                t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
        Finpour
Finpour
```

Algorithme : exemple

N=4, M=4

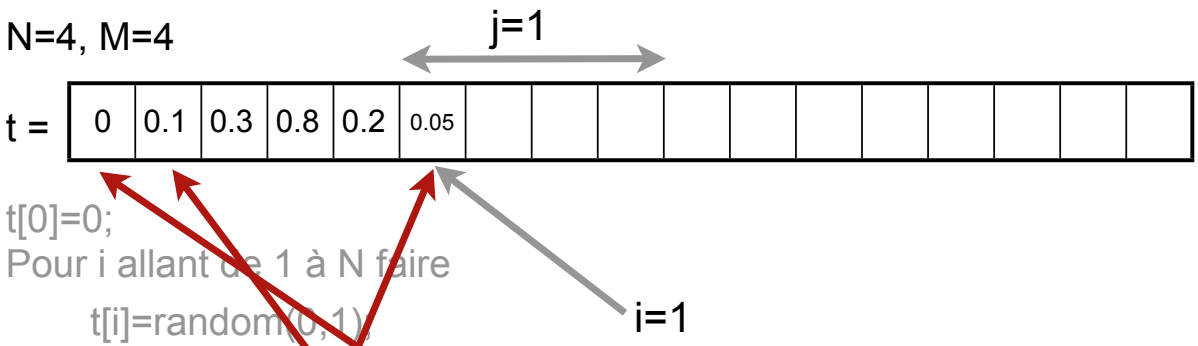



```
t[0]=0;
Pour i allant de 1 à N faire
        t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
        Pour i allant de 1 à N faire
                t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
        Finpour
Finpour
```

Algorithme : exemple

N=4, M=4

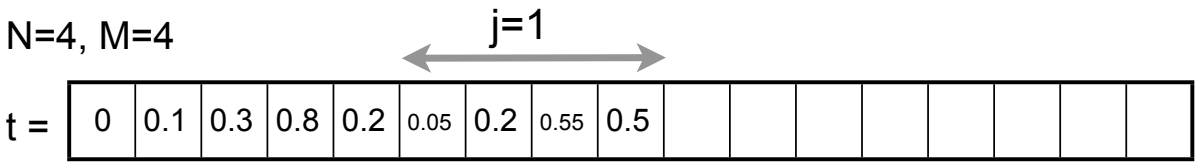



```
t[0]=0;
Pour i allant de 1 à N faire
        t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
        Pour i allant de 1 à N faire
                t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
        Finpour
Finpour
```

Algorithme : exemple

N=4, M=4

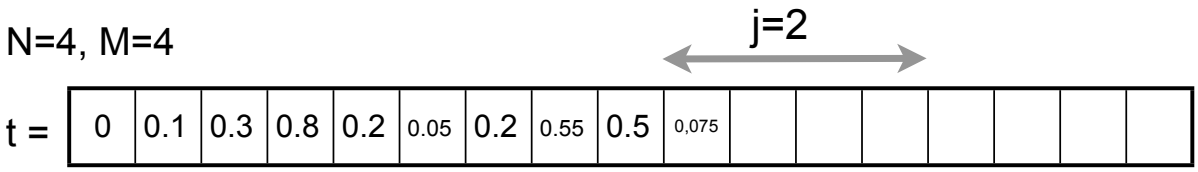



```
t[0]=0;
Pour i allant de 1 à N faire
        t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
        Pour i allant de 1 à N faire
                t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
        Finpour
Finpour
```

Algorithme : exemple

N=4, M=4




```
t[0]=0;
Pour i allant de 1 à N faire
        t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
        Pour i allant de 1 à N faire
                t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
        Finpour
Finpour
```

i=1

Algorithme : exemple

N=4, M=4

t =

0	0.1	0.3	0.8	0.2	0.05	0.2	0.55	0.5	0,075	0,175	0,025	...				

```
t[0]=0;
Pour i allant de 1 à N faire
        t[i]=random(0,1);
Finpour

Pour j allant de 1 à M faire
        Pour i allant de 1 à N faire
                t[j*N+i] = (t[(j-1)*N+i] + t[(j-1)*N+i-1]) /2;
        Finpour
Finpour
```

Algorithme : résultats

Quelques exemples

- NB : l'algorithme est souvent amélioré avec quelques paramètres supplémentaires (atténuation, probabilités...)
- Exemples issu du cours en ligne de Julius O. Smith, CCRMA, Stanford University

A retrouver en Algoscript

A tester sur Madoc !
Essayez de modifier les paramètres ou d'améliorer l'algorithme.

```
var data = [];

function KarplusStrong(freq, attenuation) {
    var Decalage = Math.floor(22050 / freq);
    for (var i = 0; i < Decalage; i++) {
        data[i] = Hasard(256);
    }
    for (var i = Decalage; i < 22050 / 3; i++) {
        data[i] = 128 + Math.floor(attenuation * ((data[i - Decalage] +
        data[i - Decalage + 1]) / 2 - 128));
    }
    return CreerSon(data, 22050);
}
```

Plan du cours

- Introduction
- Représentations du son
- Exemple d'algorithme sur le son : *Karplus-Strong*
- Exemple d'algorithme en musique : *oracle des facteurs - OMax*
- Conclusion

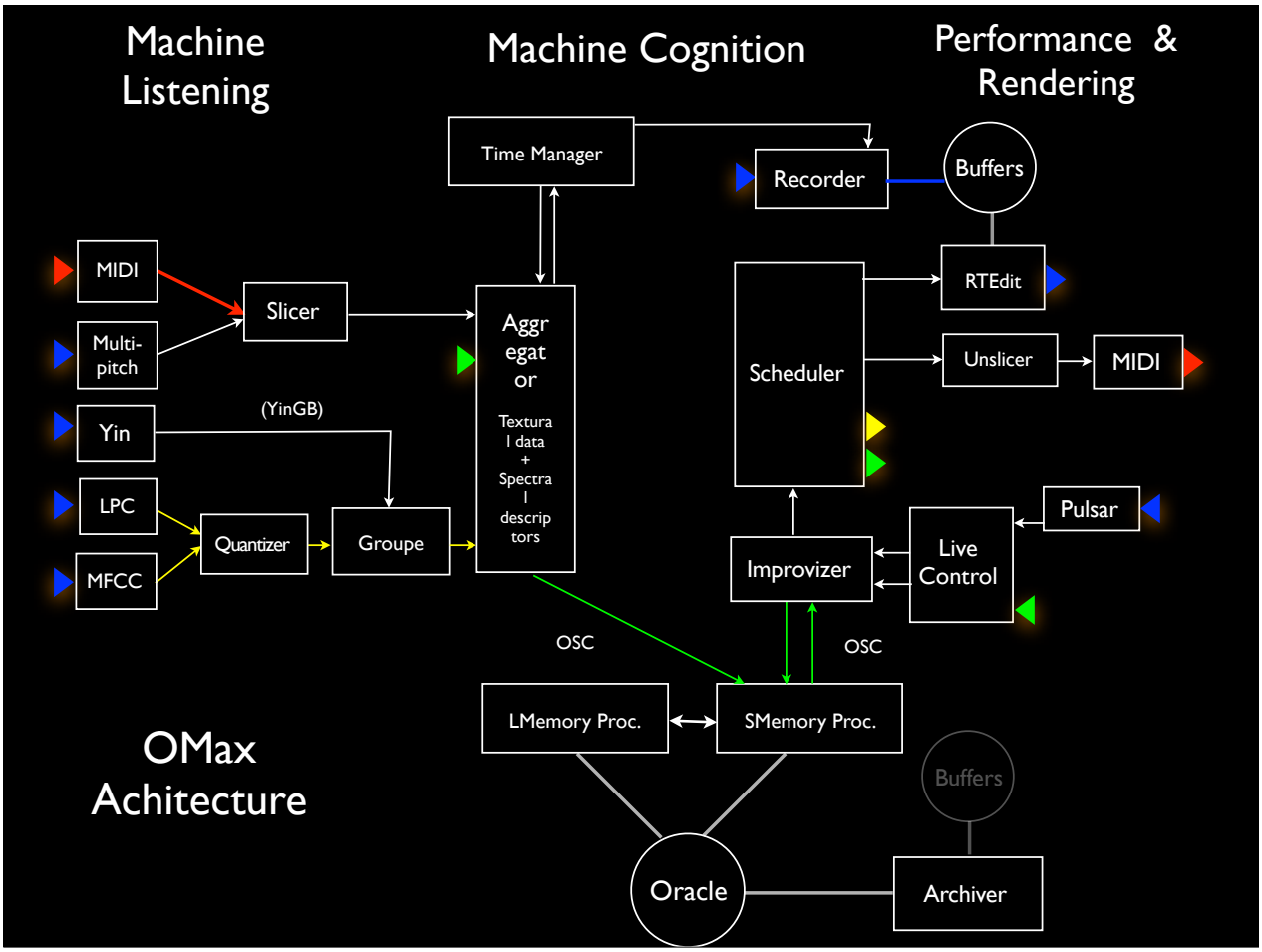
Ces transparents sont librement inspirés d'une présentation de Gérard Assayag, IRCAM

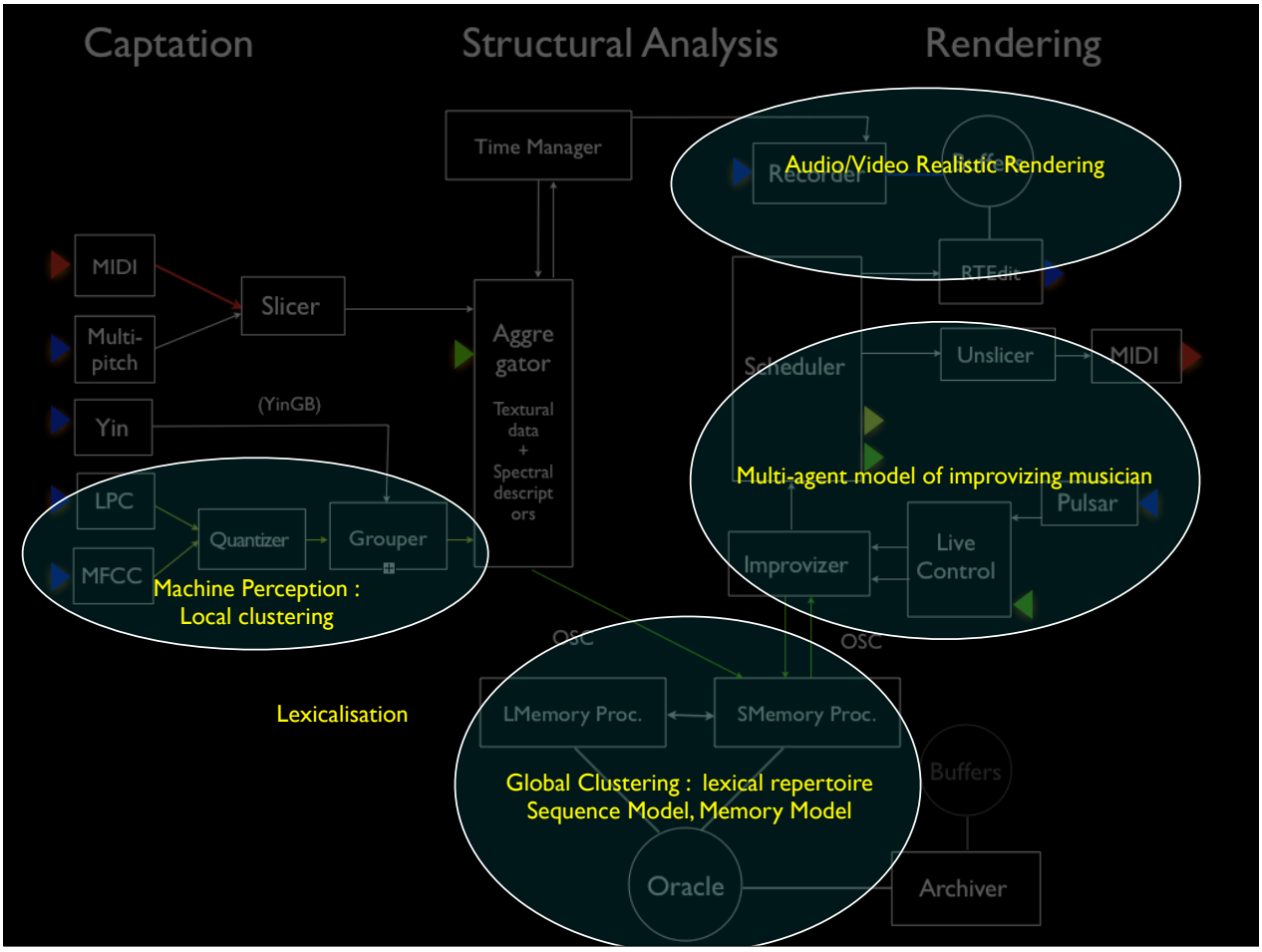
OMax

- Système d'improvisation musicale adapté à certaines musiques *live*
- Développé par G. Assayag, G. Bloch, M. Chemillier (IRCAM, Univ. Strasbourg, EHESS) depuis le début des années 2000
- Première version basée sur l'algorithme de Lempel-Ziv (compression de données, fichiers .zip)
- Version actuelle : oracle des facteurs

Improvisation stylistique

- Contexte : un musicien improvise en *live*
- Etapes :
  - captation du signal sonore en temps réel
  - segmentation du signal en données de haut niveau (notes, hauteurs, durées, rythmes...)
  - apprentissage du style sur ces données symboliques (oracle des facteurs)
  - re-synthèse d'un flux sonore à partir du style appris
- Résultat : un musicien virtuel jouant à la manière du musicien réel


Machine Listening
Machine Cognition
Performance & Rendering
Time Manager
Recorder
Buffers
MIDI
Multi-pitch
Slicer
Aggr egat or
Textura l data + Spectra l descrip tors
Scheduler
RTEdit
Unslicer
MIDI
(YinGB)
Yin
LPC
MFCC
Quantizer
Groupe
Pulsar
Improvizer
Live Control
OSC
OSC
LMemory Proc.
SMemory Proc.
OMax Achitecture
Buffers
Oracle
Archiver


Captation
Structural Analysis
Rendering
Time Manager
Audio/Video Realistic Rendering
MIDI
Multi-pitch
Slicer
Aggre gator
Textural data + Spectral descript ors
(YinGB)
Yin
Scheduler
Unslicer
MIDI
Multi-agent model of improvizing musician
LPC
Quantizer
Grouper
Pulsar
MFCC
Improvizer
Live Control
Machine Perception : Local clustering
OSC
OSC
Lexicalisation
LMemory Proc.
SMemory Proc.
Buffers
Global Clustering : lexical repertoire Sequence Model, Memory Model
Oracle
Archiver

Oracle des facteurs

- Données :
  - un alphabet de symboles (ex : notes de musique)
  - un mot (ex : sol ré si la si ré si ré sol)
- Automate reconnaissant l'ensemble des facteurs (sous-mots) d'un mot
  - dans l'exemple : sol / sol ré / sol ré si / ré / ré si / etc
- Avantage : simple à construire, économe en temps et en mémoire (linéaire)
- Défaut : reconnaît un peu trop de mots

Oracle des facteurs

Algorithme add_letter
Allauzen & al. SOFEN99

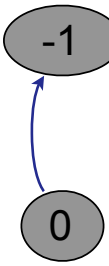
Fonction add_letter(Oracle($p = p_1 p_2 \dots p_m$), σ)

1. Create a new state $m + 1$
2. Create a new transition from m to $m + 1$ labeled by σ
3. $k \leftarrow S_p(m)$
4. **While** $k > -1$ and there is no transition from k by σ **Do**
5. Create a new transition from k to $m + 1$ by σ
6. $k \leftarrow S_p(k)$
7. **End While**
8. **If** $(k = -1)$ **Then** $s \leftarrow 0$
9. **Else** $s \leftarrow$ where leads the transition from k by σ.
10. $S_{p\sigma}(m + 1) \leftarrow s$
11. **Return** Oracle($p = p_1 p_2 \dots p_m \sigma$)

Figure4. Add a letter σ to Oracle($p = p_1 p_2 \dots p_m$) to get Oracle($p\sigma$)

Oracle des facteurs : initialisation

sol ré si la si ré si ré sol



Etat poubelle

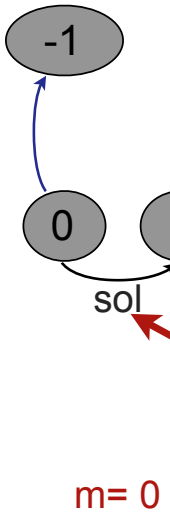
Oracle des facteurs : initialisation

sol ré si la si ré si ré sol

Etat poubelle

Oracle des facteurs : construction

sol ré si la si ré si ré sol



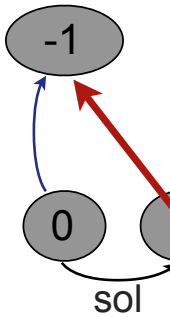
m= 0

Fonction add_letter(Oracle($p = p_1 p_2 \dots p_m$), σ)

1. Create a new state $m + 1$
2. Create a new transition from m to $m + 1$ labeled by σ

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 0

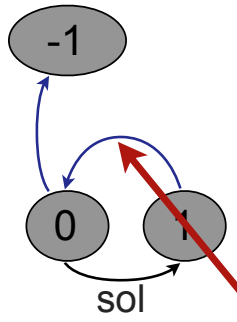
k= -1

3. $k \leftarrow S_p(m)$
4. **While** $k > -1$ and there is no transition from k by σ **Do**
5. Create a new transition from k to $m + 1$ by σ
6. $k \leftarrow S_p(k)$
7. **End While**

Faux

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 0

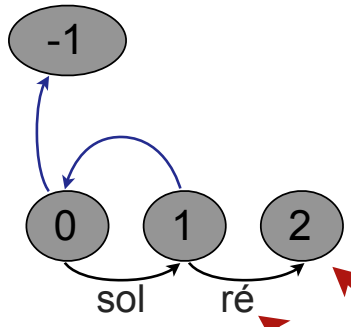
k= -1

s= 0

8. **If** $(k = -1)$ **Then** $s \leftarrow 0$
9. **Else** $s \leftarrow$ where leads the transition from k by σ.
10. $S_{p\sigma}(m+1) \leftarrow s$

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 1

k= -1

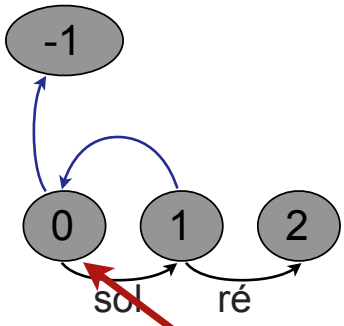
s= 0

Fonction add_letter(Oracle$(p = p_1 p_2 \dots p_m)$, σ)

1. Create a new state $m+1$
2. Create a new transition from m to $m+1$ labeled by σ

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 1

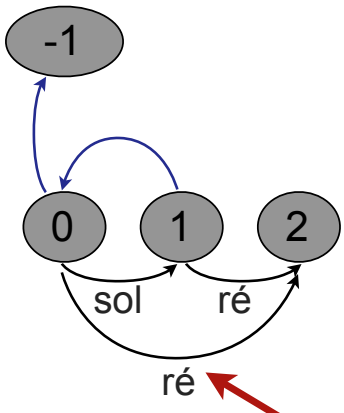
k= 0

s= 0

3. $k \leftarrow S_p(m)$
4. **While** $k > -1$ and there is no transition from k by σ **Do**
5. Create a new transition from k to $m+1$ by σ
6. $k \leftarrow S_p(k)$
7. **End While**

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 1

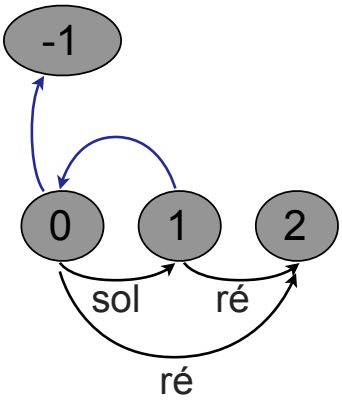
k= 0

s= 0

3. $k \leftarrow S_p(m)$
4. **While** $k > -1$ and there is no transition from k by σ **Do**
5. Create a new transition from k to $m+1$ by σ
6. $k \leftarrow S_p(k)$
7. **End While**

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 1

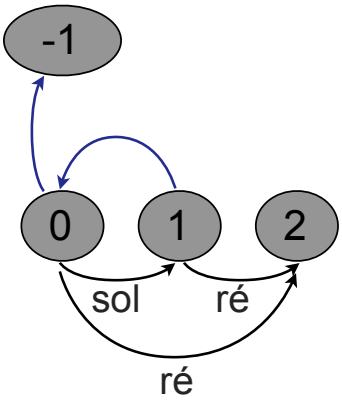
k= -1

s= 0

3. $k \leftarrow S_p(m)$
4. **While** $k > -1$ and there is no transition from k by σ **Do**
5. Create a new transition from k to $m+1$ by σ
6. $k \leftarrow S_p(k)$
7. **End While**

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 1

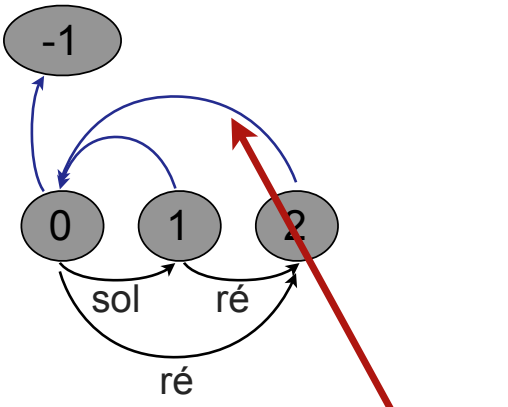
k= -1

s= 0

8. **If** $(k = -1)$ **Then** $s \leftarrow 0$
9. **Else** $s \leftarrow$ where leads the transition from k by σ.
10. $S_{p\sigma}(m+1) \leftarrow s$

Oracle des facteurs : construction

sol ré si la si ré si ré sol



m= 1

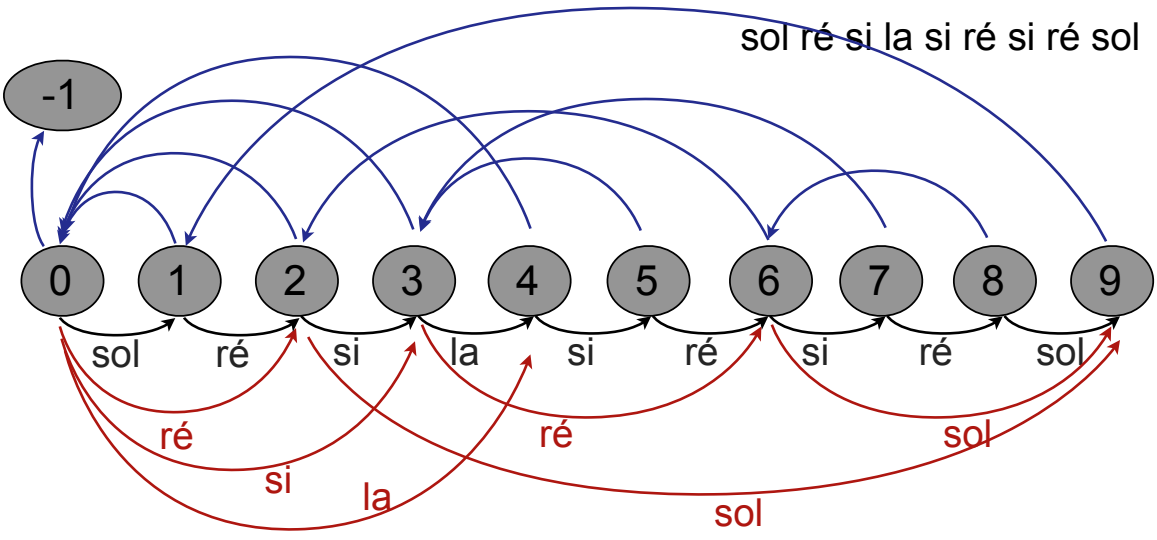
k= -1

s= 0

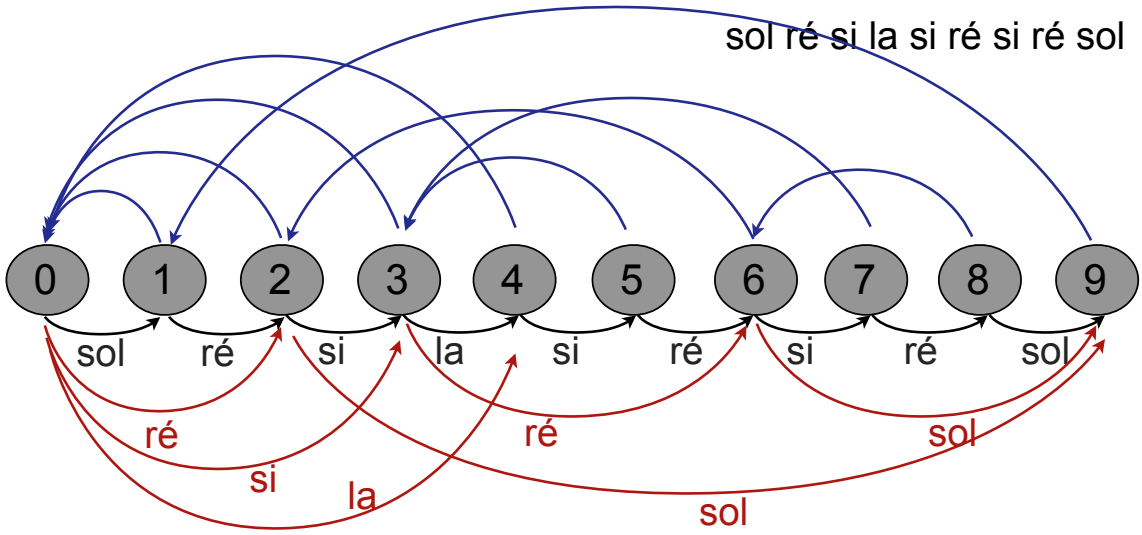
8. **If** $(k = -1)$ **Then** $s \leftarrow 0$
9. **Else** $s \leftarrow$ where leads the transition from k by σ.
10. $S_{p\sigma}(m+1) \leftarrow s$

Oracle des facteurs : construction

sol ré si la si ré si ré sol

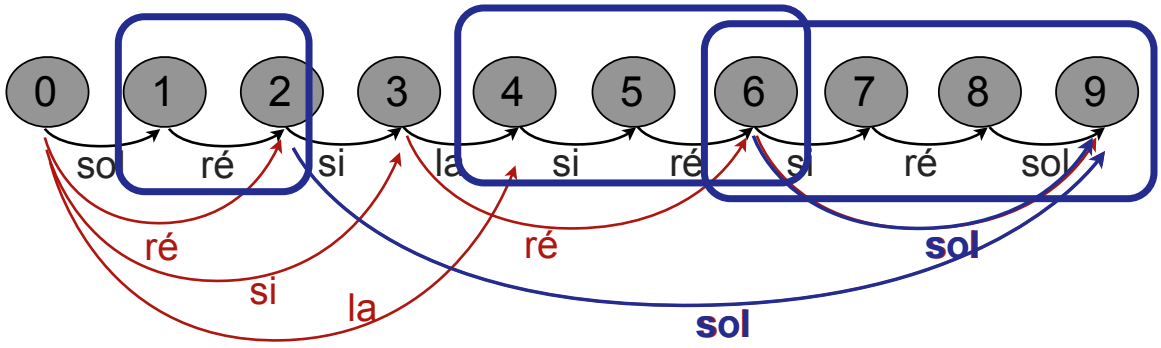


Oracle des facteurs : construction



Parcours / synthèse

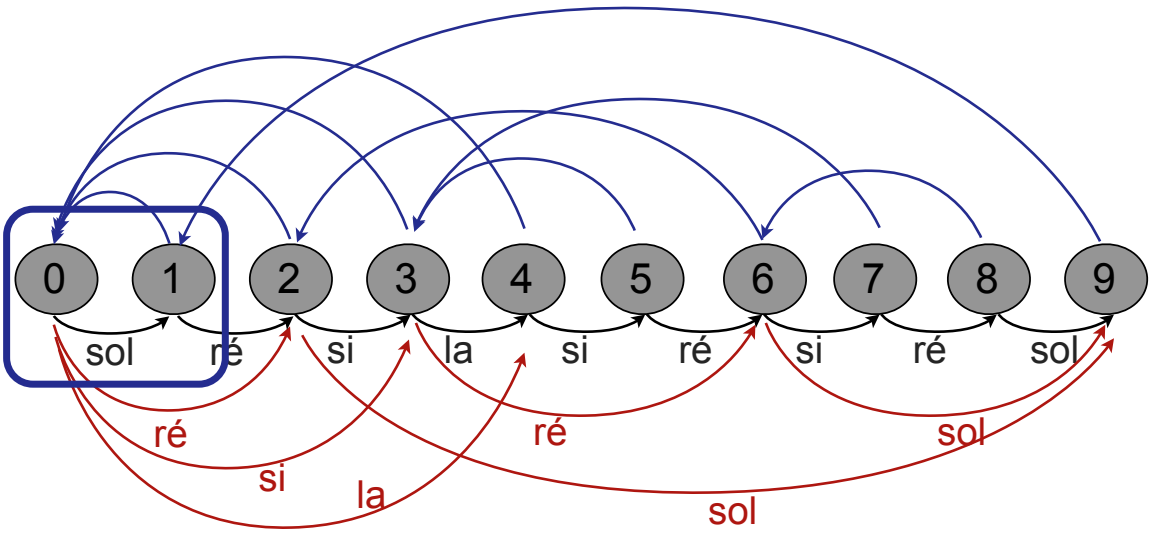
Propriété des transitions rouges : relient des points partageant les mêmes préfixes / contextes



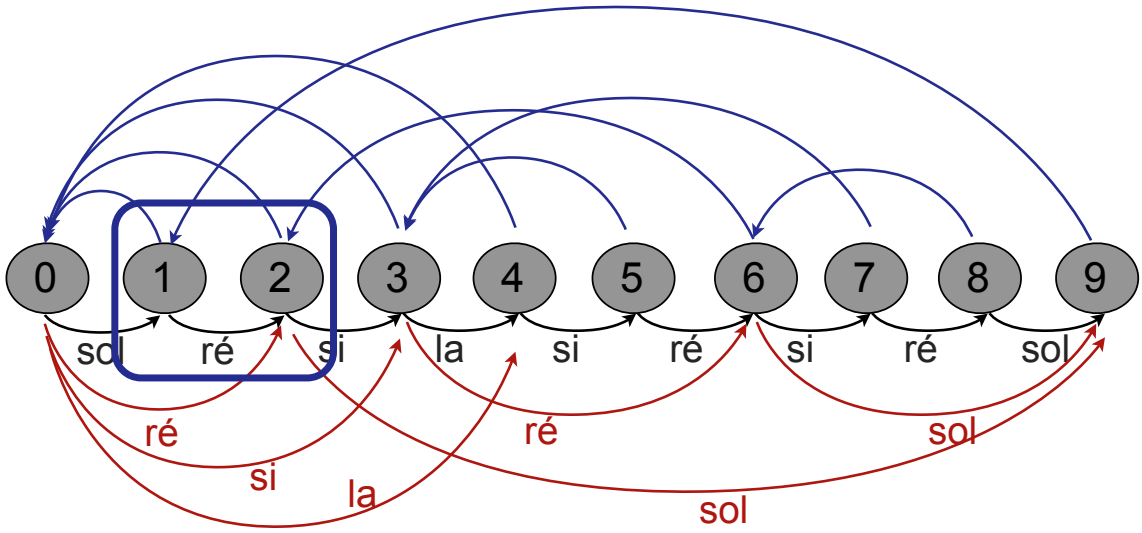
Parcours / synthèse

Stratégie de navigation quand on veut changer de contexte à partir du point courant

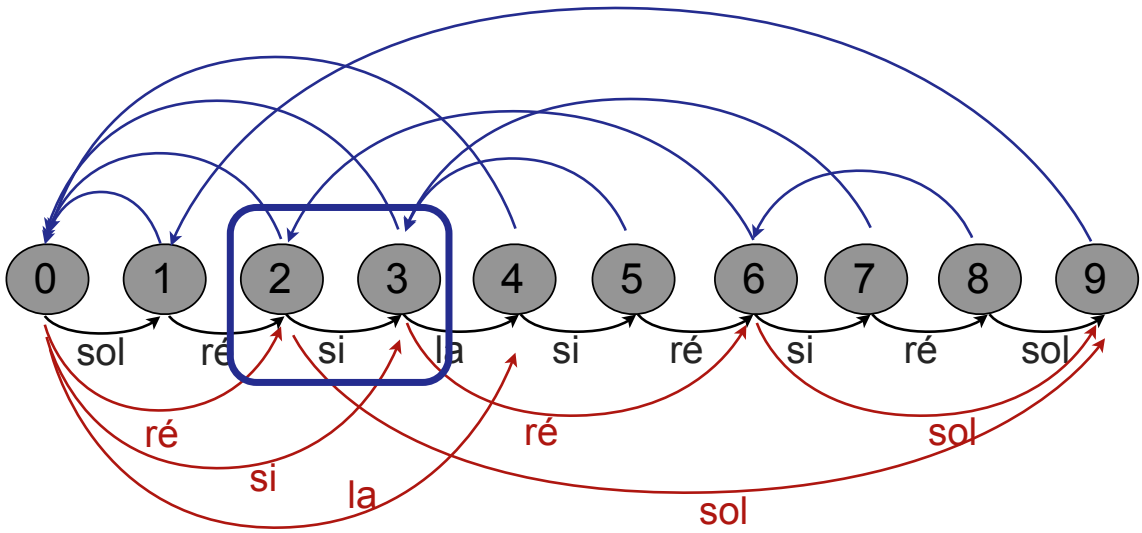
- on explore les futurs possibles (transitions)
- pour chaque état du futur, on explore les liens bleus
- on choisit un point d'arrivée selon une distribution probabiliste favorisant le contexte et la similarité rythmique globale
- on saute au point d'arrivée
- on place ce point sur une liste interdite (pendant un certain temps)



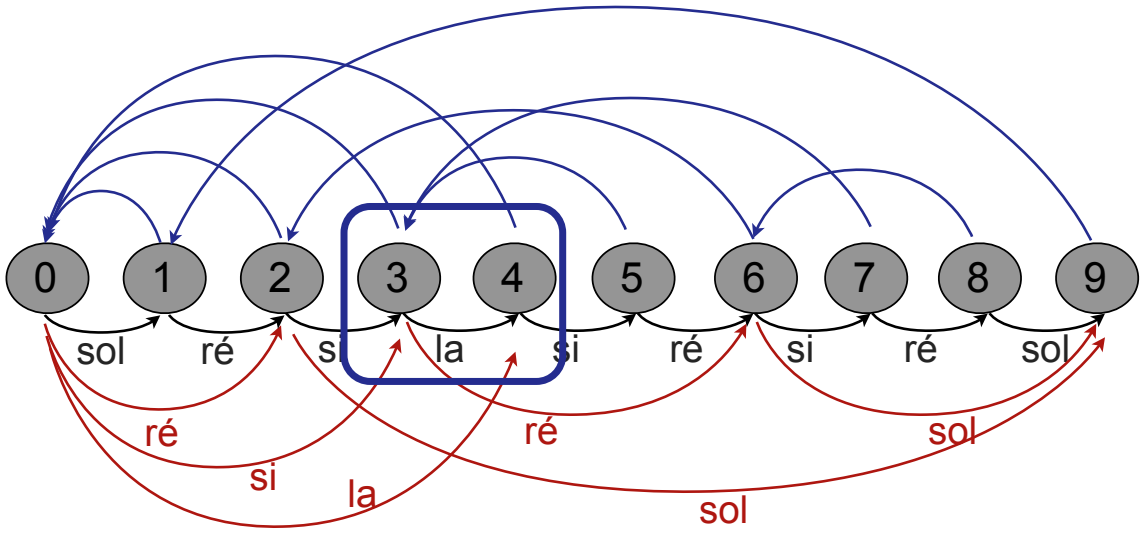
sol



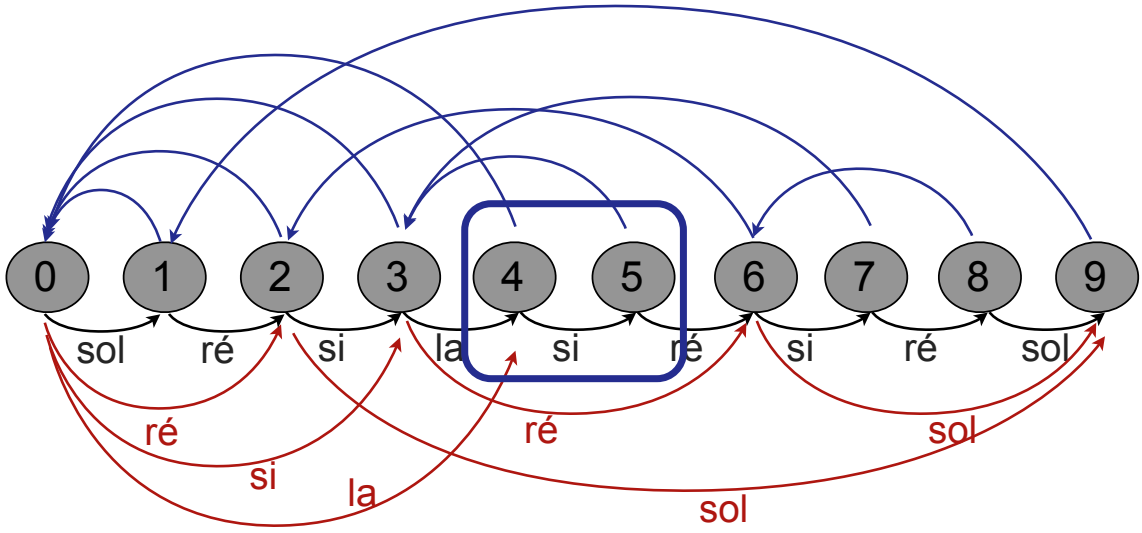
sol ré



sol ré si

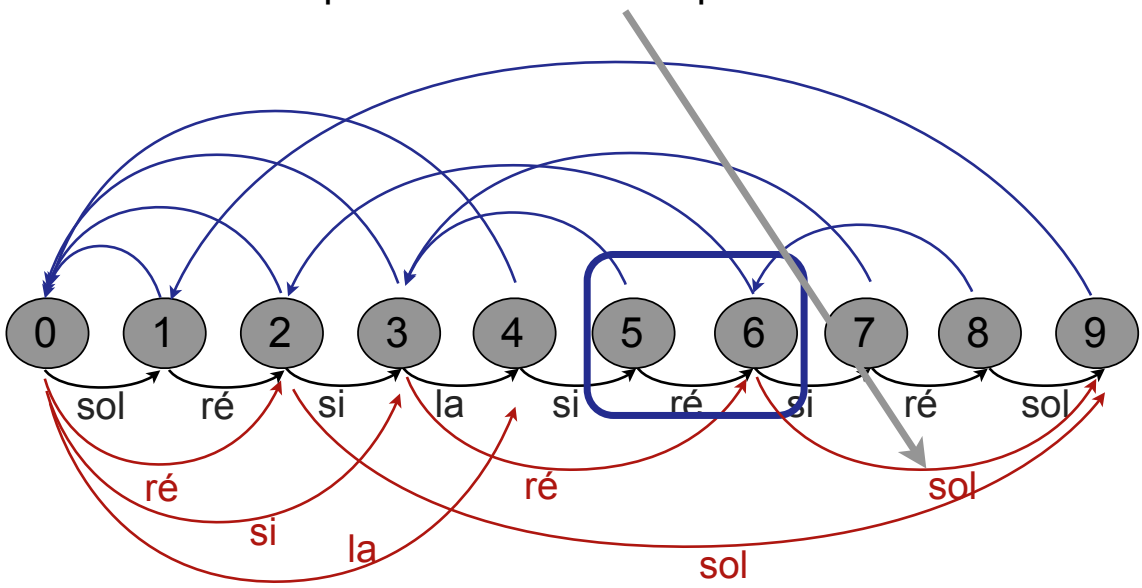


sol ré si la



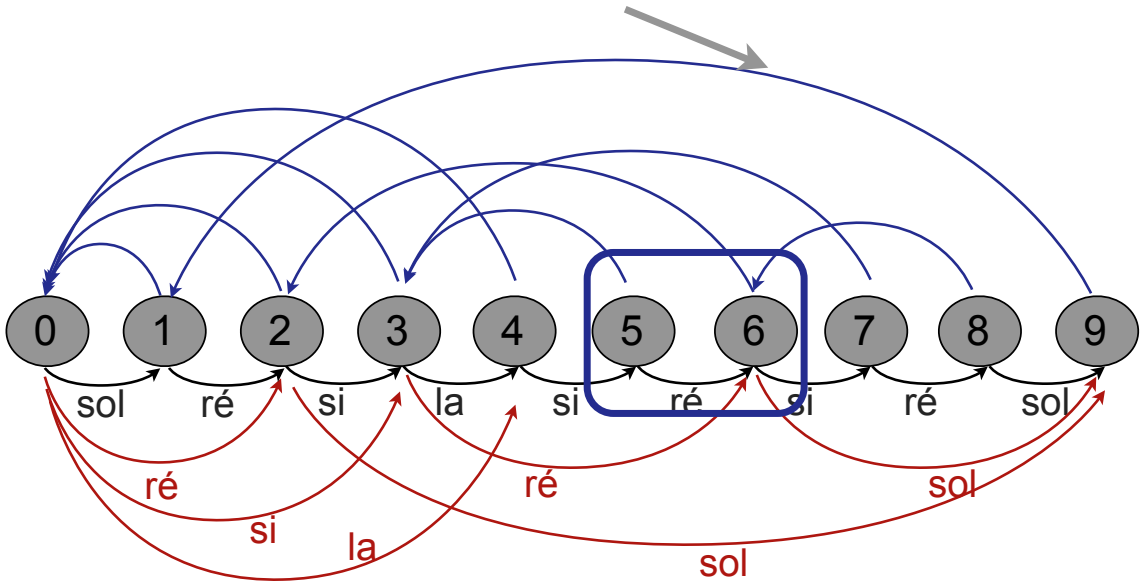
sol ré si la si

Envie de changer de contexte !
On explore les transitions possibles



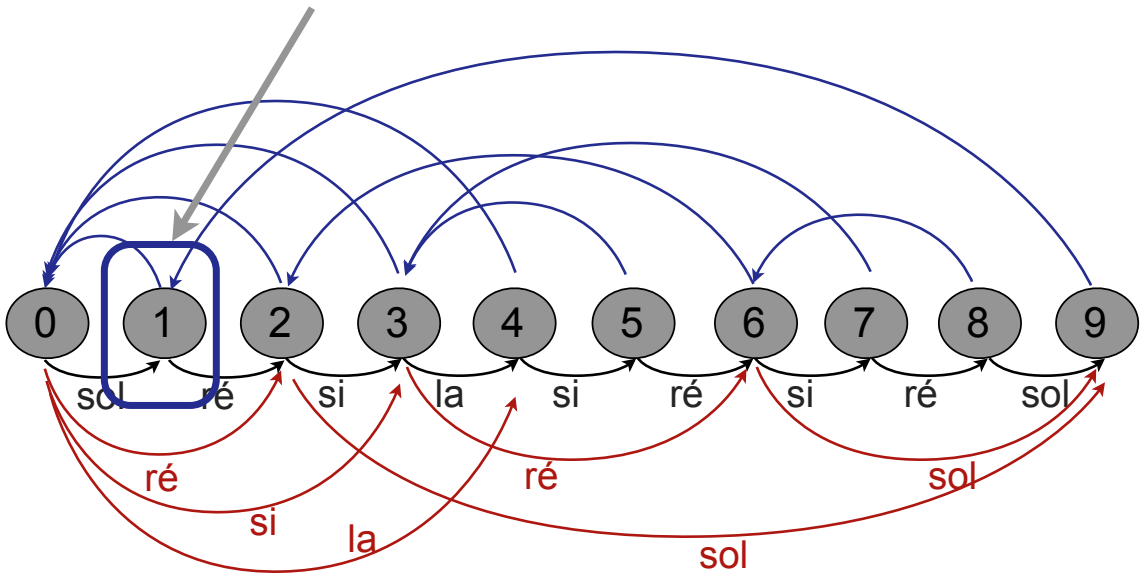
sol ré si la si ré

Envie de changer de contexte !
On suite la flèche bleue



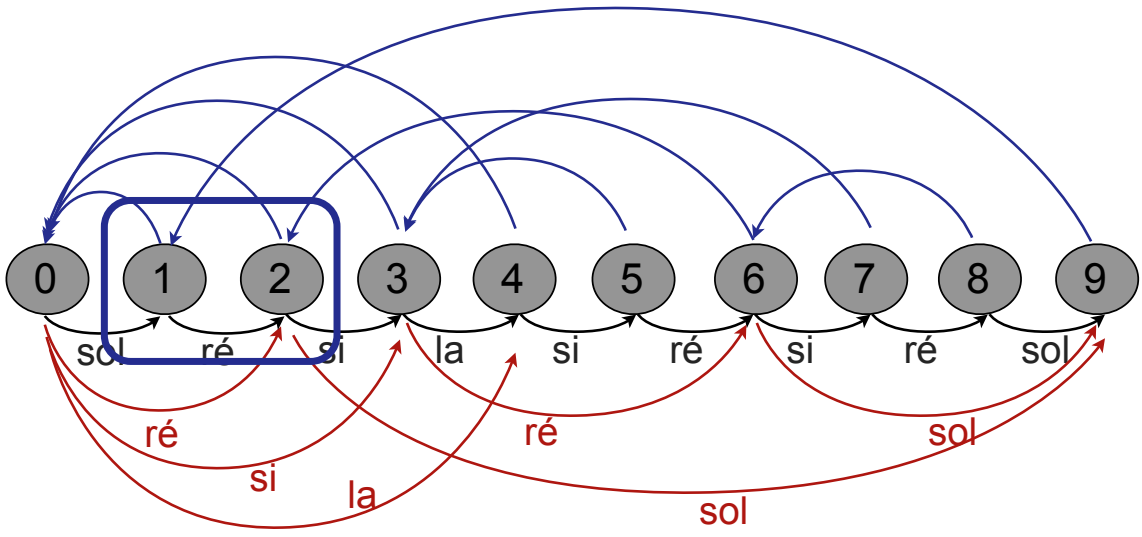
sol ré si la si ré

Envie de changer de contexte !
On saute



sol ré si la si ré sol

On reprend normalement



sol ré si la si ré sol ré ...

Oracle des facteurs

Intérêt musical

- Quand on modifie le déroulement normal de la séquence, on revient à un endroit dont le contexte (préfixe) est similaire
- En probabilisant les transitions, on obtient des séquences proches, mais toujours nouvelles
- On peut contrôler dans une certaine mesure les transitions et les adapter sur des critères musicaux (rythmes, tonalité...)

Oracle des facteurs : exemples

Bernard Lubat, atelier OMax, 2004

Lubat joue le piano, OMax joue les percussions à la manière de Lubat

Oracle des facteurs : exemples

Lubax, concert à New York, 2012 (ImproTech workshop)

Oracle des facteurs : exemples

Conclusion

- Ressources associées à ce cours :
  - Cours de Julius O. Smith, Stanford University, très complet sur le Digital Signal Processing (audio)

    https://ccrma.stanford.edu/~jos/
  - Page de OMax, articles, algorithmes et démos

    http://repmus.ircam.fr/omax/home
- Bonus :
  - synthèse temps réel de bruitages pour jeux vidéo

    http://obiwannabe.co.uk/html/sound-design/sound-design-audio.html
- Beaucoup de problèmes liés au son numérique sont de l'ordre de la recherche : caractérisation psychoacoustique, synthèse réaliste et peu coûteuse, génération ou arrangement automatique de musique, etc