



# Conversion formule

Transformé binaire en cumulatif

## Conversion cause

Le respect de la théorie musicale classique est une bonne raison pour que sa forme soit gardée, ce n'est pas parce que les musiciens l'ont adopté mais pour la logique du développement gammique entrepris.

### Visiter la méthode classique

Elle commence par la gamme naturelle nommée majeure à la tonalité en Do, et elle est notre étalonnage général car il est également usité dans la quantique musicale. Ainsi les sept degrés modaux sont **tous d'origine naturelle**, ceci qu'à chaque formulation diatonique est citée son origine.

	1	2	3	4	5	6	7	
								<b>Gamme majeure naturelle</b>
<b>1</b>	2	-3	4	5	6	-7		Gamme mélodique 7 <sup>min</sup> naturelle
<b>1</b>	-2	-3	4	5	-6	-7		Gamme tonale mélo harmonique 7 <sup>min</sup> naturelle
<b>1</b>	2	3	+4	5	6	7		<b>Gamme quarte augmentée naturelle</b>
<b>1</b>	2	3	4	5	6	-7		Gamme septième mineure 7 <sup>min</sup> naturelle
<b>1</b>	2	-3	4	5	-6	-7		Gamme mélodique harmonique 7 <sup>min</sup> naturelle
<b>1</b>	-2	-3	4	-5	-6	-7		Gam tone mélo harmo dominante 7 <sup>min</sup> naturelle

Légende : Normalement l'origine est facultative (7<sup>min</sup> = 7<sup>ème</sup> mineure)

En alignant toutes les septièmes en 7<sup>ème</sup> majeures, on obtient quelques modulations.

	1	2	3	4	5	6	7	
								<b>Gamme majeure</b>
<b>1</b>	2	-3	4	5	6	7		Gamme mélodique
<b>1</b>	-2	-3	4	5	-6	7		Gamme tonale mélo harmo
<b>1</b>	2	3	+4	5	6	7		<b>Gamme quarte augmentée</b>
<b>1</b>	2	3	4	5	6	7		Gamme majeure
<b>1</b>	2	-3	4	5	-6	7		Gamme mélo harmo
<b>1</b>	-2	-3	4	-5	-6	7		Gamme tonale mélo harmo dominante

Légende : Modulation de septième majeure (7<sup>ème</sup> majeure)

Cette technique développe les modulations des gammes tout en ayant des similitudes naturelles, on y trouve deux gammes majeures (Tonique. Dominante) et de plus deux modes n'ont pas varié (Tonique. Médiane).

Cette démarche semble aller de l'avant en gardant des anciens repères, par exemple la synchronisation au mode « maj7 » démontre la possible mécanisation des gammes musicales.



## Application d'un premier état

Il est vrai qu'en arpentant les modulations gammales on s'aperçoit qu'elles sont hiérarchisées, en commençant simplement par une note altérée à la fois on induit sept modulations par signe d'altération.

Les dièses « # » et les bémols « b », sont des altérations.

1	2	3	4	5	6	7	Gamme majeure
1	-2	3	4	5	6	7	Gamme tonale
1	2	-3	4	5	6	7	Gamme mélodique
1	2	3	-4	5	6	7	Gamme médiane
1	2	3	4	-5	6	7	Gamme dominante
1	2	3	4	5	-6	7	Gamme harmonique
1	2	3	4	5	6	-7	Gamme sensible

Légende : La gamme sensible est en septième mineure ( $7^{min}$ )

La seule restriction à retenir dans ce genre de développement, c'est qu'il faut éviter de reproduire le cas diatonique déjà développé. Pour y parvenir à chaque occurrence, il faut réaliser un développement diatonique, puisqu'à chaque mode diatonique correspond à une formulation différente.

Il peut y avoir des litiges à la nomination des gammes, en prenant un simple exemple :

Dans la gamme mélodique harmonique « **b3 et b6** », il y a la modulation dominante augmenté « **#5** ». Quel serait votre choix pour nommer cette gamme ?

- « **b3 et b6** » : A deux termes déjà existants séparément.
- « **#5** » : Est l'autre sélection modale au poids le plus léger.

L'opération effectuée sur les altérations produit un résultat réel allant du positif au négatif, et la gamme naturelle « notre étalon » après l'opération donne zéro comme poids altératif.

Cette situation présente un choix présélectionné, ou un point de divergence comme valeur de transition diatonique. Quand on opte vers le classicisme conservateur la base de la gamme est « **b3 et b6** », mais en choisissant selon le poids modal le plus proche de zéro on prend le produit « **#5** ».

C'est exactement ce qu'il s'est passé avec la programmation de `progamv6encore`, les modèles diatoniques suivent la règle de la conservation classique. Sauf qu'en évoluant parmi les formes, on a analysé le système altératif nous menant directement à des nombres réels entiers. Ce faisant de la gamme naturelle ayant un poids nul dans sa forme tonique, soit que le premier degré de la gamme c'est Do aux (zéro-poids et majeur).

La gamme dominante augmentée a une forme particulièrement adaptée pour joindre deux modèles diatoniques différents, dans le même genre on aurait une géolocalisation modale inter-diatoniques.

À des fins stratégiques le code source a été renommé `Glob Gam Vers 6` (tout lié).



## Application Glob Gam Vers 6

Cette première application se présente comme une interface utilisateur dont le but est de modéliser les gammes, afin de séparer les modes fondamentaux et toniques des autres modes du contenu diatonique. À cette époque les gammes étaient représentées dans deux listes ; une pour les noms et la deuxième pour les formules cumulatives.

### Le mode à forme cumulative

On ne fait pas de programmation sans algorithme, c'est pour cela qu'on transforme les données reçues de manière à ce qu'elles deviennent opérationnelles.

#### Définitions des formules

1. [C, D, E, F, G, A, B] = Formule alphabétique artificielle.
  - a. Permet à la note d'être visiblement compréhensible
2. [1, 2, 3, 4, 5, 6, 7] = Forme à valeur numérique.
  - a. Quand on parle des tonalités des degrés modaux
3. ['101011010101'] = Forme géolocale binaire.
  - a. Produit une neutralité à valeur locale et basique
4. [1, 1, 0, 1, 1, 1, 0] = Forme à degré cumulatif.
  - a. À chaque chiffre correspond un nombre de zéros

#### Définition du mode diatonique naturellement majeur

- Liste = {'0' : ['101011010101', [1, 1, 0, 1, 1, 1, 0]]} # Dictionnaire
- Liste = {'0' : [1, 1, 0, 1, 1, 1, 0]} # Clé(Nom), valeur(mode cumulatif)

## Application Global Applis

Cette application s'est développée sans l'évolution des modes cumulatifs, elle occupait un exercice binaire et de nombreuses autres choses. Il fallait écrire une nouvelle fonction dans le code, le premier programme traite les modes cumulatifs, le second programme fera la conversion du mode binaire vers le mode cumulatif.

Liste des noms exemplaire, sous la forme d'un dictionnaire du code source.

```
Génome = ['0', '-2', '+2', '^2', '-3', '-23', '-34x', '+34', '+23x', '-34', 'x3', '°3', '+34x', '°34x',
'^3', '-4', '-24', '^4', '°4', '-5', '-25', '-25+', '+25-', '-35', '-35+', '+45x', '+25x',
'°35-', '+35x', '-45+', '-45', 'x5', 'x45+', '-25°', '-35°', '-45°', '°45-', '°5', '°35+', '*5',
'°35x', '-45x', '°45x', '-6', '+6', '-26', '-26+', '+26-', '+26', '-36', '-36+', '-56', '-56+',
'+56', 'x46+', '-26°', '-46+', '-46°', 'x36+', '-56°', '°46-', '°36+', '*6', '°46+', '°6', 'x26-']
```

La liste des cumulatives est ignorée, elle n'est pas nécessaire à la conversion.



## Portion de script algorithmique

	<code>pratique</code>	<i>Dictionnaire : Clé(nom gamme) Valeur(forme binaire)</i>
1	<code>for k_pratic in pratique.keys():</code>	<i>k_pratic contient nom de la gamme</i>
	<code>    k0 = 0</code>	<i>Cumulateur des zéros</i>
	<code>    kn = -1</code>	<i>Nombre cyclique</i>
	<code>    k_cumul = []</code>	<i>Support forme cumulative</i>
	<code>    k_formel = pratique[k_pratic]</code>	<i>Contient forme binaire</i>
2	<code>    for k_cas in k_formel:</code>	<i>Lecture forme binaire</i>
	<code>        kn += 1</code>	
	<code>        if k_cas == '1':</code>	<i>Lecture d'un 1</i>
	<code>            if kn &lt; len(k_formel)-1 and k_formel[kn+1] == '1':</code>	<i>Alors k_cas = '1'</i>
	<code>                k_cumul.append(0)</code>	<i>Valeur cumul intervalle</i>
	<code>                continue</code>	<i>Annuler else du cycle 2</i>
	<code>            elif len(k_cumul)== 6:</code>	<i>Quand longueur k_cumul = 6</i>
	<code>                k_cumul.append(0)</code>	<i>Écriture du dernier zéro</i>
	<code>            else:</code>	<i>Alors k_formel[kn+1] == '0'</i>
3	<code>                for x in range(kn+1, len(k_formel)):</code>	<i>Lecture des zéros de k_formel</i>
	<code>                    if k_formel[x] == '0':</code>	
	<code>                        k0 += 1</code>	<i>Compter les zéros</i>
	<code>                    if k_formel[x] == '1':</code>	<i>Fin de la suite des zéros</i>
	<code>                        k_cumul.append(k0)</code>	<i>Écriture nombre de zéros</i>
	<code>                        k0 = 0</code>	
	<code>                        break</code>	<i>Terminer lecture du cycle 3</i>
		<i>Continuer le cycle 2</i>
	<code>    pratique[k_pratic] = k_cumul</code>	<i>Écriture forme cumulative</i>
		<i>Fin de partie</i>

Et voilà c'est tout