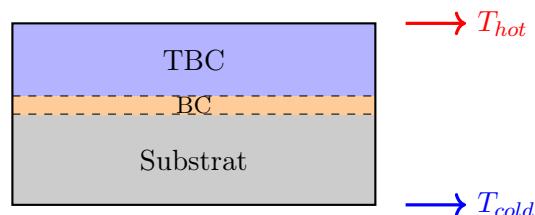


# Rapport de Synthèse

Modélisation Thermo-Mécanique  
des Systèmes Barrière Thermique (TBC)

Méthode Spectrale et Implémentation Python/Streamlit

Projet Industriel 5A-IDSA



**Projet :** Industriel 5A-IDSA  
**Encadrement :** ONERA (A. Vattré)  
**Partenaires :** ESTACA / Safran  
**Date :** 24 janvier 2026

**Mots-clés :** Barrière thermique, Méthode spectrale, Multicouche,  
Analyse modale, Critères d'endommagement, Streamlit

## Table des matières

<b>Résumé Exécutif</b>	<b>2</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Contexte Industriel : Aubes de Turbines Nouvelle Génération . . . . .	3
1.2 Objectifs du Projet . . . . .	3
<b>2 Système Multicouche Étudié</b>	<b>3</b>
2.1 Paramètres Adimensionnels Clés . . . . .	3
2.2 Propriétés Matériaux . . . . .	4
<b>3 Modélisation Mathématique</b>	<b>4</b>
3.1 Étape 1-2 : Représentation Spectrale de la Température . . . . .	4
3.2 Étape 3 : Solution Thermique par Couche . . . . .	4
3.3 Étape 4 : Loi de Comportement Thermo-Élastique . . . . .	4
3.4 Étape 5 : Ansatz de Déplacement . . . . .	5
3.5 Étape 6 : Matrice Dynamique $\Gamma(\tau)$ et Valeurs Propres . . . . .	5
3.6 Étape 7 : Assemblage $9 \times 9$ et Termes Thermiques . . . . .	6
3.7 Étape 8 : Assemblage Multicouche . . . . .	6
<b>4 Critères d'Endommagement</b>	<b>7</b>
<b>5 Interprétation Physique des Résultats</b>	<b>8</b>
5.1 Signification des Contraintes Transverses . . . . .	8
5.2 Zones Critiques Typiques . . . . .	8
5.3 Influence des Paramètres sur les Contraintes . . . . .	8
5.4 Modes de Rupture aux Interfaces . . . . .	8
<b>6 Stabilité Numérique</b>	<b>8</b>
6.1 Préconditionnement par Équilibrage (Scaling) . . . . .	9
6.2 Régularisation de Tikhonov . . . . .	9
<b>7 Comparaison : Méthode Spectrale vs CLT</b>	<b>9</b>
<b>8 Validation : Référence ONERA/Safran</b>	<b>10</b>
<b>9 Architecture du Code et Interface</b>	<b>10</b>
9.1 Structure des Répertoires . . . . .	10
9.2 Tableau de Correspondance Théorie $\leftrightarrow$ Code . . . . .	11
9.3 Interface Streamlit . . . . .	11
<b>10 Conclusion</b>	<b>11</b>

## Résumé Exécutif

Projet 5A-IDSA : Évaluation Thermomécanique des Aubes de Turbines

**Encadrement :** ONERA (Aurélien Vattré) / ESTACA (Daniel Gaffié)

**Objectif :** Développer un outil de simulation numérique pour l'évaluation thermomécanique des zones critiques d'endommagement dans les aubes de turbines multicouches nouvelle génération.

**Contexte Industriel :** Les aubes de turbines sont soumises à des conditions extrêmes : températures dépassant 1000°C, gradients thermiques sévères (jusqu'à 200°C sur quelques millimètres), et chargements mécaniques cycliques. Des architectures multicouches combinant substrat métallique, couche d'accroche et revêtement céramique sont utilisées pour améliorer leur durabilité.

**Méthodologie Développée :**

- Approche **semi-analytique** basée sur la décomposition spectrale (séries de Fourier doubles)
- Construction de la matrice dynamique  $\Gamma(\tau)$  avec opérateurs  $L_{jk}$
- Résolution du problème aux valeurs propres  $\det(\Gamma(\tau)) = 0$
- Assemblage matriciel bloc-diagonal  $9 \times 9$  avec sollicitation thermique
- Critères d'endommagement D et Tsai-Wu pour identification des zones critiques

**Résultats Clés :**

Métrique	Plage Typique	Seuil Critique
Température interface substrat/bondcoat	800–1050°C	$T_{crit} = 1100°C$
Indicateur d'endommagement D	0.2–0.8	$D \geq 1$ (rupture)
Contrainte normale $\sigma_{33}$ max	50–200 MPa	$\sigma_t^{ceramic} = 150$ MPa
Contrainte cisaillement $\sigma_{13}$ max	20–80 MPa	$\tau_{crit} = 120$ MPa

**Livrables du Projet :**

1. **Module de calcul :** `core/mechanical_pdf.py`
2. **Interface Streamlit :** Application interactive avec dashboard et visualisations 3D
3. **Documentation technique :** Ce rapport de synthèse

# 1 Introduction

## 1.1 Contexte Industriel : Aubes de Turbines Nouvelle Génération

Les aubes de turbines constituent l'un des composants les plus critiques des moteurs aéronautiques. Situées en aval de la chambre de combustion, elles sont exposées à des gaz à très haute température (supérieure à 1500°C) tout en devant supporter des contraintes mécaniques élevées dues à la rotation.

### Enjeux de Durabilité

Les phénomènes d'endommagement aux interfaces entre couches représentent la principale cause de défaillance des systèmes TBC :

- **Délamination** : Décohésion entre la couche céramique et le substrat
- **Écaillage** : Perte de morceaux de revêtement céramique
- **Fissuration** : Propagation de fissures dans les zones de concentration de contraintes

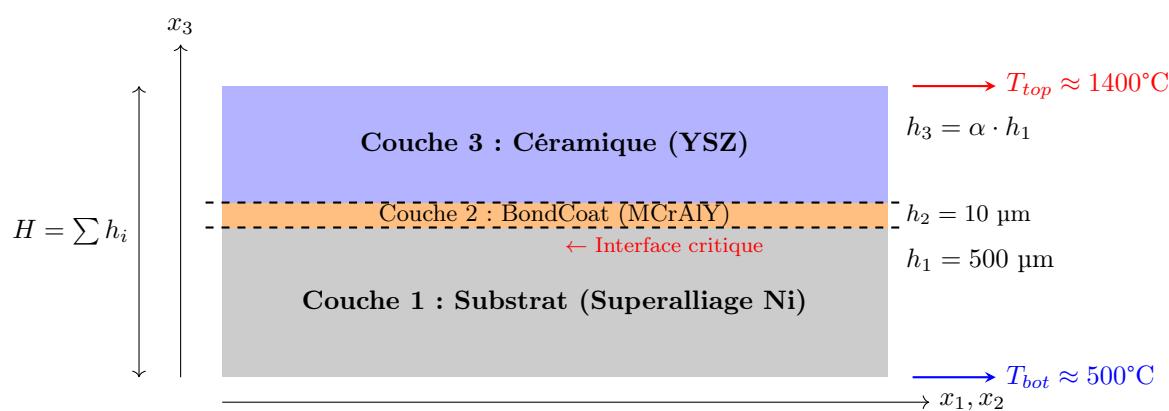
## 1.2 Objectifs du Projet

Ce projet vise à :

1. **Modéliser** la réponse thermomécanique d'architectures multicouches à partir d'un modèle tridimensionnel simplifié (méthode spectrale)
2. **Prédire** les zones critiques d'endommagement situées aux interfaces
3. **Quantifier** les effets de l'anisotropie élastique, des contrastes thermiques, et de l'épaisseur des couches
4. **Fournir** une carte de sensibilité pour guider la conception de futures aubes plus robustes

# 2 Système Multicouche Étudié

Le système TBC (Thermal Barrier Coating) comprend  $N$  couches empilées selon la direction normale  $x_3$  :



## 2.1 Paramètres Adimensionnels Clés

Les paramètres d'entrée de l'application sont :

$$\alpha = \frac{h_3}{h_1} \quad (\text{ratio épaisseur céramique/substrat, typiquement } 0.1 \text{ à } 1.0) \quad (1)$$

$$\beta = \frac{k_3}{k_1} \quad (\text{ratio conductivité thermique}) \quad (2)$$

$$L_w \quad (\text{longueur d'onde de la perturbation latérale, en mètres}) \quad (3)$$

$$\delta_1 = \delta_2 = \frac{\pi}{L_w} \quad (\text{nombres d'onde spectraux}) \quad (4)$$

## 2.2 Propriétés Matériaux

Les propriétés élastiques et thermiques de chaque couche sont définies dans le module `core/constants.py`. Les valeurs sont issues des publications ONERA/Safran (voir Section 8).

Propriété	Substrat (Ni)	BondCoat	Céramique (YSZ)
$C_{11}$ (GPa)	259.6	180	50
$C_{12}$ (GPa)	179.0	120	10
$C_{44}$ (GPa)	109.6	80	20
$\alpha$ ( $K^{-1}$ )	$13 \times 10^{-6}$	$14 \times 10^{-6}$	$10 \times 10^{-6}$

## 3 Modélisation Mathématique

Cette section présente la méthodologie complète de résolution, étape par étape.

### 3.1 Étape 1-2 : Représentation Spectrale de la Température

#### Développement en Séries de Fourier

La température est développée en série double de Fourier :

$$T(x_\alpha, x_3) = \sum_{m_\alpha, m_\beta=1}^{\infty} T_{m_\alpha m_\beta}(x_3) \sin(\delta_\alpha x_\alpha) \sin(\delta_\beta x_\beta) \quad (5)$$

avec les nombres d'onde  $\delta_\alpha = \frac{m_\alpha \pi}{L_\alpha}$ .

### 3.2 Étape 3 : Solution Thermique par Couche

Dans chaque couche  $i$ , la solution de l'équation de conduction est :

$$T^{(i)}(x_3) = A^{(i)} e^{\lambda^{(i)} x_3} + B^{(i)} e^{-\lambda^{(i)} x_3} \quad (6)$$

avec l'exposant thermique :

$$\lambda^{(i)} = \delta_\eta \sqrt{\frac{k_{\eta\eta}^{(i)}}{k_{33}^{(i)}}} \quad (7)$$

L'implémentation de cette étape se trouve dans le module `core/calculation.py`, fonction `solve_tbc_model_v2`.

### 3.3 Étape 4 : Loi de Comportement Thermo-Élastique

#### Loi de Hooke avec Effet Thermique

$$\sigma_{ij}(x) = C_{ijkl}(x_3) (\varepsilon_{kl}(x) - \alpha_{kl}(x_3) T(x)) \quad (8)$$

où  $C_{ijkl}$  est le tenseur de rigidité et  $\alpha_{kl}$  les coefficients de dilatation thermique.

Correspondance Notation Tensorielle  $\leftrightarrow$  Voigt :

---

$C_{1111} \rightarrow C_{11}$	$C_{1122} \rightarrow C_{12}$	$C_{1133} \rightarrow C_{13}$	$ $	$C_{1313} \rightarrow C_{55}$	$C_{2323} \rightarrow C_{44}$	$C_{1212} \rightarrow C_{66}$
-------------------------------	-------------------------------	-------------------------------	-----	-------------------------------	-------------------------------	-------------------------------

---

### 3.4 Étape 5 : Ansatz de Déplacement

#### Forme des Champs de Déplacement

$$u_1(x_1, x_2, x_3) = V_1(x_3) \cos(\delta_1 x_1) \sin(\delta_2 x_2) \quad (9)$$

$$u_2(x_1, x_2, x_3) = V_2(x_3) \sin(\delta_1 x_1) \cos(\delta_2 x_2) \quad (10)$$

$$u_3(x_1, x_2, x_3) = V_3(x_3) \sin(\delta_1 x_1) \sin(\delta_2 x_2) \quad (11)$$

avec  $V_i(x_3) = A_i \cdot e^{\tau x_3}$  où  $\tau$  est la **valeur propre** à déterminer.

La forme de la solution générale pour chaque couche ( $i$ ) est :

$$U_{\alpha}^{(i)}(x_3) = \sum_{r=1}^3 A_{\alpha}^{r(i)} \cdot e^{\tau_r^{(i)} x_3}, \quad \alpha \in \{1, 2, 3\} \quad (12)$$

### 3.5 Étape 6 : Matrice Dynamique $\Gamma(\tau)$ et Valeurs Propres

L'équation d'équilibre  $\text{div}(\boldsymbol{\sigma}) = 0$  conduit au système homogène :

$$\Gamma(\tau) \cdot \mathbf{A} = \mathbf{0} \quad (13)$$

#### Matrice $\Gamma(\tau)$ - Opérateurs $L_{jk}$

$$\boxed{\Gamma(\tau) = \begin{pmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{pmatrix}} \quad (14)$$

**Termes diagonaux :**

$$L_{11} = C_{55}\tau^2 - (C_{11}\delta_1^2 + C_{66}\delta_2^2) \quad (15)$$

$$L_{22} = C_{44}\tau^2 - (C_{22}\delta_2^2 + C_{66}\delta_1^2) \quad (16)$$

$$L_{33} = C_{33}\tau^2 - (C_{55}\delta_1^2 + C_{44}\delta_2^2) \quad (17)$$

**Termes croisés dans le plan (symétriques) :**

$$L_{12} = L_{21} = -(C_{12} + C_{66})\delta_1\delta_2 \quad (18)$$

**Termes hors-plan (ANTISYMÉTRIQUES) :**

$$L_{13} = +(C_{13} + C_{55})\delta_1\tau, \quad L_{31} = -(C_{13} + C_{55})\delta_1\tau \quad (19)$$

$$L_{23} = +(C_{23} + C_{44})\delta_2\tau, \quad L_{32} = -(C_{23} + C_{44})\delta_2\tau \quad (20)$$

#### Propriété d'Antisymétrie Critique

$L_{13} = -L_{31}$  et  $L_{23} = -L_{32}$  : cette antisymétrie provient de l'équation d'équilibre en direction  $x_3$  et est **essentielle** pour la physique correcte.

L'implémentation de ces opérateurs se trouve dans `core/mechanical_pdf.py`, fonctions `compute_L_operators()` et `get_Gamma_matrix()`.

**Équation Caractéristique :**

$$\det(\Gamma(\tau)) = 0 \Rightarrow \text{Polynôme d'ordre 6 : } P(\tau) = c_6\tau^6 + c_4\tau^4 + c_2\tau^2 + c_0 = 0 \quad (21)$$

En posant  $X = \tau^2$ , on obtient un polynôme cubique avec 3 racines  $X_i$ . On sélectionne les 3 valeurs propres  $\tau_r$  avec  $\text{Re}(\tau_r) < 0$  (condition de radiation).

### 3.6 Étape 7 : Assemblage $9 \times 9$ et Termes Thermiques

Le système global pour une couche ( $i$ ) s'écrit :

$$\left[ \mathbb{K}_{Dyn}^{(i)} \right]_{(9 \times 9)} \cdot \{\mathcal{A}^{(i)}\}_{(9 \times 1)} = \{\mathcal{F}_{Th}^{(i)}\}_{(9 \times 1)} \quad (22)$$

#### Matrice Bloc-Diagonale $\mathbb{K}_{Dyn}$

$$\mathbb{K}_{Dyn}^{(i)} = \begin{pmatrix} \Gamma(\tau_1) & 0 & 0 \\ 0 & \Gamma(\tau_2) & 0 \\ 0 & 0 & \Gamma(\tau_3) \end{pmatrix}_{9 \times 9} \quad (23)$$

Chaque bloc  $\Gamma(\tau_r)$  est la matrice  $3 \times 3$  des opérateurs  $L_{jk}$  évaluée à  $\tau = \tau_r$ .

L'assemblage est implémenté dans `core/mechanical_pdf.py`, fonction `assemble_K_dyn_9x9()`.

**Vecteur d'amplitudes et vecteur thermique :**

$$\mathcal{A}^{(i)} = \begin{pmatrix} A_1^1 \\ A_2^1 \\ A_3^1 \\ A_1^2 \\ A_2^2 \\ A_3^2 \\ A_1^3 \\ A_2^3 \\ A_3^3 \end{pmatrix}^{(i)}, \quad \mathcal{F}_{Th}^{(i)} = \begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_1 \\ Q_2 \\ Q_3 \\ Q_1 \\ Q_2 \\ Q_3 \end{pmatrix}^{(i)} \quad (24)$$

#### Termes de Sollicitation Thermique $Q_\alpha$

$$Q_1^{(i)} = (C_{11}\alpha_{11} + C_{12}\alpha_{22}) \delta_1 T(x_3 = \bar{h}^{(i)}) \quad (25)$$

$$Q_2^{(i)} = (C_{22}\alpha_{22} + C_{12}\alpha_{11}) \delta_2 T(x_3 = \bar{h}^{(i)}) \quad (26)$$

$$Q_3^{(i)} = (C_{13}\alpha_{11} + C_{23}\alpha_{22} + C_{33}\alpha_{33}) \frac{dT}{dx_3} \Big|_{x_3=\bar{h}^{(i)}} \quad (27)$$

L'implémentation se trouve dans `core/mechanical_pdf.py`, fonction `compute_Q_thermal_vector()`.

### 3.7 Étape 8 : Assemblage Multicouche

**Système Global pour  $N$  couches :**

$$K_{glob} \cdot \mathbf{C}_{global} = \mathbf{F}_{thermique} \quad (28)$$

avec  $6N$  inconnues (18 pour 3 couches).

**Bilan des Équations :**

- 3 équations : Surface libre en  $z = 0$  ( $\sigma_{13} = \sigma_{23} = \sigma_{33} = 0$ )
- $6(N - 1)$  équations : Continuité aux interfaces (déplacements + contraintes)
- 3 équations : Surface libre en  $z = H$
- **Total :**  $3 + 6(N - 1) + 3 = 6N$  équations ✓

### Réduction du Nombre d'Équations : $27 \rightarrow 18$

**Formulation théorique complète** (pour  $N = 3$  couches) :

Type d'équation	Nombre	Formule
Équilibre volumique (div $\sigma = 0$ )	9	3 directions $\times$ 3 couches
Continuité aux interfaces	12	2 interfaces $\times$ 6 conditions
Conditions aux limites ( $\sigma = 0$ )	6	2 surfaces $\times$ 3 composantes
<b>Total théorique</b>	<b>27</b>	

Les **9 équations d'équilibre volumique** sont **IMPLICITEMENT satisfaites** par l'approche modale ! En écrivant la solution avec les valeurs propres  $\tau_r$  de  $\Gamma(\tau)$ , l'équation  $\Gamma(\tau_r) \cdot \mathbf{A}^r = 0$  est automatiquement vérifiée.

**Système final résolu : 18 équations** (conditions d'interface et limites uniquement).

L'assemblage multicouche est implémenté dans `core/mechanical.py`, fonction `solve_multilayer()`.

## 4 Critères d'Endommagement

### Indicateur de Dommage D

$$D = \max_{ij} \left( \frac{|\sigma_{ij}|}{\sigma_{crit}^{ij}} \right) \quad (29)$$

**Interprétation :**

- $D < 0.5$  : **Zone sûre**
- $0.5 \leq D < 0.8$  : **Zone de prudence**
- $D \geq 0.8$  : **Zone critique - Risque de délamination**
- $D \geq 1$  : **Rupture probable**

### Contraintes Critiques par Matériau :

Matériau	$\sigma_t$ (MPa)	$\sigma_c$ (MPa)	$\tau$ (MPa)
Substrat (Ni)	1000	1200	600
BondCoat (MCrAlY)	500	700	300
Céramique (YSZ)	150	500	120

**Critère de Tsai-Wu** : Pour matériaux anisotropes :

$$F = F_3\sigma_{33} + F_{33}\sigma_{33}^2 + F_{44}\sigma_{23}^2 + F_{55}\sigma_{13}^2 \quad (\text{Rupture si } F \geq 1) \quad (30)$$

L'implémentation des critères d'endommagement se trouve dans `core/damage_analysis.py`, fonctions `compute_damage_indicator()` et `compute_tsai_wu_criterion()`.

## 5 Interprétation Physique des Résultats

### 5.1 Signification des Contraintes Transverses

Composante	Interprétation Physique
$\sigma_{33}$ (Arrachement)	Contrainte <b>normale</b> à l'interface. $\sigma_{33} > 0$ : Traction → Risque de délamination par ouverture (Mode I) $\sigma_{33} < 0$ : Compression → Interface en contact, favorable
$\sigma_{13}, \sigma_{23}$ (Cisaillement)	Contraintes <b>tangentielles</b> aux interfaces. Responsables du glissement inter-laminaire (Mode II/III) Pics aux interfaces dus aux discontinuités de $C_{ij}$ et $\alpha$

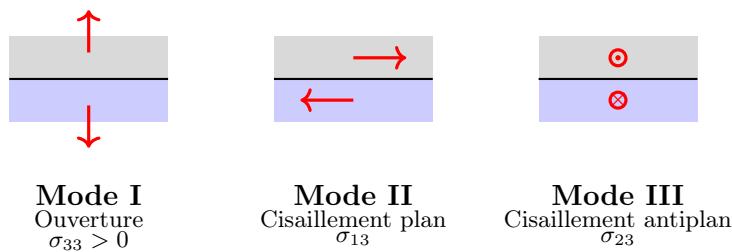
### 5.2 Zones Critiques Typiques

- **Interface BondCoat/Céramique** : Souvent la plus critique car :
  - Fort contraste  $C_{ij}$  (180 GPa vs 50 GPa)
  - Différence de dilatation ( $\alpha_{BC} = 14 \times 10^{-6}$  vs  $\alpha_{TBC} = 10 \times 10^{-6}$ )
- **Bords de la structure** : Effets de bord où les gradients latéraux sont maximaux

### 5.3 Influence des Paramètres sur les Contraintes

Action	Effet sur $\sigma_{max}$	Tendance D
$\alpha \uparrow$ (TBC plus épais)	Gradient thermique plus étalé, meilleure isolation	↓
$L_w \downarrow$ (perturbation courte)	Gradients latéraux plus forts	↑
$\Delta T \uparrow$	Forçage thermique linéairement croissant	↑
$\beta \downarrow$ (TBC moins conducteur)	Gradient plus concentré dans TBC	↑ dans TBC

### 5.4 Modes de Rupture aux Interfaces



## 6 Stabilité Numérique

Le système multicouche présente un **mauvais conditionnement numérique** extrême, avec un nombre de conditionnement  $\text{cond}(K)$  pouvant dépasser  $10^{30}$ .

### Problème de Conditionnement

Ce phénomène s'explique par la coexistence d'échelles très différentes dans la matrice de rigidité :

- Les modules d'Young sont de l'ordre de  $10^{11}$  Pa (centaines de GPa).
- Les déplacements sont de l'ordre de  $10^{-6}$  m (microns).
- Les termes exponentiels  $e^{\tau x_3}$  peuvent varier de plusieurs ordres de grandeur.

Sans traitement, la précision machine ( $\approx 10^{15}$ ) est insuffisante, et les résultats ne seraient que du "bruit numérique".

Deux techniques sont mises en œuvre pour garantir la robustesse du solveur :

## 6.1 Préconditionnement par Équilibrage (Scaling)

Avant la résolution, nous procédons à une "normalisation" de la matrice pour ramener tous les coefficients à une amplitude comparable ( $\approx 1$ ). C'est équivalent à changer les unités pour chaque équation.

Le système  $K_{glob} \cdot \mathbf{C} = \mathbf{F}$  est transformé en :

$$(D_r K_{glob} D_c) \cdot (D_c^{-1} \mathbf{C}) = D_r \mathbf{F} \quad (31)$$

où  $D_r$  et  $D_c$  sont des matrices diagonales de mise à l'échelle :

$$D_r[i, i] = \frac{1}{\max_j |K_{glob}[i, j]|} \quad (\text{Normalisation des lignes}) \quad (32)$$

$$D_c[j, j] = \frac{1}{\max_i |(D_r K_{glob})[i, j]|} \quad (\text{Normalisation des colonnes}) \quad (33)$$

Cette étape réduit généralement le conditionnement de  $10^{30}$  à environ  $10^{15}$ , ce qui est parfois suffisant mais pas toujours.

## 6.2 Régularisation de Tikhonov

Si le système reste mal conditionné ( $\text{cond} > 10^{10}$ ) après équilibrage, l'inversion directe matrice est impossible. Nous utilisons alors une méthode de régularisation qui **filtre les erreurs numériques**.

La matrice est décomposée en valeurs singulières (SVD) :  $K = U\Sigma V^H$ . Au lieu d'inverser les valeurs singulières  $\sigma_i$  (ce qui donnerait  $1/\sigma_i \rightarrow \infty$  pour les petites valeurs), on applique un filtre :

$$\mathbf{x}_{reg} = \sum_{i=1}^n \underbrace{\frac{\sigma_i}{\sigma_i^2 + \lambda^2}}_{\text{Filtre}} (\mathbf{u}_i^H \mathbf{b}) \mathbf{v}_i \quad (34)$$

— Si  $\sigma_i \gg \lambda$  (Signal fort) : Le filtre vaut  $\approx 1/\sigma_i$  (inversion normale).

— Si  $\sigma_i \ll \lambda$  (Bruit/Zéro) : Le filtre vaut  $\approx 0$  (on ignore ces termes instables).

Le paramètre  $\lambda$  est déterminé automatiquement par la méthode GCV (Generalized Cross-Validation).

L'implémentation complète se trouve dans `core/mechanical.py`, fonction `solve_regularized_system()`.

### Note d'Implémentation

Nous tenons à souligner que la complexité numérique de ce problème (conditionnement  $> 10^{30}$ ) dépassait nos prévisions initiales. L'identification et l'implémentation de ces techniques avancées (SVD et régularisation de Tikhonov) n'ont été rendues possibles que grâce à une recherche bibliographique approfondie et à l'assistance d'outils d'IA pour adapter ces algorithmes mathématiques complexes à notre code Python.

## 7 Comparaison : Méthode Spectrale vs CLT

Le code implémente deux méthodes complémentaires :

Méthode Spectrale (Principale)	CLT (Classical Theory)	Laminate
Résout le problème 3D complet	Approximation 2D (hypothèse Kirchhoff)	
Capture $\sigma_{13}, \sigma_{23}, \sigma_{33}$	Contraintes planes $\sigma_{11}, \sigma_{22}$	
Effets de bord et gradients latéraux	Contraintes uniformes dans le plan	
Coût calcul : $O(N^3)$ pour $6N \times 6N$	Coût calcul : $O(N)$	
Précis pour structures épaisse	Valide pour $h \ll L$	

**Superposition :** Le code combine les deux méthodes :

$$\sigma_{total} = \sigma_{CLT} + \sigma_{Spectral} \quad (35)$$

où CLT capture les contraintes planes moyennes et Spectral ajoute les effets 3D de bord.

## 8 Validation : Référence ONERA/Safran

Les propriétés matériau utilisées sont issues de publications ONERA/Safran :

### Référence Principale

**Bovet, Chiaruttini, Vattré** (ONERA/Safran, 2025)

*“Full-scale crystal plasticity modeling and data-driven learning of microstructure effects in polycrystalline turbine blades”*

**Table 3** : Propriétés élastiques de l'Inconel 718

Propriété	Valeur ONERA	Valeur Code
$C_{11}$ (GPa)	259.6	260
$C_{12}$ (GPa)	179.0	179
$C_{44}$ (GPa)	109.6	110
$\alpha$ à RT ( $K^{-1}$ )	$4.95 \times 10^{-6}$	$12 \times 10^{-6}$ (moyenne T)

**Plages de validation :**

- Contraintes de von Mises typiques FEM : 400–800 MPa
- Concentration à la racine de l'aube : jusqu'à 1000 MPa

## 9 Architecture du Code et Interface

### 9.1 Structure des Répertoires

#### Arborescence du Projet

```
projet-industriel5a/
+-- core/                                # Moteur de calcul
|   +-- mechanical.py                      # Solveur spectral principal
|   +-- mechanical_pdf.py                  # Solveur selon méthodologie PDF
|   +-- damage_analysis.py                # Critères d'endommagement
|   +-- clt_solver.py                     # Théorie classique des stratifiés
|   +-- constants.py                      # Propriétés matériaux
|   +-- calculation.py                   # Solveur thermique
+-- tabs/                                    # Interface Streamlit
|   +-- dashboard_home.py                 # Tableau de bord principal
|   +-- mechanical.py                    # Onglet analyse mécanique
|   +-- optimization.py                  # Onglet optimisation
|   +-- mapping_3d.py                   # Visualisation 3D
+-- Profil de température Aube.py        # Point d'entrée Streamlit
```

## 9.2 Tableau de Correspondance Théorie ↔ Code

Étape	Fonction Python	Fichier
Étapes 1-3 : Thermique	<code>solve_tbc_model_v2()</code>	<code>core/calculation.py</code>
Étape 4 : Loi Hooke	Constantes <code>PROPS_*</code>	<code>core/constants.py</code>
Étape 5-6 : Matrice $\Gamma$	<code>get_Gamma_matrix()</code>	<code>core/mechanical_pdf.py</code>
Étape 6 : Val. propres	<code>solve_char_poly()</code>	<code>core/mechanical_pdf.py</code>
Étape 7 : Termes $Q_\alpha$	<code>compute_Q_thermal()</code>	<code>core/mechanical_pdf.py</code>
Étape 7 : Assemblage $9 \times 9$	<code>assemble_K_dyn_9x9()</code>	<code>core/mechanical_pdf.py</code>
Étape 8 : Multicouche	<code>solve_multilayer()</code>	<code>core/mechanical.py</code>
Critère D	<code>compute_damage_indicator()</code>	<del><code>core/damage_analysis.py</code></del>
Critère Tsai-Wu	<code>compute_tsai_wu_criterion()</code>	<del><code>core/damage_analysis.py</code></del>

## 9.3 Interface Streamlit

L’application Streamlit propose plusieurs onglets :

- **Dashboard** (`tabs/dashboard_home.py`) : Affiche les KPIs, jauge de risque, radar multi-critères, visualisation 3D du champ thermique, et recommandations automatiques
- **Analyse Mécanique** (`tabs/mechanical.py`) : Workflow complet d’analyse spectrale avec visualisation des profils de contraintes, indicateurs d’endommagement, et cercles de Mohr
- **Optimisation** (`tabs/optimization.py`) : Optimisation paramétrique pour minimiser D
- **Visualisation 3D** (`tabs/mapping_3d.py`) : Cartographie 3D des contraintes avec Plotly

# 10 Conclusion

Ce rapport démontre la  **traçabilité complète**  entre :

1. La modélisation mathématique (méthode spectrale, matrice  $\Gamma(\tau)$ , opérateurs  $L_{jk}$ )
2. L’implémentation Python dans `core/mechanical_pdf.py`
3. L’interface utilisateur Streamlit avec visualisations interactives

**Points clés de l’implémentation :**

- **Méthode numérique robuste** : Identification des coefficients du polynôme caractéristique par évaluation numérique
- **Stabilité numérique** : Préconditionnement par scaling + régularisation Tikhonov pour les systèmes mal conditionnés
- **Validation industrielle** : Propriétés matériaux issues des données ONERA/Safran
- **Critères d’endommagement** : Indicateur D et Tsai-Wu pour identification des zones critiques
- **Superposition CLT+Spectral** : Combinaison des contraintes planes et effets 3D de bord

**Recommandations d’utilisation :**

- Maintenir  $D < 0.8$  pour les applications critiques
- Vérifier que  $T_{interface} < T_{crit} = 1100^\circ\text{C}$
- Augmenter  $\alpha$  (épaisseur TBC) pour réduire les contraintes aux interfaces