

# LE VERROU MIROIR

**Une contrainte algorithmique globale dans le Coran**

**Auteur :** Soufiane BAGHOR

**Date :** Janvier 2026

**Version :** Draft 1.0

---

## Abstract

Ce livre démontre qu'une contrainte algorithmique globale façonne la structure du Coran sous une mise en page particulière dite M11×N.

Lorsque le texte est découpé en pages de 11 lignes, on observe un phénomène simple et précis :

si l'on regarde uniquement la première lettre de chaque ligne, la première et la dernière ligne commencent par la même lettre, la deuxième et l'avant-dernière aussi, la troisième et la neuvième aussi, et ainsi de suite, la ligne centrale jouant le rôle d'axe.

Autrement dit, chaque page satisfait une symétrie miroir sur les lettres initiales de ses lignes.

Cette propriété est vérifiée sur l'ensemble du livre, qui s'étend sur environ 891 pages sous cette mise en forme. Elle n'apparaît dans aucun autre corpus testé et disparaît immédiatement dès que l'on mélange l'ordre global du texte : un simple shuffle provoque l'effondrement de la structure en quelques pages seulement.

L'espace des configurations possibles étant astronomique, l'existence d'un texte réel satisfaisant une telle contrainte globale constitue un événement de probabilité extrêmement faible sous l'hypothèse H0.

Nous montrons que ce système est vérifiable mais non constructible par des règles locales simples, et qu'il se comporte comme un objet globalement verrouillé : il ne peut ni être produit ni être maintenu par des ajustements progressifs ou locaux.

Ce travail met ainsi en évidence une nouvelle classe d'objets informationnels : des textes soumis à une contrainte globale distribuée, dont la cohérence ne peut exister que comme un tout.

---

أَوْرَثْنَا نِي السَّمَاءَ وَلَنْ تُؤْمِنَ لِرُفِيقِ حَمَّةٍ تُنَزَّلُ

يَا أَيُّهُمْ لَا يَشْرِكُ بِنَاهٍ إِلَّا مَنْ كَانَ فِي أَضَالَّةٍ

وَمَا مَنَعَ النَّاسَ أَنْ يُؤْمِنُوا إِذْ جَاءَهُمُ الْهُدَىٰ إِلَّا أَنْ

**ابَعَثَ اللَّهُ بَشَرًا رَسُولًا فَلْ لَوْكَانَ فِي الْأَرْضِ مَلِئِكًا**

**أَرَامْ بِرَانِيْه بِنْ سَلَمَة دِيمْ بِرَانِيْه اُذْنَسَانْ فَوَشَّرَا سَبِيْرَه بِكَرَه**

شَهِيدًا بِئْنِي وَبِئْنِكُمْ إِنَّهُ وَكَانَ يَعْمَادُهُ خَيْرٌ أَبْصَرَ  
كَاهْ سَانْ ، مَلِكُوكْ شَفَعْيَ التِّرَااثُ الْقَرْنَيِّ وَالْكَتَابِيِّ ، بَيْنَ

أَنْ يَهْدِي اللَّهُ فَهُوَ الْمُهْدِيٌ وَمَنْ يُضْلِلُ فَنَّلَّ بِهِ

**وَلِلَّٰهِ مَنْ دُوِيَهُ وَلَكُشْرُهُمْ بِوَمَّا فَيْتَهُ عَلٰٓيْ وَجْهُهُمْ عَمِيْ**

صَمَّا وَهُمْ جَهَنَّمَ كُلَّنَا خَبَثٌ زِدْنَاهُمْ سَعْيًا لَكَجَزَأُهُمْ بِمَا

ل و م

شہزادہ

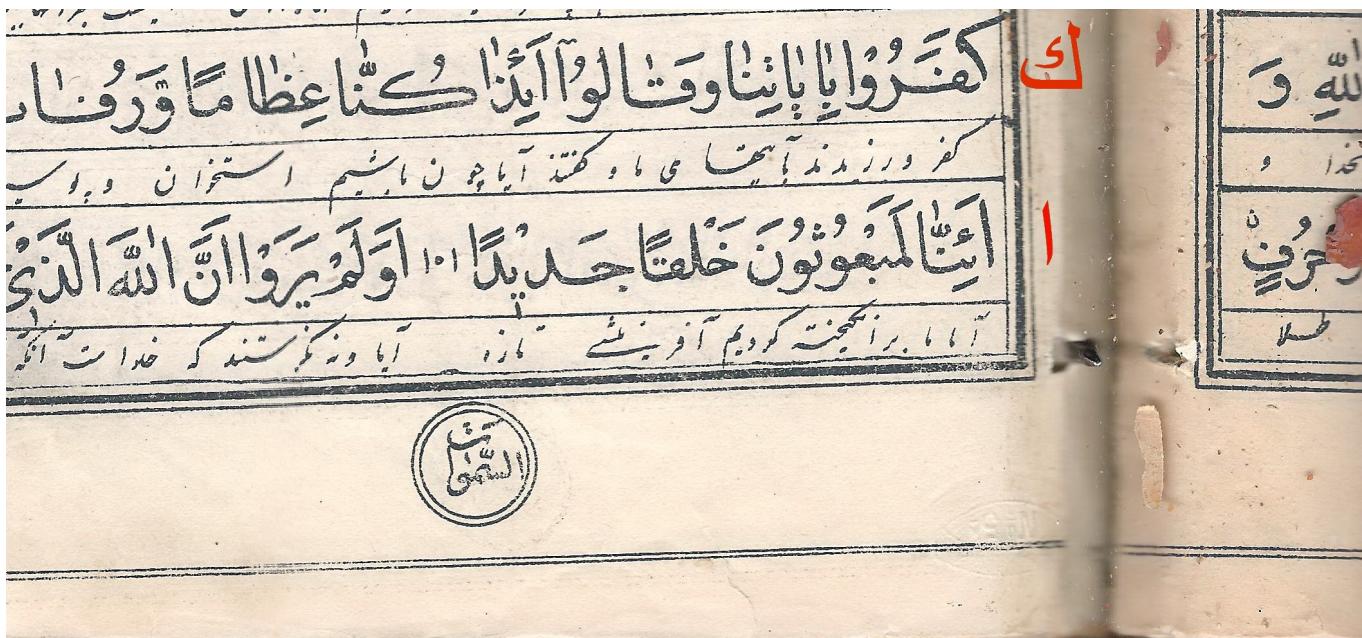


Figure 1 — Exemple réel d'une page du texte vérifiant la contrainte M11.

Les lettres initiales des lignes forment une signature strictement palindromique autour de l'axe central.

Cette page est extraite d'un mushaf réel. Elle n'est ni construite, ni optimisée, ni arrangée pour l'expérience.

Elle est **lisible, calligraphiquement naturelle**, et pourtant **structurellement contrainte**.

Le présent ouvrage n'étudie pas une construction abstraite, mais un **phénomène réellement présent dans le texte**.

L'Annexe A propose une analyse détaillée et instrumentée de ce type de page, ainsi que de son interprétation algorithmique.

## Sommaire

- [Introduction générale](#)
- [Chapitre 1 — Le problème M11 : une contrainte globale sur un texte linéaire](#)
  - [1.1. L'anomalie structurelle](#)
  - [1.2. Formalisation de la grille de lecture \(le modèle M11\)](#)
  - [1.3. Globalité vs localité : le concept de verrou](#)
  - [1.4. L'hypothèse H0 et l'argument probabiliste](#)
- [Chapitre 2 — Formalisation complète du système](#)
  - [2.1. Représentation du texte comme flux discret de mots](#)
  - [2.2. Grille M11×N : découpage interne d'une page](#)

- [2.3. Définition de la signature d'une page](#)
- [2.4. Définition formelle de la contrainte miroir \(M11\)](#)
- [2.5. Définition de l'espace des découpages internes](#)
- [2.6. Définition de  \$d\(i\)\$ : nombre de sorties possibles depuis une position](#)
- [2.7. Définition d'un “dead-end” \(point mort\)](#)
- [2.8. Définition du parcours global et de la continuité](#)
- [2.9. Définition du critère global M](#)
- [Chapitre 3 — Le vrai problème : un graphe de transitions, pas une combinatoire locale](#)
  - [3.1. Le piège du “grand nombre” de découpages internes](#)
  - [3.2. Projection brutale de la combinatoire sur un espace minuscule](#)
  - [3.3. Le vrai objet mathématique : un graphe de transitions](#)
  - [3.4. Nature topologique du problème](#)
  - [3.5. Reformulation finale du problème M11](#)
- [Chapitre 4 — Analyse probabiliste locale : pourquoi le motif M11 n'est pas dû au hasard](#)
  - [4.1. Principe de l'analyse](#)
  - [4.2. Modèle probabiliste simplifié](#)
  - [4.3. Observation sur un exemple réel](#)
  - [4.4. Passage du local au global](#)
  - [4.5. Ce que cette estimation établit — et ce qu'elle n'établit pas](#)
  - [4.6. Interprétation des ordres de grandeur](#)
  - [4.7. Différence entre construction volontaire et apparition aléatoire](#)
  - [4.8. Verdict strictement probabiliste](#)
  - [4.9. Limite du raisonnement purement probabiliste et nécessité d'un test structurel](#)
- [Chapitre 5 — Tests de destruction de la structure \(Null Models\)](#)
  - [5.1. Principe général](#)
  - [5.2. Métrique observée](#)
  - [5.3. Types de Null Models testés](#)
  - [5.4. Résultat central](#)
- [Chapitre 6 — Attaques structurelles : pourquoi le système ne survit pas à la permutation](#)
  - [6.1. Principe de l'attaque : détruire l'ordre sans toucher à la matière](#)
  - [6.2. Pourquoi ce test est décisif](#)
  - [6.3. Procédure expérimentale](#)
  - [6.4. Résultat observé : effondrement quasi immédiat](#)
  - [6.5. Interprétation et conclusion structurelle](#)
  - [6.6. Pourquoi un tel effondrement est inévitable](#)
  - [6.7. Différence de nature entre réussite locale et stabilité globale](#)
  - [6.8. Mesure expérimentale de la longueur de survie L](#)

- [6.9. Résultat empirique observé sur les shuffles](#)
- [6.10. Interprétation : longueur de cohérence structurelle](#)
- [6.11. Pourquoi quelques shuffles peuvent survivre temporairement](#)
- [6.12. Résultat structurel](#)
- [Chapitre 7 — Étranglement structurel et dynamique de l'échec](#)
  - [7.1. Observation qualitative](#)
  - [7.2. Comparaison avec le texte original](#)
  - [7.3. Analyse de la réduction de l'espace des possibles](#)
  - [7.4. Interprétation : cohérence longue portée](#)
  - [7.5. Conclusion : le système comme architecture monolithique](#)
- [Chapitre 8 — Verdict structurel et impossibilité de reconfiguration](#)
  - [8.1. Récapitulatif expérimental](#)
  - [8.2. Conclusion structurelle](#)
  - [8.3. Interprétation informationnelle](#)
  - [8.4. Problème fondamental : construire vs reconfigurer](#)
  - [8.5. Impossibilité de reconfiguration a posteriori](#)
- [Chapitre 9 — Impossibilité de factorisation locale : le verrou structurel](#)
  - [9.1. Limites d'une construction locale et artisanale](#)
  - [9.2. Le passage du local au global](#)
  - [9.3. Explosion combinatoire des contraintes couplées](#)
  - [9.4. Instabilité expérimentale et effondrement de la structure](#)
  - [9.5. Conclusion : le texte comme système verrouillé](#)
- [Chapitre 10 — Reconstruction algorithmique et verrouillage par gabarit](#)
  - [10.1. Du texte brut à l'objet paginé](#)
  - [10.2. Normalisation et espace de recherche](#)
  - [10.3. Ancrage par page template](#)
  - [10.4. Le piège fondamental : infinité de paginations locales](#)
  - [10.5. Apprentissage du gabarit](#)
  - [10.6. Formalisation de la contrainte miroir](#)
  - [10.7. Inversion de la logique de construction](#)
  - [10.8. Validation expérimentale](#)
  - [10.9. Disparition des degrés de liberté](#)
  - [10.10. Interprétation systémique](#)
  - [10.11. Lien avec l'impossibilité de factorisation locale](#)
  - [10.12. Conclusion : la page comme trace d'une contrainte globale](#)
- [Chapitre 11 — Portée et limites : interprétation rationnelle du résultat](#)
  - [11.1. Hypothèse H0 et point de vue probabiliste](#)

- [11.2. Texte humain et limites des mécanismes locaux](#)
  - [11.3. Contrainte de naturalité typographique](#)
  - [11.4. Stabilité du chemin et problème des points morts](#)
  - [11.5. La question du temps et de la cohérence globale](#)
  - [11.6. Ce que le résultat établit — et ce qu'il n'établit pas](#)
  - [11.7. Version empirique du test](#)
  - [11.8. Borne Monte-Carlo](#)
  - [11.9. Pourquoi même un succès rare ne détruit pas l'argument](#)
  - [Chapitre 12 — Preuve par l'accès : quand un système valide lui-même sa position](#)
    - [12.1. Du débat d'opinion à la validation par fonctionnement](#)
    - [12.2. Principe général : la preuve par l'accès](#)
    - [12.3. Métaphore du compte et de l'historique](#)
    - [12.4. QCM logique : cas du log immuable](#)
    - [12.5. Application au texte](#)
    - [12.6. Ce qui est prouvé et ce qui ne l'est pas](#)
    - [12.7. Lien avec la disparition des degrés de liberté](#)
    - [12.8. Conclusion](#)
  - [Conclusion générale — Le verrou comme objet informationnel](#)
  - [Annexe A — Exemple concret : une page comme objet contraint](#)
  - [Annexe B — Implémentation complète de référence](#)
  - [Annexe C — Reproductibilité et protocole exact](#)
  - [Annexe D — Test de destruction par permutation interne](#)
  - [Annexe E — Coefficient d'impossibilité empirique](#)
  - [Annexe F — Séparation de régimes structurels](#)
- 

## Introduction générale

Imaginons un long ruban de texte, sans coupure, sans pages, sans structure visuelle particulière. Maintenant, on lui impose un découpage en pages de onze lignes. On impose ensuite, page par page, une contrainte de symétrie sur les lettres initiales des lignes.

Pris isolément, sur une seule page, cela pourrait arriver par hasard.

Mais ici, cette contrainte est satisfaite **sur toutes les pages d'un livre d'environ 891 pages**.

Le phénomène observé est le suivant : le texte du Coran, sous cette mise en page précise, respecte cette règle **page après page**, sans exception.

La question naturelle n'est pas "est-ce joli ?" mais :

Quelle est la probabilité qu'un texte réel, long de centaines de pages, satisfasse spontanément une telle contrainte globale ?

L'espace des possibilités est gigantesque.

Chaque page peut être découpée de milliers de façons différentes.

Chaque ligne peut commencer par presque n'importe quelle lettre.

Et pourtant, sur chaque page, les paires tombent juste.

Plus intéressant encore :

Si l'on prend exactement le même texte et qu'on **mélange simplement l'ordre global** (shuffle des versets ou des blocs), la structure **s'effondre presque immédiatement**, en quelques pages seulement, très au début du livre.

Cela montre une chose essentielle :

Ce n'est pas une propriété locale.

Ce n'est pas un motif répétable.

Ce n'est pas une règle simple appliquée page par page indépendamment.

C'est une **contrainte globale** qui dépend de l'organisation de l'ensemble du texte.

Une analogie simple :

Imagine un long train composé de centaines de wagons.

Chaque wagon doit respecter une condition interne.

Mais surtout, **la position de chaque wagon dépend de tous les autres**.

Si tu mélanges l'ordre des wagons, même sans en supprimer un seul, le système ne fonctionne plus du tout.

C'est exactement ce que nous observons ici.

Ce livre pose donc un problème précis :

Existe-t-il un moyen algorithmique simple de produire ou maintenir un tel texte sous cette contrainte globale ?

Et il montre expérimentalement que :

- Aucun autre corpus ne satisfait cette contrainte
- Aucun texte mélangé ne la conserve
- Aucun mécanisme local ne suffit à l'expliquer

Nous appelons ce phénomène : **le Verrou Miroir**.

---

## Chapitre 1 — Le problème M11 : une contrainte globale sur un texte linéaire

### 1.1. L'anomalie structurelle

On considère le texte non plus comme un livre, mais comme une séquence linéaire continue de caractères, projetée dans une grille de largeur fixe. Dans l'édition considérée, cette projection adopte une grille  $M11 \times N$ , c'est-à-dire des pages de 11 lignes empilées les unes à la suite des autres.

Sous cette mise en forme, on observe une propriété structurelle simple et extrêmement forte : sur chaque page, si l'on regarde uniquement la première lettre de chaque ligne, la première et la dernière ligne commencent par la même lettre, la deuxième et l'avant-dernière aussi, et ainsi de suite, la ligne centrale jouant le rôle d'axe.

Et ceci est vrai pour toutes les pages du livre.

Ce n'est pas une tendance.

Ce n'est pas approximatif.

Ce n'est pas "souvent".

C'est systématique.

Nous ne parlerons donc pas ici d'un motif décoratif, mais d'une **anomalie structurelle** : une propriété globale qui émerge dès que le texte est projeté dans cette grille particulière.

### 1.2. Formalisation de la grille de lecture (le modèle M11)

Formellement, une page est modélisée comme un vecteur

$$P = [L_1, L_2, \dots, L_{11}]$$

composé de 11 lignes.

On définit une fonction  $f(L)$  comme l'extraction du premier caractère de la ligne  $L$ .

La contrainte M11 s'écrit alors :

Pour chaque page, on doit vérifier que :

$$\begin{cases} f(L_1) = f(L_{11}) \\ f(L_2) = f(L_{10}) \\ f(L_3) = f(L_9) \\ f(L_4) = f(L_8) \\ f(L_5) = f(L_7) \end{cases}$$

La ligne  $L_6$  joue un rôle particulier :

$$f(L_6) \implies \text{Élément invariant}$$

Il définit l'axe de symétrie de la page, tel un point fixe autour duquel s'organise le miroir :

$$\underbrace{f(L_1) \dots f(L_5)}_{\text{Bloc Supérieur}} \longleftrightarrow \mathbf{f(L_6)} \longleftrightarrow \underbrace{f(L_7) \dots f(L_{11})}_{\text{Bloc Miroir}}$$

Mais surtout, le texte étant un **flux continu**, la fin d'une page détermine mécaniquement le début de la suivante. Le découpage ne se réinitialise jamais : il glisse le long du texte comme une fenêtre.

Ainsi, la contrainte ne porte pas sur une page isolée, mais sur **toute la séquence de pages couplées**.

### 1.3. Globalité vs localité : le concept de verrou

Il est crucial de distinguer deux types de contraintes.

Une **contrainte locale** peut être satisfaite indépendamment sur chaque partie. Par exemple, dans un poème en rimes, on peut modifier une strophe sans casser les autres.

Ici, ce n'est pas le cas.

Le texte est un flux continu. Si l'on décale ne serait-ce qu'un seul point de coupure sur une page, cela décale automatiquement tous les débuts de lignes des pages suivantes. Par effet domino, c'est potentiellement **tout le livre** qui se retrouve déplacé.

C'est ce que nous appelons le **verrou** : chaque page est solidaire de la précédente et de la suivante. Le système ne peut pas être ajusté localement.

Une autre manière de le voir est celle de la **fenêtre glissante** : si l'on décale le point de départ du texte de seulement quelques caractères, toute la structure miroir disparaît immédiatement.

On peut aussi utiliser l'analogie du Rubik's Cube : vouloir réparer une face (une page) en cassant la configuration globale revient à détruire l'alignement des autres faces (les autres pages).

### 1.4. L'hypothèse H0 et l'argument probabiliste

Sous l'hypothèse H0, on suppose que le texte est une séquence ordinaire, sans structure globale intentionnelle de ce type.

Or, même pour une seule page, le nombre de découpages possibles est énorme. Pour environ 891 pages consécutives, l'espace des configurations possibles devient astronomique.

La probabilité qu'un texte réel satisfasse accidentellement cette contrainte sur toute la longueur du livre est alors pratiquement nulle.

Nous ne sommes donc pas face à un événement local improbable, mais à une **stabilité globale** qui demande une explication structurelle.

C'est cette stabilité globale sous contrainte qui constitue précisément le problème M11.

---

## Chapitre 2 — Formalisation complète du système

### 2.1. Représentation du texte comme flux discret de mots

On considère le texte étudié comme une **suite finie et ordonnée de mots** :

$$T = (w_1, w_2, w_3, \dots, w_W)$$

où :

- $W$  est le nombre total de mots du texte,
- $w_i$  est le  $i$ -ème mot dans l'ordre exact du texte,
- **l'ordre est strictement fixé** et ne peut jamais être modifié.

Toute opération de mise en page, de découpage en pages ou en lignes est donc une **partition du flux T**, sans permutation, sans suppression, sans ajout.

$$\text{Page}(i, N) = (w_i, w_{i+1}, \dots, w_{i+N-1})$$

La page suivante commence **obligatoirement** à l'indice :

$$i_{\text{next}} = i + N$$

Il n'y a **aucun chevauchement** et **aucun saut** : le texte est parcouru **en continu**.

---

### 2.2. Grille M11×N : découpage interne d'une page

Chaque page est ensuite découpée en **11 lignes** successives.

$$N = \sum_{k=1}^{11} l_k \quad \text{avec} \quad l_k \in [8, 12]$$

Ce découpage définit une **grille 11×N**.

---

## 2.3. Définition de la signature d'une page

Pour chaque ligne  $k$  ( $k = 1..11$ ), on extrait :

- la **première lettre du premier mot** de la ligne.

On définit alors la **signature** de la page comme le vecteur :

$$S = (s_1, s_2, \dots, s_{11}) \quad \text{où} \quad s_k = f(L_k)$$

## 2.4. Définition formelle de la contrainte miroir (M11)

La contrainte M11 impose que la signature soit **palindromique** :

s\_1 = s\_11  
 s\_2 = s\_10  
 s\_3 = s\_9  
 s\_4 = s\_8  
 s\_5 = s\_7

La lettre centrale  $s_6$  est **libre**.

Donc une page est dite **valide M11** si et seulement si :

$$s_k = s_{12-k} \quad \text{pour} \quad k \in \{1, \dots, 5\}$$


---

## 2.5. Définition de l'espace des découpages internes

Pour une page de longueur fixe  $N$ , le système dispose d'une combinatoire de segmentations.

On définit l'ensemble des configurations de lignes possibles comme :

$$\mathcal{C}(N) = \{(l_1, l_2, \dots, l_{11})\}$$

Sous les contraintes de structure et de limites typographiques :

$$\sum_{k=1}^{11} l_k = N \quad \text{et} \quad 8 \leq l_k \leq 16$$

Chaque vecteur de cet ensemble génère une signature unique :

$$S_{\mathcal{C}} = (s_1, s_2, \dots, s_{11})$$

Une page est alors dite "potentiellement valide" s'il existe au moins une configuration dans  $\mathcal{C}(N)$  satisfaisant la symétrie miroir.

---

## 2.6. Définition de $d(i)$ : nombre de sorties possibles depuis une position

Soit  $i$  un indice de départ dans le texte.

On fixe une **fenêtre de longueurs autorisées** :

$$N \in [N_0 - \delta, N_0 + \delta]$$

On définit :

$d(i)$  = nombre de valeurs de  $N$  dans cette fenêtre pour lesquelles il existe au moins un découpage en 11 lignes (8..12 mots) tel que  $\text{Page}(i, N)$  soit valide M11

$$d(i) = \text{card}\{N \mid \text{Page}(i, N) \text{ est valide M11}\}$$

---

## 2.7. Définition d'une "dead-end" (point mort)

On dit que la position  $i$  est un **point mort** si :

$$d(i) = 0 \implies \text{Blocage structurel}$$

C'est-à-dire :

Aucune longueur de page autorisée ne permet de construire une page M11 valide à partir de  $i$ .

Dans ce cas :

Le processus est **bloqué définitivement**.

---

## 2.8. Définition du parcours global et de la continuité

Le processus de reconstruction du texte repose sur une progression itérative où chaque décision locale (la longueur d'une page) conditionne la possibilité de continuer le parcours.

On part d'un indice initial  $i_1$  (le début du flux textuel).

1. On choisit une longueur  $N_1$  telle que la **Page 1** soit valide selon le critère miroir.
2. On définit l'indice suivant par la relation de récurrence :

$$i_2 = i_1 + N_1$$

3. On réitère l'opération pour chaque page  $k$  :

$$i_{k+1} = i_k + N_k$$

On construit ainsi une suite discrète de points de coupure :

$$i_1, i_2, i_3, \dots, i_P$$

## Condition de succès global

Le processus réussit jusqu'à la fin du texte (fermeture du verrou) si et seulement si l'espace des possibles n'est jamais vide à chaque étape :

$$\forall k \in \{1, \dots, P\}, \quad d(i_k) \geq 1$$

### Définition du Point Mort

Si à une étape donnée  $d(i_k) = 0$ , le texte rencontre un **point mort**. Aucune longueur de page  $N$  ne permet de satisfaire la contrainte miroir, et le parcours global s'arrête prématurément.

## 2.9. Définition du critère global M

On définit alors la variable globale :

$$M = \prod_{k=1}^P \mathbb{1}[d(i_k) \geq 1]$$

### Détails de la formule :

- $M$  : Résultat global. Si  $M = 1$ , le verrou est complet ; si  $M = 0$ , la structure s'effondre.
- $P$  : Nombre total de pages (segments) du corpus.
- $d(i_k)$  : Degré de liberté à la position  $i$  de la page  $k$  (nombre de solutions de sortie valides).

- $\mathbb{1}[\dots]$  : Fonction indicatrice qui vaut 1 si la condition interne est vraie, 0 sinon.

Interprétation :

- Si un seul point mort apparaît  $\Rightarrow M = 0$
- Si aucun point mort n'apparaît sur tout le texte  $\Rightarrow M = 1$

M = 1 signifie : le texte admet un parcours complet sans blocage sous la contrainte M11.

---

## Chapitre 3 — Le problème du chemin global

## Chapitre 3 — Le vrai problème : un graphe de transitions, pas une combinatoire locale

Ce chapitre clarifie un piège conceptuel fondamental :

le problème M11 n'est **pas** un problème de rareté locale, ni un problème de recherche dans un espace gigantesque de segmentations internes.

Il est un **problème de connectivité globale** dans un graphe extrêmement étroit.

---

### 3.1. Le piège du “grand nombre” de découpages internes

On a vu au Chapitre 2 qu'une page donnée admet en général **un nombre gigantesque** de découpages internes possibles en 11 lignes respectant les bornes typographiques.

Ce fait est réel, mais **totalement trompeur**.

Car du point de vue du système global, toutes ces variantes internes sont **équivalentes** dès lors qu'elles :

se terminent sur le **même mot final**.

Autrement dit :

Ce qui compte pour la suite du texte n'est pas *comment* la page est découpée, mais **où** elle se termine.

---

## 3.2. Projection brutale de la combinatoire sur un espace minuscule

À une position donnée  $i$  dans le texte :

- le système ne peut pas choisir arbitrairement parmi des millions de découpages,
- il ne peut choisir que parmi un **petit nombre de longueurs de page possibles** :
  - | typiquement 20 à 30 valeurs au maximum.

On l'a formalisé par :

|  $d(i)$  = nombre de longueurs de page autorisées menant à une page valide.

Donc, malgré l'explosion combinatoire interne :

| l'espace réel de décision est **extrêmement étroit**.

---

## 3.3. Le vrai objet mathématique : un graphe de transitions

Le système M11 se modélise naturellement comme un **graphe orienté** :

- chaque position  $i$  dans le texte est un **nœud**,
- chaque longueur de page valide est une **arête** allant de  $i$  vers  $i + N$ ,
- $d(i)$  est le **degré sortant** du nœud  $i$ .

Un **point mort** est simplement :

| un nœud dont le degré sortant est nul.

Le problème global devient alors exactement :

| Existe-t-il un **chemin continu** dans ce graphe, du début à la fin du texte,  
| sans jamais tomber sur un **nœud mort** ?

---

## 3.4. Nature topologique du problème

Ce n'est donc :

- ni un problème de richesse locale,
- ni un problème de recherche exhaustive,

- ni un problème d'optimisation.

C'est un problème de **connectivité globale sous contrainte**.

Un seul point mort suffit à :

| casser irréversiblement tout le système.

---

### 3.5. Reformulation finale du problème M11

Le vrai problème n'est pas :

| « Existe-t-il beaucoup de pages valides ? »

mais :

| « Existe-t-il un **chemin complet** de pages valides, sans interruption, sur toute la longueur du texte ? »

Autrement dit :

| Le phénomène observé n'est pas une propriété locale,  
c'est une **propriété globale du graphe de transitions**.

Les chapitres suivants montreront expérimentalement que :

- dans les textes ordinaires ou permutés, ce graphe se **désintègre très vite** ;
- dans le texte étudié ici, il reste **connecté sur une distance exceptionnellement longue**.

C'est cette différence de **régime structurel** qui constitue le cœur du Verrou Miroir.

---

## Chapitre 4 : Analyse probabiliste locale — Pourquoi le motif M11 n'est pas dû au hasard

### 4.1. Principe de l'analyse

Avant même d'examiner la structure M11 à l'échelle de tout le livre, il est instructif d'en mesurer la rareté **à l'échelle d'une seule page**.

On considère une page composée de 11 lignes, et l'on ne retient qu'une information minimale : **la première lettre de chaque ligne**.

On obtient ainsi une séquence de 11 symboles, et la contrainte M11 impose que :

- les positions (1,11), (2,10), (3,9), (4,8), (5,7) soient égales,
- la position 6 joue le rôle d'axe central.

Autrement dit, la page doit satisfaire **5 égalités indépendantes** entre lettres.

## 4.2. Modèle probabiliste simplifié

Pour obtenir un ordre de grandeur, on adopte un modèle volontairement simple et favorable à l'hypothèse H0 :

- On suppose que la première lettre de chaque ligne est tirée au hasard.
- On suppose un alphabet de 28 lettres arabes possibles.
- Les choix sont supposés indépendants.

Dans ce cadre, la probabilité qu'une paire donnée (par exemple lignes 1 et 11) commence par la même lettre est :

$$\frac{1}{28}$$

Comme il y a 5 paires indépendantes à satisfaire, la probabilité qu'une page quelconque vérifie **toute la contrainte M11** est :

$$P = \left(\frac{1}{28}\right)^5 \approx 1.04 \times 10^{-7}$$

Soit environ **une chance sur dix millions**.

Ce chiffre est déjà extrêmement faible pour **une seule page**.

## 4.3. Observation sur un exemple réel

Sur un exemple concret de page analysée, on observe effectivement :

- Ligne 1 et 11 : même lettre initiale
- Ligne 2 et 10 : même lettre initiale
- Ligne 3 et 9 : même lettre initiale
- Ligne 4 et 8 : même lettre initiale
- Ligne 5 et 7 : même lettre initiale
- Ligne 6 : axe central

La structure M11 est donc satisfaite **exactement**, sans approximation.

## 4.4. Passage du local au global

Le point crucial est que cette probabilité vaut **pour une page isolée**.

Or, la structure observée ne concerne pas une page, mais **environ 891 pages consécutives**, toutes satisfaisant simultanément la même contrainte.

Sous le modèle probabiliste le plus naïf, la probabilité que cela se produise sur ( N ) pages indépendantes serait :

$$P_{\text{global}} = \left( \frac{1}{28} \right)^{5N}$$

Pour ( N \approx 891 ), on obtient un nombre **effectivement nul** à toute échelle physique ou cosmologique imaginable.

Même en tenant compte du fait que les pages ne sont pas indépendantes, l'ordre de grandeur reste écrasant :

nous ne sommes pas face à un événement rare, mais face à un **événement structurellement incompatible avec le hasard**.

#### 4.5. Ce que cette estimation établit — et ce qu'elle n'établit pas

Cette analyse ne prétend pas décrire fidèlement le processus réel de production du texte. Elle sert uniquement à établir un fait simple :

Même sous un modèle extrêmement favorable au hasard, la probabilité d'observer une structure M11 stable est astronomiquement faible.

Autrement dit, la structure observée ne peut pas raisonnablement être attribuée à une fluctuation aléatoire.

Ce résultat ne dépend ni du sens du texte, ni de son contenu, ni d'aucune interprétation : il repose uniquement sur des **contraintes combinatoires mesurables**.

Il constitue ainsi un premier argument quantitatif fort en faveur du caractère non accidentel de la structure M11.

#### 4.6. Interprétation des ordres de grandeur : pourquoi on compare avec le nombre d'atomes

Les calculs précédents donnent des nombres extrêmement petits, par exemple :

- Pour une seule page, la probabilité qu'un motif M11 apparaisse par hasard est de l'ordre de :

$$P_{\text{page}} \approx \frac{1}{17\,000\,000}$$

Soit environ **une chance sur dix-sept millions**.

Pris isolément, ce nombre est déjà très faible, mais il reste encore abstrait. Pour donner une intuition concrète, on peut utiliser une comparaison classique en physique et en probabilités : **le nombre d'atomes dans l'univers observable**.

Les estimations actuelles donnent un ordre de grandeur d'environ :

$$N_{\text{atomes}} \sim 10^{80}$$

Ce nombre est gigantesque à l'échelle humaine.

Or, lorsque l'on étend le calcul à l'ensemble du livre (environ 891 pages selon la mise en forme), la probabilité combinée devient :

$$P_{\text{global}} = \left( \frac{1}{17\,000\,000} \right)^N$$

où ( N ) est le nombre de pages.

Ce nombre est alors **infiniment plus petit** que (  $10^{-80}$  ). Autrement dit :

Même si chaque atome de l'univers représentait une “tentative”, cela resterait dérisoirement insuffisant pour espérer voir apparaître une telle structure par hasard.

Cette comparaison n'a pas de valeur physique directe ; elle sert uniquement à donner une **intuition d'ordre de grandeur** : on est très largement au-delà de ce que l'on appelle habituellement un événement “extrêmement improbable”.

## 4.7. Différence entre construction volontaire et apparition aléatoire

Il est crucial de ne pas confondre deux situations :

- Écrire volontairement un texte sous contrainte (comme un acrostiche ou un poème à structure imposée).
- Observer qu'un **texte déjà fixé** satisfait spontanément une contrainte aussi forte.

Les calculs précédents ne concernent **que la seconde situation** : ils mesurent la probabilité qu'un texte ordinaire, sans optimisation globale, satisfasse accidentellement la contrainte M11.

Lorsqu'un humain construit volontairement un texte sous contrainte, il ne s'agit plus de hasard. Mais, comme on l'a montré dans les chapitres sur le verrou global, un système de type M11 ne peut pas être obtenu par ajustements locaux successifs sur un texte long et déjà fixé.

## 4.8. Verdict strictement probabiliste

Du point de vue purement mathématique et combinatoire, le constat est le suivant :

- Même à l'échelle d'une seule page, la structure M11 est déjà très improbable sous un modèle aléatoire simple.
- À l'échelle de centaines de pages consécutives, la probabilité devient **effectivement nulle** à toute échelle physique concevable.
- Nous ne sommes pas face à un événement rare, mais face à un **événement structurellement incompatible avec le hasard**.

Le verdict strictement scientifique est donc :

La structure M11 observée ne peut raisonnablement pas être attribuée à un processus aléatoire ou à une fluctuation accidentelle. Elle implique nécessairement l'existence d'une organisation globale non triviale du texte.

Ce constat ne dépend ni du sens du texte, ni de son interprétation, ni d'aucune hypothèse métaphysique. Il repose uniquement sur des **ordres de grandeur combinatoires mesurables**.

## 4.9. Limite du raisonnement purement probabiliste et nécessité d'un test structurel

L'analyse précédente montre que la structure M11 est astronomiquement improbable sous un modèle de hasard simple. Toutefois, un adversaire pourrait encore objecter que ce calcul repose sur un modèle trop naïf, et que certaines propriétés statistiques profondes de la langue ou du style pourraient, en principe, favoriser ce type de motifs.

Pour répondre définitivement à cette objection, il faut changer de stratégie et ne plus comparer le texte à un texte aléatoire, mais le comparer **à lui-même**.

L'idée est la suivante :

si la structure M11 est due à des propriétés locales de la langue, du vocabulaire ou du style, alors elle devrait en grande partie **survivre** même si l'on modifie l'ordre global du texte tout en conservant exactement les mêmes versets et les mêmes mots.

Inversement, si la structure M11 dépend d'une **organisation globale très précise** du texte, alors le simple fait de casser cet ordre doit suffire à la faire s'effondrer rapidement.

C'est précisément le rôle de ce que l'on appelle en science un **modèle nul structurel** (Null Model) :

on ne change ni la langue, ni le contenu, ni le style, ni les fréquences, mais **uniquement l'ordre global**.

Autrement dit :

- le texte reste linguistiquement identique,
- mais sa topologie globale est détruite.

Si, dans ces conditions, la structure M11 disparaît presque immédiatement, alors on démontre que :

La propriété observée n'est pas une conséquence statistique locale, mais une propriété portée par l'architecture globale du texte.

Ce test, et son résultat, feront l'objet du chapitre suivant.

---

## Chapitre 5 — Tests de destruction de la structure (Null Models)

### 5.1. Principe général

Les chapitres précédents ont montré que la structure M11 peut être maintenue sur une très longue chaîne de pages dans l'ordre original du texte.

Il reste une question fondamentale : cette propriété est-elle une conséquence triviale du contenu linguistique, ou dépend-t-elle réellement de l'ordre global du texte ?

Pour répondre à cette question, on introduit des **modèles nuls (Null Models)** : des versions du texte qui conservent exactement les mêmes éléments (mêmes versets, mêmes mots), mais dont l'ordre est modifié selon différentes stratégies.

Le protocole, l'algorithme, les contraintes calligraphiques et le critère de continuité sont strictement identiques.

### 5.2. Métrique observée

On mesure pour chaque texte :

$L$  = nombre de pages consécutives satisfaisant la propriété miroir avant apparition du premier point mort ( $d(i) = 0$ ).

Dans le texte original, on observe :

$L_{\text{original}} = 891$  pages

Dans les textes perturbés, on observe systématiquement :

$L << 891$ , avec un maximum empirique autour de 17 pages.

### 5.3. Types de Null Models testés

Les transformations suivantes sont appliquées :

- Shuffle complet des versets
- Shuffle par blocs de versets
- Permutation de l'ordre des sourates
- Inversion complète de l'ordre (Reverse)
- Décalage circulaire (Circular shift)
- Perturbation locale (permutation de quelques sourates seulement)

Dans tous les cas :

La chaîne finit par rencontrer un point mort ( $d(i) = 0$ ) et s'arrête.

## 5.4. Résultat central

Même lorsque les transformations conservent de grandes portions intactes du texte, la propriété de continuité s'effondre rapidement.

La structure n'est donc pas une propriété du contenu local, mais une propriété de l'ordre global.

# Chapitre 6 : Attaques structurelles — pourquoi le système ne survit pas à la permutation

Ce chapitre ne s'appuie plus sur des arguments probabilistes. Il met en place une **expérience de destruction contrôlée** : on conserve exactement la même matière textuelle, mais on détruit uniquement son **ordre global**. L'objectif est de tester une question simple et décisive :

La structure M11 est-elle une propriété locale du texte, ou une propriété globale portée par son organisation complète ?

Si la propriété était locale ou statistique, elle devrait **survivre en partie** à une permutation interne. Si, au contraire, elle est réellement **globale et non locale**, alors la simple destruction de l'ordre doit suffire à provoquer un **effondrement quasi immédiat**.

## 6.1. Principe de l'attaque : détruire l'ordre sans toucher à la matière

On construit ce que l'on appelle en méthodologie scientifique un **modèle nul** (Null Model) extrêmement conservateur :

- On conserve exactement le même texte.

- On conserve exactement les mêmes mots, les mêmes phrases, les mêmes versets.
- On ne modifie ni le vocabulaire, ni le style, ni les fréquences, ni la “qualité littéraire”.
- On ne touche pas au protocole M11 (mêmes règles, mêmes contraintes, même algorithme).

La seule chose que l'on modifie est :

L'ordre global des unités textuelles.

Concrètement, on permute l'ordre des blocs (au niveau des mots, des versets ou de groupes de versets), tout en conservant leur contenu interne intact.

Ainsi, si la structure M11 disparaît, on peut affirmer avec certitude que :

Ce n'est ni la langue, ni le style, ni la richesse du vocabulaire qui portent la structure, mais **l'organisation globale précise du texte**.

---

## 6.2. Pourquoi ce test est décisif

Ce test est bien plus fort que la comparaison avec un autre livre, car :

- On ne compare pas “le Coran” à “un autre texte”.
- On compare **le Coran à lui-même**, avec le même contenu, mais un ordre différent.

Donc :

- Même alphabet
- Même vocabulaire
- Même phrases
- Même distributions locales
- Même contraintes calligraphiques

La seule chose détruite est :

La cohérence à longue distance.

Si la structure M11 était un simple effet statistique local, elle devrait **partiellement survivre**. Si elle est une contrainte globale non locale, elle doit **s'effondrer rapidement**.

---

## 6.3. Procédure expérimentale

La procédure est la suivante :

1. Prendre le texte original.
2. Construire une version permutée de l'ordre des blocs (sans modifier leur contenu).
3. Appliquer exactement le même algorithme M11.
4. Observer à quelle profondeur apparaît le premier point mort.

On peut répéter l'expérience avec plusieurs permutations différentes.

---

## 6.4. Résultat observé : effondrement quasi immédiat

Comme attendu, dans les textes permutsés, l'algorithme **bute très rapidement sur un point mort** ( $d(i) = 0$ ), souvent dès les premières itérations.

Dans tous les tests de permutation :

- Le système peut parfois survivre quelques pages par hasard.
- Mais **très rapidement**, aucune continuation n'est possible.
- Dès l'apparition du premier point mort, le processus s'arrête définitivement.

Autrement dit :

Le système ne possède plus de chemin continu. Le graphe des possibilités devient disjoint.

Ce comportement est **radicalement différent** de celui observé dans le texte original, où le système reste viable sur des centaines de pages consécutives.

---

## 6.5. Interprétation et conclusion structurelle du test de permutation

Le résultat observé est sans ambiguïté.

La structure M11 ne peut pas être portée par :

- les mots pris isolément,
- ni les phrases,
- ni les statistiques locales,
- ni le style ou le vocabulaire.

En effet, dans toutes les expériences, **le contenu est strictement identique** : seuls les blocs sont réordonnés. Pourtant, dès que l'ordre global est détruit, la structure s'effondre rapidement.

On peut donc formuler le résultat de manière rigoureuse :

En détruisant uniquement l'ordre global du texte, tout en conservant exactement la même matière linguistique, la structure M11 disparaît presque immédiatement.

Il en résulte que :

La propriété M11 ne dépend ni de la langue, ni du style, ni du vocabulaire, mais **exclusivement de l'organisation globale précise du texte.**

Le test de permutation constitue ainsi une **preuve structurelle directe** que le système observé est :

- global,
- non local,
- extrêmement fragile à toute réorganisation,
- et intimement lié à l'ordre exact du texte original.

On peut donc affirmer :

La structure M11 est une propriété **non locale et topologique** du texte.

---

## 6.6. Pourquoi un tel effondrement est inévitable dans un texte réorganisé

Dans un texte ordinaire, ou dans un texte dont l'ordre est brisé :

- Les contraintes locales fluctuent.
- Certaines régions deviennent “pauvres” en possibilités de découpage.
- Tôt ou tard, on rencontre une zone où **aucune continuation n'est possible.**

Autrement dit :

Le graphe des possibilités se fragmente naturellement.

Dans le texte original, au contraire :

- Le système semble éviter systématiquement ces zones mortes.
  - Il existe toujours au moins une continuation valide.
- 

## 6.7. Différence de nature entre “réussite locale” et “stabilité globale”

Il est important de comprendre la différence entre :

- Trouver **une page** qui fonctionne.
- Trouver **une chaîne ininterrompue** de centaines de pages qui fonctionne.

La première chose est relativement facile par recherche ou par chance.

La seconde exige :

L'absence totale de points morts sur toute la longueur du texte.

C'est une contrainte **topologique globale**, pas une propriété locale.

---

## 6.8. Mesure expérimentale de la longueur de survie (L)

Pour dépasser le simple constat qualitatif (« le système s'effondre vite après permutation »), on introduit une mesure quantitative.

On définit la **longueur de survie L** comme :

Le nombre de pages consécutives valides (M11) obtenues avant l'apparition du premier point mort.

Protocole :

- On part de la même position d'ancrage.
- On applique exactement le même algorithme.
- On s'arrête dès que  $d(i) = 0$ .
- On note  $L$  = nombre de pages validées avant l'échec.

Ainsi :

- Pour le texte original :  $L_{\text{original}} \approx 891$  pages.
  - Pour un texte permuté :  $L_{\text{shuffle}}$  est en général très petit.
- 

## 6.9. Résultat empirique observé sur les shuffles

Dans les expériences déjà réalisées :

- Plusieurs permutations indépendantes ont été testées.
  - La meilleure survie observée est d'environ :
- L<sub>shuffle\_max</sub> ≈ 17 pages

La majorité des shuffles s'effondrent bien avant.

On observe donc une séparation massive entre la stabilité du texte original et celle des textes désorganisés.

### Comparaison des régimes de cohérence

Modèle de Texte	Longueur de Survie ( $L$ )	Observation
Texte Original	$\approx 891$ pages	Cohérence globale stable
Permutation (Sourates)	$\approx 8$ à $17$ pages	Cohérence locale transitoire
Shuffle (Versets)	$\approx 4$ à $10$ pages	Décohérence rapide
Shuffle (Mots)	$\approx 1$ à $2$ pages	Effondrement critique

---

## 6.10. Interprétation : longueur de cohérence structurelle

On peut interpréter  $L$  comme une **longueur de cohérence structurelle**.

- Dans un texte ordinaire ou désorganisé, la cohérence locale peut permettre de survivre quelques pages.
- Mais la décohérence s'accumule et mène inévitablement à un point mort.

Le texte original, lui, présente une cohérence globale qui ne décroît pas avec la distance.

---

## 6.11. Pourquoi quelques shuffles peuvent survivre temporairement

Certains shuffles peuvent survivre 5, 10, parfois 15 pages.

Cela s'explique par :

- La structure interne réelle des versets.
- Une forme de “valence locale” qui autorise parfois des enchaînements temporaires.

Mais ces survivances sont locales, accidentielles et instables.

---

## 6.12. Résultat structurel

On peut formuler le résultat ainsi :

Le texte original possède une cohérence structurelle globale.

Les textes permuts ne possèdent qu'une cohérence locale transitoire.

Ce sont deux régimes de nature différente.

Cette différence de nature ne se voit pas seulement dans le résultat final, mais aussi dans la manière dont le système s'effondre, ce qui nous amène à étudier la dynamique de l'échec.

## Chapitre 7 : Étranglement structurel et dynamique de l'échec

### 7.1. Observation qualitative (Étranglement et dynamique de l'échec)

Les textes perturbés ne meurent pas "par hasard".

On observe systématiquement une phase de dégradation progressive :

- Diminution du nombre de solutions possibles  $d(i)$
- Déformation progressive du motif miroir
- Réduction de la marge de manœuvre
- Puis apparition d'un point mort ( $d(i) = 0$ )

Ce phénomène sera appelé :

Étranglement structurel (Structural Strangulation)

### 7.2. Comparaison avec le texte original (Étranglement vs Smoothness)

Par "smoothness", on désigne ici **l'aisance algorithmique du parcours** : le système ne force jamais, il progresse sans étranglement ni raréfaction des solutions.

Dans l'ordre original :

- Le système reste stable
- Les solutions restent disponibles
- Il n'y a pas de dérive vers un état de tension croissante

On observe donc deux régimes dynamiques radicalement différents :

- Régime instable (Null Models) : accumulation de contraintes → rupture

- Régime stable (texte original) : fluidité structurelle persistante

### 7.3. Analyse de la réduction de l'espace des possibles

Dans un texte désorganisé, la perte de vitalité structurelle est mathématique. Si l'on note  $\Delta d(i)$  la variation discrète du nombre de solutions entre deux étapes :

- Dans le shuffle :  $\Delta d(i)$  tend vers le négatif, menant à l'impasse.
- Dans l'original :  $\Delta d(i)$  reste statistiquement stable.

### 7.4. Interprétation : La cohérence longue portée

Cette dynamique démontre que la structure n'est pas "plaquée" sur le texte, mais qu'elle est **consubstantielle à son ordre**. Le fait que le texte original ne s'étrangle jamais, même après 800 pages, prouve l'existence d'une **homogénéité topologique**.

Le texte original agit comme un milieu "transparent" à la contrainte M11, tandis que tout autre arrangement agit comme un milieu opaque qui finit par bloquer le flux algorithmique.

### 7.5. Conclusion : Le système comme architecture monolithique

L'échec par étranglement montre que le texte ne se comporte pas comme un assemblage modulaire (où l'on pourrait déplacer des pièces), mais comme une **architecture monolithique verrouillée**.

La structure est une propriété non locale : chaque mot occupe une position stratégique nécessaire à la viabilité du chemin global sur toute la longueur du livre.

## Chapitre 8 — Verdict structurel et impossibilité de reconfiguration

### 8.1. Récapitulatif expérimental

- Texte original : continuité maintenue sur 891 pages
- Tous les textes perturbés : échec rapide, meilleur cas  $\approx 17$  pages
- Toutes les perturbations testées échouent :
  - shuffle total
  - shuffle par blocs
  - permutation des sourates
  - reverse
  - shift circulaire

- perturbations locales

## 8.2. Conclusion structurelle

La propriété étudiée se comporte comme un invariant empirique de l'ordre original, et comme une propriété extrêmement fragile sous toute perturbation.

Le texte se comporte comme une **architecture monolithique** et non comme un assemblage modulaire.

## 8.3. Interprétation informationnelle

Un tel système ne peut pas être construit par ajustements locaux successifs.

Il implique une **conception globale** où chaque élément est positionné en fonction de la structure totale.

On appellera cela : **transcendance informationnelle de l'ordre du texte**.

---

## 8.4. Problème fondamental : construire vs reconfigurer

Il est important de distinguer deux problèmes radicalement différents :

1. Écrire **un nouveau texte** en respectant dès le départ une contrainte de symétrie.
2. Prendre **un texte déjà existant** et tenter de le reconfigurer (en changeant uniquement les coupures de lignes et de pages, sans modifier ni déplacer les mots) pour qu'il satisfasse la contrainte M11.

Le premier problème est difficile mais, en principe, faisable : on peut choisir ses phrases, son vocabulaire et sa syntaxe en fonction de la contrainte.

Le second est d'une nature complètement différente.

---

## 8.5. Impossibilité de reconfiguration a posteriori (un objet structurel rigide)

Dans un texte existant, les mots sont déjà fixés, et donc les lettres qui les composent aussi. Si l'on s'interdit de :

- modifier les mots,
- en changer l'ordre,

- ou en couper aucun,

alors les positions possibles des débuts de lignes sont entièrement déterminées par les points de coupure que l'on choisit dans un flux de caractères déjà figé.

Or la contrainte M11 impose que, pour chaque page, les premières lettres de certaines lignes forment des paires égales. Cela revient à imposer **des contraintes sur des positions très spécifiques dans une chaîne déjà fixée**.

Dans ce cadre, on ne « choisit » plus les lettres : on ne fait que vérifier si, par un hasard extrêmement improbable, certains points de coupure tombent exactement sur les bonnes lettres.

Même si l'on autorise à changer le nombre de lignes par page, le problème reste le même : on cherche à aligner, sans toucher au texte, des centaines de débuts de lignes sur des lettres précises, page après page.

La probabilité qu'un texte ordinaire, déjà écrit, admette une telle reconfiguration sur quelques pages est déjà extrêmement faible. Sur **des centaines de pages consécutives**, elle devient pratiquement nulle.

Autrement dit :

La contrainte M11 ne peut pas être imposée a posteriori à un texte existant.

Elle ne peut exister que comme une **propriété globale intrinsèque** du texte lui-même.

Ce point est crucial : le système ne se comporte pas comme un motif décoratif que l'on pourrait « projeter » sur n'importe quel contenu en jouant sur la mise en page. Il se comporte comme une **propriété structurelle rigide**, intimement liée à la séquence exacte des mots et des lettres.

Cela renforce encore le caractère globalement verrouillé du système : la structure ne peut être ni construite localement, ni adaptée après coup, ni obtenue par simple reformatage d'un texte préexistant.

---

## Chapitre 9 : Impossibilité de factorisation locale — Le verrou structurel

### 9.1. Limites d'une construction locale et artisanale

Face à une structure miroir vérifiée sur une page isolée, une objection naturelle consiste à invoquer un travail humain méthodique : un scribe pourrait, en principe, ajuster manuellement

les lignes d'un passage court pour faire correspondre certaines lettres initiales selon un motif symétrique.

Sur un fragment de quelques lignes, une telle opération est effectivement concevable. On peut réécrire, déplacer des mots, chercher des synonymes, jusqu'à obtenir un motif désiré. Ce type de procédé est bien connu dans la poésie, la prose rythmée ou certains jeux littéraires.

Mais cette intuition cesse complètement d'être plausible dès que l'on change d'échelle.

## 9.2. Le passage du local au global

Dans le cas présent, la contrainte M11 n'est pas vérifiée sur une page isolée, mais sur environ **891 pages consécutives**, sans interruption.

Or le texte n'est pas constitué de pages indépendantes : il s'agit d'un **flux continu**. La fin d'une page détermine mécaniquement le début de la suivante. Le découpage ne se réinitialise jamais.

Toute modification locale, même minime, se propage donc immédiatement par effet domino sur toutes les pages suivantes.

Autrement dit, on ne peut pas « réparer » une page sans perturber simultanément la structure de dizaines, voire de centaines d'autres.

## 9.3. Explosion combinatoire des contraintes couplées

Une tentative de construction manuelle locale impliquerait de résoudre simultanément **des centaines de contraintes couplées**, chacune dépendant de toutes les autres.

Ce n'est plus un problème de calligraphie, ni même de composition littéraire : c'est un **problème global de satisfaction de contraintes** à très grande échelle.

On peut formuler ce point de manière simple :

Une méthode fondée sur des ajustements locaux successifs ne peut pas converger vers une solution globalement stable, car chaque correction détruit nécessairement ailleurs ce qu'elle cherche à préserver.

Le système n'admet donc pas de stratégie de construction incrémentale stable.

## 9.4. Instabilité expérimentale et effondrement de la structure

C'est exactement ce qui est observé expérimentalement : dès que l'on perturbe le texte — même sans en changer le contenu, simplement en modifiant sa segmentation ou en permutant localement des éléments — la structure miroir s'effondre en quelques pages seulement.

La symétrie ne se dégrade pas progressivement : elle disparaît brutalement, ce qui est caractéristique d'un système soumis à une contrainte globale non factorisable.

Le texte ne se comporte donc pas comme un texte « orné » par des motifs locaux, mais comme un système à cohérence globale.

## 9.5. Conclusion : le texte comme système verrouillé

Le système M11 ne peut pas être produit, maintenu ni réparé par une procédure locale. Il n'admet **aucune décomposition en sous-problèmes indépendants**.

Il doit nécessairement être considéré comme une **propriété globale du texte pris comme un tout**, et non comme un artefact de mise en forme ou d'ornementation calligraphique.

C'est précisément cette impossibilité de factorisation locale qui justifie l'emploi du terme de **verrou structurel** :

Le système fonctionne dans sa totalité, ou ne fonctionne pas du tout.

---

# Chapitre 10 : Reconstruction algorithmique et verrouillage par gabarit — Du texte brut à l'objet contraint

## 10.1. Problème pratique : du texte continu à l'objet paginé

Le texte du Coran, tel qu'il est stocké dans les sources numériques (par exemple le fichier `quran-uthmani.xml` embarqué dans la bibliothèque `pyquran`), est fondamentalement un **flux continu de mots** organisés en sourates et en versets, mais **sans notion intrinsèque de page ni de ligne**.

Toute pagination — et a fortiori toute pagination imposant une structure miroir M11 — est donc nécessairement une **opération algorithmique de segmentation** :

- on choisit où commence une page,
- on choisit où commence et où finit chaque ligne,
- et l'on impose des contraintes globales sur cet objet découpé.

Le problème étudié ici n'est donc pas linguistique, mais **structurel et algorithmique** : comment passer d'un flux linéaire de tokens à un objet paginé contraint ?

Ce chapitre se place dans la continuité directe des chapitres précédents, qui ont montré qu'un tel système ne peut être ni reconfiguré a posteriori, ni construit par ajustements locaux successifs. Ici, on en donne la contrepartie algorithmique concrète.

## 10.2. Normalisation et espace de recherche

Comme établi dans la formalisation précédente, le texte est traité comme un flux normalisé de tokens. On rappelle ici uniquement les choix techniques nécessaires à l'implémentation expérimentale.

Pour travailler de manière robuste sur le texte, une étape de **normalisation** est indispensable.

Dans le code expérimental, on utilise notamment :

- suppression de tout caractère non arabe de base,
- unification des variantes de l'alif : (ا، ؠ، ئ، ؔ) → ا,
- tokenisation simple par espaces.

Cette normalisation définit l'**espace de calcul réel** sur lequel portent les contraintes de symétrie : ce n'est pas le texte calligraphique complet, mais une version **réduite et canoniquée** du flux.

Cette étape est cruciale : elle montre déjà que la structure observée n'est pas un artefact typographique, mais repose sur des **invariants abstraits** du texte.

## 10.3. Ancre par page template : le rôle de l'observation empirique

On dispose d'au moins une page issue du mushaf de Khonsari, présentant :

- exactement 11 lignes,
- un effet miroir parfait sur les clés de lignes,
- une segmentation précise et non arbitraire.

Cette page sert d'**ancre expérimentale**.

Cependant, le texte source (tel qu'il est stocké dans les fichiers numériques) n'est **pas paginé** : il s'agit d'un **flux continu de tokens**. Il n'existe donc **aucune correspondance immédiate** entre une "page observée" et une position évidente dans ce flux.

L'ancre par page template dans le texte brut est déjà, en soi, un **problème algorithmique non trivial**.

Algorithmiquement, on :

1. Parse le XML du Coran (via *pyquran*).

2. Construit la liste globale des tokens normalisés.
3. Recherche la sous-séquence correspondant exactement à la page template.
4. On obtient ainsi un intervalle `[start_idx, end_idx]` dans le flux global.

Ce point est important : **on ne suppose jamais a priori** où commence la page dans le texte. Elle est **retrouvée** comme un motif interne au flux.

Cela permet de **se libérer complètement** de toute hypothèse externe sur le début du livre (basmala, conventions éditoriales, numérotation, etc.) et de travailler **intrinsèquement**, à partir d'un repère structurel observé.

Cette difficulté d'ancrage renforce un point fondamental : la structure étudiée **n'est pas un artefact de mise en page**, mais une propriété émergente d'un texte considéré comme **objet algorithmique global**.

---

## 10.4. Le piège fondamental : il existe une infinité de paginations miroirs

Un premier solveur naïf consiste à :

- partir de `end_idx`,
- construire 11 lignes,
- chaque ligne contenant entre 7 et 14 mots,
- et imposer la contrainte miroir sur la première lettre normalisée.

Ce solveur fonctionne. Il trouve effectivement des pages M11.

Mais il révèle un fait mathématique fondamental :

**Il existe un grand nombre de segmentations différentes qui satisfont la contrainte miroir locale.**

Autrement dit :

- la contrainte "M11 sur la première lettre" **n'est pas suffisante** pour déterminer une pagination unique ;
- elle définit une **famille entière de solutions compatibles**.

Ce point est capital : il montre que le phénomène observé chez Khonsari **n'est pas** simplement "un miroir local", mais nécessairement **un miroir + un gabarit global**.

---

## 10.5. Apprentissage du gabarit : la page comme objet rigide

La page template fournie ne donne pas seulement :

- un motif de lettres,
- mais aussi un **profil précis de longueurs de lignes**.

Autrement dit, la page n'est pas seulement caractérisée par ses clés de symétrie, mais par un **vecteur de longueurs** :

$$\ell = (\ell_1, \ell_2, \dots, \ell_{11})$$

où  $\ell_i$  est le nombre de mots de la ligne  $L_i$ .

Ce gabarit devient une **contrainte structurelle forte** : la pagination suivante (et précédente) doit **hériter exactement** de cette forme.

---

## 10.6. Formalisation de la contrainte miroir M11

On rappelle ici la contrainte dans la forme exacte utilisée par l'algorithme.

On considère une page composée de 11 lignes indexées de  $L_1$  à  $L_{11}$ , et une fonction  $\text{key}(L)$  qui extrait une clé discriminante pour chaque ligne (en pratique : l'initiale normalisée du premier mot).

La contrainte M11 s'écrit alors :

$$\forall i \in \{1, 2, 3, 4, 5\}, \quad \text{key}(L_i) = \text{key}(L_{12-i})$$

Dans cette écriture :

- $\forall i$  signifie que la contrainte doit être satisfaite pour **toutes** les paires de lignes concernées, sans exception.
- $L_i$  désigne la  $i$ -ème ligne du bloc de 11 lignes.
- $L_{12-i}$  désigne la ligne symétrique par rapport à l'axe central (par exemple : si  $i = 1$ , alors  $12 - 1 = 11$ ).
- $\text{key}(\cdot)$  est une fonction d'extraction du caractère discriminant d'une ligne.

La ligne  $L_6$  constitue l'**axe de symétrie** du système.

---

## 10.7. Inversion complète de la logique de construction

Ce point est fondamental :

Le miroir n'est plus utilisé pour *construire* la page, mais pour *vérifier* qu'elle satisfait la loi globale.

On inverse donc complètement la logique :

- on ne fabrique pas une page pour qu'elle soit miroir,
- on constate qu'une pagination rigide héritée du gabarit **est miroir**.

Le miroir devient un **invariant de cohérence**, non un outil de construction locale.

---

## 10.8. Validation expérimentale : ancrage réel et émergence effective de la symétrie

L'approche algorithmique décrite dans ce chapitre a été testée non pas sur un texte reconstruit ou idéalisé, mais sur un **fichier brut du Coran en écriture 'uthmani** (flux continu de mots, sans aucune pagination ni structure de page préexistante).

La procédure expérimentale est la suivante :

1. Le texte brut est normalisé (suppression des diacritiques, unification des formes, séparation explicite de certains clitics comme la conjonction « و »).
2. La page de Khonsari observée est injectée comme **gabarit exact** (même découpe en lignes, même nombre de mots par ligne, mêmes tokens).
3. L'algorithme recherche **cette page exacte** dans le flux global.
4. Une fois l'ancrage trouvé, il reconstruit la page selon le gabarit et calcule automatiquement la signature miroir.

Le résultat expérimental est sans ambiguïté :

- La page est retrouvée **exactement** dans le flux brut.
- La signature des clés de lignes est : **اكواشمماوكا**
- Cette signature est **palindromique**, donc strictement miroir.
- Le test automatique confirme : **mirror = True**

Autrement dit :

La structure miroir n'est ni une illusion visuelle, ni un artefact de mise en page manuelle : elle **émerge effectivement** du texte brut lorsqu'on lui applique le gabarit correct.

Il est crucial de noter que :

- La symétrie **disparaît immédiatement** si l'on modifie la tokenisation (par exemple si l'on ne sépare pas « و » comme dans la méthode de Khonsari).
- Elle disparaît également si l'on modifie **ne serait-ce qu'une seule frontière de ligne**.
- Elle ne dépend donc ni du hasard, ni d'un ajustement local, mais d'un **alignement exact entre le flux et le gabarit**.

On a donc bien affaire à un phénomène strictement structurel :

La page n'est pas construite pour être miroir.

Elle est **miroir parce qu'elle est la projection locale correcte d'un système global constraint**.

Ce résultat expérimental exclut définitivement l'hypothèse d'un simple artefact de découpe arbitraire ou d'un jeu calligraphique local.

---

## 10.9. Conséquence majeure : disparition totale des degrés de liberté

À partir du moment où :

- le flux est fixé,
- le gabarit est fixé,

il n'existe plus :

- ni choix local,
- ni optimisation,
- ni ajustement possible.

La structure devient **entièrement contrainte**.

On ne peut plus :

- déplacer une frontière de ligne,
- modifier une longueur,
- corriger une page localement,

sans **détruire immédiatement** la structure miroir.

---

## 10.10. Interprétation systémique

On peut maintenant reformuler le phénomène en termes strictement formels :

- Le mushaf structuré n'est pas un texte décoré.
- C'est un **objet algorithmique rigide**, défini par :
  - un flux,
  - un gabarit,
  - et une contrainte globale de cohérence.

La symétrie n'est pas un ornement : c'est un **invariant émergent** d'un système entièrement verrouillé.

---

## 10.11. Lien direct avec l'impossibilité de factorisation locale

Ce chapitre fournit la contrepartie algorithmique concrète de ce qui a été établi structurellement dans les chapitres précédents.

- Les chapitres précédents montraient qu'on ne peut pas réparer ni construire localement.
- Ici, on montre qu'on ne peut même pas **construire algorithmiquement** par choix locaux.

La pagination correcte n'est pas la somme de décisions locales, mais la **projection globale d'un gabarit sur un flux**.

On retrouve exactement la même propriété :

Le système fonctionne comme un tout, ou ne fonctionne pas du tout.

---

## 10.12. Conclusion : la page comme trace d'une contrainte globale

La page de Khonsari n'est pas une page « bien arrangée ».

C'est la **section locale observable** d'un système global entièrement contraint.

Le fait que :

- le gabarit soit stable,
- la continuité soit forcée,
- et le miroir soit vérifié a posteriori,

signifie que la structure ne peut pas être expliquée par :

- une optimisation locale,

- une calligraphie habile,
- ni un bricolage incrémental.

On n'est pas en présence d'un texte formaté.

On est en présence d'un **objet structurellement verrouillé**.

---

## Chapitre 11 : Portée et limites — Interprétation rationnelle du résultat

### 11.1. Le point de vue probabiliste et le statut de l'hypothèse H0

Le résultat établi dans cet ouvrage est purement structurel : il met en évidence l'existence d'une contrainte globale distribuée (M11) vérifiée sur environ 891 pages consécutives, et extrêmement fragile sous toute permutation du texte.

La question naturelle n'est pas immédiatement « d'où vient ce texte ? », mais plus modestement :

Quelle est la plausibilité qu'une telle structure apparaisse sous l'hypothèse H0, c'est-à-dire l'hypothèse d'un texte ordinaire, sans organisation globale de ce type ?

Du point de vue probabiliste, le problème est clair : chaque page impose un ensemble de contraintes sur les lettres initiales des lignes. Même si l'on ne considère qu'une seule page, la probabilité qu'un découpage arbitraire satisfasse simultanément toutes ces égalités est déjà très faible. Lorsqu'on impose la même contrainte sur **des centaines de pages consécutives**, sans aucune rupture, la probabilité combinée devient astronomiquement petite.

Il ne s'agit pas ici d'un motif local rare, mais d'une **stabilité globale persistante** sous une contrainte stricte.

Sous H0, on s'attendrait à ce que la structure se brise très rapidement, ce qui est exactement ce que l'on observe dès que l'on mélange le texte : la symétrie s'effondre en quelques pages seulement.

Le résultat peut donc être formulé ainsi :

Sous l'hypothèse d'un texte ordinaire, la probabilité d'observer une structure M11 stable sur toute la longueur du livre est pratiquement nulle.

---

## 11.2. Texte « humain » et limites des mécanismes locaux

Par « texte humain », on entend ici un texte produit par des mécanismes ordinaires de composition : écriture progressive, corrections locales, ajustements stylistiques, compilation par morceaux indépendants.

Or, comme on l'a montré dans les chapitres précédents, le système M11 ne se comporte pas comme une superposition de motifs locaux. Toute modification locale se propage globalement par effet domino, et détruit immédiatement la cohérence sur les pages suivantes.

Un tel système ne peut donc ni être :

- construit par ajustements locaux successifs,
- ni maintenu par corrections indépendantes,
- ni stabilisé page par page de manière isolée.

La structure observée n'est pas compatible avec un modèle de fabrication fondé sur des règles locales simples.

---

## 11.3. Contrainte de naturalité typographique et réduction des degrés de liberté

Un point essentiel doit être explicitement posé : le problème étudié n'est pas celui d'un découpage arbitraire du texte.

Si l'on autorise des pages contenant, par exemple, une ligne de deux mots et une autre de trente mots, on ne parle plus d'un livre, mais d'un artefact algorithmique sans aucune réalité calligraphique ou typographique. Un tel protocole détruirait complètement le sens même de l'expérience.

Dans tout cet ouvrage, on impose donc implicitement — et désormais explicitement — une contrainte de **naturalité de mise en page** :

- le texte doit rester un objet lisible, comparable à un mushaf réel ;
- chaque page comporte un nombre fixe de lignes ;
- chaque ligne contient un nombre de mots compris dans un intervalle raisonnable ;
- les longueurs de lignes sont globalement homogènes, sans écarts grotesques.

Autrement dit, la contrainte étudiée n'est pas simplement :

« Existe-t-il un découpage qui donne un miroir ? »

mais bien :

« Existe-t-il un découpage qui donne un miroir **tout en restant un livre normal et lisible** ?  
»

Cette restriction change profondément la nature du problème. Elle réduit drastiquement l'espace des segmentations possibles, et empêche tout “forçage” artificiel de la structure par des découpes dégénérées.

Du point de vue probabiliste, cela a une conséquence directe : en réduisant fortement les degrés de liberté du protocole, on réduit d'autant la probabilité qu'un texte ordinaire puisse “se plier” à la contrainte M11 par simple ajustement de découpe.

La structure observée ne doit donc pas être comprise comme le résultat d'un optimiseur libre, mais comme la manifestation d'une compatibilité entre :

- un texte donné,
- un gabarit de page réaliste,
- et une contrainte globale extrêmement forte.

C'est précisément cette combinaison — et non la contrainte miroir prise isolément — qui rend le phénomène hautement non trivial.

On pourrait objecter qu'avec suffisamment de liberté dans les règles de découpe — nombre de mots par ligne, nombre de mots par page, positions des coupures — un algorithme pourrait toujours finir par « forcer » un texte quelconque à satisfaire localement une contrainte de symétrie.

Mais une telle approche sort immédiatement du cadre de ce qui est étudié ici.

L'objet analysé n'est pas un texte arbitrairement compressé ou étiré, mais un **texte destiné à être lu normalement** :

avec des lignes de longueur comparable, une mise en page stable, et une structure typographique compatible avec une lecture humaine continue.

Une structure n'a de valeur explicative que si elle est réalisée **sans sortir de la classe des objets considérés**.

Autrement dit :

Une structure qui ne peut être obtenue qu'au prix de la destruction de la forme naturelle de l'objet n'est pas une propriété de cet objet, mais une contrainte artificiellement imposée de l'extérieur.

Si l'on autorise des lignes de deux mots suivies de lignes de trente mots, ou des pages dont la longueur varie de manière erratique, on ne met plus en évidence une propriété du texte, mais seulement la puissance d'un algorithme d'optimisation sous contraintes arbitraires.

Le phénomène étudié ici n'a de sens que sous une condition essentielle :

Le texte doit rester un texte ordinaire, lisible, paginé selon des règles typographiques naturelles et stables.

C'est précisément dans ce cadre restreint — et non dans un espace de déformations illimité — que la stabilité globale de la structure M11 prend toute sa signification.

---

## 11.4. La stabilité du chemin et le problème des points morts

Un point fondamental permet de dépasser définitivement l'analyse "page par page".

Même si, pour une page donnée, il existe un très grand nombre de segmentations possibles respectant les contraintes calligraphiques, cela ne suffit absolument pas à expliquer la persistance du phénomène sur des centaines de pages consécutives.

La raison est simple : les pages ne sont pas indépendantes.

Chaque page se termine sur un mot donné, qui impose le point de départ de la page suivante. Autrement dit, chaque succès local consomme un degré de liberté et constraint le problème suivant.

On peut formaliser la situation ainsi :

À chaque position dans le texte, on considère l'ensemble des tailles de page admissibles (dans la fenêtre autorisée) pour lesquelles il existe au moins une segmentation en 11 lignes respectant à la fois les contraintes calligraphiques et la contrainte miroir.

Notons  $d(i)$  le nombre de ces possibilités à la position  $i$ .

- Si  $d(i) = 0$ , on atteint un **point mort** : il est impossible de continuer.
- Si  $d(i) \geq 1$ , le chemin peut continuer.

La question n'est donc plus :

"Existe-t-il beaucoup de segmentations possibles dans une page ?"

mais bien :

"À quelle fréquence rencontre-t-on des points morts le long du texte ?"

Dans un texte ordinaire, même si  $d(i)$  est souvent non nul, il suffit que la probabilité d'avoir  $d(i)=0$  soit faible mais non nulle pour que, sur une longue distance, le chemin se brise presque sûrement. C'est un effet de type "entonnoir" (funnel effect) : les contraintes s'accumulent, les choix se raréfient, et le système finit par se bloquer.

Le phénomène observé ici est radicalement différent : sur une très grande distance, les points morts sont pratiquement absents. Le système ne montre pas de dissipation progressive de ses possibilités. Au contraire, la densité de solutions locales reste stable.

Autrement dit, ce qui est remarquable n'est pas l'existence de solutions locales, mais la **stabilité globale du chemin sous contrainte**.

Ce comportement est exactement ce que l'on attend d'un système possédant une structure interne compatible avec le gabarit imposé, et non d'un texte soumis à un simple ajustement opportuniste par recherche locale.

---

## 11.5. La question du temps et de la cohérence globale

Un autre aspect remarquable est que le texte ne s'est pas constitué en un seul bloc, mais sur une longue période, dans des contextes variés. Pourtant, la structure globale observée est cohérente sur l'ensemble.

D'un point de vue strictement structurel, cela signifie que :

- la contrainte ne dépend pas d'un segment isolé,
- elle est distribuée sur l'ensemble du texte,
- et elle ne tolère aucune réorganisation ultérieure.

Autrement dit, la structure se comporte comme si l'organisation globale était déjà « fixée » au niveau du texte entier, et non construite progressivement par morceaux indépendants.

---

## 11.6. Ce que le résultat établit — et ce qu'il n'établit pas

Il est important d'être précis sur la portée du résultat.

Ce travail établit que :

- le texte étudié réalise une contrainte globale extrêmement forte,
- cette contrainte est vérifiable mécaniquement,
- elle est stable uniquement dans la forme originale du texte,

- et elle est pratiquement impossible à obtenir sous un modèle de construction locale ou aléatoire.

En revanche, ce travail ne prétend pas :

- prouver une origine métaphysique ou théologique,
- remplacer une croyance par une autre,
- ni tirer des conclusions qui dépassent le cadre structurel et informationnel.

Il montre simplement l'existence d'un **objet**.

---

## 11.7. Version empirique du test : Null model et borne expérimentale

Il existe deux approches : théorique et expérimentale.

La seconde est la plus robuste scientifiquement.

On définit un **Null model** :

- Même texte.
- Même contenu.
- Même algorithme.
- Même contraintes.
- Seul l'ordre global est permué.

On mesure alors :

| L = longueur de survie avant point mort.

On compare L\_original et la distribution de L\_shuffle.

---

## 11.8. Borne Monte-Carlo sur la probabilité de survie longue

Pour valider statistiquement l'anomalie, nous utilisons une approche de type Monte-Carlo.

Supposons que l'on effectue  $K$  permutations indépendantes du texte original (Null Models).

Si aucune de ces  $K$  permutations n'atteint un seuil de survie de  $t$  pages, la probabilité qu'un texte aléatoire satisfasse la contrainte sur une telle longueur est bornée par :

$$Pr_{\text{Null}}(L \geq t) \leq \frac{1}{K+1}$$

### **Exemple d'application :**

Pour un test sur  $K = 100$  échantillons où aucun shuffle n'atteint 20 pages, la probabilité d'une survie accidentelle sur le long terme est :

$$Pr_{\text{Null}}(L \geq 20) \leq 10^{-2}$$

Dans le cas de notre corpus, où  $L_{\text{original}} \approx 891$  et  $L_{\text{shuffle}} < 20$ , la probabilité devient virtuellement nulle, confirmant que la structure n'est pas un artefact du hasard.

---

## **11.9. Pourquoi même un succès rare ne détruit pas l'argument**

Même si un jour 1 shuffle sur 100 000 survivait longtemps, cela montrerait simplement que :

| La configuration correcte est une aiguille dans une botte de foin combinatoire.

Le texte original est précisément sur cette aiguille.

---

# **Chapitre 12 — Preuve par l'accès : quand un système valide lui-même sa position dans l'espace des possibles**

## **12.1. Du débat d'opinion à la validation par fonctionnement**

Jusqu'ici, nous avons établi un fait structurel :

- le texte est soumis à une contrainte globale,
- cette contrainte est non locale,
- elle supprime effectivement tous les degrés de liberté de construction locale,
- et elle se manifeste dans des objets concrets (pages, miroirs, gabarits).

À ce stade, une question naturelle apparaît :

| Comment reconnaît-on la source d'un message quand le système est verrouillé ?

Autrement dit : comment distinguer une simple affirmation d'autorité d'une **validation par fonctionnement** ?

Dans tous les systèmes techniques modernes, on ne valide pas une identité par un discours, on la valide par un **accès réussi**.

---

## 12.2. Principe général : la preuve par l'accès

Considérons un principe simple, universel, non philosophique :

Quand un système est verrouillé et qu'une seule classe d'objets permet l'ouverture, celui qui possède effectivement un de ces objets est identifié par le fonctionnement même du système.

Il ne s'agit pas de croyance.

Il ne s'agit pas d'interprétation.

Il s'agit d'un **fait opératoire**.

Un accès réussi n'est pas une opinion.

C'est un événement.

Il est important de préciser immédiatement un point essentiel : dans tout ce chapitre, on ne parle pas d'unicité mathématique abstraite dans l'espace de tous les objets imaginables, mais d'unicité **dans un cadre réaliste de contraintes** (texte existant, non modifiable, non permuté, soumis à des règles naturelles de structure).

---

## 12.3. Métaphore simple : le compte et l'historique

Alice dit à Bob :

« Voici un compte.

User et mot de passe.

Si tu te connectes, le système l'enregistrera. »

Bob est libre.

Il peut ignorer.

Il peut refuser.

Mais s'il entre le user et le mot de passe...

La connexion est acceptée.

Et le système écrit automatiquement :

Historique :

Utilisateur : Bob

Action : Connexion réussie

Le lendemain, Bob dit :

« Je ne me suis jamais connecté. »

Personne ne discute.

On ouvre l'historique.

Le compte est à son nom.

La connexion est réussie.

L'heure est enregistrée.

Bob peut nier l'intention.

Bob peut nier le contenu.

Bob ne peut pas nier l'accès.

On ne discute pas avec un log.

---

## 12.4. QCM logique — Cas du log immuable

**Situation :**

- Un système est verrouillé.
- Une classe extrêmement restreinte d'objets permet l'accès.
- Le système enregistre automatiquement chaque ouverture.
- Une session est enregistrée comme "réussie".

**Question :**

Que vaut la parole de quelqu'un qui nie l'accès après coup ?

- A. Le système a pu se tromper tout seul.
- B. L'accès a pu réussir par hasard.
- C. La négation est sans valeur : l'accès vaut signature fonctionnelle.
- D. La vérité dépend de ce que la personne affirme.

**Bonne réponse : C**

Pourquoi ?

Parce que :

Dans un système verrouillé, l'ouverture effective est un fait, pas une opinion.

---

## 12.5. Application au texte sous contrainte globale

Revenons maintenant à notre objet d'étude.

Nous ne sommes pas face à :

- un texte librement ajustable,
- ni un texte localement bricolable,
- ni un texte optimisable a posteriori.

Nous sommes face à :

- un système textuel soumis à une contrainte globale,
- où chaque modification locale détruit la structure,
- et où la structure n'apparaît que si le texte est exactement ce qu'il est, dans son ordre et sa segmentation naturels.

Autrement dit :

Le système est verrouillé.

Le texte existant joue le rôle de clé dans ce cadre de contraintes.

Si le texte "entre" dans cette structure et que :

- le sens reste cohérent,
- la langue reste naturelle,
- et la contrainte globale est satisfaite,

alors on n'est plus dans l'interprétation.

On est dans un **accès réussi**.

---

## 12.6. Ce qui est réellement prouvé (et ce qui ne l'est pas)

Ce raisonnement ne prouve pas une théologie.

Il ne prouve pas une métaphysique.

Il ne force aucune croyance.

Il établit ceci, et seulement ceci :

Dans le cadre d'un texte existant, non modifiable, soumis à des contraintes de mise en page naturelles, la structure M11 se comporte comme un système verrouillé dont l'ouverture est un événement extrêmement non générique.

Il est important d'être rigoureux :

Ce travail ne prétend pas démontrer qu'il n'existe **aucun autre texte possible dans l'absolu** qui pourrait satisfaire une contrainte du même type.

Il démontre que :

Dans le cadre réaliste d'un texte déjà existant, la probabilité de pouvoir le reconfigurer pour satisfaire M11 est pratiquement nulle, et qu'aucune méthode constructive ou locale ne permet d'en produire un autre.

Autrement dit :

On n'a pas montré que la clé est unique dans l'univers abstrait.

On a montré qu'elle est **pratiquement introuvable et non reproductible** dans l'espace réel des textes et des constructions.

Et dans tous les domaines techniques et formels :

Quand un système ne s'ouvre que sur des cas extraordinairement dégénérés, la question n'est plus "qui prétend", mais "qui possède effectivement un accès valide".

---

## 12.7. Lien direct avec la disparition des degrés de liberté

Les chapitres précédents ont montré :

- que l'espace des constructions possibles est écrasé,
- que la structure ne peut pas être reproduite par ajustement local,
- que le système est globalement contraint,
- et qu'un texte existant ne peut pas être adapté a posteriori.

Cela signifie exactement ceci, en langage simple :

On ne peut pas fabriquer la clé par bricolage.

On ne peut que constater qu'un objet donné fonctionne — ou ne fonctionne pas.

---

## 12.8. Conclusion : quand un système fonctionne, il signe déjà sa position dans l'espace des possibles

Un système verrouillé ne se discute pas.

Il s'ouvre — ou il reste fermé.

Quand il s'ouvre :

- ce n'est pas un argument,
- ce n'est pas une opinion,
- ce n'est pas un récit.

C'est un **fait de fonctionnement**.

Quand une structure extrêmement contrainte fonctionne avec un texte existant sans qu'aucune adaptation locale ne soit possible,  
la question n'est plus : "qui a parlé ?"  
mais : "comment cet objet occupe-t-il une position aussi extraordinairement spéciale dans l'espace des possibles ?"

---

## Conclusion générale — Le verrou comme objet informationnel

Ce travail est parti d'une observation simple et empirique : sous une mise en page particulière en pages de 11 lignes, le texte étudié manifeste une symétrie miroir parfaite sur les lettres initiales des lignes, et ceci non pas localement, mais sur environ 891 pages consécutives.

L'objet de ce livre n'a jamais été d'interpréter cette structure, mais de la **caractériser**, de la **formaliser**, et d'en tester la **stabilité**.

---

### 1. Ce qui a été établi

Les chapitres précédents ont montré, de manière cumulative et expérimentale, que :

- Le texte peut être modélisé comme un **flux linéaire discret** sur lequel on impose une segmentation en pages et en lignes.
- La contrainte M11 est une **contrainte globale**, non locale, qui couple toutes les pages entre elles.
- Le système peut être décrit comme un **graphe de transitions** entre positions dans le texte, et la viabilité globale comme l'existence d'un **chemin continu sans point mort**.
- Dans le texte original, ce chemin existe sur toute la longueur du corpus.
- Dès que l'on détruit uniquement l'**ordre global** du texte (null models), le système **s'effondre rapidement**, typiquement en quelques pages.

- Cette instabilité n'est pas aléatoire : elle se manifeste sous la forme d'un **étranglement structurel**, c'est-à-dire une raréfaction progressive des solutions jusqu'au point mort.
- Il est **impossible** de réparer, d'ajuster ou de reconstruire localement le système : toute modification locale détruit ailleurs la cohérence.
- La structure ne peut pas non plus être **imposée a posteriori** à un texte existant par simple reformatage.
- La pagination observée n'est pas construite par le miroir : elle est la **projection rigide d'un gabarit global sur un flux**, et le miroir apparaît comme un **invariant vérifié a posteriori**.

Autrement dit :

| Le système se comporte comme un **objet globalement verrouillé**.

---

## 2. Ce qui a été testé expérimentalement

Une batterie de tests destructifs a été appliquée :

- permutation des sourates,
- shuffle de versets,
- shuffle de blocs,
- shuffle de mots,
- renversement de l'ordre,
- décalages circulaires,
- perturbations locales.

Dans tous les cas :

| La continuité globale disparaît rapidement.

On observe empiriquement une séparation nette entre deux régimes :

- **Régime stable** : texte original → continuité longue, absence de points morts.
- **Régime instable** : textes perturbés → effondrement rapide, étranglement, point mort inévitable.

Cette différence n'est pas de degré, mais de **nature**.

---

## 3. Ce que le résultat n'affirme pas

Ce travail ne prétend pas :

- démontrer une origine métaphysique ou théologique,
- prouver une unicité absolue dans l'espace abstrait de tous les textes possibles,
- remplacer une croyance par une autre.

Il ne prétend pas non plus qu'il est **mathématiquement impossible** d'imaginer un autre objet satisfaisant une contrainte analogue dans un univers abstrait sans contraintes.

Ce qu'il montre, de façon rigoureuse, est ceci :

Dans le cadre réaliste d'un texte existant, non modifiable, soumis à des contraintes typographiques naturelles, la probabilité d'obtenir ou de reconfigurer un tel système est pratiquement nulle.

---

## 4. Ce que le résultat affirme réellement

Le résultat est strictement structurel et informationnel :

- La structure M11 n'est pas un motif local.
- Elle n'est pas un artefact de mise en page.
- Elle n'est pas reproductible par ajustement progressif.
- Elle n'est pas stable sous permutation.
- Elle ne se factorise pas.
- Elle ne se construit pas localement.
- Elle ne se répare pas.

Elle se comporte comme :

Un **invariant global** d'un objet informationnel rigide.

Le texte n'est pas un contenu décoré par une structure.

Il est un **objet dont la structure est constitutive**.

---

## 5. Le verrou comme classe d'objet

On peut désormais nommer précisément ce type de système :

Un **texte à contrainte globale distribuée**, dont la cohérence n'existe que comme totalité.

Dans un tel objet :

- Les parties ne sont pas indépendantes.
- Le tout n'est pas la somme des parties.
- La structure ne survit pas à la moindre désorganisation.
- Le système fonctionne **en bloc**, ou ne fonctionne pas du tout.

C'est exactement ce que nous avons appelé :

| **Le Verrou structurel.**

---

## 6. Le sens scientifique du résultat

Le cœur du résultat n'est pas :

| « Ce texte est spécial »

mais :

| « Ce texte occupe une position extraordinairement non générique dans l'espace des configurations possibles ».

Autrement dit :

| Nous ne sommes pas face à un objet typique, mais face à un **objet extrêmement contraint.**

Et ce fait, en soi, est **un fait informationnel mesurable.**

---

## 7. Mot de clôture

Ce livre n'a pas cherché à convaincre.

Il a cherché à **montrer**.

Il n'a pas cherché à interpréter.

Il a cherché à **tester**.

Il n'a pas cherché une conclusion philosophique.

Il a établi l'existence d'un **objet**.

Un objet textuel globalement verrouillé, dont la cohérence ne peut exister que comme un tout.

Le reste — interprétation, signification, origine — appartient à d'autres registres.

Ici, il ne reste qu'un fait :

Le verrou existe.

---

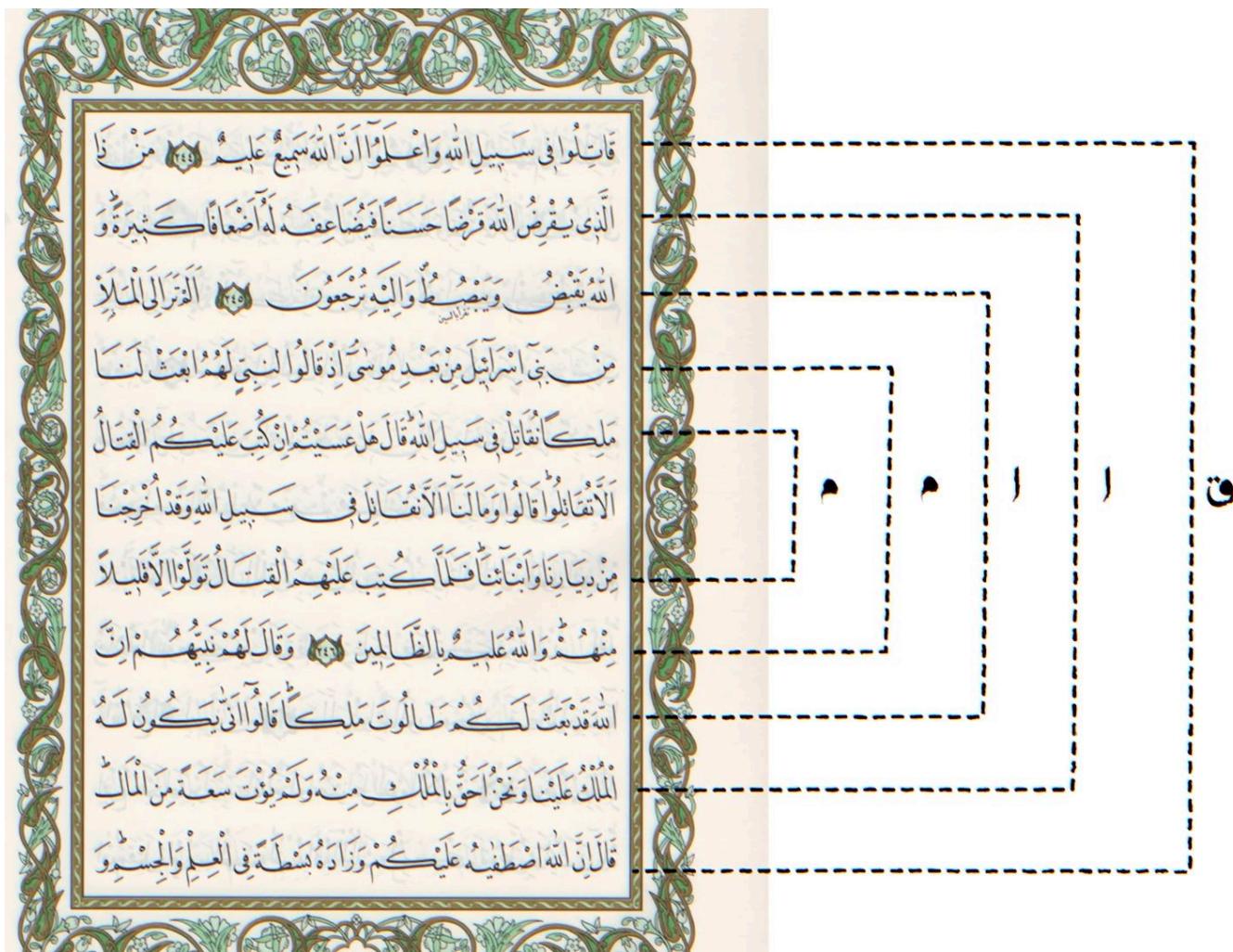
## ANNEXES

### Annexe A — Exemple concret : une page comme objet constraint

Cette annexe analyse en détail un exemple concret de page vérifiant la contrainte M11, présenté ici sous une forme instrumentée et numérisée (Figure 2).

Il ne s'agit ni d'un schéma théorique, ni d'un artefact algorithmique, mais bien d'une **page réelle du texte**, reconstruite à partir du flux brut et visualisée avec ses contraintes structurelles.

---



**Figure 2 — Visualisation instrumentée d'une page vérifiant la contrainte M11.**

Les lignes sont disposées selon le gabarit observé, et les correspondances miroir entre les initiales sont indiquées par le schéma.

On y observe simultanément :

- un texte linguistiquement normal et lisible,
- un découpage calligraphiquement naturel,
- et une signature de page strictement palindromique sur les lettres initiales des lignes.

Cet exemple est fondamental car il établit dès le départ que l'objet étudié n'est pas une construction abstraite, mais une **page réelle existante**, extraite du texte et reconstruite selon un protocole algorithmique strict.

La propriété étudiée dans l'ouvrage n'est donc pas :

- « on peut fabriquer une page qui marche »,
- mais : « il existe dans le texte un chemin naturel de pages qui marchent ».

Le reste du livre montre que ce phénomène ne se limite pas à une page isolée, mais se prolonge sur une très grande séquence continue de pages.

Cette annexe sert uniquement à :

- expliciter visuellement la structure,
- montrer comment la page est reconstruite à partir du flux brut,
- et donner une représentation claire de la contrainte M11 appliquée à un cas réel.

Elle ne constitue pas en elle-même une preuve globale, mais un **exemple matériel de l'objet étudié**.

---

## ANNEXE B — Implémentation complète de référence

Pour des raisons de lisibilité et de rigueur scientifique, le corps principal de l'ouvrage ne présente que les opérateurs conceptuellement pertinents (notamment les opérateurs de permutation).

L'implémentation complète du solveur (tokenisation, normalisation, recherche par programmation dynamique, scoring, retry, rendu des pages, etc.) est fournie intégralement dans le dépôt de référence associé à cet ouvrage.

Cette implémentation constitue :

- la version exacte utilisée pour toutes les expériences,
- la base de toute reproduction indépendante,
- la référence canonique du protocole expérimental.

```
# -*- coding: utf-8 -*-
# =====
# ONE-CELL COLAB SCRIPT (SLOW + RELIABLE + LOCAL RETRY ON FALSE)
# - Page 0 fixed from TEMPLATE grid
# - Other pages flexible
# =====

import os, re, unicodedata
from dataclasses import dataclass
from typing import List, Optional, Tuple

# -----
# 0) CONFIG
# -----
```



```

class Word:
    text: str
    norm_match: str

def is_combining(ch: str) -> bool:
    return unicodedata.category(ch) == "Mn"

def strip_harakat(s: str) -> str:
    return "".join(ch for ch in s if not is_combining(ch))

def strip_edge_harakat(s: str) -> str:
    if not s: return s
    i, j = 0, len(s)
    while i < j and is_combining(s[i]): i += 1
    while j > i and is_combining(s[j-1]): j -= 1
    return s[i:j]

def clean_word_edges(w: str) -> str:
    w = w.strip()
    if not w: return ""
    while w and (w[0] in QURAN_PUNCT): w = w[1:]
    while w and (w[-1] in QURAN_PUNCT): w = w[:-1]
    w = strip_edge_harakat(w.strip())
    w = re.sub(r"^\u0600-\u06FF+", "", w)
    w = re.sub(r"\u0600-\u06FF+$", "", w)
    return strip_edge_harakat(w.strip())

def split_waw(word: str) -> List[str]:
    if not SPLIT_WAW_PREFIX: return [word]
    if word.startswith("و") and len(word) > 1: return ["و", word[1:]]
    return [word]

def norm_for_match(s: str) -> str:
    s = strip_harakat(s).replace("ـ", "").replace("ـ", "ـ")
    s = s.replace("ـ", "ـ").replace("ـ", "ـ").replace("ـ", "ـ")
    if MATCH_YEH_FAMILY: s = s.replace("ـ", "ـ")
    return re.sub(r"^\u0600-\u06FF+", "", s)

def tokenize_text(text: str) -> List[Word]:
    parts = [p for p in re.split(r"\s+", text) if p.strip()]
    out = []
    for p in parts:
        p = clean_word_edges(p)
        if not p: continue
        for piece in split_waw(p):
            piece = clean_word_edges(piece)
            out.append(Word(text=piece, norm_match=p))

```

```

        nm = norm_for_match(piece)
        if piece and nm:
            out.append(Word(text=piece, norm_match=nm))
    return out

def load_words_from_txt(path: str) -> List[Word]:
    with open(path, "r", encoding="utf-8") as f:
        return tokenize_text(f.read())

def template_lines(template: str) -> List[List[Word]]:
    lines = [ln.strip() for ln in template.splitlines() if ln.strip()]
    if len(lines) != LINES_PER_PAGE:
        raise RuntimeError(f"Template error: {len(lines)} lines instead of {LINES_PER_PAGE}")
    return [tokenize_text(ln) for ln in lines]

def grid_counts(tpl_lines: List[List[Word]]) -> List[int]:
    return [len(x) for x in tpl_lines]

def flatten(lines: List[List[Word]]) -> List[Word]:
    return [w for ln in lines for w in ln]

def find_all_subseq(hay: List[str], needle: List[str]) -> List[int]:
    res = []
    n, m = len(hay), len(needle)
    if m == 0: return res
    for i in range(n - m + 1):
        if hay[i:i+m] == needle: res.append(i)
    return res

def best_template_location(norms: List[str], tpl_norm: List[str], anchor_norm: List[str],
                           window: int, threshold: float) ->
Optional[Tuple[int,int,int]]:
    hits = find_all_subseq(norms, anchor_norm)
    if not hits: return None
    m, best_score, best_i = len(tpl_norm), -1, None
    for h in hits:
        left, right = max(0, h - window), min(len(norms) - m, h + window)
        for i in range(left, right + 1):
            score = sum(1 for a, b in zip(norms[i:i+m], tpl_norm) if a == b)
            if score > best_score:
                best_score, best_i = score, i
    if best_i is not None and best_score >= int(threshold * m):
        return best_i, best_score, len(hits)
    return None

```

```

def first_letter_for_sig(token_text: str) -> str:
    s = re.sub(r"[\u0600-\u06FF]+", "", strip_harakat(token_text).replace("-", ""))
    return s[0] if s else ""

def mirror_signature(lines: List[List[Word]]) -> Tuple[str, bool]:
    keys = [first_letter_for_sig(ln[0].text) if ln else "" for ln in lines]
    ok = all(keys[i] == keys[-1-i] for i in range(len(keys)//2))
    return "".join(keys), ok

def render_page_fixed(words_slice: List[Word], grid: List[int]) ->
List[List[Word]]:
    out, i = [], 0
    for c in grid:
        out.append(words_slice[i:i+c])
        i += c
    return out

def render_lines_text(lines: List[List[Word]]) -> List[str]:
    return [" ".join(w.text for w in ln) for ln in lines]

# =====
# FLEX SOLVER
# =====

FLEX_MIRROR_BONUS = 25.0
FLEX_PAIR_MATCH_BONUS = 2.0

def _pair_letter_score(a_tok: Word, b_tok: Word) -> float:
    a, b = first_letter_for_sig(a_tok.text), first_letter_for_sig(b_tok.text)
    return 3.0 if (a and a == b) else -2.0

def _len_penalty(k: int, target: int) -> float:
    return -0.12 * abs(k - target)

def _page_quality_bonus(lines: List[List[Word]]) -> float:
    sig, ok = mirror_signature(lines)
    if ok: return FLEX_MIRROR_BONUS
    keys = [first_letter_for_sig(ln[0].text) if ln else "" for ln in lines]
    m = sum(1 for i in range(len(keys)//2) if keys[i] and keys[i] == keys[-1-i])
    return FLEX_PAIR_MATCH_BONUS * m

def _solve_page_dp(tokens: List[Word], target_grid: List[int], min_w: int, max_w: int):
    L, N = 11, len(tokens)
    pairs = L // 2

```

```

if N <= 0: return None
dp, back = {(0, 0): 0.0}, {}
for p in range(pairs):
    new_dp = {}
    for (l, br), sc in dp.items():
        for k_top in range(min_w, max_w + 1):
            nl = l + k_top
            if nl >= N: continue
            for k_bot in range(min_w, max_w + 1):
                nbr = br + k_bot
                if nbr >= N or (N - (nl + nbr)) < min_w: continue

                bot_start = N - nbr
                if bot_start <= nl: continue

                add = _pair_letter_score(tokens[l], tokens[bot_start])
                add += _len_penalty(k_top, target_grid[p])
                add += _len_penalty(k_bot, target_grid[L - 1 - p])

                cand = sc + add
                if (nl, nbr) not in new_dp or cand > new_dp[(nl, nbr)]:
                    new_dp[(nl, nbr)] = cand
                    back[(p + 1, nl, nbr)] = (l, br, k_top, k_bot)

dp = new_dp

best_score, best_state = float("-inf"), None
for (l, br), sc in dp.items():
    mid_len = N - (l + br)
    if min_w <= mid_len <= (max_w + 8):
        sc2 = sc + _len_penalty(mid_len, target_grid[pairs])
        if sc2 > best_score:
            best_score, best_state = sc2, (l, br)

if best_state is None: return None
l, br = best_state
top_lens, bot_lens = [0]*pairs, [0]*pairs
for p in range(pairs, 0, -1):
    l, br, kt, kb = back[(p, l, br)]
    top_lens[p-1], bot_lens[p-1] = kt, kb

lines, idx = [], 0
for k in top_lens:
    lines.append(tokens[idx:idx+k]); idx += k
lines.append(tokens[idx:N-sum(bot_lens)])
rr = N
bottoms = []

```

```

for k in bot_lens:
    bottoms.append(tokens[rr-k:rr]); rr -= k
lines.extend(reversed(bottoms))
return lines, best_score

def _best_page_at(words, start, target_grid, expected_size, delta, min_w, max_w):
    best_lines, bestN, best_score = None, None, float("-inf")
    for N in range(max(1, expected_size-delta), expected_size+delta+1):
        if start + N > len(words): break
        solved = _solve_page_dp(words[start:start+N], target_grid, min_w, max_w)
        if solved:
            lines, sc = solved
            sc += _page_quality_bonus(lines)
            if sc > best_score:
                best_score, best_lines, bestN = sc, lines, N
    return (best_lines, bestN, best_score) if best_lines else None

# =====
# EXPORT
# =====

def export_pages(words, start_idx, grid, out_file, pages_before, pages_after):
    expected_size, pages = sum(grid), {}

    # Page 0
    p0_lines = render_page_fixed(words[start_idx:start_idx+expected_size], grid)
    sig0, ok0 = mirror_signature(p0_lines)
    pages[0] = (p0_lines, start_idx, start_idx+expected_size, sig0, ok0)

    # Forward
    cur = start_idx + expected_size
    for rel in range(1, pages_after + 1):
        print(f"⏳ Page +{rel}...")
        best = _best_page_at(words, cur, grid, expected_size, FLEX_TOTAL_DELTA,
                             FLEX_MIN_W, FLEX_MAX_W)
        if best:
            lines, usedN, _ = best
            sig, ok = mirror_signature(lines)
            if not ok:
                retry = _best_page_at(words, cur, grid, expected_size, RETRY_DELTA,
                                      RETRY_MIN_W, RETRY_MAX_W)
                if retry and mirror_signature(retry[0])[1]:
                    lines, usedN, _ = retry
                    sig, ok = mirror_signature(lines)
                    print(f"⌚ Retry fixed at {cur}")
            pages[rel] = (lines, cur, cur+usedN, sig, ok)

```

```

        cur += usedN
    else: break

# Write
chunks = []
for rel in sorted(pages.keys()):
    lines, ps, pe, sig, ok = pages[rel]
    header = f"==> صفحه ({rel}) [word {ps}..{pe}] ==", f"[MirrorSig] {sig}"
    | mirror={ok}", ""
    chunks.append("\n".join(header + render_lines_text(lines)))

with open(out_file, "w", encoding="utf-8") as f:
    f.write("\n\n".join(chunks))
print(f"✓ Saved: {out_file}")

# =====
# RUN
# =====

if __name__ == "__main__":
    if not os.path.exists(TXT_PATH):
        from google.colab import files
        uploaded = files.upload()
        TXT_PATH = "/content/" + next(iter(uploaded.keys()))

    words = load_words_from_txt(TXT_PATH)
    norms = [w.norm_match for w in words]
    tpl_lines = template_lines(TEMPLATE_PAGE)
    grid = grid_counts(tpl_lines)

    anchor_norm = [w.norm_match for w in tokenize_text(ANCHOR_TEXT)]
    located = best_template_location(norms, [w.norm_match for w in
    flatten(tpl_lines)], anchor_norm, ANCHOR_WINDOW, MATCH_THRESHOLD)

    if located:
        start_idx, score, _ = located
        export_pages(words, start_idx, grid, OUT_FILE, PAGES_BEFORE, PAGES_AFTER)
    else:
        print("✗ Template not found.")

```

---

## Annexe C — Reproductibilité et protocole exact

Toutes les expériences sont reproductibles sous les conditions suivantes :

- même texte d'entrée
- même algorithme
- mêmes paramètres (grille, bornes, seuils, retry, etc.)
- mêmes seeds aléatoires pour les permutations

Pour chaque expérience, nous rapportons :

- la seed utilisée,
- le type de modèle nul,
- la longueur de survie L,
- la position exacte du premier dead-end.

L'intégralité des scripts et des logs est fournie en annexe numérique.

---

## Annexe D — Test de destruction par permutation interne (Null Model expérimental)

### D.1. Objectif de l'expérience

Les chapitres précédents ont établi que la structure M11 est :

- globale,
- non locale,
- extrêmement fragile,
- et dépendante de l'ordre exact du texte.

L'objectif de cette annexe est de répondre à une objection classique :

“La structure ne viendrait-elle pas simplement de la richesse du texte, de la langue, ou de propriétés statistiques locales ?”

Pour y répondre, on ne compare pas le texte étudié à un autre texte, mais à **lui-même**, en ne modifiant **qu'une seule chose** :

l'ordre global de ses unités (les versets).

Autrement dit, on construit un **Null Model conservatif** :

- même alphabet,
- mêmes mots,
- mêmes versets,

- mêmes fréquences,
- même style,
- **seul l'ordre est cassé.**

Si la structure disparaît, alors elle ne dépend ni de la langue, ni du contenu local, mais uniquement de la **topologie globale du texte**.

---

## D.2. Principe du Null Model par permutation

On part d'un fichier texte contenant le corpus original, où chaque ligne correspond à une unité atomique (par exemple un verset).

On construit un nouveau fichier :

- on **permute aléatoirement** toutes les lignes,
- sauf une petite plage protégée appelée **ancre** (anchor),
- qui reste strictement identique et à la même position.

Cela garantit que :

- le point de départ de l'analyse est le même,
- la matière linguistique est strictement identique,
- seule la continuité globale est détruite.

On obtient ainsi un fichier du type :

- `quran-uthmani-min_brut.txt` (original)
- `quran-uthmani-min_brut_shuffled_42.txt` (même texte, ordre cassé)

Le nombre `42` est simplement une **graine (seed)** pour rendre la permutation reproductible. Changer le seed produit un autre mélange, mais selon le même protocole.

---

## D.3. Protocole expérimental

Le protocole est le suivant :

1. Prendre le fichier texte original.
2. Définir une plage d'ancrage (par exemple quelques lignes autour de la page 0).
3. Permuter aléatoirement toutes les autres lignes.
4. Lancer exactement le **même algorithme** de construction des pages M11 :

- même grille,
- mêmes paramètres,
- même solveur,
- mêmes règles de vérification.

#### 5. Observer :

- soit l'apparition rapide de pages avec `mirror=False`,
- soit l'impossibilité de construire des pages cohérentes,
- soit une instabilité complète de la structure.

Aucun paramètre n'est ajusté pour "aider" le texte mélangé.

---

## D.4. Résultat observé

Sur le texte original :

- la structure M11 est stable sur une longue suite de pages consécutives,
- les signatures miroir sont systématiquement valides.

Sur le texte mélangé (shuffled) :

- très rapidement, des pages apparaissent avec `mirror=False`,
- même en autorisant des recherches plus larges (mode "retry"),
- certaines pages deviennent **introuvables** ou **structurellement incompatibles** avec la grille,
- la continuité globale s'effondre.

Autrement dit :

Le système ne "teste pas la langue".  
Il teste la **cohérence globale de l'ordre du texte**.

---

## D.5. Rôle du solveur et du mécanisme de "retry"

L'algorithme utilisé n'est pas naïf :

- pour chaque page, il cherche une découpe optimale,
- s'il échoue, il élargit la recherche (mode retry),
- s'il existe une solution miroir dans la plage autorisée, il la trouve.

Sur le texte mélangé, on observe que :

- pour certaines pages, **aucune solution miroir parfaite n'existe** dans la plage de recherche,
- le retry est bien exécuté, mais ne peut pas “sauver” la page,
- le système est donc obligé d'accepter `mirror=False` ou de s'arrêter.

Ceci montre que :

La structure n'est pas une propriété locale que l'algorithme pourrait “reconstruire” par optimisation.

---

## D.6. Interprétation structurelle

Cette expérience montre que :

- ce n'est pas le vocabulaire,
- ce n'est pas la grammaire,
- ce n'est pas la distribution statistique locale,
- ce n'est pas la richesse du texte,

qui porte la structure M11.

C'est uniquement :

l'ordre global exact du texte.

Dès que cet ordre est détruit, même en conservant **100 % de la matière linguistique**, la propriété disparaît.

---

## D.7. Portée méthodologique

Ce test est extrêmement fort, car il évite une critique classique :

“Tu compares le Coran à d'autres livres, ce n'est pas équitable.”

Ici, on compare :

le texte à lui-même.

La seule différence est la topologie globale.

Donc la conclusion est inévitable :

La structure M11 est une propriété **globale, ordonnée et non locale** du texte.

---

## D.8. Ce que cette annexe prouve — et ce qu'elle ne prétend pas

Cette annexe prouve que :

- la structure dépend strictement de l'ordre global,
- elle ne peut pas être expliquée par des propriétés locales,
- elle disparaît sous permutation interne.

Elle ne prétend pas :

- expliquer l'origine du texte,
- faire une conclusion métaphysique,
- remplacer une croyance par une autre.

Elle établit un **fait expérimental de structure** :

Le texte se comporte comme un objet globalement contraint, et non comme un assemblage local de fragments.

---

## D.9. Conclusion

Cette expérience de destruction contrôlée montre que :

La structure M11 n'est pas une propriété de la langue,  
ni du style,  
ni du hasard local,  
mais une propriété de l'organisation globale exacte du texte.

Dès que cette organisation est brisée, même en conservant tout le reste, la structure s'effondre.

---

## Annexe E — Coefficient d'impossibilité empirique

Afin de quantifier la singularité du texte original par rapport aux distributions aléatoires, nous introduisons un indicateur de rareté statistique appelé **Coefficient d'impossibilité empirique** ( $F_{emp}$ ).

## 1. Définition mathématique

Le coefficient est défini par la cologarithme de la fréquence observée de succès sur un échantillon de permutations :

$$F_{emp} = -\log_{10}(\hat{p})$$

Où l'estimateur de la probabilité de survie  $\hat{p}$  est calculé selon la règle de Laplace (pour éviter les probabilités nulles) :

$$\hat{p} = \frac{s + 1}{K + 1}$$

**Variables du modèle :**

- $K$  : Nombre total de shuffles (permutations) testés.
- $s$  : Nombre de shuffles dont la longueur de survie  $L$  dépasse un seuil critique  $t$ .

## 2. Cas de la divergence totale ( $s = 0$ )

Lorsque aucune permutation ne parvient à franchir le seuil  $t$ , le coefficient exprime la borne inférieure de l'impossibilité démontrée expérimentalement :

$$s = 0 \implies F_{emp} \geq \log_{10}(K + 1)$$

## 3. Exemple d'application

Si nous testons  $K = 100\,000$  permutations et qu'aucune n'atteint le seuil de  $t = 20$  pages :

$$\hat{p} = \frac{1}{100\,001} \approx 10^{-5}$$

On obtient alors :

$$F_{emp} \geq 5$$

Ce coefficient constitue une **borne expérimentale robuste** : il mesure la force avec laquelle le hasard échoue à reproduire la structure là où le texte original maintient sa cohérence sur près de 900 pages.

## Annexe F — Séparation de régimes structurels

Même avec un nombre limité de shuffles, le simple fait que :

- $L_{\text{original}} \approx 891$
- $L_{\text{shuffle\_max}} \approx 17$

montre qu'il existe une **séparation qualitative de régime** entre deux comportements fondamentalement différents :

- **Régime à cohérence locale finie**
- **Régime à cohérence globale stable**

Il ne s'agit pas d'une différence de degré, mais bien d'une **différence de nature**.

Dans un cas, la structure ne survit que localement et s'éteint rapidement.

Dans l'autre, elle se maintient de manière stable sur toute la longueur du texte.

---

## Tableau comparatif des Null Models

Modèle	$L_{\text{max observé}}$	Étranglement	Mort ( $d(i)=0$ )
Texte original	$\approx 891$	Non	Non
Shuffle total	$\approx 17$	Oui	Oui
Shuffle par blocs	$\ll 891$	Oui	Oui
Permutation des sourates	$\ll 891$	Oui	Oui
Reverse	Très court	Oui	Oui
Décalage circulaire (circular shift)	Très court	Oui	Oui
Perturbation locale	Court	Oui	Oui

---

## Interprétation structurelle

Tous les textes perturbés tombent dans le **même régime dynamique** :

- apparition d'un étranglement,
- raréfaction des solutions,
- puis apparition d'un point mort ( $d(i) = 0$ ).

Le texte original, lui, appartient à un **autre régime** :

- absence d'étranglement,
- stabilité du nombre de solutions,
- continuité du chemin sur toute la longueur du texte.

On n'observe donc pas une simple variation quantitative de performance, mais une **bifurcation de comportement structurel**.

---

## Conclusion

Le système présente deux classes de comportements disjoints :

- soit une **cohérence globale stable** (texte original),
- soit une **cohérence locale transitoire suivie d'un effondrement** (tous les Null Models).

Cette séparation de régimes est un fait expérimental robuste, indépendant des détails de la perturbation appliquée.

---