



HAL
open science

Méthode combinée volumes finis et meshless local Petrov Galerkin appliquée au calcul de structures

Mohammad-Réza Moosavi

► **To cite this version:**

Mohammad-Réza Moosavi. Méthode combinée volumes finis et meshless local Petrov Galerkin appliquée au calcul de structures. Autre. Université Henri Poincaré - Nancy 1, 2008. Français. NNT : 2008NAN10080 . tel-01748434

HAL Id: tel-01748434

<https://hal.univ-lorraine.fr/tel-01748434>

Submitted on 29 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

**NANCY UNIVERSITÉ
UNIVERSITÉ HENRI POINCARÉ, NANCY I
ÉCOLE DOCTORALE « EMMA »
GFD Mécanique-Energétique**

2008

THÈSE

Présentée pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITÉ
HENRI POINCARÉ, NANCY I**
Spécialité : Mécanique et Energétique

par

Mohammad-Réza MOOSAVI

**MÉTHODE COMBINÉE VOLUMES FINIS ET MESHLESS LOCAL PETROV
GALERKIN APPLIQUÉE AU CALCUL DE STRUCTURES**

Directeur de thèse : Abdelouahab KHELIL

Soutenue publiquement le 12 novembre 2008 devant la commission d'examen :

Rapporteurs :

Francisco CHINESTA
Nguyen-Dang HUNG

Examineurs :

Tarak BEN ZINEB (Président du Jury)
Julien YVONNET

Remerciements

Cette thèse a été effectuée au Laboratoire d’Energétique et de Mécanique Théorique et Appliquée (LEMETA UMR CNRS 7563) sous la direction de Monsieur KHELIL, Maître de Conférences HDR à l’Université Henri Poincaré, Nancy 1.

Je voudrais exprimer toute ma gratitude à Monsieur KHELIL qui m’a guidé dans ce thème de recherche, pour sa disponibilité permanente, ses conseils éclairés et motivants.

J’adresse mes remerciements à Monsieur le Professeur CHINESTA, de l’école centrale de Nantes (ancien Directeur du laboratoire de Mécanique des Systèmes & des Procédés, UMR CNRS 8106 de l’ENSAM de PARIS), qui a bien voulu être le rapporteur de ce travail malgré ses lourdes responsabilités.

Mes très sincères remerciements vont également à Monsieur le Professeur DANG HUNG, de l’Institut de Mécanique et Génie Civil, de l’université de Liège, pour le grand honneur qu’il m’a fait en acceptant de rapporter mes travaux.

Mes remerciements vont également à Monsieur Julien YVONNET de l’université de Marne La Vallée, d’avoir accepté d’examiner ce travail de recherche dans un domaine qu’il connaît bien.

Je tiens à remercier particulièrement Monsieur le professeur BEN ZINEB, de l’université Henri Poincaré Nancy 1, responsable de l’équipe mécanique du solide à l’ESSTIN, de son intérêt à ce travail de recherche.

Je tiens à associer à ces remerciements : Monsieur le professeur HORNUT, Directeur de l’IUT Nancy Brabois, Monsieur BLIN-LACROIX Vice-Président de l’UHP chargé des moyens immobiliers et Monsieur ROY, Chef du département Génie Civil de l’IUT, qui ont soutenu chacun à leur façon, mon activité de recherche au département génie civil.

Pour finir je ne peux oublier de mentionner ma famille : tout d’abord mon père qui aurait bien voulu assister à cette thèse, mais le destin en a décidé autrement, ma mère et toute ma famille qui, malgré l’éloignement m’a toujours aidé et soutenu dans ce travail.

Je tiens également à remercier le ministère de la science, de la recherche et de la technologie d’Iran en particulier Madame KHODAMI ainsi que Messieurs les Docteurs RAHMATI et ABDOLLAHI du service scientifique de l’Ambassade d’Iran à Paris pour le soutien financier à la réalisation de ce travail de thèse.

Sommaire

Remerciement	2
Abréviations	5
Chapitre 1 Introduction	6
1.1 Motivation et objectif	6
1.2 Méthodes sans maillages	7
1.3 Principe des méthodes sans maillage	10
1.3.1 Modélisation de la géométrie	11
1.3.2 Génération des nœuds	11
1.3.3 Construction des fonctions de forme	11
1.3.4 Conditions aux limites initiales, et conditions de charge	12
1.4 Contenu	12
Chapitre 2 Méthode « FVMLPG » appliquée aux problèmes unidimensionnels de classe C^0	14
2.1 Forme faible pour les problèmes 1-D de classe C^0	14
2.2 Approximation des moindres carrés mobiles	23
2.2.1 Méthode moindres carrés	23
2.2.2 Moindres carrés pondérés	25
2.2.3 Méthode des moindres carrés mobiles	26
2.3 Équations du système	30
2.4 Méthode de collocation (conditions aux limites essentielles - Dirichlet)	33
2.5 Exemples numériques	33
2.6 Remarques et conclusion	39
Chapitre 3 FVMLPG appliquée aux problèmes C^0 élastostatiques bidimensionnels	40
3.1 Forme faible symétrique locale (LSWF) de problèmes élastostatiques unidimensionnels	40
3.2 Approche volumes finis moindres carrés mobiles (FVMLPG)	47
3.3 Imposition des conditions aux limites	51
3.4 Intégration numérique	51
3.5 Routine d'exécution	53
3.6 Exemples numériques	53
3.7 Remarques et conclusion	69

Chapitre 4 FVMLPG appliquée aux poutres minces	70
4.1 Rappel succinct de la théorie des poutres	70
4.2 Forme faible locale pour des problèmes de poutre d'Euler Bernoulli	73
4.3 Exemples numériques	76
4.3.1 Patch tests	77
4.3.2 Problèmes aux valeurs limites mixtes	80
4.3.3 Poutre continue	85
4.4 Remarques et conclusion	87
Chapitre 5 FVMLPG appliquée aux problèmes C^0 élastodynamiques bidimensionnels	88
5.1 Forme faible symétrique locale (LSWF) de problèmes C^0 élastodynamiques bidimensionnelle	88
5.2 Imposition des conditions aux limites	94
5.3 Intégration temporelle	96
5.4 Exemple numérique	97
5.5 Remarques et conclusion	104
Chapitre 6 FVMLPG appliquée aux instabilités de poutres minces	105
6.1 Équations d'équilibres	105
6.2 Remarques et conclusion	111
Conclusions et perspectives	112
Références bibliographiques	115
Annexe Code développé sous MATLAB	125
Résumé	151

Abréviations

ALE	Arbitrary Lagrangian Eulerian
DEM	Diffuse Element Method
EFGM	Element Free Galerkin Method
ELB	Équation de Lattice Boltzmann
EDP	Équations aux dérivées Partielles
FVMLPG	Finite Volume Meshless Local Petrov-Galerkin
LWF	Local Weak Form
MEF	Méthode des Éléments Finis
MLPG	Meshless Local Petrov Galerkin
MN	Mécanique Numérique
MTA	Mécanique Théorique et Appliquée
MVF	Méthode des Volumes Finis
PIC	Particle-in-cell
RKPM	Reproducing Kernel Particle Method
SDMC	Simulation Directe Monte Carlo
SPH	Smoothed Particle Hydrodynamics

Chapitre 1

Introduction

Ce chapitre décrit l'objectif de ce travail ainsi que les développements récents des méthodes sans maillages et de leurs applications dans la mécanique appliquée.

1.1 Motivation et objectif

Le champ de la mécanique numérique (MN) est généralement dominé par les méthodes de calcul de type éléments finis (EF) particulièrement pour la mécanique des solides et par la méthode des volumes finis (VF), pour la mécanique des fluides. La méthode des éléments finis (MEF) est de nos jours la méthode la plus utilisée pour résoudre des systèmes d'équations aux dérivées partielles (EDP) issus de problèmes de modélisation physique. Cette méthode bénéficie d'un fondement théorique très solide, et de nombreuses techniques sont venues l'améliorer au fil des ans. Cependant, sa mise en œuvre reste difficile et parfois coûteuse dans certains cas, notamment dans le domaine de la modélisation de grandes déformations. Des techniques de remaillage adaptatives automatiques très performantes ont été développées, mais restent gourmandes en temps de calcul et posent souvent des problèmes pour les géométries complexes. Ces dernières années de nouvelles méthodes sans maillages sont apparues comme solutions alternatives à la méthode des éléments finis (MEF). Comme leur nom l'indique, ces méthodes ne requièrent pas de maillage pour construire l'approximation du champ inconnu dans le domaine mais seulement un nuage de points. Afin de contribuer à l'amélioration des performances de ces méthodes, on propose dans ce travail de thèse une nouvelle approche intitulée « finite volume meshless local Petrov Galerkin » (FVMLPG). La méthode repose sur la combinaison d'une méthode sans maillage de type Galerkin et de l'approche par volume finis. L'approximation par les moindres carrés mobiles conduit à une distribution lisse du champ de variables. Cette propriété a un rôle efficace dans l'exactitude des résultats. Avec ce concept de volumes finis, la fonction de base du premier ordre dans l'interpolation produit des résultats précis et par conséquent plus d'efficacité que les autres méthodes sans maillage de type MLPG qui nécessitent une fonction de base d'ordre supérieur.

1.2 Méthodes sans maillages

La mécanique théorique et appliquée (MTA) est la branche des sciences appliquées relative à l'étude des phénomènes mécaniques : le comportement des fluides, des solides, et des matériaux complexes sous les actions des forces extérieures. Le développement de cette discipline a eu un grand impact sur le monde industriel, permettant des modifications technologiques très importantes dans pratiquement chaque secteur qui affecte notre sécurité et bien-être.

La mécanique numérique (MN) est la sous discipline de la MTA concernée par l'utilisation des méthodes informatiques dans l'étude des événements régis par les principes de la mécanique. La mécanique numérique a eu un impact profond sur la science et la technologie ces trois dernières décennies. La MN a complètement transformé la théorie newtonienne classique en outils pratiques pour la prévision et la compréhension des systèmes complexes. Ceux-ci ont eu un impact dominant sur la fabrication, la communication, le transport, la médecine, la défense et beaucoup d'autres secteurs d'activité dans le monde moderne.

Depuis son invention dans les années 50, la méthode des éléments finis (MEF) est devenue la méthode la plus populaire et largement la plus répandue en calcul mécanique. Le principe de la MEF repose sur la subdivision d'un corps continu en éléments discrets. Cette subdivision s'appelle la discrétisation. Dans la MEF, les différents éléments sont reliés entre eux par une carte topologique, qui s'appelle habituellement le maillage. Les fonctions d'interpolation d'éléments finis sont alors établies sur le maillage, qui assure la compatibilité de l'interpolation. Cependant, ce procédé n'est pas toujours avantageux, parce que l'état numérique de compatibilité n'est pas souvent identique à celui de l'état physique du corps continu. Par exemple, dans un calcul de type lagrangien, la distorsion d'une maille peut bloquer le processus de calcul. Bien que la MEF basée sur le maillage soit dominante dans l'ingénierie, elle bute toutefois sur quelques difficultés en particulier dans la résolution des problèmes impliquant les grandes déformations, transformations et discontinuité (distorsion du maillage dans l'analyse du comportement des pneus, la propagation de fissures). Dans ces problèmes, afin de maintenir la connectivité d'élément, des algorithmes spécifiques de génération et d'amélioration de maillages sont développés. Parfois, le coût pour le pré-traitement et le post-traitement est encore plus élevé que le coût de résolution des équations linéaires. En outre, la MEF exige souvent un maillage raffiné dans les problèmes avec des gradients élevés ou avec un caractère local distinct, ce qui peut augmenter le temps de calcul. Pour cette raison, la méthode des éléments finis adaptative est devenue une nécessité.

Aujourd'hui, les procédures de remaillages adaptatifs pour des simulations des problèmes d'impact/pénétration, des problèmes d'explosion/fragmentation, des obstacles de passage d'écoulement, et des problèmes d'interaction de fluide structure sont devenues des défis pour les recherches avenir. Les difficultés rencontrées sont non seulement le remaillage mais également le passage des variables de l'ancien maillage au nouveau. Par conséquent, les formulations Arbitrary Lagrangian Eulérien (ALE) ont été développées [1-4]. L'objectif de la formulation ALE est d'introduire un maillage indépendant du matériau de sorte que la déformation du corps n'affecte pas le maillage. Malheureusement, dans des simulations numériques de très grande déformation et/ou de systèmes mécaniques et structuraux à grande vitesse, même avec la formulation de type ALE, une maille tordue présente des erreurs graves dans des calculs numériques. Par conséquent, il serait plus efficace au plan informatique de discrétiser le milieu continu par seulement un ensemble de points nodaux, ou des particules, sans les contraintes de maillage. C'est le *leitmotiv* principal des méthodes sans maillage de Galerkin.

Ces dernières années, les méthodes sans maillages sont très développées pour la résolution de problèmes avec des discontinuités ou avec des frontières mobiles et également pour le cas des déformations importantes du matériau [116-129]. Plusieurs méthodes ont été mises au point ; comme la méthode dite « smoothed particle hydrodynamics » (SPH) [1], la « méthode des éléments diffus » (MED) [2], la « méthode element free Galerkin » (MEFG) [3], la méthode « reproducing kernel particle » (MRKP) [4], la méthode « HP-clouds » [5], la partition d'unité (PUM) [6], et de la méthode « boundary node » (MBN) [7]. La plupart de ces méthodes sont basées sur une forme faible globale. En faisant face aux formulations sans maillage avec une forme faible globale, la question principale est le calcul de quadrature impliqué dans le principe variationnel lié au problème de valeur. Cette quadrature exige des cellules de fond qui rendent ces méthodes pas vraiment sans maillage. En outre, pour atteindre une exactitude correcte, la forme sans maillage exige beaucoup plus de temps de calcul que la méthode EF. Généralement les fonctions de forme d'ordre supérieur sont utilisées dans la méthode sans maillage nécessitant plus de points de quadrature pour une intégration exacte. Par conséquent, en raison du coût de calcul élevé, les méthodes sans maillage mentionnées ci-dessus ne sont pas assez efficaces pour des problèmes compliqués et particulièrement dans les cas non linéaires.

Plusieurs approches sont proposées dans la littérature pour éliminer le maillage de fond en plus de la méthode de collocation. L'une de ces méthodes est la méthode dite meshless local Petrov-Galerkin (MLPG), de type méthode résiduelle pondérée locale [10-16]. Pour la méthode de conventionnel de Galerkin, la fonction d'essai et la fonction test sont choisies dans un même

espace de fonction, alors que pour la méthode de MLPG, qui est réellement une méthode de résiduel pondéré, ces fonctions sont choisies dans des espaces différents.

Avantages des méthodes sans maillage :

- 1) Elles s'appliquent facilement pour les grandes déformations puisque la connectivité entre les noeuds est générée en tant qu'élément de calcul et peut changer en fonction du temps ;
- 2) La méthodologie peut être liée plus facilement avec une base de données de CAD que les éléments finis, puisqu'il n'est pas nécessaire de produire un maillage d'éléments ;
- 3) La méthode peut facilement simuler des propagations de fissures ;
- 4) L'exactitude peut être améliorée plus facilement en rajoutant si nécessaire des noeuds supplémentaires en cours de calcul. (h-adaptative) ;
- 5) Les méthodes sans maillage peuvent être utilisées pour modéliser des structures en grande déformation ainsi que les structures à parois minces de type coque tel que les nano tubes.
- 6) La discrétisation peut fournir une représentation plus précise de la géométrie de l'objet.

En général, les méthodes sans maillage peuvent être classées suivant deux critères différents : principes physiques, ou formulations informatiques. Selon la modélisation physique du milieu, elles peuvent être classées en deux catégories suivant les concepts déterministes ou probabilistes. On distingue également les méthodes basées sur les approximations des formes faibles et fortes des EDP.

Pour approximer la forme forte d'une EDP en utilisant une méthode sans maillage, l'équation aux dérivées partielles est discrétisée par une technique spécifique de collocation comme :

- smoothed particle hydrodynamics (SPH) [7-11],
- la méthode de vortex [12-17],
- la méthode des différences finies généralisées [18, 19],

Il faut noter que, quelques méthodes sans maillage, telles que SPH et les méthodes de vortex, ont été développées dans un premier temps en tant que méthodes probabilistes [9, 13]. Aujourd'hui, elles sont utilisées en tant que méthodes déterministes. Néanmoins, la majorité des méthodes sans maillage dans cette catégorie sont basées sur des principes probabilistes, ou exploitées en tant qu'outils probabilistes de simulation. Il y a trois méthodes principales dans cette catégorie :

- 1) dynamique moléculaire (dynamique moléculaire de quantum [20-25] et dynamique moléculaire classique [26-31]) ;
- 2) la simulation directe Monte Carlo (SDMC), ou la méthode de Monte Carlo sont basées sur la dynamique moléculaire [32-40] (on note que toutes les méthodes de Monte Carlo ne sont pas des méthodes sans maillage, comme la méthode des éléments finis probabiliste qui est une méthode avec maillage [41-43]) ;
- 3) le lattice gas automaton (LGA), [44-48] ainsi que la méthode Lattice Boltzmann Equation (LBE) [49-53] ne sont pas des méthodes sans maillage et nécessitent une grille.

La deuxième classe des méthodes de particules est utilisée avec les diverses formulations faibles de Galerkin, qui s'appelle les méthodes de Galerkin comme :

- Diffuse Element Method (DEM) [54-57],
- Element Free Galerkin Method (EFGM) [58-62],
- Reproducing Kernel Particle Method (RKPM) [63-71],
- h-p Cloud Method [72-75],
- Partition of Unity Method [76-78],
- méthode de Meshless Local Petrov-Galerkin (MLPG) [79-82],
- Free mesh method [83-87],

Il y a des exceptions à cette classification, quelques méthodes de particules peuvent être utilisées aussi bien dans la forme forte de collocation, que dans la discrétisation de la forme faible (exemple, la méthode « particle-in-cell (PIC) »). La forme forte collocation PIC s'appelle souvent la méthode de particule dans une cellule de volumes finis [88-90], et la forme faible PIC s'appelle la méthode du point matériel [91], ou simplement la méthode dite « particle -in-cell [92-94].

1.3 Principe des méthodes sans maillage

La différence fondamentale entre la méthode des éléments finis et les méthodes sans maillage est la construction des fonctions de forme. La génération de la géométrie, l'entrée des données ; matérielle, matrice globale, les étapes de post-traitement (qui n'existent pas dans des méthodes sans maillage) sont les procédures habituelles que les deux méthodes partagent. Ces deux méthodes divergent à l'étape de la création du maillage.

1.3.1 Modélisation de la géométrie

Dans la méthode EF, les parties complexes de la géométrie et des frontières peuvent être modélisées en utilisant les éléments (surfaciques ou curvilignes) d'ordre élevé. Le nombre et l'ordre des éléments utilisés conditionnent l'exactitude de la solution. Un maillage plus fin peut généralement mener à des résultats plus précis. Dans les méthodes sans maillage, la frontière (non discrétisée) est représentée par des noeuds. À un point quelconque entre deux noeuds sur la frontière, on peut interpoler en utilisant une fonction de forme sans maillage. Une frontière incurvée peut être rapprochée très exactement même avec une base de polynômes linéaires. En effet, les fonctions de forme sans maillages sont créées en utilisant des noeuds dans un domaine local mobile.

1.3.2 Génération des noeuds

La génération du maillage est une partie de prétraitement très importante dans la MEF. Le domaine doit être correctement maillé avec des éléments bien précis et sans recouvrement ni trou. Dans les méthodes sans maillage, le domaine du problème est représenté par un ensemble de noeuds arbitrairement distribués. Pour les méthodes sans maillage, le processus de la génération de noeud peut être entièrement automatisé sans modélisations particulières comme pour la MEF. Toutefois, quelques méthodes sans maillage exigent un maillage de fond des cellules pour l'intégration des matrices de système (par exemple la méthode élément free Galerkin). Puisque le maillage de fond est seulement exigé pour l'intégration numérique, n'importe quelle forme de cellules est acceptée à condition qu'elle fournisse une exactitude suffisante dans les intégrations numériques.

1.3.3 Construction des fonctions de forme

Pour construire les fonctions de forme, on utilise la technique des moindres carrés mobiles, dans laquelle l'influence d'un noeud émane d'une fonction de poids définie sur un sous-ensemble compact arbitraire du domaine. Les fonctions de forme peuvent être construites avec une continuité arbitraire. Les frontières des appuis des noeuds n'affectent pas le lissage de la fonction de forme. La douceur de la fonction de forme est donnée par la douceur de la fonction de poids et des prétendues fonctions de base.

La construction de la fonction de forme est effectuée pendant l'analyse. Elles sont complexes et doivent être calculées avec l'utilisation de la connaissance prédéfinie du rapport des noeuds. Le problème principal est que les fonctions de forme construites n'ont pas la propriété du symbole de Kronecker.

Les recouvrements arbitraires des appuis nodaux conduisent à une connectivité variable ; le nombre de noeuds affectant l'approximation change d'un point à un autre.

1.3.4 Conditions aux limites initiales, et conditions de charge

Puisque les fonctions de forme ne tiennent pas compte de la propriété du symbole de Kronecker, des techniques spéciales sont exigées pour imposer les conditions aux limites essentielles.

1.4 Contenu

Le chapitre 2 décrit la méthode de FVMLPG pour les problèmes d'ordre C^0 . Ces problèmes sont décrits par une équation ordinaire du second degré. Les variables dépendantes sont continues, mais leurs dérivés peuvent ne pas l'être. Une forme faible locale de l'équation régissant l'équilibre est développée. Des approximations à la solution connue sous le nom de fonctions d'essai sont établies en utilisant l'interpolation des moindres carrés mobiles. La formulation de Petrov Galerkin pour ces problèmes C^0 est présentée. Un système d'équations algébriques est déduit à partir de l'interpolation de MLS et des fonctions test de Petrov Galerkin sous la forme faible locale. Des exemples numériques avec des patches tests, et des conditions aux limites mixtes sont détaillés.

Le chapitre 3 décrit la méthode de FVMLPG pour les problèmes C^0 , appliquée spécifiquement aux problèmes de contraintes planes. Une forme faible locale est développée à partir de la forme résiduelle pondérée classique de l'équation d'équilibre. Une interpolation des moindres carrés mobiles est utilisée pour construire les approximations à la solution connue sous le nom de fonctions d'essai. Des fonctions test sont choisies dans un espace différent de celui des fonctions d'essai (c'est le cas de la méthode de Petrov-Galerkin). Un système d'équations algébriques est déduit en substituant les fonctions d'essai et test dans la forme faible locale. La méthode est

évaluée en l'appliquant à plusieurs problèmes de patch tests avec des conditions aux limites mixtes. Une comparaison en coûts de calcul et en exactitude par rapport à la MEF est présentée.

Le chapitre 4 concerne le développement la méthode de FVMLPG pour les problèmes d'ordre C^1 , telle les poutres d'Euler Bernoulli. Ces problèmes sont décrits par des équations différentielles ordinaires de quatrième ordre. Dans les problèmes C^1 , les variables dépendantes et leurs premières dérivées sont continues, mais les dérivés d'ordre supérieur peuvent ne pas l'être. Une forme faible locale (LWF) de l'équation d'équilibre est développée. Les interpolations par les moindres carrés mobiles sont généralisées pour inclure des dérivés des variables dépendantes afin de construire les fonctions d'essai. Les fonctions d'essai sont choisies dans un espace différent de celui des fonctions test. Les fonctions d'essai et test sont alors utilisée dans le LWF pour déduire un système d'équations algébriques.

Dans le chapitre 5, la méthode de FVMLPG est étendue aux problèmes élastodynamique bidimensionnels de classe C^0 . Cette approche dérive de la forme faible locale des équations d'équilibre en utilisant le concept général de la méthode des volumes finis (VF) et de la méthode de « Meshless Local Petrov-Galerkin » (MLPG). En incorporant les approximations des moindres carrés mobiles (MLS) pour la fonction d'essai, la forme faible locale est discrétisée puis intégrée au-dessus du volume de contrôle ou du sous domaine local pour l'analyse structurale modale. La méthode est testée sur plusieurs cas d'analyse modale des structures élastiques. Les résultats démontrent l'efficacité et l'exactitude de cette résolution par rapport à la méthode des éléments finis de Galerkin.

Enfin, dans le chapitre 6, une extension de cette méthode au problème de stabilité élastique est présentée.

Chapitre 2

Méthode « FVMLPG » appliquée aux problèmes unidimensionnels de classe C^0

Une nouvelle méthode intitulée « finite volume meshless local Petrov Galerkin » (FVMLPG) est développée pour les problèmes unidimensionnels (1-D) de C^0 . Ce chapitre présente une description détaillée de cette méthode appliquée aux problèmes unidimensionnels.

D'abord, une forme faible locale est développée à partir de la forme résiduelle pondérée classique de l'équation d'équilibre. Une interpolation des moindres carrés mobiles est utilisée pour construire les approximations. Des fonctions tests sont choisies dans un espace différent de celui des fonctions d'essai considérées dans la méthode de Petrov-Galerkin. Un système d'équations algébriques est obtenu en substituant les fonctions d'essai et les fonctions tests dans la forme faible locale. La méthode est évaluée en l'appliquant sur plusieurs exemples avec différentes conditions aux limites.

2.1 Forme faible pour les problèmes 1-D de classe C^0

Soit une fonction $u(x)$ de classe C^0 dans un domaine Ω

$$-\frac{d}{dx}\left(b(x)\frac{du}{dx}\right) + c(x)u = f(x) \tag{2.1.1}$$

dans le domaine Ω ($0 \leq x \leq l$) avec la frontière Γ , où $b(x)$ et $c(x)$ sont les paramètres du problème qui peuvent être des fonctions de x et $f(x)$ la fonction décrivant le chargement. Les conditions aux limites essentielles et naturelles sont :

$$u = \tilde{u} \quad \text{sur } \Gamma_u, \quad q = \tilde{q} \quad \text{sur } \Gamma_q \quad (2.1.2)$$

où

$$q = b \frac{du}{dx} \quad (2.1.3)$$

et Γ_u et Γ_q sont les régions de frontière où la variable primaire « u » et la variable secondaire « q » sont respectivement imposées. Dans les problèmes 1-D, ces régions de frontière sont les points $x=0$ et $x=l$. Les variables u et q représentent différentes quantités physiques selon le type de problème considéré. Par exemple, dans le problème de la déformation axiale d'une barre, la variable primaire u est le déplacement longitudinal, $b=EA$ où E est le module d'élasticité et A la section, f est la force appliquée sur la surface de la barre, et $b(du/dx)$, la variable secondaire, est la force axiale. Pour un problème de transfert thermique, u est la température, b est la conductivité thermique, f est la conductivité thermique, et $b(du/dx)$ est le flux de la chaleur [100].

Pour obtenir une solution approximative à l'équation (2.1.1), une technique résiduelle pondérée est utilisée. L'erreur résiduelle pour cette solution approchée est

$$R = -\frac{d}{dx} \left(b \frac{du}{dx} \right) + cu - f \quad (2.1.4)$$

Le contrôle des erreurs est effectué en multipliant le résiduel par une fonction de poids $v(x)$, intégrée au-dessus du domaine entier, et en posant l'intégrale égale à zéro.

$$\int_{\Omega} v \left[-\frac{d}{dx} \left(b \frac{du}{dx} \right) + cu - f \right] dx = 0 \quad (2.1.5)$$

L'équation (2.1.5) représente la forme résiduelle pondérée classique de l'équation différentielle. Une solution approximative pour u est choisie telle que chaque limite dans la solution approchée doit être deux fois différentiable et satisfasse toutes les conditions aux limites (Eq. 2.1.2). Ces conditions sont généralement faciles à respecter pour les problèmes 1-D, par contre pour des

dimensions élevées, le problème devient compliqué. Par conséquent, une formulation qui accepte des conditions plus faibles sur u est recherchée. La forme faible de l'équation résiduelle pondérée est mise en place en transférant la différentiation à partir de la variable primaire u à la fonction du poids v . Ceci est réalisé par l'intégration par partie en 1-D et par l'application du théorème de divergence en 2-D et 3-D.

En intégrant l'équation (2.1.5) par partie, on obtient :

$$\int_{\Omega} b \frac{du}{dx} \frac{dv}{dx} dx + \int_{\Omega} cvu dx - \int_{\Omega} fv dx - \left[vb \frac{du}{dx} \right]_{\Gamma} = 0 \quad (2.1.6)$$

Cette intégration par partie produit un terme de frontière $[vb(du/dx)]$. L'imposition de la variable secondaire $b(du/dx)$ sur Γ constitue la condition au limite naturelle (NBC) qui fait partie maintenant de la forme faible. Les conditions pour la solution approximative ont été ainsi réduites, c.-à-d., u doit être différentiable une fois et doit satisfaire seulement les conditions de frontière essentielles alors que les NBCs sont inclus sous la forme faible. Dans l'équation (2.1.6), appelée la forme faible de l'équation différentielle, les fonctions d'approximation choisies pour u et v s'appellent respectivement fonction test et fonction d'essai. Cette forme faible est le point de départ de la méthode des éléments finis (MEF).

Dans la méthode des éléments finis (MEF), u est choisi comme une fonction linéaire par morceaux (exemple représenté sur la figure 2.1.1).

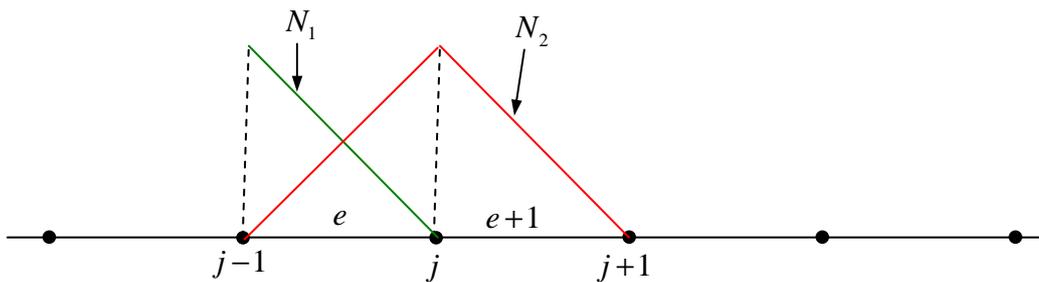


Figure 2.1.1 fonctions d'essai (forme) des éléments finis au noeud j

Les fonctions d'essai pour l'élément e sont choisies comme suit :

$$u^{(e)} = N_1 u_{j-1}^{(e)} + N_2 u_j^{(e)} \quad (2.1.7)$$

N_1 et N_2 sont des fonctions de forme de l'élément $e^{ème}$ et $u = \sum_{e=1}^{N_{el}} u^{(e)}$ et N_{el} est le nombre d'éléments dans le modèle. La fonction test v est choisie comme variation de u

$$v^{(e)} = \delta u^{(e)} = N_1 \delta u_{j-1}^{(e)} + N_2 \delta u_j^{(e)}. \quad (2.1.8)$$

Ces choix procurent plusieurs avantages à la méthode EF :

- les fonctions d'essai sont linéaires par morceaux, la méthode EF a un caractère local, et la matrice de rigidité est par conséquent une matrice bande,
- le choix $v = \delta u$ conduit à une matrice de rigidité symétrique,
- la matrice de rigidité devient définie positive après l'imposition des conditions aux limites (la première fonction à intégrer dans l'équation (2.1.6) représente une quantité d'énergie).

Les variables secondaires sont habituellement les grandeurs recherchées dans cette analyse. Pour les problèmes C^0 considérés ici, la variable secondaire s'écrit :

$$q = b \frac{du}{dx}. \quad (2.1.9)$$

La variable secondaire q_j pour la fonction d'essai u_j (figure 2.1.1) est la pente au noeud j . Les pentes au noeud j pour les éléments e et $e+1$ sont évidemment inégales. En général, toutes les variables secondaires dans la méthode EF sont discontinues à travers les frontières d'élément en raison de la nature de l'approximation par morceaux pour les fonctions de forme. Des techniques de traitement post–processeur sont indispensables pour réaliser des distributions lisses pour les variables secondaires. Ceci est l'un des inconvénients de la méthode EF.

Pour surmonter ce problème de discontinuité dans les EF, Nayroles et d'autres chercheurs ont proposé une formulation dite « éléments diffus » [101]. Ce concept a été utilisé par la suite par Belytschko *et autres* [102] qui ont introduit la méthode dite « Element Free Galerkin » (EFG). Dans ces méthodes aucun élément n'est présent, et les fonctions d'essais u sont constituées en passant une fonction lisse par des valeurs nodales fictives (section 2.2). Ces fonctions d'essai sont écrites comme dans les méthodes d'EFG [103]

$$u(x) \cong \sum_{j=1}^n \hat{u}_j \phi_j(x) \quad (2.1.10a)$$

n est le nombre de noeuds dans le domaine de définition de la fonction d'essai, \hat{u}_j sont les valeurs nodales fictives du déplacement, et $\phi_j(x)$ sont des fonctions de forme. Comme les fonctions d'essai sont lisses, les variables secondaires sont continues à chaque point dans le domaine des fonctions d'essai. En utilisant la méthodologie de Galerkin, les fonctions d'essai sont choisies comme variation de u et de $v = \delta u$, et sont écrites de la même manière que les fonctions d'essai.

$$v(x) \cong \mu_i^{(u)} \chi_i^{(u)}(x) \quad (2.1.10b)$$

$\mu_i^{(u)}$ sont les constantes arbitraires pour le déplacement, et $\chi_i^{(u)}$ sont les composantes des fonctions d'essai. Les détails du développement des fonctions test et d'essai sont présentés dans la section 2.2. Le domaine d'intégration pour les termes i - j de la forme faible de l'équation (2.6) est l'intersection des fonctions d'essai et test (région Ω_d , sur la Figure 2.1.2).

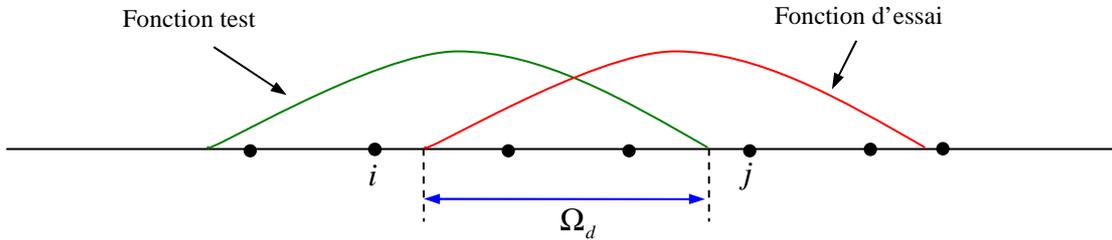


Figure 2.1.2 Fonctions d'essai et test et domaine d'intégration

Ce domaine peut être grand, et surtout sa forme est difficile à déterminer pour les problèmes en 2-D et 3-D. Puisqu'une forme bien définie est souhaitable pour les problèmes d'intégration, un maillage de fond (maillage rectangulaire en 2-D [102]) est utilisé. En conséquence, l'élaboration des fonctions test et d'essai n'exige pas d'éléments. L'utilisation de ce maillage de fond permettant l'intégration, réduit l'avantage de la méthode d'EFG et par conséquent cette méthode n'est pas véritablement une méthode sans maillage.

Pour développer une méthode véritablement sans maillage, Atluri et Zhu [104] ont suggéré le choix de la fonction d'essai dans une espace différent, et par conséquent,

$$v \neq \delta u \quad (2.1.11)$$

Par exemple, une fonction de poids dont les valeurs non nulles définissent une forme bien définie peut être utilisée. Les formes communes dans 2-D incluent des cercles, des ellipses, et des rectangles. Une fonction test commune v_i pour le noeud i dans 1-D (en comparaison d'une fonction d'essai pour le noeud j) est présentée sur la figure 2.1.3. Ces fonctions test peuvent être choisies pour s'annuler à une certaine distance contrôlable, R_0 , du noeud i . Cette propriété locale des fonctions tests donne à la méthode son caractère local.

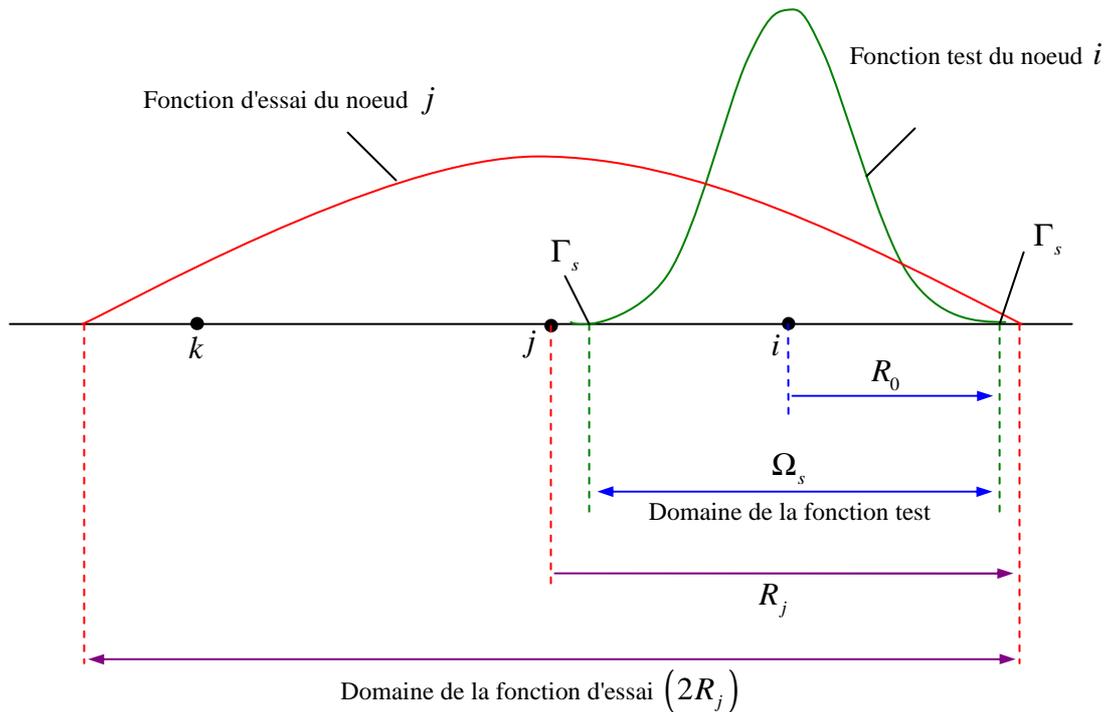


Figure 2.1.3 Comparaison des domaines des fonctions d'essai et test

En raison de la forme bien définie et des valeurs nulles en dehors du sous domaine local Ω_s les intégrations des fonctions d'essai peuvent par conséquent être limitées à Ω_s et s'obtiennent par extension des fonctions d'essai (figure 2.1.3). Ce choix élimine ainsi le maillage fictif en arrière plan. Le choix de la fonction test dans un espace différent de celui de la fonction d'essai conduit à la méthode de Petrov-Galerkin. Cette méthode s'appelle aussi « Meshless Local Petrov Galerkin » (MLPG) [104].

La forme faible pour le sous domaine local Ω_s s'écrit donc :

$$\int_{\Omega_s} b \frac{du}{dx} \frac{dv}{dx} dx + \int_{\Omega_s} cvu dx - \int_{\Omega_s} fv dx - \left[vb \frac{du}{dx} \right]_{\partial\Omega_s} = 0. \quad (2.1.12)$$

En utilisant la fonction de Heaviside

$$v_i = \begin{cases} 0 & x < x_{i-1} \\ 1 & x_{i-1} \leq x \leq x_{i+1} \\ 0 & x > x_{i+1} \end{cases} \quad (2.1.13)$$

Comme fonction test sous la forme faible symétrique locale dans l'équation (2.1.12), on obtient ainsi :

$$\int_{\Omega_s} cu dx - \int_{\Omega_s} f dx - \left[b \frac{du}{dx} \right]_{\partial\Omega_s} = 0. \quad (2.1.14)$$

Etant donné que la frontière locale $\partial\Omega_s$ peut intersecter la frontière globale Γ , $\partial\Omega_s$ est divisé en sous-ensembles qui couvrent toutes les conditions limites possibles.

$$\partial\Omega_s \cap \Gamma_u, \partial\Omega_s \cap \Gamma_q, L_s \quad (\Omega_s \text{ est complètement à l'intérieur de } \Omega). \quad (2.1.15)$$

En considérant la condition limite de l'équation (2.1.2), l'équation (2.1.14) s'écrit :

$$\int_{\Omega_s} cu dx - \int_{\Omega_s} f dx - [q]_{\Gamma_{su}} - [\tilde{q}]_{\Gamma_{sq}} - [q]_{L_s} = 0 \quad (2.1.16)$$

L'équation (2.1.16) représente l'équation d'équilibre du sous domaine local Ω_s comme pour la méthode VF conventionnelle avec les conditions aux limites de traction. La forme faible de l'équation (2.1.16) est locale parce que les intégrations sont effectuées au-dessus du sous domaine local Ω_s . Si les fonctions d'essai et test de l'équation (2.1.16) sont choisies dans un même espace par l'intermédiaire d'une méthode de Galerkin, l'évaluation des termes de l'équation (2.1.16) conduit aux matrices de rigidité symétriques. Ainsi la forme faible peut s'appeler dans «forme faible symétrique locale» (LSWF).

Dans ce travail, la méthode de Petrov Galerkin est utilisée. Les matrices résultantes de rigidité ne sont pas symétriques, et le terme symétrique est omis ainsi. La substitution de la fonction d'essai dans l'équation (2.1.16) conduit à un système d'équations sous la forme :

$$K^{(node)}\hat{u} + K^{(bdry)}\hat{u} - f^{(node)} - f^{(bdry)} = 0 \quad (2.1.17)$$

L'indice supérieur "bdry" dénote la frontière, et \hat{u} est la valeur nodale fictive \hat{u}_j (équation 2.10a).

Le système d'équations est présenté en détail dans la section 2.3.

Considérant maintenant les deux termes de l'équation (2.1.16)

$$[q]_{\Gamma_{su}} \text{ et } [\tilde{q}]_{\Gamma_{sq}} \quad (2.1.18)$$

ces termes doivent être évalués aux points de frontière. Les détails de ces évaluations sont expliqués à l'aide d'un domaine 1-D modélisé avec 17 noeuds équidistants (figure 2.1.4). L'espacement nodal dans ce modèle est $\Delta x = l/16$. On suppose que la variable primaire, u , est donnée au noeud 1, et la variable secondaire « q » au noeud 17. Sur la figure 2.1.4, les fonctions test sont indiquées sur certains noeuds dans le modèle.

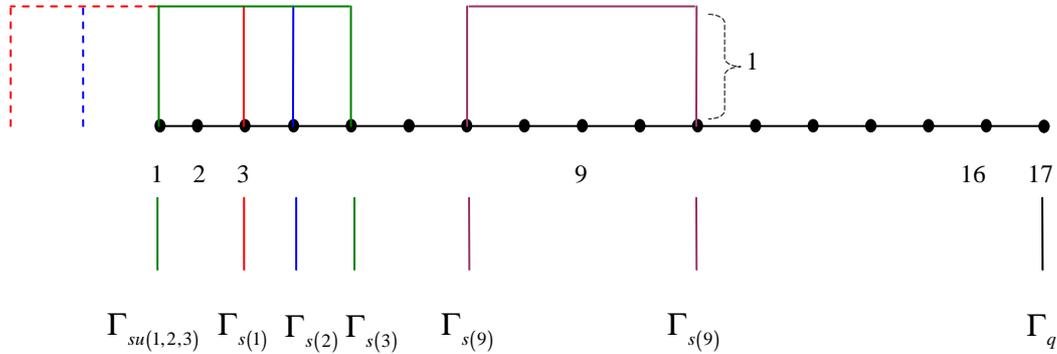


Figure 2.1.4 Fonctions test pour divers noeuds dans un modèle à 17 noeuds

Le terme $[q]_{\Gamma_{su}}$ doit être évalué pour chaque noeud du modèle dont le Ω_s est intersecté par Γ_u . Le modèle de la figure 2.1.4 comporte trois noeuds (noeud 1, noeud 2, et noeud 3). La clef de contribution de chacun des noeuds au terme $[q]_{\Gamma_{su}}$ se situe en valeurs de v_1 , de v_2 , et de v_3 au noeud 1, où $x = 0$.

v_i : valeur de la fonction test au noeud i :

par exemple pour le nœud 3

$$v_3 = 0 \text{ au nœud 1,} \quad (2.1.19)$$

et donc

$$[q]_{\Gamma_{su}} = 0. \quad (2.1.20)$$

Pour le nœud 1

$$v_1 = 1 \text{ au nœud 1,} \quad (2.1.21)$$

en utilisant les équations (2.1.9 et 2.1.10a), on obtient :

$$[q]_{\Gamma_{su}} = \left[b \frac{du}{dx} \right]_{\Gamma_{su}} = b \left[\frac{d\phi_1}{dx} \quad \frac{d\phi_2}{dx} \quad \dots \quad \frac{d\phi_n}{dx} \right]_{x=0} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix} \quad (2.1.22)$$

Pour le nœud 2

$$v_2 = 1 \text{ au nœud 1,} \quad (2.1.23)$$

et donc

$$[q]_{\Gamma_{su}} = \left[b \frac{du}{dx} \right]_{\Gamma_{su}} = b \left[\frac{d\phi_1}{dx} \quad \frac{d\phi_2}{dx} \quad \dots \quad \frac{d\phi_n}{dx} \right]_{x=0} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{bmatrix} \quad (2.1.24)$$

Les termes $b \left[\frac{d\phi_1}{dx} \quad \frac{d\phi_2}{dx} \quad \dots \quad \frac{d\phi_n}{dx} \right]_{\Gamma_{su}}$ dans les équations (2.1.22 et 2.1.24) sont évalués au nœud 1 et contribuent à l'évaluation de la matrice $\mathbf{K}^{(\text{bdry})}$ de l'équation (2.1.17). La contribution

du noeud 2 au terme $[q]_{\Gamma_{su}}$, et finalement à la matrice $\mathbf{K}^{(\text{bdry})}$, est d'une importance extrême et ne peut pas être négligée.

Le terme $[\tilde{q}]_{\Gamma_{sq}}$ contribue au vecteur force $\mathbf{f}^{(\text{bdry})}$ de l'équation (2.1.17) et doit être évalué pour chaque noeud du modèle dont le Ω_s intersecté Γ_q . Pour un noeud dont $v = 0$ au noeud 17,

$$[\tilde{q}]_{\Gamma_{sq}} = 0. \quad (2.1.25)$$

Pour un noeud dont $v \neq 0$ au noeud 17, $[\tilde{q}]_{\Gamma_{sq}}$ n'est pas évalué en tant que zéro à moins que la variable secondaire imposée soit zéro. La contribution de tels noeuds au terme $[\tilde{q}]_{\Gamma_{sq}}$, et finalement au vecteur force $\mathbf{f}^{(\text{bdry})}$, est également d'une importance extrême et ne peut pas être négligée.

Un arrangement approprié de la façon dont les termes de l'équation (2.1.18) sont calculés donne la possibilité aux futurs utilisateurs de la méthode de FVMLPG la liberté considérable dans les choix de l'espacement nodal et des tailles des fonctions test. Pour le cas présenté sur la figure 2.1.4 d'un modèle avec des noeuds équidistants, le choix d'un plus petit (R_o / l) pour les noeuds 2 et $N - 1$ (par exemple, ici (R_o / l) = Δx pour les noeuds 2 et 16) assure que $[\tilde{q}]_{\Gamma_{sq}} = 0$ et peut être ainsi préférable.

Cependant, les noeuds n'ont pas besoin d'être équidistants. De même, la taille de Ω_s pour chaque v_i n'a pas besoin d'être uniforme. Quand c'est le cas, le choix simple d'un plus petit (R_o / l) pour les noeuds 2 et le $N - 1$ ne permet pas de garantir que tous les termes de l'équation de l'équation (2.1.18) soient tous nuls pour des noeuds additionnels près des frontières. Par ailleurs, les utilisateurs de l'algorithme de FVMLPG ne peuvent pas supposer qu'une attribution simple de (R_o / l) expliquera les termes de l'équation (2.1.18) comme dans l'exemple ci-dessus. Afin d'exploiter la pleine utilité de la méthode, les termes de l'équation (2.1.18) doivent être évalués.

2.2 Approximation des moindres carrés mobiles

2.2.1 Méthode moindres carrés

Considérant un problème classique de données bien connues. Soit la fonction $u^h(x)$ une fonction d'application pour les points (x_i, u_i) (figure 2.2.1) avec $u_i = u(x_i)$.

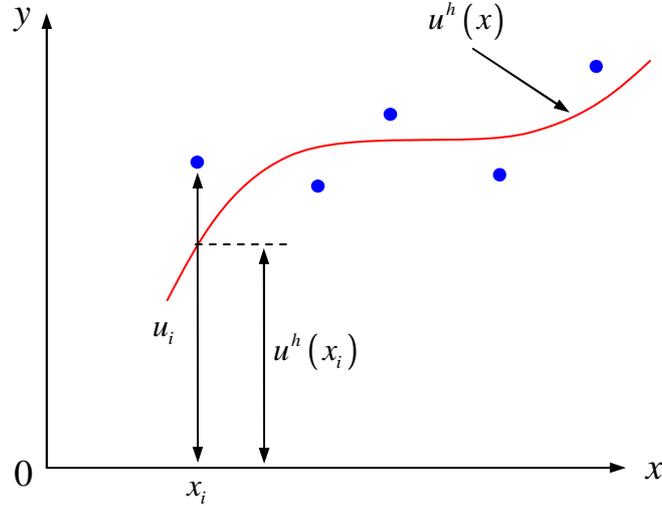


Figure 2.2.1 Ajustement par moindres carrés

Supposant que la fonction d'approximation $u^h(x)$ est un polynôme d'ordre m

$$u^h(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m \quad (2.2.1)$$

Soit sous une forme compacte

$$u^h(x) = p^T(x)a \quad (2.2.2)$$

Les paramètres « a » peuvent être déterminés en minimisant le carré de différence entre u_i et $u^h(x)$

$$J = \sum_{i=1}^n [u^h(x_i) - u_i]^2 = \sum_{i=1}^n [p^T(x_i)a - u_i]^2 \quad (2.2.3)$$

La différenciation par rapport à « a » conduit à l'équation suivante

$$\sum_{i=1}^n \mathbf{p}(x_i) \mathbf{p}^T(x_i) \mathbf{a} = \sum_{i=1}^n \mathbf{p}(x_i) u_i \quad (2.2.4)$$

Ce qui nous permet de déterminer les inconnus \mathbf{a} , puis la fonction d'approximation $u^h(x)$.

Soit l'exemple suivant :

x_i	1	2	3
u_i	1	1.5	4

$$\mathbf{p}^T(x) = [1 \quad x] \quad \mathbf{a}^T = [a_0 \quad a_2] \quad (2.2.5)$$

L'équation (2.2.4) devient maintenant

$$\sum_{i=1}^3 \begin{bmatrix} 1 & x_i \\ x_i & x_i^2 \end{bmatrix} \mathbf{a} = \sum_{i=1}^3 \begin{bmatrix} 1 \\ x_i \end{bmatrix} u_i \quad (2.2.6)$$

$$\begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix} \mathbf{a} = \begin{bmatrix} 6.5 \\ 16 \end{bmatrix} \quad (2.2.7)$$

La solution est $a_0 = -5/6$ et $a_1 = 1.5$, par conséquent l'approximation est donnée par

$$u^h(x) = -\frac{5}{6} + \frac{3}{2}x \quad (2.2.8)$$

2.2.2 Moindres carrés pondérés

Le modèle linéaire est remplacé par le modèle « moindres carrés pondérés » dont l'objectif est de réduire l'influence de points situés dans des zones de forte variance.

L'ajustement des moindres carrés pondérés s'écrit dans ce cas :

$$J = \sum_{i=1}^n w(x_i - x_0) [u^h(x_i) - u_i]^2 \quad (2.2.9)$$

2.2.3 Méthode des moindres carrés mobiles

Cette méthode est développée à la fin des années 60 par Shepard [105] comme un moyen de lisser l'approximation autour des valeurs nodales.

Dans cette méthode, le procédé est exactement le même que les moindres carrés pondérés.

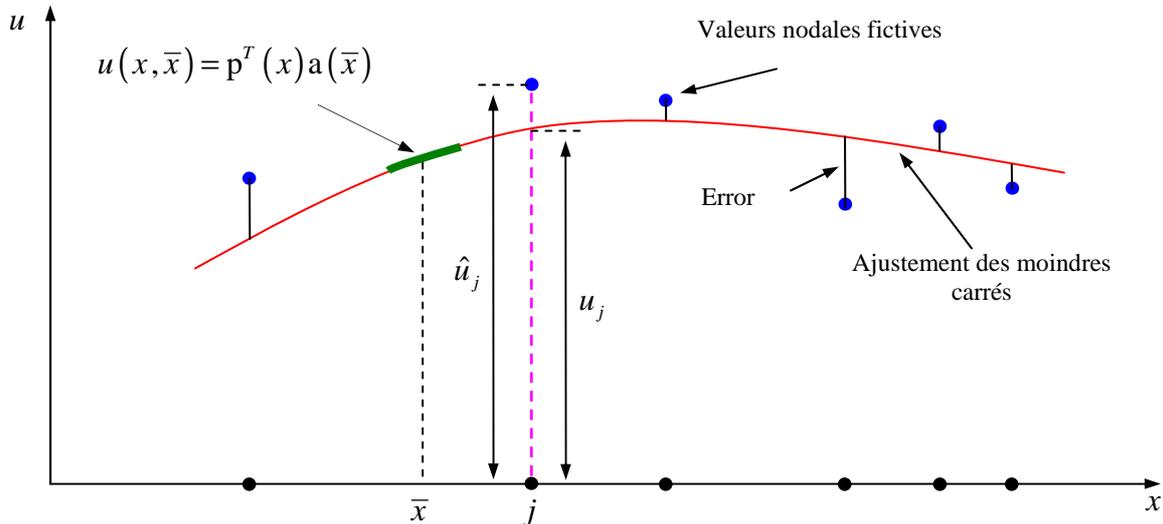


Figure 2.2.1 Interpolation des moindres carrés mobiles (MLS)

Les interpolations de MLS sont utilisées pour former les fonctions d'essai « u » de la méthode de FVMLPG. On suppose que les fonctions d'essai sont lisses et non nulles au-dessus d'une distance contrôlable R_j du nœud j . Cette distance R_j est habituellement choisie pour couvrir un domaine beaucoup plus large que celui des fonctions de forme des éléments finis (figure 2.2.2). Une approximation de type MLS peut alors être faite pour u^h (u^h est la valeur de u dans le domaine Ω_h). La valeur pour u^h est égale à zéro à l'extérieur du domaine Ω_h .

L'approximation de la fonction $u(x)$, à savoir $u^h(x)$, repose sur une base de fonction polynomiale d'ordre m à coefficients *non constants*. L'approximation locale autour d'un point \bar{x} est donnée par :

$$u(x, \bar{x}) \cong u^h(x, \bar{x}) = p^T(x) a(\bar{x}) \quad (2.2.10)$$

$p(x)$ est un polynôme complet d'ordre m

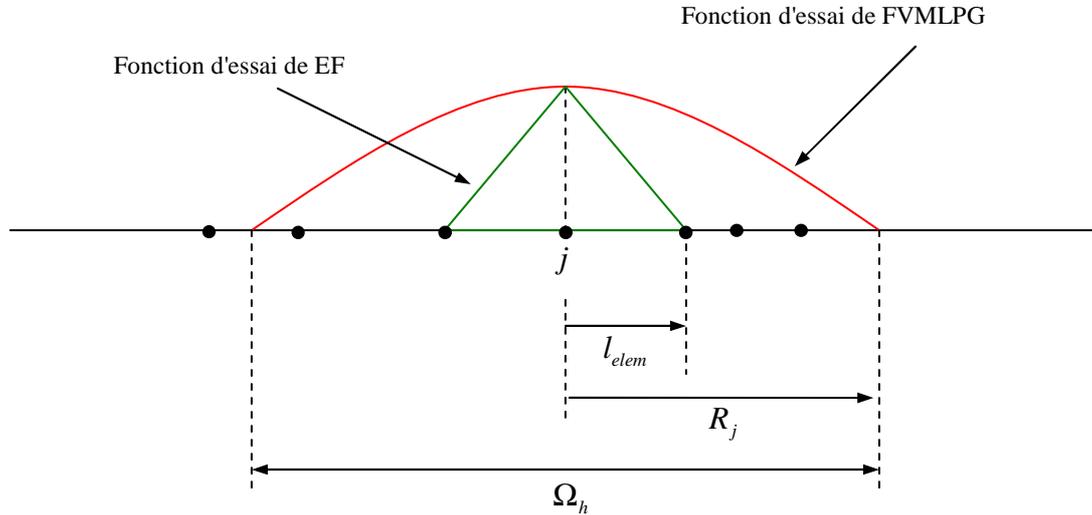


Figure 2.2.2 Comparaison entre les fonctions d'essai des EF et de FVMLPG

$$p^T(x) = [1 \quad x \quad x^2 \quad \cdots \quad x^m] \quad (2.2.11)$$

et $a(x)$ contient des coefficients non constants

$$a^T(x) = [a_0(x) \quad a_1(x) \quad a_2(x) \quad \cdots \quad a_m(x)] \quad (2.2.12)$$

Les paramètres inconnus $a_j(x)$ à un point quelconque sont déterminés en minimisant la différence entre l'approximation locale en ce point et les paramètres nodaux u_l .

En réduisant au minimum la norme pondérée L_2 par rapport à $a(x)$ on obtient

$$\begin{aligned} J &= \sum_{l=1}^n w(x-x_l) [u_L^h(x_l, x) - u_l]^2 \\ &= \sum_{l=1}^n w(x-x_l) [p^T(x_l) a(x) - u_l]^2 \end{aligned} \quad (2.2.13)$$

n est le nombre de noeuds dans le voisinage x , d'où la fonction de poids $w(x-x_i) \neq 0$.

Le minimum de J , dans l'équation (2.2.13) par rapport au coefficient $a(x)$ peut être obtenu en posant égale à zéro la dérivée de J par rapport $a(x)$

$$\begin{aligned} \sum_{l=1}^n w(x-x_l) 2p_1(x_l) [p^T(x_l)a(x)-u_l] &= 0 \\ \sum_{l=1}^n w(x-x_l) 2p_2(x_l) [p^T(x_l)a(x)-u_l] &= 0 \\ \vdots & \\ \sum_{l=1}^n w(x-x_l) 2p_m(x_l) [p^T(x_l)a(x)-u_l] &= 0 \end{aligned} \tag{2.2.14}$$

Soit en écriture vectorielle :

$$\sum_{l=1}^n w(x-x_l) 2p(x_l) [p^T(x_l)a(x)-u_l] = 0 \tag{2.2.15}$$

L'élimination du facteur 2 conduit à :

$$\sum_{l=1}^n w(x-x_l) p(x_l) p^T(x_l) a(x) = \sum_{l=1}^n w(x-x_l) p(x_l) u_l \tag{2.2.16}$$

Soit en notation compacte :

$$A(x)a(x) = B(x)u \tag{2.2.17}$$

Où

$$A(x) = \sum_{l=1}^n w(x-x_l) p(x_l) p^T(x_l) \tag{2.2.18}$$

et

$$B(x) = [w(x-x_1)p(x_1) \quad w(x-x_2)p(x_2) \quad \cdots \quad w(x-x_n)p(x_n)] \tag{2.2.19}$$

La détermination de $a(x)$ de l'équation (2.2.17) et la substituant dans l'équation (2.2.10), construit à l'approximation MLS suivante :

$$u^h(x) = p^T(x) [A(x)]^{-1} B(x) u \quad (2.2.20)$$

L'introduction de la forme d'approximation bien connue

$$u^h(x) = \sum_{I=1}^n \Phi_I(x) u_I = \Phi^T(x) u \quad (2.2.21)$$

Permet d'écrire les fonctions de forme de MLS

$$\Phi^T(x) = p^T(x) [A(x)]^{-1} B(x) \quad (2.2.22)$$

La fonction de forme Φ_I pour un nœud I à un point x s'écrit :

$$\Phi_I(x) = p^T(x) [A(x)]^{-1} w(x-x_I) p(x_I) \quad (2.2.23)$$

La matrice $A(x)$ s'appelle « matrice moment », de taille $m \times m$. Cette matrice doit être inversée partout où les fonctions de forme de MLS sont évaluées. Ce qui constitue un inconvénient principal des méthodes sans maillage (temps de calcul élevé ; possibilité d'obtenir une matrice de moment singulière).

Soit une base linéaire dans une dimension, la matrice moment s'écrit :

$$A(x) = w(x-x_1) \begin{bmatrix} 1 & x_1 \\ x_1 & x_1^2 \end{bmatrix} + w(x-x_2) \begin{bmatrix} 1 & x_2 \\ x_2 & x_2^2 \end{bmatrix} + \dots + w(x-x_n) \begin{bmatrix} 1 & x_n \\ x_n & x_n^2 \end{bmatrix} \quad (2.2.24)$$

Il est clair que si dans cette équation $n=1$ c.-à-d., le point x est couvert par seulement un appui nodal tandis que la base est linéaire ($m=2$), alors la matrice est singulière et ne peut pas être inversée. Par conséquent, la condition pour que la matrice moment soit inversible est que $n \geq m$.

Les calculs de la fonction de forme de MLS comme ses dérivés impliquent l'inverse de la matrice moment qui devient onéreuse en deux et trois dimensions.

Une fonction de poids est généralement utilisée dans des méthodes sans maillage. Il s'agit de la fonction de poids de spline quartique. La fonction de poids de spline est donnée par l'équation (2.2.25),

$$w_i(x) = \begin{cases} 1 - 6\left(\frac{d_i}{r_i}\right)^2 + 8\left(\frac{d_i}{r_i}\right)^3 - 3\left(\frac{d_i}{r_i}\right)^4 & 0 \leq d_i \leq r_i \\ 0 & d_i \geq r_i \end{cases} \quad (2.2.25)$$

$d_i = |x - x_i|$ est la distance euclidienne du noeud x_i au point x ; et r_i est la taille du support de la fonction de poids w_i qui détermine l'appui du noeud x_i . Si la fonction de poids de spline de l'équation (2.2.25) est de l'ordre C^1 continue au-dessus du domaine entier Ω , alors les fonctions de forme $\Phi(x)$ et la fonction d'essai sont également de classe C^1 .

2.3 Équations du système

La fonction d'essai de l'équation 2.1.10 et la fonction test de l'équation 2.1.13 sont substituées dans la forme faible locale (équation 2.1.12) pour obtenir les matrices du système :

$$\int_{\Omega_s} cu \, dx - \int_{\Omega_s} f \, dx - [q]_{\Gamma_{su}} - [\tilde{q}]_{\Gamma_{sq}} - [q]_{L_s} = 0 \quad (2.3.1)$$

La variable primaire, u , est approchée en utilisant l'équation (2.2.21)

$$u(x) = \sum_{j=1}^n \phi_j(x) \hat{u}_j \quad (2.3.2)$$

ϕ_j sont les fonctions de forme, et \hat{u}_j sont les valeurs nodales fictives de u . En substituant l'équation (2.3.2) dans l'équation (2.3.1), la forme faible locale est discrétisée par :

$$\int_{\Omega_s} c \sum_{j=1}^n \phi_j \hat{u}_j dx - \int_{\Omega_s} f dx - [q]_{\Gamma_{su}} - [\tilde{q}]_{\Gamma_{sq}} - [q]_{L_s} = 0 \quad (2.3.3)$$

Dans cette équation il n'y a aucune dérivée de la fonction de forme dans les intégrales locales. C'est l'un des avantages de la méthode de FVMLPG parce qu'il est bien connu que l'approximation sans maillage n'est pas efficace en calculant de telles dérivées partout dans le domaine, particulièrement pour les approximations de type de MLS.

La substitution des fonctions de forme dans l'équation (2.3.3) conduit au système d'équations suivant :

$$\mathbf{K}^{(node)} \hat{\mathbf{u}} + \mathbf{K}^{(bdry)} \hat{\mathbf{u}} - \mathbf{f}^{(node)} - \mathbf{f}^{(bdry)} = 0 \quad (2.3.4)$$

$$K_{ij}^{(node)} = \int_{\Omega_s^{(i)}} c \phi_j dx \quad (2.3.5a)$$

$$K_{ij}^{(bdry)} = \left[\frac{d\phi_j}{dx} \right]_{\Gamma_{su}^{(i)}} \quad (2.3.5b)$$

$$f_i^{(node)} = \int_{\Omega_s^{(i)}} f dx \quad (2.3.6a)$$

$$f_i^{(bdry)} = [\tilde{q}]_{\Gamma_{sq}^{(i)}} \cdot \quad (2.3.6b)$$

La matrice de « rigidité » \mathbf{K} , composée de $\mathbf{K}^{(node)}$ et de $\mathbf{K}^{(bdry)}$, n'est pas symétrique. Les matrices antisymétriques ne sont pas nécessairement indésirables. Plusieurs méthodes numériques, comme la méthode d'éléments de frontière ou la méthode de collocation de sous domaine, ont comme conséquence les matrices antisymétriques. Dans cette formulation sans maillage, l'antisymétrique \mathbf{K} n'est pas incorrecte parce que, à la différence de la méthode EF, la matrice de \mathbf{K} dans la méthode sans maillage n'est pas évaluée à partir de l'énergie de déformation de traction du problème, mais obtenu en imposant le résiduel pondéré égal à zéro au sens intégral.

L'intégration numérique est utilisée pour résoudre le système d'équations (équations 2.3.5a et 2.3.6a). Dans le schéma d'intégration de Gauss, un point gaussien intégrera de façon exacte un polynôme de degré de $2n-1$. Les équations (2.2.23, 2.2.18, et 2.2.19) sont rappelées ici par convenance.

$$\phi_j(x) = \sum_{g=1}^m p_g(x) [A^{-1}B]_{gj} \quad (2.3.7)$$

$$\phi_{j,x}(x) = \sum_{g=1}^m \left[p_{g,x}(A^{-1}B)_{gi} + p_g(A^{-1}B_{,x} - A^{-1}A_{,x}A^{-1}B)_{gi} \right] \quad (2.3.8)$$

$$[A] = P^T W P = [B] P = \sum_{j=1}^n w(x-x_j) p(x_j) p^T(x_j) \quad (2.3.9)$$

et

$$[B] = P^T W = \left[w(x-x_1) p(x_1) \quad w(x-x_2) p(x_2) \quad \cdots \quad w(x-x_n) p(x_n) \right] \quad (2.3.10)$$

L'ordre d'intégration gaussienne exigé pour des résultats acceptables dépend des fonctions de base et des fonctions de poids utilisées. L'ordre le plus élevé de la fonction de base considérée est quadratique (x^2). La fonction de poids disponible pour construire les fonctions d'essai est la fonction quartique de poids de spline (équation 2.2.25). Une quadrature gaussienne plus élevée intégrerait avec succès les termes des équations (2.3.5a et 2.3.6a). Par simulation numérique on montre que trois points de quadrature gaussienne donnent de très bons résultats. Par conséquent, c'est cette méthode qui est utilisée pour la résolution des problèmes du paragraphe 2.5.

2.4 Méthode de collocation (conditions aux limites essentielles - Dirichlet)

L'imposition des conditions limites essentielles (EBCs) dans les méthodes de FVMLPG est difficile parce que les fonctions de forme de l'approximation des moindres carrés mobiles n'ont pas la propriété du symbole de Kronecker. Théoriquement, les fonctions des moindres carrés mobiles (MLS) ne passent pas par les valeurs nodales fictives utilisées pour le lissage comme dans les EF.

$$\phi_j(x_k) \neq \delta_{jk} \quad (2.3.11)$$

$\phi_j(x_k)$ est la fonction de forme pour le noeud j évaluée au point nodal k , et δ_{jk} est le symbole de Kronecker.

Puisque les conditions aux limites ne peuvent pas être directement imposées, une méthode de collocation est appliquée. Pour un point nodal x_j , si son degré de liberté (DL) en déplacement « j » appartient à la frontière de déplacement, c.-à-d. $u_j \in \Gamma_{su}$, l'équation correspondante du système peut être remplacée par celle produite par collocation pour ce DL particulier :

$$\alpha u(x_j) = \alpha \bar{u}(x_j) \quad (2.3.12)$$

Cette collocation standard garde les équations du système creuse et en bande.

2.5 Exemples numériques

Cette méthode « FVMLPG », est évaluée à partir de quelques exemples de problèmes unidimensionnels de classe C^0 . Les solutions exactes suivantes sont considérées comme des « patch test »

I) $u = \text{constant}$

II) $u = x$ (2.5.1)

III) $u = x^2$.

Pour réaliser un patch test, chaque solution exacte est imposée comme conditions aux limites essentielles et le problème est ainsi analysé avec l'algorithme de FVMLPG. Pour passer le patch test, l'algorithme de FVMLPG doit reproduire la solution exacte à tous les noeuds intérieurs du modèle.

Paramètres de problème

On considère une barre uniforme de longueur l . La barre est modélisée en utilisant 5, 9, 17, et 33 noeuds équidistants. Le modèle à 17 noeuds est présenté sur la figure 2.5.1.

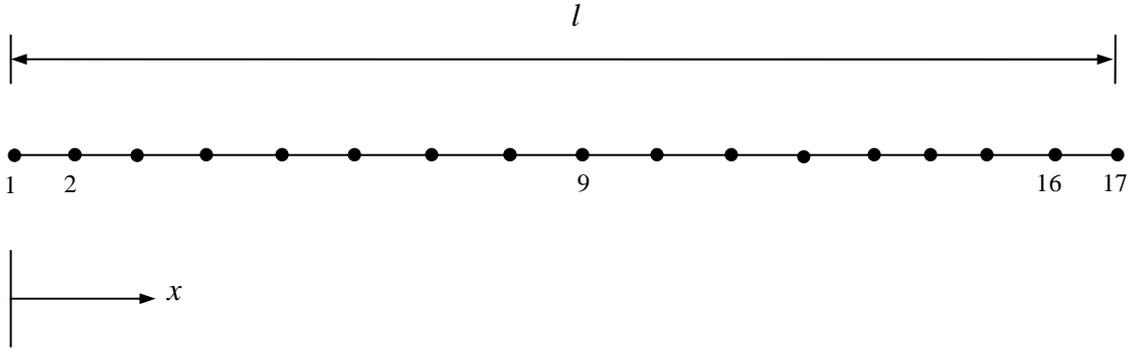


Figure 2.5.1 modèle à 17 noeuds d'une barre de longueur l

Une fonction linéaire de base $(1, x)$ devrait reproduire exactement les solutions linéaires (x^1) et d'ordre inférieure. De même, une base quadratique $(1, x, x^2)$ devrait reproduire les solutions quadratiques (x^2) et d'ordre inférieure. La forme faible (équation 2.1.16) nécessite que la fonction approchée, u , soit différentiable au moins une fois. La fonction linéaire de base est la fonction d'ordre le plus faible qui répond au critère de la méthode FVMLPG pour les problèmes C^0 .

$$-\frac{d}{dx} \left(b \frac{du}{dx} \right) + cu = f. \quad (2.5.2)$$

b et c sont des constantes. Les patches test sont réalisés pour différentes valeurs choisies de ces constantes.

- I. Patch test 1 : $b = 1$; $c = 0$; $u = \text{constante} = \beta_1$, où β_1 est une constante arbitraire. La substitution de ces valeurs dans l'équation (2.5.2) donne $f = 0$. Les conditions aux limites sont imposées aux noeuds 1 ($x = 0$) et N ($x = l$) d'un modèle à N noeuds comme suit:

$$\begin{aligned} u|_{x=0} &= \beta_1 \\ u|_{x=l} &= \beta_1 \end{aligned} \quad (2.5.3)$$

Ce patch test correspond à un déplacement d'un corps rigide non contraint (de grandeur β_1) de la barre. Les valeurs de (R_o/l) et (R_j/l) ont été choisies comme $(R_o/l) = 2\Delta x$ et $(R_j/l) = 1.0$. Pour les modèles à 5 et à 9 nœuds, l'algorithme calcule les solutions exactes pour les valeurs nodales fictives et les variables primaires et secondaires interpolées (figure 2.5.2). Pour le modèle à 17 nœuds, l'algorithme ne calcule pas la solution exacte pour les valeurs nodales fictives, mais les valeurs interpolées sont exactes. La valeur de (R_j/l) est réduite à $(R_j/l) = 8\Delta x$, et avec cette valeur dans le modèle à 17 nœuds, l'algorithme calcule les valeurs nodales fictives exactes. Des résultats semblables ont été obtenus pour le modèle à 33 nœuds. Ceci suggère que l'algorithme soit capable de reproduire des valeurs interpolées exactes, mais les valeurs nodales fictives exactes dépendent du paramètre (R_j/l) .

Les valeurs de (R_o/l) et (R_j/l) sont dorénavant choisies comme $(R_o/l) = 2\Delta x$ pour tous les modèles et $(R_j/l) = \text{longueur de barre}$ pour les modèles à 5 et à 9 nœuds et $(R_j/l) = 8\Delta x$ pour les modèles à 17 et 33 nœuds.

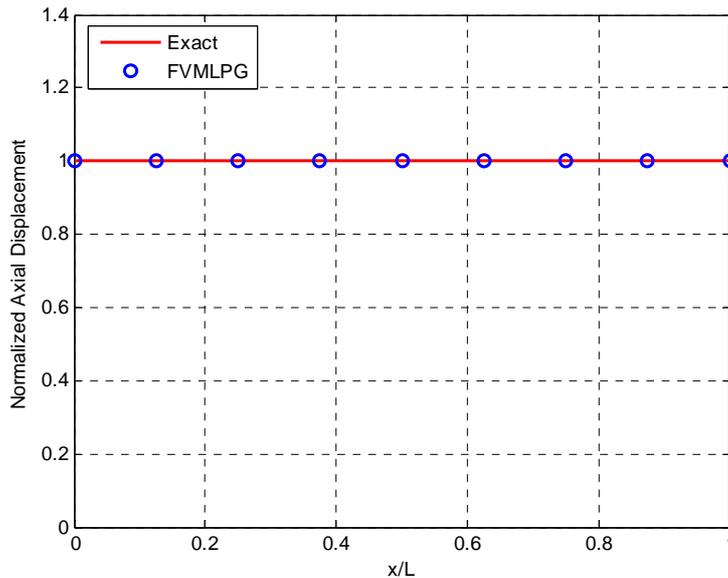


Figure 2.5.2 Déplacement pour $u = \text{constante}$.

II. Patch test 2: $b = 1$; $c = 0$; $u = x/l$

La substitution de b , c , et u dans l'équation (2.5.2) conduit au chargement $f = 0$. Les conditions aux limites sont imposées aux nœuds 1 ($x = 0$) et N ($x = l$) d'un modèle à N nœuds

$$\begin{aligned} u|_{x=0} &= 0 \\ u|_{x=l} &= 1 \end{aligned} \quad (2.5.4)$$

Les modèles à 5 -, 9 -, 17, et 33 nœuds donnent des solutions exactes avec ces conditions aux limites aux nœuds et à chaque point interne dans le domaine. Ce problème peut être traité comme dans le cas d'une barre uniforme avec une charge d'extrémité, \tilde{q} (figure 2.5.3), c'est-à-dire avec les conditions aux limites essentielles à l'une des extrémités et les conditions aux limites naturelles à l'autre extrémité.

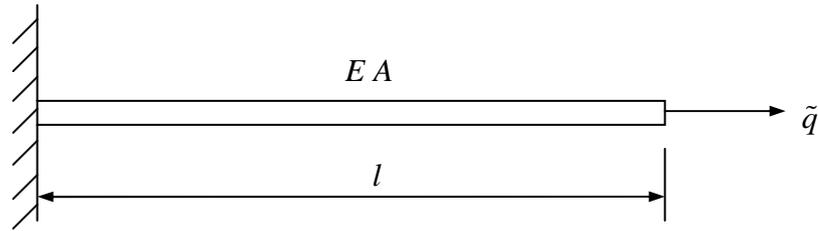


Figure 2.5.3 Barre uniforme de la longueur l soumise à une charge \tilde{q} à son extrémité

Les conditions aux limites imposées et le chargement appliqué sont

$$\begin{aligned} u|_{x=0} &= \tilde{u} = 0 \\ b \frac{du}{dx} \Big|_{x=l} &= \tilde{q} = 1, \quad b = EA \end{aligned} \quad (2.5.5)$$

et $f = 0$.

De nouveau, l'algorithme de FVMLPG avec chacun des quatre modèles a reproduit les solutions exactes pour les conditions limites mixtes (figure 2.5.4).

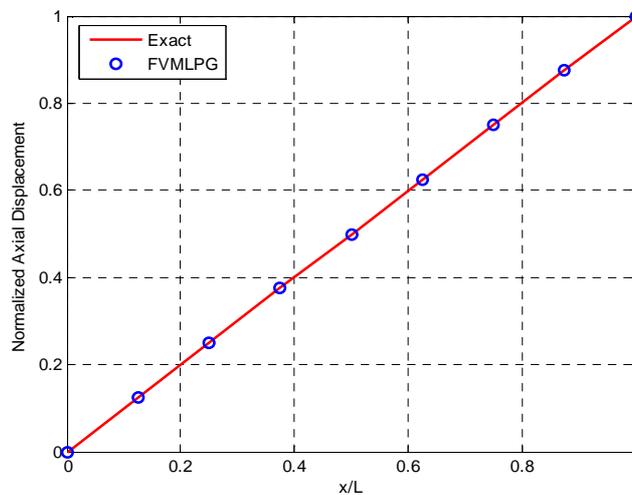


Figure 2.5.4 Déplacement axial pour $u = x$

III. Patch test 3: $b = 0$; $c = 1$

La solution exacte est

$$u = \left(\frac{x}{l} \right)^2. \quad (2.5.6)$$

La substitution de b , c , et de l'équation (2.5.6) dans l'équation (2.5.2) conduit au chargement $f = (x/l)^2$. Cette analyse peut être exécutée en utilisant trois ensembles différents de conditions aux limites.

i) Pour réaliser le patch test 2, les conditions aux limites à $x = 0$ et $x = l$ sont :

$$\begin{aligned} u|_{x=0} &= 0 \\ u|_{x=l} &= 1 \end{aligned} \quad (2.5.7)$$

ii) Alternativement, les conditions aux limites mixtes sont :

$$\begin{aligned} u|_{x=0} &= 0 \\ b \frac{du}{dx} \Big|_{x=l} &= 2b \frac{x}{l} = 0 \end{aligned} \quad (2.5.8)$$

iii) Les conditions aux limites mixtes sont imposées comme suit :

$$\begin{aligned} b \frac{du}{dx} \Big|_{x=0} &= 0 \\ u|_{x=l} &= 1 \end{aligned} \quad (2.5.9)$$

Comme prévu, l'analyse de FVMLPG a reproduit les solutions exactes pour les trois cas et pour tous les noeuds des quatre modèles considérés (figure 2.5.5).

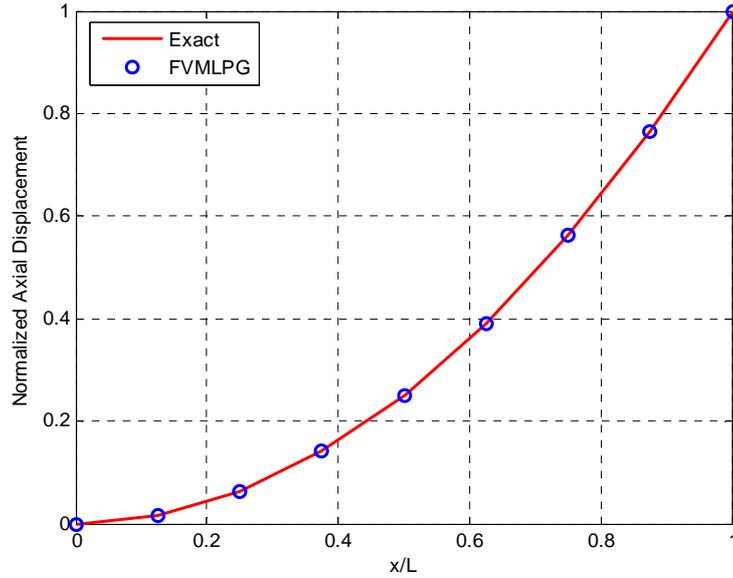


Figure 2.5.5 Déplacement axial pour $u = x^2$

2.6 Remarques et conclusion

Dans ce chapitre la méthode de FVMLPG appliquée aux problèmes unidimensionnels (1-D) est présentée. Sous la forme faible locale (LWF) de l'équation d'équilibre, une interpolation des moindres carrés mobiles (MLS) a été utilisée pour former les approximations à la solution connue sous le nom de fonctions d'essai. Les fonctions test, également nécessaires pour le LWF ont été choisies dans un espace différent de celui des fonctions d'essai. Ce choix des fonctions test a conduit aux matrices de rigidité antisymétriques. Les conditions aux limites essentielles ont été imposées par une méthode de collocation. La méthode de FVMLPG a été évaluée à partir de plusieurs tests. Les résultats obtenus ont montré un bon comportement de la méthode.

Chapitre 3

FVMLPG appliquée aux problèmes C^0 élastostatiques bidimensionnels

Ce chapitre présente une description de la méthode de « finite volume meshless local Petrov Galerkin » (FVMLPG) appliquée aux problèmes élastostatiques bidimensionnels. D'abord, une forme faible locale est développée par la méthode des résidus pondérés. Une interpolation des moindres carrés mobiles est utilisée pour construire les approximations connues sous le nom de fonctions d'essai. Des fonctions tests sont choisies dans un espace différent de celui des fonctions d'essai, relative à la méthode de Petrov Galerkin. Un système d'équations algébriques est obtenu en substituant les fonctions d'essai et test dans la forme faible locale. La méthode est évaluée ensuite en l'appliquant sur plusieurs problèmes aux limites. Une étude comparative entre cette méthode et la méthode des éléments finis est effectuée. L'efficacité informatique ainsi que l'exactitude entre la formulation combinée (FVMLPG) et la méthode des éléments finis est présentée.

3.1 Forme faible symétrique locale (LSWF) de problèmes élastostatiques unidimensionnels

Soit un corps Ω de frontière Γ en 2-D, de coordonne $\mathbf{x}_I = \{x_I^1, x_I^2\}$ avec des conditions aux limites essentielles et naturelles, soumis à une force de volume (figure 3.1.1). Pour un corps élastique linéaire subissant des déformations infinitésimales, l'équation d'équilibre peut être obtenue par l'équilibre linéaire des quantités de mouvement

$$\sigma_{ij,j} + f_i = 0 \quad (3.1.1)$$

Conditions aux limites

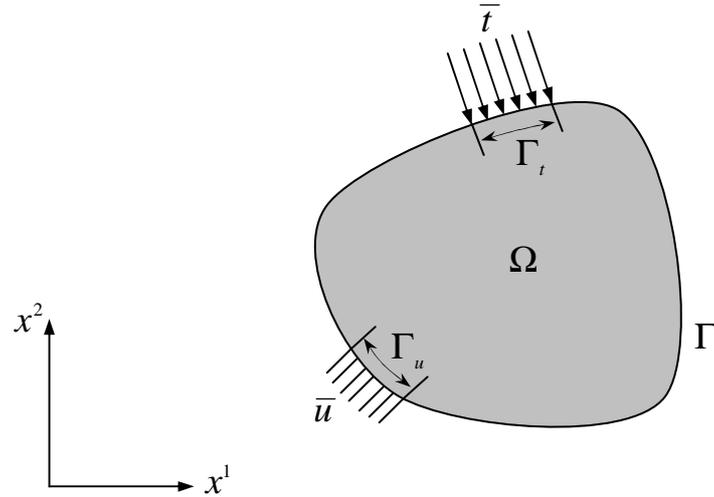


Figure 3.1.1 Corps arbitraire avec des conditions aux limites en déplacement et en force (traction)

$$u_i = \bar{u}_i \quad \text{sur} \quad \Gamma_u \quad (3.1.2a)$$

$$t_i = \sigma_{ij} n_j = \bar{t}_i \quad \text{sur} \quad \Gamma_t \quad (3.1.2b)$$

σ_{ij} est le tenseur de contrainte, f_i est la force de volume, \bar{u}_i sont les déplacements imposés sur la frontière de déplacement Γ_u et \bar{t}_i sont les forces de tractions imposées sur la frontière de traction Γ_t , n_j est le vecteur unitaire normal au bord Γ .

L'équation constitutive du comportement linéaire élastique s'écrit :

$$\sigma_{ij} = E_{ijkl} u_{(k,l)} \quad (3.1.3)$$

où

$$u_{(k,l)} = \frac{1}{2} (u_{k,l} + u_{l,k}). \quad (3.1.4)$$

La méthode des éléments finis de Galerkin (FEGM) est basée sur la forme faible symétrique globale (GSWF)

$$\int_{\Omega} \left[E_{ijkl} v_{(i,j)} u_{(k,l)} - b_i v_i \right] d\Omega - \int_{\Gamma_t} \bar{t}_i v_i d\Gamma = 0 \quad (3.1.5)$$

Pour toutes les fonctions d'essai u_i et de test v_i , continues dans chaque élément, comme aux frontières d'inter élément, et qui a priori, respectent la condition aux limites essentielles sur Γ_u (en particulier $v_i = 0$); dans ces conditions, l'équation (3.1.5) peut s'écrire sous la forme suivante :

$$\sum_m \left\{ \int_{\Omega_m} \left[E_{ijkl} v_{(k,l)} - b_i v_i \right] d\Omega - \int_{\Gamma_m} \bar{t}_i v_i d\Gamma \right\} = 0 \quad (3.1.6)$$

La sommation se prolonge au-dessus de tous les éléments finis Ω_m , et de leurs frontières Γ_m . Dans la méthode des éléments finis classiques, les fonctions d'essai et de test dans chaque élément s'expriment par :

$$u_i(x) = \sum_{\alpha} \phi^{\alpha}(x) u_i^{\alpha}, \quad (3.1.7)$$

$$v_i(x) = \sum_{\alpha} \phi^{\alpha}(x) v_i^{\alpha} \quad (3.1.8)$$

Dans les équations (3.1.7) et (3.1.8), la sommation s'effectue sur tous les noeuds α appartenant aux éléments. Dans ce cas, les valeurs nodales des fonctions d'essai et test sont u_i^{α} et v_i^{α} . Ainsi les fonctions $\phi_i^{\alpha}(x)$ des équations en (3.1.7) et (3.1.8) peuvent être désignées comme des fonctions de forme et chacune, a une valeur unitaire au noeud α . Les équations (3.1.6)-(3.1.8) peuvent être utilisées non seulement pour les éléments mais aussi pour l'ensemble du maillage. C'est également valable pour l'intégration des formes faibles de l'équation (3.1.6). Si l'élément introduit est un triangle dans le problème d'élasticité (3.1.6), les fonctions de forme représentant les fonctions d'essai et test sont linéaires.

Par ailleurs, les fonctions de forme globales au noeud pour le problème d'élasticité en 2-D, sont telle que leur valeur aux noeuds sont égales à l'unité et à zéro à tous les noeuds voisins et aussi qu'à la frontière Ω_r^J entourant le noeud J en question (figure 3.1.2). La forme faible globale (3.1.6) peut être alternativement écrite, en termes de fonctions de forme globale aux noeuds :

$$\sum_{IJ} \int_{\Omega_{te}^I} (E_{ijkl} \phi_{,j}^I v_i^I \phi_{,l}^J u_k^J - b_i \phi^I v_i^I) d\Omega - \int_{\Gamma_I} \bar{t}_i \phi^I v_i^I d\Gamma = 0 \quad (3.1.9)$$

où ϕ^J sont les fonctions de forme globales aux nœuds centrées au nœud J (prend des valeurs unitaires au nœud J) et sont non nulles seulement dans Ω_{tr}^J . ϕ^I s'exprime de la même façon au nœud I et sur l'élément Ω_{te}^I , (figure 3.1.2).

u_k^J et v_i^I sont respectivement les valeurs nodales de la fonction d'essai u_k et de la fonction test v_i .

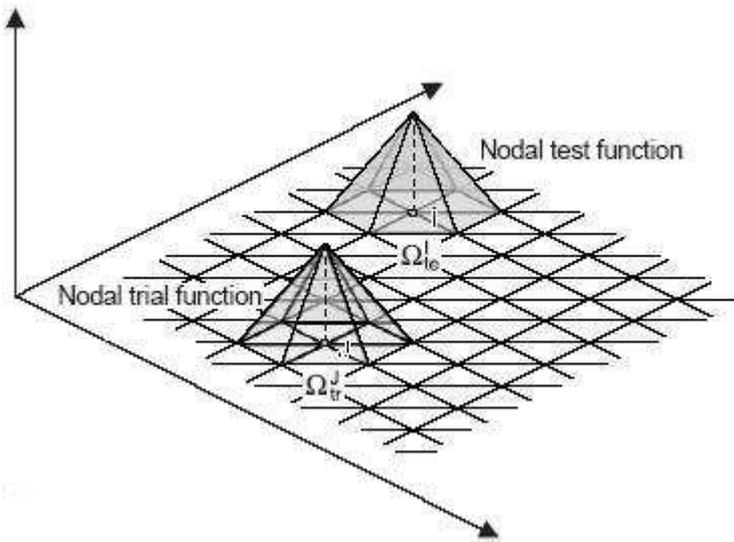


Figure 3.1.2 (a) Représentation des fonctions d'essai et test centrées au nœud I et J

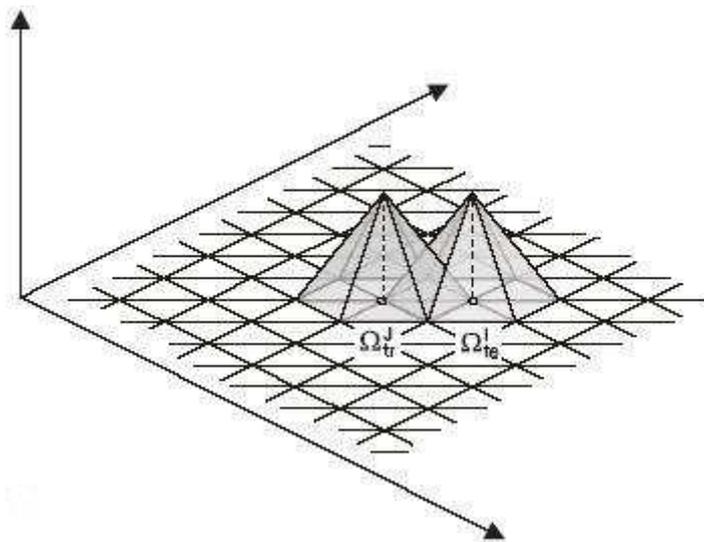


Figure 3.1.2 (b) Intersection des fonctions d'essai et test

La zone d'intersection permet immédiatement à partir des équations (3.1.9) et la figure 3.1.1, d'identifier la matrice globale de rigidité K_{IJ} tels que :

$$K_{IJ} = 0 \quad (3.1.10)$$

Pour obtenir une zone d'intersection, il faut que le noeud J soit un nœud voisin immédiat du nœud I (c'est-à-dire que le noeud J est le sommet de Ω_{te}^I entourant le nœud I). C'est la raison qui conduit à la matrice bande symétrique de K_{IJ} dans la méthode FEGM. Il est également important de noter, que la matrice de rigidité K_{IJ} est différente de zéro du fait que le résultat d'intégration est différent de zéro seulement au-dessus de l'intersection du domaine Ω_{tr}^J et Ω_{te}^I des nœuds environnants J et I , respectivement, comme le montre la figure 3.1.2. Cependant, dans la méthode FEGM, la forme faible est intégrée au-dessus de tous les éléments finis, et les 'contributions nodales' respectives sont assemblées, pour former la matrice globale de rigidité.

Dans toutes les méthodes numériques (éléments finis (MEF), éléments de frontières (BEM), différences finies (MDF), volumes finis (MVF)) les sous domaines sont considérés, comme des éléments dans les approches de type (MEF et BEM) et des mailles dans les méthodes DF et VF. La solution du problème dans les sous domaines est une manière d'approcher la solution dans la totalité du domaine. Dans la méthode de « finite volume meshless local Petrov Galerkin » FVMLPG, le sous domaine Ω_s est défini seulement en quelques points dans une zone surfacique ou dans un volume avec une forme quelconque (figure 3.1.3). Il y a plusieurs méthodes d'approximation du champ de variables. L'une de ces méthodes est la méthode des moindres carrés mobiles (MLS). Dans cette méthode, il est possible d'interpoler des données aléatoires avec une exactitude raisonnable [106]. Si le champ de variable est la fonction $u(x)$ dans l'approximation MLS, la distribution u dans Ω_s peut être approchée à un certain nombre de points locaux dispersés x_I , ($I = 1, 2, 3, \dots, n$) comme

$$u(x) = p^T(x) a(x) \quad \forall x \in \Omega_s \quad (3.1.11)$$

$p^T(x) = [p_1(x), p_2(x), \dots, p_m(x)]_{1 \times m}$ est une base de monôme d'ordre m , en 2-D

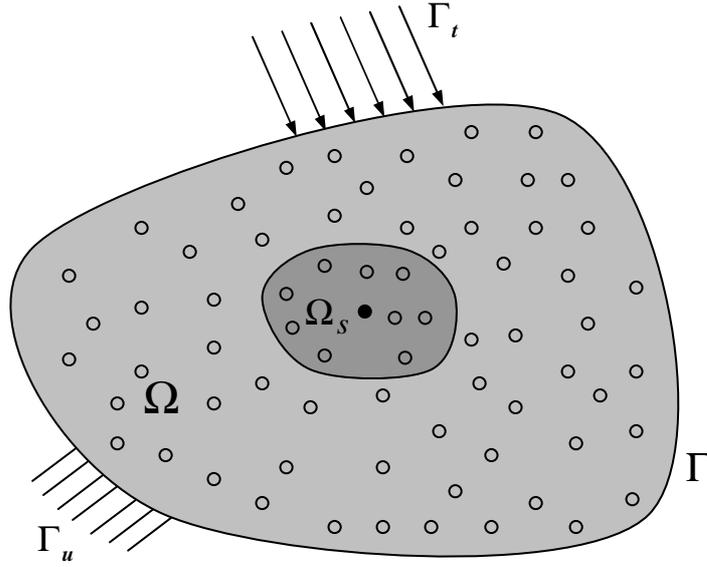


Figure 3.1.3 Sous domaine arbitraire autour d'un point

$$\mathbf{p}^T(\mathbf{x}) = [1, x^1, x^2] \quad \text{base linéaire pour } m = 3 \quad (3.1.12a)$$

$$\mathbf{p}^T(\mathbf{x}) = [1, x^1, x^2, (x^1)^2, x^1 x^2, (x^2)^2] \quad \text{base quadratique } m = 6 \quad (3.1.12b)$$

et $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \dots, a_m(\mathbf{x})]_{1 \times m}^T$ est un vecteur contenant les coefficients qui sont aussi des fonctions de coordonnées cartésiennes globales $[x^1, x^2]^T$, suivant la base de monômes. Ces coefficients sont déterminés en minimisant la norme pondérée discrète L_2 aux points nodaux :

$$\begin{aligned} J(\mathbf{x}) &= \sum_{I=1}^m w_I(\mathbf{x}) [\mathbf{p}^T(\mathbf{x}_I) \mathbf{a}(\mathbf{x}) - \hat{u}^I]^2 \\ &= [\mathbf{P} \cdot \mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}]^T \mathbf{W} [\mathbf{P} \cdot \mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}] \end{aligned} \quad (3.1.13)$$

où $w_I(\mathbf{x})$ est la fonction de poids associée au nœud I , avec $w_I(\mathbf{x}) > 0$. Pour tous \mathbf{x} appartenant à $w_I(\mathbf{x})$ où \mathbf{x}_I est la valeur de \mathbf{x} au nœud I et n est le nombre de nœuds dans Ω_s , les matrices \mathbf{P} et \mathbf{W} sont définis telle que $w_I(\mathbf{x}) > 0$,

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_n) \end{bmatrix}_{n \times m} \quad (3.1.14)$$

$$\mathbf{W} = \begin{bmatrix} w_1(\mathbf{x}) & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & w_n(\mathbf{x}) \end{bmatrix}_{n \times n} \quad (3.1.15)$$

et

$$\hat{\mathbf{u}}^T = [\hat{u}_1 \quad \hat{u}_2 \quad \cdots \quad \hat{u}_n] \quad (3.1.16)$$

Il convient de noter que \hat{u}_i , $i=1,2,\dots,n$ dans l'équations (3.1.13) et (3.1.16) sont les valeurs nodales fictives. La stationnarité de J dans l'équation (3.1.13) en ce qui concerne $a(\mathbf{x})$ conduit à la relation linéaire suivante entre $a(\mathbf{x})$ et \hat{u}_i .

$$\mathbf{A}(\mathbf{x})a(\mathbf{x}) = \mathbf{B}(\mathbf{x})\hat{\mathbf{u}} \quad (3.1.17)$$

les matrices $\mathbf{A}(\mathbf{x})$ et $\mathbf{B}(\mathbf{x})$ sont définies par :

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T \mathbf{W} \mathbf{P} = \mathbf{B}(\mathbf{x})\mathbf{P} = \sum_{i=1}^n w_i(\mathbf{x})\mathbf{p}(\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i) \quad \forall \mathbf{x} \in \Omega_s \quad (3.1.18)$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{P}^T \mathbf{W} = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2), \dots, w_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)] \quad \forall \mathbf{x} \in \Omega_s \quad (3.1.19)$$

L'approximation MLS est bien définie seulement quand la matrice \mathbf{A} dans l'équation (3.1.17) est non singulière.

La fonction de forme peut être obtenue de la façon suivante :

$$u(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\hat{\mathbf{u}} \equiv \Phi^T(\mathbf{x})\hat{\mathbf{u}} \quad \forall \mathbf{x} \in \Omega_s \quad (3.1.20)$$

La fonction de poids dans l'équation (3.1.13) a un support circulaire et compact définissant ainsi le domaine d'influence du nœud I . Dans ce travail, on considère que la fonction de poids est une spline de quatrième ordre :

$$w_I(\mathbf{x}) = \begin{cases} 1 - 6\left(\frac{d_I}{r_I}\right)^2 + 8\left(\frac{d_I}{r_I}\right)^3 - 3\left(\frac{d_I}{r_I}\right)^4 & 0 \leq d_I \leq r_I \\ 0 & d_I \geq r_I \end{cases} \quad (3.1.21)$$

$d_I = |\mathbf{x} - \mathbf{x}_I|$ est la distance entre le noeud \mathbf{x}_I et le noeud \mathbf{x} ; et r_I est le rayon du domaine d'influence de la fonction de poids w_I qui détermine le support du noeud \mathbf{x}_I . La fonction de poids de spline (équation (3.1.21)) est de classe C^1 continue au-dessus du domaine entier Ω et les fonctions de forme $\Phi(\mathbf{x})$ ainsi que les fonctions d'essai sont également de classe C^1 .

3.2 Approche finite volume meshless local Petrov Galerkin (FVMLPG)

La discrétisation par volumes finis (FV) est basée sur la forme intégrale de l'équation au-dessus du volume ou du sous domaine Ω_s . En d'autres termes, la discrétisation de FV introduit la forme intégrale de l'équation (3.1.1) au-dessus du sous domaine Ω_s autour du noeud I comme en [107, 108, 109]

$$\int_{\Omega_s} (\sigma_{ij,j} + f_i) d\Omega = 0 \quad (3.2.1)$$

Si le théorème de divergence est appliqué à la première limite intégrale alors

$$\int_{\partial\Omega_s} \sigma_{ij} n_j d\Gamma - \int_{\Omega_s} f_i d\Omega = 0. \quad (3.2.2)$$

n_j est la normale extérieure à la frontière locale $\partial\Omega_s$. La nature conservatrice de la méthode des volumes finis (FVM) est établie comme un flux de contrainte σ_{ij} intégré au-dessus de la frontière locale $\partial\Omega_s$.

En introduisant les conditions aux limites de traction de l'équation (3.1.2) dans l'équation (3.2.2), on obtient :

$$\int_{L_s} t_i d\Gamma + \int_{\Gamma_{su}} t_i d\Gamma + \int_{\Gamma_{st}} \bar{t}_i d\Gamma - \int_{\Omega_s} f_i d\Omega = 0. \quad (3.2.3)$$

On distingue dans l'équation (3.2.3), deux frontières locales (figure 3.2.1) :

- La frontière intérieure du domaine global L_s
- La frontière qui a le côté commun avec la frontière du domaine global.

Soit Γ_{su} la partie commune entre Γ_s et Γ_u c'est-à-dire $\Gamma_{su} = \Gamma_s \cap \Gamma_u$.

Soit Γ_{st} la partie commune entre Γ_s et Γ_t c'est-à-dire $\Gamma_{st} = \Gamma_s \cap \Gamma_t$.

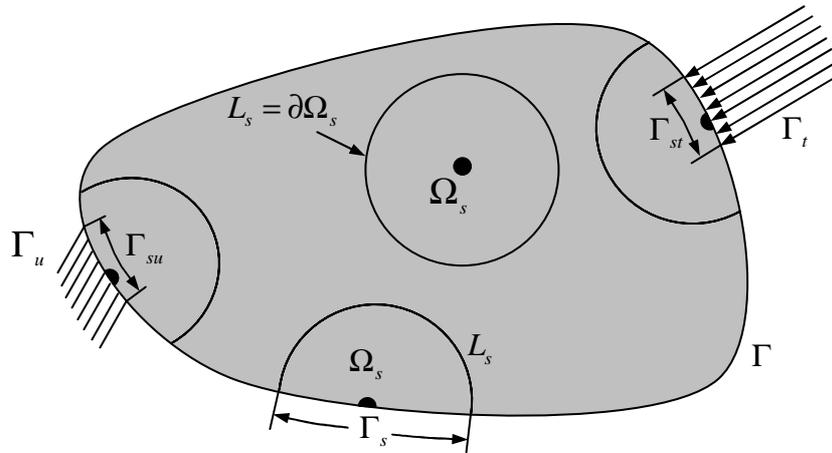


Figure 3.2.1 Différents types de frontière de domaine local

L'équation (3.2.3) représente une signification physique dans la loi d'équilibre du sous domaine local Ω_s comme FVM conventionnel avec les conditions de frontière de traction. Par conséquent cette approche s'appelle les formulations dites de « finite volume meshless local Petrov Galerkin » (FVMLPG) de l'équation d'équilibre dans le problème unidimensionnel de l'élasticité linéaire, de l'équation (3.1.1).

Les relations constitutives du solide homogène élastique linéaire isotrope sont introduites dans l'équation (3.1.2b)

$$t_i = \sigma_{ij} n_j = E_{ijkl} \epsilon_{kl} n_j \quad (3.2.4)$$

Les relations de contrainte déplacement pour le problème linéaire d'élastostatiques sont

$$\boldsymbol{\varepsilon}_{kl} = \frac{1}{2} (u_{k,l} + u_{l,k}). \quad (3.2.5)$$

La contrainte peut être interpolée avec la même fonction de forme dans l'équation (3.1.12)

$$\boldsymbol{\varepsilon}_{kl}(\mathbf{x}) = \sum_{K=1}^n \Phi^{(K)}(\mathbf{x}) \boldsymbol{\varepsilon}_{kl}^{(K)}. \quad (3.2.6)$$

L'équation (3.2.3) est discrétisée par substitution des équations (3.2.4) et (3.2.6) dans l'équation (3.2.7)

$$\begin{aligned} & - \sum_{K=1}^n \left[\int_{L_s} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] \boldsymbol{\varepsilon}_{kl}^{(K)} - \sum_{K=1}^n \left[\int_{\Gamma_{su}} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] \boldsymbol{\varepsilon}_{kl}^{(K)} \\ & = \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega. \end{aligned} \quad (3.2.7)$$

L'avantage de cette équation (3.2.7) est qu'il n'y a aucune dérivée de fonction de forme. L'approximation sans maillage n'est pas efficace en calculant de telles dérivées partout dans le domaine. Par conséquent c'est l'un des avantages résultant de ce travail en comparaison avec la méthode de MLPG traditionnelle. En clair, dans la méthode MLPG le déplacement est approché directement donc la dérivée de la fonction de forme est nécessaire dans la forme locale discrétisée. En outre dans l'approche FVMLPG, la déformation (qui est la variable secondaire du champ), est approchée indépendamment du déplacement. De ce fait on réduit l'ordre des fonctions d'approximation. Le domaine d'influence nodal est par conséquent de taille réduite. Ce qui permet des calculs rapides des fonctions de forme.

D'autre part, l'équation (3.2.7) montre que le nombre d'équations est inférieur au nombre de variables indépendantes de déformations. Par exemple dans un problème en 2-D, on a trois variables nodales en déformation, mais seulement deux variables nodales de déplacement. Il est possible de réduire le nombre de variables en transformant les variables de déformation de nouveau en déplacements sans aucune modification de l'équation (3.2.7). Par conséquent en appliquant l'équation (3.2.6) seulement à chaque point nodal \mathbf{x}_I au lieu du domaine entier, les variables nodales de déformations sont exprimées en termes de variables nodales de déplacement

$$\boldsymbol{\varepsilon}_{kl}(\mathbf{x}_I) = \boldsymbol{\varepsilon}_{kl}^I = \frac{1}{2} (u_{k,I} + u_{l,k}) \quad I = 1, 2, \dots, n. \quad (3.2.8)$$

soit l'approximation en déplacements suivante

$$u(x) = \sum_{J=1}^n \phi^J(x) \hat{u}^J \quad (3.2.9)$$

Les deux ensembles de variables nodales peuvent être transformés par une matrice algébrique linéaire

$$\varepsilon_{kl}^I = H_{klm}^{IJ} u_m^J \quad (3.2.10)$$

la matrice de transformation H est diagonalisée. Après substitution de l'équation (3.2.10) dans l'équation (3.2.7) on obtient la relation force – déplacement suivante :

$$K u = f \quad (3.2.11)$$

$$K_{IJ} = \left[\int_{\Gamma_{L_s}} \Phi^{(I)}(x) E_{ijkl} n_j d\Gamma - \int_{\Gamma_{su}} \Phi^{(I)}(x) E_{ijkl} n_j d\Gamma \right] H_{klm}^{(I)(J)} \quad (3.2.12)$$

K_{IJ} est la matrice de rigidité, u est le vecteur déplacement, et

$$f_I = \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega \quad (3.2.13)$$

est le vecteur de force.

3.3 Imposition des conditions aux limites

Avec l'approximation sans maillage, les conditions aux limites essentielles ne peuvent pas être imposées directement. Dans notre approche FVMLPG, les paramètres de pénalité ainsi que multiplicateurs de Lagrange n'apparaissent pas dans les équations parce que les conditions aux limites essentielles peuvent être imposées par la méthode de collocation. Pour un point nodal $x^{(I)}$, si son $i^{\text{ème}}$ degré de liberté (DL) en déplacement appartient à la frontière de déplacement, c'est-à-

dire $u_i^{(l)} \in \Gamma_{su}$, l'équation correspondante du système peut être remplacée par celle résultant de la collocation pour ce DL, comme

$$\alpha u_i(x^{(l)}) = \alpha \bar{u}_i(x^{(l)}). \quad (3.3.1)$$

3.4 Intégration numérique

L'intégration numérique est un ingrédient important d'une méthode sans maillage. Ce calcul nécessite l'évaluation des intégrales au-dessus de la frontière des sous domaines locaux. Par définition les forces de traction $\bar{t}_i = \sigma_{ij} n_j$, contiennent la normale n à la frontière. Compte tenu de la forme circulaire du domaine, le vecteur normal nécessite des fonctions de type trigonométriques. Des expressions polynomiales ne peuvent pas être prévues au-dessus de la frontière locale non couverte par tous les noeuds locaux. Par conséquent il n'est pas possible d'appliquer les arrangements numériques conventionnels de quadrature conçus pour des polynômes avec des fonctions trigonométriques. Cependant une quadrature avec 2 points de gauss évalue l'intégrale $\int_0^1 x^3 dx$ exactement. Dans ce cas, les expériences numériques donnent une erreur d'environ 0.07% quand une quadrature de 4 point de gauss est utilisée pour évaluer l'intégrale unidimensionnelle en 2D de type $\int_0^1 x^3 \sin 2\pi x dx$ dans laquelle seulement une fonction trigonométrique est impliquée.

Il y a deux manières de contrôler les erreurs numériques ; soit on augmente l'ordre d'approximation de quadrature de gauss, soit on subdivise le domaine d'intégration en petits segments pour une meilleure exactitude. La deuxième méthode est beaucoup plus efficace que la première. Pour les noeuds situés sur la frontière globale, on considère un sous domaine circulaire local centré au noeud I , $x^{(l)}$, avec un rayon $r_0^{(l)}$. En traçant une droite du noeud I à ses noeuds voisins $J \{x^{(j)}\}$, ($J = 1, 2, \dots, m$), un point peut être obtenu à l'intersection entre cette droite et le cercle local ; ce point est désigné par $\{y^{(j)}\}$, ($J = 1, 2, \dots, m$). Un sous-ensemble de ces points d'intersection est utilisé pour diviser le domaine d'intégration, c'est-à-dire le cercle local. Les points d'intersection entre les frontières locales et globales sont automatiquement inclus dans $y^{(j)}$.

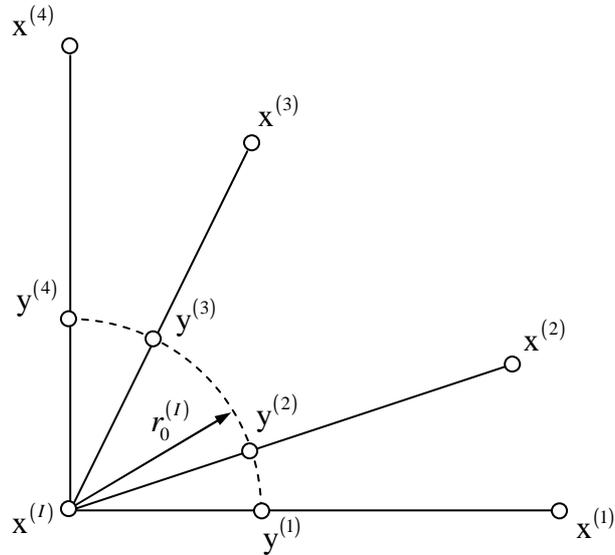


Figure 3.4.1 Intégration numérique sur la frontière locale

La forme du sous domaine est circulaire et sa taille est liée à la distance nodale. Normalement, ce domaine devrait être plus grand afin d'assurer un nombre de points suffisants pour soutenir les noeuds sur la frontière globale. La taille du sous domaine affecte l'exactitude de la solution et l'efficacité de la méthode.

3.5 Routine d'exécution

L'exécution de la méthode peut être effectuée selon la routine suivante [110] :

1. Choix d'un nombre fini de noeuds dans le domaine Ω et sur la frontière Γ du domaine physique donné ; et s'assurer que les fonctions de poids et de base sont correctement définies dans l'approximation de MLS.
2. Détermination du sous domaine local Ω_s et sa frontière locale correspondante $\partial\Omega_s$ pour chaque noeud.
3. Itération au-dessus de tout le noeud situé à l'intérieur du domaine global et à la frontière globale Γ .
 - Détermination des points de quadrature de Gauss x_Q dans Ω_s et au dessus $\partial\Omega_s$.
 - Introduction d'une boucle sur les points finis de quadrature x_Q dans le sous domaine Ω_s et sur la frontière locale $\partial\Omega_s$

- (a) détermination des noeuds x_i situés dans le domaine de définition de l'approximation de MLS pour la fonction d'essai au point x_Q , c.-à-d., pour les noeuds avec $w_i(x_Q) > 0$;
 - (b) pour les noeuds situés dans le domaine de définition de l'approximation de MLS et de la fonction d'essai au point x_Q ; on calcule des fonctions $\phi_i(x_Q)$;
 - (c) évaluation des intégrales numériques ;
 - (d) assemblage des contributions au système linéaire pour tous les noeuds dans K, f ;
- Fin de la boucle sur le point de quadrature
4. Fin de la boucle sur le noeud.
 5. Résolution du système linéaire pour les valeurs nodales fictives.
 6. Calcul de la valeur de la variable inconnue en ces points témoins.

3.6 Convergence et Stabilité

La convergence et la stabilité des résultats numériques sont vérifiées vis-à-vis de l'augmentation du nombre de noeuds et de l'irrégularité dans leur distribution. ¶ Pour la convergence, la solution numérique obtenue par cette méthode doit approcher la solution exacte quand l'espacement nodal approche zéro. Pour cela, on utilise la norme $L2$ ainsi que la norme d'erreurs d'énergie dans quelques exemples. La norme d'erreur en ¶ déplacement $L2$ $\|\mathbf{u}\|$ et la norme d'énergie $\|\boldsymbol{\sigma}\|$ ¶ sont définies comme suit : ¶

$$\|\mathbf{u}\| = \left(\int_{\Omega} \mathbf{u}^T \mathbf{u} d\Omega \right)^{\frac{1}{2}} \quad (3.6.1)$$

$$\|\boldsymbol{\sigma}\| = \left(\frac{1}{2} \int_{\Omega} \boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma} d\Omega \right)^{\frac{1}{2}} \quad (3.6.2)$$

La norme d'erreur en ¶ déplacement et en énergie sont définies par :

$$r_u = \frac{\|\mathbf{u}^{num} - \mathbf{u}^{exact}\|}{\|\mathbf{u}^{exact}\|} \quad (3.6.3)$$

$$r_{\sigma} = \frac{\|\sigma^{num} - \sigma^{exact}\|}{\|\sigma^{exact}\|} \quad (3.6.4)$$

La distribution non-uniforme ou arbitraire des nœuds est une condition essentielle pour la robustesse de la méthode sans maillage. ¶Deux éléments sont importants en état de stabilité. ¶Le premier élément est la stabilité d'interpolation qui garantit la stabilité de la fonction de forme vis-à-vis des petites perturbations de la position des nœuds dans chaque sous-domaine. Dans ce cas la matrice de moment, $\mathbf{A}(\mathbf{x})$ dans l'équation ¶(3.1.18), doit être conditionnée dans la distribution arbitraire des nœuds. ¶Cette matrice est bien-conditionnée et inversible quand un nombre approprié de nœuds existent dans le sous-domaine. Ce nombre est déterminé par expériences numériques. ¶Le deuxième élément est la stabilité de la solution qui est induite par la fonction de forme ainsi que par la formulation de la méthode. ¶La solution numérique ne doit pas avoir des oscillations numériques ou physiques. ¶¶Afin d'étudier la stabilité, le processus de calcul doit être stable vis-à-vis des petites perturbations de la position des nœuds dans le sous-domaine. En utilisant un ensemble de nœuds distribués arbitrairement, les résultats ne doivent en en cas changer considérablement par rapport aux résultats obtenus avec une distribution régulière des nœuds.¶

3.7 Exemples numériques

Cette méthode est évaluée par les trois exemples ci-dessous.

Exemple 1 Poutre encastree soumise à une charge transversale à son d'extrémité libre

On considère la poutre console représentée par la figure 3.7.1. Une charge transversale P est appliquée à l'extrémité libre.

L : longueur,

c : demi hauteur,

E : module d'élasticité

ν : coefficient de Poisson

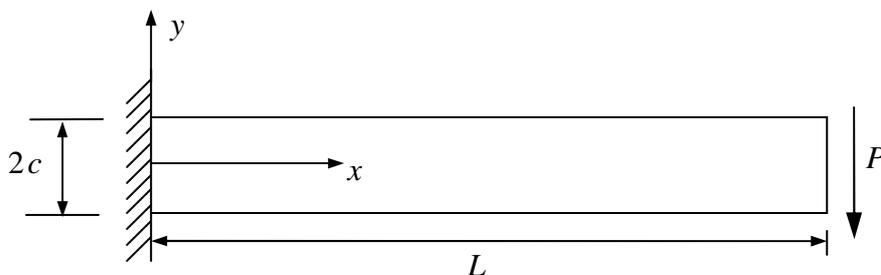


Figure 3.6.1 Poutre console sous une charge transversale

Cet exemple est résolu en considérant l'état de contraintes planes

$$L = 24, c = 2, E = 1, \nu = 0.25, P = 1.$$

Avec les conditions aux limites en déplacement suivantes :

$$u_x(0,0) = u_y(0,0) = 0 \text{ et } u_x(0,c) = u_x(0,-c) = 0$$

La solution exacte pour ce problème est donnée dans Timoshenko et Goodier [111] par :

$$u_x = -\frac{Py}{6EI} \left[3x(2L-x) + (2+\nu)(y^2 - c^2) \right] \quad (3.7.1a)$$

$$u_y = \frac{P}{6EI} \left[x^2(3L-x) + 3\nu(L-x)y^2 + (4+5\nu)c^2x \right] \quad (3.7.1b)$$

$I = 2tc^3/3$ est le moment d'inertie de la section et t est l'épaisseur considérée constante de long de la poutre. Dans cet exemple $t = 1$. Les contraintes correspondantes sont

$$\sigma_x = -\frac{P}{I}(L-x)y \quad (3.7.2a)$$

$$\sigma_y = 0 \quad (3.7.2b)$$

$$\sigma_{xy} = -\frac{P}{2I}(y^2 - c^2) \quad (3.7.2c)$$

Afin de lancer la procédure de calcul suivant notre méthode « FVMLPG » au problème de la console, on doit d'abord considérer quelques nœuds choisis par des tests numériques. On considère des configurations nodales uniformes et régulières avec des distances nodales d de 2.0, 1.0 et 0.5. Le nombre adimensionnel d est la distance entre deux nœuds voisins suivant les directions x ou y . Il est déterminé par division de la longueur de poutre par le nombre de nœuds dans la direction de la longueur. Les nombres de nœuds sont respectivement 39, 125 et 441. Un maillage d'éléments finis est également construit autour des configurations nodales à partir de l'élément incompatible à 4 nœuds défini dans ABAQUS. Le maillage EF est composé de 12×2 ,

24×4 et 48×8 nœuds suivant les directions x et y . Ce type d'élément donne les meilleurs résultats pour ce problème comparativement à d'autres éléments dans ABAQUS. La figure 3.7.2 montre la configuration nodale de FVMLPG ainsi que le maillage par EF avec la distance nodale $d = 1.0$.

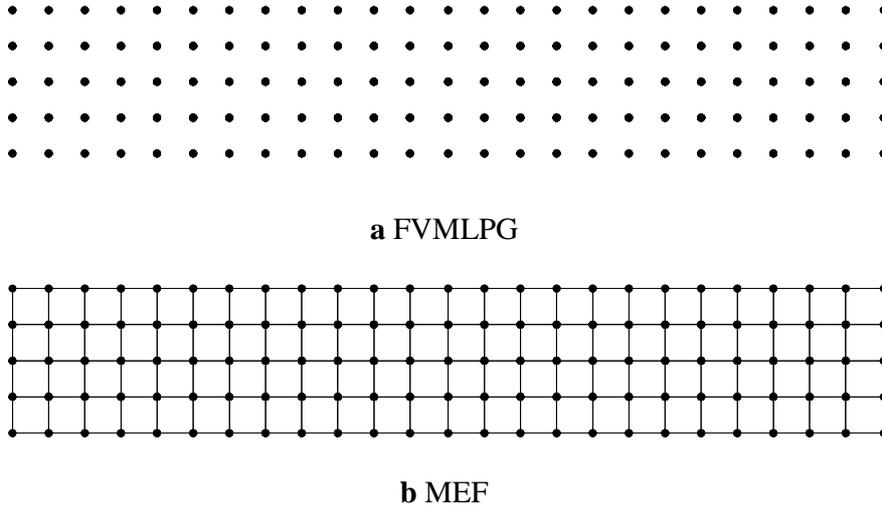


Figure 3.7.2 a Configuration nodale de FVMLPG avec la distance nodale $d = 1.0$

b Maillage EF avec la distance nodale $d = 1.0$

La figure 3.7.3 montre le déplacement vertical de la poutre avec la distance nodale de 0.5 avec les conditions de frontière essentielles imposées uniquement en trois points de la section d'encastrement.

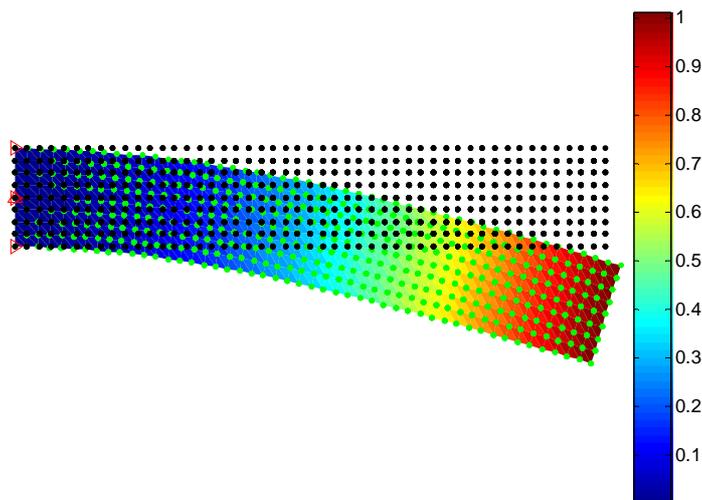


Figure 3.7.3 Déplacement vertical normalisé d'une poutre encastree sous une charge transversale
(441 noeuds)

Les résultats numériques de la déformation de cisaillement sont montrés sur la figure 3.7.4. Ces résultats sont en bon accord avec ceux fournis par la solution analytique. L'erreur relative maximale du déplacement obtenu par notre approche est de 0.0001389% . L'erreur de calcul par éléments finis est de 0.07661% bien que la distance nodale est la même dans les deux configurations. La figure 3.7.4 montre bien l'exactitude de cette méthode par rapport aux calculs fournis par ABAQUS.

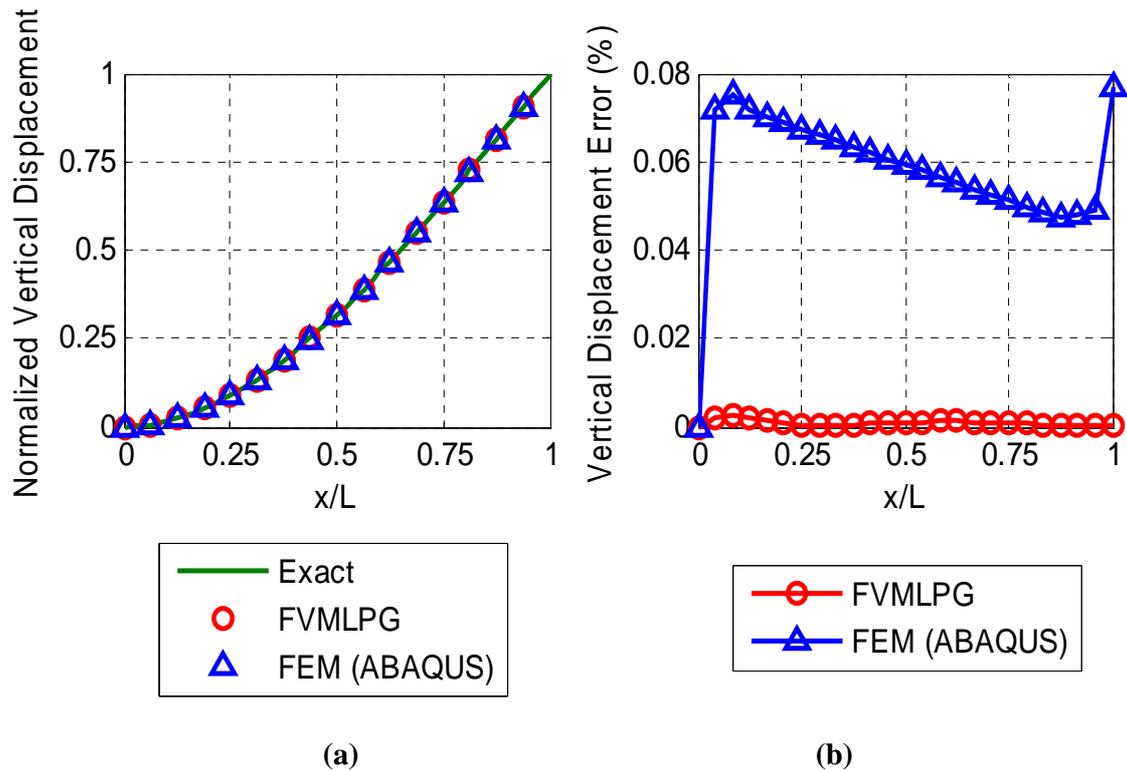


Figure 3.7.4 (a) Déplacement vertical normalisé

(b) Erreur par rapport à la solution exacte

Afin d'étudier la convergence des résultats en déplacement, la norme relative est calculée. ¶ Cette erreur est calculée pour différentes distances d entre les nœuds : 2.0, 1.0, et 0.5 et pour différents rayon normalisés pour chaque sous-domaine r/d , 0.5, 0.6, 0.7, et 0.8. La figure. 3.7.5 montre clairement que le taux de convergence dans cet exemple augmente en fonction de la diminution de la distance entre les nœuds. ¶ En d'autres termes, en augmentant le nombre de nœuds, l'erreur diminue de façon significative.

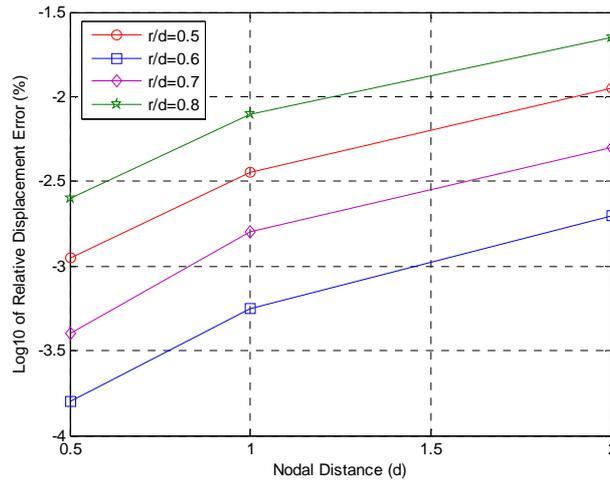


Figure 3.7.5 Norme d'erreur en déplacement (FVMLPG)

Pour l'étude de la stabilité, cet exemple est résolu en considérant une distribution irrégulière des nœuds. Les déplacements obtenus par cette méthode avec 441 nœuds et r irréguliers / $d = 0.6$ indiquent une bonne concordance avec la solution exacte, comme le montre la figure 3.7.6 pour un nombre typique de nœuds. Par conséquent, cette méthode FVMLPG a une bonne stabilité. ¶

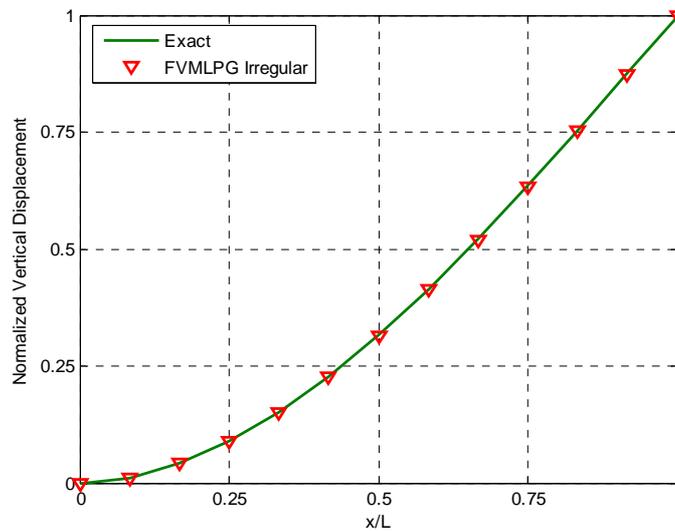


Figure 3.7.6 Déplacement vertical avec une distribution irrégulière de nœuds (441 nœuds)

La convergence de l'erreur relative en déplacement pour la distribution irrégulière de nœuds indique également la bonne exécution de la méthode. ¶ En fait, non seulement il n'y a aucune instabilité avec la distribution irrégulière de nœuds mais il y a également un bon taux de convergence de la norme d'erreur $L2$ (figure. 3.7.7) ¶

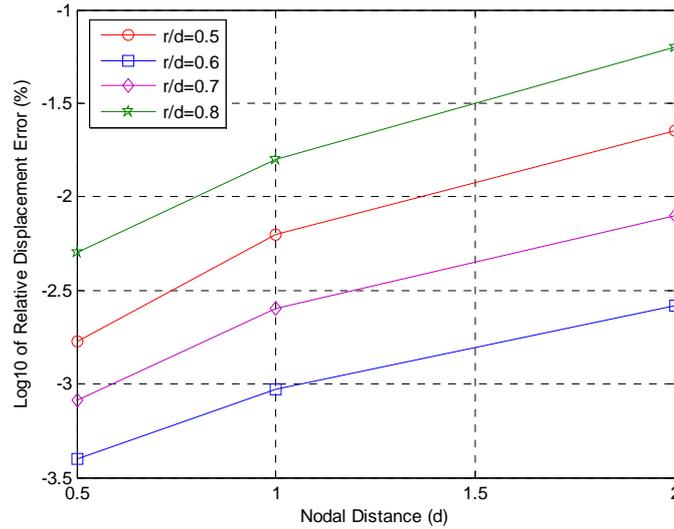


Figure 3.7.7 Norme d'erreur L_2 en fonction de distance entre les nœuds (Distribution irrégulière de nœuds)

La figure 3.7.8 donne les indications sur l'efficacité informatique durant le calcul du déplacement vertical maximum pour les deux méthodes. Il est clair le « TEMPS-CPU » de notre méthode FVMLPG pour les distances nodales $d = 2.0$ (39 nœuds) et $d = 1.0$ (125 noeud) est inférieur à celui de la méthode EF. De même, l'exactitude de FVMLPG pour les distances nodales 2.0 et 1.0 est inférieure à l'exactitude correspondante à la méthode EF. On note toutefois que l'exactitude de FVMLPG pour la distance nodale $d = 0.5$ (441 nœuds) est nettement supérieure à celle de la MEF ; cependant son TEMPS-CPU est plus grand.

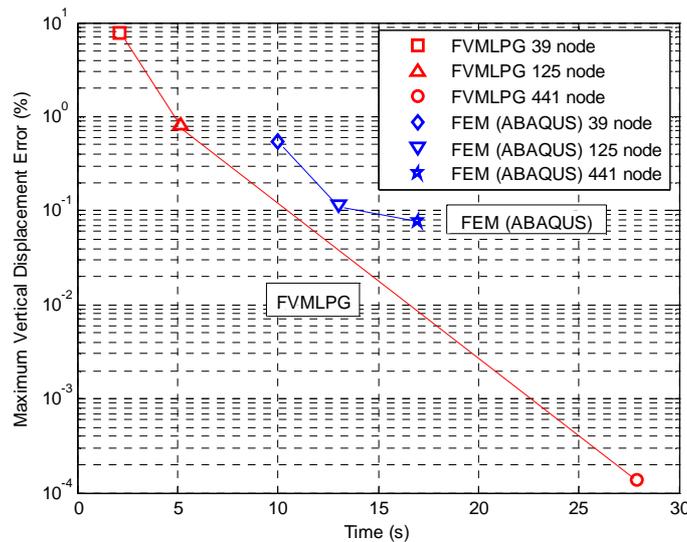


Figure 3.7.8 Efficacité informatique des méthodes FVMLPG et EF

Les résultats de la contrainte normale de flexion dans la section centrale de la poutre en $x = L/2$, s'approche de la solution exacte [111], pour les deux méthodes FVMLPG et EF (figure 3.7.9). L'erreur maximale est de 0.02% dans la FVMLPG et de 0% pour la MEF.

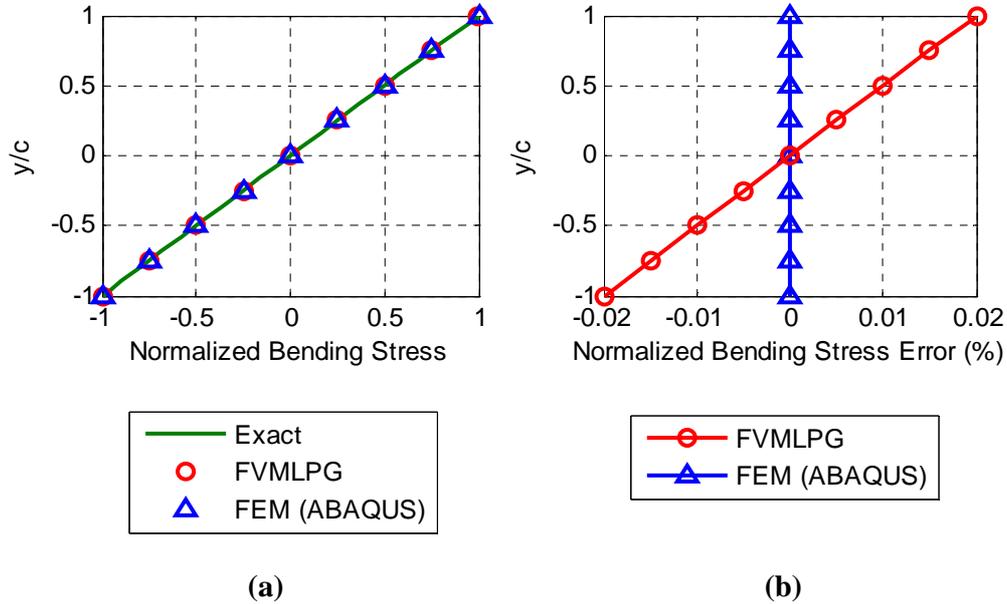


Figure 3.7.9 (a) Contrainte normale de flexion en $x = L/2$

(b) Erreur par rapport à la solution exacte

L'effet de convergence des résultats de contraintes de flexion obtenus par cette approche FVMLPG est montré sur la figure 3.7.10. On constate que l'erreur diminue avec l'augmentation du nombre de nœuds. Ce qui garantit la bonne convergence de cette méthode pour ce calcul de contrainte de flexion. ¶

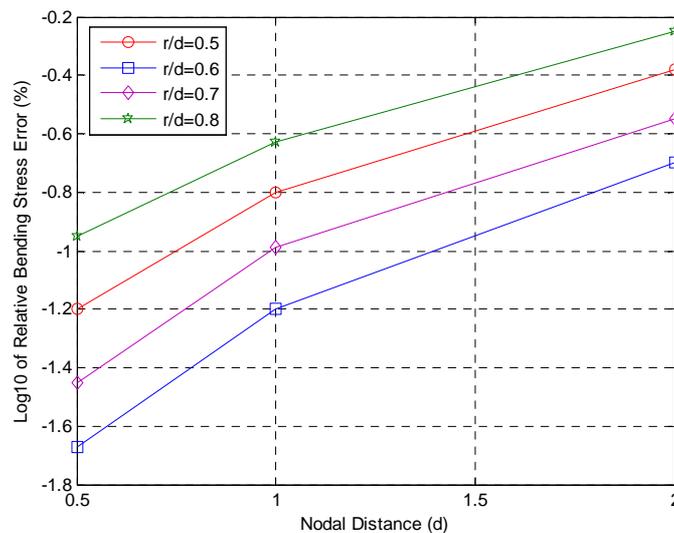


Figure 3.7.10 Erreur relative dans le calcul des contraintes de flexion (FVMLPG)

La distribution irrégulière de nœuds ne provoque pas d'erreur importante dans le calcul des contraintes de flexion (figure 3.7.11). Les résultats numériques coïncident avec la solution exacte. La méthode est par conséquent stable. Ces résultats sont calculés en considérant 441 nœuds irréguliers avec $r/d = 0.6$.

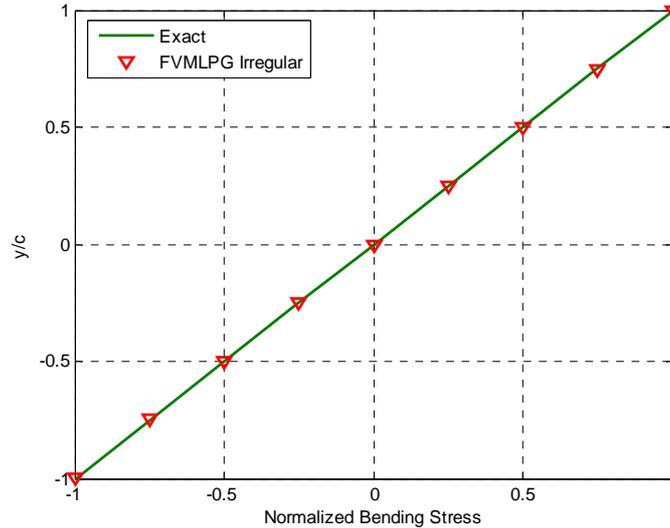


Figure 3.7.11 Contrainte de flexion en fonction pour une distribution irrégulière de (441 nœuds)

Pour la recherche de convergence, l'erreur relative de la norme d'énergie est calculée dans le cas d'une distribution irrégulière de nœuds. Les résultats montrent une bonne convergence quand le nombre de nœuds augmente. Par conséquent, cette méthode a une bonne convergence avec la distribution irrégulière de nœuds (figure. 3.7.12)

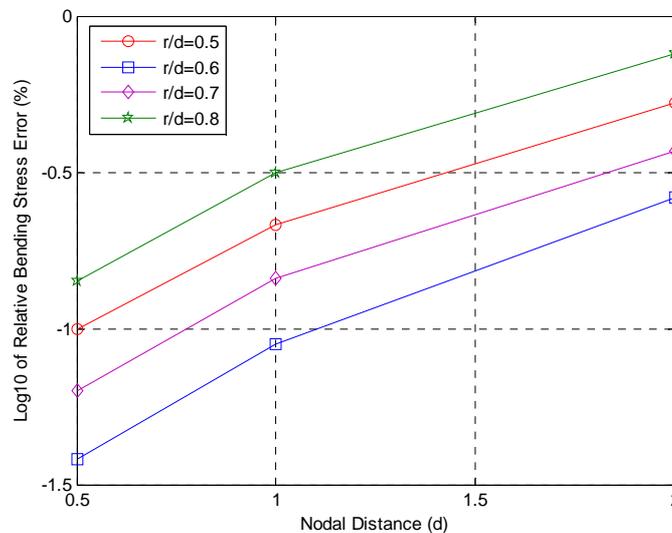


Figure 3.7.12 Erreur relative de calcul de la contrainte dans la section médiane (441 nœuds)

Pour ce calcul de contrainte, la figure 3.7.13 montre que l'efficacité de notre approche est plus faible par rapport au calcul par éléments finis. Ceci est dû au fait que l'exactitude de notre méthode FVMLPG pour les distances nodale 2.0 et 1.0 est inférieure à celle de la méthode EF. La MEF donne la solution exacte de la contrainte de flexion dans les trois distances nodales suivantes : 2 - 1 - 0.5.

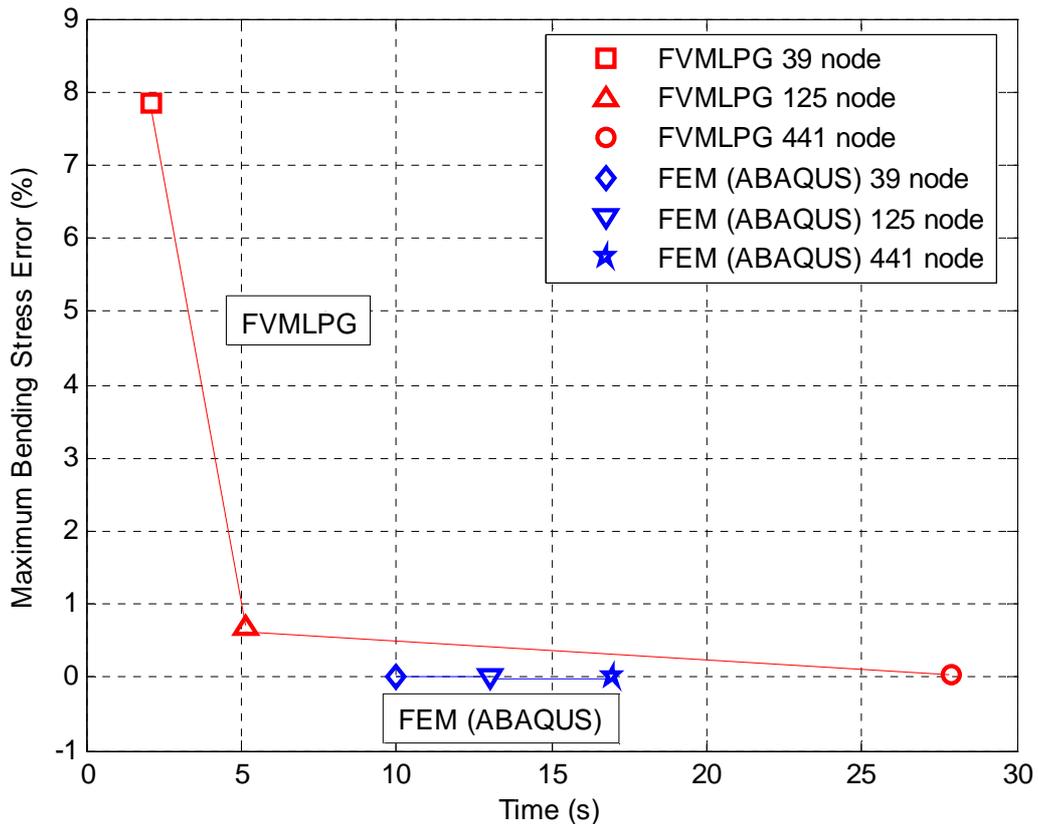


Figure 3.7.13 Efficacité informatique des méthodes FVMLPG et EF (section centrale)

Enfin dans la figure 3.7.14 la distribution normale de contrainte de cisaillement dans la section centrale de la poutre $x = L/2$ donnée par la méthode FVMLPG est bonne. Pour les résultats par EF, il y a une certaine déviation près des frontières supérieures et inférieures de la poutre. L'erreur minimum d'effort de cisaillement du FVMLPG est de 0.01251% et l'erreur maximum est de 0.7086% mais dans le calcul EF l'erreur minimum est de 2.083% et l'erreur maximum est de 22.92% sur les noeuds supérieurs et inférieurs de frontières.

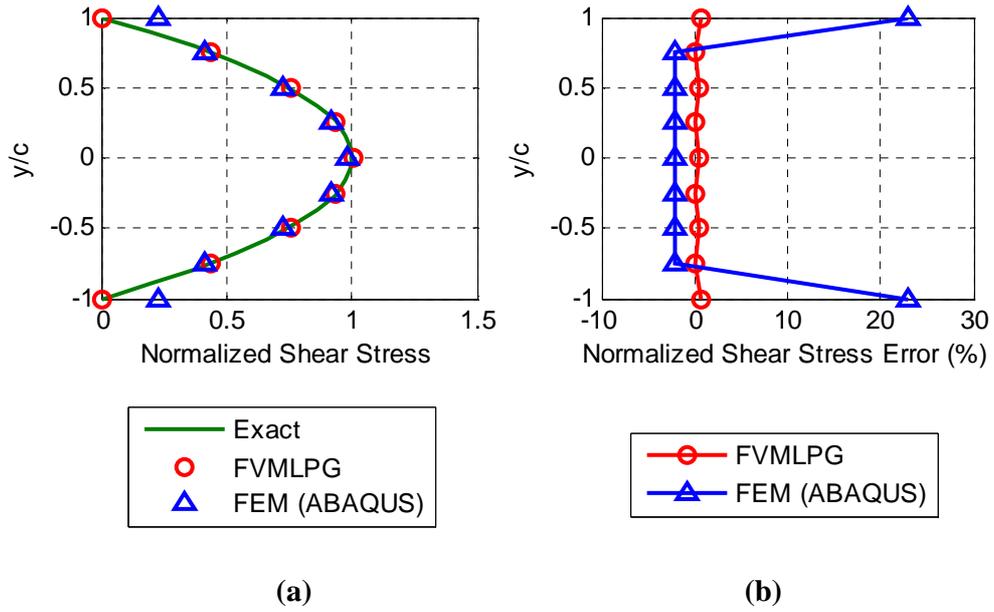


Figure 3.7.14 (a) Contrainte de cisaillement normalisé dans la section centrale
 (b) Erreur par rapport à la solution exacte (FVMLPG et MEF)

Comme pour les exemples précédents, le taux de convergence augmente en fonction du nombre de nœuds n (figure 3.7.15). \blacksquare

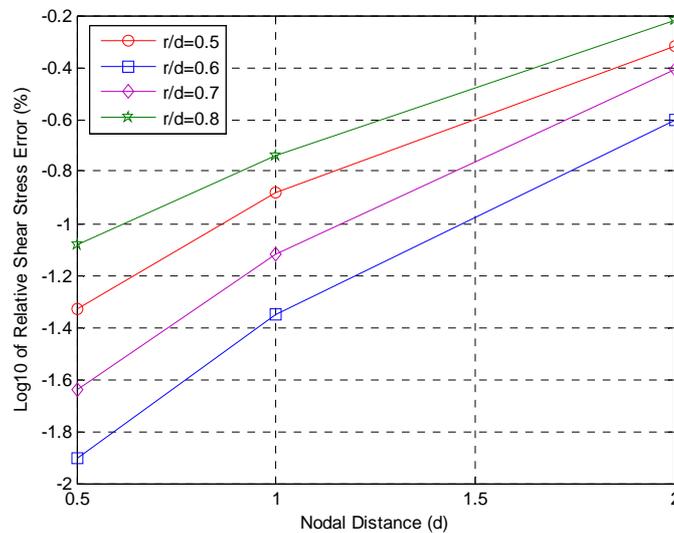


Figure 3.7.15 Erreur relative dans le calcul des contraintes de cisaillement

La stabilité de la méthode dans le calcul des contraintes de cisaillement est démontrée en étudiant l'effet de la distribution irrégulière des nœuds sur les résultats. La figure 3.7.16 montre que cette méthode est stable par comparaison avec la solution exacte (441 nœuds et r irréguliers / $d = 0.6$) \blacksquare

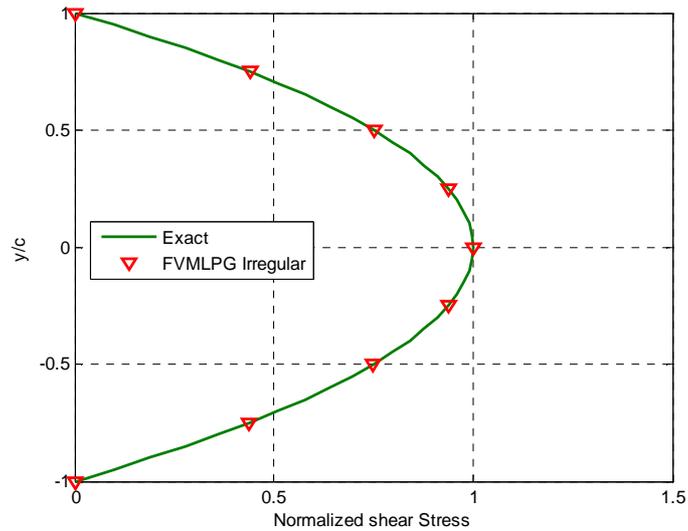


Fig. 3.7.16 Contrainte de cisaillement pour une distribution irrégulière des nœuds. (441 nœuds)

La figure 3.7.17 montre l'erreur en fonction de la distance entre les nœuds pour une distribution irrégulière des nœuds. On constate Les courbes montrent une bonne convergence.

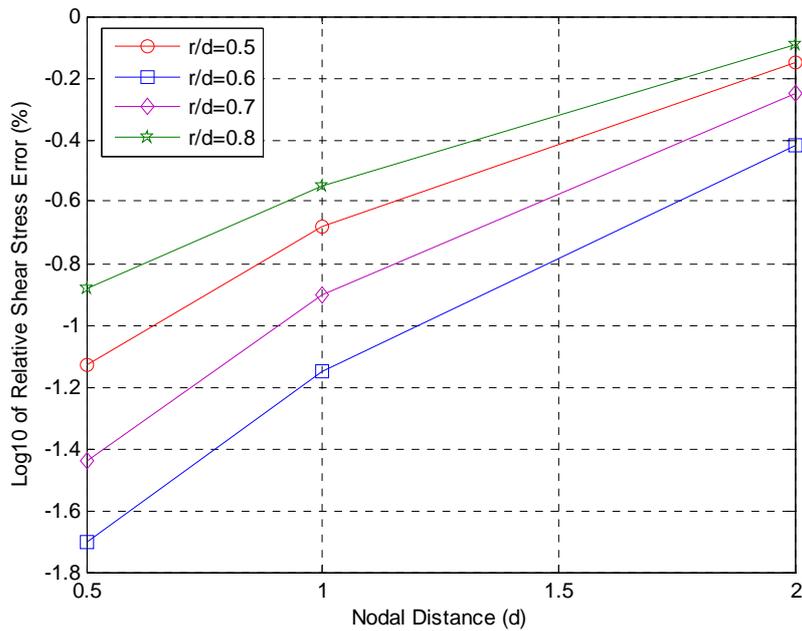


Figure 3.7.17 Erreur relative de la contrainte de cisaillement dans la section médiane (Distribution irrégulière des nœuds (441 nœuds))

Dans la figure 3.7.18, l'erreur minimum de la contrainte de cisaillement maximum dans la distance nodale 0.5 est de 0.7086% cependant cette erreur est de 22.92% par un calcul EF.

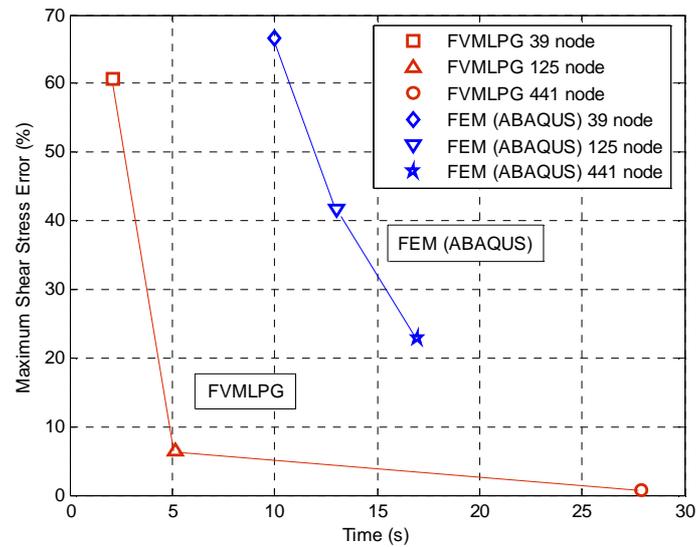


Figure 3.7.18 Efficacité informatique des méthodes FVMLPG et EF - contrainte de cisaillement maximum dans la section centrale en $x = L/2$

Exemple 2 Poutre encastree soumise à une charge de compression

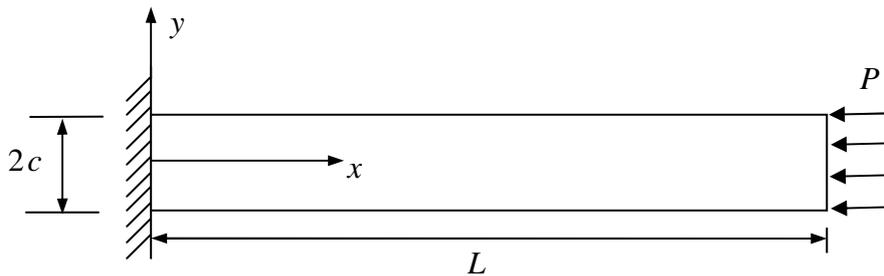


Figure 3.7.19 Poutre encastree sous une charge de compression

Cet exemple est résolu en contrainte plane

$$L = 24, c = 2, E = 1, \nu = 0.25, P = 1$$

Le déplacement axial normalisé obtenu à partir de la méthode FVMLPG est proche de la solution exacte. La géométrie primaire de la poutre et son déplacement axial normalisé sont montrés sur la figure 3.7.20. La distance nodale est de 0.5.

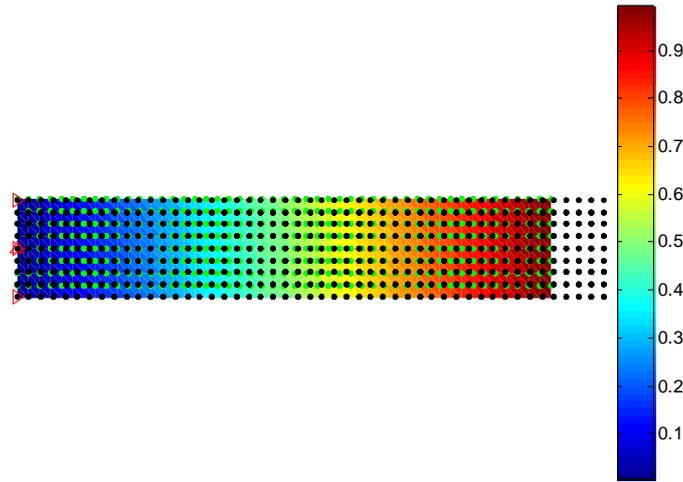
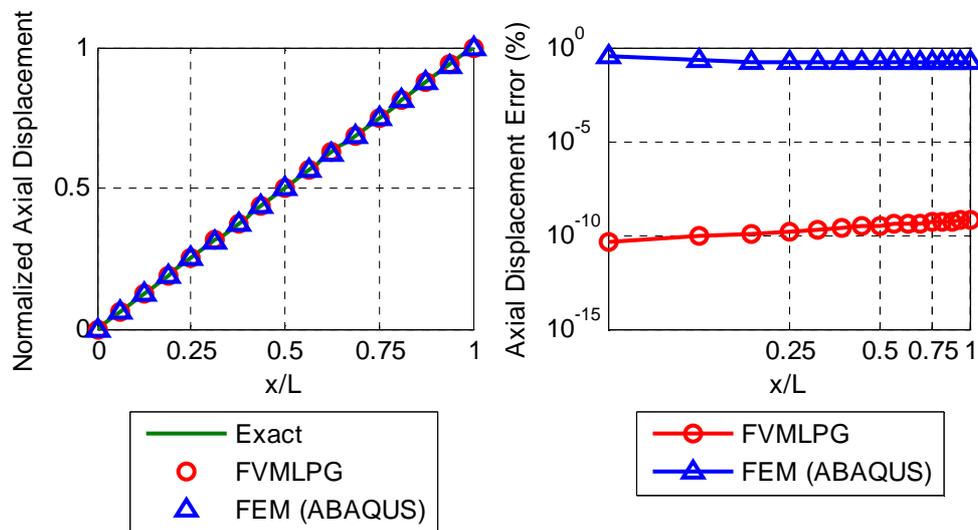


Figure 3.7.20 Déplacement axial normalisé d'une poutre encastée soumise à la compression (441 noeuds)

L'erreur maximale dans le déplacement axial est de $6.14 \times 10^{-10} \%$ pour notre approche FVMLPG. Cette erreur est de 0.1546% pour le calcul par EF. L'exactitude de la méthode FVMLPG est bien meilleure en comparaison avec les résultats par EF obtenus par ABAQUS.



(a)

(b)

Figure 3.7.21 (a) Déplacement axial normalisé

(b) Erreur par rapport à la solution exacte

La figure 3.7.22 montre la norme d'erreur L_2 pour différentes distances d entre les noeuds ; 2.0, 1.0, et 0.5. L'augmentation du nombre de noeuds diminue les erreurs.

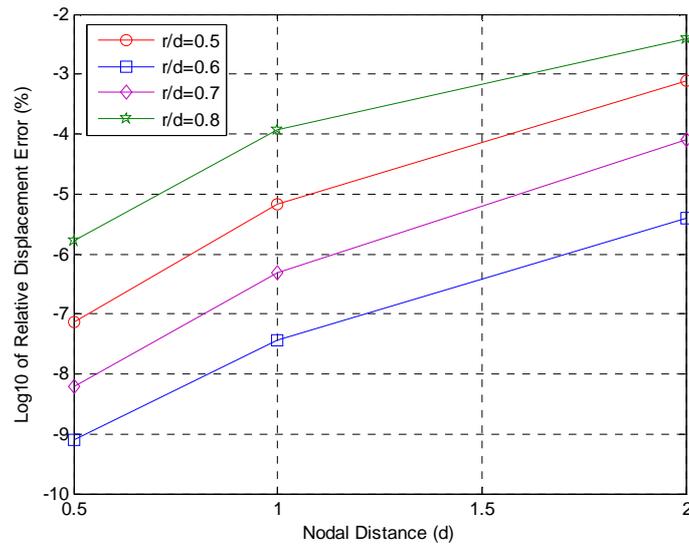


Figure 3.7.22 Norme d'erreur en déplacement L_2

Pour l'étude de la stabilité, l'exemple est résolu en considérant une distribution irrégulière de nœuds. La figure 3.7.23 montre les déplacements axiaux de 441 nœuds irréguliers avec $r / d = 0.6$. ¶Les résultats montrent une bonne stabilité du déplacement axial.¶

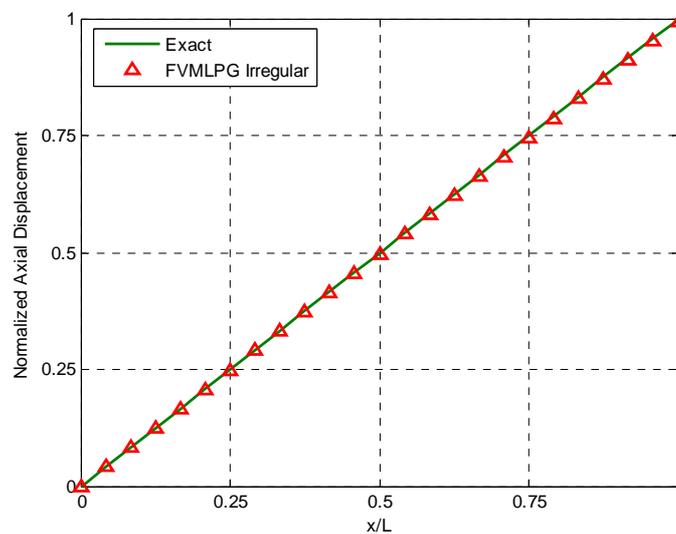


Figure 3.7.23 Déplacement axial pour une distribution irrégulière de nœuds (441 nœuds)

La figure 3.7.24 montre une bonne convergence de l'erreur. ¶Par conséquent, l'irrégularité dans la distribution de nœuds ne peut pas induire l'instabilité ou la divergence des résultats.

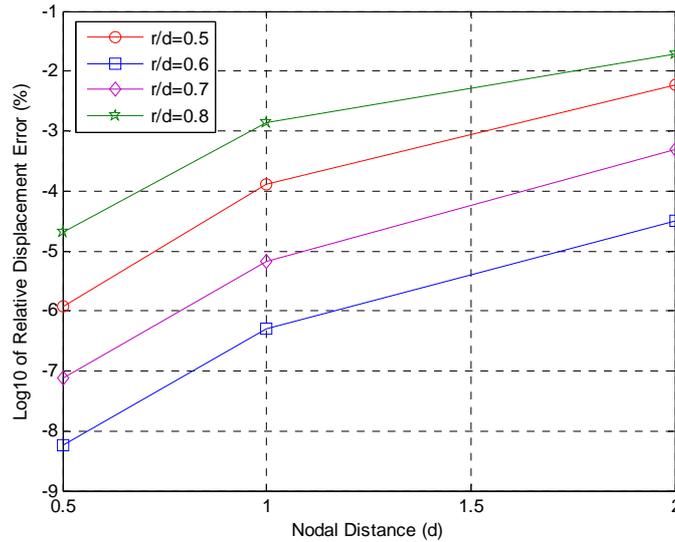


Figure 3.7.24. Norme d'erreur L_2 en fonction pour une distribution irrégulière des nœuds.

La figure 3.7.25 montre que l'efficacité informatique de la méthode FVMLPG dans le déplacement axial est plus élevée que celle de la MEF. Cependant, la période de fonctionnement de notre approche FVMLPG est plus élevée que celle de la MEF mais l'exactitude FVMLPG compense cette lacune.

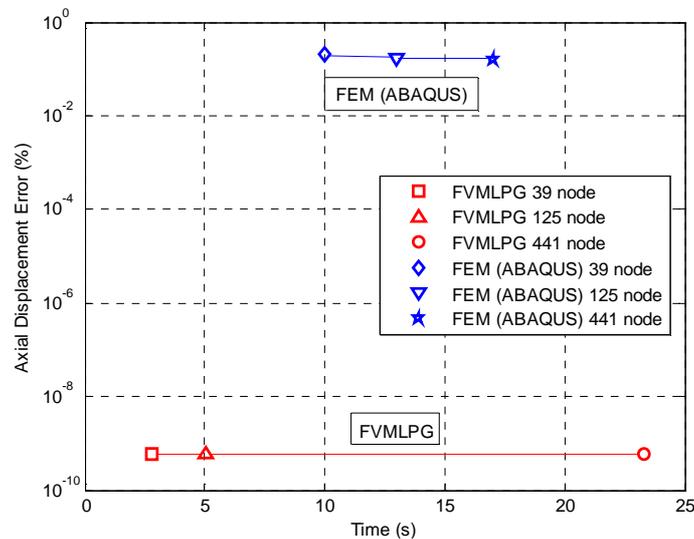


Figure 3.7.25 Efficacité informatique des méthodes FVMLPG et EF (déplacement axial)

La figure 3.7.26 montre l'effort normal ainsi que l'erreur par rapport à la solution exacte dans la section centrale. Les deux méthodes donnent de bons résultats par rapport à la solution exacte. Il y a seulement une très faible dérive des résultats de la MEF dans la fibre supérieure de la section (0.0001%). Cette déviation n'est pas apparue dans le résultat de la méthode FVMLPG.

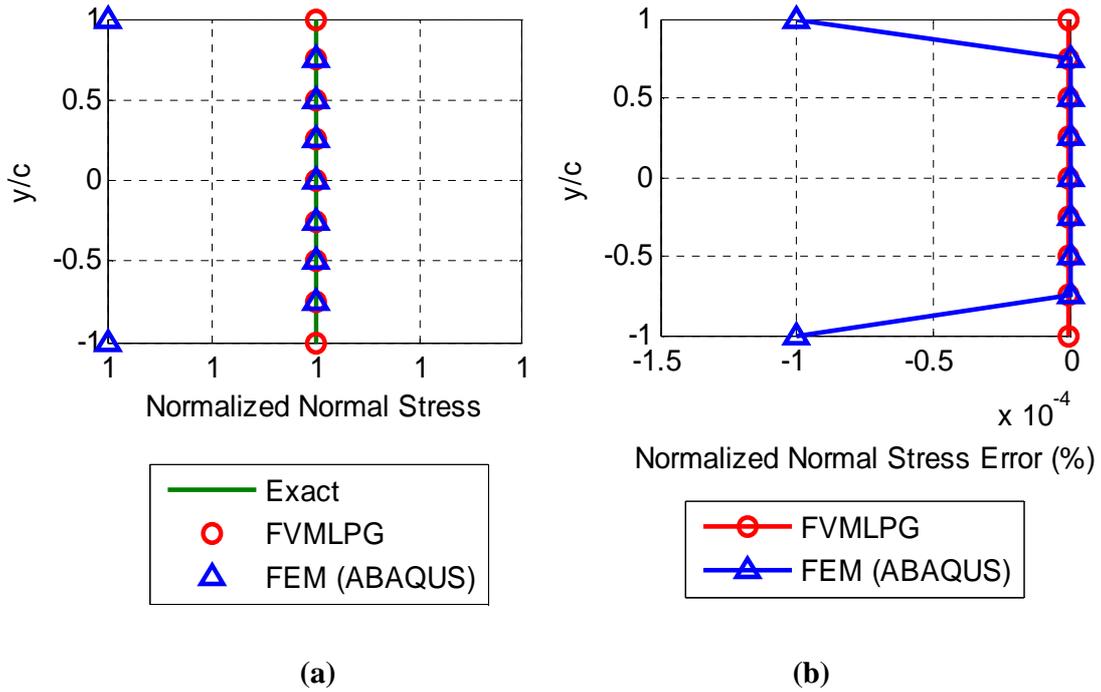


Figure 3.7.26 (a) Contrainte normale
 (b) Erreur par rapport à la solution exacte

Les figures 3.7.27 et 3.7.28 montrent la convergence et la stabilité des résultats de la contrainte normale.

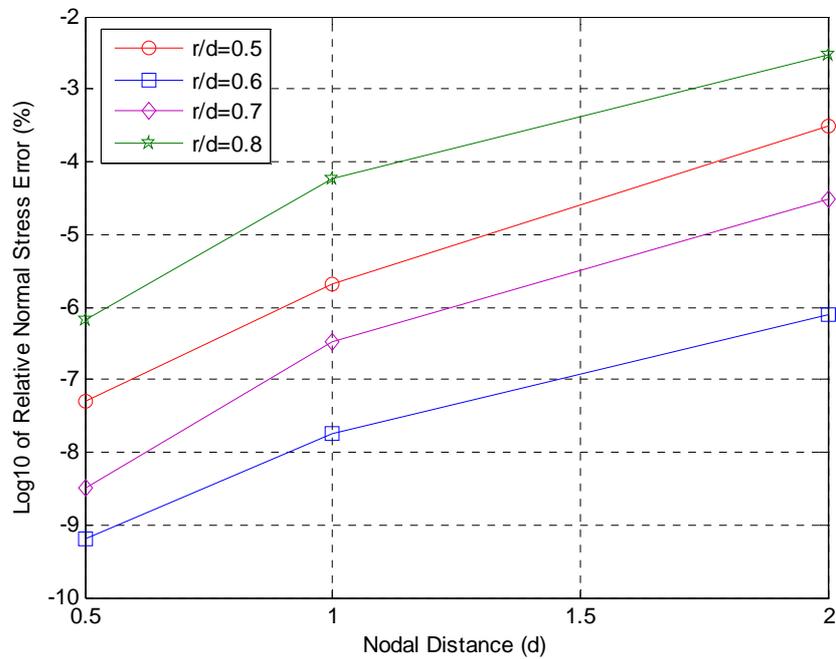


Figure 3.7.27 Erreur de calcul de la contrainte normale en fonction de la distance entre les nœuds.

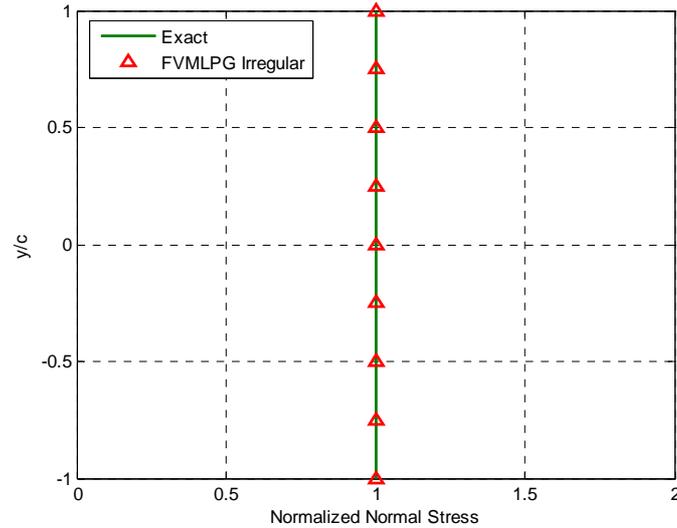


Figure 3.7.28 Contrainte normale pour une distribution irrégulière des nœuds.
Pour $r/d = 0.6$ avec 441 nœuds

La figure 3.7.29 montre l'erreur en fonction de la distance entre les nœuds pour une distribution irrégulière. Ces résultats montrent une bonne convergence de la méthode quand on augmente le nombre de nœuds.

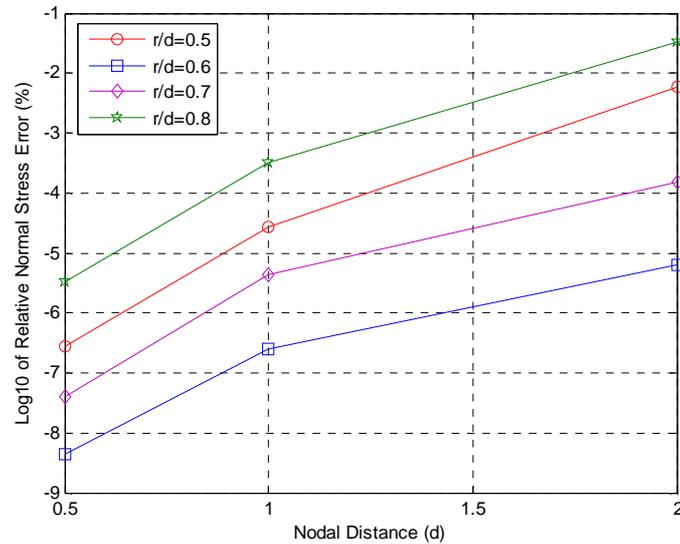


Figure 3.7.29 Erreur de calcul de la contrainte normale en fonction de la distance nodale

La figure 3.7.30 montre l'efficacité informatique de deux méthodes. Il n'y a pas une différence importante entre ces deux méthodes.

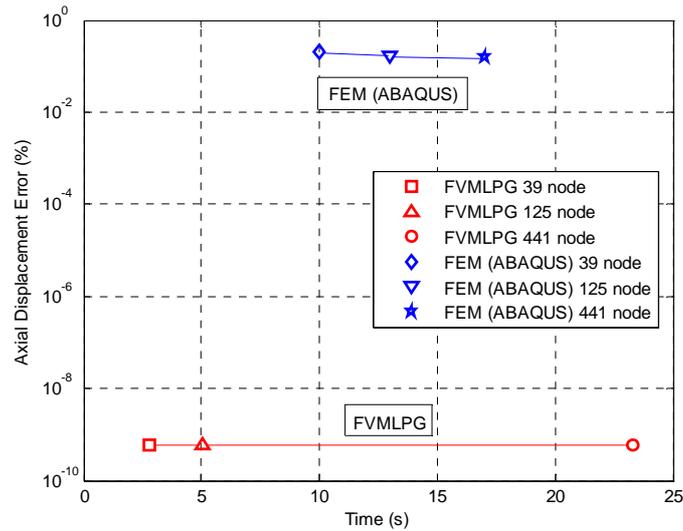


Figure 3.7.30 Efficacité informatique des méthodes FVMLPG et EF (contrainte normale)

Exemple 3 Poutre simplement appuyée soumise à une charge répartie uniforme.

Le troisième exemple est une poutre simplement appuyée soumise à une charge répartie uniforme (figure 3.7.31).

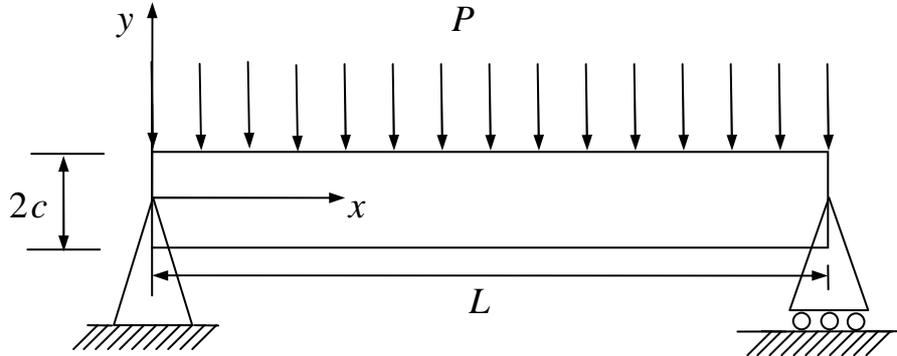


Figure 3.7.31 Poutre simplement appuyée

Cet exemple est résolu en contrainte plane :

$$L = 24, c = 2, E = 1, \nu = 0.25, P = 1$$

Le déplacement latéral de la poutre obtenu par le FVMLPG est montré sur la figure 3.7.32.

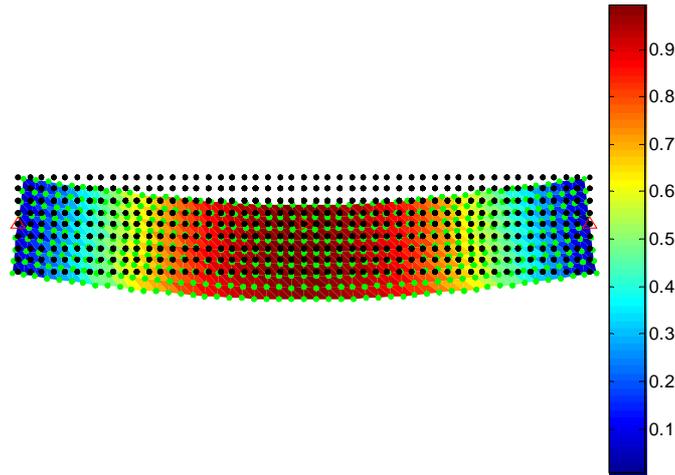


Figure 3.7.32 Le déplacement latéral normalisé d'une poutre simplement appuyée (441 nœuds)

L'erreur maximale par rapport à la solution exacte (figure 3.7.33) du déplacement latéral obtenue par la méthode FVMLPG est de 0.03078% . Le calcul par EF donne une erreur de 1.507% .

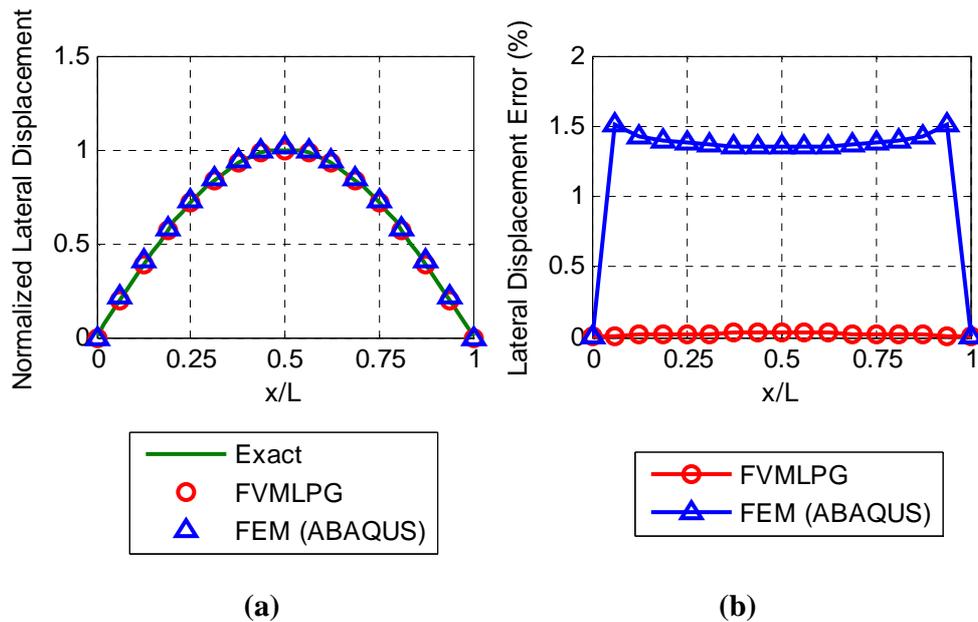


Figure 3.7.33 (a) Déplacement latéral normalisé

(b) Erreur par rapport à la solution exacte

Comme les exemples précédents, les figures 3.7.34 à 3.7.36 montrent la convergence et la stabilité dans ces calculs de déplacements.

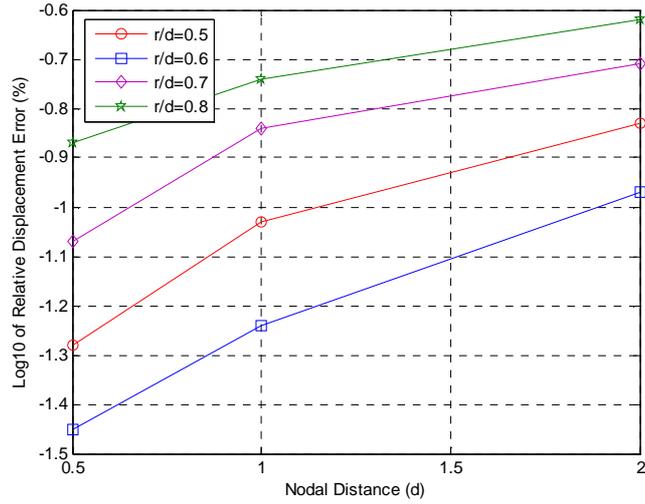


Figure 3.7.34 Norme d'erreur L2

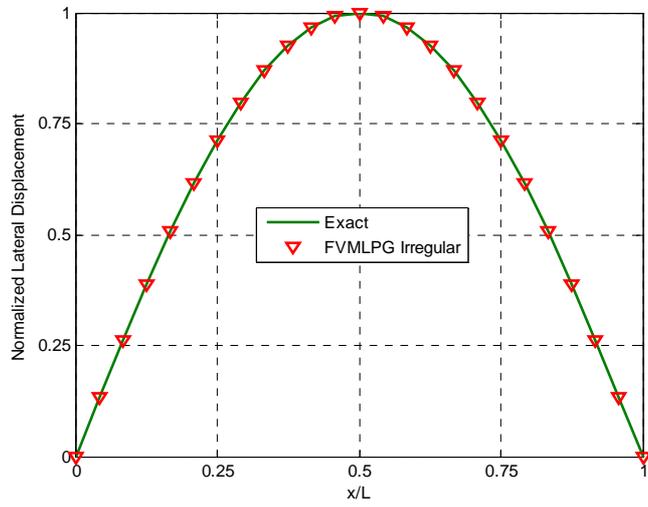


Figure 3.7.35 Déplacement latéral pour une distribution irrégulière des nœuds. (441 nœuds irréguliers pour $r/d = 0.6$)

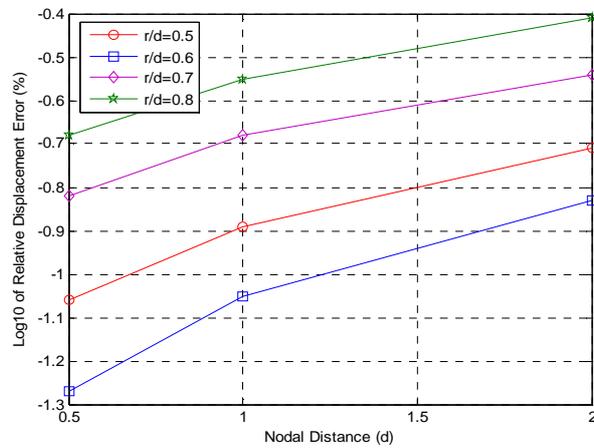


Figure 3.7.36 Norme d'erreur L_2 pour une distribution irrégulière de nœuds.

La figure 3.7.37 montre l'efficacité informatique de ces méthodes dans le calcul du déplacement latéral. L'efficacité informatique de notre approche FVMLPG est bien meilleure que celle de la MEF.

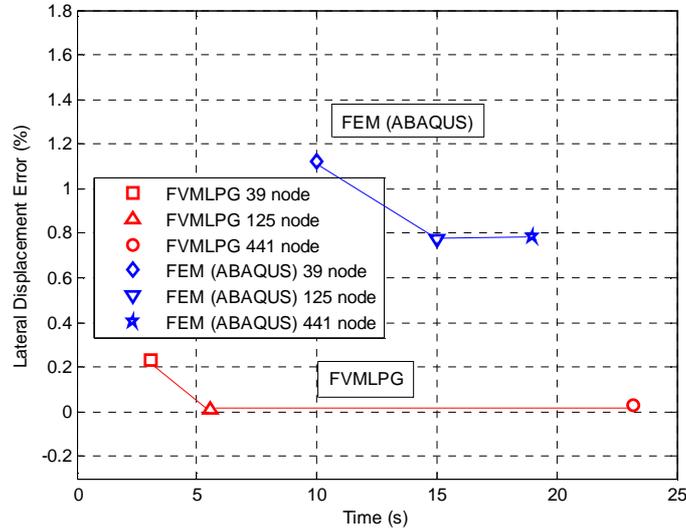


Figure 3.7.37 Efficacité informatique des méthodes FVMLPG et MEF (déplacement latéral)

L'erreur maximale pour la contrainte de flexion est respectivement de 0.03153% pour la méthode FVMLPG et de 0.1555% pour le calcul par EF.

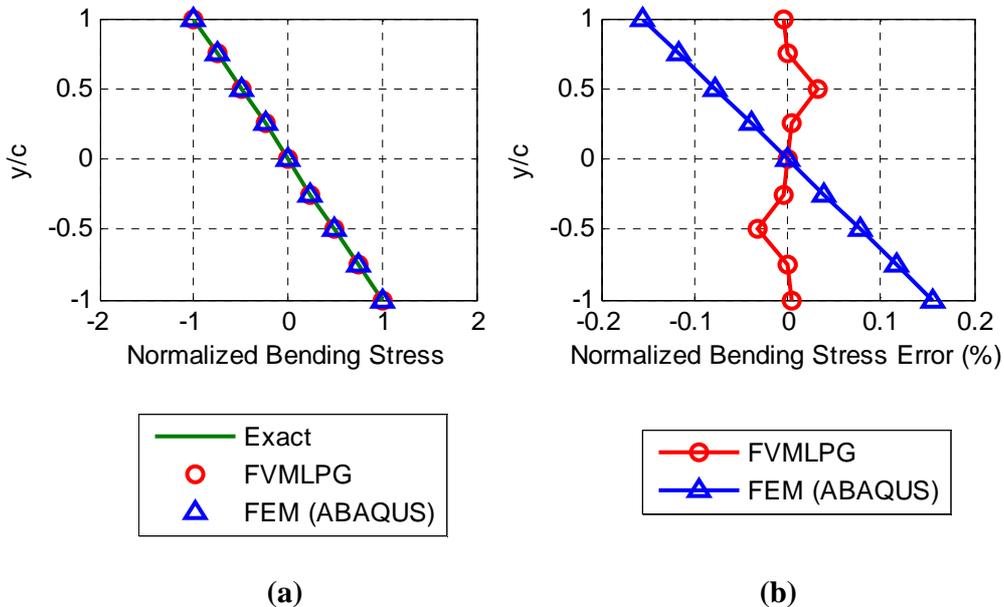


Figure 3.7.38 (a) Contrainte de flexion normalisée

(b) Erreur par rapport à la solution exacte

Les figures 3.7.39 à 3.7.41 montrent les tests de convergence pour ces calculs. Les résultats restent stables.

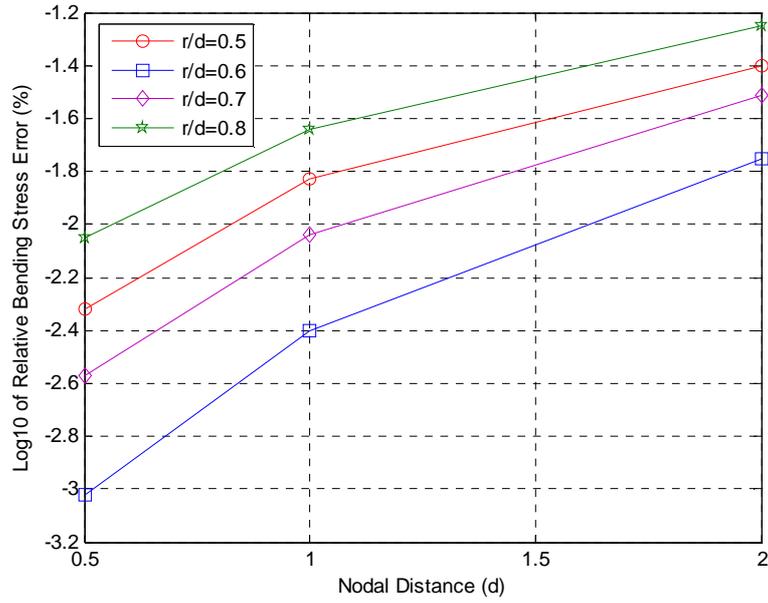


Figure 3.7.39 Erreur de calcul de la contrainte de flexion

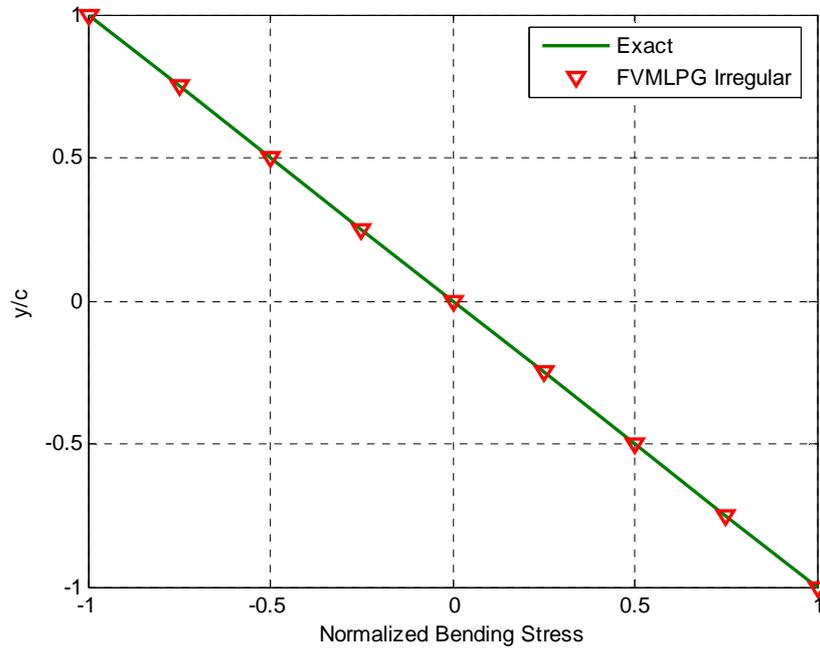


Figure 3.7.40 Contrainte de flexion pour une distribution irrégulière de 441 nœuds.

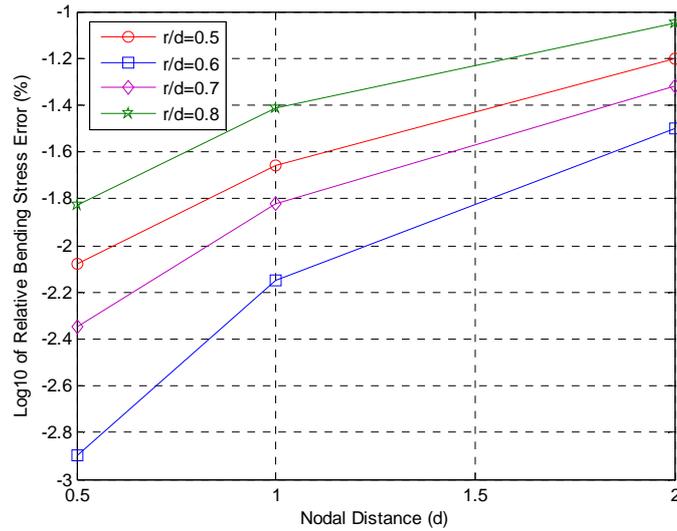


Figure 3.7.41 Erreur de calcul de la contrainte de flexion dans la section médiane (Distribution irrégulière de 441 nœuds)

L'efficacité informatique des deux méthodes FVMLPG et MEF dans le calcul de la contrainte de flexion dans la section centrale de la poutre, $x = L/2$ est montrée sur la figure 3.7.42. L'exactitude augmente en diminuant la distance nodale mais l'exactitude du FVMLPG est plus élevée que celle de la MEF. Cependant le TEMPS-CPU de la méthode FVMLPG pour la distance nodale 0.5 est plus grand que le TEMPS-CPU correspondant à la MEF.

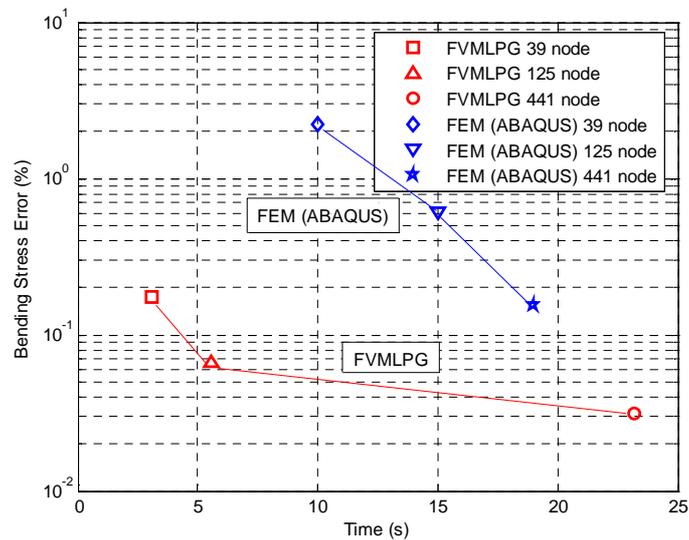


Figure 3.7.42 Efficacité informatique des méthodes FVMLPG et MEF : contrainte de flexion dans la section centrale de la poutre, $x = L/2$

Le comportement des deux méthodes dans le calcul de la contrainte de cisaillement est indiqué sur la figure 3.7.43. Les résultats montrent une bonne concordance entre les résultats de la méthode FVMLPG et la solution exacte. L'erreur maximale de la FVMLPG est 0.6127% .

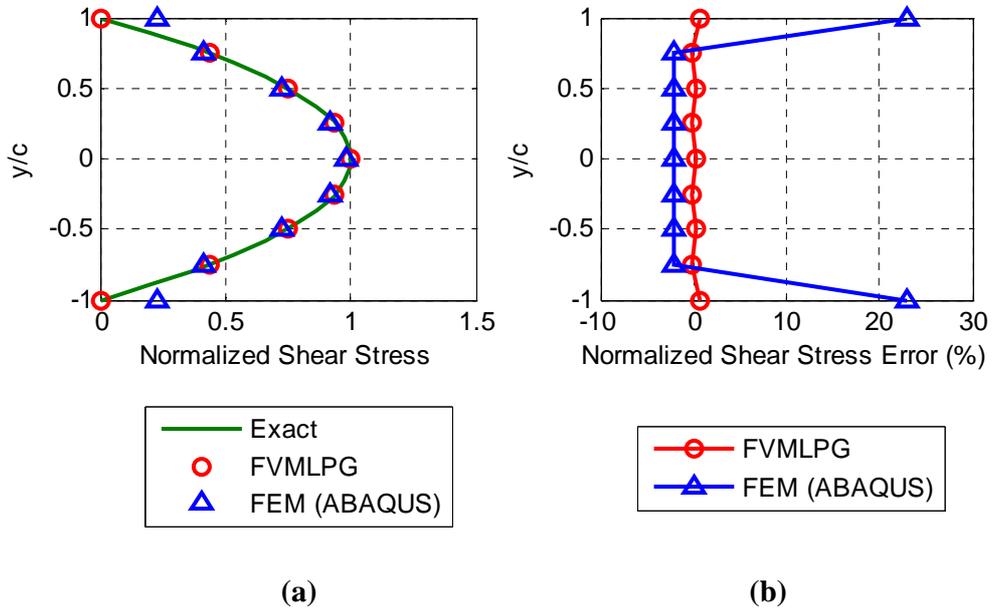


Figure 3.7.43 (a) Contrainte de cisaillement
(b) Erreur par rapport à la solution exacte

Les figures 3.7.44 à 3.7.46 montrent les tests de convergence pour les calculs de contrainte de cisaillement. Les résultats montrent encore un bon comportement de la méthode.

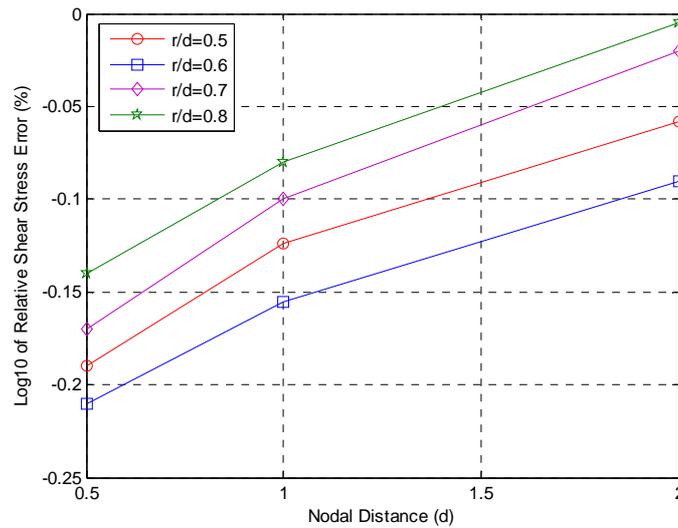


Figure 3.7.44 Erreur de calcul de la contrainte de cisaillement en fonction de la distance entre les noeuds

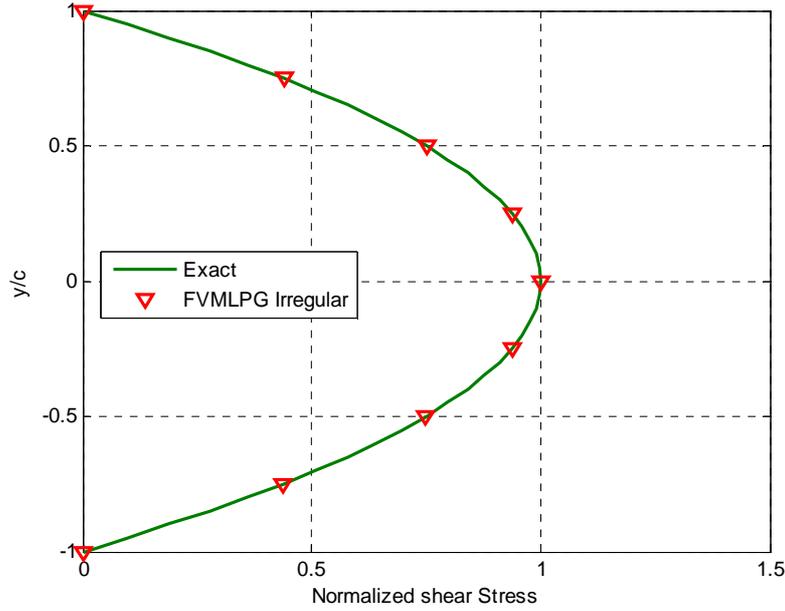


Figure 3.7.45 Contrainte de cisailment avec une distribution irrégulière de 441 nœuds.

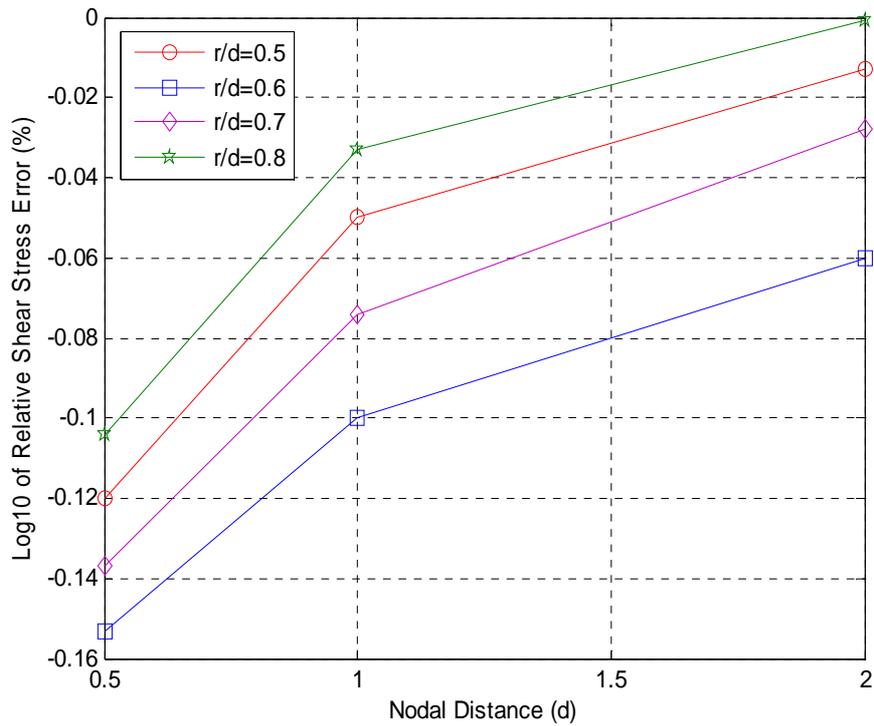


Figure 3.7.46 Erreur de calcul de la contrainte de cisailment dans la section médiane (Distribution irrégulière de 441 nœuds)

La figure 3.7.47 montre l'efficacité informatique de la méthode FVMLPG par rapport à la méthode des éléments finis.

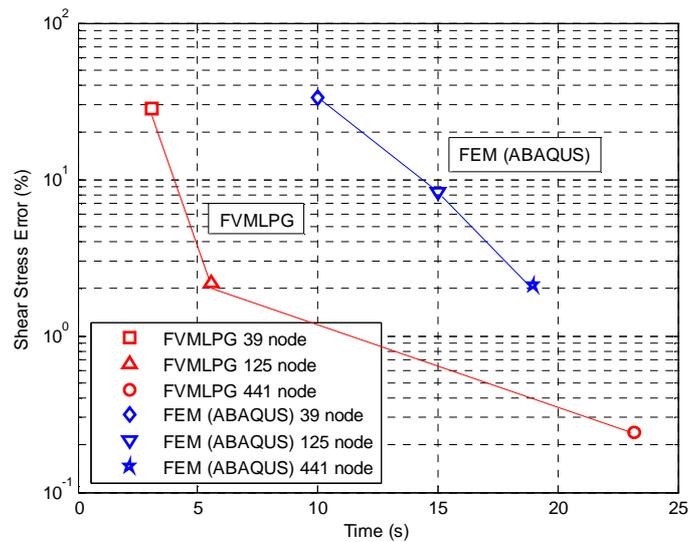


Figure 3.7.47 Efficacité informatique dans le calcul de la contrainte de cisaillement dans la section centrale de la poutre, $x = L/2$

3.8 Remarques et conclusion

Dans ce chapitre l'approche FVMLPG est appliquée aux problèmes bidimensionnels. La procédure de calcul numérique est implémentée dans un code développé dans MATLAB. Dans cette méthode, les variables indépendantes sous la forme faible locale sont des contraintes. L'interpolation directe en contrainte simplifie le problème (pas de différentiation des fonctions de forme). L'exactitude et l'efficacité informatique de cette méthode sont étudiées à partir de quelques exemples types en élastostatiques. Une étude comparative entre cette approche et la méthode EF en utilisant ABAQUS est également effectuée. La méthode de FVMLPG a été évaluée à partir de plusieurs tests. Les résultats obtenus ont montré un bon comportement de la méthode par rapport à la solution exacte.

Chapitre 4

FVMLPG appliquée aux poutres minces

Dans le chapitre 2 la méthode de FVMLPG a été développée pour l'étude des déformations des barres dans le cas de problèmes C^0 . Dans ce chapitre, la méthode de FVMLPG est étendue pour l'étude de la flexion des poutres. Une forme faible locale est développée à partir de la forme pondérée résiduelle classique des équations d'équilibre. La méthode d'interpolation des moindres carrés mobiles est utilisée pour construire les fonctions d'approximations. Avec la procédure de Petrov-Galerkin, les fonctions tests sont choisies dans un espace différent de celui des fonctions d'essai. Les matrices du système d'équation sont déduites en substituant les fonctions d'essai et test dans la forme faible locale.

4.1 Rappel succinct de la théorie des poutres

La méthode de FVMLPG pour les problèmes C^1 est développée en utilisant les hypothèses classiques de poutres d'Euler Bernoulli. Dans ce type de poutre (figure 4.1.1), les sections planes, normales aux fibres avant déformation demeurent normales aux fibres après. Le déplacement w suivant l'axe z est une fonction x

$$w = w(x), \quad u = u(x), \quad \text{et} \quad v = 0. \quad (4.1.1)$$

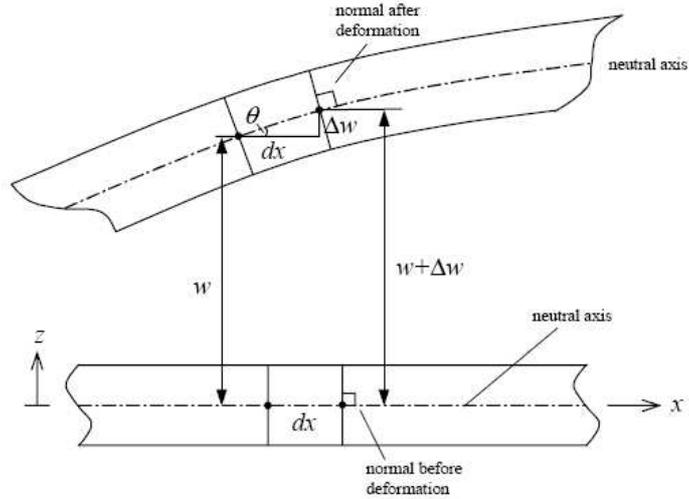
$$\tan \theta = \frac{CD}{AC} = \frac{\Delta w}{\Delta x}. \quad (\text{figure 4.1.1 b}) \quad (4.1.2)$$

En petites déformation, on peut écrire $\Delta x \rightarrow 0$, et, $\tan \theta \approx \theta$, ce qui donne

$$\theta = \frac{dw}{dx} \tag{4.1.3}$$

θ est la pente de la déformée par rapport à l'axe neutre. Soit le triangle ABC représenté sur la dans la figure 4.1.1c.

$$BAC = CAD = \theta \quad (\text{hypothèse de Bernoulli}) \tag{4.1.4}$$



(a) configuration de la poutre avant et après déformation

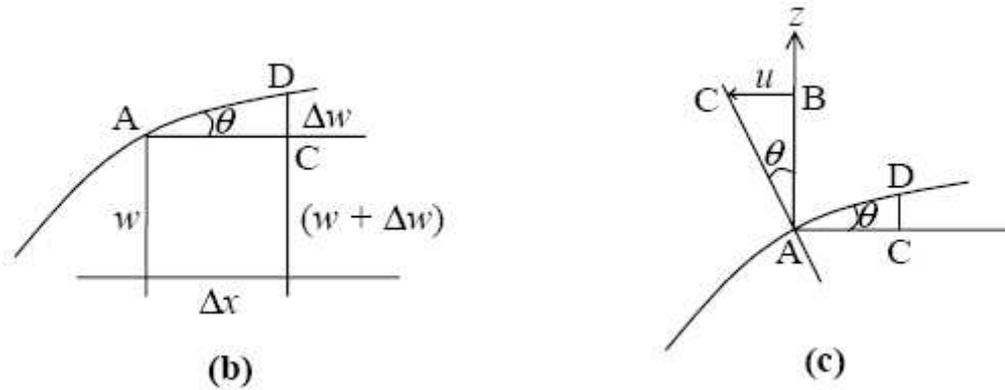


Figure 4.1.1 Poutre d'Euler Bernoulli

Dans le triangle ABC de la figure 4.1.1c,

$$\frac{BC}{AB} = \tan \theta \tag{4.1.5a}$$

où

$$u = -z \tan \theta \approx -z \frac{dw}{dx}. \quad (4.1.5b)$$

Les déformations correspondantes aux déplacements $u = -z (dw / dx)$, $v = 0$, et $w = w(x)$ peuvent alors s'écrire :

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} = -z \frac{d^2 w}{dx^2} & \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0 \\ \varepsilon_y &= \frac{\partial v}{\partial y} = 0 & \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} = 0 \end{aligned} \quad (4.1.6)$$

$$\varepsilon_z = \frac{\partial w(x)}{\partial z} = 0 \quad \gamma_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} = \frac{dw}{dx} - \frac{dw}{dx} = 0$$

Toutes les déformations sont nulles à l'exception de ε_x . En utilisant les relations contraintes déformations, la contrainte σ_x peut donc s'écrire :

$$\sigma_x = E \varepsilon_x = -Ez \frac{d^2 w}{dx^2}. \quad (4.1.7)$$

Soit un petit élément de poutre soumis à un moment (figure 4.1.2).

$$M = - \int_A z (\sigma_x b) dz = \int_A E z^2 b \frac{d^2 w}{dx^2} dz = E \frac{d^2 w}{dx^2} \int_A b z^2 dz. \quad (4.1.8)$$

La terme $\int_A b z^2 dz$ désigne le moment quadratique de la section par rapport à l'axe y (inertie de flexion, I)

$$M = EI \frac{d^2 w}{dx^2} \quad (4.1.9)$$

$$V = -EI \frac{d^3 w}{dx^3} \quad (4.1.10)$$

Dans ce chapitre, on considère les poutres minces d'Euler Bernoulli en petites déformations.

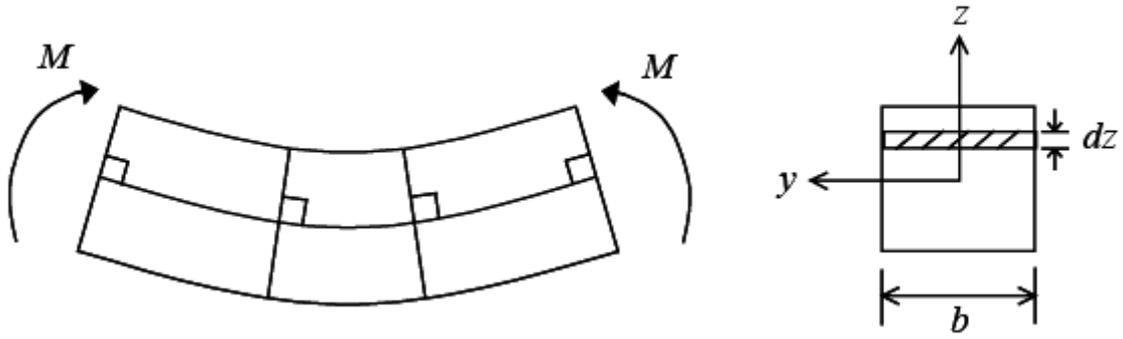


Figure 4.1.2 Élément de poutre soumis à un moment M

4.2 Forme faible locale pour des problèmes de poutre d'Euler Bernoulli

L'équation d'équilibre pour une poutre d'Euler Bernoulli est

$$EI \frac{d^4 w}{dx^4} = f \quad \text{dans le domaine } \Omega \ (0 \leq x \leq l) \text{ avec la frontière } \Gamma \quad (4.2.1)$$

où l est la longueur et EI est la rigidité de flexion de la poutre, et f est la charge répartie. Les conditions aux limites à $x = 0$ et à $x = l$ peuvent prendre plusieurs formes :

Conditions aux limites essentielles (EBCs) :

$$w = \tilde{w} \quad \text{sur } \Gamma_w \text{ et } \frac{dw}{dx} = \tilde{\theta} \quad \text{sur } \Gamma_\theta \quad (4.2.2)$$

Conditions aux limites naturelles (NBCs) :

$$V = \tilde{V} \quad \text{sur } \Gamma_V \text{ et } M = \tilde{M} \quad \text{sur } \Gamma_M \quad (4.2.3)$$

V et M désignent respectivement l'effort tranchant et le moment fléchissant correspondant au déplacement w (équations 4.1.9 et 4.1.10)

$$V = -EI \frac{d^3 w}{dx^3} \quad \text{et} \quad M = EI \frac{d^2 w}{dx^2} \quad (4.2.4)$$

Γ_w , Γ_θ , Γ_V , et Γ_M sont les points de frontière où le déplacement (w), la pente (θ), le cisaillement (V)

et le moment (M) sont respectivement imposés. \tilde{w} , \tilde{V} , $\tilde{\theta}$ et \tilde{M} désignent les valeurs aux point limites [100, 112].

Pour la simplicité de l'écriture, nous considérons une équation générique adimensionnelle comme suit :

$$\frac{d^4 w}{dx^4} = 1 \quad (4.2.5)$$

La forme faible de l'équation d'équilibre est obtenue de la même façon que pour les problèmes C^0 . L'erreur résiduelle à minimiser est :

$$R = \frac{d^4 w}{dx^4} - 1. \quad (4.2.6)$$

La forme résiduelle pondérée classique de l'équation pour les problèmes d'ordre quatre est constituée, en multipliant le résiduel par une fonction de poids $v(x)$, intégrée au-dessus du sous domaine Ω_s ,

$$\int_{\Omega_s} \left(\frac{d^4 w}{dx^4} - 1 \right) v dx = 0. \quad (4.2.7)$$

Une solution approximative pour w est choisie tels que chaque terme dans la solution approximative doit être quatre fois différentiables et satisfasse toutes les conditions aux limites. En fait il est difficile d'obtenir des dérivées précises d'ordre supérieur de w dans l'approximation des moindres carrés mobiles (MLS). Par conséquent, cette forme faible n'est pas appropriée pour l'exécution numérique, en utilisant l'approximation de type MLS.

Pour obtenir une méthode sans maillage précise et efficace, on doit diminuer l'ordre des dérivés de la fonction d'essai dans la forme faible locale. Il y a deux manières d'atteindre cet objectif. Soit au moyen d'intégration par parties dans lesquelles la différentiation peut être transférée à partir de la fonction d'essai, w , à la fonction test, v . Puis, les dérivées d'ordre supérieur dans l'intégration du domaine disparaîtront. Cependant, les dérivées d'ordre supérieur de la fonction d'essai apparaissent toujours dans l'intégrale de frontière locale (frontière locale pour des problèmes 1-D). Une autre approche prometteuse est l'utilisation de l'approche de FVMLPG, où, des interpolations

sans maillage indépendantes sont également utilisées pour les dérivées, aussi bien que pour la fonction.

Le problème décrit par l'équation (4.2.5), est décomposé en termes de quatre équations de premier ordre, avec les variables Φ_i :

$$w = \phi_1 \quad (4.2.8)$$

$$\frac{d\phi_1}{dx} = \phi_2 \quad (4.2.9)$$

$$\frac{d\phi_2}{dx} = \phi_3 \quad (4.2.10)$$

$$\frac{d\phi_3}{dx} = \phi_4 \quad (4.2.11)$$

$$\frac{d\phi_4}{dx} = 1 \quad (4.2.12)$$

Soit en notation matricielle,

$$\begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{Bmatrix}' + \begin{Bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{Bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \quad (4.2.13)$$

où

$$A_{ij}\phi_j' + B_{ij}\phi_j = f_i \quad j=1,\dots,4 \quad (4.2.14)$$

Chacun terme Φ_i est interpolé par MLS, comme suit :

$$\phi_j^h(x) = \sum_{l=1}^N \phi_j^l(x) \hat{\phi}_j^l; \quad x \in \Omega_s \quad (4.2.15)$$

avec $j = 1 \dots 4$

En utilisant une fonction test qui est une fonction d'étape de Heaviside dans chaque Ω_s [c'est-à-dire, $x_{k-1} \leq x \leq x_k$, qui peut recouvrir des sous domaines comme dans la méthode des volumes finis], le système d'équation linéaire simple suivant est obtenu pour les valeurs nodales de $\hat{\phi}_j^I$

$$\int_{x_{k-1}}^{x_k} [A_{ij}\phi_j' + B_{ij}\phi_j - f_i] dx = 0 \quad (4.2.16)$$

L'équation (4.2.16) peut également être écrite comme suite

$$\int_{x_{k-1}}^{x_k} [B_{ij}\phi_j - f_i] dx + [A_{ij}\phi_j]_{x_{k-1}}^{x_k} = 0 \quad (4.2.17)$$

L'équation (4.2.17) comporte seulement les interpolations de MLS pour chaque Φ_i , ($I = 1 \dots 4$), mais non pour leurs dérivées. Ainsi, la méthode de FVMLPG présentée dans ce chapitre est entièrement analogue à la méthode des volumes finis.

4.3 Exemples numériques

Plusieurs exemples numériques sont étudiés pour tester l'efficacité de la méthode de FVMLPG pour des problèmes de poutres.

On considère une poutre de rigidité de flexion constante EI et de longueur $5l$. La longueur $5l$ a été spécifiquement choisie pour éviter la graduation par l'unité. Six modèles avec 6, 11, 21, 41, 81, et 161 noeuds uniformément distribués sur la longueur de la poutre sont considérés. La figure 4.3.1 montre un exemple type de modèle avec 11 noeuds.

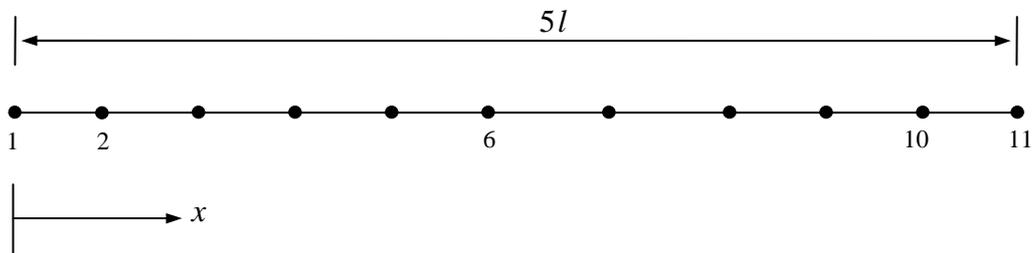


Figure 4.3.1 modèle à 11 noeuds

Les distances entre les noeuds ($\Delta x / l$) sont 1.0, 0.5, 0.25, 0.125, 0.0625, et 0.03125 pour les modèles à 6 -, 11 -, 21 -, 41 -, 81 et 161 noeuds. Une fonction de base linéaire est utilisée. Les matrices du système sont de la forme de l'équation 5.2.17

$$\int_{x_{k-1}}^{x_k} [B_{ij}\phi_j - f_i] dx + [A_{ij}\phi_j]_{x_{k-1}}^{x_k} = 0. \quad (4.3.1)$$

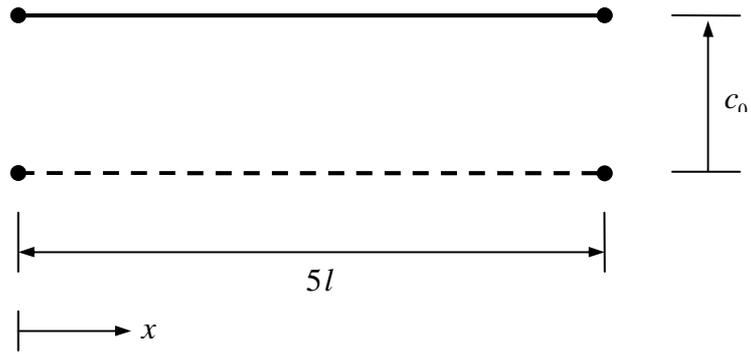
Dans un premier temps, des patches tests sont appliqués aux problèmes simples où une approche locale du même rang est développée pour améliorer l'exactitude de la méthode. Les normes d'erreur des patches tests pour les méthodes globales et locales sont comparées pour démontrer la validité de l'approche locale. Puis, les principes généraux de base pour choisir les divers paramètres définis pour l'utilisateur sont discutés. Plusieurs problèmes aux valeurs limites mixtes sont considérés. Enfin la méthode est appliquée aux cas de poutres continues.

4.3.1 Patch tests

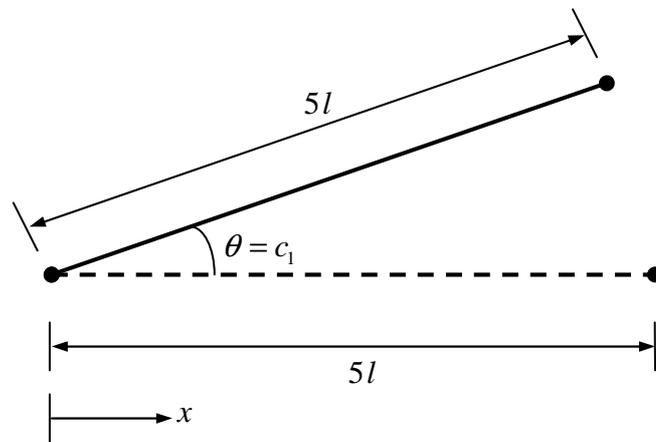
La formulation de FVMLPG pour les problèmes C^1 est évaluée en l'appliquant aux cas suivants de patch test.

1. $w(x) = c_0$, $\theta = \frac{dw}{dx} = 0$; Translation de corps rigide
2. $w(x) = c_1 x$, $\theta = c_1$; Rotation de corps rigide (4.3.2)
3. $w(x) = c_2 \frac{x^2}{2}$, $\theta = c_2 x$; Courbure constante

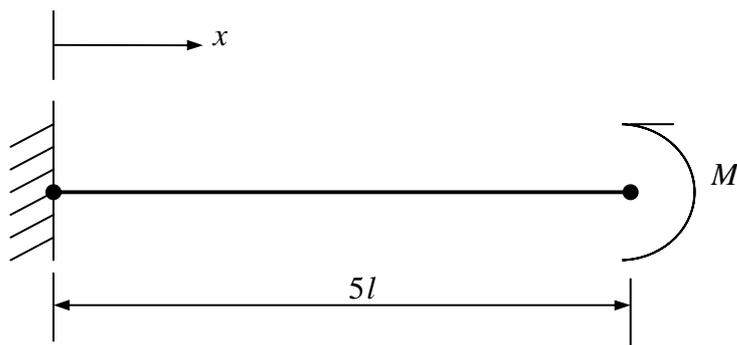
c_0 , c_1 , et c_2 sont des constants arbitraires. Le troisième patch test peut être considéré comme le cas d'une poutre encastree avec un moment, $M = EI(d^2w/dx^2) = E I c^2$, appliqué en $x = 5l$. La déformation W et la pente θ correspondant aux problèmes 1, 2, et 3 sont imposées en tant que conditions aux limites essentielles (EBCs) à $x = 0$ et $x = 5l$. Avec ces EBCs, le problème de poutre a été résolu en utilisant la méthode de FVMLPG. Si la méthode de FVMLPG récupère la solution exacte à tous les noeuds intérieurs et à chaque point arbitraire de la poutre, dans ce cas, la méthode de FVMLPG passe le patch test.



(a) Translation de corps rigide



(b) Rotation de corps rigide



(c) Courbure constante

Figure 4.3.2 Patch test

Dans ces évaluations préliminaires, les fonctions de poids $\lambda_j(x)$ utilisées pour construire les

fonctions d'essai sont

$$\lambda_j(\bar{x}) = \begin{cases} 1 - 6\left(\frac{d_j}{R_j}\right)^2 + 8\left(\frac{d_j}{R_j}\right)^3 - 3\left(\frac{d_j}{R_j}\right)^4 & \text{si } 0 \leq d_j \leq R_j \\ 0 & \text{si } d_j > R_j \end{cases} \quad (4.3.3)$$

$d_j = \|x - x_j\|$ ce $x = x - x_j$ est utilisé dans l'approximation de MLS pour construire les fonctions d'essai au voisinage local x de x_j . Ainsi, $d_j = \|x - x_j\|$ peut également s'écrire $d_j = \bar{x}$. Le (R_j/l) est $(R_j/l = 2.5\Delta x)$ pour les modèles 6 -, 11 -, et 21 nœuds et $(R_j/l = 5\Delta x)$ pour les modèles 41 -, 81 -, et 161 nœuds.

Pour un déplacement $w(x) = c_0$ et $w(x) = c_1x$, les conditions aux limites sont

$$\text{Translation: } \begin{cases} w|_{x=0} = c_0 & w|_{x=4l} = c_0 \\ \theta|_{x=0} = 0 & \theta|_{x=4l} = 0 \end{cases} \quad (4.3.4a)$$

$$\text{Rotation: } \begin{cases} w|_{x=0} = 0 & w|_{x=4l} = 5c_1l \\ \theta|_{x=0} = c_1 & \theta|_{x=4l} = c_1 \end{cases} \quad (4.3.4b)$$

Puisque les solutions exactes sont soit constantes ou linéaires, la méthode de FVMLPG développée avec une fonction de base linéaire doit reproduire les solutions exactes.

Dans le cas d'une courbure constante, $w = c_2 x^2 / 2$, le problème est modélisé avec les conditions EBCs

$$\begin{aligned} w|_{x=0} &= 0 & w|_{x=4l} &= 25/2 c_2 l^2 \\ \theta|_{x=0} &= 0 & \theta|_{x=4l} &= 5 c_2 l \end{aligned} \quad (4.3.5)$$

Bien que la solution exacte soit quadratique en x , la méthode de FVMLPG développée avec une fonction linéaire de base doit reproduire la solution exacte.

Table 4.3.1 Norme d'erreur pour les patch tests

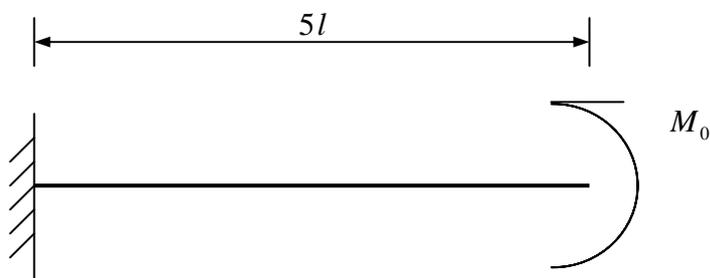
Norme d'erreur	Nombre de noeuds					
	6	11	21	41	81	161
$w(x) = c_0$	0.446e-14	0.372e-15	0.586e-13	0.217e-11	0.156e-10	0.743e-8
$w(x) = c_1 x$	0.527e-14	0.568e-15	0.714e-12	0.351e-10	0.640e-9	0.185e-7
$w(x) = c_2 \frac{x^2}{2}$	0.832e-14	0.672e-15	0.437e-11	0.558e-9	0.192e-8	0.243e-6

4.3.2 Problèmes aux valeurs limites mixtes

La méthode de FVMLPG est appliquée aux problèmes de poutre avec des conditions aux limites mixtes.

1. Poutre encastree soumise à un moment concentré à l'extrémité libre

Soit $M = M_0$ à $x = 5l$

**Figure 4.3.3** Poutre encastree

La solution exacte pour ce problème est $w = M_0 x^2 / 2EI$ et $\theta = M_0 x / EI$. La méthode FVMLPG a reproduit la solution exacte (figure 4.3.4) avec l'utilisation de la fonction de spline à 4 termes dans l'équation (4.3.3).

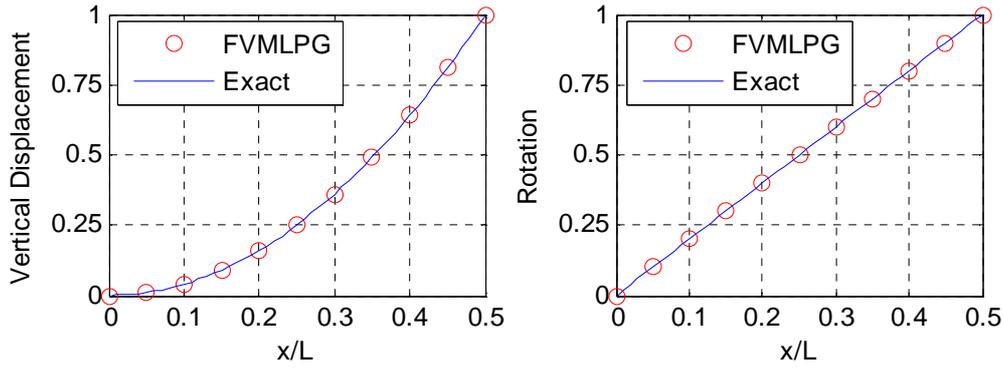


Figure 4.3.4 Déplacement et rotation normalisés de la poutre

2. Poutre simplement appuyée avec une charge uniforme

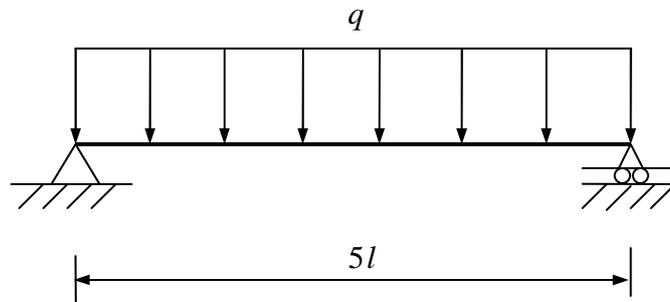


Figure 4.3.5 Poutre simplement appuyée soumise à une charge uniforme

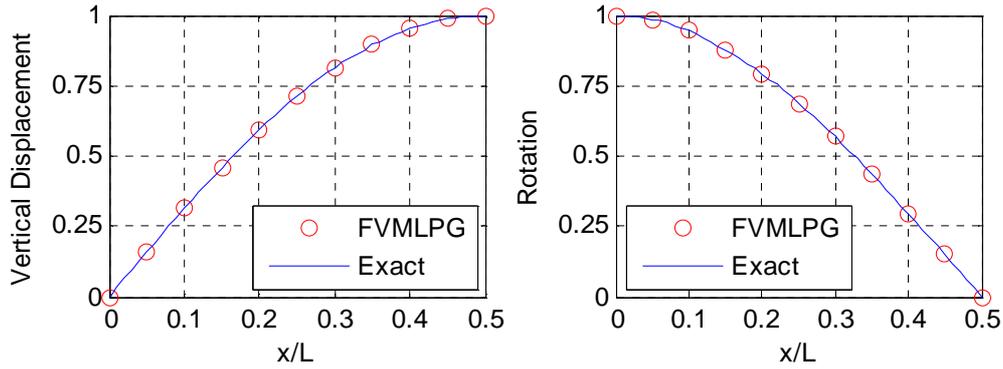
La solution exacte pour ce problème est

$$w = \frac{q}{24EI} (-2Lx^3 + x^4 + L^3x) \quad (4.3.6)$$

$$\frac{dw}{dx} = \frac{q}{24EI} (-6Lx^2 + 4x^3 + L^3)$$

Dans les équations (4.3.6) nous avons posé $L = 5l$.

Compte de la symétrie, nous avons modélisé uniquement la moitié de la poutre. La solution exacte pour ce problème est quadratique. La méthode de FVMLPG avec une fonction de base linéaire reproduit la solution exacte (figure 4.3.6).


Figure 4.3.6 Déplacement et rotation normalisés

Les normes d'erreur définies par

$$\|E_w\|_2 = \sqrt{\frac{1}{2} \sum_{k=1}^g \left[\frac{(w_{FVMLPG} - w_{exact})}{w_{exact}} \right]^2} \quad (4.3.7)$$

$$\|E_M\|_2 = \sqrt{\frac{1}{2} \sum_{k=1}^g \left[\frac{(M_{FVMLPG} - M_{exact})}{M_{exact}} \right]^2}$$

les calculs sont effectués aux points uniformément espacés de g le long de la poutre ($g = 20$). Tous les modèles donnent des solutions presque exactes. Au fur et à mesure que le nombre de nœuds augmente, la norme $\|E_w\|_2$ varie de façon marginale. Par ailleurs, la norme $\|E_M\|_2$ est du même ordre de grandeur que la norme $\|E_w\|_2$. Pour obtenir des résultats acceptables en utilisant une formulation de Galerkin, Ω_s est subdivisé en sous domaines dans lesquels, par exemple, une quadrature à 3 points de Gauss est introduite pour effectuer les intégrations.

Table 4.3.2 Norme d'erreur de $\|E\|_2$ pour une poutre simplement soumise à une charge uniforme

Norme d'erreur	Nombre de noeuds					
	6	11	21	41	81	161
$\ E_w\ _2$	0.1662e-1	0.1306e-2	0.4573e-2	0.3829e-1	0.1742e-1	0.2368e-1
$\ E_M\ _2$	0.2774e+0	0.1057e-1	0.1704e-1	0.3680e-1	0.1763e-1	0.2340e-1

3. Poutre simplement appuyée soumise à une charge concentrée au milieu

Soit une poutre simplement appuyée soumise à une charge concentrée au milieu (figure 4.3.7).

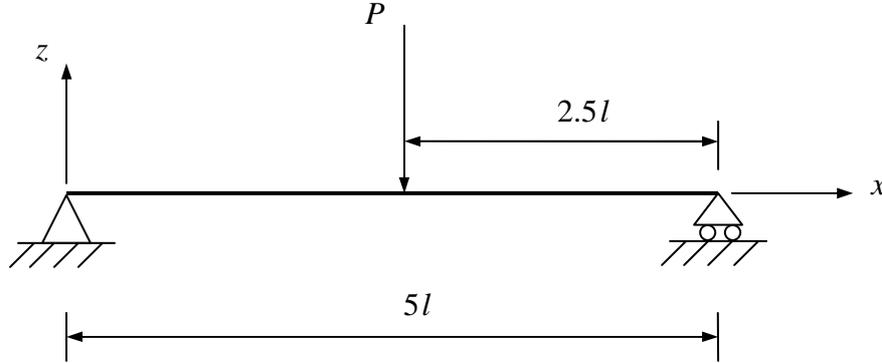


Figure 4.3.7 Poutre simplement appuyée soumise à une charge concentrée au milieu

La solution exacte pour ce problème est

$$EIw = -\frac{P}{12}x^3 + \frac{PL^2}{16}x \quad 0 \leq x \leq \frac{L}{2} \quad (4.3.8)$$

$$EI \frac{dw}{dx} = EI\theta = -\frac{P}{4}x^2 + \frac{PL^2}{16}$$

et

$$EIw = \frac{P}{12}x^3 - \frac{PL}{4}x^2 + \frac{3PL^2}{16}x - \frac{PL^3}{48} \quad \frac{L}{2} \leq x \leq L \quad (4.3.9)$$

$$EI \frac{dw}{dx} = EI\theta = \frac{P}{4}x^2 - \frac{PL}{2}x + \frac{3PL^2}{16}$$

dans l'équation (4.3.9), nous avons $L = 5l$. Le problème a été analysé de deux manières différentes. D'abord, en utilisant les conditions de symétrie (modélisation de la moitié de la poutre) puis en modélisant l'ensemble de la poutre.

Pour les deux cas, le problème est traité en utilisant une fonction de base linéaire et une intégration à 3 points de gauss. La fonction de poids de l'équation (4.3.3) a été introduite pour construire les fonctions d'essai.

$(R_0 / l) = 2.5x$ pour tous les noeuds. La solution exacte pour le déplacement sous la charge concentrée est donnée par l'équation suivante :

$$w_{\max} = \frac{PL^3}{48EI} = \frac{P(5l)^3}{48EI}. \quad (4.3.10)$$

Les solutions exactes pour les rotations aux appuis sont :

$$\theta|_{x=0} = \frac{PL^2}{16EI} = \frac{P(5l)^2}{16EI} \text{ and } \theta|_{x=5l} = -\frac{PL^2}{16EI} = -\frac{P(5l)^2}{16EI} \quad (5.3.11)$$

La figure 4.3.8 montre les conditions aux limites dans le cas de la symétrie. La méthode de FVMLPG a reproduit les solutions exactes pour tous les modèles à tous les noeuds et à tous les points intérieurs de la poutre.

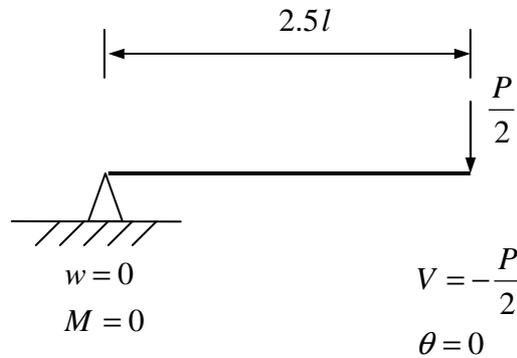


Figure 4.3.8 Poutre simplement soutenue soumise à une charge concentrée centrale (considération de la symétrie)

La poutre entière est modélisée pour étudier la validité de la méthode de FVMLPG. Les valeurs de w_{\max} et de θ_{\max} obtenues par FVMLPG pour chacun des modèles étudiés sont présentés dans le tableau 4.3.3.

Table 4.3.3 Valeurs de FVMLPG de la déformation et de la pente pour différents modèles

	Nombre de noeuds					
	6	11	21	41	81	161
$w_{(\max)}/w_{(\max)exact}$	0.9746	1.0882	1.0368	0.9982	0.9992	1.0120
$\theta_{(\max)}/\theta_{(\max)exact}$	0.3717	1.1003	1.0380	1.0012	0.9975	1.0126

Les résultats obtenus par notre approche FVMLPG et ceux fournis par les solutions exactes pour la déformation et la rotation du modèle à 21 noeuds sont montrés sur la figure 4.3.9.

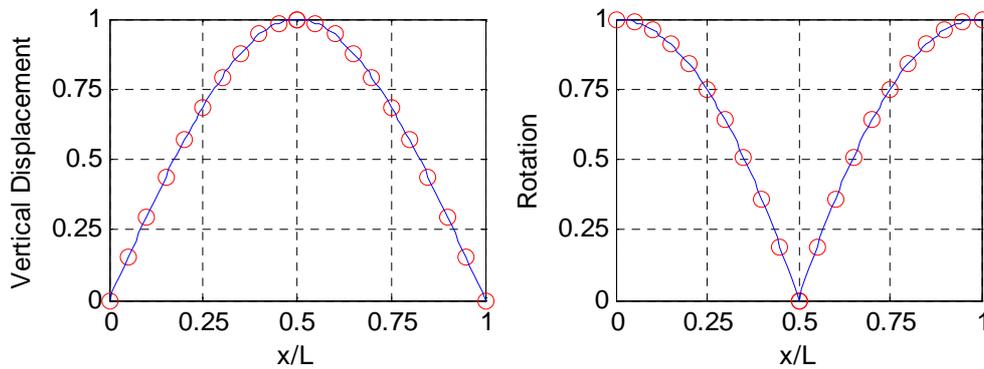


Figure 4.3.9 FVMLPG et solutions exactes pour une poutre simplement appuyée avec une charge concentrée au milieu

Ces figures et les résultats présentés dans le tableau 4.3.2 démontrent que la méthode de FVMLPG donne d'excellents résultats tant pour les déformations que pour les rotations. Ces résultats ont été obtenus sans utilisation des techniques de post-traitement. L'augmentation du nombre de noeuds de 21 à 161, n'a pas d'effet sensible sur l'exactitude. Le modèle à 21 noeuds semble toutefois donner la solution la plus précise. La méthode FVMLPG a par conséquent effacé la discontinuité provoquée par la charge concentrée centrale.

4.3.3 Poutre Continue

Soit une poutre continue à 3 travées identiques (figure 4.3.10)

La solution exacte pour ce cas est

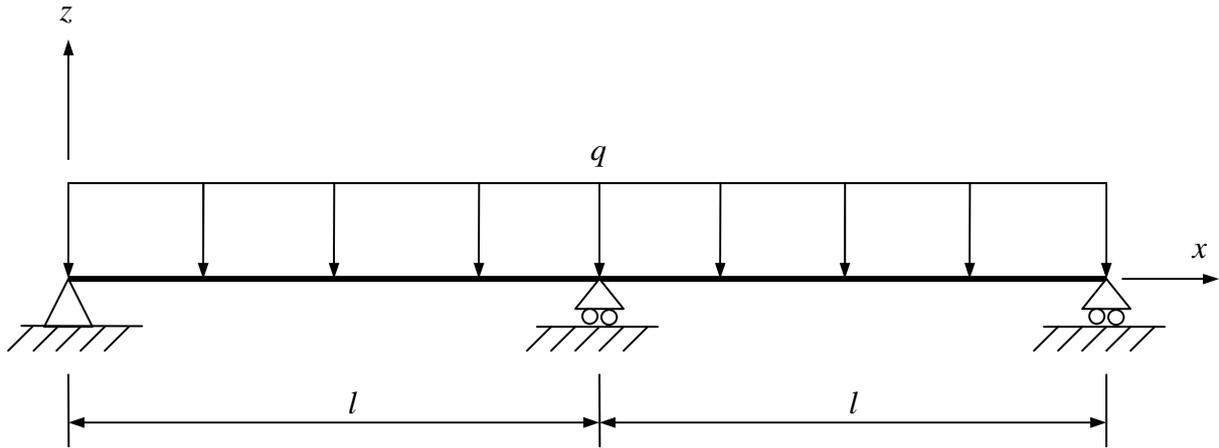


Figure 4.3.10 Poutre continue soumise à une charge uniformément répartie

$$w(x) = \frac{q}{48EI} [3lx^3 - 2x^4 - l^3x] \quad 0 \leq x \leq l \quad (4.3.12)$$

$$w(x) = \frac{q}{48EI} [13lx^3 - 2x^4 - 30l^2x^2 + 29l^3x - 10l^4] \quad l \leq x \leq 2l$$

et

$$\frac{d}{dx} w(x) = \frac{q}{48EI} [9lx^2 - 8x^3 - l^3] \quad 0 \leq x \leq l \quad (4.3.13)$$

$$\frac{d}{dx} w(x) = \frac{q}{48EI} [39lx^2 - 8x^3 - 60l^2x + 29l^3] \quad l \leq x \leq 2l$$

La méthode FVMLPG et les solutions exactes pour la déformation, la pente, et le moment obtenu à partir du modèle à 41 nœuds sont montrées sur la figure 4.3.11. La méthode FVMLPG a fournit des résultats très précis pour les variables primaires et secondaires et a bien franchit la discontinuité provoquée par l'appui intermédiaire.

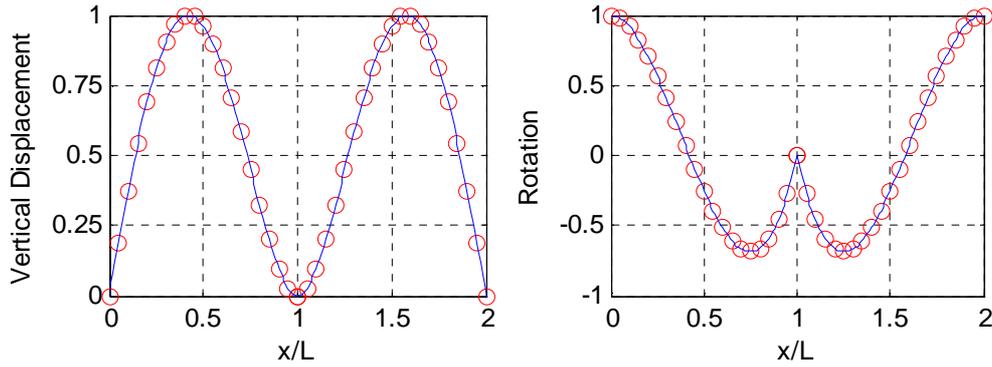


Figure 4.3.11 FVMLPG et solutions exactes d'une poutre continue soumise à une charge uniformément répartie

4.4 Remarques et conclusion

Ce chapitre présente la méthode de FVMLPG appliquée aux poutres fléchies. Sous la forme faible locale (LWF) de l'équation régissant l'équilibre, une interpolation des moindres carrés mobiles (MLS) a été utilisée pour former les approximations à la solution connue sous le nom de fonctions d'essai. Les fonctions tests, également nécessaires pour le LWF ont été choisies dans un espace différent de celui des fonctions d'essai. Ce choix des fonctions test a conduit aux matrices anti-symétriques de rigidité. Les conditions aux limites essentielles ont été imposées par une méthode de collocation. La méthode FVMLPG a été évaluée à partir plusieurs problèmes de patch test. Les résultats obtenus sont en bon accord avec la solution exacte pour le champ de variables. Une distribution de la variable secondaire a été obtenue sans utilisation des techniques de post traitement.

Chapitre 5

FVMLPG appliquée aux problèmes C^0 élastodynamiques bidimensionnels

Dans ce chapitre, la méthode « finite volume meshless local Petrov Galerkin » (FVMLPG) est étendue aux problèmes élastodynamique bidimensionnels de classe C^0 . Cette approche dérive de la forme faible locale des équations d'équilibre en utilisant le concept général de la méthode des volumes finis (VF) et de la méthode de « meshless local Petrov Galerkin » (MLPG). En incorporant les approximations des moindres carrés mobiles (MLS) pour la fonction d'essai, la forme faible locale est discrétisée puis intégrée au-dessus du volume de contrôle ou du sous domaine local pour l'analyse de vibration des structures. La méthode est testée sur plusieurs cas d'analyse de vibration des structures élastiques.

5.1 Forme faible symétrique locale (LSWF) de problèmes C^0 élastodynamiques bidimensionnels

Soit un corps élastique linéaire 2-D avec domaine Ω , et une frontière Γ . On suppose que le solide subit des déformations infinitésimales. L'équation d'équilibre s'exprime par [113] :

$$\sigma_{ij,j} + f_i - \rho a_i = 0 \quad (5.1.1)$$

σ_{ij} tenseur de contrainte correspond au champ de déplacement u_i

a_i champ d'accélération

f_i force de volume.

Les conditions aux limites sont :

$$u_i = \bar{u}_i \quad \text{sur} \quad \Gamma_u \quad (5.1.2a)$$

$$t_i = \sigma_{ij} n_j = \bar{t}_i \quad \text{sur} \quad \Gamma_t \quad (5.1.2b)$$

u_i est les déplacements sur la frontière de déplacement Γ_u et t_i est les forces appliquées sur la frontière de traction Γ_t , et n_j est la normale extérieure à la frontière Γ .

Les relations déplacements - déformations sont

$$\varepsilon_{kl} = \frac{1}{2} (u_{k,l} + u_{l,k}) \quad (5.1.3)$$

Les relations de comportement d'un solide homogène élastique linéaire isotrope sont

$$\sigma_{ij} = E_{ijkl} \varepsilon_{kl} = E_{ijkl} u_{k,l} \quad (5.1.4)$$

avec

$$E_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \quad (5.1.5)$$

λ et μ qui sont les constantes de Lamé.

Dans les approches locales Petrov Galerkin, on peut écrire une forme faible au-dessus d'un volume de contrôle ou d'un sous domaine local Ω_s , ce qui peut avoir une forme arbitraire, et contient un point \mathbf{X} en question (figure 5.1.1). Une forme faible locale de l'équation (5.1.1) au-dessus d'un sous domaine local Ω_s , peut s'écrire :

$$\int_{\Omega_s} (\sigma_{ij,j} + f_i - \rho a_i) v_i d\Omega = 0 \quad (5.1.6)$$

u_i et v_i sont les fonctions d'essai et test.

L'équation (5.1.6) peut être réécrite sous une forme faible symétrique. L'application du théorème

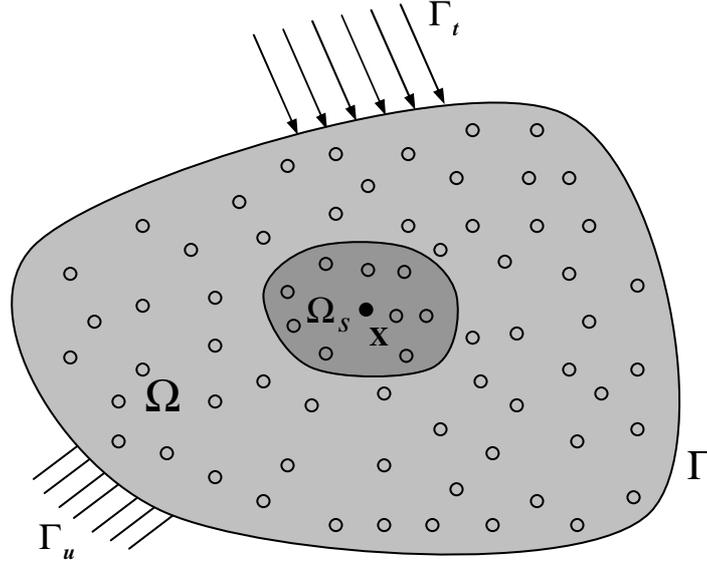


Figure 5.1.1 Sous domaine arbitraire autour d'un point

de divergence donne :

$$\int_{\partial\Omega_s} \sigma_{ij} n_j v_i d\Gamma - \int_{\Omega_s} (\sigma_{ij} v_{i,j} - f_i v_i + \rho a_i) d\Omega = 0 \quad (5.1.7)$$

Avec les conditions aux limites de traction dans l'équation (5.1.2b) on obtient

$$\int_{L_s} t_i v_i d\Gamma + \int_{\Gamma_{su}} t_i v_i d\Gamma + \int_{\Gamma_{st}} \bar{t}_i v_i d\Gamma - \int_{\Omega_s} (\sigma_{ij} v_{i,j} - f_i v_i + \rho a_i) d\Omega = 0 \quad (5.1.8)$$

Γ_{su} est une partie du $\partial\Omega_s$ de frontière de Ω_s , au-dessus duquel les conditions aux limites essentielles sont indiquées. En général, $\partial\Omega_s = \Gamma_s \cup L_s$, avec Γ_s est une partie de la frontière locale située sur la frontière globale, et L_s est l'autre partie de la frontière locale qui est à l'intérieur du domaine de solution.

$\Gamma_{su} = \Gamma_s \cap \Gamma_u$ est l'intersection entre la frontière locale $\partial\Omega_s$ et la frontière globale de déplacement Γ_u ;

$\Gamma_{st} = \Gamma_s \cap \Gamma_t$ est une partie de la frontière où les conditions aux limites naturelles sont spécifiées.

Par conséquent, une forme faible symétrique locale (LSWF) en élasticité linéaire peut s'écrire :

$$\int_{\Omega_s} \sigma_{ij} v_{i,j} d\Omega - \int_{L_s} t_i v_i d\Gamma - \int_{\Gamma_{su}} t_i v_i d\Gamma + \int_{\Omega_s} \rho a_i v_i d\Omega = \int_{\Gamma_{st}} \bar{t}_i v_i d\Gamma + \int_{\Omega_s} f_i v_i d\Omega \quad (5.1.9)$$

En utilisant la fonction de Heaviside

$$v_i = \begin{cases} 1 & \mathbf{x} \in \Omega_s \\ 0 & \mathbf{x} \notin \Omega_s \end{cases} \quad (5.1.10)$$

Comme la fonction d'essai dans le LSWF de l'équation (5.1.9), on obtient

$$-\int_{L_s} \bar{t}_i d\Gamma - \int_{\Gamma_{su}} t_i d\Gamma + \int_{\Omega_s} \rho a_i d\Omega = \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega. \quad (5.1.11)$$

En introduisant les relations constitutives d'un solide homogène élastique linéaire isotrope dans l'équation (5.1.11), on obtient :

$$-\int_{L_s} E_{ijkl} \varepsilon_{kl} n_j d\Gamma - \int_{\Gamma_{su}} E_{ijkl} \varepsilon_{kl} n_j d\Gamma + \int_{\Omega_s} \rho a_i d\Omega = \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega \quad (5.1.12)$$

Dans l'équation (5.1.12), on remplace les déplacements et les déformations par les équations d'approximation suivantes :

$$u_i(\mathbf{x}, t) = \sum_{K=1}^n \Phi^{(K)}(\mathbf{x}) u_i^{(K)}(t) \quad (5.1.13)$$

$$\varepsilon_{kl}(\mathbf{x}, t) = \sum_{K=1}^n \Phi^{(K)}(\mathbf{x}) \varepsilon_{kl}^{(K)}(t) \quad (5.1.14)$$

$\Phi^{(K)}(\mathbf{x})$ est la fonction de forme, $u_i^{(K)}$ et $\varepsilon_{kl}^{(K)}$ sont respectivement le déplacement et la déformation dans le nœud I . Par substitution des équations (5.1.13) et (5.1.14) dans l'équation (5.1.12), on obtient l'équation discrétisée suivante :

$$\begin{aligned} & -\sum_{K=1}^n \left[\int_{L_s} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] \varepsilon_{kl}^{(K)}(t) - \sum_{K=1}^n \left[\int_{\Gamma_{su}} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] \varepsilon_{kl}^{(K)}(t) \\ & + \sum_{K=1}^n \left[\int_{\Omega_s} \Phi^{(K)}(\mathbf{x}) \rho d\Omega \right] a_i^{(K)}(t) = \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega \end{aligned} \quad (5.1.15)$$

L'équation (5.1.15) représente la formulation de type finite volume meshless local Petrov Galerkin (FVMLPG) de l'équation d'équilibre dans le problème bidimensionnel de l'équation (5.1.1).

Dans l'analyse modale, a_i peut s'écrire:

$$a_i = \ddot{u}_i(x, t) = -\omega_i^2 u_i \sin(\omega_i t + \psi_i) = -\omega_i^2 u_i(x, t) \quad (5.1.16)$$

u_i est l'amplitude du déplacement

ω_i est la fréquence.

Dans l'analyse modale \bar{t}_i sur Γ_i est égale zéro et on suppose que f_i s'annule. En substituant l'équation (5.1.16) dans l'équation (5.1.15) on obtient :

$$\begin{aligned} & -\sum_{K=1}^n \left[\int_{L_s} \Phi^{(K)}(x) E_{ijkl} n_j d\Gamma \right] \varepsilon_{kl}^{(K)}(t) - \sum_{K=1}^n \left[\int_{\Gamma_{su}} \Phi^{(K)}(x) E_{ijkl} n_j d\Gamma \right] \varepsilon_{kl}^{(K)}(t) \\ & = \sum_{K=1}^n \left[\int_{\Omega_s} \Phi^{(K)}(x) \rho \omega_i^2 d\Omega \right] u_i^{(j)}(t) \end{aligned} \quad (5.1.17)$$

On constate dans cette équation qu'aucune dérivée de fonctions de forme n'est impliquée dans les intégrales locales. Il est bien connu que l'approximation sans maillage n'est pas efficace en calculant de telles dérivées partout dans le domaine, particulièrement quand l'approximation de MLS est utilisée. Ainsi, l'efficacité de la méthode actuelle est améliorée et dépasse les méthodes de MLPG traditionnelles.

La méthode standard de collocation peut être appliquée pour imposer l'équation (5.1.3) seulement à chaque point nodal $\mathbf{x}^{(l)}$. Ainsi, les variables nodales de déformation sont exprimées en termes de variables nodales de déplacement

$$\varepsilon_{kl}(\mathbf{x}^{(l)}) = \frac{1}{2} \left[u_{k,l}(\mathbf{x}^{(l)}) + u_{l,k}(\mathbf{x}^{(l)}) \right] \quad (5.1.18)$$

Avec l'approximation en déplacement dans l'équation (5.1.13), les deux ensembles de variables nodales peuvent être transformés par une matrice algébrique linéaire

$$\boldsymbol{\varepsilon}_{kl}^I = \mathbf{H}_{klm}^{IJ} u_m^J \quad (5.1.19)$$

la matrice H de transformation est une matrice bande.

Après la transformation, le nombre d'équations du système est alors réduit au même nombre que les variables nodales en déplacement. L'équation (5.1.19) est substituée dans le LWF (5.1.17) pour tous les nœuds, ce qui conduit aux équations discrètes suivantes

$$\begin{aligned} -\sum_{K=1}^n \left[\int_{L_s} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] \mathbf{H}_{klm}^{(K)(J)} u_m^{(J)}(t) - \sum_{K=1}^n \left[\int_{\Gamma_{su}} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] \mathbf{H}_{klm}^{(K)(J)} u_m^{(J)}(t) \\ = \sum_{K=1}^n \left[\int_{\Omega_s} \Phi^{(K)}(\mathbf{x}) \rho \omega_i^2 d\Omega \right] u_i^{(K)}(t) \end{aligned} \quad (5.1.20)$$

L'équation (5.1.20) peut s'écrire sous la forme suivante :

$$\mathbf{K}u - \omega^2 \mathbf{M}u = 0 \quad (5.1.21)$$

la matrice de « rigidité » \mathbf{K} et la matrice de « masse » \mathbf{M} sont définies par

$$\mathbf{K}_{IJ} = \left[\int_{L_s} \Phi^{(I)}(\mathbf{x}) E_{ijkl} n_j d\Gamma - \int_{\Gamma_{su}} \Phi^{(I)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] \mathbf{H}_{klm}^{(I)(J)} \quad (5.1.22)$$

$$\mathbf{M}_{IJ} = \int_{\Omega_s} \rho \Phi^{(I)}(\mathbf{x}) d\Omega. \quad (5.1.23)$$

Pour des analyses de vibration libres, l'équation (5.1.21) peut également s'écrire comme suit :

$$(\mathbf{K} - \omega^2 \mathbf{M})\mathbf{q} = 0 \quad (5.1.24)$$

\mathbf{q} est le vecteur propre. L'équation (5.1.24) est la formulation de FVMLPG pour l'analyse de vibration libre. Afin de déterminer les fréquences, ω , et les modes de vibration libres, il est nécessaire de résoudre l'équation linéaire aux valeurs propres.

Dans l'analyse de vibrations forcées, l'équation (5.1.15) peut être développée en utilisant l'équation (5.1.19) sous la forme suivante :

$$\begin{aligned}
 & - \sum_{K=1}^n \left[\int_{L_s} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] H_{klm}^{(K)(J)} u_m^{(J)}(t) - \sum_{K=1}^n \left[\int_{\Gamma_{su}} \Phi^{(K)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] H_{klm}^{(K)(J)} u_m^{(J)}(t) \\
 & + \sum_{K=1}^n \left[\int_{\Omega_s} \Phi^{(K)}(\mathbf{x}) \rho d\Omega \right] a_i^{(K)}(t) = \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega
 \end{aligned} \tag{5.1.25}$$

Dans l'équation (5.1.25), on identifie

- la matrice de « rigidité » K ,
- la matrice de « masse » M
- la matrice de « force » f

avec

$$K_{IJ} = \left[\int_{L_s} \Phi^{(I)}(\mathbf{x}) E_{ijkl} n_j d\Gamma - \int_{\Gamma_{su}} \Phi^{(I)}(\mathbf{x}) E_{ijkl} n_j d\Gamma \right] H_{klm}^{(I)(J)} \tag{5.1.26}$$

$$M_{IJ} = \int_{\Omega_s} \rho \Phi^{(I)}(\mathbf{x}) d\Omega. \tag{5.1.27}$$

$$f_I = \int_{\Gamma_{st}} \bar{t}_i d\Gamma + \int_{\Omega_s} f_i d\Omega \tag{5.1.28}$$

5.2 Imposition des conditions aux limites

Il est bien connu que l'approximation de type MLS donne les fonctions de forme basées sur les valeurs nodales virtuelles, qui ne possèdent pas la propriété du symbole de Kronecker. Les champs sur la frontière, où des conditions aux limites essentielles sont imposées dépendent non seulement des valeurs nodales situées sur la frontière mais également de celles situées à l'intérieur du domaine de solution. Ils ne sont également pas linéaires entre les noeuds de frontière. Du point de vue numérique, si les conditions aux limites sont imposées le long de la totalité de la frontière essentielle, elles engendrent trop de contraintes aux noeuds internes. Ces conditions rendent la structure beaucoup plus raide. Dans la présente étude, la méthode standard de collocation est appliquée pour l'analyse modale. Pour les noeuds qui appartiennent à la frontière essentielle, c.-à-d. $u_i^{(I)} \in \Gamma_{su}$, on peut introduire la méthode standard de collocation,

$$\alpha u_i(\mathbf{x}^{(I)}) = \alpha \bar{u}_i(\mathbf{x}^{(I)}) \tag{5.2.1}$$

pour imposer les conditions aux limites essentielles.

On impose les accélérations « \hat{a} » égales à zéro pour tous les noeuds appartenant à la frontière essentielle :

« \hat{a} » : valeur de l'accélération au nœud

$$a_i(x^{(I)}) = 0 \quad (5.2.2)$$

Soit sous forme matricielle

$$\mathbf{Q}^T \cdot \hat{\mathbf{a}} = 0 \quad (5.2.3)$$

Après l'orthogonalisation de la matrice \mathbf{Q} , l'équation (5.2.3) peut s'écrire sous une forme équivalente

$$\mathbf{G}^T \cdot \hat{\mathbf{a}} = 0 \quad (5.2.4)$$

la matrice \mathbf{G} satisfait la condition suivante :

$$\mathbf{G}^T \cdot \mathbf{G} = \mathbf{I} \quad (5.2.5)$$

\mathbf{I} est la matrice d'identité.

Sous forme matricielle, la forme faible locale de l'équation (5.1.25) peut être intégrée numériquement au-dessus de chaque sous domaine local

$$\mathbf{M} \cdot \hat{\mathbf{a}} + \mathbf{K} \cdot \hat{\mathbf{u}} = \hat{\mathbf{f}} \quad (5.2.6)$$

la matrice de masse, \mathbf{M} , est diagonale quand la masse du nœud est localisée. Avec une intégration temporelle explicite, les accélérations sont obtenues à partir de l'équation (5.2.6),

$$\hat{\mathbf{a}}' = \mathbf{M}^{-1} \cdot (\hat{\mathbf{f}} - \mathbf{K} \cdot \hat{\mathbf{u}}) \quad (5.2.7)$$

Avec la prise en compte des conditions aux limites essentielles dans l'équation (5.2.4), les accélérations peuvent être corrigées :

$$\hat{\mathbf{a}} = \hat{\mathbf{a}}' - \mathbf{G} \cdot \mathbf{G}^T \cdot \hat{\mathbf{a}}' \quad (5.2.8)$$

5.3 Intégration temporelle

La méthode β de Newmark [114], bien connue, est utilisée pour intégrer les équations d'équilibres. L'équation (5.2.8) permet de déterminer les accélérations, les déplacements ainsi que les vitesses à partir de la méthode standard β de Newmark,

$$\begin{aligned} \mathbf{u}^{t+\Delta t} &= \mathbf{u}^t + \Delta t \mathbf{v}^t + \frac{\Delta t^2}{2} \left[(1-2\beta) \mathbf{a}^t + 2\beta \mathbf{a}^{t+\Delta t} \right] \\ \mathbf{v}_c^{t+\Delta t} &= \mathbf{v}^t + \Delta t \left[(1-\gamma) \mathbf{a}^t + \gamma \mathbf{a}^{t+\Delta t} \right] \end{aligned} \quad (5.3.1)$$

Pour le système sans amortissement, cette méthode est inconditionnellement stable si

$$2\beta \geq \gamma \geq \frac{1}{2} \quad (5.3.2)$$

et conditionnellement stable si

$$\gamma \geq \frac{1}{2}, \beta \leq \frac{1}{2} \text{ et } \Delta t \leq \frac{1}{\omega_{\max} \sqrt{\gamma/2 - \beta}} \quad (5.3.3)$$

ω_{\max} est la fréquence maximale dans le système structural.

Après l'introduction des conditions initiales, les intégrations temporelles pour chaque incrément de temps peuvent être établies de la façon suivante :

Étape 1 : calcul des vitesses et déplacements

$$\begin{aligned}\hat{\mathbf{u}}_c^{t+\Delta t} &= \hat{\mathbf{u}}^t + \Delta t \hat{\mathbf{v}}^t + \frac{\Delta t^2}{2} (1 - 2\beta) \hat{\mathbf{a}}^t \\ \hat{\mathbf{v}}_c^{t+\Delta t} &= \hat{\mathbf{v}}^t + \Delta t (1 - \gamma) \hat{\mathbf{a}}^t\end{aligned}\quad (5.3.4)$$

Étape 2 : calcul des accélérations

$$\begin{aligned}\hat{\mathbf{a}}_{c1}^{t+\Delta t} &= \mathbf{M}^{-1} \cdot (\hat{\mathbf{f}}^{t+\Delta t} - \mathbf{K} \cdot \hat{\mathbf{u}}_c^{t+\Delta t}) \\ \hat{\mathbf{a}}_{c2}^{t+\Delta t} &= \hat{\mathbf{a}}_{c1}^{t+\Delta t} - \mathbf{G} \cdot \mathbf{G}^T \cdot \hat{\mathbf{a}}_{c1}^{t+\Delta t}\end{aligned}\quad (5.3.5)$$

Étape 3 : correction des vitesses et déplacements

$$\begin{aligned}\hat{\mathbf{u}}^{t+\Delta t} &= \hat{\mathbf{u}}_c^{t+\Delta t} + \Delta t^2 \beta \hat{\mathbf{a}}_{c2}^{t+\Delta t} \\ \hat{\mathbf{v}}^{t+\Delta t} &= \hat{\mathbf{v}}_c^{t+\Delta t} + \Delta t \gamma \hat{\mathbf{a}}_{c2}^{t+\Delta t}\end{aligned}\quad (5.3.6)$$

Étape 4 : correction des accélérations

$$\begin{aligned}\hat{\mathbf{a}}_{c3}^{t+\Delta t} &= \mathbf{M}^{-1} \cdot (\hat{\mathbf{f}}^{t+\Delta t} - \mathbf{K} \cdot \hat{\mathbf{u}}^{t+\Delta t}) \\ \hat{\mathbf{a}}^{t+\Delta t} &= \hat{\mathbf{a}}_{c3}^{t+\Delta t} - \mathbf{G} \cdot \mathbf{G}^T \cdot \hat{\mathbf{a}}_{c3}^{t+\Delta t}\end{aligned}\quad (5.3.7)$$

5.4 Exemples numériques

Cette méthode est évaluée par les exemples suivants.

Exemple 1 Vibration libre d'une poutre encastree

On considère la poutre console représentée par la figure 5.3.1. Les caractères de la poutre sont

L : longueur,

c : demi hauteur,

E : module d'élasticité

ν : coefficient de Poisson

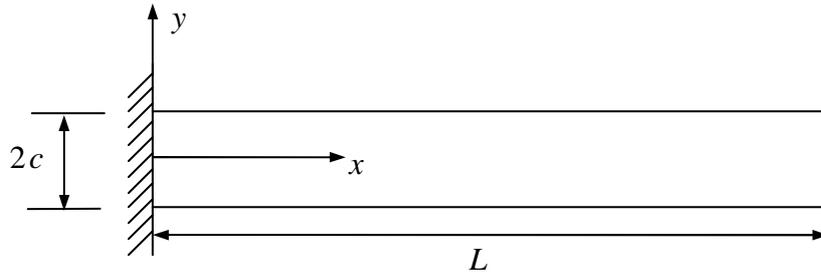


Figure 5.3.1 Poutre encastrée

Cet exemple est résolu en considérant l'état de contraintes planes

$$L = 24, c = 2, E = 1, \nu = 0.25.$$

Avec les conditions aux limites en déplacement suivantes

$$u_x(0,0) = u_y(0,0) = 0 \text{ et } u_x(0,c) = u_x(0,-c) = 0$$

On considère des configurations nodales uniformes et régulières avec des distances nodales d de 2.0, 1.0 et 0.5. Le nombre adimensionnel d est la distance entre deux nœuds voisins suivant les directions x ou y . Il est déterminé par division de la longueur de la poutre par le nombre de noeuds dans la direction de la longueur. Les nombres de noeuds sont respectivement 39, 125 et 441.

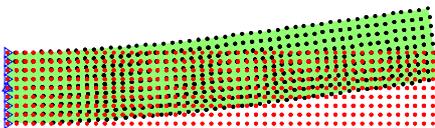


Figure 5.3.2 Mode 1

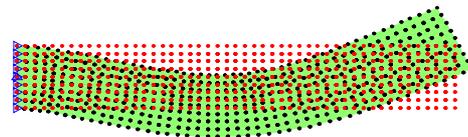


Figure 5.3.3 Mode 2

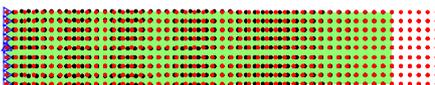


Figure 5.3.4 Mode 3

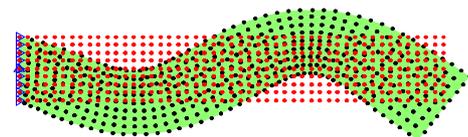


Figure 5.3.5 Mode 4

Erreur	FVMLPG %	ABAQUS %
ω_1	0.015	0.6231
ω_2	0.0087	0.6225
ω_3	0.0053	0.6088
ω_4	0.0034	0.5836

Table 5.3.1 Erreur de fréquence

Exemple 2 Poutre soumise à une excitation à son extrémité libre

Soit une poutre console encadrée à une extrémité et libre à l'autre, soumise à la fonction d'excitation $f(t)$ (figure 5.4.6)

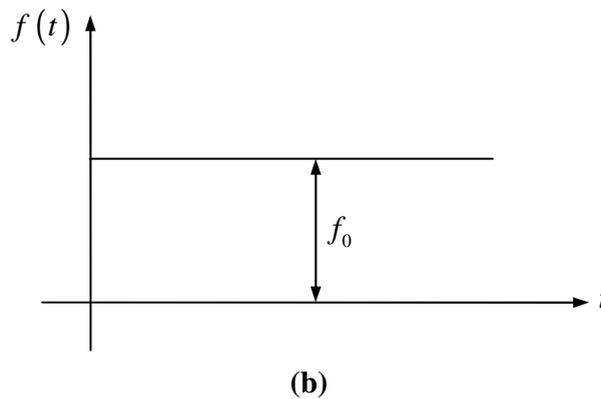
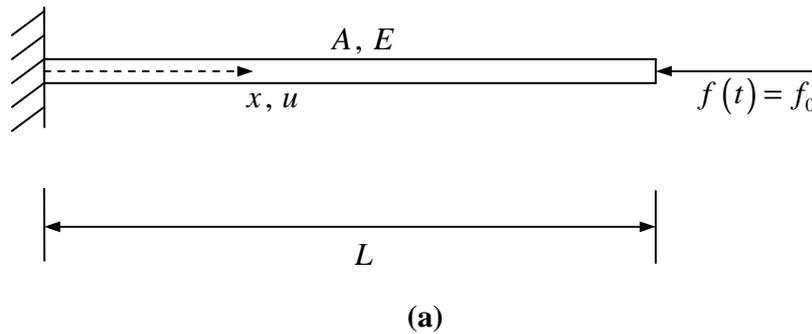


Figure 5.4.6 Poutre soumise à une excitation à son extrémité libre

(a) configuration géométrique

(b) fonction d'excitation

$$u(x, t) = \frac{8f_0}{\pi^2} \frac{L}{AE} \sum_{n=1}^{\infty} \left[(\pm) \frac{1 - \cos \omega_n t}{(2n-1)^2} \sin \frac{2n-1}{2} \frac{\pi x}{L} \right] \tag{5.4.1}$$

$$F(x, t) = \frac{4f_0}{\pi} \sum_{n=1}^{\infty} \left[(\pm) \frac{1 - \cos \omega_n t}{2n-1} \cos \frac{2n-1}{2} \frac{\pi x}{L} \right] \quad (5.4.2)$$

Pour la résolution du problème par la méthode de FVMLPG, nous avons considéré les caractéristiques suivantes :

$$L = 24, E = 1, \nu = 0.25, \rho = 1, A = 1$$

$f(t) = H(t)$ où $H(t)$ est la fonction d'excitation de Heaviside.

La barre est divisée en 49 noeuds équidistants. La méthode « β » de Newmark avec l'incrément de temps Δt , égale $1.0 \mu s$ est appliquée. La fréquence maximum ω_{\max} calculée est égale à 0.0654 hertz. La figure 5.4.7 montre le déplacement ainsi que l'effort axial le long de la poutre. Les résultats montrent une bonne concordance entre notre approche FVMLPG et la solution exacte.

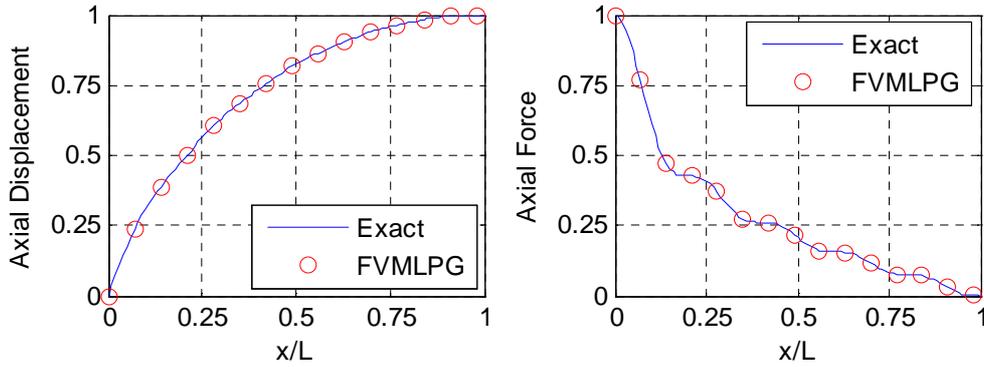


Figure 5.4.7 Effort et déplacement axial de la poutre

La figure 5.4.8 montre l'évolution de l'effort et du déplacement axial en fonction du temps au milieu de la poutre ($x = L / 2$). On constate que les résultats numériques se superposent avec ceux fournis par la solution exacte.

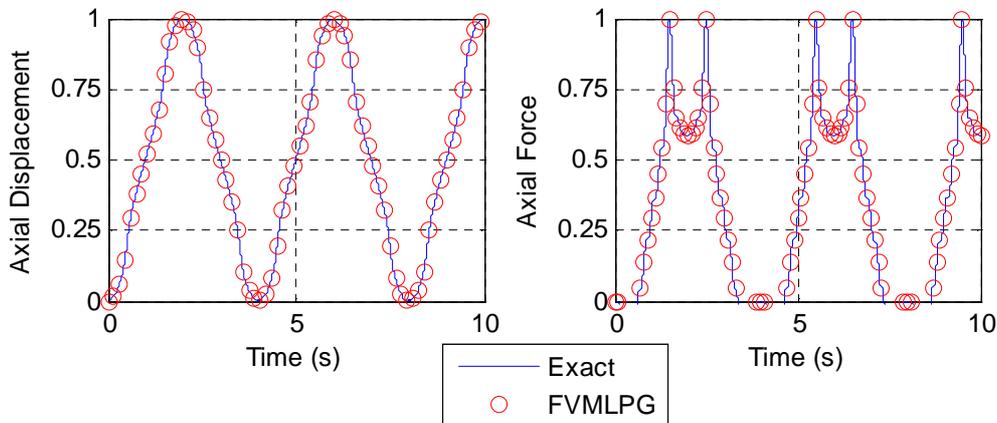


Figure 5.4.8 Effort et déplacement axial en fonction du temps

Exemple 3 Poutre console soumise à une charge impulsive

On considère une poutre de longueur $L = 10$ m et de largeur $2c = 1$ m, (figure 5.4.9). Cette poutre est encastree à une extrémité et libre à l'autre.

$E = 100$ t/m², $\nu = 0$, $\rho = 10$ kg/m³, vitesse d'impulsion P : $C_p = 100$ m/sec.

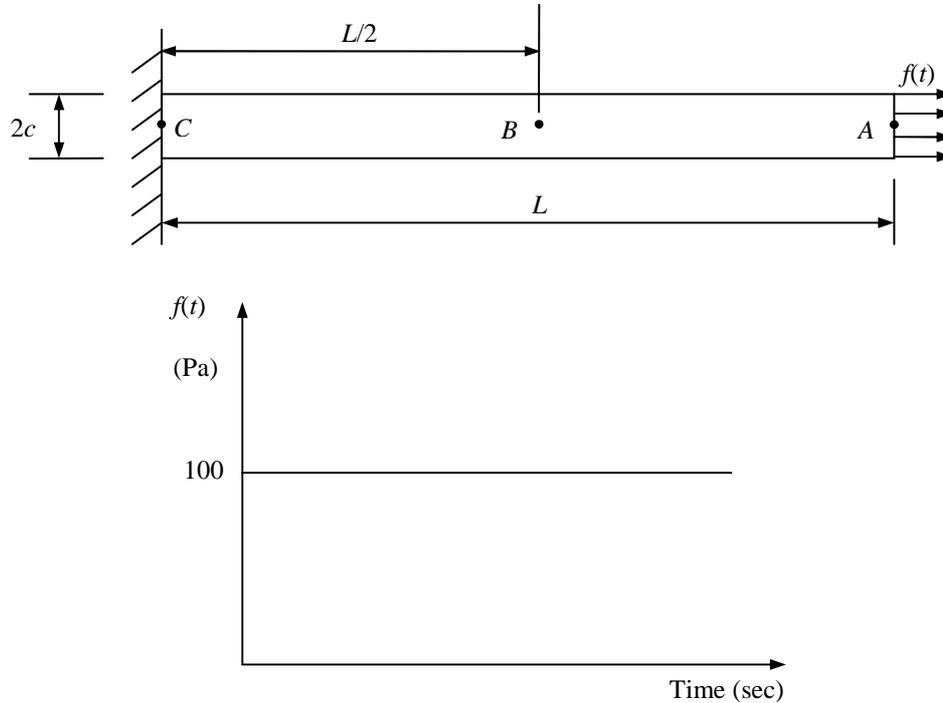


Figure 5.4.9 Poutre sous charge impulsive

En contrainte plane, le domaine de l'espace est discrétisé uniformément avec 441 nœuds (48×8). Chaque sous domaine circulaire est intégré par le point de quadrature de gauss 3×6 dans la direction de r et de θ . L'incrément de temps choisi pour l'intégration est $\Delta t = 0.4/C_p = 0.004$ sec. Les réponses en déplacement et en effort en fonction du temps sont analysées au point B de cette poutre (figures 5.4.10 et 5.4.11.)

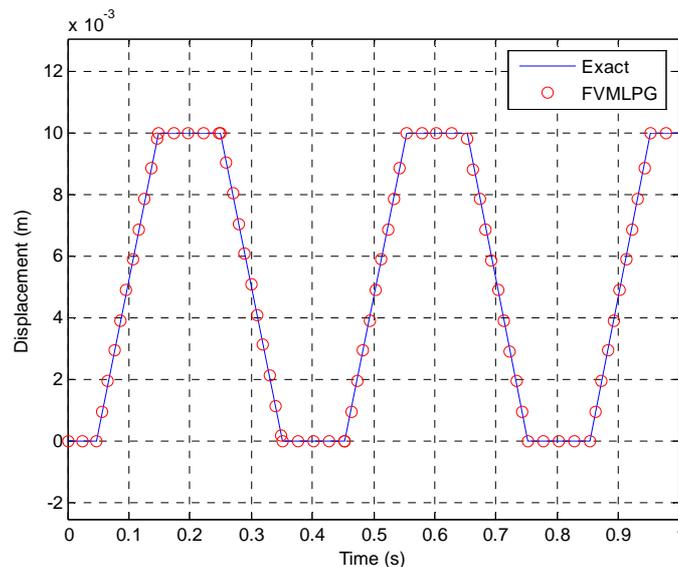


Figure 5.4.10 Déplacement au point B en fonction du temps

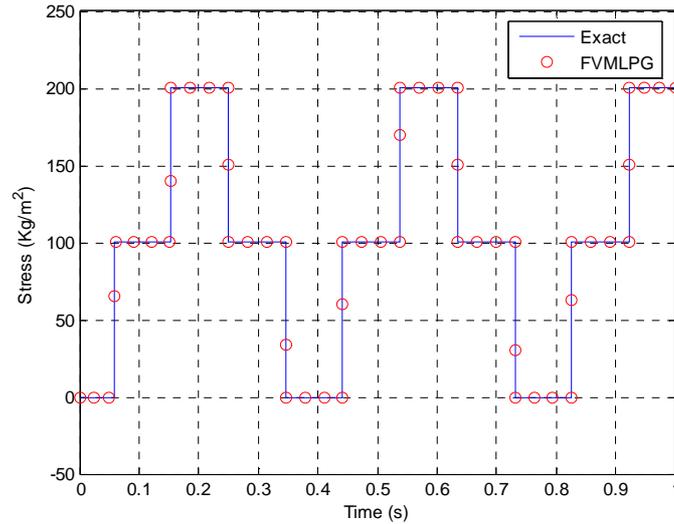


Figure 5.4.11 Effort au point B en fonction du temps

Les réponses en contraintes et déplacements sont obtenus au point B par la méthode de FVMLPG en considérant trois rayons différents de sous-domaine, 0.05 m, 0.1 m, et 0.2 m. On constate que Les résultats numériques obtenus par la méthode FVMLPG s'approchent de la solution exacte pour les réponses en déplacement quand le rayon du sous domaine augmente de 0.05 m jusqu'à 0.2 m. Cependant, le temps d'exécution est un peu élevé pour un rayon du sous domaine dépassant 0.2 m.

5.5 Remarques et conclusion

Dans ce chapitre, la méthode « finite volume meshless local Petrov Galerkin » (FVMLPG) est étendue aux problèmes élastodynamique bidimensionnels de classe C^0 . Cette approche dérive de la forme faible locale des équations d'équilibre en utilisant le concept général de la méthode des volumes finis (VF) et de la méthode de « meshless local Petrov Galerkin » (MLPG). En utilisant l'approche de Newmark pour l'intégration temporelle, un traitement numérique est développé pour l'application des conditions aux limites cinématiques. La méthode est testée sur plusieurs cas d'analyse de vibration des structures élastiques. Les résultats démontrent l'efficacité et l'exactitude de cette résolution.

Chapitre 6

FVMLPG appliquée aux instabilités de poutres minces

La méthode « finite volume meshless local Petrov Galerkin » (FVMLPG) est développée pour l'étude de l'instabilité d'une barre soumise à un effort normal axial.

6.1 Équations d'équilibres

On considère une barre soumise à un effort normal de compression (figure 6.1.1). On suppose que la barre au repos est parfaitement rectiligne et que l'effort de compression est appliqué suivant la ligne moyenne sans excentricité. A l'état déformé de la poutre, on détermine les sollicitations d'effort normal et de flexion due à la déformée de la ligne moyenne.

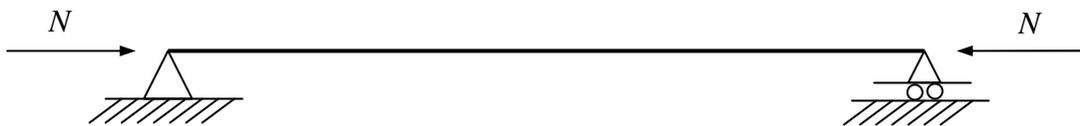


Figure 6.1.1 Barre sous effort axial de compression (état initial non déformé)

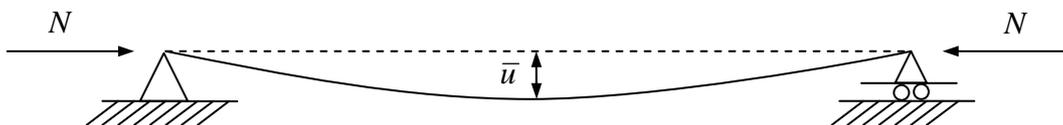


Figure 6.1.2 Etat déformé de la barre

Le moment de flexion s'exprime par :

$$M^N(x) = -N \cdot \bar{u}(x) \quad (6.1.1)$$

L'équation d'équilibre de la barre peut s'écrire :

$$(M + M^N)'' = 0. \quad (6.1.2)$$

Soit en introduisant le déplacement \bar{u} , l'équation (6.1.2) s'écrit :

$$EI \frac{d^4 \bar{u}}{dx^4} + N \frac{d^2 \bar{u}}{dx^2} = 0 \quad (6.1.3)$$

Sous forme adimensionnelle, l'équation (6.1.3) devient

$$\frac{d^4 u}{d\xi^4} + N \frac{d^2 u}{d\xi^2} = 0. \quad (6.1.4)$$

Pour une charge axiale donnée N , on peut trouver une solution particulière u_0 qui satisfasse l'équation 6.1.4

$$\frac{d^4 u_0}{d\xi^4} + N \frac{d^2 u_0}{d\xi^2} = 0. \quad (6.1.5)$$

$u = 0$ est une solution de l'équation (6.1.4). Il peut y avoir des valeurs caractéristiques (valeurs propres) de N , pour lesquelles les solutions non triviales pour le déplacement transversal u peuvent être possibles. Ces valeurs propres de N sont les charges axiales de bifurcation de flambage de la barre.

Les solutions génériques de l'équation (6.1.4) sont

$$\psi^i = \sin 2i\pi\xi, \quad i = 1, 2, \dots \quad (6.1.6)$$

En substituant l'équation (6.1.6) dans l'équation (6.1.4), on obtient les équations caractéristiques pour N

$$(2i\pi)^4 - N(2i\pi)^2 = 0 \quad (6.1.7)$$

qui constituent les charges critiques (valeurs propres pour N)

$$N^i = (2i\pi)^2, \quad i = 1, 2, \dots \quad (6.1.8)$$

L'objectif dans ce chapitre est d'appliquer notre approche « FVMLPG » pour résoudre l'équation homogène (6.1.4). La forme faible locale de cette équation peut être obtenue en introduisant une fonction test dans cette équation et en l'intégrant dans le sous domaine

$$\int_{\Omega_s} \left(\frac{d^4 u}{d\xi^4} + N \frac{d^2 u}{d\xi^2} \right) v d\xi = 0 \quad (6.1.9)$$

Comme mentionné dans le chapitre 4, il est nécessaire de réduire l'ordre de l'équation en termes de quatre équations de premier ordre. Le premier terme peut être écrit de la façon suivante :

$$u = \phi_1 \quad (6.1.10)$$

$$\frac{d\phi_1}{dx} = \phi_2 \quad (6.1.11)$$

$$\frac{d\phi_2}{dx} = \phi_3 \quad (6.1.12)$$

$$\frac{d\phi_3}{dx} = \phi_4 \quad (6.1.13)$$

$$\frac{d\phi_4}{dx} = 1 \quad (6.1.14)$$

en notation matriciel

$$\begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{Bmatrix}' + \begin{Bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{Bmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (6.1.15)$$

ou

$$A_{ij}\phi_j' + B_{ij}\phi_j = 0 \quad j = 1, \dots, 4 \quad (6.1.16)$$

Chacun terme Φ_j est interpolé par l'approximation MLS comme suit

$$\phi_j^h(x) = \sum_{l=1}^N \phi_j^l(x) \hat{\phi}_j^l; \quad x \in \Omega_s \quad (6.1.17)$$

avec $j = 1, \dots, 4$, comme dans l'équation (6.1.16).

En utilisant une fonction test qui est une fonction de Heaviside dans chaque Ω_s [c.-à-d., $x_{k-1} \leq x \leq x_k$, qui peut recouvrir des sous domaines comme dans la méthode volume finis], le système d'équations linéaire simple suivant est obtenu pour les valeurs nodales de $\hat{\phi}_j^l$

$$\int_{x_{k-1}}^{x_k} [A_{ij}\phi_j' + B_{ij}\phi_j] dx = 0 \quad (6.1.18)$$

L'équation (6.1.18) peut également s'écrire :

$$\int_{x_{k-1}}^{x_k} B_{ij}\phi_j dx + [A_{ij}\phi_j]_{x_{k-1}}^{x_k} = 0 \quad (6.1.19)$$

L'équation (6.1.19) comporte seulement les interpolations de type MLS pour chaque Φ_j , ($j = 1, \dots, 4$), mais non leurs dérivées. Cette équation peut également s'écrire sous la forme discrète suivante :

$$\sum_{l=1}^N \left(\int_{x_{k-1}}^{x_k} B_{ij}\phi_j^l(x) dx + [A_{ij}\phi_j^l(x)]_{x_{k-1}}^{x_k} \right) \hat{\phi}_j^l = 0 \quad (6.1.20)$$

Le système d'équations est obtenu pour le premier terme,

$$\frac{d^4 u}{d\xi^4} = K_{ij} a_j \quad (6.1.21)$$

K_{ij} est défini par

$$K_{ij} = \sum_{l=1}^N \left(\int_{x_{k-1}}^{x_k} B_{ij} \phi_j'(x) dx + \left[A_{ij} \phi_j'(x) \right]_{x_{k-1}}^{x_k} \right) \quad (6.1.22)$$

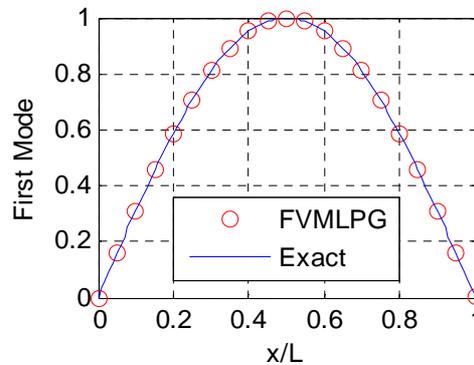
avec la même approche, nous pouvons ainsi obtenir les équations semblables pour le deuxième terme,

$$N \frac{d^2 u}{d\xi^2} = NC_{ij} a_j \quad (6.1.23)$$

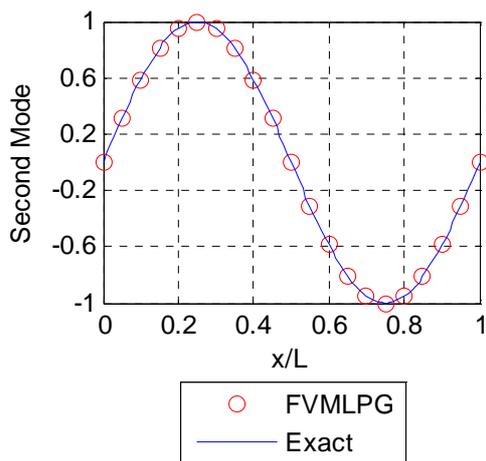
En ajoutant les équations (6.1.21) et (6.1.23), on peut déduire les équations du système pour les problèmes de flambage linéaire,

$$(K_{ij} + NC_{ij}) a_j = 0 \quad (6.1.24)$$

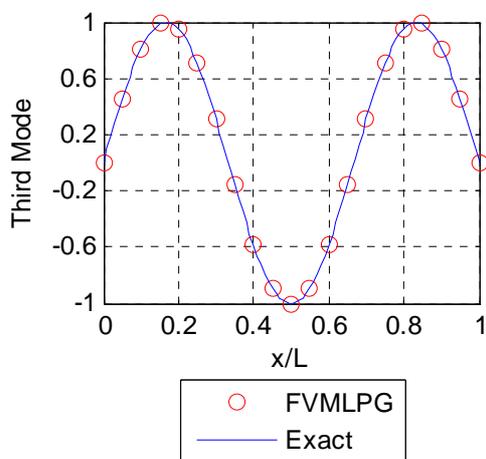
L'équation (6.1.24) est un problème généralisé aux valeurs propres. Ses solutions non triviales, c.-à-d. les valeurs propres et les vecteurs propres (N^i, a_j^i) , sont respectivement les charges critiques et les modes de flambages correspondants (figure 6.1.3).



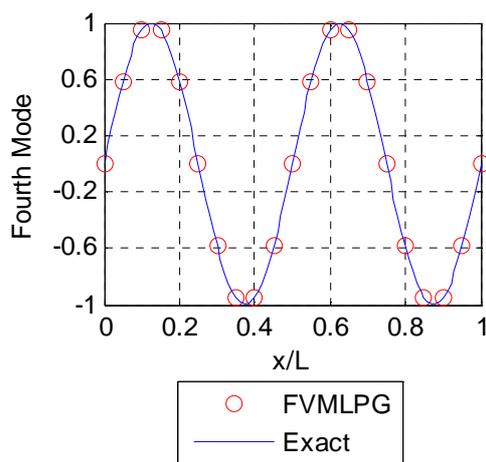
(a) Première mode de flambage



(b) Deuxième mode de flambage



(c) Troisième mode de flambage



(d) Quatrième mode de flambage

Figure 6.1.3 Modes de flambages de barre simplement bi articulée

Les résultats numériques se superposent avec la solution exacte pour les 4 modes de flambage.

6.2 Remarque et conclusion

En ce chapitre, la méthode de FVMLPG est étendue pour résoudre les problèmes de stabilité des poutres minces. Pour le schéma d'interpolation sans maillage dans la méthode FVMLPG, l'interpolation des moindres carrés mobiles est utilisée. Les résultats du mode de flambage indiquent une bonne exactitude de cette approche pour l'analyse statique d'instabilité. Cette partie est toutefois incomplète et mérite plus d'approfondissement à l'avenir pour prendre en compte les non linéarités géométriques dues aux grands déplacements par exemple.

Conclusions et perspectives

Dans ce travail, nous avons proposé une nouvelle méthode sans maillage intitulée « finite volume meshless local Petrov Galerkin (FVMLPG) » pour résoudre les problèmes élastostatiques, élastodynamiques et de stabilités. Dans un premier temps l'approche FVMLPG est mise au point pour les problèmes de classe C^0 unidimensionnels (chapitre 2). Une forme faible locale de l'équation d'équilibre est développée. Des approximations à la solution connue sous le nom de fonctions d'essai sont formées en utilisant l'interpolation des moindres carrés mobiles. La formulation de type Petrov Galerkin pour ces problèmes C^0 est présentée en détail. Un système d'équations algébriques est construit en utilisant l'interpolation MLS ainsi que les fonctions test de Petrov Galerkin sous la forme faible locale. Plusieurs exemples numériques, y compris les problèmes de patch tests, ainsi que les problèmes aux valeurs limites mixtes sont étudiés pour évaluer l'efficacité de la méthode. Les résultats numériques obtenus sont en parfaite concordance avec les solutions exactes. Une distribution lisse de la variable secondaire a été également observée sans utilisation de techniques de post traitement habituellement utilisées dans les autres méthodes numériques.

La méthode FVMLPG est ensuite présentée pour les problèmes de classe C^0 bidimensionnel, spécifiquement en contraintes planes (chapitre 3). Une forme faible locale est développée à partir de la forme pondérée résiduelle classique de l'équation d'équilibre. Des fonctions tests sont choisies dans un espace différent de celui des fonctions d'essai. Un système d'équations algébriques est construit en substituant les fonctions d'essai et test dans la forme faible locale. La méthode est évaluée à partir de plusieurs cas simples et quelques patches tests. Une étude comparative avec la méthode des éléments finis est également réalisée pour tester l'efficacité de cette nouvelle approche.

La méthode est ensuite étendue pour les problèmes de classe C^1 , comme les poutres d'Euler Bernoulli (chapitre 4). Dans les problèmes C^1 , les variables dépendantes et leurs premières dérivées sont continues, mais les dérivés d'ordre supérieur peuvent ne pas l'être. Une forme faible locale (LWF) de l'équation équilibre est développée. Comme pour les cas décrits dans les chapitres 3, les fonctions d'essai et test sont utilisées dans le LWF pour déduire un système d'équations algébriques. La méthode FVMLPG est testée sur plusieurs cas (patch test, poutre simple et poutre continue). Pour tous les exemples étudiés, les résultats concernant les deux

variables sont concordants avec les solutions exactes. Par ailleurs, une distribution lisse de la variable secondaire a été obtenue sans utilisation des techniques de post traitement.

En ce qui concerne les problèmes élastodynamiques bidimensionnels, l'équation discrétisée découle de la forme faible locale des équations d'équilibre. En utilisant les mêmes concepts de volume finis généraux (VF) et de meshless local Petrov Galerkin (MLPG) ainsi que les approximations des moindres carrés mobiles (MLS) pour la fonction d'essai, la forme faible locale est discrétisée, puis intégrée au-dessus du volume de contrôle ou du sous domaine local. Les résultats obtenus à partir d'exemples d'analyse de vibration des structures élastiques démontrent l'efficacité et l'exactitude de cette méthode.

Enfin, la méthode de FVMLPG est appliquée pour l'étude de l'instabilité d'une barre élastique sollicitée en compression. Mais, cette partie reste toutefois incomplète et mérite plus d'approfondissement à l'avenir.

En comparaison avec la MEF, on peut conclure que l'un des avantages majeur de cette méthode est la distribution lisse des variables secondaires. En effet dans la MEF les variables secondaires sont discontinues à travers les frontières d'inter élément. Ces discontinuités apparaissent du fait de l'utilisation des fonctions de forme linéaire par morceaux dans la construction des fonctions d'essai. Des techniques de post traitement sont donc nécessaires pour obtenir des distributions lisses de ces variables secondaires. Dans la méthode de FVMLPG, les éléments sont éliminés, et uniquement les nœuds sont utilisés dans le domaine du problème. Une fonction d'essai diffuse (non pas linéaire par morceaux) telle que l'interpolation des moindres carrés mobiles (MLS) permet d'obtenir directement ces variables secondaires.

Perspectives

Dans l'état actuel de son développement, La méthode FVMLPG ne permet pas l'analyse des problèmes complexes déjà traités par la MEF (non linéarité, grande déformation...). Mais, son développement futur nous semble prometteur au vue des résultats performants obtenus sur les cas simples. Le temps de calcul, qui souvent considéré comme paramètre important dans un traitement informatique mérite une attention particulière à l'avenir pour le ramener au niveau de la méthode des éléments finis. L'intégration numérique joue un rôle important dans la convergence des solutions numériques des méthodes sans maillages. Malheureusement, les fonctions de forme

nodales des interpolations sans maillages, telles que MLS, sont fortement complexes; ce qui rend difficile une intégration numérique plus précise de la forme faible et particulièrement dans les méthodes classiques de type Galerkin.

La poursuite du développement de cette méthode portera sur plusieurs aspects :

- Formulation non linéaire en grandes déformations
- Traitement de problèmes à valeurs propres complexes
- Modélisation de la fissuration des poutres en béton armé
- Etude de l'interaction milieu granulaire structures et/ou fluide - structures

Références bibliographiques

- [1] Liu WK, Belytschko T, and Chang H (1986), An arbitrary Lagrangian-Eulerian finite element method for path-dependent materials. *Comput. Methods Appl. Mech. Eng.* **58**, 227–246.
- [2] Liu WK, Chang H, Chen JS, and Belytschko T (1988), Arbitrary Lagrangian and Eulerian Petrov-Galerkin finite elements for nonlinear problems, *Comput. Methods Appl. Mech. Eng.* **68**, 259–310.
- [3] Huerta A and Liu WK (1988), Viscous flow with large free surface motion, *Comput. Methods Appl. Mech. Eng.* **9**, 277–324.
- [4] Liu WK, Chen JS, Belytschko T, and Zhang YF (1991), Adaptive ALE finite elements with particular reference to external work rate on frictional interface, *Comput. Methods Appl. Mech. Eng.* **93**, 189–216.
- [5] Needleman A (1988), Material rate dependent and mesh sensitivity in localization problems, *Comput. Methods Appl. Mech. Eng.* **67**, 68–85.
- [6] Needleman A (1989), Dynamic shear band development in plane strain, *ASME J. Appl. Mech.* **56**, 1–9.
- [7] Benz W (1990), Smooth particle hydrodynamics: a review. In: *Numerical Modeling of Non-linear Stellar Pulsation: Problems and Prospects*, Kluwer Academic, Boston.
- [8] Monaghan JJ (1982), Why particle methods work (Hydrodynamics), *SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput.* **3**, 422–433.
- [9] Monaghan JJ (1985), Particle methods for hydrodynamics, *Comput. Phys. Rep.* **3**, 71–124.
- [10] Bernard PS (1995), A deterministic vortex sheet method for boundary layer flow, *J. Comput. Phys.* **117**, 132–145.
- [11] Chorin AJ (1973), Numerical study of slightly viscous flow, *J. Fluid Mech.* **57**, 785–796.
- [12] Chorin AJ (1973), Discretization of a vortex sheet, with an example of roll-up, *J. Comput. Phys.* **13**, 423–429.
- [13] Chorin AJ (1978), Vortex sheet approximation of boundary layers, *J. Comput. Phys.* **27**, 428–442.
- [14] Leonard A (1980), Vortex methods for flow simulation, *J. Comput. Phys.* **37**, 289–335.
- [15] Leonard A (1985), Computing three-dimensional incompressible flows with vortex elements, *Annu. Rev. Fluid Mech.* **17**, 523–529.

- [16] Liszka T, and Orkisz J (1980), The finite difference method at arbitrary irregular grids and its application in applied mechanics, *Comput. Struct.* **11**, 83–95.
- [17] Liszka T (1984), An interpolation method for an irregular net of nodes, *Int. J. Numer. Methods Eng.* **20**, 1599–1612.
- [18] Feldneier H and Schnack J (2000). Molecular dynamics for fermions. Technical report, Gesellschaft fur Schwerionenforschung mbH.
- [19] Kobrak MN, and Bittner ER (2000), Quantum molecular dynamics study of polaron recombination in conjugated polymers, *Phys. Rev. B* **62**, 11473–11486.
- [20] Krumrine JR, Jang S, Alexander MH, and Voth GA (2000), Quantum molecular dynamics and spectral simulation of a boron impurity in solid para-hydrogen, *J. Chem. Phys.* **113**, 9079–9089.
- [21] Kihe C, Yildirim T, Mehrez H, and Ciraci S (2000), A first-principles study of the structure and dynamics of C_8H_8 , Si_8H_8 , and Ge_8H_8 moleculars, *J. Phys. Chem. A* **104**, 2724–2728.
- [22] Hedman F, and Laaksonen A (2000), Parallel aspects of quantum molecular dynamics simulations of liquids, *Comput. Phys. Commun.* **128**, 284–294.
- [23] Hong J and Zhao XS (2000), New propagators for quantum-classical molecular dynamics simulations, *J. Chem. Phys.* **113**, 930–935.
- [24] Rapaport DC (1995), *The Art of Molecular Dynamics Simulation*, Cambridge Univ Press, Cambridge, UK.
- [25] Allen MP and Tildesley DJ (1987), *Computer Simulation of Liquids*, Oxford Univ Press, Oxford, UK.
- [26] Allen MP and Tildesley DJ ed, (1993) *Computer Simulation of Chemical Physics*, Kluwer Academic Pub, Dordrecht.
- [27] Catlow CRA, Parker SC, and Allen MP ed, (1990), *Computer Modelling of Fluids Polymers and Solids*, Kluwer Academic Pub, Dordrecht.
- [28] Ciccotti G and Hoover WG (eds) (1986), *Molecular Dynamics Simulation of Statistical Mechanical Systems*, North-Holland, Amsterdam.
- [29] Ciccotti G, Frenkel D, and McDonald IR (eds) (1987), *Simulation of Liquids and Solids. Molecular Dynamics and Monte Carlo Methods in Statistical Mechanics*, North-Holland, Amsterdam.
- [30] Bird GA (ed) (1994), *Molecular Gas Dynamics and the Direct Simulation of Gas Flow*, Oxford Univ Press, Oxford, UK.
- [31] Oran ES, Oh CK, and Cybyk BZ (1998), Direct simulation Monte Carlo: Recent advances and applications, *Annu. Rev. Fluid Mech.* **30**, 403–441.

- [32] Tunon I, Martins-Costa MTC, Millot C, Ruiz-Lopez MF, and Rivail JL (1996), A coupled density functional-molecular mechanics Monte Carlo simulation: the water molecule in liquid water, *J. Comput. Chem.* **17**, 19–29.
- [33] Gross WJ, Vasileska D, and Ferry DK (1999), A novel approach for introducing the electron-electron and electron-impurity interactions in particle-based simulations, *IEEE Electron Device Lett.* **20**, 463–465.
- [34] Drovetsky BY, Chu JC, and Mak CH (1998), Computer simulations of self-avoiding polymerized membranes, *J. Chem. Phys.* **108**, 6554–6557.
- [35] Acioli PH (1997), Review of quantum monte carlo methods and their applications, *J. Mol. Struct.* **394**, 75–85.
- [36] Binder K (ed) (1988), *The Monte Carlo Method in Condensed Matter Physics*, Springer, Berlin, Heidelberg.
- [37] Binder K (ed) (1992), *The Monte Carlo Simulation in Statistical Physics*, Springer, Berlin, Heidelberg.
- [38] Baer R (2000), Ab-initio molecular deformation barriers using auxiliary-field quantum Monte Carlo with application to the inversion barrier of water, *Chem. Phys. Lett.* **324**, 101–107.
- [39] Liu WK, Belytschko T, and Mani A (1986), Probabilistic finite elements for nonlinear structural dynamics, *Comput. Methods Appl. Mech. Eng.* **56**, 61–81.
- [40] Liu WK, Belytschko T, and Mani A (1986), Random field finite elements, *Int. J. Numer. Methods Eng.* **23**, 1831–1845.
- [41] Liu WK, Chen YJ, and Belytschko T (1996), Three reliability methods for fatigue crack growth, *Eng. Fract. Mech.* **53**, 733–752.
- [42] Frisch U, Hasslacher B, and Pomeau Y (1986), Lattice gas cellular automata for the Navier-Stokes equations, *Phys. Rev. Lett.* **56**, 1505.
- [43] Kadanoff L (1986), On two levels, *Phys. Today* **39**, 7–9.
- [44] Kadanoff L, McNamara GR, and Zanetti G (1987), A Poiseuille viscometer for lattice gas automata, *Complex Syst.* **1**, 791.
- [45] Kadanoff L, McNamara GR, and Zanetti G (1989), From automata to fluid flow: comparisons of simulation and theory, *Phys. Rev. A* **40**, 4527.
- [46] Henon M (1987), Viscosity of a lattice gas, *Complex Syst.* **1**, 763.
- [47] Qian YH, d’Humières D, and Lallemand P (1992), Lattice BGK models for the Navier-Stokes equation, *Europhys. Lett.* **17**, 479–484.
- [48] Qian YH and Orszag SA (1993), Lattice BGK models for the Navier-Stokes equation: Nonlinear deviation in compressible regimes, *Europhys. Lett.* **21**, 255–259.

- [49] Qian YH, Succi S, and Orszag SA (2000), Recent advances in lattice Boltzmann computing, In: *Annual Reviews of Computational Physics, Volume III*, D Stauffer ~ed World Scientific, Singapore, 195–242.
- [50] Chen S, Wang Z, Shan XW, and Doolen GD ~1992!, Lattice Boltzmann computational fluid dynamics in three dimensions, *J. Stat. Phys.* **68**, 379–400.
- [51] Chen S and Doolen GD (1998), Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* **30**, 329–364.
- [52] Nayroles B, Touzot G, and Villon P (1992), Generalizing the finite element method: Diffuse approximation and diffuse elements, *Computational Mech., Berlin* **10**, 307–318.
- [53] Breitkopf P, Touzot G, and Villon P (1998), Consistency approach and diffuse derivation in element free methods based on moving least squares approximation, *Comp. Assist. Mech. Eng. Sci.* **5**, 479–501 ISSN: 1232-308X.
- [54] Breitkopf P, Touzot G, and Villon P (2000), Double grid diffuse collocation method. *Computational Mech., Berlin* **25**, 199–206.
- [55] Breitkopf P, Rassineux A, Touzot G, and Villon P (2000), Explicit form and efficient computation of MLS shape function and their derivatives, *Int. J. Numer. Methods Eng.* **48**, 451–466.
- [56] Belytschko T, Lu YY, and Gu L (1994), Element free galerkin methods. *Int. J. Numer. Methods Eng.* **37**, 229–256.
- [57] Belytschko T, Krongauz Y, Organ D, Fleming M, and Krysl P (1996), Meshless methods: An overview and recent developments, *Comput. Methods Appl. Mech. Eng.* **139**, 3–48.
- [58] Belytschko T, Krongauz Y, Dolbow J, and Gerlach C (1998), On the completeness of meshfree particle methods, *Int. J. Numer. Methods Eng.* **43**, 785–819.
- [59] Belytschko T, Organ D, and Gerlach C (2000), Element-free Galerkin methods for dynamic fracture in concrete, *Comput. Methods Appl. Mech. Eng.* **187**, 385–399.
- [60] Lu YY, Belytschko T, and Tabbara M (1995), Element-free Galerkin method for wave propagation and dynamic fracture, *Comput. Methods Appl. Mech. Eng.* **126**, 131–153.
- [61] Liu WK, Adee J, and Jun S (1993), Reproducing kernel and wavelets particle methods for elastic and plastic problems, In: *Advanced Computational Methods for Material Modeling*, AMD 180/PVP 268 ASME, 175–190.
- [62] Liu WK and Oberste-Brandenburg C (1993), Reproducing kernel and wavelets particle methods, In: *Aerospace Structures: Nonlinear Dynamics and System Response*, AD 33 ASME, 39–56.
- [63] Liu WK, Jun S, and Zhang YF (1995), Reproducing kernel particle methods, *Int. J. Numer. Methods Eng.* **20**, 1081–1106.

- [64] Liu WK, Jun S, Li S, Adee J, and Belytschko T (1995), Reproducing kernel particle methods for structural dynamics, *Int. J. Numer. Methods Eng.* **38**, 1655–1679.
- [65] Liu WK, Chen Y, Chang CT, and Belytschko T (1996), Advances in multiple scale kernel particle methods, *Computational Mech., Berlin* **18**, 73–111.
- [66] Liu WK, Chen Y, Jun S, Chen JS, Belytschko T, Uras RA, and Chang CT (1996), Overview and applications of the reproducing kernel particle methods, *Arch. Comput. Mech. Eng.: State of Rev.* **3**, 3–80.
- [67] Liu WK, Li S, and Belytschko T (1997), Moving least square reproducing kernel method Part I: Methodology and convergence, *Comput. Methods Appl. Mech. Eng.* **143**, 422–453.
- [68] Chen JS, Pan C, Wu CT, and Liu WK (1996), Reproducing kernel particle methods for large deformation analysis of nonlinear structures, *Comput. Methods Appl. Mech. Eng.* **139**, 195–227.
- [69] Chen JS, Wu CT, Yoon S, and You Y (2001), A stabilized conforming nodal integration for Galerkin meshfree methods, *Int. J. Numer. Methods Eng.* **50**, 435–466.
- [70] Duarte CA, and Oden JT (1996), hp Clouds—an hp meshless method, *Numer. Methods Partial Diff. Eqs.* **12**, 673–705.
- [71] Duarte CA, and Oden JT (1996), An hp adaptive method using clouds, *Comput. Methods Appl. Mech. Eng.* **139**, 237–262.
- [72] Liszka T, Duarte CAM, and Tworzydło WW (1996), hp-meshless cloud method, *Comput. Methods Appl. Mech. Eng.* **139**, 263–288.
- [73] Oden JT, Duarte CAM, and Zienkiewicz OC (1998), A new Cloud based hp finite element method, *Comput. Methods Appl. Mech. Eng.* **153**, 117–126.
- [74] Babuska I and Melenk JM (1997), The partition of unity method, *Int. J. Numer. Methods Eng.* **40**, 727–758.
- [75] Babuska I and Zhang Z (1998), The partition of unity method for the elastically supported beam, *Comput. Methods Appl. Mech. Eng.* **152**, 1–18.
- [76] Melenk JM and Babuska I (1996), The partition of unity finite element method: Basic theory and applications, *Comput. Methods Appl. Mech. Eng.* **139**, 289–314.
- [77] Atluri SN, and Zhu T (1998), A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation, *Comput. Model. Simul. Eng.* **3**, 187–196.
- [78] Atluri SN, Kim HG, and Cho JY (1999), A critical assessment of the truly meshless local Petrov-Galerkin (MLPG) and local boundary integral equation (LBIE) methods, *Computational Mech., Berlin* **24**, 348–372.

- [79] Atluri SN, Cho JY, and Kim HG (1999), Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least square interpolations, *Computational Mech., Berlin* **24**, 334–347.
- [80] Atluri SN and Zhu T (2000), The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics, *Computational Mech., Berlin* **25**, 169–179.
- [81] Furukawa T, Yang C, Yagawa G, and Wu CC (2000), Quadrilateral approaches for accurate free mesh method, *Int. J. Numer. Methods Eng.* **47**, 1445–1462.
- [82] Shirazaki M and Yagawa G (1999), Large-scale parallel flow analysis based on free mesh method: A virtually meshless method, *Comput. Methods Appl. Mech. Eng.* **174**, 419–431.
- [83] Yagawa G and Yamada T (1996), Free mesh method: A new meshless finite element method, *Computational Mech., Berlin* **18**, 383–386.
- [84] Yagawa G and Yamada T (1998), Meshless method on massively parallel processors with application to fracture mechanics, *Key Eng. Mater.* **145–149**, 201–210.
- [85] Yagawa G and Furukawa T (2000), Recent development of free mesh method, *Int. J. Numer. Methods Eng.* **47**, 1419–1417.
- [86] Cushman-Roisin B, Esenkov OE, and Mathias BJ (2000), A particle-in-cell method for the solution of two-layer shallow-water equations, *Int. J. Numer. Methods Fluids* **32**, 515–543.
- [87] Munz CD, Schneider R, Sonnendrücker E, Stein E, Voss U, and Westermann T (1999), A finite-volume particle-in-cell method for the numerical treatment of Maxwell-Lorentz equations on boundary-fitted meshes, *Int. J. Numer. Methods Eng.* **44**, 461–487.
- [88] Munz CD, Schneider R, and Voss U (1999), A finite-volume particle-in-cell method for the numerical simulation of devices in pulsed power technology, *Surv. Math. Ind.* **8**, 243–257.
- [89] Bardenhagen SG, Brackbill JU, and Sulsky D (2000), The material point method for granular materials, *Comput. Methods Appl. Mech. Eng.* **187**, 529–541.
- [90] Brackbill JU and Ruppel HM (1986), FLIP: A method for adaptively zoned, particle-in-cell calculations in two dimensions, *J. Comput. Phys.* **65**, 314–343.
- [91] Brackbill JU (1987), On modeling angular momentum and velocity in compressible fluid flow, *Comput. Phys. Commun.* **47**, 1.
- [92] Sulsky D and Schreyer HL (1996), Axisymmetric form of the material point with applications to upsetting and Taylor impact problems, *Comput. Methods Appl. Mech. Eng.* **139**, 409–429.
- [93] Aluru NR (2000), A point collocation method based on reproducing kernel approximations, *Int. J. Numer. Methods Eng.* **47**, 1083–1121.
- [94] Anderson C and Greengard C (1985), On vortex methods, *SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal.* **22**, 413–440.

- [95] Abraham FF (1996), Parallel simulations of rapid fracture, In: *Fracture-Instability Dynamics, Scaling and Ductile/Brittle Behavior Symp.*, Mater. Res. Soc, Pittsburgh PA, 311–320.
- [96] Abraham FF, Bernstein N, Broughton JQ, and Hess D (2000), Dynamic fracture of silicon: Concurrent simulation of quantum electrons, classical atoms, and the continuum solid, *MRS Bull.* **25**, 27–32.
- [97] Foiles SM, Baskes MI, and Daw MS (1986), Embedded-atom-method functions for FCC metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys, *Phys. Rev. B* **33**, 7983–7991.
- [98] Falk ML and Langer JS (2000), From simulation to theory in the physics of deformation and fracture, *MRS Bull.* **25**, 40–45.
- [99] Galli G, Cygi F, and Catellani A (1996), Quantum mechanical simulations of microfracture in a complex material, *Phys. Rev. Lett.* **82**, 3476–3479.
- [100] Reddy JN (2006), *An Introduction to the Finite Element Method (Third Edition)*. McGraw-Hill.
- [101] B. Nayroles, G. Touzot, and P. Villon (1992), Generalizing the finite element method: Diffuse approximation and diffuse elements, *Computational Mechanics*, **10**, 307–318.
- [102] Belytschko T, Lu YY, and Gu L (1994), Element-free Galerkin methods, *International Journal for Numerical Methods in Engineering*, **37**, 229–256.
- [103] Mukherjee YX and Mukherjee S (1997), On boundary conditions in the element-free Galerkin method, *Computational Mechanics*, **19**, 264–270.
- [104] Atluri SN and Zhu T (1998), A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics, *Computational Mechanics*, **22**, 117–127.
- [105] Shepard D (1968), A two-dimensional function for irregularly spaced points. In *23rd ACM National Conference*, 517–524.
- [106] Atluri SN (2005), *Methods of computer modeling in engineering & the sciences Vol. I*, 600 pages.
- [107] Oñate E, Cervera M, Zienkiewicz OC, (1994), A finite volume format for structural mechanics, *Int. J. Numer. Methods Engrg.* **37**, 181–201.
- [108] Demirdzic I, Muzaferija S. (1994), Finite volume method for stress analysis in complex domains, *Int. J. Numer. Methods Engrg.* **37**, 3751–3766.
- [109] Jasak H, Weller HG, (2000), Application of the finite volume method and unstructured meshes to linear elasticity, *Int. J. Numer. Methods Engrg.* **48**, 267–287.
- [110] Atluri SN, Zhu T, (1998), A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics, *Comput. Mech.* **22**, 117–127.
- [111] Timoshenko SP, and Goodier JN, (1976), *Theory of elasticity*, 3rd edition, McGraw Hill.

- [112] Atluri SN, Cho JY, and Kim HG, (1999), Analysis of thin beams, using the meshless local Petrov-Galerkin method, with generalized moving least squares interpolations, *Computational Mechanics*, **24**, 334-347.
- [113] Zienkiewicz OC, Taylor RL and Zhu JZ, (2006), *The Finite Element Method Its Basis & Fundamentals*, Butterworth-Heinemann; 6th edition.
- [114] Newmark NM, (1959), A method of computation for structural dynamics, *Journal of the Engineering Mechanics Division, ASCE*, **85**, 67-94.
- [115] Clough R. W. and Penzien J, (1975), *Dynamics of Structures*, New York McGraw-Hill.
- [116] Duflot M and Nguyen-Dang H, (2004), A meshless method with enriched weight functions for fatigue crack, *Int. J. Num. M. Eng.*, **59**, 1945-1961.
- [117] Duflot M and Nguyen-Dang H, (2004), Fatigue crack growth analysis by an enriched meshless method, *J. Comp. Appl. Mech.*, **168**, 155-164.
- [118] Duflot M and Nguyen-Dang H, (2004), Global error estimation with displacement and equilibrium meshless methods, In Yao ZH, Yuan MW, Zhong WX (editors), *Computational Mechanics*, 316-321, Tsinghua University Press, Beijing.
- [119] Duflot M and Nguyen-Dang H, (2002), Dual analysis by a meshless method, *Commun. Numer. Meth. Eng.*, **18**, 621-631.
- [120] Duflot M and Nguyen-Dang H, (2002), A truly meshless Galerkin method based on a moving least squares quadrature, *Commun. Numer. Meth. Eng.*, **18**, 441-449.
- [121] Duflot M (2006), A meshless method with enriched weight functions for three-dimensional crack propagation, *Int. J. Num. M. Eng.*, **65**, 1970-2006.
- [122] Vinh Phu N, Rabczuk T, Bordas S and Duflot M (2008), Meshless methods: a review and computer implementation aspects, *Math. Comp. Simu.* (in press).
- [123] Joyot P, Trunzler J, Chinesta F, (2006), Accounting for incompressibility in reproducing kernel particle meshless approximations, *Revue Européenne de Mécanique Numérique (European Journal of Computational Mechanics)*, **15 (5)**, 513-527.
- [124] Martinez MA, Cueto E, Doblare M and Chinesta F, (2001), A meshless simulation of injection processes involving short fibers molten composites, *Int. J. Form. Proc.*, **4 (3-4)**, Special Issue on Material Forming, Habraken AM.
- [125] Yvonnet J, Ryckelynck D, Lorong Ph and Chinesta F, (2005), The C-NEM for Discontinuous Natural Element Galerkin Interpolation and Moving Interfaces, *Springer, Lectures Notes on Computational Science and Engineering*, **43**, 255-270.
- [126] Yvonnet J, Lorong Ph, Ryckelynck D and Chinesta F, (2005), Simulating Dynamic Thermo-Elasto-Plasticity in Large Transformations with Adaptive Refinement in the Natural

- Element Method: Application to Shear Banding, *Int. J. Form. Proc., Special Issue, Material Forming Process Optimization*, 317-345.
- [127] Missoum Benziane D, Ryckelynck D and Chinesta F, (2007) A new fully coupled two-scales modelling for mechanical problems involving microstructures: the 95/5 technic, *Comp. Meth. Appl. Mech. Eng.*, **196 (21-24)**, 2325-2337.
- [128] Yvonnet J, Coffignal G, Ryckelynck D, Lorong Ph and Chinesta F, (2006), A simple error indicator for meshfree methods based on natural neighbors, *Computers & Structures*, *84 (21)*, 1301-1312.
- [129] Yvonnet J, Ryckelynck D, Lorong Ph and Chinesta F, (2004), A new extension of the natural element method for non-convex and discontinuous problems: the constrained natural element method (C-NEM), *Int. J. Num. Meth. Eng.*, **60**, 1451-1474.

Publications liées aux travaux de la thèse :

- [130] M. R. Moosavi, A. Khelil, (2008), Accuracy and Computational Efficiency of the Finite Volume Method Combined with the Meshless Local Petrov-Galerkin in Comparison with the Finite Element Method in Elasto-static Problem, *ICCES*, **113**, no.1, 1-28.
- [131] M. R. Moosavi, A. Khelil, (2008), The finite volume meshless local Petrov-Galerkin method for solving Euler-Bernoulli beam problems, submitted in *Computers and Structures*.
- [132] M. R. Moosavi, A. Khelil, (2008), The Finite Volume Meshless Local Petrov-Galerkin Method in Elasto-static Problems, submitted in *Computers and Structures*.
- [133] M. R. Moosavi, A. Khelil, (2008), Finite Volume Meshless Local Petrov-Galerkin Method in Vibration of Structures, The Fourth International Conference on Advances in Mechanical Engineering and Mechanics, Sousse-Tunisia.
- [134] M. R. Moosavi, A. Khelil, (2008), Finite Volume Meshless Local Petrov-Galerkin Method in Structural Transient Analysis, 30th International Conference on Boundary Elements and other Mesh Reduction Methods, Maribor, Slovenia.
- [135] M. R. Moosavi, A. Khelil, (2008), The finite volume meshless local Petrov-Galerkin method for solving Euler-Bernoulli beam, World Congress on Engineering, London, U.K.
- [136] M. R. Moosavi, A. Khelil, (2008), A Combined Formulation of the Finite Volume Method and the Meshless Local Petrov-Galerkin Method applied to Solve Linear Elastostatic Problems, 8th World Congress on Computational Mechanics (WCCM8) and the 5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008), Venice, Italy.

- [137] M. R. Moosavi, A. Khelil, (2008), A Combined Formulation of the Finite Volume Method and the Meshless Local Petrov-Galerkin Method applied to Modal Analysis, 8th World Congress on Computational Mechanics (WCCM8) and the 5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008), Venice, Italy.

Annexe

Code développé sous MATLAB

```

clc; clear;
L = 4;   c = 0.25;   E0 = 1;   nu0 = 0.25;   I0=2*c^3/3;
elemType = 'T3';   numx   = 80;   numy   = 2;
plotMesh = 0;   plotDeformed = 0;   stressContour = 0;
loadType='uniform transverse';   stressState='PLANE_STRESS';
P = -1;   mls_type = 'MLS2D1';
supportSize = 1.72;   testSize = 0.257;
meshSize = mean([c*2/numy, L/numx]);
supportSize = supportSize * meshSize;
testSize = testSize * meshSize;
if ( strcmp(stressState,'PLANE_STRESS') )
    C=E0/(1-nu0^2)*[1 nu0 0; nu0 1 0; 0 0 (1-nu0)/2 ];
else
    C=E0/(1+nu0)/(1-2*nu0)*[ 1-nu0 nu0 0; nu0 1-nu0 0; 0 0 1/2-nu0 ];
end
[U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,L/2,0);
u2L = U_beam(1,2);
switch elemType
case 'Q4'
    nnx=numx+1;   nny=numy+1;
    node=square_node_array([0 -c],[L -c],[L c],[0 c],nnx,nny);
    inc_u=1; inc_v=nnx; node_pattern=[ 1 2 nnx+2 nnx+1 ];
    element=make_elem(node_pattern,numx,numy,inc_u,inc_v);
case 'Q9'
    nnx=2*numx+1;   nny=2*numy+1;
    node=square_node_array([0 -c],[L -c],[L c],[0 c],nnx,nny);

```

```

inc_u=2;    inc_v=2*nnx;
node_pattern=[ 1 3 2*nnx+3 2*nnx+1 2 nnx+3 2*nnx+2 nnx+1 nnx+2 ];
element=make_elem(node_pattern,numx,numy,inc_u,inc_v);
otherwise
nnx=numx+1;  nny=numy+1;
node=square_node_array([0 -c],[L -c],[L c],[0 c],nnx,nny);
node_pattern1=[ 1 2 nnx+1 ]; node_pattern2=[ 2 nnx+2 nnx+1 ];
inc_u=1;    inc_v=nnx;
element=[make_elem(node_pattern1,numx,numy,inc_u,inc_v);
         make_elem(node_pattern2,numx,numy,inc_u,inc_v) ];
end
numnode=size(node,1);  numelem=size(element,1);
uln=nnx*(nny-1)+1;    urn=nnx*nny;
lrn=nnx;              lln=1;
cln=nnx*(nny-1)/2+1;  crn=nnx*(nny-1)/2+nnx;
switch elemType
case 'Q9'
rightEdge=[ lrn:2*nnx:(uln-1); (lrn+2*nnx):2*nnx:urn; (lrn+nnx):2*nnx:urn ];
leftEdge =[ uln:-2*nnx:(lrn+1); (uln-2*nnx):-2*nnx:1; (uln-nnx):-2*nnx:1 ];
topEdge = [ uln:2:urn-2; uln+1:2:urn-1; uln+2:2:urn ];
edgeElemType='L3';
otherwise
rightEdge=[ lrn:nnx:(uln-1); (lrn+nnx):nnx:urn ];
leftEdge =[ uln:-nnx:(lrn+1); (uln-nnx):-nnx:1 ];
topEdge =[ uln:urn-1; uln+1:urn ];
edgeElemType='L2';
end
fixedNodeX=[(crn-cln)/2+cln]';  fixedNodeY=[cln crn]';
fn=1;
if ( plotMesh )
figure(fn)
clf
plot_mesh(node,element,elemType,'g. ');
hold on
plot_mesh(node,rightEdge,edgeElemType,'g. ');

```

```

plot_mesh(node,leftEdge,edgeElemType,'g. ');
plot(node(fixedNodeX,1),node(fixedNodeX,2),'r>');
plot(node(fixedNodeY,1),node(fixedNodeY,2),'r^');
axis off
axis([0 L -c c])
fn=fn+1;
disp('(paused)')
pause
end
mlpgnode = zeros(numnode,1);
d = zeros(numnode,1);
for i=1:numnode
    d =sqrt((node(:,1)-node(i,1)).^2 + (node(:,2)-node(i,2)).^2);
    neighNode = find(d < supportSize+testSize);
    mlpgnode(i,1:size(neighNode,1)+1) = [size(neighNode,1) i setdiff(neighNode,[i])];
end
clear d neighNode
mlpgelement = zeros(numnode,1);
for i=1:numnode
    [neighElem, col] = find(element == i);
    mlpgelement(i,1:size(neighElem,1)+1,1:2) = [size(neighElem,1) 0; neighElem col];
end
switch elemType
case 'Q4'
    mlpgedge = setdiff( ...
        [element(:,[1 2]);element(:,[2 3]);element(:,[3 4]);element(:,[4 1])], ...
        [element(:,[2 1]);element(:,[3 2]);element(:,[4 3]);element(:,[1 4])], ...
        'rows');
case 'T3'
    mlpgedge = setdiff( ...
        [element(:,[1 2]);element(:,[2 3]);element(:,[3 1])], ...
        [element(:,[2 1]);element(:,[3 2]);element(:,[1 3])], ...
        'rows');
otherwise
    disp('Unsupported FVMLPG elements');

```

```

end
mlpgangle = zeros(numnode,2);
dr = node(mlpgedge(:,2),:) - node(mlpgedge(:,1),:);
mlpgangle(mlpgedge(:,1),1) = cart2pol(dr(:,1),dr(:,2));
mlpgangle(find(mlpgangle>pi-eps),1)=-pi;
mlpgangle(mlpgedge(:,2),2) = pi+mlpgangle(mlpgedge(:,1),1);
dr = mlpgangle(:,2)-mlpgangle(:,1);
ii = find(dr > 2*pi); mlpgangle(ii,2) = mlpgangle(ii,2)-2*pi;
ii = find(dr < 0); mlpgangle(ii,1) = mlpgangle(ii,1)-2*pi;
ii = setdiff([1:numnode]', mlpgedge(:,1));
mlpgangle(ii,1) = -inf; mlpgangle(ii,2) = inf;
f0 = linspace(0,1,9)';
for i=1:numnode
    ii = mlpnode(i,3:mlpnode(i,1));
    sita = cart2pol(node(ii,1)-node(i,1),node(ii,2)-node(i,2));
    sita = pi*[-1:0.5:1]';
    startangle = max(mlpgangle(i,1),min(sita));
    ii = find(sita < startangle);
    sita(ii) = sita(ii)+2*pi;
    endangle = min(startangle+2*pi,mlpgangle(i,2));
    sita = sita(find(sita <=endangle));
    [f,sita] = ecdf(sita);
    sita = sort(unique(sita(dsearchn(f,f0))));
    if(sita(1) > startangle+eps)
        sita = [startangle; sita];
    end
    if(sita(end) < endangle-eps)
        sita = [sita;endangle];
    end
    mlpgintangle(i,1:size(sita,1)+1) = [size(sita,1) sita]';
end
uFixed=zeros(size(fixedNodeX)); vFixed=zeros(size(fixedNodeY));
U=zeros(2*numnode,1); f=zeros(2*numnode,1);
KS=zeros(2*numnode,3*numnode); xs=[1:numnode]';
ys=xs+numnode; sxx=xs;

```

```

syy=sxx+numnode;          sxy=syy+numnode;
xs_ip = 0;                ys_ip = numnode;
sxx_ip = xs_ip;          syy_ip = sxx_ip + numnode;
sxy_ip = syy_ip + numnode;
[W,Q]=quadrature( 3, 'GAUSS', 1 );
Q = Q/2.+0.5;           W = W/2.;
topNode = find(node(:,2) == c);
for e=1:size(topNode,1)
    n = topNode(e);       sctr=[n];
    sctrx=sctr;          sctry=sctrx+numnode;
    for q=1:size(W,1)
        pt=Q(q,:);       wt=W(q);
        detJ0=testSize;   fxPt = 0.0;   fyPt = P;
        if(mlpgangle(n,1) == 1.0*pi)
            xPt=node(n,1)-testSize*pt(1);   yPt=node(n,2);
            [U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,xPt,yPt);
            fyPt = fyPt - S_beam(1,2);
        end

        if(mlpgangle(n,2) == 0.0*pi );
            xPt=node(n,1)+testSize*pt(1);   yPt=node(n,2);
            [U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,xPt,yPt);
            fyPt = fyPt - S_beam(1,2);
        end

        if(mlpgangle(n,2) == 2.0*pi );
            xPt = node(n,1)+testSize*pt(1);   yPt=node(n,2);
            [U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,xPt,yPt);
            fyPt = fyPt + S_beam(1,2);
        end
    end
    f(sctry)=f(sctry)-fyPt*detJ0*wt;
end
end
rightNode = find(node(:,1) == L);
for e=1:size(rightNode,1)

```

```

n = rightNode(e);    sctr=[n];
sctrx=sctr;         sctry=sctrx+numnode;
for q=1:size(W,1)
    pt=Q(q,:);      wt=W(q);
    detJ0=testSize;  fxPt = 0.0;    fyPt = 0.0;
    if(mlpgangle(n,1) == 0.5*pi)
        xPt = node(n,1);        yPt=node(n,2)+testSize*pt(1);
        [U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,xPt,yPt);
        fyPt = fyPt + S_beam(1,3);
    end
    if(mlpgangle(n,2) == 1.5*pi)
        xPt = node(n,1);        yPt=node(n,2)-testSize*pt(1);
        [U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,xPt,yPt);
        fyPt = fyPt + S_beam(1,3);
    end
    f(sctry)=f(sctry)-fyPt*detJ0*wt;
end
end
leftNode = find(node(:,1) == 0);
for e=1:size(leftNode,1)
    n = leftNode(e);    sctr=[n];
    sctrx=sctr;         sctry=sctrx+numnode;
    for q=1:size(W,1)
        pt=Q(q,:);      wt=W(q);    detJ0=testSize;
        fxPt = 0.0;    fyPt = 0.0;
        if(mlpgangle(n,2) == 0.5*pi)
            xPt = node(n,1);
            yPt=node(n,2)+testSize*pt(1);
            [U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,xPt,yPt);
            fyPt = fyPt + S_beam(1,3);
        end
        if(mlpgangle(n,1) == -0.5*pi)
            xPt = node(n,1);    yPt=node(n,2)-testSize*pt(1);
            [U_beam,S_beam]=exact_solution(loadType,P,E0,nu0,L,c,xPt,yPt);
            fyPt = fyPt + S_beam(1,3);
        end
    end
end

```

```

    end
    f(sctry)=f(sctry)+fyPt*detJ0*wt;
end
end
[W,Q]=quadrature( 6, 'GAUSS', 1 );
Q = Q/2.+0.5;    W = W/2.;
for n=1:numnode
    nn=mlpgnode(n,1);          sctr=mlpgnode(n,2:nn+1);
    sctrN=[n+sxx_ip n+syy_ip n+sxy_ip];  sctrB=[ sctr sctr+numnode ];
    nseg = mlp gintangle(n,1);    angle = mlp gintangle(n,2:nseg+1);
    xn = node(n,:);              pts = node(sctr,:);
    for(seg = 1:nseg-1)
        angle0 = angle(seg);      dangle = angle(seg+1)-angle0;
        arclength = dangle*testSize;
        for q=1:size(W,1)
            pt=Q(q,1);            wt=W(q);
            [nx, ny] = pol2cart(angle0+dangle*pt, 1);
            x0 = xn + [nx ny]* testSize;
            [N,dNdx]=mls_basis(mls_type,x0,pts,supportSize);
            KS(n+sxx_ip,sctr+sxx_ip) = KS(n+sxx_ip,sctr+sxx_ip) + ...
                (nx * wt * arclength) * N';
            KS(n+sxx_ip,sctr+sxy_ip) = KS(n+sxx_ip,sctr+sxy_ip) + ...
                (ny * wt * arclength) * N';
            KS(n+syy_ip,sctr+syy_ip) = KS(n+syy_ip,sctr+syy_ip) + ...
                (ny * wt * arclength) * N';
            KS(n+syy_ip,sctr+sxy_ip) = KS(n+syy_ip,sctr+sxy_ip) + ...
                (nx * wt * arclength) * N';
        end
    end
end
end
TU=zeros(3*numnode,2*numnode);
TS=zeros(3*numnode,3*numnode);
for n=1:numnode
    nn=mlpgnode(n,1);          sctr=mlpgnode(n,2:nn+1);
    sctrN=[n+sxx_ip n+syy_ip n+sxy_ip];

```

```

sctrU=[sctr+xs_ip sctr+ys_ip];
sctrS=[sctr+sxx_ip sctr+syy_ip sctr+sxy_ip];
xn = node(n,:);      pts = node(sctr,:);
[N,dNdx]=mls_basis(mls_type,xn,pts,supportSize);
TS(n+sxx_ip,sctr+sxx_ip) = TS(n+sxx_ip,sctr+sxx_ip)+ N';
TS(n+syy_ip,sctr+syy_ip) = TS(n+syy_ip,sctr+syy_ip)+ N';
TS(n+sxy_ip,sctr+sxy_ip) = TS(n+sxy_ip,sctr+sxy_ip)+ N';
B=zeros(3,2*nn);
B(1,1:nn)   = dNdx(:,1)';      B(2,nn+1:2*nn) = dNdx(:,2)';
B(3,1:nn)   = dNdx(:,2)';      B(3,nn+1:2*nn) = dNdx(:,1)';
TU(sctrN,sctrU)=TU(sctrN,sctrU)+C*B;
end
TSI = inv(TS);   K = zeros(2*numnode);   K = KS * TSI * TU;
bcwt=max(diag(K));   udofs=fixedNodeX;
vdofs=fixedNodeY+numnode;
fixedNode = union(fixedNodeX, fixedNodeY);
for i = 1:size(fixedNode,1)
    n = fixedNode(i);   nn=mlpgnode(n,1);
    sctr=mlpgnode(n,2:nn+1);
    xn = node(n,:);      pts = node(sctr,:);
    [N,dNdx]=mls_basis(mls_type,xn,pts,supportSize);
    if ismember(n, fixedNodeX)
        f(n) = 0.;      K(n,:) = 0.;      K(n,sctr) = N'*bcwt*1.e10;
    end
    if ismember(n, fixedNodeY)
        f(n+numnode) = 0.;      K(n+numnode,:) = 0.;
        K(n+numnode,sctr+numnode) = N'*bcwt*1.e10;
    end
end
end
U=K\f;
virtualU = U;   nodestress=zeros(numnode,3);
for n=1:numnode
    nn=mlpgnode(n,1);      sctr=mlpgnode(n,2:nn+1);
    sctrB=[ sctr sctr+numnode ];
    xn = node(n,:);      pts = node(sctr,:);

```

```

[N,dNdx]=mls_basis(mls_type,xn,pts,supportSize);
U(n) = N'*virtualU(sctr);
U(n+numnode) = N'*virtualU(sctr+numnode);
strain = zeros(3,1);
strain([1,3],1) = dNdx'*virtualU(sctr);
strain([3,2],1) = strain([3,2],1) + dNdx'*virtualU(sctr+numnode);
nodestress(n,:) = (C* strain)';
end

```

```

function [Nv,dNdx]=lagrange_basis(type,coord,dim)

```

```

if ( nargin == 2 )

```

```

    dim=1;

```

```

end

```

```

switch type

```

```

case 'L2'

```

```

    if size(coord,2) < 1

```

```

        disp('Error coordinate needed for the L2 element')

```

```

    else

```

```

        xi=coord(1);

```

```

        N=([1-xi,1+xi]/2)';

```

```

        dNdx=[-1;1]/2;

```

```

    end

```

```

case 'T3'

```

```

    if size(coord,2) < 2

```

```

        disp('Error two coordinates needed for the T3 element')

```

```

    else

```

```

        xi=coord(1); eta=coord(2);

```

```

        N=[1-xi-eta;xi;eta];

```

```

        dNdx=[-1,-1;1,0;0,1];

```

```

    end

```

```

case 'T3fs'

```

```

if size(coord,2) < 2
    disp('Error two coordinates needed for the T3 element')
else
    xi=coord(1); eta=coord(2);
    N=[1-xi-eta;xi;eta];
    dNdx1=[-1,-1;1,0;0,1];
end

```

```

case 'Q4'
if size(coord,2) < 2
    disp('Error two coordinates needed for the Q4 element')
else
    xi=coord(1); eta=coord(2);
    N=1/4*[ (1-xi)*(1-eta);
            (1+xi)*(1-eta);
            (1+xi)*(1+eta);
            (1-xi)*(1+eta)];
    dNdx1=1/4*[-(1-eta), -(1-xi);
                1-eta,   -(1+xi);
                1+eta,   1+xi;
                -(1+eta), 1-xi];
end

```

```

otherwise
    disp(['Element ',type,' not yet supported'])
    N=[]; dNdx1=[];
end

```

```

if(dim == 1)
    Nv = N;
else
    I=eye(dim);
    Nv=[];
    for i=1:size(N,1)
        Nv=[Nv;I*N(i)];
    end
end

```

```
    end
end

function element=make_elem(node_pattern,num_u,num_v,inc_u,inc_v)

if ( nargin < 5 )
    disp(['Not enough parameters specified for make_elem function'])
end

inc=[zeros(1,size(node_pattern,2))];
e=1;
element=zeros(num_u*num_v,size(node_pattern,2));

for row=1:num_v
    for col=1:num_u
        element(e,:)=node_pattern+inc;
        inc=inc+inc_u;
        e=e+1;
    end
    inc=row*inc_v;
end

function [Nv,dNdx]=mls_basis(type,x0,pts,supportSize)

nn = size(pts,1);

switch type(1:3)
    case 'MLS'
        interpolation = 1;
    otherwise
        interpolation = 0;
        disp(['Interpolation ',type(1:3),' not yet supported'])
        N=[]; dNdx=[];
end
```

```

switch type(4:5)
    case '2D'
        numdim = 2;
    otherwise
        numdim = 0;
        disp(['Dimension ',type(4:5),' not yet supported'])
        N=[]; dNdx=[];
end

switch type(6)
    case '1'
        numorder = 1;
    otherwise
        numorder = 0;
        disp(['Order ',type(6),' not yet supported'])
        N=[]; dNdx=[];
end

if size(pts,2) ~= numdim
    disp('Error', num2str(numdim), ' coordinates needed for the MLS')
end

if nn < numorder
    disp('Error minimum', num2str(numorder), ' points needed for the MLS')
end

if interpolation == 1
    switch numdim
        case 1
            xi=[pts(:,1)-x0(1)];
            ri = sqrt(xi(:,1).^2);
            [wt,dwdr] = weight('SPLINE4',ri, supportSize);
            if(numorder == 1)
                P = [ones(nn,1) xi];
                B = [P(:,1).*wt ...

```

```

    P(:,2).*wt]';
elseif(numorder == 2)
    P = [ones(nn,1) xi ...
        xi(:,1).^2];
    B = [P(:,1).*wt ...
        P(:,2).*wt ...
        P(:,3).*wt]';
elseif(numorder == 3)
    P = [ones(nn,1) xi ...
        xi(:,1).^2 ...
        xi(:,1).^3];
    B = [P(:,1).*wt ...
        P(:,2).*wt ...
        P(:,3).*wt ...
        P(:,4).*wt]';
end

A = B*P;
A1 = inv(A);
A1B = A1*B;
Nv = A1B(1,:);
IPA1B = eye(nn) - P*A1B;
A1P = A1*P';
pA1Pdwr = A1P(1,:).'*dwdr;
dNdx1 = A1B(2,:).'...
    + IPA1B' * ...
    [pA1Pdwr.*xi(:,1)];

case 2
    xi=[pts(:,1)-x0(1) pts(:,2)-x0(2)];
    ri = sqrt(xi(:,1).^2 + xi(:,2).^2);
    [wt,dwdr] = weight('SPLINE4',ri, supportSize);
    if(numorder == 1)
        P = [ones(nn,1) xi];
        B = [P(:,1).*wt ...

```

```

        P(:,2).*wt ...
        P(:,3).*wt]';
elseif(numorder == 2)
    P = [ones(nn,1) xi ...
        xi(:,1).^2 xi(:,1).*xi(:,2) xi(:,2).^2];
    B = [P(:,1).*wt ...
        P(:,2).*wt ...
        P(:,3).*wt ...
        P(:,4).*wt ...
        P(:,5).*wt ...
        P(:,6).*wt]';
end

A = B*P;
A1 = inv(A);
A1B = A1*B;
Nv = A1B(1,:);
IPA1B = eye(nn) - P*A1B;
A1P = A1*P';
pA1Pdwdr = A1P(1,:).*dwdr;
dNdxi = A1B(2:3,:)'...
    - IPA1B' * ...
    [pA1Pdwdr.*xi(:,1) pA1Pdwdr.*xi(:,2)];
end
end

function plot_field(X,connect,elem_type,field)

if ( nargin == 4 )
    nodesoff=0;
end

if ( size(field) == size(connect) )
    elementalField=1;
else

```

```
    elementalField=0;
end

% fill X if needed
if (size(X,2) < 3)
    for c=size(X,2)+1:3
        X(:,c)=[zeros(size(X,1),1)];
    end
end

holdState=ishold;
hold on

if ( strcmp(elem_type,'T3') )
    ord=[1,2,3,1];
elseif ( strcmp(elem_type,'L2') )
    ord=[1,2];
end

for e=1:size(connect,1)

    xpt=X(connect(e,ord),1);
    ypt=X(connect(e,ord),2);
    zpt=X(connect(e,ord),3);

    if ( elementalField )
        fpt=field(e,ord);
    else
        fpt=field(connect(e,ord));
    end

    fill3(xpt,ypt,zpt,fpt)
end

shading interp
```

```
axis equal
```

```
if ( ~holdState )
```

```
    hold off
```

```
end
```

```
function plot_mesh(X,connect,elem_type,se)
```

```
if ( nargin < 4 )
```

```
    se='w-';
```

```
end
```

```
holdState=ishold;
```

```
hold on
```

```
% fill X if needed
```

```
if (size(X,2) < 3)
```

```
    for c=size(X,2)+1:3
```

```
        X(:,c)=[zeros(size(X,1),1)];
```

```
    end
```

```
end
```

```
for e=1:size(connect,1)
```

```
    if ( strcmp(elem_type,'T3') )
```

```
        ord=[1,2,3,1];
```

```
    elseif ( strcmp(elem_type,'L2') )
```

```
        ord=[1,2];
```

```
    end
```

```
    for n=1:size(ord,2)
```

```
        xpt(n)=X(connect(e,ord(n)),1);
```

```
        ypt(n)=X(connect(e,ord(n)),2);
```

```
        zpt(n)=X(connect(e,ord(n)),3);
```

```
    end
```

```
    plot3(xpt,ypt,zpt,se)
end

rotate3d on
axis equal

if ( ~holdState )
    hold off
end

function [W,Q] = quadrature( quadorder, qt, sdim )

if ( nargin < 3 )
    if ( strcmp(qt,'GAUSS') == 1 )
        dim = 1;
    else
        dim = 2;
    end
end

if ( nargin < 2 )
    type = 'GAUSS';
end

if ( strcmp(qt,'GAUSS') == 1 )

    if ( quadorder > 8 )
        disp('Order of quadrature too high for Gaussian Quadrature');
        quadorder =8;
    end

    quadpoint=zeros(quadorder^sdim ,sdim);
    quadweight=zeros(quadorder^sdim,1);

    r1pt=zeros(quadorder,1); r1wt=zeros(quadorder,1);
```

switch (quadorder)

case 1

r1pt(1) = 0.0000000000000000;
r1wt(1) = 2.0000000000000000;

case 2

r1pt(1) = 0.577350269189626;
r1pt(2) = -0.577350269189626;

r1wt(1) = 1.0000000000000000;
r1wt(2) = 1.0000000000000000;

case 3

r1pt(1) = 0.774596669241483;
r1pt(2) = -0.774596669241483;
r1pt(3) = 0.0000000000000000;

r1wt(1) = 0.5555555555555556;
r1wt(2) = 0.5555555555555556;
r1wt(3) = 0.8888888888888889;

case 4

r1pt(1) = 0.861134311594053;
r1pt(2) = -0.861134311594053;
r1pt(3) = 0.339981043584856;
r1pt(4) = -0.339981043584856;

r1wt(1) = 0.347854845137454;
r1wt(2) = 0.347854845137454;
r1wt(3) = 0.652145154862546;
r1wt(4) = 0.652145154862546;

case 5

r1pt(1) = 0.906179845938664;

r1pt(2) = -0.906179845938664;
r1pt(3) = 0.538469310105683;
r1pt(4) = -0.538469310105683;
r1pt(5) = 0.000000000000000;

r1wt(1) = 0.236926885056189;
r1wt(2) = 0.236926885056189;
r1wt(3) = 0.478628670499366;
r1wt(4) = 0.478628670499366;
r1wt(5) = 0.568888888888889;

case 6

r1pt(1) = 0.932469514203152;
r1pt(2) = -0.932469514203152;
r1pt(3) = 0.661209386466265;
r1pt(4) = -0.661209386466265;
r1pt(5) = 0.238619186003152;
r1pt(6) = -0.238619186003152;

r1wt(1) = 0.171324492379170;
r1wt(2) = 0.171324492379170;
r1wt(3) = 0.360761573048139;
r1wt(4) = 0.360761573048139;
r1wt(5) = 0.467913934572691;
r1wt(6) = 0.467913934572691;

case 7

r1pt(1) = 0.949107912342759;
r1pt(2) = -0.949107912342759;
r1pt(3) = 0.741531185599394;
r1pt(4) = -0.741531185599394;
r1pt(5) = 0.405845151377397;
r1pt(6) = -0.405845151377397;
r1pt(7) = 0.000000000000000;

```
r1wt(1) = 0.129484966168870;  
r1wt(2) = 0.129484966168870;  
r1wt(3) = 0.279705391489277;  
r1wt(4) = 0.279705391489277;  
r1wt(5) = 0.381830050505119;  
r1wt(6) = 0.381830050505119;  
r1wt(7) = 0.417959183673469;
```

case 8

```
r1pt(1) = 0.960289856497536;  
r1pt(2) = -0.960289856497536;  
r1pt(3) = 0.796666477413627;  
r1pt(4) = -0.796666477413627;  
r1pt(5) = 0.525532409916329;  
r1pt(6) = -0.525532409916329;  
r1pt(7) = 0.183434642495650;  
r1pt(8) = -0.183434642495650;
```

```
r1wt(1) = 0.101228536290376;  
r1wt(2) = 0.101228536290376;  
r1wt(3) = 0.222381034453374;  
r1wt(4) = 0.222381034453374;  
r1wt(5) = 0.313706645877887;  
r1wt(6) = 0.313706645877887;  
r1wt(7) = 0.362683783378362;  
r1wt(8) = 0.362683783378362;
```

otherwise

```
disp('Order of quadrature to high for Gaussian Quadrature');
```

```
end % end of quadorder switch
```

```
n=1;
```

```
if ( sdim == 1 )
```

```

for i = 1:quadorder
    quadpoint(n,:) = [ r1pt(i) ];
    quadweight(n) = r1wt(i);
    n = n+1;
end

elseif ( sdim == 2 )
    for i = 1:quadorder
        for j = 1:quadorder
            quadpoint(n,:) = [ r1pt(i), r1pt(j)];
            quadweight(n) = r1wt(i)*r1wt(j);
            n = n+1;
        end
    end
end

else % sdim == 3
    for i = 1:quadorder
        for j = 1:quadorder
            for k = 1:quadorder
                quadpoint(n,:) = [ r1pt(i), r1pt(j), r1pt(k) ];
                quadweight(n) = r1wt(i)*r1wt(j)*r1wt(k);
                n = n+1;
            end
        end
    end
end

end

Q=quadpoint;
W=quadweight;
% END OF GAUSSIAN QUADRATURE DEFINITION

elseif ( strcmp(qt,'TRIANGULAR') == 1 )

    if ( sdim == 3 ) %%% TETRAHEDRA

```

```

if ( quadorder ~= 1 & quadorder ~= 2 & quadorder ~= 3 )
    % check for valid quadrature order
    disp('Incorect quadrature order for triangular quadrature');
    quadorder = 1;
end

```

```

if ( quadorder == 1 )
    quadpoint = [ 0.25 0.25 0.25 ];
    quadweight = 1;

```

```

elseif ( quadorder == 2 )
    quadpoint = [ 0.58541020 0.13819660 0.13819660;
                 0.13819660 0.58541020 0.13819660;
                 0.13819660 0.13819660 0.58541020;
                 0.13819660 0.13819660 0.13819660];
    quadweight = [1; 1; 1; 1]/4;

```

```

elseif ( quadorder == 3 )
    quadpoint = [ 0.25 0.25 0.25;
                 1/2 1/6 1/6;
                 1/6 1/2 1/6;
                 1/6 1/6 1/2;
                 1/6 1/6 1/6];
    quadweight = [-4/5 9/20 9/20 9/20 9/20]';

```

```

end

```

```

Q=quadpoint;
W=quadweight/6;

```

```

else %%% TRIANGLES

```

```

if ( quadorder > 7 ) % check for valid quadrature order
    disp('Quadrature order too high for triangular quadrature');

```

```
quadorder = 1;
end

if ( quadorder == 1 ) % set quad points and quadweights
quadpoint = [ 0.33333333333333, 0.33333333333333 ];
quadweight = 1;

elseif ( quadorder == 2 )
quadpoint = zeros( 3, 2 );
quadweight = zeros( 3, 1 );

quadpoint(1,:) = [ 0.16666666666667, 0.16666666666667 ];
quadpoint(2,:) = [ 0.66666666666667, 0.16666666666667 ];
quadpoint(3,:) = [ 0.16666666666667, 0.66666666666667 ];

quadweight(1) = 0.33333333333333;
quadweight(2) = 0.33333333333333;
quadweight(3) = 0.33333333333333;

elseif ( quadorder <= 5 )
quadpoint = zeros( 7, 2 );
quadweight = zeros( 7, 1 );

quadpoint(1,:) = [ 0.1012865073235, 0.1012865073235 ];
quadpoint(2,:) = [ 0.7974269853531, 0.1012865073235 ];
quadpoint(3,:) = [ 0.1012865073235, 0.7974269853531 ];
quadpoint(4,:) = [ 0.4701420641051, 0.0597158717898 ];
quadpoint(5,:) = [ 0.4701420641051, 0.4701420641051 ];
quadpoint(6,:) = [ 0.0597158717898, 0.4701420641051 ];
quadpoint(7,:) = [ 0.33333333333333, 0.33333333333333 ];

quadweight(1) = 0.1259391805448;
quadweight(2) = 0.1259391805448;
quadweight(3) = 0.1259391805448;
quadweight(4) = 0.1323941527885;
```

```
quadweight(5) = 0.1323941527885;  
quadweight(6) = 0.1323941527885;  
quadweight(7) = 0.2250000000000;
```

```
else
```

```
quadpoint = zeros( 13, 2 );  
quadweight = zeros( 13, 1 );
```

```
quadpoint(1,:) = [ 0.0651301029022, 0.0651301029022 ];  
quadpoint(2,:) = [ 0.8697397941956, 0.0651301029022 ];  
quadpoint(3,:) = [ 0.0651301029022, 0.8697397941956 ];  
quadpoint(4,:) = [ 0.3128654960049, 0.0486903154253 ];  
quadpoint(5,:) = [ 0.6384441885698, 0.3128654960049 ];  
quadpoint(6,:) = [ 0.0486903154253, 0.6384441885698 ];  
quadpoint(7,:) = [ 0.6384441885698, 0.0486903154253 ];  
quadpoint(8,:) = [ 0.3128654960049, 0.6384441885698 ];  
quadpoint(9,:) = [ 0.0486903154253, 0.3128654960049 ];  
quadpoint(10,:) = [ 0.2603459660790, 0.2603459660790 ];  
quadpoint(11,:) = [ 0.4793080678419, 0.2603459660790 ];  
quadpoint(12,:) = [ 0.2603459660790, 0.4793080678419 ];  
quadpoint(13,:) = [ 0.33333333333333, 0.33333333333333 ];
```

```
quadweight(1) = 0.0533472356088;  
quadweight(2) = 0.0533472356088;  
quadweight(3) = 0.0533472356088;  
quadweight(4) = 0.0771137608903;  
quadweight(5) = 0.0771137608903;  
quadweight(6) = 0.0771137608903;  
quadweight(7) = 0.0771137608903;  
quadweight(8) = 0.0771137608903;  
quadweight(9) = 0.0771137608903;  
quadweight(10) = 0.1756152576332;  
quadweight(11) = 0.1756152576332;  
quadweight(12) = 0.1756152576332;  
quadweight(13) = -0.1495700444677;
```

```
end

Q=quadpoint;
W=quadweight/2;
end

end

function X=square_node_array(pt1,pt2,pt3,pt4,numnod_u,numnod_v,uratio,vratio)

if ( nargin < 6 )
    disp(['Not enough parameters specified for quare_node_array function'])
elseif ( nargin == 6 )
    uratio=1;
    vratio=1;
elseif ( nargin == 7 )
    vratio=1;
end

if ( uratio == 1 )
    xi_pts=linspace(-1,1,numnod_u);
elseif ( uratio > 0 )
    ru=uratio^(1/(numnod_u-2));
    xi_pts(1)=0;
    d=1;
    for i=2:numnod_u
        xi_pts(i)=xi_pts(i-1)+d;
        d=d/ru;
    end
    xi_pts=2*xi_pts/xi_pts(numnod_u)-1;
else
    disp('uratio must be greater than 0');
    xi_pts=linspace(-1,1,numnod_u);
end
```

```
if ( vratio == 1 )
    eta_pts=linspace(-1,1,numnod_v);
elseif ( vratio > 0 )
    rv=vratio^(1/(numnod_v-2));
    eta_pts(1)=0;
    d=1;
    for i=2:numnod_v
        eta_pts(i)=eta_pts(i-1)+d;
        d=d/rv;
    end
    eta_pts=2*eta_pts/eta_pts(numnod_v)-1;
else
    disp('vratio must be greater than 0');
    eta_pts=linspace(-1,1,numnod_v);
end

x_pts=[pt1(1),pt2(1),pt3(1),pt4(1)];
y_pts=[pt1(2),pt2(2),pt3(2),pt4(2)];

for r=1:numnod_v
    eta=eta_pts(r);
    for c=1:numnod_u
        xi=xi_pts(c);
        N=lagrange_basis('Q4',[xi,eta]);
        N=N(:,1);
        X((r-1)*numnod_u+c,:)=x_pts*N,y_pts*N];
    end
end
```

Résumé

Ce travail porte sur le développement d'une nouvelle méthode numérique intitulée « Meshless local Petrov Galerkin (MLPG) combinée à la méthode des volumes finis (MVF) » appliquée au calcul de structures. Elle est basée sur la résolution de la forme faible des équations aux dérivées partielles par une méthode de Petrov Galerkin comme en éléments finis, mais par contre l'approximation du champ de déplacement introduite dans la forme faible ne nécessite pas de maillage. Seul un ensemble de nœuds est réparti dans le domaine et l'approximation du champ de déplacement en un point ne dépend que de la distance de ce point par rapport aux nœuds qui l'entourent et non de l'appartenance à un certain élément fini. Les déformations et les déplacements sont déterminés aux différents nœuds par interpolation locale en utilisant les moindres carrés mobiles (MLS). Les valeurs des déformations aux nœuds sont exprimées en termes de valeurs nodales interpolées indépendamment des déplacements, en imposant simplement la relation déformation déplacement directement par collocation aux points nodaux. La procédure de calcul pour cette méthode est implémentée dans un programme de calcul développé sous MATLAB. Le code obtenu a été validé sur un certain nombre de cas tests par comparaison avec des solutions analytiques de référence et des calculs éléments finis comme ABAQUS. L'ensemble de ces tests a montré un bon comportement de la méthode (environ 0.0001% d'erreurs par rapport à la solution exacte). L'approche est étendue pour l'étude des poutres minces et pour l'analyse dynamique et stabilité.

Mots clés : Méthode combinée, volumes finis, meshless local Petrov Galerkin, calcul de structures

Summary

This work concerns the development of a new numerical method entitled “Meshless Local Petrov-Galerkin (MLPG) combined with the Finite Volumes Method (FVM)” applied to the structural analysis. It is based on the resolution of the weak form of the partial differential equations by a method of Petrov Galerkin as in finite elements, but the approximation of the field of displacement introduced into the weak form does not require grid. The displacements and strains are given with the various nodes by local interpolation by using moving least squares (MLS). The values of the nodal strains are expressed in terms of interpolated nodal values independently of displacements, by simply imposing the strain displacement relationship directly by collocation at the nodal points. The procedure of calculation for this method is implemented in a computer code developed in MATLAB. The developed code was validated on a certain number of test cases by comparison with analytical solutions and finite elements results like ABAQUS. The whole of these tests showed a good behaviour of the method (about 0.0001% of errors in compared to the exact solution). The approach is also extended for the study of the thin beams and the dynamic analysis and stability.

Keywords: Combined method, Finite volume, Meshless local Petrov Galerkin, Calculation of structures