



CACHAN

ENSC-2012/2013



CACHAN

MÉMOIRE DE STAGE DE MASTER 2  
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Présentée par

**Pierre NARGIL**

pour obtenir le diplôme de

**Master TACS**

Sujet du stage

**Extension de la PGD à des problèmes de dynamique  
non-linéaires**

Soutenue à Cachan le 28 juin 2013, Encadré par :

Pierre-Alain BOUCARD	Professeur des Universités, ENS Cachan
Francois LOUF	Maître de conférences, ENS Cachan

**LMT-Cachan**

ENS Cachan / CNRS / UPMC / PRES UniverSud Paris  
61 avenue du Président Wilson, F-94235 Cachan cedex, France

# Sommaire

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	La réduction de modèles . . . . .	4
1.2	La PGD . . . . .	5
1.2.1	Généralités . . . . .	5
1.2.2	Mise en place de nouvelles variables : nouvelles coordonnées . . . . .	5
<b>2</b>	<b>Les méthodes de réduction de modèles</b>	<b>7</b>
2.1	Condensation . . . . .	7
2.2	Rayleigh-Ritz . . . . .	9
2.3	POD . . . . .	10
2.3.1	La SVD, Singular Value Decomposition . . . . .	10
2.3.2	La formulation continue . . . . .	13
2.4	PGD . . . . .	15
<b>3</b>	<b>Le calcul dynamique</b>	<b>17</b>
3.1	Les schémas d'intégration . . . . .	17
3.1.1	Schémas de Newmark . . . . .	18
3.1.2	HHT . . . . .	18
3.2	La PGD . . . . .	20
3.2.1	Définition du problème . . . . .	20
3.2.2	Problème en espace . . . . .	21
3.2.3	Problème en temps . . . . .	22
3.2.4	Problème en paramètre . . . . .	23
<b>4</b>	<b>Programmation</b>	<b>24</b>
4.1	Plan du programme . . . . .	24
4.2	Génération d'un problème . . . . .	25
4.3	Résolution temporelle sur base complète . . . . .	25
4.4	POD . . . . .	25
4.5	Résolution temporelle sur base réduite . . . . .	25
4.5.1	Indicateurs d'erreur . . . . .	26
4.5.2	Post-traitement . . . . .	26
4.6	Méthode de Rayleigh-Ritz . . . . .	27
4.7	PGD . . . . .	27
<b>5</b>	<b>Résultats</b>	<b>28</b>
5.1	Condensation . . . . .	28
5.2	Méthode Rayleigh-Ritz . . . . .	28
5.3	POD . . . . .	28
5.4	PGD . . . . .	29

<b>6</b>	<b>La PDG pour les problèmes de dynamique non-linéaire</b>	<b>32</b>
6.1	La Méthode LATIN . . . . .	32
6.2	Les Résolutions non-linéaires . . . . .	33
6.2.1	Condensation . . . . .	33
6.2.2	Rayleigh-Ritz . . . . .	33
6.2.3	POD . . . . .	33
6.3	Ouverture . . . . .	33
	 <b>Bibliographie</b>	 <b>33</b>
<b>A</b>	<b>Remerciements</b>	<b>36</b>
<b>B</b>	<b>Compression d'image par SVD</b>	<b>37</b>
<b>C</b>	<b>Calcul du problème PGD étape par étape</b>	<b>40</b>
<b>D</b>	<b>Intégration de produits</b>	<b>45</b>
D.1	Intégration d'un produit de fonctions linéaires par morceaux . . . . .	46
D.2	Intégration de produit de fonctions constantes par morceaux . . . . .	47
<b>E</b>	<b>Problème en temps avec utilisation d'éléments finis temporels</b>	<b>48</b>
<b>F</b>	<b>Algorithmes PGD à deux variables</b>	<b>50</b>
F.1	Résumé . . . . .	50
F.2	Orthogonalisation . . . . .	50
F.2.1	Dans le point fixe . . . . .	50
F.2.2	Amélioration de la base . . . . .	51

# Chapitre 1

## Introduction

L'objectif de ce travail est d'appliquer une méthode de résolution PGD (Proper Generalized Decomposition), qui est une méthode de réduction de modèles où la solution est à variables séparées, à des problèmes de dynamique présentant des non-linéarités. Ce rapport propose ici introduction à la PGD avec une présentation d'autres méthodes de réductions de modèles, tel que la condensation, la méthode de Rayleigh-Ritz et la POD.

### 1.1 La réduction de modèles

La résolution numérique de problèmes de mécanique fait partie intégrante du travail de la plupart des ingénieurs, et connaît évidemment des limitations. Ces résolutions représentent des calculs gourmands en ressources et en temps, et la volonté croissante des industriels de s'appuyer le plus souvent sur la simulation plutôt que sur l'essai, place l'amélioration des méthodes utilisées au cœur de la problématique de nombreuses recherches. Pour l'industriel le gain de temps est un enjeu majeur pour être concurrentiel, voire pour réduire le coût d'une étude et donc au final de son produit. Les méthodes de réductions de modèles tendent à simplifier le problème pour réduire sa taille. Elles sont basées sur une diminution de la dimension de l'espace de recherche des solutions. Ceci nécessite la construction d'une base réduite. On peut alors distinguer parmi ces méthodes, celles qui nécessitent une connaissance préalable d'une solution pour construire la base sur laquelle le problème sera projeté, elles sont dites "a posteriori", de celles dites "a priori" qui construisent la base pendant le calcul de résolution.

Parmi les méthodes "a posteriori" se trouve la POD (Proper Orthogonal Decomposition), qui a fait l'objet de nombreux travaux d'études ces vingt dernières années. Il s'agit d'une méthode basée sur la séparation de variables qui sera présentée plus en détail par la suite.

Outre une amélioration du temps de résolution pour des calculs couramment effectués par les industriels, la réduction de modèle peut se montrer si efficace dans la diminution du temps de calcul que de nouvelles résolutions peuvent être envisagées. En effet l'industrie rencontre des problèmes de tailles telles qu'elle ne peut envisager de les résoudre pour des questions de temps, voire de limites des performances même des plus puissantes machines. La première possibilité apportée par une réduction de modèle est l'utilisation pour l'optimisation. Puisque même si les méthodes "a pos-

teriori" entraînent une première résolution du problème par une méthode standard, elles permettent, une fois la base réduite obtenue, de faire une série de calculs rapides pouvant permettre une optimisation ou une étude de type Monte Carlo sur le modèle réduit.

## 1.2 La PGD

### 1.2.1 Généralités

La PGD, pour "Proper Generalized Decomposition", trouve son origine dans [1] sous le nom de "radial time-space approximation", comme faisant partie la méthode LATIN (voir également [2]).

Parmi les possibilités apportées par la réduction de modèles, elle peut permettre de résoudre des problèmes jusque là inenvisageable par exemple dans la chimie quantique (quand des processus chimiques impliquent un nombre de molécules de réactifs si petit que le concept de continuité de concentration n'est plus valide, par exemple dans la recherche génétique). De tels problèmes peuvent rapidement être impossibles à résoudre par des méthodes standards, car ils souffrent de ce que l'on appelle la malédiction de la dimension. La solution est inaccessible par des approches traditionnelles de discrétisation utilisant un maillage car elles présentent une taille du système à résoudre qui croît exponentiellement en fonction du nombre de particules. Plus de détail sur ce sujet sont donné dans [3].

La diminution drastique de temps de calcul apportée par la PGD permet d'envisager de nouvelles utilisations de la simulation, notamment dans les DDDAS (pour Dynamic Data-Driven Application System), dont l'enjeu est de permettre à la simulation d'interagir avec une manipulation en temps réel. C'est-à-dire que les instruments actionneurs seront influencés par les résultats de calculs apportés par la simulation et les capteurs du système renverront des données à la simulation pour réajuster les calculs. Les DDDAS sont un sujet d'études identifié par la USNSF (United States National Science Foundation) dans son rapport en 2006 sur les SBES (Simulation-based Engineering Sciences) comme étant parmi les cinq challenges majeurs de la prochaine décennie. Pour plus de détails sur la PGD dans le cadre des DDDAS le lecteur pourra consulter [4].

La PDG rencontre donc un franc succès depuis quelques années et ses applications sont pléthores. Un aperçu des différents travaux concernant cette méthode peut être trouvé dans [5].

### 1.2.2 Mise en place de nouvelles variables : nouvelles coordonnées

La PGD présente un autre avantage, par la mise en place de coordonnées supplémentaires, comme le décrivent [3] et [4]. Il s'agit de rajouter au produit de fonctions à variables séparées, une fonction d'une nouvelle variable. Ceci permet de conserver une variable comme une inconnue et ainsi d'avoir un problème résolu quelque soit la valeur de cette variable. Pour obtenir un tel résultat sans cette méthode il faudrait effectuer de nombreux tirages de valeurs pour la variable et réaliser une approche de Monte Carlo, ce qui requerrait autant de résolutions. L'ajout de cette coordonnée

permet donc de passer outre la nécessité de la valeur d'une variable au calcul, ou de résoudre une famille de problèmes qui dépendent de cette variable. C'est le principe de base de l'analyse stochastique, qui fait l'objet d'un intérêt grandissant. De cette manière la résolution du problème par la PGD apporte plus encore qu'une résolution directe, et peut permettre facilement la mise en place d'une optimisation. Dans les articles cités précédemment, les auteurs présentent comme exemple un problème de thermique à résoudre pour lequel la conductivité matériau est inconnue. Mais ceci ne se limite pas à une valeur matériau et peut s'appliquer à une condition initiale ou à une valeur géométrique...

# Chapitre 2

## Les méthodes de réduction de modèles

Les notations dans ce rapport seront les suivantes (elles n'ont de sens que pour les élément discrétisé) :

- pour un vecteur :  $\mathbf{v}$  (en minuscule et gras)
- pour un élément du vecteur  $\mathbf{v}$  :  $v_i$  (en minuscule et non gras)
- pour une matrice :  $\mathbf{M}$  (en majuscule et gras)
- pour un élément de la matrice  $\mathbf{M}$  :  $M_{ij}$  (en majuscule et non gras)
- pour un scalaire en minuscule et non gras.
- $\mathbf{I}$  est la matrice identité.
- $.^T$  est l'opérateur transposition.
- $\delta_{..}$  et le symbole de Kronecker.

Un problème de dynamique peut s'écrire sous la forme (pour le cas discret) :

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \quad (2.1)$$

- $\mathbf{M}$  est la matrice de masse
- $\mathbf{C}$  est une matrice d'amortissement
- $\mathbf{K}$  est une matrice de raideur
- $\mathbf{u}$  est le vecteur contenant les déplacements aux nœuds
- $\mathbf{f}$  est le vecteur les efforts extérieurs sur les nœuds

Il existe plusieurs méthodes de résolution, et celles de réductions de modèles se basent sur une projection du problème sur une base, pour en diminuer la taille.

### 2.1 Condensation

La résolution classique à l'aide d'un modèle de discrétisation tel que les éléments finis se base sur une matrice dont la taille dépend de la finesse du maillage. Une géométrie complexe dans une zone d'intérêt nécessitera un raffinement qui entraîne un agrandissement du problème à résoudre. Avant de se lancer dans une résolution coûteuse, il peut être intéressant de réduire la taille du modèle en condensant les degrés de liberté possédant peu ou pas d'inertie et qui ne sont pas soumis à des efforts extérieurs. Les nœuds amenés à être condensés sont appelés nœuds esclaves et ceux qui seront conservés sont appelés nœuds maîtres.

Le système se présente alors sous cette forme si l'on néglige l'inertie des nœuds

esclaves :

$$\begin{bmatrix} \mathbf{M}_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_1 \\ \ddot{\mathbf{u}}_2 \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ 0 \end{Bmatrix} \quad (2.2)$$

Où  $\mathbf{u}_1$  contient les degrés de libertés des nœuds maîtres et  $\mathbf{u}_2$  ceux des nœuds esclaves. On obtient alors les deux équations :

$$\begin{cases} \mathbf{M}_{11}\ddot{\mathbf{u}}_1 + \mathbf{K}_{11}\mathbf{u}_1 + \mathbf{K}_{12}\mathbf{u}_2 = \mathbf{f}_1 \\ \mathbf{K}_{21}\mathbf{u}_1 + \mathbf{K}_{22}\mathbf{u}_2 = \mathbf{0} \end{cases} \quad (2.3)$$

On peut alors déterminer le comportement des nœuds esclaves à partir des nœuds maîtres ainsi :

$$\mathbf{u}_2 = \mathbf{K}_{22}^{-1} \mathbf{K}_{21} \mathbf{u}_1 \quad (2.4)$$

Alors la résolution du système se réduit à :

$$\mathbf{M}_{11}\ddot{\mathbf{u}}_1 + (\mathbf{K}_{11}\mathbf{u}_1 - \mathbf{K}_{12}\mathbf{K}_{22}^{-1}\mathbf{K}_{21})\mathbf{u}_1 = \mathbf{f}_1 \quad (2.5)$$

qui ne tient compte que des nœuds maîtres. La résolution du problème modifié par cette condensation apporte la solution exacte en statique, puisqu'alors l'inertie n'a pas de rôle. L'approximation apportée par la réduction de Guyan est de conserver la relation entre maîtres et esclaves 2.4 même quand les seconds possèdent un peu d'inertie. On introduit  $\mathbf{L}$  :

$$\mathbf{u} = \begin{Bmatrix} \ddot{\mathbf{u}}_1 \\ \ddot{\mathbf{u}}_2 \end{Bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{K}_{22}^{-1}\mathbf{K}_{21} \end{bmatrix} \mathbf{u}_1 = \mathbf{L}\mathbf{u}_1 \quad (2.6)$$

Le calcul de l'énergie cinétique et de l'énergie potentielle deviennent :

$$\begin{cases} e_c = \frac{1}{2}\ddot{\mathbf{u}}\mathbf{M}\ddot{\mathbf{u}} = \frac{1}{2}\ddot{\mathbf{u}}_1\mathbf{L}^T\mathbf{M}\mathbf{L}\ddot{\mathbf{u}}_1 = \frac{1}{2}\ddot{\mathbf{u}}_1\hat{\mathbf{M}}\ddot{\mathbf{u}}_1 \\ e_p = \frac{1}{2}\mathbf{u}\mathbf{K}\mathbf{u} = \frac{1}{2}\mathbf{u}_1\mathbf{L}^T\mathbf{K}\mathbf{L}\mathbf{u}_1 = \frac{1}{2}\mathbf{u}_1\hat{\mathbf{K}}\mathbf{u}_1 \end{cases} \quad (2.7)$$

Ce qui permet de définir les matrices de masse et de raideur réduites :

$$\begin{aligned} \hat{\mathbf{M}} &= \mathbf{L}^T\mathbf{M}\mathbf{L} \\ \hat{\mathbf{K}} &= \mathbf{L}^T\mathbf{K}\mathbf{L} \end{aligned} \quad (2.8)$$

De la même manière si l'on souhaite introduire des efforts extérieurs sur des nœuds esclaves on calcule des efforts réduits :

$$\hat{\mathbf{f}} = \mathbf{L}^T\mathbf{f} \quad (2.9)$$

Le système final est alors :

$$\hat{\mathbf{M}}\ddot{\mathbf{u}}_1 + \hat{\mathbf{K}}\mathbf{u}_1 = \hat{\mathbf{f}} \quad (2.10)$$

Cette méthode de condensation assure un résultat exact si au moment de la sélection des nœuds maîtres on les choisit tels que les degrés de liberté condensés n'ont pas d'inertie. Il est possible de démontrer que l'erreur dans la forme d'un mode due à la condensation de Guyan est une fonction croissante de rapport :  $\omega_i^2/\eta^2$  où  $\omega_i$  est la pulsation propre associée au mode  $i$ , et  $\eta$  est la première pulsation propre du système contraint, c'est à dire du système où les nœuds maîtres sont bloqués ( $\eta$  est la solution de  $\det(\mathbf{K}_{22} - \eta^2\mathbf{M}_{22}) = 0$  la plus faible). Alors  $\eta$  doit être suffisamment grand comparé à la bande de fréquences dans laquelle le modèle est étudié.

Une utilisation de réduction de modèle par condensation appelée Equivalent reduced model technique est définie dans [6] pour résoudre des problèmes comprenant des non-linéarités. Mais celles-ci sont limitées à des interfaces non-linéaires entre des sous-structures au comportement linéaire.



## 2.2 Rayleigh-Ritz

La méthode de Rayleigh-Ritz est une approche modale basée sur le quotient de Rayleigh généralisée par Ritz. Elle part de l'hypothèse que l'amortissement du système est faible et qu'il a peu d'influence sur les modes et leurs fréquences propres associées. Cette méthode est notamment utilisée par le code Aster. Elle utilise comme base réduite, l'ensemble des premiers modes de vibration de la structure. Ce qui motive l'utilisation d'une base de modes propres est que ceux-ci sont **M**-orthogonaux et **K**-orthogonaux. Les projections de **M** et **K** dans cette base sont donc diagonales. Les modes sont définis comme les couples  $(\omega_i, \varphi_i)$  solutions de l'équation :

$$(\mathbf{K} - \omega^2 \mathbf{M})\varphi = 0 \quad (2.11)$$

Pour calculer les modes comme on résout un problème aux valeurs propres, on pose :

$$\mathbf{A}\varphi = \frac{1}{\omega^2}\varphi \quad \text{avec} \quad \mathbf{A} = \mathbf{K}^{-1}\mathbf{M} \quad (2.12)$$

On sait que l'ensemble des modes propres engendre l'espace des champs cinématiquement admissibles. On cherche donc le mode propre de plus basse fréquence en initialisant avec un  $\varphi^{(0)}$  quelconque, dont on fait l'hypothèse qu'il se décompose sur tous les modes qui forment la base. Alors il s'écrit :

$$\varphi^{(0)} = \sum_{i \in \mathbb{N}} a_i \varphi_i \quad \text{avec} \quad a_i \neq 0 \quad \forall i \in \mathbb{N} \quad (2.13)$$

Si le système étudié a un nombre de modes limité, il faut remplacer :

$$\forall i \in \mathbb{N} \text{ par } \forall i \in \llbracket 1; NbMode \rrbracket$$

On multiplie par **A** et on utilise la formule 2.12

$$\varphi^{(1)} = \mathbf{A}\varphi^{(0)} = \sum_{i \in \mathbb{N}} a_i \mathbf{A}\varphi_i = \sum_{i \in \mathbb{N}} a_i \frac{1}{\omega_i^2} \varphi_i \quad (2.14)$$

On itère ainsi pour trouver le terme dominant :

$$\varphi^{(r)} = \sum_{i \in \mathbb{N}} a_i \frac{1}{\omega_i^{2r}} \varphi_i \approx a_1 \frac{1}{\omega_1^{2r}} \varphi_1 \quad (2.15)$$

Une fois le terme déterminé on peut calculer sa fréquence propre associée :

$$\mathbf{A}\varphi^{(r)} \approx \mathbf{A}a_1 \frac{1}{\omega_1^{2r}} \varphi_1 = \frac{1}{\omega_1^2} a_1 \frac{1}{\omega_1^{2r}} \varphi_1 \approx \frac{1}{\omega_1^2} \varphi^{(r)} \quad (2.16)$$

Les deux équations précédentes permettent de déterminer  $\varphi_1$  et  $\omega_1$  d'autant plus rapidement que  $\omega_1 \gg \omega_2$  (un  $r$  suffisant pour atteindre une convergence souhaitée sera d'autant plus petit). Pour calculer le mode suivant on choisit un  $\varphi^{(0)}$  d'initialisation tel que :

$$\varphi_1^T \mathbf{M} \varphi^{(0)} = 0 \quad (2.17)$$

donc d'après l'équation 2.13 :

$$\varphi_1^T \mathbf{M} \varphi^{(0)} = 0 = \varphi_1^T \mathbf{M} \sum_{i \in \mathbb{N}} a_i \varphi_i = \sum_{i \in \mathbb{N}} \varphi_1^T \mathbf{M} a_i \varphi_i = \varphi_1^T \mathbf{M} a_1 \varphi_1 \quad \text{car} \quad \varphi_i^T \mathbf{M} \varphi_j = \delta_{ij} \quad (2.18)$$

Cela donne que  $a_1$  (la composante de  $\varphi^{(0)}$  sur le premier mode :  $\varphi_1$ ) est nulle, i.e.  $\varphi^{(0)}$  est orthogonal au premier mode.

Ainsi l'hypothèse 2.17 suffit à n'avoir pas de composante sur  $\varphi_1$  et l'application de la méthode présentée ci-dessus convergera pour donner  $\varphi_2$  et  $\omega_2$ . Dans la procédure on peut utiliser  $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$  à la place de  $\mathbf{A} = \mathbf{K}^{-1}\mathbf{M}$ . Ceci présente l'avantage d'inverser  $\mathbf{M}$  plutôt que  $\mathbf{K}$ , ce qui peut être plus aisé si  $\mathbf{M}$  est diagonale. Cependant la procédure convergera alors vers les modes associés aux fréquences les plus élevées, ce qui fait qu'elle ne peut être utilisée que pour des systèmes ayant un nombre de modes limité donc un nombre de degrés de liberté limité. Le fait de projeter sur une base qui contient un nombre de modes propres inférieurs au nombre qu'en possède le système est ce qui permet la réduction de modèle. Mais cela entraîne forcément une perte d'information, et ainsi les excitations de fréquences supérieures aux modes propres utilisées seront mal représentées. Pour atténuer l'erreur de troncature il existe des corrections tel qu'en présente [7]. Si l'amortissement n'est pas négligé,  $\mathbf{C}$  dans le cas général, n'est pas diagonale dans la base des modes propres. Mais dans certain cas on peut faire l'hypothèse de Basile qui permet d'exprimer  $\mathbf{C}$  comme une combinaison linéaire de  $\mathbf{M}$  et  $\mathbf{K}$ .

## 2.3 POD

La POD (pour Proper Orthogonal Decomposition), connue précédemment sous le nom de "Principal Component Analysis" ou "Karhunen-Loève Expansion", permet à partir d'une grande quantité de données, d'extraire des modes principaux permettant de représenter au mieux l'ensemble des données. Elle est à rapprocher de la SVD (Singular Value Decomposition) qui peut être vue comme sa forme discrète. La SVD a notamment été utilisée pour de la reconnaissance de formes et de visages.

### 2.3.1 La SVD, Singular Value Decomposition

Soit une matrice  $\mathbf{A}(n \times m)$ , il existe  $\mathbf{U}(n \times n)$ ,  $\mathbf{V}(m \times m)$  orthogonales et  $\mathbf{S}(n \times m)$  diagonale aux coefficients positifs ou nuls, tel que :

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad A_{ij} = \mathbf{u}_i \mathbf{S} \mathbf{v}_j^T \quad \text{où } \mathbf{u}_i \text{ et } \mathbf{v}_j \text{ sont les lignes de } \mathbf{U} \text{ et } \mathbf{V} \quad (2.19)$$

Cette décomposition n'est pas unique, mais pour que  $\mathbf{S}$  le soit on choisit par convention d'ordonner les valeurs sur sa diagonale de manière décroissante :  $S_{ii} \geq S_{(i+1)(i+1)}$ . Ces valeurs sont appelées valeurs singulières et pour une matrice  $\mathbf{A}$  de rang  $r$  :  $S_{ii} = 0 \forall i > r$ , on peut alors tronquer la décomposition précédente à l'ordre  $r$  (voir dans la suite). Note :  $r \leq \min(m, n)$ .

Ces valeurs singulières ne sont pas sans rappeler les valeurs propres et elles sont liées à celles-ci. On notera que les valeurs singulières peuvent se calculer pour des matrices rectangulaires alors que les valeurs propres n'ont de sens que pour des endomorphismes et donc se limitent aux matrices carrées. La relation liant les deux décompositions est que  $\mathbf{V}$  contient les vecteurs propres de  $\mathbf{A}^T \mathbf{A}$  (symétriquement  $\mathbf{U}$  contient les vecteurs propres de  $\mathbf{A} \mathbf{A}^T$ ), et les carrés des valeurs singulières de  $\mathbf{A}$  sont les modules des valeurs propres de  $\mathbf{A} \mathbf{A}^T$ . Si  $\mathbf{A}$  est symétrique définie positive alors ses valeurs propres sont ses valeurs singulières et  $\mathbf{U} = \mathbf{V}$ . Si  $\mathbf{A}$  est symétrique

mais possède des valeurs propres négatives, alors les valeurs singulières associées à ces colonnes en seront les valeurs absolues.

## La SVD Tronquée

L'intérêt d'obtenir cette décomposition est de pouvoir réduire la quantité de données en conservant au mieux les informations qu'elle contenait. Les vecteurs singuliers à gauche et à droite (respectivement les colonnes de  $\mathbf{U}$  et  $\mathbf{V}$ ) nous donnent des modes d'autant plus représentatifs de l'ensemble des données qu'ils sont associés à des valeurs singulières élevées. On choisit un ordre  $k$  pour ne garder que les plus grandes valeurs singulières et calculer :

$$\mathbf{A}^{(k)} = \mathbf{U}\mathbf{S}^{(k)}\mathbf{V}^T \quad A_{ij} = \mathbf{u}_i\mathbf{S}^{(k)}\mathbf{v}_j^T \quad (2.20)$$

où  $S_{ij}^{(k)} = S_{ij} \quad \forall i, j < k$ , et nul au delà

Ceci donne la matrice approximée  $\mathbf{A}^{(k)}$  mais une grande partie du système  $\mathbf{U}\mathbf{S}^{(k)}\mathbf{V}^T$  est alors inutile car elle n'apporte plus de contribution au calcul à cause des termes nuls de  $\mathbf{S}^{(k)}$ . On peut donc tronquer chacune des matrices. En forme matricielle on voit où agit la troncature à l'ordre  $k$  :

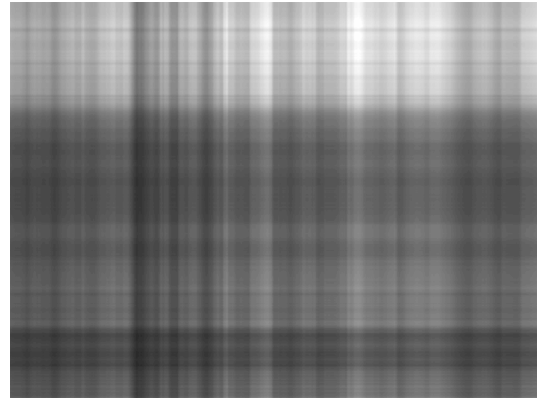
$$\begin{bmatrix} U_{11} & \cdots & U_{1k} \\ \vdots & \ddots & \vdots \\ U_{n1} & \cdots & U_{nk} \end{bmatrix} \quad (n \times k) \quad \begin{bmatrix} S_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S_{kk} \end{bmatrix} \quad \begin{bmatrix} V_{11} & \cdots & V_{m1} \\ \vdots & \ddots & \vdots \\ V_{1k} & \cdots & V_{mk} \end{bmatrix} \quad (n \times m) \quad (2.21)$$

On tronque  $\mathbf{U}$  et  $\mathbf{V}$  après leur  $k^{ieme}$  colonnes, et  $\mathbf{S}$  après sa  $k^{ieme}$  ligne et sa  $k^{ieme}$  colonne, et  $\mathbf{A}^{(k)}$  reste de la même taille.

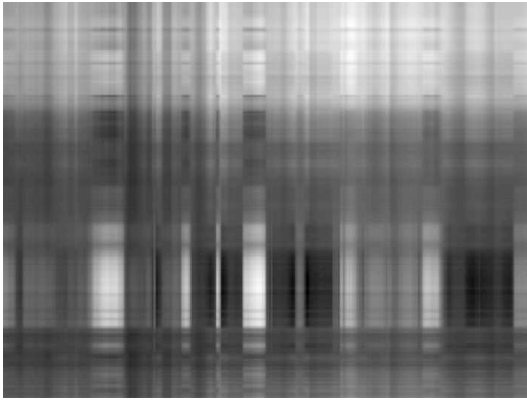
Pour illustrer, dans l'exemple qui suit, on considère une image comme une matrice qui décrit la couleur de chacun des pixels, et on lui applique la SVD. À partir des résultats plus ou moins tronqués de la SVD on obtient une compression -en terme de données- de l'image, qui a pour conséquence de représenter plus ou moins bien l'image d'origine (Le code *Matlab* est fourni en annexes).



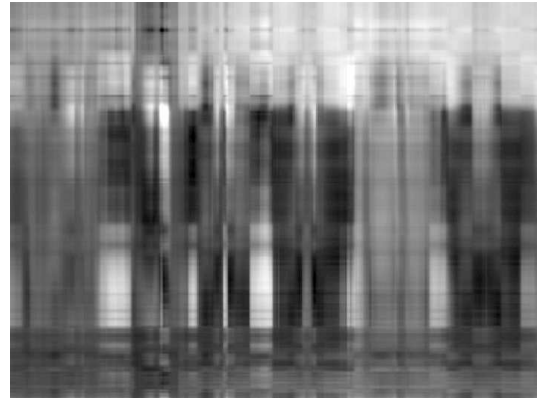
(a) Image originale



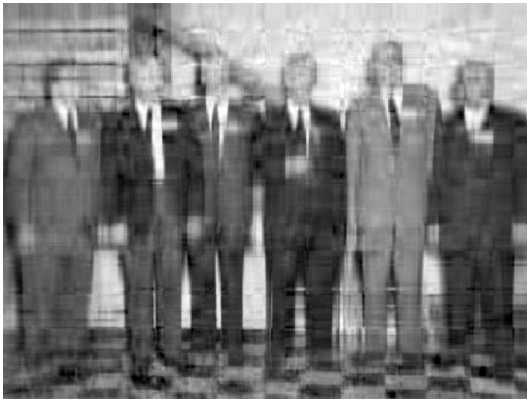
(b) Image obtenue par une troncature de la SVD à l'ordre 1



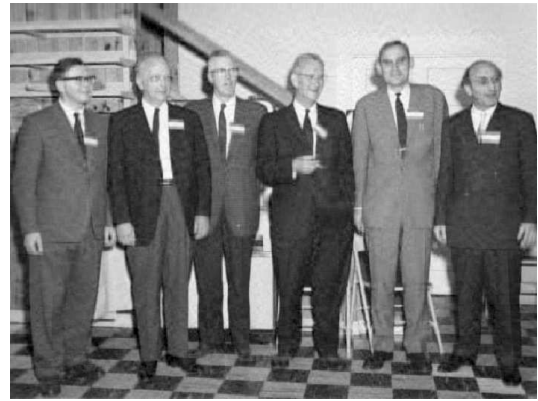
(c) à l'ordre 2



(d) à l'ordre 4



(e) à l'ordre 16



(f) à l'ordre 64



(g) à l'ordre 256



(h) Image obtenue par SVD sans troncature, rigoureusement semblable à l'originale

FIGURE 2.1 – Compression d'une image par SVD

On constate donc que plus l'ordre de la troncature de la SVD est faible plus l'image obtenue est détériorée, et on peut voir dans le programme que le poids en est évidemment réduit. Ainsi l'image obtenue grâce à 256 modes ne semble pas distinguable de l'originale (même sans le rétrécissement nécessaire pour l'afficher dans ce rapport), qui en compte 480, ceci permet donc de réaliser une compression des données jusqu'à une précision souhaitée (note : Il existe des algorithmes plus performants que la SVD pour compresser les images). Voici la répartition des valeurs singulières de l'image (affichée avec un axe des ordonnées en semi-log pour être lisible étant donnée la disparité des valeurs) :

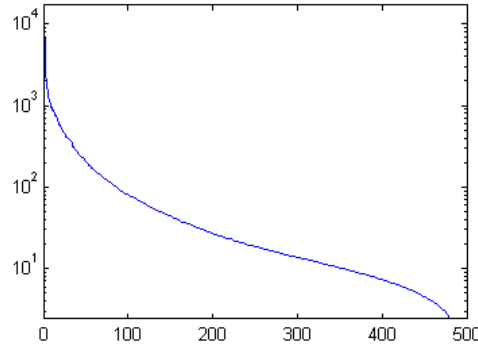


FIGURE 2.2 – Valeurs singulières

Évidemment plus les valeurs singulières décroissent rapidement, plus la troncature sera efficace.

La SVD Tronquée assure l'optimalité de l'approximation car la matrice obtenue par une troncature à l'ordre  $k$ , est la matrice de rang  $k$  qui minimise l'erreur :  $\|\mathbf{A} - \mathbf{A}^{(k)}\|_F$ . La norme de Frobenius est égale à la racine carrée de la somme des carrés de tous les termes d'une matrice (norme 2) et est aussi égale à plus grande valeur singulière.

La SVD tronquée nécessite le calcul total de la SVD, ce qui représente un nombre d'opérations numériques de l'ordre de  $O(mn^2)$  (qui est lié au temps de calcul). Et même si l'on peut choisir de transposer la matrice pour échanger le rôle de  $m$  et  $n$ , cela représente un handicap qui s'amplifie en même temps que la taille de la matrice étudiée. Pour remédier à ceci des méthodes de décomposition approchées ont été développées, les méthodes LRMA (pour Low-Rank Matrix Approximation), et notamment la méthode QUIC-SVD, qui permet de maîtriser l'erreur causée par l'approximation [8].

La POD est le plus souvent utilisée en version discrète, car les bases de données le sont (par exemple due à l'échantillonnage expérimental), ou simplement à cause de la discrétisation utilisée par un solveur numérique. Une utilisation de la version continue se trouve dans [9], en voici la description.

### 2.3.2 La formulation continue

La POD est une méthode qui propose une solution à variables séparées sous la forme d'une somme de produits de fonctions. Prenons la fonction originale  $z(x, t)$ ,

elle sera approximée sous la forme :

$$z(x, t) \approx \sum_{k=1}^m a_k(t) \varphi_k(x) \quad (2.22)$$

Dans la suite de ce rapport,  $x$  désignera la coordonnée spatiale (donc est éventuellement un vecteur) et  $t$  la coordonnée temporelle. La décomposition 2.22 n'est pas unique car il faut choisir l'espace d'approximation contenant les  $\varphi_k(x)$ . Par exemple si  $x$  est contenue dans un intervalle  $\Omega$  de réel, alors les fonctions  $\varphi_k(x)$  peuvent être choisies comme des séries de Fourier, des polynômes de Legendre, des polynômes de Tchebychev etc... Et ce choix influence évidemment le résultat obtenu pour les fonctions  $a_k(t)$ . On fait l'hypothèse que l'espace d'approximation des fonctions d'espace est bien choisi et que la somme tendra vers la solution  $z(x, t)$  quand  $m$  tend vers l'infini (plus précisément sur le domaine de définition de  $z(x, t)$ , hormis des éventuels ensembles de mesure nulle. Les lecteurs qui ne sont pas familiers avec la théorie des mesures peuvent consulter [10]. Ces considérations ne seront pas développées ici. Il en sera de même pour les questions d'intégrabilité des fonctions qui ne se posent pas en calcul discret. Si on choisit pour les fonctions  $\varphi_k(x)$  une base orthonormée :

$$\int_{\Omega} \varphi_i(x) \varphi_j(x) dx = \delta_{ij} \quad (2.23)$$

On obtient la formule suivante pour  $a_k(t)$  :

$$a_k(t) = \int_{\Omega} z(x, t) \varphi_k(x) dx \quad (2.24)$$

alors la détermination de la fonction  $a_k(t)$  ne dépend que de  $\varphi_k(x)$ .

Si la décomposition est optimale, i.e. si pour chaque terme de la somme, il n'existe pas de meilleure approximation, alors les termes sont appelés proper orthogonal nodes, et la somme proper orthogonal decomposition.

Pour faire correspondre avec la formulation discrète on pose :

$$\mathbf{A} = \mathbf{Q} \mathbf{V}^T = \sum_{i=1}^m \mathbf{q}_i \mathbf{v}_i^T \text{ où } \mathbf{Q} = \mathbf{U} \mathbf{S} \quad (2.25)$$

Dans cette forme discrète de l'équation 2.22, la matrice  $\mathbf{A}$  représente  $z(x, t)$  et les colonnes  $\mathbf{q}_i$  représentent les fonctions  $a_k(t)$ . Le lien entre valeurs singulières et valeurs propres a été décrit précédemment, et ce rapprochement peut servir à la résolution du problème ; On définit alors  $\mathbf{B} = \mathbf{A}^T \mathbf{A}$ , et on aura dans le cas continu en temps mais décrit de manière discrète sur l'espace :

$$B_{ij} = \frac{1}{T} \int_{t_0}^{t_0+T} z(x_i, t) z(x_j, t) dt \quad (2.26)$$

Et si l'on passe à une formulation en dimension infinie, il y a une infinité de valeurs propres et  $B$  n'est plus une matrice mais une fonction de deux variables :

$$B(x_1, x_2) = \frac{1}{T} \int_{t_0}^{t_0+T} z(x_1, t) z(x_2, t) dt \quad (2.27)$$

où  $x_1$  et  $x_2$  sont des variables continues définies sur  $\Omega$ .

Soit  $\psi$  un vecteur propre de  $\mathbf{B}$ , de valeur propre associée  $\lambda$ , alors  $\mathbf{B}\psi = \lambda\psi$ . Ce qui s'écrit sous forme discrète :

$$\sum_{j=1}^m B_{ij} \psi_j^T = \lambda \psi_i \quad (2.28)$$

ce qui devient une intégrale en forme continue :

$$\int_{\Omega} B(x_1, x_2) \psi(x_2) dx_2 = \lambda \psi(x_1) \quad (2.29)$$

Un des problèmes posés par cette réduction de modèles "a posteriori" est qu'elle dépend de la solution obtenue "a priori". En effet les modes trouvés sont représentatifs des données de départ, et si au cours d'un calcul la configuration du système considéré devait s'éloigner du résultat préalable, les modes ne seraient plus représentatifs de la solution. La représentation à variables séparées utilisée par la POD possède des limites, en particulier la mauvaise représentation d'une turbulence locale qui se déplace, typiquement : une onde de se propageant (la solution de l'équation de d'Alembert n'est pas à variable séparables :  $f(x \pm \beta t)$ ).

## 2.4 PGD

Comme la POD, la PDG approxime la solution à l'aide de l'hypothèse de séparation des variables comme une somme de produits de fonctions ; prenons la fonction solution  $z(x, t)$ , elle sera approximée sous la forme :

$$z(x, t) \approx \sum_{k=1}^m a_k(t) \varphi_k(x) \quad (2.30)$$

Mais à la différence de la POD, la fonction  $z(x, t)$  est inconnue au préalable, on ne connaît alors pas les modes et on les calcule "à la volée", grâce à l'algorithme appelé algorithme des puissances itérées qui est le suivant :

---

### Algorithme 1 Résolution PGD

---

```

1: for  $m = 1$  à  $m_{max}$  do
2:   initialiser  $a_1$ 
3:   for  $k = 1$  à  $k_{max}$  do
4:      $\varphi_k \leftarrow S_m(a_k)$ 
5:     normer  $\varphi_k$  ( $\|\varphi_k\|_{\Omega} = 1$ )
6:      $a_k \leftarrow T_m(\varphi_k)$ 
7:     vérifier la convergence de  $(\varphi_k a_k)$ 
8:   end for
9:    $\varphi_m \leftarrow \varphi_k$  et  $a_m \leftarrow a_k$ 
10:   $z_m = z_{m-1} + \varphi_m a_m$ 
11: end for
```

---

Où  $S_m$  est l'application qui associe à une fonction du temps  $a$ , une fonction de l'espace  $\varphi$ . Et  $T_m$  l'application qui associe à une fonction spatiale  $\varphi$ , une fonction

temporelle  $a$ . On part de la formulation faible d'un problème :

$$B(u, v) = L(v) \quad \forall v \in H \quad (2.31)$$

Sous les hypothèses du théorème de Lax-Milgram :

- $H$  un espace de Hilbert
  - $B$  une forme bilinéaire, continue et coercive sur  $H$
  - $L$  une forme linéaire continue sur  $H$
- il existe un unique  $u \in H$  solution de l'équation 2.31.

On applique alors la démarche classique pour la PGD, appelée Galerkin progressive. On suppose  $z_{m-1}$  (l'approximation de  $z$  à l'ordre  $m-1$ ) connue. Un nouveau couple  $(\varphi, a) \in U \otimes V$  est défini pour la décomposition d'ordre  $m$  comme celui qui vérifie le critère de double orthogonalité de Galerkin suivant :

$$B(z_{m-1} + \varphi a, \varphi a^* + \varphi^* a) = L(\varphi a^* + \varphi^* a) \quad \forall \varphi^* \in U \text{ et } \forall a^* \in V \quad (2.32)$$

Alors  $S_m(a)$  est définie par :

$$B(z_{m-1} + \varphi a, \varphi^* a) = L(\varphi^* a) \quad \forall \varphi^* \in U \quad (2.33)$$

et  $T_m(\varphi)$  par :

$$B(z_{m-1} + \varphi a, \varphi a^*) = L(\varphi a^*) \quad \forall a^* \in V \quad (2.34)$$

Le couple  $(\varphi, a)$  vérifie l'équation 2.32 si et seulement si  $\varphi = S_m(a)$  et  $a = T_m(\varphi)$ , qui est un problème non-linéaire, dont la résolution par la PGD peut être vue comme un problème aux valeurs propres où  $\varphi$  et  $a$  sont respectivement les fonctions propres dominantes de  $S_m \circ T_m$  et  $T_m \circ S_m$ . Ce qui permet en s'inspirant d'algorithmes utilisés dans les problèmes aux valeurs propres d'obtenir l'algorithme 1, qui donne la décomposition. Cet algorithme peut être modifié, notamment par l'ajout de l'orthogonalisation qui peut généralement améliorer les résultats (dans le cas où il n'y a que deux variables). S'il on veut orthogonaliser  $\varphi_m$  par rapport à la base spatiale existante  $\{\varphi_1, \varphi_2, \dots, \varphi_{m-1}\}$ , il faut définir un produit scalaire sur  $\Omega$ , par exemple :

$$\langle u, v \rangle_\Omega = \int_\Omega uv \, d\Omega \quad (2.35)$$

Une autre modification consiste à introduire une fonction  $T(\mathbf{p}_m) = \boldsymbol{\alpha}_m$ , où  $\mathbf{p}_m$  est un vecteur qui contient les  $\varphi_m$ , et  $\boldsymbol{\alpha}_m$  un vecteur contenant les  $a_m$ . Permettant ainsi la mise à jour de l'ensemble des fonctions du temps après chaque calcul d'un nouveau couple  $(\varphi_m, a_m)$ , en remplaçant les lignes 9 et 10 de l'algorithme 1 par :

$$\begin{aligned} \varphi_m &\leftarrow \varphi_k \\ \boldsymbol{\alpha}_m &\leftarrow T(\varphi_m) \\ z_m &= \mathbf{p}_m \cdot \boldsymbol{\alpha}_m \end{aligned}$$

Pour voir un exemple d'utilisation de la méthode, [11] propose une résolution d'un problème thermique avec la PGD. Le calcul pour un problème dynamique sera détailler dans une autre partie.



# Chapitre 3

## Le calcul dynamique

L'équation de la dynamique 2.1, est valable à tout instant, rappel :

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}$$

Il s'agit d'une équation différentielle en  $\mathbf{u}$ . Pour résoudre une équation différentielle il existe une méthode dite de résolution par quadrature. Cette méthode fournit une solution exacte et explicite pour l'équation, le problème est que cette résolution n'est possible que dans quelques cas classiques et elle est donc impossible dans le cas général. Dans ce cas on résout l'équation par une méthode numérique, cela induit la discrétisation de l'intervalle d'étude. En espace la discrétisation se traduit par le maillage, ce qui permet de définir les quantités dépendantes de l'espace comme des matrices. Le temps ne se représentant que sur une dimension, il suffit de choisir un pas de temps fixe pour découper le temps d'étude, et le représenter plus ou moins finement.

### 3.1 Les schémas d'intégration

Le fait de découper l'axe temporelle permet de résoudre l'équation à un instant donné (à condition de connaître la solution à l'instant précédent\*). Pour ce faire il faut définir les liens entre les différentes dérivées de  $\mathbf{u}$ , on choisit alors un schéma d'intégration.

\*Si un schéma n'utilise que la solution au pas de temps précédent et pas aux pas de temps antérieurs à celui-ci, on dit que ce schéma est à pas unique, sinon il est dit à pas multiples.

Pour décrire la dérivée d'une fonction discrète on peut utiliser les schémas d'Euler, de Runge-Kutta ou de Crank-Nicholson. Dans le cas particulier de la dynamique on utilise la méthode de Newmark qui définit les liens entre accélération, vitesse et déplacement. Ces liens sont définis dans deux équations comportant des paramètres qui peuvent rendre l'algorithme explicite ou implicite, et qui influencent sa stabilité.

Un schéma est dit stable si il contrôle l'accumulation d'erreur de calcul (arrondis). Certains schémas sont conditionnellement stables et la solution qu'ils fournissent peut alors diverger. La condition de stabilité en dynamique et en générale une condition imposée à la discrétisation en temps : Choisir un pas de temps qui ne soit pas trop petit pour définir la propagation de l'information dans le maillage. La taille critique

en dessous de laquelle le pas de temps ne doit pas se trouver est donc dépendante de la taille de maille.

### 3.1.1 Schémas de Newmark

Le schéma d'intégration est donné par les deux équations :

$$\mathbf{u}_{n+1} = \mathbf{u}_n + dt \cdot \dot{\mathbf{u}}_n + \frac{dt^2}{2} [(1 - 2\beta)\ddot{\mathbf{u}}_n + 2\beta\ddot{\mathbf{u}}_{n+1}] \quad (3.1)$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + dt[(1 - \gamma)\ddot{\mathbf{u}}_n + \gamma\ddot{\mathbf{u}}_{n+1}] \quad (3.2)$$

La stabilité du schéma dépend des deux paramètres  $\beta$  et  $\gamma$  :

$$\begin{array}{ll} \gamma \leq 1/2 & \text{Instable} \\ \gamma \geq 1/2 \text{ et } 2\beta \leq \gamma & \text{conditionnellement stable} \\ \gamma \geq 1/2 \text{ et } 2\beta \geq \gamma & \text{inconditionnellement stable} \end{array}$$

Pour des valeurs particulières des paramètres on retrouve des méthodes classiques :

Nom	$\gamma$	$\beta$	Propriétés
Différences Centrées	1/2	0	explicite et conditionnellement stable
Fox Goodwin	1/2	1/12	conditionnellement stable
Accélération linéaire	1/2	1/6	conditionnellement stable
Accélération Moyenne	1/2	1/4	inconditionnellement stable

L'équation de la dynamique peut être formulée ainsi :

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \quad (3.3)$$

Ce qui donne avec les équations 3.1 et 3.2, si  $\mathbf{K}$  ne dépend pas de  $\mathbf{u}$  :

$$\overbrace{(\mathbf{M} + \mathbf{C}\gamma dt + \mathbf{K}\beta dt^2)}^{\mathbf{S}} \ddot{\mathbf{u}}_{n+1} = \mathbf{f}_{n+1} - \mathbf{C} \overbrace{[\dot{\mathbf{u}}_n + dt(1 - \gamma)\ddot{\mathbf{u}}_n]}^{\dot{\mathbf{u}}_{n+1}^p} - \mathbf{K} \overbrace{[\mathbf{u}_n + dt\dot{\mathbf{u}}_n + \frac{dt^2}{2}(1 - 2\beta)\ddot{\mathbf{u}}_n]}^{\mathbf{u}_{n+1}^p} \quad (3.4)$$

Permettant de calculer en trois étapes, les valeurs de  $\ddot{\mathbf{u}}$ ,  $\dot{\mathbf{u}}$  et  $\mathbf{u}$  :

$$\begin{cases} \ddot{\mathbf{u}}_{n+1} = \mathbf{S}^{-1}(\mathbf{f}_{n+1} - \mathbf{C}\dot{\mathbf{u}}_{n+1}^p - \mathbf{K}\mathbf{u}_{n+1}^p) \\ \dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_{n+1}^p + dt\gamma\ddot{\mathbf{u}}_{n+1} \\ \mathbf{u}_{n+1} = \mathbf{u}_{n+1}^p + dt^2\beta\ddot{\mathbf{u}}_{n+1} \end{cases} \quad (3.5)$$

### 3.1.2 HHT

La méthode HHT (ou HHT- $\alpha$ ) qui se base sur la méthode de Newmark, a été introduite dans [12] pour rajouter un paramètre  $\alpha$  qui permet d'introduire un amortissement numérique.

Le schéma d'intégration est toujours défini par 3.1 et 3.2, mais il y a cette fois trois paramètres à choisir, que l'on peut réduire à un choix sur  $\alpha$  uniquement, en suivant les conseils donnés par [12] :

$$\begin{cases} \beta = (1 - \alpha)^2/4 \\ \gamma = 1/2 - \alpha \\ \alpha \in [-1/3; 0] \end{cases} \quad (3.6)$$

L'équation de la dynamique donnée dans [13] est formulée ainsi :

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} + (1 + \alpha)\mathbf{C}\dot{\mathbf{u}}_{n+1} - \alpha\mathbf{C}\dot{\mathbf{u}}_n + (1 + \alpha)\mathbf{K}\mathbf{u}_{n+1} - \alpha\mathbf{K}\mathbf{u}_n = (1 + \alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_n \quad (3.7)$$

Ce qui donne, si  $\mathbf{K}$  ne dépend pas de  $\mathbf{u}$  :

$$\left\{ \begin{array}{l} \mathbf{M}\ddot{\mathbf{u}}_{n+1} + (1 + \alpha)\mathbf{C}\dot{\mathbf{u}}_n + (1 + \alpha)\mathbf{C}dt(1 - \gamma)\ddot{\mathbf{u}}_n + (1 + \alpha)\mathbf{C}dt\gamma\ddot{\mathbf{u}}_{n+1} - \alpha\mathbf{C}\dot{\mathbf{u}}_n \\ + (1 + \alpha)\mathbf{K}\mathbf{u}_n + (1 + \alpha)\mathbf{K}dt\dot{\mathbf{u}}_n + (1 + \alpha)\mathbf{K}\frac{dt^2}{2}(1 - 2\beta)\ddot{\mathbf{u}}_n + (1 + \alpha)\mathbf{K}\frac{dt^2}{2}2\beta\ddot{\mathbf{u}}_{n+1} - \alpha\mathbf{K}\mathbf{u}_n \\ = (1 + \alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_n \\ \text{Soit, en factorisant les inconnues :} \\ \overbrace{[\mathbf{M} + (1 + \alpha)(\mathbf{C}dt\gamma + \mathbf{K}dt^2\beta)]}^{\mathbf{S}_{\text{HHT}}} \ddot{\mathbf{u}}_{n+1} \\ + (1 + \alpha)[\mathbf{C}dt(1 - \gamma) + \mathbf{K}\frac{dt^2}{2}(1 - 2\beta)]\ddot{\mathbf{u}}_n \\ + [(1 + \alpha)(\mathbf{C} + \mathbf{K}dt) - \alpha\mathbf{C}]\dot{\mathbf{u}}_n \\ + [(1 + \alpha)\mathbf{K} - \alpha\mathbf{K}]\mathbf{u}_n \\ = (1 + \alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_n \\ \text{Soit, en factorisant les matrices :} \\ \overbrace{\mathbf{S}_{\text{HHT}}\ddot{\mathbf{u}}_{n+1} + \mathbf{C}[\dot{\mathbf{u}}_n + (1 + \alpha)dt(1 - \gamma)\ddot{\mathbf{u}}_n]}^{\dot{\mathbf{u}}_{n+1}^{\text{pHHT}}} + \overbrace{\mathbf{K}[\mathbf{u}_n + (1 + \alpha)(dt\dot{\mathbf{u}}_n + \frac{dt^2}{2}(1 - 2\beta)\ddot{\mathbf{u}}_n)]}^{\mathbf{u}_{n+1}^{\text{pHHT}}} \\ = (1 + \alpha)\mathbf{f}_{n+1} - \alpha\mathbf{f}_n \end{array} \right. \quad (3.8)$$

Permettant de calculer en trois étapes :

$$\left\{ \begin{array}{l} \ddot{\mathbf{u}}_{n+1} = \mathbf{S}_{\text{HHT}}^{-1}(\mathbf{f}_{n+1}^{\text{HHT}} - \mathbf{C}\dot{\mathbf{u}}_{n+1}^{\text{pHHT}} - \mathbf{K}\mathbf{u}_{n+1}^{\text{pHHT}}) \\ \dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + dt[(1 - \gamma)\ddot{\mathbf{u}}_n + \gamma\ddot{\mathbf{u}}_{n+1}] \\ \mathbf{u}_{n+1} = \mathbf{u}_n + dt.\dot{\mathbf{u}}_n + \frac{dt^2}{2}[(1 - 2\beta)\ddot{\mathbf{u}}_n + 2\beta\ddot{\mathbf{u}}_{n+1}] \end{array} \right. \quad (3.9)$$

avec :

$$\mathbf{f}_{n+1}^{\text{HHT}} = (1 + \alpha)\mathbf{f}_{n+1} - \alpha\mathbf{f}_n \quad (3.10)$$

Si on ajoute un terme non-linéaire, on peut le représenter par l'ajout d'un élément au second membre  $\mathbf{f}$ , ou recalculer la matrice  $\mathbf{K}$ .

L'équation de la dynamique devient :

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} + (1 + \alpha)\mathbf{C}\dot{\mathbf{u}}_{n+1} - \alpha\mathbf{C}\dot{\mathbf{u}}_n + (1 + \alpha)\mathbf{K}\mathbf{u}_{n+1} - \alpha\mathbf{K}\mathbf{u}_n = (1 + \alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_n - \underline{((1 + \alpha)f_{(u_{n+1})} - \alpha f_{(u_n)})} \quad (3.11)$$

Ou :

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} + (1 + \alpha)\mathbf{C}\dot{\mathbf{u}}_{n+1} - \alpha\mathbf{C}\dot{\mathbf{u}}_n + \underline{(1 + \alpha)\mathbf{K}_{(u_{n+1})}\mathbf{u}_{n+1} - \alpha\mathbf{K}_{(u_n)}\mathbf{u}_n} = (1 + \alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_n \quad (3.12)$$

Si l'on choisit la représentation 3.11 il suffit d'ajouter le terme souligné au second membre. Si l'on choisit la représentation 3.12 les équations sont modifiées ainsi :

\* Pour simplifier on notera  $\mathbf{K} = \mathbf{K}_{(u_{n+1})}$  et on obtient des équations légèrement modifiées par les termes soulignés.

$$\left\{ \begin{array}{l}
\mathbf{M}\ddot{\mathbf{u}}_{n+1} + (1+\alpha)\mathbf{C}\dot{\mathbf{u}}_{\mathbf{n}} + (1+\alpha)\mathbf{C}dt(1-\gamma)\ddot{\mathbf{u}}_{\mathbf{n}} + (1+\alpha)\mathbf{C}dt\gamma\ddot{\mathbf{u}}_{n+1} - \alpha\mathbf{C}\dot{\mathbf{u}}_{\mathbf{n}} \\
+ (1+\alpha)\mathbf{K}\mathbf{u}_{\mathbf{n}} + (1+\alpha)\mathbf{K}dt\dot{\mathbf{u}}_{\mathbf{n}} + (1+\alpha)\mathbf{K}\frac{dt^2}{2}(1-2\beta)\ddot{\mathbf{u}}_{\mathbf{n}} + (1+\alpha)\mathbf{K}\frac{dt^2}{2}2\beta\ddot{\mathbf{u}}_{n+1} - \underline{\alpha\mathbf{K}_{(u_n)}\mathbf{u}_{\mathbf{n}}} \\
= (1+\alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_{\mathbf{n}} \\
\text{Soit, en factorisant les inconnues :} \\
\overbrace{[\mathbf{M} + (1+\alpha)(\mathbf{C}dt\gamma + \mathbf{K}dt^2\beta)]}^{\mathbf{S}_{\text{HHT}}} \ddot{\mathbf{u}}_{n+1} \\
+ (1+\alpha)[\mathbf{C}dt(1-\gamma) + \mathbf{K}\frac{dt^2}{2}(1-2\beta)]\ddot{\mathbf{u}}_{\mathbf{n}} \\
+ [(1+\alpha)(\mathbf{C} + \mathbf{K}dt) - \alpha\mathbf{C}]\dot{\mathbf{u}}_{\mathbf{n}} \\
+ [(1+\alpha)\mathbf{K} - \underline{\alpha\mathbf{K}_{(u_n)}}]\mathbf{u}_{\mathbf{n}} \\
= (1+\alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_{\mathbf{n}} \\
\text{Soit, en factorisant les matrices :} \\
\overbrace{\ddot{\mathbf{u}}_{n+1}}^{\mathbf{u}_{n+1}^{\text{pHHT}}} \overbrace{[\mathbf{S}_{\text{HHT}} + \mathbf{C}(\dot{\mathbf{u}}_{\mathbf{n}} + (1+\alpha)dt(1-\gamma)\ddot{\mathbf{u}}_{\mathbf{n}}) + \mathbf{K}(u_{\mathbf{n}} + (1+\alpha)(dt\dot{\mathbf{u}}_{\mathbf{n}} + \frac{dt^2}{2}(1-2\beta)\ddot{\mathbf{u}}_{\mathbf{n}})]}^{\mathbf{u}_{n+1}^{\text{pHHT}}} \\
= (1+\alpha)\mathbf{f}_{n+1} - \alpha\mathbf{f}_{\mathbf{n}} - \underline{\alpha(\mathbf{K} - \mathbf{K}_{(u_n)})\mathbf{u}_{\mathbf{n}}}
\end{array} \right. \quad (3.13)$$

Et on retrouve les équations 3.9 mais cette fois :

$$\mathbf{f}_{n+1}^{\text{HHT}} = (1+\alpha)\mathbf{f}_{n+1} - \alpha\mathbf{f}_{\mathbf{n}} - \underline{\alpha(\mathbf{K} - \mathbf{K}_{(u_n)})\mathbf{u}_{\mathbf{n}}} \quad (3.14)$$

## 3.2 La PGD

### 3.2.1 Définition du problème

- Admissibilité cinématique :  
 $U \in [H^1(\Omega)]^3, U|_{\partial\Omega_U} = U_d$
- Admissibilité statique :  
 $\text{div}\sigma + f_v = \rho\ddot{U} + \mu\dot{U}$  sur  $\Omega$  et  $\sigma.n|_{\partial\Omega_f} = f_d$
- Relation de comportement :  
 $\sigma = H : \varepsilon(U)$  sur  $\Omega$
- Conditions initiales :  
 $U(0) = U_0$  et  $\dot{U}(0) = \dot{U}_0$

Formulation variationnelle :

$$\forall U^* \text{ CA0, } \int_{\Omega} [\varepsilon(U^*) : \sigma + U^* \mu \dot{U} + U^* \rho \ddot{U} - U^* f_d] d\Omega - \int_{\partial\Omega_f} U^* f_d ds - \underbrace{\int_{\partial\Omega_U} U^* \sigma.n ds}_{=0} = 0 \quad (3.15)$$

Où

- $U$  est le déplacement
- $\Omega$  l'espace, avec le bord  $\partial\Omega = \partial\Omega_U \cup \partial\Omega_f$
- $\sigma$  le tenseur des contraintes
- $\varepsilon$  le tenseur des déformations
- $\rho$  la masse volumique
- $\mu$  l'amortissement volumique (cette donnée est généralement mal connue)
- $H$  le tenseur de Hook
- $f_v$  effort volumique

—  $f_d$  effort imposé sur  $\partial\Omega_f$

Mise en place de la PGD, utilisation de l'hypothèse des variables séparables :

$$U(x, t, \theta) = \sum_{k=1}^n \varphi_k(x) g_k(t) h_k(\theta) \quad (3.16)$$

Où

- $x$  est la variable d'espace
- $t$  la variable de temps
- $\theta$  une variable paramètre utilisée à titre d'exemple

Pour chaque  $k$ , il faut discrétiser les fonctions sur leur domaine de définition, ici la discrétisation en espace donne :

$$\varphi(x) = \sum_{i=1}^{Nbc_x} N_i(x) \varphi_i = N_\varphi(x) \boldsymbol{\varphi}_q \quad (3.17)$$

$Nbc_x$  représente le nombre de composante de discrétisation spatiale, l'indice  $q$  représente le vecteur sur le domaine discrétisé.

On remplace dans la formulation variationnelle, en choisissant  $U^* = (\varphi gh)^* = (\varphi^* gh + \varphi g^* h + \varphi gh^*)$ .

On rassemble les termes en  $N_\varphi(x)$  ce qui permet de définir :

$$\mathbf{K} = \int_{\Omega} \varepsilon(N_\varphi(x)) : H : \varepsilon(N_\varphi(x)) \, dx \quad (3.18)$$

$$\mathbf{C} = \int_{\Omega} N_\varphi(x) \mu N_\varphi(x) \, dx \quad (3.19)$$

$$\mathbf{M} = \int_{\Omega} N_\varphi(x) \rho N_\varphi(x) \, dx \quad (3.20)$$

$$\mathbf{f} = \int_{\Omega} N_\varphi(x) f_v \, dx + \int_{\partial\Omega_f} N_\varphi(x) f_d \, ds dt d\theta \quad (3.21)$$

Les calculs sont déroulés en annexes et donnent :

$$\begin{aligned} \int_T \int_{\Theta} (\varphi_q^* gh + \varphi_q g^* h + \varphi_q gh^*)^T & \left[ \mathbf{K} \varphi_q gh + \mathbf{C} \varphi_q \dot{g} h + \mathbf{M} \varphi_q \ddot{g} h \right. \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k \\ & \left. - \mathbf{f} \right] dt d\theta = 0 \end{aligned} \quad (3.22)$$

Ce qui permet de dégager un problème à résoudre pour chaque variable.

### 3.2.2 Problème en espace

$$\begin{aligned} \forall \varphi_q^* \quad \int_T \int_{\Theta} \varphi_q^{*T} gh & \left[ \mathbf{K} \varphi_q gh + \mathbf{C} \varphi_q \dot{g} h + \mathbf{M} \varphi_q \ddot{g} h \right. \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k \\ & \left. - \mathbf{f} \right] dt d\theta = 0 \end{aligned} \quad (3.23)$$

Donc :

$$\int_T \int_{\Theta} gh [ \mathbf{K} \varphi_q gh + \mathbf{C} \varphi_q \dot{g}h + \mathbf{M} \varphi_q \ddot{g}h + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f} ] dtd\theta = 0 \quad (3.24)$$

Soit :

$$\begin{aligned} & \int_T \int_{\Theta} gh [ \mathbf{K} gh + \mathbf{C} \dot{g}h + \mathbf{M} \ddot{g}h ] dtd\theta \varphi_q \\ = & - \int_T \int_{\Theta} gh [ \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f} ] dtd\theta \end{aligned} \quad (3.25)$$

Système auquel il faut rajouter les conditions limites en déplacement  $U_d$  par une méthode quelconque : substitution, multiplicateur de Lagrange... Dans le deuxième cas on change :

$$\mathbf{M} = \begin{bmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{C} & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} \mathbf{K} & \mathbf{D}^T \\ \mathbf{D} & 0 \end{bmatrix}, \varphi_q = \begin{bmatrix} \varphi_q \\ \lambda \end{bmatrix} \text{ et } \mathbf{f} = \begin{bmatrix} \mathbf{f} \\ \mathbf{b} \end{bmatrix} \quad (3.26)$$

Où  $\mathbf{D}$  et  $\mathbf{b}$  représentent les déplacement imposés de la manière suivante :  $\mathbf{D}\varphi_q = \mathbf{b}$ .

### 3.2.3 Problème en temps

$$\forall g^* \quad \int_T \int_{\Theta} \varphi_q^T g^* h [ \mathbf{K} \varphi_q gh + \mathbf{C} \varphi_q \dot{g}h + \mathbf{M} \varphi_q \ddot{g}h + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f} ] dtd\theta = 0 \quad (3.27)$$

Donc :

$$\int_{\Theta} \varphi_q^T h [ \mathbf{K} \varphi_q gh + \mathbf{C} \varphi_q \dot{g}h + \mathbf{M} \varphi_q \ddot{g}h + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f} ] d\theta = 0 \quad (3.28)$$

Soit :

$$\begin{aligned} & \int_{\Theta} \varphi_q^T h [ \mathbf{K} \varphi_q h ] d\theta g + \int_{\Theta} \varphi_q^T h [ \mathbf{C} \varphi_q h ] d\theta \dot{g} + \int_{\Theta} \varphi_q^T h [ \mathbf{M} \varphi_q h ] d\theta \ddot{g} \\ = & - \int_{\Theta} \varphi_q^T h [ \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f} ] d\theta \end{aligned} \quad (3.29)$$

On se trouve donc en présence d'une équation différentielle en  $g$ , que l'on peut résoudre comme on résout classiquement un problème de dynamique, i.e. en utilisant un schéma d'intégration comme il en est question dans la section précédente. On peut aussi choisir résoudre en utilisant des élément finis temporels, le calcul est fournis en annexes.

### 3.2.4 Problème en paramètre

$$\begin{aligned} \forall h^* \quad \int_T \int_{\Theta} \varphi_q^T g h^* [ & \mathbf{K} \varphi_q g h + \mathbf{C} \varphi_q \dot{g} h + \mathbf{M} \varphi_q \ddot{g} h \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k \\ & - \mathbf{f} ] dt d\theta = 0 \end{aligned} \quad (3.30)$$

Donc :

$$\begin{aligned} \int_T \varphi_q^T g [ & \mathbf{K} \varphi_q g h + \mathbf{C} \varphi_q \dot{g} h + \mathbf{M} \varphi_q \ddot{g} h \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k \\ & - \mathbf{f} ] dt = 0 \end{aligned} \quad (3.31)$$

Soit :

$$\begin{aligned} & \int_T \varphi_q^T g [ \mathbf{K} \varphi_q g + \mathbf{C} \varphi_q \dot{g} + \mathbf{M} \varphi_q \ddot{g} ] dt h \\ = & - \int_T \varphi_q^T g [ \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f} ] dt \end{aligned} \quad (3.32)$$

# Chapitre 4

## Programmation

### 4.1 Plan du programme

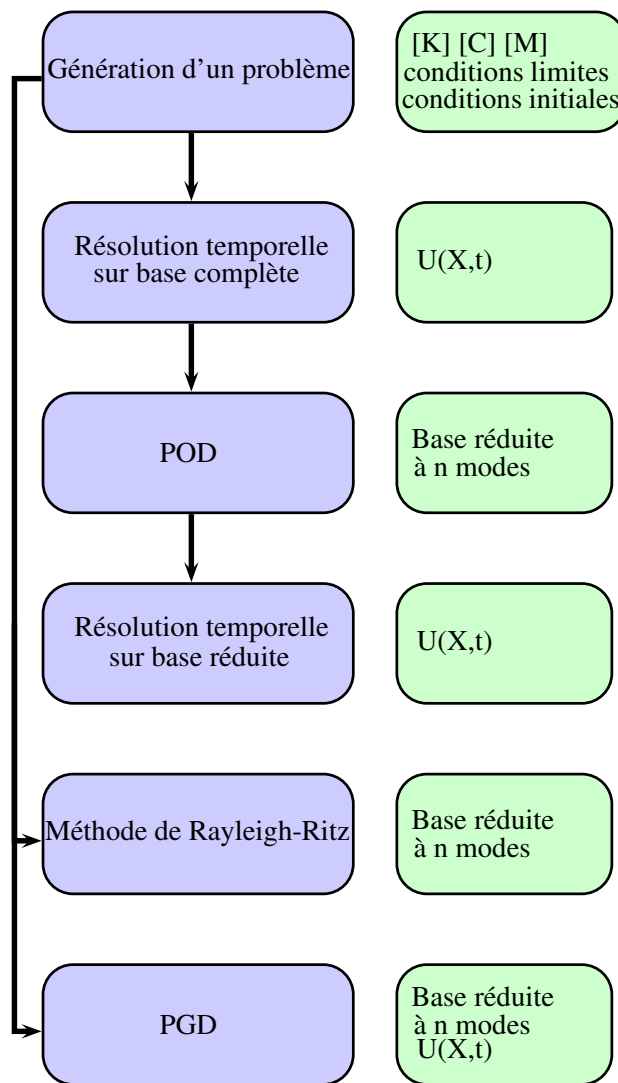


FIGURE 4.1 – Plan du Programme *Matlab*



## 4.2 Génération d'un problème

Le programme commence par générer un problème mécanique pour pouvoir lui appliquer différentes méthodes de résolution. Par simplicité et pour pouvoir visualiser simplement les résultats dépendants du temps, le problème de base est un problème poutre.

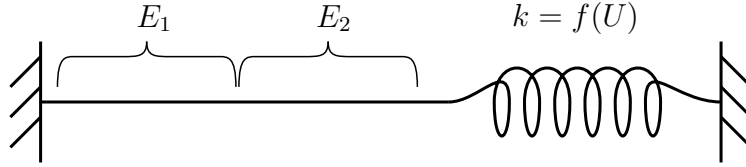


FIGURE 4.2 – Système mécanique du problème type

Comme on voit sur la figure 4.2, il s'agit d'une poutre encastrée à une extrémité et associée à un ressort sur l'autre. Le ressort peut éventuellement avoir une raideur dépendante du déplacement du bout de la poutre et ainsi introduire de la non-linéarité, ou il peut avoir une raideur nul pour simuler une poutre encastrée-libre. Il est aussi possible de séparer la poutre en plusieurs parties pour pouvoir leur choisir des propriétés (comme le module d'Young) différentes, et chaque partie est divisée en éléments.

## 4.3 Résolution temporelle sur base complète

Il s'agit d'une fonction utilisant en paramètres les objets définis par la génération de problème : raideur, amortissement, masse, efforts imposés au cours du temps et déplacement imposés au cours du temps. Elle prend aussi en paramètre le choix du schémas d'intégration dont les possibilités sont discuté en partie 3.1. En sortie de la fonction sont générées des matrices contenant respectivement le déplacement, la vitesse et l'accélération en tout temps. Peuvent aussi être générées des données énergétique pour vérifier le bon fonctionnement du calcul.

## 4.4 POD

Le calcul de la base POD est assez simple puisqu'il s'agit d'appliquer la SVD au résultat trouvé précédemment. Ce qui nous donne des modes en espaces, des modes en temps et des valeurs singulières. Les premiers sont utilisés pour construire une base réduite en fixant un nombre de modes. Les modes en temps permettent de constater le rôle de l'amortissement (numérique ou matériau) et d'en comparer l'influence selon la fréquence. Les valeurs singulières peuvent donner un aperçu de la qualité de la projection, comme il en est question dans la partie 2.3.1.

## 4.5 Résolution temporelle sur base réduite

La résolution sur la nouvelle base se fait en utilisant la même fonction que sur base complète simplement en projetant au préalable les données : raideur, amortissement,

masse, efforts et déplacements imposés. On peut constater que pour représenter ces derniers sur une base réduite, la projection peut détériorer grandement le calcul si le "mode" de déplacement imposé est mal représenté sur la base. Il a donc été choisi de rajouter à la base réduite les "modes" de déplacements imposés. Si le déplacement est imposé à un seul nœud, le mode ajouté est nul sur tous les nœuds sauf à celui où le déplacement est fixé, et ceci peut être fait automatiquement à partir des conditions limites.

#### 4.5.1 Indicateurs d'erreur

Après avoir calculé une deuxième solution, il est nécessaire de pouvoir les comparer. Au fil de la programmation différents indicateurs d'erreur ont été mis en place :

- $s_{Ref}$  est la solution de référence
- $s_{Cal}$  est la solution calculée par la résolution à étudier
- $e_{Amp}$  est l'erreur commise sur l'amplitude totale de la solution
- $e_{Max}$  est l'écart local maximal avec la solution de référence
- $e_{Vol}$  est une indication globale en intégrant l'écart au carré.
- $max(\bullet) = max_{T,\Omega}(\bullet)$
- $min(\bullet) = min_{T,\Omega}(\bullet)$

$$e_{Amp} = \left| \frac{max(s_{Ref}) - min(s_{Ref}) - [max(s_{Cal}) - min(s_{Cal})]}{max(s_{Ref}) - min(s_{Ref})} \right| \quad (4.1)$$

$$e_{Max} = \left| \frac{max(s_{Ref} - s_{Cal})}{max(s_{Ref}) - min(s_{Ref})} \right| \quad (4.2)$$

$$e_{Vol} = \left[ \frac{\int_T \int_\Omega (s_{Ref} - s_{Cal})^2 dx dt}{\int_T \int_\Omega (s_{Ref})^2 dx dt} \right]^{\frac{1}{2}} \quad (4.3)$$

#### 4.5.2 Post-traitement

Pour visualiser les résultats, *Matlab* fournit une panoplie d'outils graphique qui permettent notamment au programme d'afficher une animation pour visualiser de manière efficace la propagation de la déformation à la fois dans le modèle complet et dans le modèle réduit. Le programme peut aussi afficher sous forme de courbes les modes obtenus par les différentes méthodes, ou afficher des surfaces qui décrivent le déplacement en fonction du temps et de l'espace. Et grâce à un diagramme barre on peut aussi visualiser les nombres de MAC des modes, pour voir si les modes d'une méthode sont indépendants, ou pour comparer les modes entre les méthodes.

La figure 4.3 représente la fenêtre type de *Matlab* comparant une solution à la solution de référence :

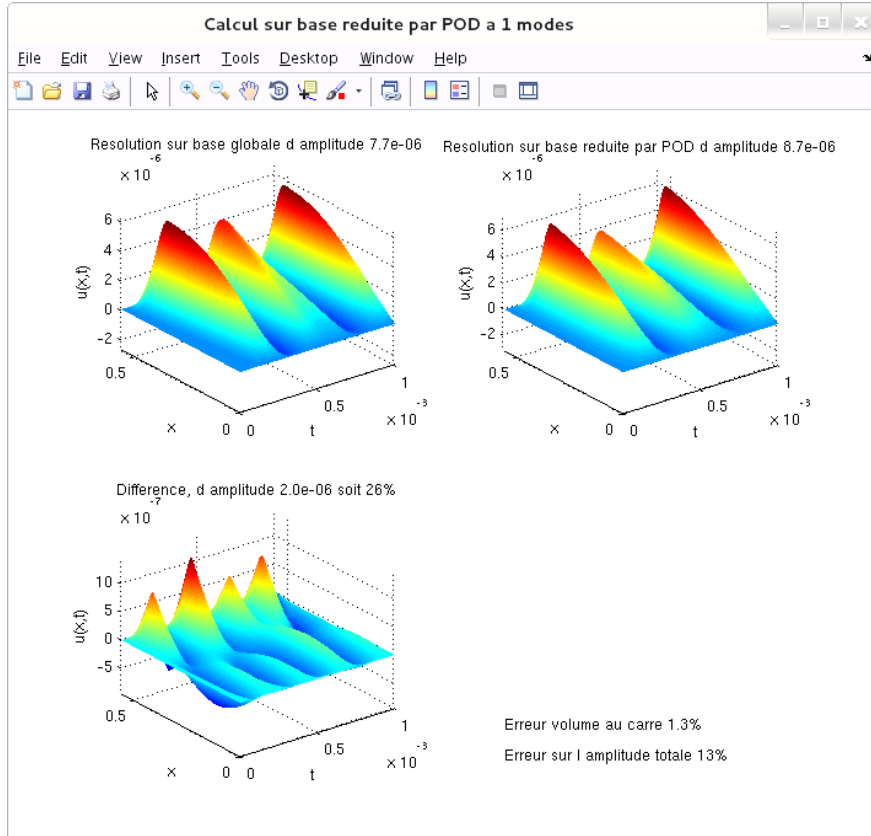


FIGURE 4.3 – Fenêtre type de *Matlab*

## 4.6 Méthode de Rayleigh-Ritz

Cette méthode permettant d'avoir des modes espace, l'analyse par nombre de MAC a été mise en place pour pouvoir les comparer aux précédents. La méthode itérative utilisée est détaillée dans la partie 2.2, et il a pu être constaté que la forme de la matrice de masse choisie pouvait influer sur la convergence de cette méthode.

## 4.7 PGD

La difficulté particulière dans la programmation de la PGD est que le code n'est testable que globalement.

Le code en place réalise une PGD à deux variables uniquement : espace et temps. Les modes en espace sont normalisés pour les raisons évoquées en 5.4. Le besoin qui s'impose pour la résolution du problème en espace comme il est défini en 3.2.2, est d'intégrer des produits de deux fonctions comme on peut le voir dans le déroulement des calculs en annexes D.

Quand la PGD est réalisée avec uniquement deux variables, il est possible de mettre en place une orthogonalisation (cela peut rester possible avec plus de variables mais devient plus complexe à mettre en place). Ceci permet d'avoir des modes indépendants et d'améliorer les modes calculés précédemment comme le montre l'annexe F.2.2. Comme au-delà de deux variables, l'orthogonalisation pose des problèmes, et même si le code actuel n'en possède que deux, dans la perspective d'ajouter d'autres variables on ne va pas s'intéresser plus à l'orthogonalisation.

# Chapitre 5

## Résultats

### 5.1 Condensation

Comme décrite dans la partie 2.1, la condensation repose sur le fait que l'inertie d'une partie des nœuds est négligeable. Elle semble donc inadaptée à un calcul sur un système où la masse est répartie de manière homogène, comme ici. Et quand bien même il serait possible de répartir la masse des nœuds esclaves sur les nœuds maîtres (cela paraît envisageable sur le problème poutre en utilisant une méthode non systématique), dans le meilleur des cas on retrouverait le système à résoudre comme si on avait discrétisé l'espace moins finement.

### 5.2 Méthode Rayleigh-Ritz

La méthode de Rayleigh-Ritz peut être plus rapide que la POD pour calculer des modes espace (La POD fournit en plus des modes en temps), selon la taille du snapshot\* choisi pour la POD (\*explication la partie suivante sur la POD). En comparant les modes POD et Rayleigh-Ritz par une analyse de MAC, on peut voir qu'il s'agit des mêmes si le snapshot choisi est suffisamment long.

En revanche on atteint les limites de la méthode si l'on choisit d'introduire une non-linéarité ou un déplacement imposé qui varie au cours du temps. Ces faiblesses sont dues au fait que la méthode de Rayleigh-Ritz ne se base que sur les données initiales du système, la non-linéarité n'est donc pas visible ou bien il faut recalculer les modes comme expliqué dans la partie 6.2.2. Le fait d'imposer un déplacement qui varie au cours du temps ne se voit pas dans la méthode Rayleigh-Ritz, et les nœuds en question sont considérés comme encastres, alors les modes trouvés ne décrivent pas la réponse du système de manière satisfaisante.

### 5.3 POD

Comme cela est expliqué dans la partie 2.3, la POD repose sur une résolution préalable d'un problème sur base complète. On peut utiliser la réduction de modèle qu'elle apporte pour effectuer un grand nombre de calculs semblables ou pour calculer la réponse du système sur temps plus long. On appelle l'échantillon du temps

sur lequel est faite la résolution préalable : le snapshot. Il représente un temps de calcul et donc un coût ce qui évidemment pousse à le choisir court. Mais la durée du snapshot conditionne aussi la qualité des modes trouvés par la SVD, et cela limite la volonté de le raccourcir. Dans le programme, la longueur du snapshot est choisi en fonction de la période du premier mode pour pouvoir observer quelques oscillations.

## 5.4 PGD

La PGD fournis des modes en espace et en temps sans résolution préalable, fournis une solution, permet d'ajouter des paramètres, mais elle a quelques faiblesses. Notamment vis-à-vis de la convergence.

Premièrement à l'intérieur du point fixe : On peut observer que le point fixe ne converge pas toujours rapidement, et si on arrête le calcul du point fixe après un nombre fixe d'itérations ( $k_{max}$  dans l'algorithme 1 de la partie 2.4) on peut trouver un mode défectueux qui va détériorer la solution globale voire empêcher sa convergence. Heureusement il est possible de remédier à ce problème en normalisant les fonctions en espace. On garde donc la forme du mode en espace mais pas son intensité, qui est répercuter sur la fonction en temps. Avec la normalisation il ne se manifeste plus de divergence cependant, la convergence du point fixe n'est pourtant pas assurée.

Dans la figure 5.1, qui se base sur une résolution PGD d'un problème où l'on cherche la solution sous la forme d'une somme de produit de fonctions du temps et de fonctions de l'espace. La courbe est une indication de la convergence obtenue par la formule suivante calculer à chaque itération :

$$\left[ \frac{\int_T (g_k(t) - g_{k-1}(t))^2 dt \quad ((\varphi_q)_k - (\varphi_q)_{k-1})^T \mathbf{K} ((\varphi_q)_k - (\varphi_q)_{k-1})}{\int_T g_k(t)^2 dt \quad (\varphi_q)_k^T \mathbf{K} (\varphi_q)_k} \right]^{\frac{1}{2}} \quad (5.1)$$

ce qui correspond à :

$$\frac{\|g_k(t) - g_{k-1}\|_T \quad \|(\varphi_q)_k - (\varphi_q)_{k-1}\|_\Omega}{\|g_k(t)\|_T \quad \|(\varphi_q)_k\|_\Omega} \quad (5.2)$$

Sur la figure 5.1 on voit que même après 8 itérations du point fixe le mode peut subir de grande variations. Le mode a été choisi pour illustrer le problème mais la plus part des modes stagnant (convergent) après quelques itérations. Dans le cas où un point fixe ne converge pas, on compte sur les modes suivants pour corriger l'apport défectueux. Ceci semble fonctionner à condition de normer les fonctions d'espace.

Sur la figure 5.2 sont affichés deux indicateurs d'erreur de la solution PGD par rapport à la solution de référence éléments finis. La courbe en bleu représente l'erreur sur l'amplitude totale du signal. La courbe rouge en pointillés représente le maximum de l'erreur. Les formules sont données dans la partie 4.5.1, respectivement l'équation 4.1 et l'équation 4.2.

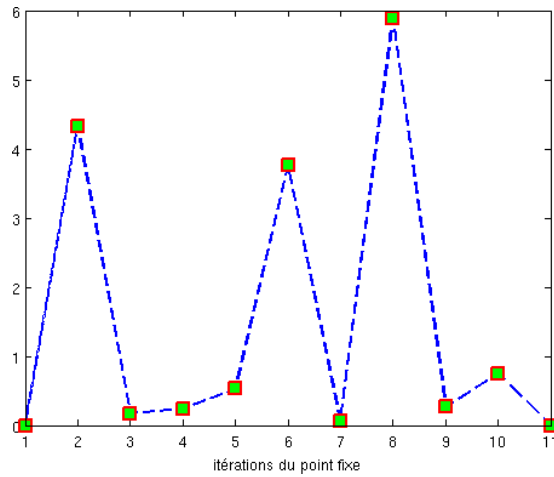


FIGURE 5.1 – Convergence du point fixe au 14<sup>e</sup> mode

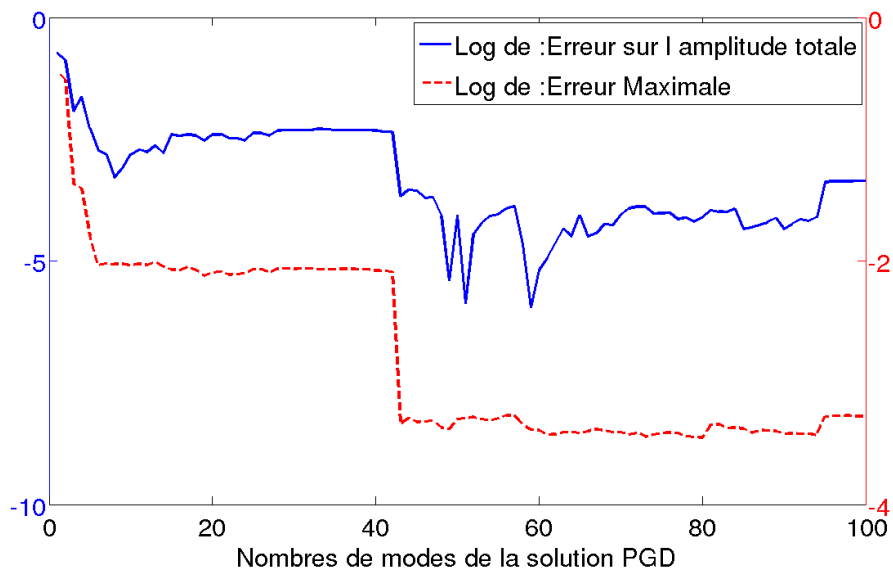


FIGURE 5.2 – Évolution des indicateurs d'erreur d'une solution PGD

La courbe bleu étant sujet à de nombreuses fluctuations, on peut observer plus nettement la diminution de l'erreur sur la courbe rouge. On peut aussi remarquer que tous les modes n'ont pas une contribution de même importance à la diminution de l'erreur. En effet en l'absence d'orthogonalisation des modes, les nouveaux modes trouvés par l'algorithme peuvent être plus moins colinéaires aux modes déjà présent dans la base PGD.

La figure 5.3 représente les mêmes quantités que la précédente mais évaluées pour des solutions calculées par réduction de modèle par POD en faisant varier le nombre de modes. Évidemment la POD fournis des résultats d'erreur moindre pour un même nombre de modes, mais il y a une différence majeure entre la réalisation de ces deux figures : Dans le cas de la PGD, pour avoir un mode supplémentaire on fait juste converger le point fixe une fois de plus. Alors que pour la POD l'ajout d'un mode agrandis la base de projection et pour avoir la nouvelle solution il faut résoudre entièrement le problème sur cette nouvelle base.

Sur la figure 5.4 est représentée la comparaison des modes PGD aux modes POD par analyse de MAC. On peut voir qu'après une centaine de modes trouvés par la PGD, on représente en réalité moins d'une quinzaine de modes indépendants

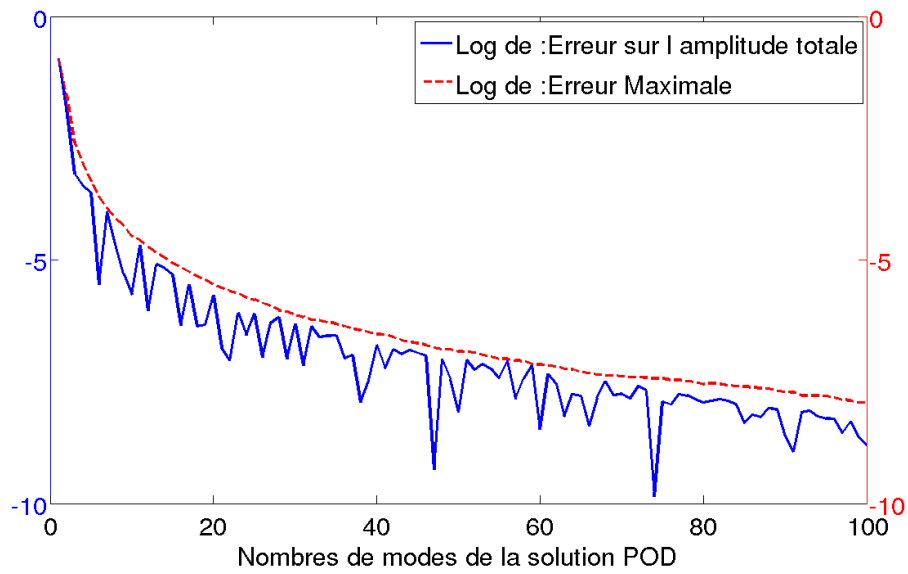


FIGURE 5.3 – Évolution des indicateurs d’erreur d’une solution POD

(orthogonaux), qui correspondent aux premiers modes de la POD. Pour remédier à ce problème il faudrait orthogonaliser les modes. Mais comme expliqué précédemment, l’orthogonalisation ne sera pas mise en place.

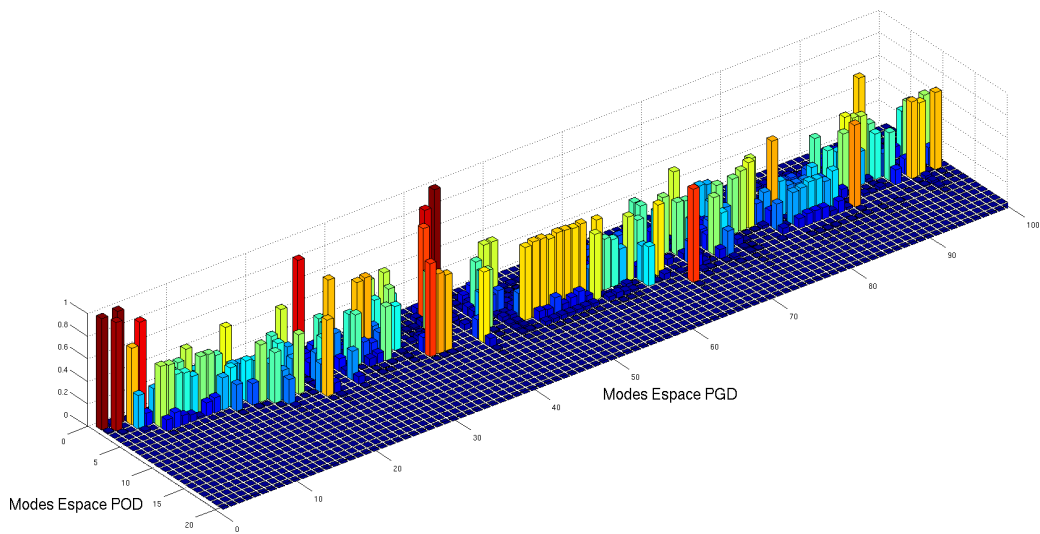


FIGURE 5.4 – Comparaisons des modes PGD et POD

# Chapitre 6

## La PDG pour les problèmes de dynamique non-linéaire

Les problèmes non-linéaires posent des problèmes de résolution car l'absence de linéarité proscriit l'utilisation de nombreux théorèmes comme celui de Lax-Milgram décrit en partie 2.4, puisqu'elle contredit ses hypothèses. Les algorithmes de résolution appliqués aux problèmes linéaires perdent donc ici des propriétés de convergence et c'est pour étendre les possibilités de résolution des problèmes non-linéaires qu'a été développée la méthode LATIN.

### 6.1 La Méthode LATIN

Introduite par [1] et [2], la méthode LATIN (pour LArge Time INcrement) a permis d'établir la convergence de résolutions pour des problèmes mettant en jeu des matériaux non-linéaires à la condition qu'ils respectent l'inégalité de Drucker :

$$\oint_C \sigma d\varepsilon \geq 0, \forall C \quad (6.1)$$

Cette inégalité se traduit par la positivité du travail plastique, puisque le travail des contraintes est nul sur un trajet élastique (pour un trajet fermé). Ceci comprend de nombreux matériaux plastiques ou viscoplastiques. Cette méthode atteint ses limites en revanche pour des matériaux instables, et notamment les matériaux dommageables, pour lesquels la convergence n'est pas assurée. Son nom lui vient du fait qu'elle permet de résoudre des problèmes en temps sur l'ensemble du domaine temporel. Cette capacité a été apportée par la formulation qui est à l'origine de la PGD, utilisant la séparation des variables. La méthode LATIN est itérative puisqu'elle se base sur la résolution alternative de deux problèmes dont l'un et local est potentiellement non-linéaire, et l'autre est linéaire et potentiellement global. Pour plus de détails voir [2] et [14]. La PGD est utilisée pour résoudre le problème linéaire, ce qui est particulièrement pertinent quand la solution de ce problème ne change que légèrement d'une itération à l'autre. L'objectif du travail qui suivra cette bibliographie est d'étudier la possibilité de l'utiliser même pour des problèmes non-linéaires.



## 6.2 Les Résolutions non-linéaires

### 6.2.1 Condensation

Une méthode de condensation traitant des problèmes non-linéaire est décrite par [6], mais se limite à une utilisation dans le cas d'éléments linéaires connectés de manière non- linéaire. La méthode développée dans cette article (SEREP) permet de conserver les valeurs propres et vecteurs propres dans le modèle réduit.

### 6.2.2 Rayleigh-Ritz

Le code Aster propose un extension de la méthode de Rayleigh-Ritz à des cas non-linéaires. Le problème est alors formulé ainsi :

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}(\mathbf{u}) = \mathbf{f}$$

$\mathbf{K}(\mathbf{u})$  est une fonction non linéaire qui dépend des déplacements, tel une raideur non linéaire ou un contact. Alors on définit une matrice de rigidité tangente :

$$\mathbf{K}^{tg}_u = \partial H(u) \partial u$$

On définit alors une base modale à l'aide de modes solution de l'équation :

$$(\mathbf{K}^{tg}\mathbf{u} - \omega_i^{tg}\mathbf{M}) \boldsymbol{\varphi}_i^{tg} = 0$$

Ces modes dépendent de  $\mathbf{K}^{tg}$  et donc de  $\mathbf{u}$ .

### 6.2.3 POD

La SVD peut permettre de décomposer un opérateur, mais si le problème se présente comme dans ce qui précède, il devra subir la décomposition pour chaque  $u$  donc à chaque pas de temps.

## 6.3 Ouverture

La PDG non linéaire est un domaine de recherche qui est tout à fait d'actualité, puisque l'application de la PGD à des problèmes non-linéaires fait l'objet de travaux en cours ou à publier comme [15] et [16], qui présente des problèmes liés à la thermique non-linéaire. Ceci sera l'objet de l'étude du travail qui suivra.

# Bibliographie

- [1] P. Ladevèze and G. Duvaut. Sur une famille d’algorithmes en mécanique des structures. *Compte rendu de l’académie des Sciences*, Vol 300 :pp 41–44, 1985. No. 2.
- [2] P. Ladevèze. La méthode à grand incrément de temps pour l’analyse de structures à comportement non linéaire décrit par variables internes. *Compte rendu de l’académie des Sciences*, Vol 309 :pp 1095–1099, 1989. No. 2.
- [3] F. Chinesta, P. Ladevèze, A. Ammar, E. Cueto, and A. Nouy. Proper generalized decomposition in extreme simulations : Towards a change of paradigm in computational mechanics. *IACM expressions*, No 26 :pp 2–7, 2009.
- [4] Ch. Ghnatios, F. Masson, A. Huerta, A. Leygue, E. Cueto, and F. Chinesta. Proper generalized decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering*, Vol 213-216 :pp 29–41, 2012.
- [5] F. Chinesta, P. Ladevèze, and E. Cueto. A short review on model order based on proper generalized decomposition. *Computational Methods in Engineering*, Vol 18 :pp 395–404, 2011. Issue 4.
- [6] L. Thibault, P. Avitabile, J. Foley, and J. Wolfson. Equivalent reduced model technique development for nonlinear system dynamic response. *Mechanical Systems and Signal Processing*, Vol 36 :pp 422–455, Avril 2013. Issue 2.
- [7] A. Alarcon. Réduction de modèle en dynamique linéaire et non linéaire. *Documentation du code Aster*, Clé : R5.06.01.
- [8] P. Holmes, A. Gray, and C.L. Isbell Jr. Quic-svd : Fast svd using cosine trees. *NIPS*, pages pp 673–680, 2008.
- [9] P. Holmes, J.L. Lumley, and G. Berkooz. Turbulence, coherent structures, dynamical systems and symmetry. *Cambridge Monographs on Mechanics*, 1996.
- [10] W. Rudin. Principles of mathematical analysis. *International Series in Pure and Applied Mathematics*, 1976. 3<sup>e</sup> edition.
- [11] P. Ladevèze L. Chamoin. Toward guaranteed pgd-reduced models. *Bytes and Science*, 2012. CIMNE Barcelona.
- [12] H. Hilber, T. Hughes, and R. Taylor. Improved numerical dissipation for the time integration algorithms in structural dynamics. *Earthquake engineering and structural dynamics*, Vol 5 :pp 283–292, 1977.
- [13] C. Chen and J. Ricles. Analysis of implicit hht-alpha integration algorithm for real-time hybrid simulation. *Earthquake engineering and structural dynamics*, 41 :pp 1021–1041, 2012.
- [14] P. Ladevèze, J.-C. Passieux, and D. Néron. The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, Vol 199 :pp 1287–1296, 2010. Issues 21–22.

- [15] P. Joyot, N. Bur, F. Chinesta, and P. Villon. La méthode pgd appliquée à l'équation de la chaleur non-linéaire : vers une formulation performante. *CSMA*, 11ieme Colloque National en Calcul des Structures, Mai 2013.
- [16] P. Despret, J.-L. Dulong, F. Druesne, and P. Villon. Méthode pgd pour non linéarité matériaux appliquée à la fonderie. *CSMA*, 11ieme Colloque National en Calcul des Structures, Mai 2013.

# Annexe A

## Remerciements

**Mes encadrants** Je voudrais remercier Francois LOUF et Pierre-Alain BOUCARD, pour m'avoir orienté, aidé et avoir répondu à mes nombreuses questions autant par e-mail que de vive voix.

**Mes collègues** Je remercie mes collègues de master pour l'aide donné comme reçue, que se soit pour des réflexions théoriques ou dans la recherche d'une syntaxe *Matlab* ou  $\text{\LaTeX}$ . Et je les remercie aussi pour avoir maintenue une ambiance agréable dans une salle souterraine.

**Le LMT** Ce stage c'est bien passé aussi grâce au laboratoire, qui non-seulement a mis à ma disposition un poste mais me permet de détendre au bar et lors d'éventuel évènements. J'espère que les choses continueront ainsi en thèse.

**Wilson** La mascotte de la salle master 2.

# Annexe B

## Compression d'image par SVD

```
1 clear all;
2 clc;
3 clf;
4
5 % Image originale
6 S = load('gatlin.mat'); %image existante dans Matlab
7 map = rgb2gray(S.map);
8 memory = ceil(numel(S.X)/1024);
9
10 b=4; % nombre de SVD tronquees, precise suffisamment tot
11 % pour utiliser 'subplot' a la place des 'figure'
12 CalculDErreur=1;
13
14 %% Calcul de l'erreur
15 if(CalculDErreur)
16     b = min(size(S.X));
17     Verreur = zeros(min(size(S.X)),1);
18 end
19 %% Calcul de la SVD sur l'image
20
21 % subplot(2,b+1,1);
22 image(S.X);
23 axis('image')
24 axis off
25 colormap(map);
26 title(['Image originale : ' num2str(memory) 'ko']);
27
28 % Image apres SVD
29 A = double(S.X);
30 [U0,S0,V0] = svd(A);
31 ASVD = U0*S0*V0';
32 memory = ceil((size(U0,1) + size(V0,1))*min(size(S0))/1024);
33
34 % subplot(2,b+1,b+2);
35 figure
36 image(ASVD);
37 axis('image')
```

```

38     axis off
39     colormap(map);
40     title(['Image apres SVD : ' num2str(memory) 'ko']);
41
42 VS = zeros(min(size(S0),1)); %Valeurs singulieres
43 for j = 1:min(size(S0))
44     VS(j) = S0(j,j);
45 end
46
47
48 %% Calcul des SVD tronquees
49
50 for i = 1:b
51     % svd tronque au k-ieme ordre
52     if (CalculDErreur)
53         k=i;
54     else
55         k = 4^i;
56     end
57     % Tronquer la SVD premiere manniere
58     S=S0(1:k,1:k);
59     V=V0(:,1:k);
60     U=U0(:,1:k);
61     Ak = U*S*V';
62     % Tronquer la SVD deuxieme maniere
63     %     Vk = V0(:,1:k);
64     %     M = A*Vk;
65     %     [Tild,S,V] = svd(M*M);
66     %     S = S^(1/2);
67     %     U = M*(V/S);
68     %     V = Vk*V;
69     %     Ak = U*S*V';
70
71
72     memory = ceil((size(U,1) + size(V,1))*k/1024);
73
74     if(CalculDErreur)
75         Verreur(i) = norm(ASVD-Ak,'fro')/norm(A,'fro');
76         disp([' ' num2str(i*100/b) '%']);
77     else
78     %         subplot(2,b+1,i+1);
79         figure
80         image(Ak);
81         axis('image')
82         axis off
83         colormap(map);
84         title(['Image apres une SDS d ordre ' num2str(k) ' : ' ←
85             num2str(memory) 'ko.']);
86
87 % Differences
88 %         subplot(2,b+1,b+i+2);
89         figure
90         image(ASVD-Ak);

```

```

90         axis('image')
91         axis off
92         colormap(map);
93         title(['|A-A_k| Erreur : ' num2str(norm(ASVD-Ak,'fro')/←
          norm(A,'fro'))]);
94     end
95 end
96
97 %% Post traitement
98
99 figure % Affichage des valeurs singulieres avec ordonnees en ←
    semilog
100 semilogy(VS(1:end,:))
101 axis([0 500 0 17000]) % pour cadrer un peu mieux dans le cas de ←
    gatlin.mat
102 title('Valeur singulieres');
103
104 if(CalculDErreur)
105     figure % Affichage des valeurs singulieres avec ordonnees en ←
        semilog
106     semilogy(Verreur(1:(size(Verreur)-1)))
107     title('Erreur');
108 end

```

Code source B.1 – Traitement d'image par SVD

# Annexe C

## Calcul du problème PGD étape par étape

- Admissibilité cinématique :  
 $U \in [H^1(\Omega)]^3, U|_{\partial\Omega_U} = U_d$
- Admissibilité statique :  
 $\text{div}\sigma + f_v = \rho\ddot{U} + \mu\dot{U}$  sur  $\Omega$  et  $\sigma.n|_{\partial\Omega_f} = f_d$
- Relation de comportement :  
 $\sigma = H : \epsilon(U)$  sur  $\Omega$
- Conditions initiales :  
 $U(0) = U_0$  et  $\dot{U}(0) = \dot{U}_0$

Formulation variationnelle :

$$\forall U^* \text{ CA0, } \int_{\Omega} [\epsilon(U^*) : \sigma + U^* \mu \dot{U} + U^* \rho \ddot{U} - U^* f_d] d\Omega - \int_{\partial\Omega_f} U^* f_d ds - \underbrace{\int_{\partial\Omega_U} U^* \sigma.n ds}_{=0} = 0 \quad (\text{C.1})$$

Où

- $U$  est le déplacement
- $\Omega$  l'espace, avec le bord  $\partial\Omega = \partial\Omega_U \cup \partial\Omega_f$
- $\sigma$  le tenseur des contraintes
- $\epsilon$  le tenseur des déformations
- $\rho$  la masse volumique
- $\mu$  expression de la viscosité (cette donnée est généralement mal connue)
- $H$  le tenseur de Hook
- $f_v$  effort volumique
- $f_d$  effort imposé que le bord de  $\Omega$

Mise en place de la PGD, utilisation de l'hypothèse des variables séparables :

$$U(x, t, \theta) = \sum_{k=1}^n \phi_k(x) g_k(t) h_k(\theta) \quad (\text{C.2})$$

Pour chaque  $k$ , il faut discrétiser les fonctions sur leur domaine de définition :

$$\begin{aligned} \phi(x) &= \sum_{i=1}^{Nbc_x} N_i(x) \phi_i &= N_{\phi}(x) \phi_{\mathbf{q}} \\ g(t) &= \sum_{i=1}^{Nbc_t} N_i(t) g_i &= N_g(t) \mathbf{g}_{\mathbf{q}} \\ h(\theta) &= \sum_{i=1}^{Nbc_{\theta}} N_i(\theta) h_i &= N_h(\theta) \mathbf{h}_{\mathbf{q}} \end{aligned} \quad (\text{C.3})$$

$Nbc_x$  Représente le nombre de composante de discrétisation spatiale, l'indice  $q$  représente le vecteur sur le domaine discrétisé.



On remplace dans la formulation variationnelle, en choisissant  $U^* = (\phi gh)^* = (\phi^* gh + \phi g^* h + \phi gh^*)$  :

$$\begin{aligned}
& \int_{\Omega} \int_T \int_{\Theta} [\epsilon(\phi^* gh) : H : \epsilon(U_n) + \phi^* gh \mu \dot{U}_n + \phi^* gh \rho \ddot{U}_n - \phi^* gh f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} [\epsilon(\phi g^* h) : H : \epsilon(U_n) + \phi g^* h \mu \dot{U}_n + \phi g^* h \rho \ddot{U}_n - \phi g^* h f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} [\epsilon(\phi gh^*) : H : \epsilon(U_n) + \phi gh^* \mu \dot{U}_n + \phi gh^* \rho \ddot{U}_n - \phi gh^* f_v] dx dt d\theta \\
- & \int_{\partial\Omega_f} \int_T \int_{\Theta} (\phi^* gh + \phi g^* h + \phi gh^*) f_a ds dt d\theta \\
= & 0
\end{aligned} \tag{C.4}$$

Puis  $U = U_n = U_{n-1} + \phi gh$  :

$$\begin{aligned}
& \int_{\Omega} \int_T \int_{\Theta} gh [\epsilon(\phi^*) : H : \epsilon(U_{n-1} + \phi gh) + \phi^* \mu (U_{n-1} + \phi gh) + \phi^* \rho (U_{n-1} + \phi gh) - \phi^* f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} g^* h [\epsilon(\phi) : H : \epsilon(U_{n-1} + \phi gh) + \phi \mu (U_{n-1} + \phi gh) + \phi \rho (U_{n-1} + \phi gh) - \phi f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh^* [\epsilon(\phi) : H : \epsilon(U_{n-1} + \phi gh) + \phi \mu (U_{n-1} + \phi gh) + \phi \rho (U_{n-1} + \phi gh) - \phi f_v] dx dt d\theta \\
- & \int_{\partial\Omega_f} \int_T \int_{\Theta} (\phi^* gh + \phi g^* h + \phi gh^*) f_a ds dt d\theta \\
= & 0
\end{aligned} \tag{C.5}$$

On distribue la dérivation :

$$\begin{aligned}
& \int_{\Omega} \int_T \int_{\Theta} gh [\epsilon(\phi^*) : H : \epsilon(U_{n-1} + \phi gh) + \phi^* \mu (U_{n-1} + \phi gh) + \phi^* \rho (U_{n-1} + \phi gh) - \phi^* f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} g^* h [\epsilon(\phi) : H : \epsilon(U_{n-1} + \phi gh) + \phi \mu (U_{n-1} + \phi gh) + \phi \rho (U_{n-1} + \phi gh) - \phi f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh^* [\epsilon(\phi) : H : \epsilon(U_{n-1} + \phi gh) + \phi \mu (U_{n-1} + \phi gh) + \phi \rho (U_{n-1} + \phi gh) - \phi f_v] dx dt d\theta \\
- & \int_{\partial\Omega_f} \int_T \int_{\Theta} (\phi^* gh + \phi g^* h + \phi gh^*) f_a ds dt d\theta \\
= & 0
\end{aligned} \tag{C.6}$$

On sépare les termes en  $n - 1$  des termes virtuels :

$$\begin{aligned}
& \int_{\Omega} \int_T \int_{\Theta} gh [\epsilon(\phi^*) : H : \epsilon(\phi gh) + \phi^* \mu \phi gh + \phi^* \rho \phi gh - \phi^* f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh [\epsilon(\phi^*) : H : \epsilon(U_{n-1}) + \phi^* \mu U_{n-1} + \phi^* \rho U_{n-1}] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} g^* h [\epsilon(\phi) : H : \epsilon(\phi gh) + \phi \mu \phi gh + \phi \rho \phi gh - \phi f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} g^* h [\epsilon(\phi) : H : \epsilon(U_{n-1}) + \phi \mu U_{n-1} + \phi \rho U_{n-1}] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh^* [\epsilon(\phi) : H : \epsilon(\phi gh) + \phi \mu + \phi gh + \phi \rho \phi gh - \phi f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh^* [\epsilon(\phi) : H : \epsilon(U_{n-1}) + \phi \mu U_{n-1} + \phi \rho U_{n-1} - \phi f_v] dx dt d\theta \\
- & \int_{\partial\Omega_f} \int_T \int_{\Theta} (\phi^* gh + \phi g^* h + \phi gh^*) f_a ds dt d\theta \\
= & 0
\end{aligned} \tag{C.7}$$

On peut regrouper les termes qui n'ont pas de partie virtuel en espace i.e.  $\phi^*$  :

$$\begin{aligned}
& \int_{\Omega} \int_T \int_{\Theta} gh [\epsilon(\phi^*) : H : \epsilon(\phi gh) + \phi^* \mu \phi gh + \phi^* \rho \phi gh - \phi^* f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh [\epsilon(\phi^*) : H : \epsilon(U_{n-1}) + \phi^* \mu U_{n-1} + \phi^* \rho U_{n-1}] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} (g^* h + gh^*) [\epsilon(\phi) : H : \epsilon(\phi gh) + \phi \mu \phi gh + \phi \rho \phi gh - \phi f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} (g^* h + gh^*) [\epsilon(\phi) : H : \epsilon(U_{n-1}) + \phi \mu U_{n-1} + \phi \rho U_{n-1}] dx dt d\theta \\
- & \int_{\partial\Omega_f} \int_T \int_{\Theta} (\phi^* gh + \phi g^* h + \phi gh^*) f_a ds dt d\theta \\
= & 0
\end{aligned} \tag{C.8}$$

On discrétise en espace :

$$\begin{aligned}
& \int_{\Omega} \int_T \int_{\Theta} gh \phi_{\mathbf{q}}^* [\epsilon(N_{\phi}(x)) : H : \epsilon(N_{\phi}(x) \phi_{\mathbf{q}} gh) + N_{\phi}(x) \mu N_{\phi}(x) \phi_{\mathbf{q}} \dot{g}h \\
& \quad + N_{\phi}(x) \rho N_{\phi}(x) \phi_{\mathbf{q}} \ddot{g}h - N_{\phi}(x) f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh \phi_{\mathbf{q}}^* [\epsilon(N_{\phi}(x)) : H : \epsilon(U_{n-1}) + N_{\phi}(x) \mu \dot{U}_{n-1} + N_{\phi}(x) \rho \ddot{U}_{n-1}] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} (g^*h + gh^*) \phi_{\mathbf{q}} [\epsilon(N_{\phi}(x)) : H : \epsilon(N_{\phi}(x) \phi_{\mathbf{q}} gh) + N_{\phi}(x) \mu N_{\phi}(x) \phi_{\mathbf{q}} \dot{g}h \\
& \quad + N_{\phi}(x) \rho N_{\phi}(x) \phi_{\mathbf{q}} \ddot{g}h - N_{\phi}(x) f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} (g^*h + gh^*) \phi_{\mathbf{q}} [\epsilon(N_{\phi}(x)) : H : \epsilon(U_{n-1}) + N_{\phi}(x) \mu \dot{U}_{n-1} + N_{\phi}(x) \rho \ddot{U}_{n-1}] dx dt d\theta \\
- & \int_{\partial\Omega_{\mathbf{f}}} \int_T \int_{\Theta} N_{\phi}(x) (\phi_{\mathbf{q}}^* gh + \phi_{\mathbf{q}} g^*h + \phi_{\mathbf{q}} gh^*) f_d ds dt d\theta \\
= & 0
\end{aligned} \tag{C.9}$$

$$U_{n-1}(x, t, \theta) = \sum_{k=1}^{n-1} \phi_k(x) g_k(t) h_k(\theta) = \sum_{k=1}^{n-1} N_{\phi}(x) (\phi_{\mathbf{q}})_k g_k(t) h_k(\theta) \tag{C.10}$$

En utilisant la formule précédente :

$$\begin{aligned}
& \int_{\Omega} \int_T \int_{\Theta} gh \phi_{\mathbf{q}}^* [\epsilon(N_{\phi}(x)) : H : \epsilon(N_{\phi}(x) \phi_{\mathbf{q}} gh) \\
& \quad + N_{\phi}(x) \mu N_{\phi}(x) \phi_{\mathbf{q}} \dot{g}h \\
& \quad + N_{\phi}(x) \rho N_{\phi}(x) \phi_{\mathbf{q}} \ddot{g}h - N_{\phi}(x) f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} gh \phi_{\mathbf{q}}^* [\epsilon(N_{\phi}(x)) : H : \epsilon(\sum_{k=1}^{n-1} N_{\phi}(x) (\phi_{\mathbf{q}})_k g_k h_k) \\
& \quad + N_{\phi}(x) \mu \sum_{k=1}^{n-1} N_{\phi}(x) (\phi_{\mathbf{q}})_k \dot{g}_k h_k \\
& \quad + N_{\phi}(x) \rho \sum_{k=1}^{n-1} N_{\phi}(x) (\phi_{\mathbf{q}})_k \ddot{g}_k h_k] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} (g^*h + gh^*) \phi_{\mathbf{q}} [\epsilon(N_{\phi}(x)) : H : \epsilon(N_{\phi}(x) \phi_{\mathbf{q}} gh) \\
& \quad + N_{\phi}(x) \mu N_{\phi}(x) \phi_{\mathbf{q}} \dot{g}h \\
& \quad + N_{\phi}(x) \rho N_{\phi}(x) \phi_{\mathbf{q}} \ddot{g}h - N_{\phi}(x) f_v] dx dt d\theta \\
+ & \int_{\Omega} \int_T \int_{\Theta} (g^*h + gh^*) \phi_{\mathbf{q}} [\epsilon(N_{\phi}(x)) : H : \epsilon(\sum_{k=1}^{n-1} N_{\phi}(x) (\phi_{\mathbf{q}})_k \dot{g}_k h_k) \\
& \quad + N_{\phi}(x) \mu \sum_{k=1}^{n-1} N_{\phi}(x) (\phi_{\mathbf{q}})_k \ddot{g}_k h_k \\
& \quad + N_{\phi}(x) \rho \sum_{k=1}^{n-1} N_{\phi}(x) (\phi_{\mathbf{q}})_k \ddot{g}_k h_k] dx dt d\theta \\
- & \int_{\partial\Omega_{\mathbf{f}}} \int_T \int_{\Theta} N_{\phi}(x) (\phi_{\mathbf{q}}^* gh + \phi_{\mathbf{q}} g^*h + \phi_{\mathbf{q}} gh^*) f_d ds dt d\theta \\
= & 0
\end{aligned} \tag{C.11}$$

On peut sortir  $N_{\phi}(x)$  de la somme apportée par  $U_{n-1}$ . Ce qui permet de créer les termes :

$$\mathbf{K} = \int_{\Omega} \epsilon(N_{\phi}(x)) : H : \epsilon(N_{\phi}(x)) dx \tag{C.12}$$

$$\mathbf{C} = \int_{\Omega} N_{\phi}(x) \mu N_{\phi}(x) dx \tag{C.13}$$

$$\mathbf{M} = \int_{\Omega} N_{\phi}(x) \rho N_{\phi}(x) dx \tag{C.14}$$

$$\mathbf{f} = \int_{\Omega} N_{\phi}(x) f_v dx + \int_{\partial\Omega_{\mathbf{f}}} N_{\phi}(x) f_d ds dt d\theta \tag{C.15}$$

On a alors :

$$\begin{aligned}
& \int_T \int_\Theta gh \phi_q^* [ \mathbf{K} \phi_q gh + \mathbf{C} \phi_q \dot{g}h + \mathbf{M} \phi_q \ddot{g}h - N_\phi(x) f_v ] dt d\theta \\
+ & \int_T \int_\Theta gh \phi_q^* [ \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k ] dt d\theta \\
+ & \int_T \int_\Theta (g^*h + gh^*) \phi_q [ \mathbf{K} \phi_q gh + \mathbf{C} \phi_q \dot{g}h + \mathbf{M} \phi_q \ddot{g}h - N_\phi(x) f_v ] dt d\theta \\
+ & \int_T \int_\Theta (g^*h + gh^*) \phi_q [ \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k ] dt d\theta \\
- & \int_{\partial\Omega_f} \int_T \int_\Theta N_\phi(x) (\phi_q^* gh + \phi_q g^*h + \phi_q gh^*) f_a ds dt d\theta \\
= & 0
\end{aligned} \tag{C.16}$$

Puis avec  $\mathbf{f}$  :

$$\begin{aligned}
& \int_T \int_\Theta gh \phi_q^* [ \mathbf{K} \phi_q gh + \mathbf{C} \phi_q \dot{g}h + \mathbf{M} \phi_q \ddot{g}h ] dt d\theta \\
+ & \int_T \int_\Theta gh \phi_q^* [ \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k ] dt d\theta \\
+ & \int_T \int_\Theta (g^*h + gh^*) \phi_q [ \mathbf{K} \phi_q gh + \mathbf{C} \phi_q \dot{g}h + \mathbf{M} \phi_q \ddot{g}h ] dt d\theta \\
+ & \int_T \int_\Theta (g^*h + gh^*) \phi_q [ \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k ] dt d\theta \\
- & \int_T \int_\Theta \mathbf{f} (\phi_q^* gh + \phi_q g^*h + \phi_q gh^*) dt d\theta \\
= & 0
\end{aligned} \tag{C.17}$$

On peut alors réarranger les lignes :

$$\begin{aligned}
& \int_T \int_\Theta gh \phi_q^* [ \mathbf{K} \phi_q gh + \mathbf{C} \phi_q \dot{g}h + \mathbf{M} \phi_q \ddot{g}h ] dt d\theta \\
+ & \int_T \int_\Theta gh \phi_q^* [ \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k ] dt d\theta \\
+ & \int_T \int_\Theta (g^*h + gh^*) \phi_q [ \mathbf{K} \phi_q gh + \mathbf{C} \phi_q \dot{g}h + \mathbf{M} \phi_q \ddot{g}h ] dt d\theta \\
+ & \int_T \int_\Theta (g^*h + gh^*) \phi_q [ \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k ] dt d\theta \\
- & \int_T \int_\Theta \mathbf{f} (\phi_q^* gh + \phi_q g^*h + \phi_q gh^*) dt d\theta \\
= & 0
\end{aligned} \tag{C.18}$$

Et en factorisant :

$$\begin{aligned}
& \int_T \int_\Theta (\phi_q^* gh + \phi_q g^*h + \phi_q gh^*) [ \mathbf{K} \phi_q gh + \mathbf{C} \phi_q \dot{g}h + \mathbf{M} \phi_q \ddot{g}h ] dt d\theta \\
+ & \int_T \int_\Theta (\phi_q^* gh + \phi_q g^*h + \phi_q gh^*) [ \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k ] dt d\theta \\
- & \int_T \int_\Theta (\phi_q^* gh + \phi_q g^*h + \phi_q gh^*) \mathbf{f} dt d\theta \\
= & 0
\end{aligned} \tag{C.19}$$

Puis en regroupant les intégrales :

$$\begin{aligned}
& \int_T \int_{\Theta} (\phi_{\mathbf{q}}^* gh + \phi_{\mathbf{q}} g^* h + \phi_{\mathbf{q}} gh^*) [ \mathbf{K} \phi_{\mathbf{q}} gh + \mathbf{C} \phi_{\mathbf{q}} \dot{g} h + \mathbf{M} \phi_{\mathbf{q}} \ddot{g} h \\
& \quad + \mathbf{K} \sum_{k=1}^{n-1} (\phi_{\mathbf{q}})_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_{\mathbf{q}})_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_{\mathbf{q}})_k \ddot{g}_k h_k \\
& \quad - \mathbf{f} ] dt d\theta \\
& = 0
\end{aligned}
\tag{C.20}$$

# Annexe D

## Intégration de produits

On désire calculer les intégrales pour résoudre les problèmes comme celui en espace :

$$\begin{aligned}
 & \int_T \int_{\Theta} gh [ K gh + C \dot{g}h + M \ddot{g}h ] dt d\theta f_q \\
 = & - \int_T \int_{\Theta} gh [ K \sum_{k=1}^{n-1} (f_q)_k g_k h_k + C \sum_{k=1}^{n-1} (f_q)_k \dot{g}_k h_k + M \sum_{k=1}^{n-1} (f_q)_k \ddot{g}_k h_k - F ] dt d\theta
 \end{aligned} \tag{D.1}$$

On peut obtenir des termes développés ainsi :

$$\begin{aligned}
 & \left[ \int_T \int_{\Theta} g^2 h^2 dt d\theta K + \int_T \int_{\Theta} g \dot{g} h^2 dt d\theta C + \int_T \int_{\Theta} g \ddot{g} h^2 dt d\theta M \right] f_q \\
 = & - \int_T \int_{\Theta} gh \sum_{k=1}^{n-1} (f_q)_k g_k h_k K dt d\theta \\
 & - \int_T \int_{\Theta} gh \sum_{k=1}^{n-1} (f_q)_k \dot{g}_k h_k C dt d\theta \\
 & - \int_T \int_{\Theta} gh \sum_{k=1}^{n-1} (f_q)_k \ddot{g}_k h_k M dt d\theta \\
 & + \int_T \int_{\Theta} gh F dt d\theta
 \end{aligned} \tag{D.2}$$

En inversant somme et intégrale on a :

$$\begin{aligned}
 & \left[ \int_T \int_{\Theta} g^2 h^2 dt d\theta K + \int_T \int_{\Theta} g \dot{g} h^2 dt d\theta C + \int_T \int_{\Theta} g \ddot{g} h^2 dt d\theta M \right] f_q \\
 = & - \sum_{k=1}^{n-1} \int_T \int_{\Theta} gh (f_q)_k g_k h_k K dt d\theta \\
 & - \sum_{k=1}^{n-1} \int_T \int_{\Theta} gh (f_q)_k \dot{g}_k h_k C dt d\theta \\
 & - \sum_{k=1}^{n-1} \int_T \int_{\Theta} gh (f_q)_k \ddot{g}_k h_k M dt d\theta \\
 & + \int_T \int_{\Theta} gh F dt d\theta
 \end{aligned} \tag{D.3}$$

En faisant l'hypothèse que les matrices ne dépendent pas des paramètres intégrés (hypothèse fausse en non-linéaire), la séparation des variables nous donne :

$$\begin{aligned}
& \left[ \int_T g^2 dt \int_{\Theta} h^2 d\theta K + \int_T g \dot{g} dt \int_{\Theta} h^2 d\theta C + \int_T g \ddot{g} dt \int_{\Theta} h^2 d\theta M \right] f_q \\
= & - \sum_{k=1}^{n-1} \int_T g g_k dt \int_{\Theta} h h_k d\theta (f_q)_k K \\
& - \sum_{k=1}^{n-1} \int_T g \dot{g}_k dt \int_{\Theta} h h_k d\theta (f_q)_k C \\
& - \sum_{k=1}^{n-1} \int_T g \ddot{g}_k dt \int_{\Theta} h h_k d\theta (f_q)_k M \\
& + \int_T g F dt \int_{\Theta} h d\theta
\end{aligned} \tag{D.4}$$

On voit que une intégrale simple d'un produit de deux fonctions est un élément calculer de nombreuse fois, il faut donc en créer un programme.

## D.1 Intégration d'un produit de fonctions linéaires par morceaux

Les calculs sont évidemment effectués sur des objet discrétiser pour pouvoir utiliser des outils numériques. Alors on considère deux vecteurs  $G_1$  et  $G_2$  contenant chacun les valeurs d'une fonction discrétisée, respectivement  $g_1(t)$  et  $g_2(t)$ , définie  $\forall t \in [t_0; t_0 + T]$  tels que :

$$\forall n \in [1; (T/\delta t)] G_1(n) = g_1((n-1)\delta t + t_0) \text{ idem pour } G_2 \tag{D.5}$$

Il faut alors choisir une méthode de calcul d'intégrale, la première présentée ici fais l'hypothèse que les fonction sont linéaires entre les valeurs discrètes connues.

$$\int_T g_1(t) g_2(t) dt = \sum_{n=1}^{T/\delta t} \int_{t_0+(n-1)\delta t}^{t_0+n\delta t} g_1(t) g_2(t) dt \tag{D.6}$$

Et si on applique l'hypothèse de linéarité, on peut écrire localement (sur le domaine de l'intégrale précédente)  $g_1$  sous la forme :  $at + b$ , et  $g_2$  sous la forme :  $et + f$ , où  $a, b, e$  et  $f$  sont des réels. Pour alléger la notation on définit  $t_1 = t_0 + (n-1)\delta t$  et  $t_2 = t_0 + n\delta t$ , alors :

$$\begin{aligned}
& \int_{t_1}^{t_2} g_1(t) g_2(t) dt \\
= & \int_{t_1}^{t_2} (at + b)(et + f) dt \\
= & \int_{t_1}^{t_2} (aet^2 + (af + be)t + bf) dt \\
= & \left[ ae \frac{t^3}{3} + (af + be) \frac{t^2}{2} + bft \right]_{t_1}^{t_2} \\
= & ae \frac{t_2^3 - t_1^3}{3} + (af + be) \frac{t_2^2 - t_1^2}{2} + bf(t_2 - t_1)
\end{aligned} \tag{D.7}$$

Donc :

$$\int_T g_1(t)g_2(t)dt = \sum_{n=1}^{T/\delta t} \left[ ae \frac{t_2^3 - t_1^3}{3} + (af + be) \frac{t_2^2 - t_1^2}{2} + bf(t_2 - t_1) \right] \quad (D.8)$$

On peut identifier :

$$\begin{aligned} a &= \frac{g_1(t_1) - g_1(t_2)}{\delta t} &= \frac{G_1(n) - G_1(n+1)}{\delta t} \\ b &= \frac{g_1(t_1) - at_1}{\delta t} &= \frac{G_1(n) - at_1}{\delta t} \\ e &= \frac{g_2(t_1) - g_2(t_2)}{\delta t} &= \frac{G_2(n) - G_2(n+1)}{\delta t} \\ f &= \frac{g_2(t_1) - et_1}{\delta t} &= \frac{G_2(n) - et_1}{\delta t} \end{aligned} \quad \text{et identiquement} \quad (D.9)$$

## D.2 Intégration de produit de fonctions constantes par morceaux

Si on considère les fonctions sont constantes par morceaux, on peut écrire l'intégrale sous la forme :

$$\int_T g_1(t)g_2(t)dt = \sum_{n=1}^{T/\delta t} \int_{t_0+(n-1)\delta t}^{t_0+n\delta t} g_1(t)g_2(t)dt = \sum_{n=1}^{T/\delta t} G_1(n)G_2(n)dt \quad (D.10)$$

Ceci correspond à une intégrale de Riemann, et on peut donc en avoir quelques variantes en changeant les indices de sommation :

$$\begin{aligned} &\sum_{n=1}^{T/\delta t} G_1(n)G_2(n)dt \\ &\sum_{n=2}^{T/\delta t+1} G_1(n)G_2(n)dt \\ &\sum_{n=1}^{T/\delta t+1} G_1(n)G_2(n)dt = G_1 \cdot G_2 dt \end{aligned} \quad (D.11)$$

# Annexe E

## Problème en temps avec utilisation d'éléments finis temporels

### Problème en temps

$$\begin{aligned} \forall g^* \quad \int_T \int_{\Theta} \phi_q^T g^* h [ & \mathbf{K} \phi_q g h + \mathbf{C} \phi_q \dot{g} h + \mathbf{M} \phi_q \ddot{g} h \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{g}_k h_k + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{g}_k h_k \\ & - \mathbf{f} ] dt d\theta \end{aligned} \quad (E.1)$$

$$= 0$$

En discrétisant en temps on obtient :

$$\begin{aligned} \forall \mathbf{g}_q^* \quad \int_T \int_{\Theta} \phi_q^T N_g(t) \mathbf{g}_q^* h [ & \mathbf{K} \phi_q N_g(t) \mathbf{g}_q h + \mathbf{C} \phi_q \dot{N}_g(t) \mathbf{g}_q h + \mathbf{M} \phi_q \ddot{N}_g(t) \mathbf{g}_q h \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k N_g(t) (\mathbf{g}_q)_k h_k \\ & + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{N}_g(t) (\mathbf{g}_q)_k h_k \\ & + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{N}_g(t) (\mathbf{g}_q)_k h_k \\ & - \mathbf{f} ] dt d\theta \end{aligned} \quad (E.2)$$

$$= 0$$

Donc :

$$\begin{aligned} \int_T \int_{\Theta} \phi_q^T N_g(t) h [ & \mathbf{K} \phi_q N_g(t) \mathbf{g}_q h + \mathbf{C} \phi_q \dot{N}_g(t) \mathbf{g}_q h + \mathbf{M} \phi_q \ddot{N}_g(t) \mathbf{g}_q h \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\phi_q)_k N_g(t) (\mathbf{g}_q)_k h_k \\ & + \mathbf{C} \sum_{k=1}^{n-1} (\phi_q)_k \dot{N}_g(t) (\mathbf{g}_q)_k h_k \\ & + \mathbf{M} \sum_{k=1}^{n-1} (\phi_q)_k \ddot{N}_g(t) (\mathbf{g}_q)_k h_k \\ & - \mathbf{f} ] dt d\theta \end{aligned} \quad (E.3)$$

$$= 0$$

Soit :

$$\begin{aligned} & \int_T N_g(t) \left( N_g(t) \int_{\Theta} \phi_q^T h \mathbf{K} \phi_q h d\theta + N_g(t) \int_{\Theta} \phi_q^T h \mathbf{C} \phi_q h d\theta + N_g(t) \int_{\Theta} \phi_q^T h \mathbf{M} \phi_q h d\theta \right) dt \mathbf{g}_q \\ = & - \int_T N_g(t) \int_{\Theta} \phi_q^T h \left[ \left[ \mathbf{K} N_g(t) + \mathbf{C} \dot{N}_g(t) + \mathbf{M} \ddot{N}_g(t) \right] \sum_{k=1}^{n-1} (\phi_q)_k (\mathbf{g}_q)_k h_k - \mathbf{f} \right] dt d\theta \end{aligned} \quad (E.4)$$



Pour alléger l'écriture on définit :

$$\mathbf{G} = \int_T N_g(t) N_g(t) dt, \quad \dot{\mathbf{G}} = \int_T N_g(t) \dot{N}_g(t) dt, \quad \ddot{\mathbf{G}} = \int_T N_g(t) \ddot{N}_g(t) dt, \quad \mathbf{f}_g = \int_T N_g(t) \mathbf{f} dt \quad (\text{E.5})$$

Alors on a :

$$\begin{aligned} & \left( \mathbf{G} \int_{\Theta} \phi_q^T h \mathbf{K} \phi_q h d\theta + \dot{\mathbf{G}} \int_{\Theta} \phi_q^T h \mathbf{C} \phi_q h d\theta + \ddot{\mathbf{G}} \int_{\Theta} \phi_q^T h \mathbf{M} \phi_q h d\theta \right) \mathbf{g}_q \\ = & - \int_{\Theta} \phi_q^T h \left[ \mathbf{K} \mathbf{G} + \mathbf{C} \dot{\mathbf{G}} + \mathbf{M} \ddot{\mathbf{G}} \right] \sum_{k=1}^{n-1} (\phi_q)_k (\mathbf{g}_q)_k h_k \Big] d\theta + \mathbf{f}_g \end{aligned} \quad (\text{E.6})$$

Note : Le produit  $\mathbf{K} \mathbf{G}$  n'est pas possible, ils n'ont pas de raison d'avoir la même dimension. L'expression précédente est une forme factorisée qu'il faut développer pour retrouver les produits entre les termes qui conviennent (termes discrétisés en espace entre eux, termes discrétisés en temps entre eux...)

# Annexe F

## Algorithmes PGD à deux variables

### F.1 Résumé

Séparation des variables :

$$U_n(X, t) = \sum_{k=1}^n f_k(X)g_k(t) \quad (\text{F.1})$$

Formulation variationnelle :

$$\begin{aligned} f_k &= F(U_{k-1}, g_k) \\ g_k &= G(U_{k-1}, f_k) \end{aligned} \quad (\text{F.2})$$

Algorithme

$$\begin{aligned} & \text{for } k = 1 \text{ à } n \\ & \quad \text{initialiser } g_k \\ & \quad \text{for } j = 1 \text{ à } j_{max} \\ & \quad \quad f_k = F(U_{k-1}, g_k) \\ & \quad \quad g_k = G(U_{k-1}, f_k) \\ & \quad \text{end} \\ & \text{end} \end{aligned} \quad (\text{F.3})$$

### F.2 Orthogonalisation

#### F.2.1 Dans le point fixe

$$\begin{aligned} & \text{for } k = 1 \text{ à } n \\ & \quad \text{initialiser } g_k \\ & \quad \text{for } j = 1 \text{ à } j_{max} \\ & \quad \quad f_k = F(U_{k-1}, g_k) \\ & \quad \quad f_k = f_k - \sum_{i=1}^{k-1} (f_k \cdot f_i) f_i \\ & \quad \quad f_k = f_k / \text{norme}(f_k) \\ & \quad \quad g_k = G(U_{k-1}, f_k) \\ & \quad \text{end} \\ & \text{end} \end{aligned} \quad (\text{F.4})$$

## F.2.2 Amélioration de la base

$$\begin{array}{l}
\text{for } k = 1 \text{ à } n \\
\quad \text{initialiser } g_k \\
\quad \text{for } j = 1 \text{ à } j_{max} \\
\quad \quad f_k = F(U_{k-1}, g_k) \\
\quad \quad g_k = G(U_{k-1}, f_k) \\
\quad \text{end} \\
\quad \left\{ \begin{array}{l}
\text{Si on enlève une partie ainsi :} \\
f_k = f_k - \sum_{i=1}^{k-1} (f_k \cdot f_i) f_i \\
\text{On veut améliorer la base par :} \\
U_{k-1} = U_{k-1} + \sum_{i=1}^{k-1} (f_k \cdot f_i) f_i g_k \\
\text{Une manière de le faire est d'impacter les } g_i : \\
U_{k-1} = \sum_{i=1}^{k-1} f_i g_i + \sum_{i=1}^{k-1} (f_k \cdot f_i) f_i g_k \\
U_{k-1} = \sum_{i=1}^{k-1} f_i (g_i + (f_k \cdot f_i) g_k) \\
\text{Donc :}
\end{array} \right. \quad (F.5) \\
\quad \text{for } i = 1 \text{ à } (k-1) \\
\quad \quad g_i = g_i + (f_k \cdot f_i) g_k \\
\quad \text{end} \\
\quad f_k = f_k - \sum_{i=1}^{k-1} (f_k \cdot f_i) f_i \\
\quad \left\{ \begin{array}{l}
\text{Si le mode } f_k \text{ orthogonalisé est très petit :} \\
\quad \text{C'est que le mode trouvé par le point fixe pouvait} \\
\quad \text{être représenté dans la base des modes précédents.} \\
\text{Alors on choisit de l'ignorer pour ne pas polluer la base.} \\
\text{Si le mode } f_k \text{ orthogonalisé satisfait un critère de norme :} \\
\quad \text{On l'ajoute à la base comme précédemment.} \\
\text{Donc :}
\end{array} \right. \\
\quad \text{if } (\|g_k\| \cdot \|f_k\|) < \epsilon \\
\quad \quad k = k - 1 \\
\quad \text{end} \\
\text{end}
\end{array}$$

Sans les commentaires cela peut être plus lisible :

$$\begin{aligned}
& \textit{for } k = 1 \text{ à } n \\
& \quad \textit{initialiser } g_k \\
& \quad \textit{for } j = 1 \text{ à } j_{max} \\
& \quad \quad f_k = F(U_{k-1}, g_k) \\
& \quad \quad g_k = G(U_{k-1}, f_k) \\
& \quad \textit{end} \\
& \quad \textit{for } i = 1 \text{ à } (k - 1) \\
& \quad \quad g_i = g_i - (f_k \cdot f_i) g_k \\
& \quad \textit{end} \\
& \quad f_k = f_k - \sum_{i=1}^{k-1} (f_k \cdot f_i) f_i \\
& \quad \textit{if } (\|g_k\| \cdot \|f_k\|) < \epsilon \\
& \quad \quad k = k - 1 \\
& \quad \textit{end} \\
& \textit{end}
\end{aligned} \tag{F.6}$$