



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Hassiba Benbouali – Chlef

Faculté des Sciences et Sciences d'Ingénieur

Département de Mécanique

Mémoire de Magister

SPECIALITE : Mécanique

OPTION : Simulation Mécanique

Thème:

**Analyse hydrodynamique d'un piston de Moteur à
combustion interne par Simulation de Système
Multi Agents (SMA)**

Soutenu le : 09/11/2009

Présenté par:

TAHRI Samira

Jury :

Mrs LOUKARFI Larbi
BOUNIF Abdelhamid
MERIEM BENZIANE Madjid

Professeur (UHBC/FSSI)
Professeur (USTOran/FM)
Docteur (UHBCChlef/FSSI)

Président
Examineur
Examineur

TAHAR ABBES Miloud
HARBOUCHE Ahmed

MC (UHBCChlef/FSSI)
MACC(UHBCChlef/FSSI)

Directeur de thèse
Co-directeur de thèse

Promotion: 2005/2006

Résumé

Les travaux de recherche ont été effectués simultanément à la faculté des Sciences et des Sciences de la Technologie aux départements de Mécanique et de l'informatique à l'Université de Hassiba Ben Bouali de Chlef. L'encadrement a été assuré par Monsieur le docteur M. TAHAR ABBES et Mr A. HARBOUCHE.

Le but de ce mémoire de magister est d'étudier en utilisant les systèmes multi agents (SMA) le comportement hydrodynamique d'un piston de moteur à combustion interne.

Une application est faite sur le piston à jupe solide de moteur Deutz type F8L413, monté actuellement sur les camions TB230 de la SNVI de Rouiba (Algérie).

L'étude des systèmes multi agents passe par le développement et la compréhension du modèle mécanique. Ainsi dans une première partie, il est étudié le comportement hydrodynamique de l'ensemble jupe cylindre avec effet du film lubrifiant. Le mouvement global, défini par les excentricités du haut et du bas de la jupe, est décomposé en trois mouvements hydrodynamiques simples : les mouvements axial, latéral et de rotation.

La résolution du modèle est réalisée grâce au développement d'un code de calcul écrit en java. On détermine la pression hydrodynamique générée dans le film, les charges hydrodynamiques et les excentricités.

Dans une deuxième partie l'étude hydrodynamique est réalisée à l'aide d'un système multi agents dont la conception est basée sur le modèle mécanique suivant une méthodologie informatique définie.

La mise en pratique du système SMA est réalisée grâce à un réseau local formé de quatre PC prenant en charge la résolution de chacun des mouvements composés.

Le mouvement global est obtenu par la superposition de chacun des mouvements au niveau du PC serveur.

Mots clés :

Mouvement secondaire de piston , lubrification hydrodynamique, agent SMA.

Résumé

Les travaux de recherche ont été effectués simultanément à la faculté des Sciences et des Sciences de la Technologie aux départements de Mécanique et de l'informatique à l'Université de Hassiba Ben Bouali de Chlef. L'encadrement a été assuré par Monsieur le docteur M. TAHAR ABBES et Mr A. HARBOUCHE.

Le but de ce mémoire de magister est d'étudier en utilisant les systèmes multi agents (SMA) le comportement hydrodynamique d'un piston de moteur à combustion interne.

Une application est faite sur le piston à jupe solide de moteur Deutz type F8L413, monté actuellement sur les camions TB230 de la SNVI de Rouiba (Algérie).

L'étude des systèmes multi agents passe par le développement et la compréhension du modèle mécanique. Ainsi dans une première partie, il est étudié le comportement hydrodynamique de l'ensemble jupe cylindre avec effet du film lubrifiant. Le mouvement global, défini par les excentricités du haut et du bas de la jupe, est décomposé en trois mouvements hydrodynamiques simples : les mouvements axial, latéral et de rotation.

La résolution du modèle est réalisée grâce au développement d'un code de calcul écrit en java. On détermine la pression hydrodynamique générée dans le film, les charges hydrodynamiques et les excentricités.

Dans une deuxième partie l'étude hydrodynamique est réalisée à l'aide d'un système multi agents dont la conception est basée sur le modèle mécanique suivant une méthodologie informatique définie.

La mise en pratique du système SMA est réalisée grâce à un réseau local formé de quatre PC prenant en charge la résolution de chacun des mouvements composés.

Le mouvement global est obtenu par la superposition de chacun des mouvements au niveau du PC serveur.

Mots clés :

Mouvement secondaire de piston , lubrification hydrodynamique, agent SMA.

Remerciements

Je remercie Dieu tout puissant de m'avoir permis d'entamer et d'achever ce travail.

Cette étude a été effectuée sous la direction du docteur M. TAHAR ABBES de l'Université Hassiba Ben Bouali de Chlef. La partie informatique a été effectuée dans la salle réseau du département d'informatique sous la direction de Mr A. HARBOUCHE.

Je remercie Monsieur le docteur M. TAHAR ABBES pour m'avoir permis d'entreprendre cette recherche et pour l'aide qu'il m'a apporté durant mes travaux, sans oublier Mr A. HARBOUCHE pour son aide et ses orientations.

Je tiens à remercier tout particulièrement les membres de mon jury pour l'intérêt qu'ils ont porté à mon travail. Je remercie également Mr LOUKARFI Larbi de m'avoir fait l'honneur de présider le jury.

Mes sincères remerciements s'adressent :

- Aux professeurs du département de mécanique et du département d'informatique qui m'ont assistés et guidés tout au long de ma modeste recherche.
- A Mr.Omar El hadj le chef du département d'informatique.
- A tous mes camarades ingénieurs qui m'ont aidés et qui se reconnaîtront d'eux mêmes.
- A Mr. Mahfoudh M et ses collègues de travail pour m'avoir aidé.

Merci à tous les personnes, nombreuses, qui m'ont aidé et encouragé jusqu'à l'achèvement de ce travail de recherche.

Je remercie tout spécialement mes parents pour m'avoir activement apporté un soutien inconditionnel tout au long de mes études. Que ce travail soit leur récompense tant méritée.

Je remercie chaleureusement ceux avec qui j'ai partagé des moments inoubliables mes sœurs Salima, Naffissa , Fatima zohra et Imane et mon petit frère Ghoulem Ellah, et toute ma grande famille.

Je termine par une pensée particulière pour mon mari, Je te remercie de tout mon coeur pour avoir cru en moi durant les moments difficiles, pour m'avoir soutenu et épauler, pour ton sacrifice permanent lors de ces longs mois de programmation.

Je dédie ce mémoire à mon petit ange *Mohammed Amine*, mes chers parents.

TABLE DES MATIERES

Résumé	1
Remerciements	2
Table des matières	3
Liste des figures	6
Liste des tableaux	8
Nomenclature	9
Introduction générale	11
CHAPITRE 1 : Etude bibliographique	12
1.1 La simulation mécanique du modèle	12
1.1.1 L'hydrodynamique en général.....	12
a)- Introduction.....	12
b)- Définition de la Tribologie.....	12
1.1.2 Lubrification hydrodynamique.....	13
1.1.2.1 La lubrification fluide.....	13
1.1.2.2 Lubrification hydrodynamique.....	13
1.1.3 Modèle Mécanique.....	13
1.1.3.1 Introduction.....	13
1.1.3.2 Le modèle hydrodynamique.....	14
1.2 La simulation informatique du modèle	17
CHAPITRE 2 : Problématique et équations de base	21
2. Problématique et équations de base.....	21
2.1 Introduction.....	21
2.2 Problématique.....	21
2.2.1 Problématique mécanique.....	21
2.2.2 Problématique informatique.....	25

2.3	Equations de base du mouvement.....	26
2.3.1	Le mouvement hydrodynamique axial.....	26
2.3.2	Le mouvement hydrodynamique de translation.....	28
2.3.3	Le mouvement hydrodynamique de rotation.....	29
2.3.4	Le mouvement hydrodynamique global.....	30
3.2.5	Equations des charges hydrodynamiques.....	32
3.2.5.1	Charges hydrodynamiques des mouvements composés.....	32
	CHAPITRE 3 : Les Systèmes Multi Agents.....	36
3.1	Système Multi agents.....	36
3.2	L'agent.....	38
3.2.1	Caractéristique d'un agent.....	39
3.2.2	Typologie des agents.....	40
3.2.3	Comportement d'un agent.....	42
3.2.3.1	Environnement	42
3.2.3.2	Interaction.....	43
3.2.3.2.1	Communication.....	44
3.2.3.2.2	Négociation.....	52
3.2.3.2.3	Coopération.....	53
3.2.3.2.4	Planification.....	54
3.2.3.3	Contrôle.....	54
3.2.3.4	Organisation.....	55
	CHAPITRE 4 : Méthode de résolution mécanique.....	56
4.	Le modèle hydrodynamique.....	56
	Méthode de résolution.....	56
	Algorithme de résolution du modèle hydrodynamique.....	58
	Algorithme de Newton-Raphson modifié.....	61
	Organigramme des mouvements composés.....	63
	Organigramme de calcul de la pression.....	68
	Calcul des charges hydrodynamiques.....	69
	Organigramme de la solution avec effet des forces d'inertie.....	70
	CHAPITRE 5 : Choix de la méthodologie et conception du SMA.....	73
5.1	Choix de la méthodologie des SMA.....	73

5.1.1 Introduction.....	73
5.1.2 La Méthodologie Aalaadin	74
5.1.2.1 Concepts centraux.....	74
5.1.2.2 Niveau méthodologique.....	74
5.1.2.3 Processus de développement.....	75
5.2 Conception du Système Multi Agents.....	76
5.2.1 Diagramme de classe.....	76
5.2.2 Diagramme d'activités de l'Agent Mouvement_Global.....	79
5.2.3 Diagramme d'activités de l'Agent Mouvement_Axial.....	81
5.2.4 Diagramme d'activités de l'Agent Mouvement_Latéral.....	83
5.2.5 Diagramme d'activités de l'Agent Mouvement_rotation.....	85
5.2.6 Diagramme d'activité de rôle.....	89
5.2.6.1 Diagramme d'activité du rôle calcule de pression.....	89
5.2.6.2 Diagramme d'activité du rôle calcule des charges hydrodynamiques	90
CHAPITRE 6 : Implémentations de méthode mécanique et SMA : Résultats et discussion.....	91
6.1 Introduction.....	91
6.2 Simulation des agents.....	91
6.3 Programme principal.....	92
6.4 Communication du Système Multi Agents.....	108
6.5 Déroulement de la communication.....	109
6.6 Exemple d'application de notre Système Multi Agents.....	111
6.7 Affichage des résultats.....	117
6.7.1 Le mouvement global.....	117
6.7.2 Calcul des excentricités.....	123
6.8 Résultats et discussion.....	129
6.9 L'apport de la simulation des Systèmes Multi Agents au modèle mécanique...	134
CONCLUSION GENERALE.....	136
PERCPECTIVES.....	138
REFERENCES BIBLIOGRAPHIQUES	139

Liste des figures

N° de figure	Nom de figure	page
Figure 1.1	Etablissement du frottement sec au démarrage	16
Figure 1.2	Etablissement du régime hydrodynamique au-delà d'une vitesse ω	17
Figure 2.1	Représentation schématique du mouvement du piston (a) et des charges appliquées (b)	20
Figure 2.2	Condition de pression dans la zone inactive	22
Figure 2.3	Zone de pression active avec vitesse positive	24
Figure 2.4	Profil adopté de la jupe	25
Figure 2.5	Le mouvement de translation latéral du piston	27
Figure 2.6	Le mouvement hydrodynamique de rotation pure	28
Figure 2.7	Le mouvement hydrodynamique global	29
Figure 2.8	Forces hydrodynamiques du mouvement axial	31
Figure 2.9	Forces hydrodynamiques du mouvement latéral	31
Figure 2.10	Forces hydrodynamiques du mouvement de rotation	32
Figure 2.11	Forces hydrodynamiques globales	33
Figure 3.1	Système multi-agents	36
Figure 3.2	Structure générale d'un agent dans un système multi-agents	39
Figure 3.3	Architecture d'un agent cognitif	41
Figure 3.4	Architecture hybride en couches verticales	42
Figure 3.5	Communication par envoi de messages	46
Figure 4.1	Organigramme principal de résolution	59
Figure 4.2	Organigramme schématique de la solution cyclique	60
Figure 4.3	Organigramme de Newton-Raphson modifié	62
Figure 4.4	Organigramme de calcul de la pression globale	66
Figure 4.5	Organigramme du mouvement hydrodynamique axial	67
Figure 4.6	Méthode SOR de calcul de la pression hydrodynamique globale	68
Figure 4.7	Organigramme de calcul des charges \bar{F} et \bar{M}	69
Figure 4.8	Organigramme de la solution avec effet d'inertie	72
Figure 5.1	Le méta-modèle de la méthodologie Aalaadin	75
Figure 5.2	Modèle AGR	76
Figure 5.3	Diagramme de classe	78
Figure 5.4	Diagramme d'activités de l'Agent Mouvement_Global	80

Figure 5.5	Diagramme d'activités de l'Agent Mouvement_Axial	82
Figure 5.6	Diagramme d'activités de l'Agent Mouvement_Latéral	84
Figure 5.7	Diagramme d'activités de l'Agent Mouvement_Rotation	87
Figure 5.8	Diagramme d'activité du rôle calcule de pression	89
Figure 5.9	Diagramme d'activité du rôle calcule des charges hydrodynamiques	90
Figure 6.1	Schéma du Système Multi Agents du modèle mécanique sous réseau	108
Figure 6.2	Schéma du déroulement de la communication entre les agents	109
Figure 6.3	Champ de pression P_u ($P_{max}=0.0145$)	118
Figure 6.4	Champ de pression P_u ($P_{max}=0.0145$), [10]	118
Figure 6.5	Pression P_{1c} ($P_{max}=0.109$)	119
Figure 6.6	Pression P_{1c} ($P_{max}=0.09$), [10]	119
Figure 6.7	Pression P_{2c} ($P_{max}= 0.024$)	120
Figure 6.8	Pression P_{2c} ($P_{max}= 0.019$), [10]	120
Figure 6.9	Pression globale \bar{P} ($P_{max}=0.125$)	121
Figure 6.10	Pression globale \bar{P} ($P_{max}=0.104639$), [10]	121
Figure 6.11	Configuration du mouvement adopté dans l'exemple	122
Figure 6.12	Forces hydrodynamiques de la configuration de la figure 6.11	122
Figure 6.13	Pression des gaz de combustion sur un cycle moteur	123
Figure 6.14	Force d'inertie axiale \bar{F}_{IP} et accélération axiale du piston	123
Figure 6.15	Progression de la convergence de la solution dans le cas où $C_p=0mm$	124
Figure 6.16	Progression de la convergence de la solution dans le cas où $C_p=0mm$, [10]	125
Figure 6.17	a) Pression des gaz, b) Pression des gaz, [10]	126
Figure 6.18	Progression de la solution dans le cas où $C_p=1.5mm$	126
Figure 6.19	Progression de la solution dans le cas où $C_p=1.5mm$, [10]	127
Figure 6.20	Mouvement secondaire avec prise en considération de l'inertie du piston ($c = 35\mu$)	128
Figure 6.21	Mouvement secondaire avec prise en considération de l'inertie du piston ($c = 20\mu$)	128
Figure 6.22	Effet du jeu radial sur le mouvement secondaire.	129
Figure 6.23	courbe du résultats du temps/type programmation, $k=1$	133
Figure 6.24	Histogramme du temps d'exécution, $k=1$	133
Figure 6.25	courbe du résultats du temps/type programmation, $k=720$	133
Figure 6.26	Histogramme du temps d'exécution, $k=720$	133
Figure 6.27	courbe des résultats de temps d'exécution/ nombre d'itération	134

Liste des tableaux

N° de tableau	Nom de tableau	page
Tableau 6.1	Données de base utilisées dans la simulation	117

Dictionnaire des mots clés

MOT	SIGNIFICATION
JADE	Java Agent DEvelopment Framework, est une plateforme Java de développement et d'exécution de SystemeMultiAgents.
JAVA	Langage de programmation orienté objet crée par la société Sun.
SMA	Système multi agents
Socket	permettent l'échange de données entre 2 processus
TCP/IP	Un service réseau TCP/IP est accessible par un port ce qui permet d'aiguiller les données vers le processus dans le SE.
Threads	Processus légers qui s'exécutent en même temps

Nomenclature

a	Quantité distance entre l'axe et le haut du piston (m)
c	Jeu radial entre le piston et le cylindre (à température ambiante) (m)
C_g	Désaxage du centre de gravité (m)
C_p	Désaxage (offset piston-axe) (m)
e_h, e_b	Excentricité du haut et du bas respectivement de la jupe (m)
F	Force hydrodynamique développée dans le film sépare le piston du cylindre
h	Epaisseur du film lubrifiant
I_{pist}	Moment d'inertie du piston par rapport à l'axe ($Kg.m^2$)
L	Longueur de la jupe du piston (m)
M	Moment hydrodynamique par rapport à l'axe (N.m)
M_{IC}	Moment d'inertie du piston (N.m)
m_{axe}	Masse de l'axe (Kg)
m_{pist}	Masse du piston (Kg)
p	Pression hydrodynamique dans le film (Pa)
P_g	Pression des gaz de combustion (Pa)
R	Rayon du piston (m)
r	Rayon du vilebrequin (m)
t	temps (s)
U	Vitesse axiale du piston (m/s)
y	coordonnée lié au piston mesurée à partir du haut de la jupe du piston (m)
μ	Viscosité dynamique de l'huile (Pa.s)
α	Angle fluide (rad)
ω	Vitesse de rotation du moteur (s^{-1})

Quantités adimensionnelles

$$\bar{a} = \frac{a}{R}$$

$$\bar{F} = \frac{Fc^2}{6\mu r \omega R^3}$$

$$\bar{h} = \frac{h}{c}$$

$$\bar{L} = \frac{L}{c}$$

$$\bar{M} = \frac{Mc^2}{6\mu r \omega R^4}$$

$$\bar{p} = \frac{pc^2}{6\mu r \omega R}$$

$$\bar{t} = t\omega$$

$$\bar{U} = \frac{U}{r\omega}$$

$$\bar{y} = \frac{y}{R}$$

$$\beta = \frac{2R}{r}$$

$$\varepsilon_h = \frac{e_h}{c}$$

$$\varepsilon_b = \frac{e_b}{c}$$

Introduction Générale

Cette étude entre dans le cadre de la simulation du modèle de lubrification hydrodynamique d'un piston de moteur à combustion interne MCI par une analyse mécanique dans une 1^{ière} partie et des **Systèmes Multi Agents SMA** dans une 2^{ième} partie.

Ce modèle tient en compte de l'effet de la lubrification hydrodynamique sur le mouvement dynamique du piston. Le comportement dynamique du piston résulte de la composition d'un mouvement oscillatoire axial (mouvement primaire) et d'un mouvement latéral (mouvement secondaire) qui se compose d'un mouvement de translation, perpendiculaire à l'axe du cylindre et d'un mouvement de rotation (ou de basculement) du piston autour de son axe. Le mouvement dynamique secondaire résulte du déséquilibre des forces et des moments agissant sur le piston dans le plan normal à l'axe et contenant les cotés de poussée latérale maximum et minimum. Ce mouvement est défini par les excentricités du haut et du bas de la jupe.

Différents travaux ont été menés dans le développement de ce modèle au début des années 80. Knoll et Peeken [27] ont considéré la lubrification de la jupe comme un problème isolé c'est à dire sans prendre en considération la dynamique du piston.

Ils présentent une analyse de base sur l'inclusion de l'effet des forces de lubrification hydrodynamique sur le mouvement primaire, défini par la vitesse transversale de translation et la vitesse de rotation autour de l'axe du piston. Le comportement du modèle à la lubrification est décrit comme un système de palier hydrodynamique. Dans cette étude l'hypothèse de base est que la capacité des forces hydrodynamiques de supporter la jupe soit le résultat de l'intégration de la répartition de la pression dans le film d'huile. L'apport de ce modèle fait ressortir la différence entre les valeurs expérimentales et théoriques du mouvement transversal analysé dans les études précédentes et qui omettaient les forces hydrodynamiques agissant sur la jupe.

Alors notre objectif est de simuler le mouvement hydrodynamique global on appliquant une nouvelle approche informatique qui est les SMA et améliorer les résultats obtenus par le modèle numérique.

Les **Systèmes Multi Agent** sont un domaine relativement jeune, apparu dans les années 80. Ils font intervenir un ensemble d'entités pro-actives et relativement indépendantes appelés «Agents» et dotés d'un comportement intelligent, et qui communiquent dont l'objectif est de coordonner leurs buts et leurs plans d'actions pour atteindre un objectif global qui est la résolution du problème. C'est une technique utilisable dans de nombreuses disciplines notamment en simulation.

Chapitre 1

ETUDE BIBLIOGRAPHIQUE

1.1 La simulation mécanique du modèle

1.1.1 L'hydrodynamique en général

a)- Introduction

L'étude scientifique de la tribologie a une longue histoire, et plusieurs des lois fondamentales du frottement limité, telles que la proportionnalité entre la force appliquée et la force de frottement, sont pensées pour avoir été développées par Leonardo da Vinci vers la fin du 15^{ème} siècle. Cependant, l'effet du frottement et de l'usure a été omis pendant plusieurs siècles avec seulement des concepts simplifiés pour expliquer les mécanismes fondamentaux. En 1699 Amontons a proposé que des surfaces ont été couvertes par de petites sphères et que le coefficient de frottement était un résultat de l'angle du contact entre les sphères des surfaces de contact. Une valeur raisonnable de coefficient de frottement près de 0.3 a été donc trouvée en supposant que le mouvement se fait par contact avec des sphères. C'est vers la fin du 18^{ème} siècle, en 1886 avec la publication d'Osborne Reynolds; qui sont apparus les concepts de l'hydrodynamique. Reynolds a montré que la pression hydrodynamique du liquide entraînée entre les surfaces de glissement était suffisante pour empêcher le contact entre les surfaces même aux vitesses coulissantes très basses. Sa recherche a eu l'application pratique immédiate et mène au déplacement d'un trou d'huile de la ligne de charge des roulements ferroviaires d'axe. L'huile, au lieu d'être vidangé loin par le trou, pouvait maintenant produire un film hydrodynamique et le frottement a aussi diminué. Le travail de Reynolds lance d'autres efforts de recherches visant d'améliorer l'interaction entre deux surfaces de contact. L'usure et les principes fondamentaux du frottement sont des problèmes plus complexes, la recherche expérimentale sur lesquels dépend des moyens expérimentaux utilisés.

b)- Définition de la tribologie

Le mot tribologie, construit à partir des racines grecques **tribein** (frotter) et **logos** (discours, étude), a été proposé en 1968 par G. Salomon pour désigner la science qui étudie les phénomènes susceptibles de se produire lorsque deux corps en contact sont animés de mouvements relatifs. Il recouvre, entre autres, tous les domaines du **frottement** proprement dit, de l'**usure** et de la **lubrification**. En termes simples il s'avère que l'objectif pratique de la tribologie est de réduire au minimum les deux inconvénients principaux du solide au contact plein : le frottement et l'usure, mais ce n'est pas toujours le cas. Dans quelques situations, minimiser le frottement et maximiser de l'usure ou minimiser de l'usure et maximiser le frottement ou maximiser du frottement et de l'usure est souhaitable. Par exemple, la réduction d'usure mais pas de frottement est souhaitable dans les freins et des embrayages lubrifiés, la réduction de frottement mais pas d'usure est souhaitable dans des crayons, augmentation de frottement et l'usure est souhaitable dans des gommes à effacer.

1.1.2 Lubrification hydrodynamique

1.1.2.1 La lubrification fluide

Historiquement, l'étude de la lubrification a commencé avec l'ère de la révolution industrielle.

C'est à cette époque que Reynolds a développé sa théorie. Cependant, bien avant, Léonard de Vinci et Coulomb s'étaient intéressés au frottement. Ce dernier a mis en évidence la relation de proportionnalité entre la charge appliquée et la force de frottement.

L'objectif de la lubrification est de séparer des surfaces en mouvement relatif à l'aide d'un fluide qui n'endommagera pas ces surfaces. La résistance au cisaillement de ce matériau doit causer le moins de résistance possible. Si la lubrification fluide a été longuement étudiée, celle basée sur la graisse, ou encore la lubrification sèche ne l'ont été que bien plus récemment. Il n'existe pas de formalisme mathématique en décrivant physiquement les mécanismes.

1.1.2.2 Lubrification hydrodynamique

La lubrification hydrodynamique est un domaine important de la tribologie, c'est l'étude des contacts dans lesquels un film de fluide sépare les surfaces en présence. Dans le cas où le film de fluide sépare totalement les surfaces, les aspérités et les défauts de forme ont des dimensions inférieures à l'épaisseur du film.

Dans le cas de la lubrification des parois du système piston cylindre peut être décrite comme un ensemble de surfaces lubrifiées. Les charges hydrodynamiques supportant les charges dynamiques latérales résultent de l'intégration du champ de pression né dans les zones actives du film d'huile.

Le champ de pression est régi par l'équation de Reynolds et qui fait partie de la MDF.

1.1.3 Modèle mécanique

1.1.3.1 Introduction

Le modèle est analysé sans prendre en considération l'effet de la lubrification sur le mouvement dynamique. Il a été développé dans le but de déterminer le frottement et le bruit moteur puis de réduire leur portée.

A l'origine, puisqu'il est difficile de mesurer la translation du piston, les méthodes numériques n'ayant pas fait leur apparition, des méthodes analytiques sont utilisées pour la prédiction du bruit provoqué par les pistons. En 1953, Burrell G. et F.G. Butler [1] présentent une méthode analytique sur la dynamique du piston et déterminer la position de l'axe du piston qui puisse réduire l'impact du piston contre la paroi du cylindre. Ils concluent, qu'avec la longueur de bielle et la position de l'axe du piston prédéterminé, la seule variable par laquelle le clappement du piston est affecté est le désaxage (offset du piston) de l'axe du piston par rapport à la ligne médiane du piston.

En 1972-73, A. M. Laws et al. [2] présentent une analyse expérimentale combinée avec une analyse par ordinateur du mouvement latéral (transversal) et de rotation du piston. Les résultats obtenus sont représentés par les vibrations du bloc moteur et le jeu piston-cylindre. Ces vibrations causées, d'après l'auteur, par l'impact de la force latérale, sont analysées dans le but d'atténuer le clappement du piston.

La deuxième partie concerne la réduction du frottement mécanique car le piston est connu comme étant une source importante génératrice de frottement. Ces frottements se manifestent aussi bien au niveau de la jupe par sa double fonction d'appui et de guidage qu'au niveau des segments. Des modèles de frottement furent mis au point. Ces modèles s'appuient principalement sur des procédures expérimentales. L'analyse du mouvement du piston est ignorée dans ces travaux.

Les premières mesures de frottement de pistons de moteur à combustion interne sont réalisées par Forbès et Taylors et datent de 1943 [3]. S. Furuhashi et al. [4-6]. Tagiguchi et al. [7] surmontent les difficultés expérimentales et réussissent à mesurer les forces de frottement piston-cylindre dans les conditions de fonctionnement du moteur. Uras et Paterson utilisent la méthode 'IMEP' [8] (méthode qui utilise l'équilibre des forces appliquées sur le corps isolé du piston) pour déterminer les frottements instantanés. Hoshi et Baba [9] présentent une méthode théorique pour déterminer les pertes par frottement hydrodynamique de l'ensemble piston-segments. L'équipement conçu 'motoring method' permet le fonctionnement de l'ensemble bielle-manivelle-piston d'être entraîné par un moteur électrique sans moteur thermique. Ils montrent que la moitié des pertes par frottements dans un moteur à combustion interne le sont par le système piston dont 70 à 80 pour cent sont par les segments de piston et le reste par la jupe. Il est en général difficile de savoir à quelle partie de l'ensemble piston-cylindre, les frottements sont attribués aux segments ou à la jupe. [10]

1.1.3.2 Le Modèle hydrodynamique

Les équations de base de la lubrification hydrodynamique des paliers ont été établies pour les systèmes de paliers par Osborn Reynolds [11-14] en 1886 et constituent la base de la lubrification hydrodynamique. L'analyse hydrodynamique se définit par le fait que la pression développée dans le film d'huile n'a pas d'effet sur les surfaces en contact.

Les études sur la lubrification hydrodynamique du piston de moteur à combustion interne, dont la majorité a trait à la lubrification des segments [15-23], débutent avec l'analyse de la lubrification de la jupe, par des procédés expérimentaux employant des méthodes de visualisation. Ainsi l'analyse expérimentale de pistons dans des cylindres transparents [24, 25] permet de visualiser le comportement du piston sous l'effet du lubrifiant. Il est montré, à l'aide de traces laissées par le film d'huile, l'existence de la force d'impact. Il est montré [LI CH] que l'injection de lubrifiant sous pression vers la jupe réduit

substantiellement le bruit émis dans le moteur Diesel. Il est connu que la jupe a une double fonction: guidage du piston dans le cylindre et supporter l'impact de la force de poussée dynamique latérale (de l'anglais 'thrust side') due au mouvement secondaire. Il est montré que le film d'huile amortit le claquement du piston donc réduit le bruit moteur. En plus le film d'huile assure une fonction de protection des surfaces de contact contre le grippage. Ainsi il est montré que le mouvement du piston et le clappement qui en résulte sont donc liés directement à la lubrification de la jupe [1, 2, 26]. Dans ces analyses les auteurs tentent de prédire l'impact du piston entre la paroi du cylindre basés sur les équations d'équilibre; cependant de bons accords avec les mesures ne sont pas achevés à cause du manque de considérations propres sur la lubrification hydrodynamique.

Les études qui prennent en considération l'effet de la lubrification du piston ne prennent de l'ampleur qu'au début des années 80. Knoll et Peeken [27] développent un modèle de lubrification hydrodynamique basé sur l'équation de Reynolds couplée avec les équations de mouvement du piston. Dans ce modèle les forces hydrodynamiques dues au film d'huile entre la jupe et la paroi du cylindre pour une position et une vitesse donnée du piston sont calculées. Cette étude fournit une base pour l'inclusion de l'effet des forces de lubrification hydrodynamique sur le mouvement primaire et secondaire des pistons

Li et al [28] performant une analyse dynamique du piston dans laquelle ils incorporent un modèle de lubrification hydrodynamique de la jupe. La trajectoire complète du piston et les forces de frottement en fonction des conditions de fonctionnement du moteur sont ainsi calculées. Les résultats de cette étude indiquent que l'inclinaison dynamique d'un piston de moteur à combustion interne influence le comportement au frottement de la jupe de piston. Cette étude constitue notre référence de base pour l'analyse hydrodynamique du piston.

Mourelatos [29] présente un modèle théorique simplifié de piston, sans segments, de moteur LHR 'Low-Heat Rejection (à 'faible rejet de chaleur') à combustion interne typique. La particularité de ce moteur, dont la lubrification se fait au moyen de gaz de combustion est que les frottements inter paroi sont très faibles. L'auteur résout l'équation de Reynolds compressible à deux dimensions par la méthode des éléments finis. Le modèle ainsi construit est résolu en le simulant par des forces dynamiques latérales données. La trajectoire du piston est obtenue à partir d'une solution quasi-statique qui ne prend pas en considération les équations de mouvement du piston. Il conclut que le piston sans segments ne peut supporter que de petites charges latérales et que le profil à côtés doubles est plus performant que le profil à côté unique.

Gommed et Etsion [30] développent un modèle mathématique pour l'analyse dynamique de piston, sans segments, de moteur de type LHR identique à celui présenté par Mourelatos, lubrifié par gaz. Le système d'équations complètes de mouvement du piston et de la bielle sont formulées simultanément avec l'équation de Reynolds pour les gaz et les équations d'énergie. Ils montrent que l'incorporation de la

dynamique de la bielle joue un rôle important dans le mouvement du piston. Ils trouvent que la stabilité du mouvement du piston dépend fortement des conditions thermohydrodynamiques de fonctionnement.

Plus récemment Liu et al [31] présentent un modèle de lubrification mixte hydrodynamique basé sur l'équation de Reynolds à deux dimensions couplée avec les équations de mouvement secondaire du piston. Cette étude, basée sur le modèle de Li, tient compte de l'effet de la rugosité des surfaces en contact. Les résultats obtenus montrent que le profil de la jupe, la vitesse du moteur, le désaxage (offset) de l'axe du piston ainsi que le jeu radial piston-cylindre jouent un rôle important dans la détermination du mouvement secondaire du piston. Ils concluent aussi que pour des vitesses élevées et un faible jeu le mouvement secondaire tend à s'atténuer.

La complexité du modèle de lubrification de la jupe du piston fait que ce sujet continue d'être traité. Siyoul [32] présente une étude du mouvement dynamique secondaire avec incorporation de la lubrification. La procédure de construction et de résolution du modèle se base essentiellement sur l'analyse de Li. Il axe son étude sur les paramètres qui influencent le mouvement secondaire et principalement le profil de la jupe. Les résultats trouvés montrent que le profil en tonneau de la jupe est le profil qui donne la force d'impact la plus faible.

Au démarrage du moteur le frottement est métal-métal (figure 1.1), avec l'augmentation de la vitesse du piston dans le cylindre il se forme un film d'huile au-delà d'une certaine vitesse : le régime hydrodynamique du piston est alors atteint (figure 1.2).

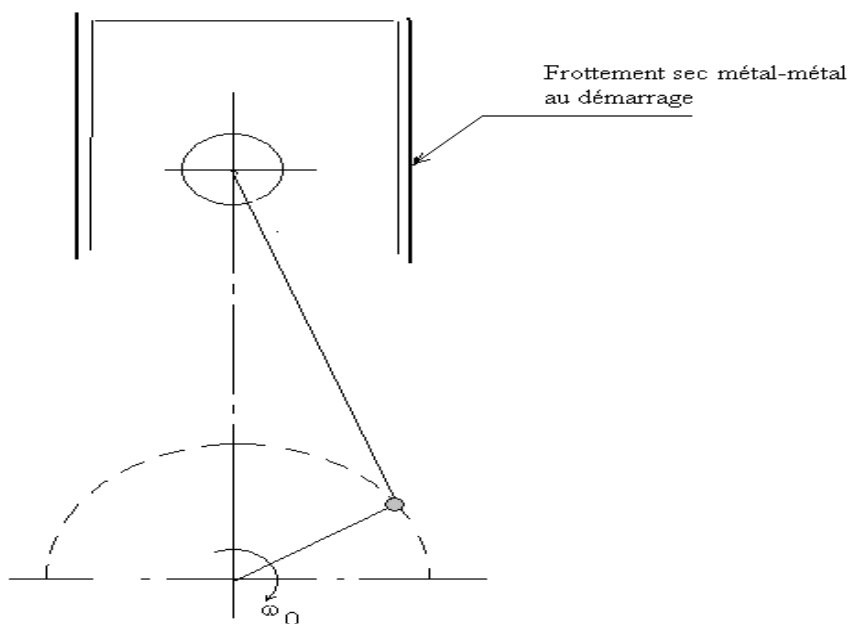


Figure 1.1 : Etablissement du frottement sec au démarrage

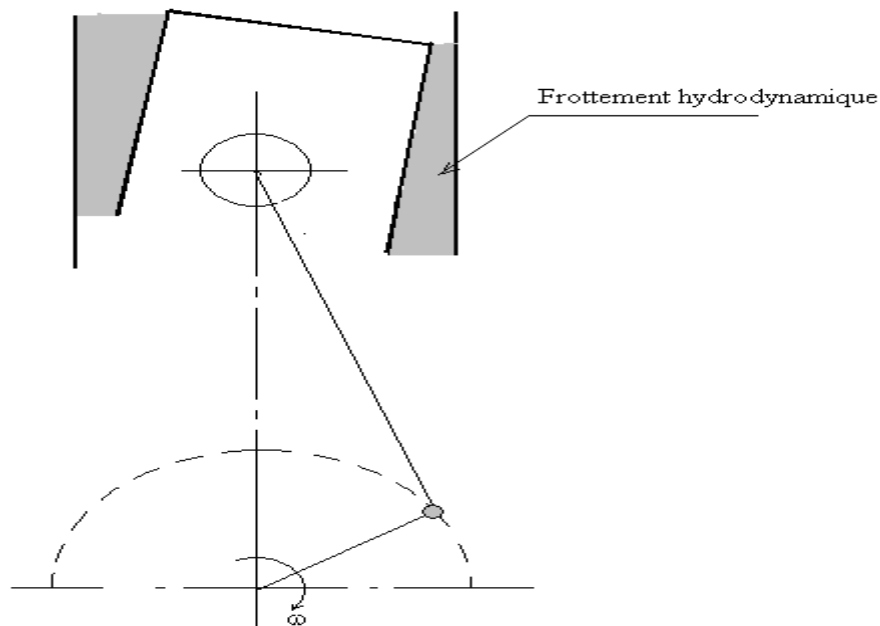


Figure 1.2 : Etablissement du régime hydrodynamique au-delà d'une vitesse ω

1.2 La simulation informatique du modèle

Actuellement les systèmes informatiques deviennent de plus en plus complexes et répartis sur différents sites. La nécessité de passer du comportement individuel au comportement collectif, a poussé la communauté de l'Intelligence Artificielle classique à se tourner vers des systèmes plus autonomes, plus coopérants afin de pouvoir résoudre des problèmes complexes.

L'intelligence artificielle cherche à rendre les ordinateurs doués, dans une certaine limite, de raison et de faculté d'action en fonction de leur raisonnement. Les recherches issues de ce domaine tendent à créer des entités intelligentes autonomes pouvant rivaliser d'une certaine manière avec l'être humain dans des domaines précis et résolvant des problèmes complexes qui nécessitaient encore l'homme pour son savoir-faire et ses facultés intellectuelles.

L'intelligence artificielle classique consiste à modéliser le comportement intelligent d'un seul agent.

L'intelligence artificielle distribuée IAD s'intéresse à des comportements intelligents qui sont le produit de l'activité coopérative de plusieurs entités autonomes (appelées agents).

La résolution d'un problème en IAD se fait grâce à ces entités qui se coordonnent et interagissent afin de tirer le meilleur de leurs caractéristiques et de leurs savoir-faire individuels pour une résolution collective du problème.

La résolution ne consiste pas seulement à l'exécution d'une entité informatique, mais d'un système composé de plusieurs entités informatique. De plus, cette résolution ne se contente pas d'être la somme

des résolutions partielles de chaque entité composant le système mais de combiner ces résolutions par des interaction entre les agents.

Les principes de base de l'IAD constituent en la distribution des connaissances et des informations nécessaires à la résolution d'un problème parmi les différentes composantes en interaction d'un système. Les composantes de ces systèmes sont des entités plus au moins intelligentes désignées par le terme « agent ».

Il existe trois axes fondamentaux (thèmes) de la recherche en IAD :

- **Système multi-agents (SMA)** : il s'agit de faire coopérer un ensemble d'agents dotés d'un comportement intelligent et de coordonner leurs plans d'action pour la résolution d'un problème.
- La résolution distribuée de problème (RDP) : elle s'intéresse à la manière de diviser un problème particulier sur un ensemble d'entités distribuées et coopérantes. Elle s'intéresse aussi à la manière de partage de la connaissance du problème et d'en obtenir la solution.
- L'intelligence Artificielle parallèle (IAP) : elle concerne le développement de langages et d'algorithmes parallèles pour l'IAD. L'IAP vise l'amélioration des performances des systèmes d'intelligence artificielle sans toutefois, s'intéresser à la nature du raisonnement ou au comportement intelligent d'un groupe d'agents. Cependant il est vrai que le développement de langages concurrents et d'architectures parallèle peut avoir un impact important sur les systèmes d'IAD.

Avant les années 80, les ordinateurs étaient encombrants et chers (les systèmes centralisés), mais à partir de la mi-80, deux nouveautés qui sont les microprocesseurs (moins chers et très rapide) et les LANs et les WANs.

Les ordinateurs en réseaux non seulement faisables, mais simples. Donc Les systèmes distribués sont un besoin réel dans l'informatique d'aujourd'hui !

Les réseaux d'ordinateurs sont partout (Internet, réseaux de téléphones mobiles, réseaux locaux, etc.).

Les progrès technologiques des ordinateurs et le haut débit permettent aux différentes applications sur des ordinateurs distincts (et distants) de coopérer pour effectuer des tâches coordonnées.

L'informatique répartie concerne les applications parallèles tournant sur plusieurs machines reliées par un réseau de communication.

Les systèmes distribués sont définis comme suit :

1. [Tanenbaum]: Un ensemble d'ordinateurs indépendants qui apparaît à un utilisateur comme un système unique et cohérent où les machines sont autonomes et les utilisateurs ont l'impression d'utiliser un seul système.

2. [Lamport]: A distributed system is one on which I can't do my work some computer has failed that I never heard of.

Un système réparti est un système qui vous empêche de travailler quand une machine dont vous n'avez jamais entendu parler tombe en panne.

3. Système formé de composants matériels ou logiciels localisés sur des ordinateurs en réseau qui communiquent et coordonnent leurs actions uniquement par l'envoi de messages, le tout en vue de répondre à un besoin donné.

4. Un système distribué est un ensemble d'entités autonomes de calcul (ordinateurs, PDA, processeurs, processus, processus léger etc.) interconnectées et qui peuvent communiquer (donnant l'exemple d'un réseau physique de machines, un logiciel avec plusieurs processus sur une même machine).

Les fonctions des systèmes distribués (réseaux) peut se résumer dans l'aspect économiques (rapport prix/performance), l'adaptation de la structure d'un système à celle des applications (géographique ou fonctionnelle), le besoin d'intégration (applications existantes) de communication et de partage d'information, la réalisation de systèmes à haute disponibilité, le partage et meilleure gestion des ressources (programmes, données, services), la résistance aux pannes, mode dégradé et la réalisation de systèmes à grande capacité d'évolution.

Une solution adéquate vu qu'il pouvait répondre aux nouveaux besoins applicatifs tels que : l'hétérogénéité, l'autonomie, l'ouverture, l'évolutivité, la flexibilité..., c'est les systèmes multi agents, il représente un nouveau moyen de modélisation qui découle de l'intelligence artificielle distribuée et de la modélisation objet.

Les systèmes multi agent sont un domaine relativement jeune, apparu dans les années 80. On peut dire que les SMA ont vu le jour avec l'avènement de l'intelligence artificielle distribuée (IAD). A ses débuts toutefois l'IAD s'intéresse à des comportements intelligents qui résultent de l'activité coopérative de plusieurs agents. Suite à la distribution de l'expertise sur un ensemble de composants qui communiquent pour atteindre un objectif global ou résoudre un problème, il est nécessaire de diviser le problème en sous-problèmes. Cette division n'est pas toujours aisée car beaucoup de problèmes ne peuvent être divisés. Ainsi une extension des systèmes d'IAD est proposée : les composants doivent être capables de raisonner sur les connaissances et les capacités des autres dans le but d'une coopération effective. Pour ce faire, ils doivent être dotés de capacités de perception et d'action sur l'environnement et doivent posséder une certaine autonomie de comportement, on parle alors d'agents et par conséquent de système multi-agents.

Les systèmes multi agents sont des systèmes idéaux dans la résolution collective des problèmes, ils permettent de résoudre des problèmes de taille et de complexité telle qu'il n'est pas réaliste d'essayer de les résoudre avec un seul agent.

Un SMA est composé d'un ensemble d'agents qui simulent les capacités du raisonnement humain, ces agents ont un comportement qui fait appel aux notions suivantes : environnement, interaction, contrôle et l'organisation.

L'environnement celui qui contient l'ensemble des entités avec les quelles interagissent les agents.

L'interaction c'est toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent. Elles permettent aux agents de participer à la satisfaction d'un but global. Les systèmes multi-agents s'intéressent aux comportements collectifs produits par ces interactions selon plusieurs schémas d'interactions dont : la communication qui est le moyen que possèdent les agents pour se mettre en relation, la coopération, la planification et la négociation.

Le contrôle qui dirige et guide le processus de résolution et de construction de la solution globale.

Et enfin l'organisation qui permet de organiser logiquement les agents.

Chapitre 2

PROBLEMATIQUE ET EQUATIONS DE BASE

2. Problématique et équations de base

2.1 Introduction

On présente dans ce chapitre les problèmes que notre sujet s'attache à résoudre qui sont divisées en deux parties, la première partie est classique par les méthodes numériques et la seconde concerne l'analyse par une méthode informatique.

Le modèle hydrodynamique se base sur le modèle de Li [28]. Le modèle décrit l'analyse du mouvement dynamique secondaire du piston avec incorporation des forces hydrodynamiques générées dans le film d'huile de lubrification. Cette lubrification concerne l'ensemble jupe-cylindre. L'effet des segments, n'est pas traité dans cette étude.

Le mouvement du piston est défini dans cette étude par les excentricités du haut et du bas (e_h, e_b) de la jupe. Le mouvement est décomposé en trois mouvements indépendants simples : le mouvement axial, le mouvement de translation latéral et le mouvement de rotation. Le comportement hydrodynamique global est alors obtenu par la résolution puis la superposition des mouvements hydrodynamiques simples.

2.2 Problématique

2.2.1 Problématique mécanique

Soit à considérer le mouvement du piston dans le cylindre comme le montre la figure 2.1.a. Le mouvement secondaire est défini par les excentricités du haut e_h et du bas e_b de la jupe. La position du piston, la vitesse de glissement (ou vitesse axiale) et l'accélération le long du cylindre, sont en fonction de l'angle de rotation du vilebrequin $\bar{\theta}$. L'axe y est introduit pour le calcul hydrodynamique du piston. Il représente la coordonnée axiale du film lubrifiant. Puisque le film lubrifiant suit à chaque instant le piston, l'axe y est fixé avec le piston et se déplace en même temps que le piston. Les angles θ_1 et θ_2 représentent les angles fluide de pression de la jupe sont égaux et ont pour valeur empirique 15° [1].

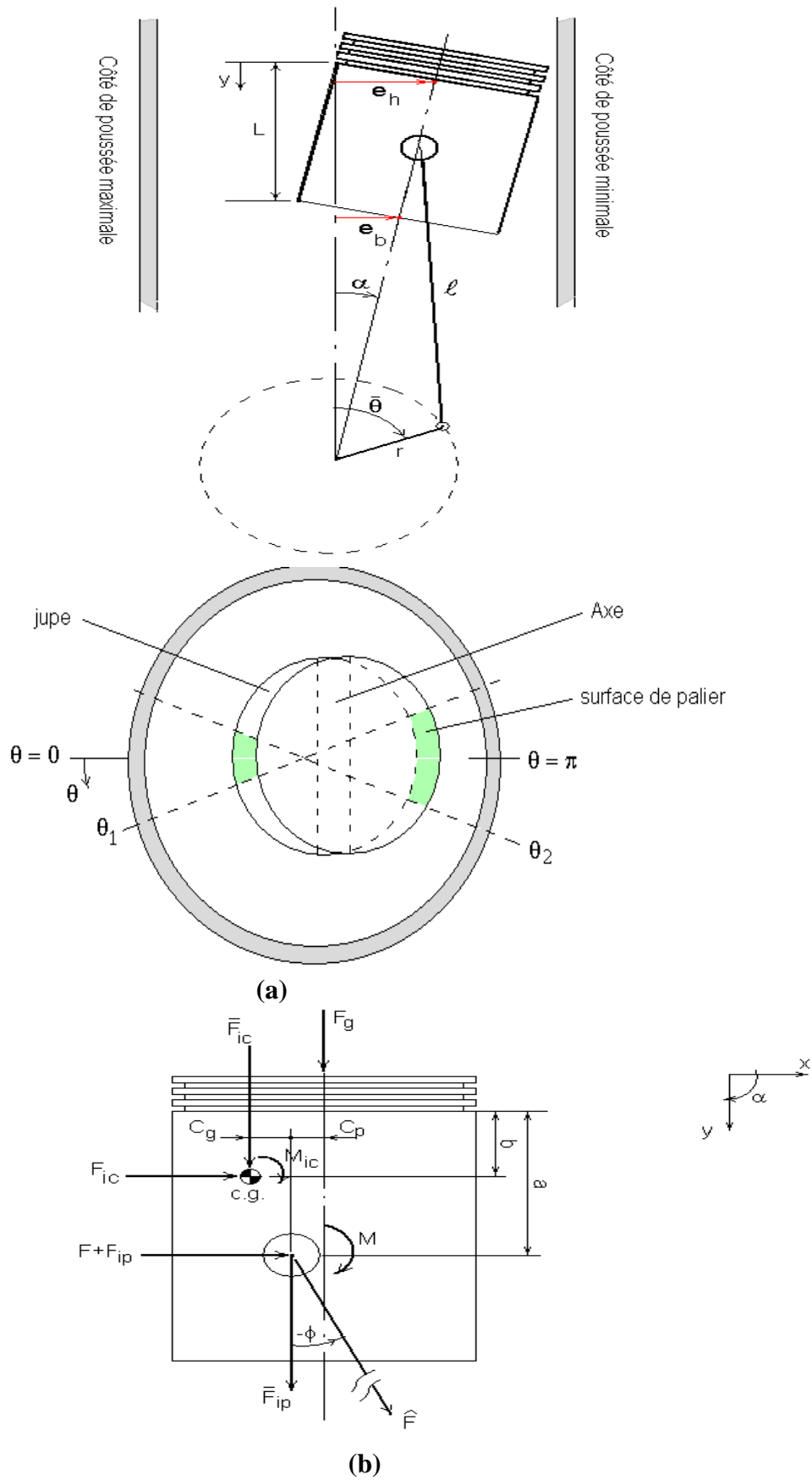


Figure 2.1 : Représentation schématique du mouvement du piston (a) et des charges appliquées (b)

L'analyse de l'équilibre des forces et des moments appliqués au piston (figure 2.1.b) permet d'obtenir le système d'équations du mouvement dynamique secondaire du piston.[10]

$$\begin{bmatrix} m_{\text{pist}} \left(1 - \frac{b}{L}\right) + m_{\text{axe}} \left(1 - \frac{a}{L}\right) & m_{\text{pist}} \frac{b}{L} + m_{\text{axe}} \frac{a}{L} \\ \frac{I_{\text{pist}}}{L} + m_{\text{pist}} (a - b) \left(1 - \frac{b}{L}\right) & -\frac{I_{\text{pist}}}{L} + m_{\text{pist}} (a - b) \frac{b}{L} \end{bmatrix} \begin{Bmatrix} \ddot{e}_h \\ \ddot{e}_b \end{Bmatrix} = \begin{Bmatrix} F + F_S \\ M + M_S \end{Bmatrix} \quad (2.1)$$

où F_S est la force latérale de paroi, définie comme

$$F_S = (F_G + \bar{F}_{IP} + \bar{F}_{IC}) \text{tg}\phi \quad (2.2)$$

Et M_S est le moment de basculement du piston selon l'angle α est défini par :

$$M_S = F_g C_p - \bar{F}_{IC} C_g \quad (2.3)$$

Pour résoudre l'équation (2.1) il faudrait connaître les charges hydrodynamiques F et M . ces charges sont dues à la pression hydrodynamique développée dans le film d'huile dans les zones de pression limitées par les arcs $-\theta_1, \theta_1$ et $\theta_2, 2\pi - \theta_2$ (figure 2.1.a). La pression hydrodynamique est régie par l'équation adimensionnelle de Reynolds, (hypothèse d'un film mince incompressible avec écoulement laminaire) :

$$\begin{aligned} \frac{\partial}{\partial \theta} \left(\bar{h}^3 \frac{\partial \bar{p}}{\partial \theta} \right) + \frac{\partial}{\partial \bar{y}} \left(\bar{h}^3 \frac{\partial \bar{p}}{\partial \bar{y}} \right) = -\bar{U} \left(\frac{(\varepsilon_b - \varepsilon_h) \cos \theta}{L} + \frac{\partial \bar{f}}{\partial \bar{y}} \right) \\ + \beta \left(\dot{\varepsilon}_h \cos \theta + (\dot{\varepsilon}_b - \dot{\varepsilon}_h) \frac{\bar{y}}{L} \cos \theta \right) \end{aligned} \quad (2.4)$$

Et les conditions limites de pression dans les zones inactives (pas de pression dans ces zones) est donnée par :

$$\bar{p}(\bar{y}, \theta^*) = 0 \quad \theta_1 \leq \theta^* \leq \theta_2 \text{ (zone inactive)} \quad (2.5)$$

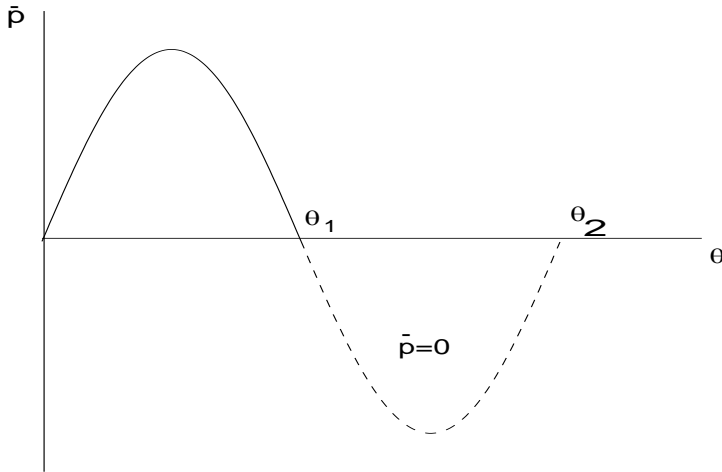


Figure 2.2 : Condition de pression dans la zone inactive

Aux points $\theta = 0$ et $\theta = \pi$, la pression est maximale et s'exprime par un gradient de pression nul

$$\left. \frac{\partial \bar{p}}{\partial \theta} \right|_{\theta=0} = \left. \frac{\partial \bar{p}}{\partial \theta} \right|_{\theta=\pi} = 0 \quad (2.6)$$

La condition de symétrie de pression dans le film est donnée par

$$\bar{p}(\bar{y}, -\theta) = \bar{p}(\bar{y}, \theta) \quad (2.7)$$

La pression est nulle sur le haut et le bas de la jupe

$$\bar{p}(0, \theta) = \bar{p}(\bar{L}, \theta) = 0, \quad (2.8)$$

Puisque e_h et e_b sont très petits devant L , l'épaisseur dimensionnée du film d'huile est approchée par :

$$h = c + e_h(t) \cos \theta + \frac{y}{L} [e_b(y) - e_h(y)] \cos \theta + f(y, \theta) \quad (2.9)$$

où c est le jeu radial nominal entre le piston et le cylindre, $f(y, \theta)$ est le profil de la surface de la jupe du piston. Ce profil est mesuré à partir de la surface d'un cylindre de base de rayon R représentant le piston. Le profil de la jupe $f(y, \theta)$ est construit dans le but d'obtenir une lubrification performante de la jupe durant le mouvement alternatif du piston. Ce profil tient également compte de la rugosité de la surface de contact.

Sous forme adimensionnelle l'équation 2.9 s'écrit :

$$\bar{h} = 1 + \varepsilon_h(\bar{t}) \cos \theta + \frac{\bar{y}}{L} [\varepsilon_b(\bar{t}) - \varepsilon_h(\bar{t})] \cos \theta + \bar{f}(\bar{y}) \quad (2.10)$$

La pression totale \bar{p} est obtenue peut être considérée comme la somme de trois termes de pression

$$\bar{p}(\bar{t}) = -\bar{U}(\bar{t})\bar{p}_u + \dot{\varepsilon}_h(\bar{t})\bar{p}_{1c} + [\dot{\varepsilon}_h(\bar{t}) - \dot{\varepsilon}_b(\bar{t})]\bar{p}_{2c} \quad (2.11)$$

Où chacun des termes représente la pression d'un mouvement simple :

- \bar{p}_u : Pression adimensionnée dans le film lors du déplacement du piston avec une vitesse axiale adimensionnée $\bar{U}(\bar{t})$

- \bar{p}_{1c} et \bar{p}_{2c} Pression adimensionnée dans le film lors du mouvement secondaire du piston avec des vitesses latérales $\dot{\varepsilon}_h$ et $\dot{\varepsilon}_b$

Ces pressions sont solutions respectives des équations de Reynolds pour chacun des mouvements :

-Mouvement axial ou mouvement de glissement du piston:

\bar{p}_u est solution de l'équation de Reynolds:

$$\frac{\partial}{\partial \theta} \left(\bar{h}^3 \frac{\partial \bar{p}_u}{\partial \theta} \right) + \frac{\partial}{\partial \bar{y}} \left(\bar{h}^3 \frac{\partial \bar{p}_u}{\partial \bar{y}} \right) = \frac{\varepsilon_b - \varepsilon_h}{L} \cos \theta + \frac{\partial \bar{f}}{\partial \bar{y}} \quad (2.12)$$

-Mouvement de translation ou d'écrasement

\bar{p}_{1c} est solution de l'équation de Reynolds pour le mouvement secondaire de translation latérale

$$\frac{\partial}{\partial \theta} \left(\bar{h}^3 \frac{\partial \bar{p}_{1c}}{\partial \theta} \right) + \frac{\partial}{\partial \bar{y}} \left(\bar{h}^3 \frac{\partial \bar{p}_{1c}}{\partial \bar{y}} \right) = \beta \cos \theta \quad (2.13)$$

-Mouvement de rotation

\bar{p}_{2c} est solution de l'équation de Reynolds pour le mouvement secondaire de rotation

$$\frac{\partial}{\partial \theta} \left(\bar{h}^3 \frac{\partial \bar{p}_{2c}}{\partial \theta} \right) + \frac{\partial}{\partial \bar{y}} \left(\bar{h}^3 \frac{\partial \bar{p}_{2c}}{\partial \bar{y}} \right) = \beta \frac{y}{L} \cos \theta \quad (2.14)$$

2.2.2 Problématique informatique

Génomique, météorologie, dynamique des fluides, simulations d'accidents, ou conception de médicaments : ces domaines nécessitent des temps et des puissances de calcul extrêmement importants. Pour cela dans tous les domaines l'homme est amené à faire mieux, plus, et plus rapidement notamment dans le domaine de l'informatique. Il se retrouve face à des défis colossaux, nécessitant énormément d'unités de calcul et traitements de très gros volumes de données.

Les laboratoires de recherche, et certaines grandes sociétés avec des centres de Recherches et Développement, ont un besoin toujours grandissant de force de calcul. Des applications "très gourmandes" en terme de puissance nécessiteraient plusieurs dizaines d'années pour pouvoir être menée à bien sur une machine standard, ou alors une autre solution serait d'investir dans la construction d'un supercalculateur à coup de Millions d'euros, ce qui n'est pas à la portée de tout le monde.

Pour certaines de ses composantes, l'achat de matériel dédié à ces calculs n'est pas possible, car un système de cluster est très coûteux, que ce soit en terme de coût du matériel informatique que des infrastructures qui vont l'accueillir (salle serveur, climatisation, installation électrique et de sécurité...). De plus, tous ces éléments prennent une place qui n'est pas négligeable dans un bâtiment.

Le réel problème qui se pose en plusieurs questions intéressantes. En arrivant à un constat très simple: toutes seconde de CPU inutilisée est perdue à jamais ! Alors, pourquoi ne pas les utiliser à bon escient? Pourquoi ne pourrait-on pas créer un système dont chaque unité de calcul serait indépendante, sans contrainte pour les utilisateurs des postes? De ce fait, l'idée de distribuer un calcul imposant entre plusieurs machines indépendantes commence à émerger. Un ensemble de machines hétérogènes, toutes regroupées pour une même cause, représente une puissance de calcul exceptionnelle.

2.3 Equations de base du mouvement

2.3.1 Le mouvement hydrodynamique axial

Le mouvement axial est obtenu par le déplacement alternatif du piston entre le PMH et le PMB. Ce mouvement est défini par l'équation :

$$\varepsilon_h^a = \varepsilon_b^a = k \tag{2.15}$$

où k est nul si le mouvement axial est un mouvement axial pur, k est non nul si le mouvement est considéré comme global.

ε_h^a et ε_b^a sont les excentricités du haut et du bas de la jupe du mouvement axial.

La zone de pression active dépend du sens de déplacement du piston

- a) si $\bar{U}(\bar{t}) > 0$ la zone active 2 (figure 2.3) est définie par les deux domaines fluides de pression 3D suivants :
- le domaine ayant pour section transversale l'angle fluide de pression $(-\theta_1, \theta_1)$, dans la direction circonférentielle θ , et une hauteur égale à $\frac{L}{2}$ délimitée par l'intervalle $(\frac{L}{2}, L)$ dans la direction y .
 - le domaine ayant pour section transversale l'angle fluide de pression $(\theta_2, 2\pi - \theta_2)$ avec une hauteur identique au domaine fluide ci-dessus.
- b) si $\bar{U}(\bar{t}) < 0$, la zone active est la zone 1 définie comme la zone 2 par les deux domaines fluides 3D avec comme hauteur délimité dans l'intervalle $(0, \frac{L}{2})$

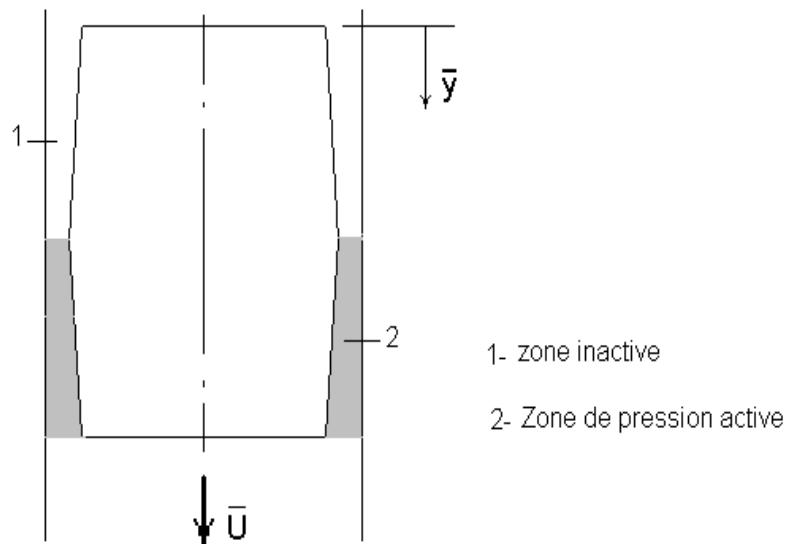


Figure 2.3 : Zone de pression active avec vitesse positive

La zone de pression active existe ainsi quelque soit le sens de déplacement du piston dans le cylindre. Cela est possible grâce au profil que l'on a adopté et qui permet la formation du coin d'huile entre les surfaces en mouvement. La géométrie la plus simple d'un tel profil est montrée sur la figure 2.4.

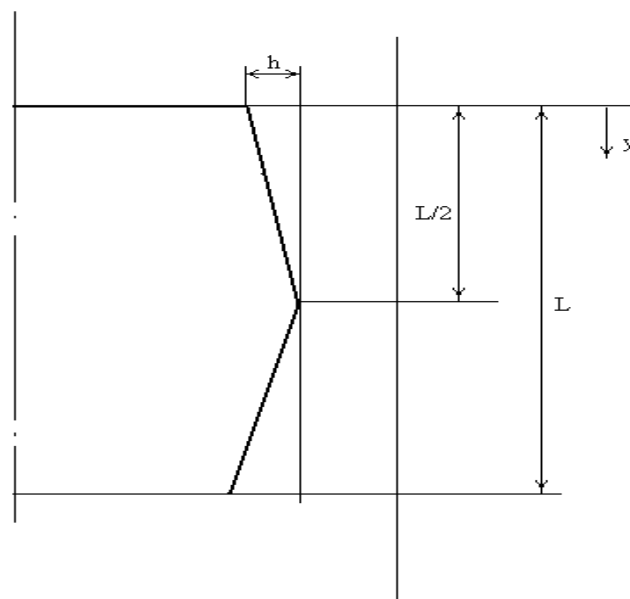


Figure 2.4 : Profil adopté de la jupe

Le profil adopté est linéaire selon \bar{y} et axisymétrique selon la direction circonférentielle. Il est donné par les équations suivantes

$$f(y) = \begin{cases} h\left(1 - \frac{2y}{L}\right) & 0 \leq y \leq \frac{L}{2} \\ h\left(\frac{2y}{L} - 1\right) & \frac{L}{2} \leq y \leq L \end{cases} \quad (2.16)$$

Pour être introduites dans l'équation (2.10), les équations du profil doivent être écrites sous forme adimensionnelle. Les équations (2.16) deviennent

$$\bar{f}(\bar{y}) = \begin{cases} \bar{h}\left(1 - \frac{2\bar{y}}{\bar{L}}\right) & 0 \leq \bar{y} \leq \frac{\bar{L}}{2} \\ \bar{h}\left(\frac{2\bar{y}}{\bar{L}} - 1\right) & \frac{\bar{L}}{2} \leq \bar{y} \leq \bar{L} \end{cases} \quad (2.17)$$

2.3.2 Le mouvement hydrodynamique de translation

Le mouvement hydrodynamique latéral de translation ou mouvement d'écrasement est obtenu par le déplacement latéral du piston dans le confinement du jeu radial (figure 2.5). Ce mouvement s'exprime par les équations suivantes :

$$\varepsilon_h^{\text{Lat}} = \varepsilon_b^{\text{Lat}} = d \quad (2.18)$$

où d représente la valeur algébrique du déplacement latéral du piston par rapport à l'axe du cylindre, $\varepsilon_h^{\text{Lat}}$ et $\varepsilon_b^{\text{Lat}}$ sont les déplacements du haut et du bas de la jupe lors du mouvement latéral du piston. La position de la zone de pression active dépend du sens de déplacement du piston

- a) si $d > 0$ la zone active est définie par le domaine fluide 3D ayant pour section l'angle fluide $(-\theta_1, \theta_1)$ et pour hauteur la longueur L de la jupe
- b) si $d < 0$ (cas de la figure 2.5) la zone active définie par le domaine 3D ayant pour section l'angle fluide de pression $(\theta_2, 2\pi - \theta_2)$ et pour hauteur la longueur L de la jupe

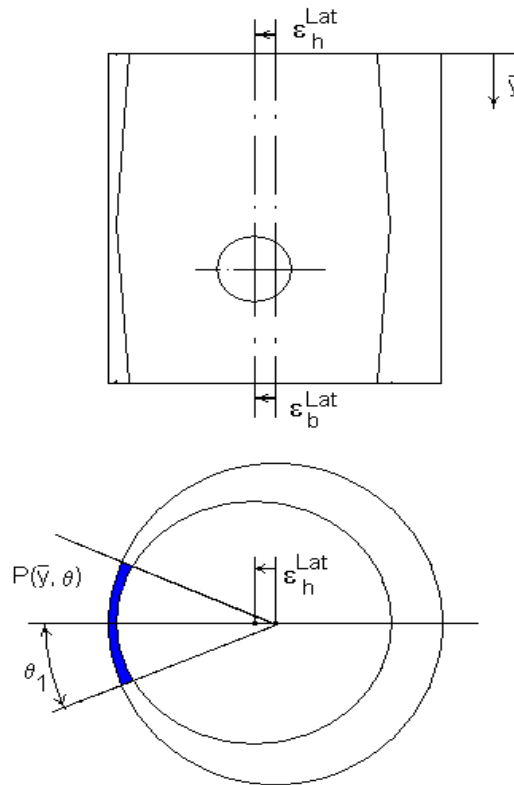


Figure 2.5 : Le mouvement de translation latéral du piston

2.3.3 Le mouvement hydrodynamique de rotation

Ce mouvement est défini par le basculement du piston autour de son axe (figure 2.6). Ce mouvement génère des zones de pression du côté de poussée maximale en $\theta=0$ et du côté de poussée minimale en $\theta=\pi$.

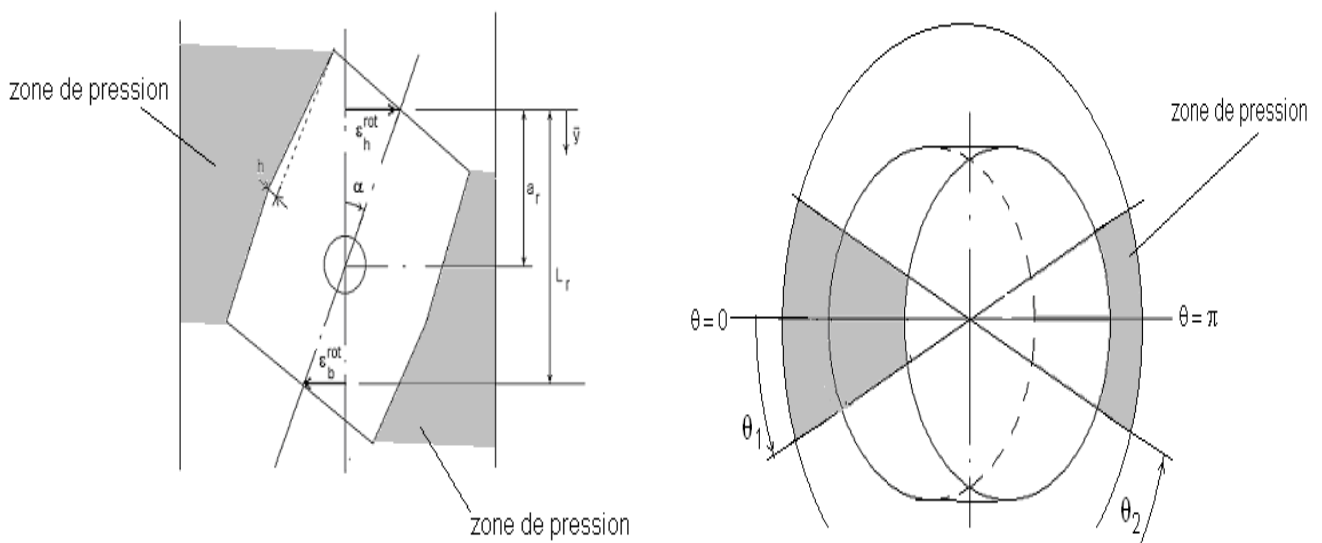


Figure 2.6 : Le mouvement hydrodynamique de rotation pure

Ce mouvement peut être formulé par les équations suivantes

$$e_h^{\text{rot}} = \alpha a \quad (2.19)$$

$$e_b^{\text{rot}} = -(L-a)\alpha \quad (2.20)$$

où α est l'angle de rotation ou de basculement du piston autour de son axe

Les relations liant les excentricités dimensionnelles et adimensionnelles sont données par :

$$e_h^{\text{rot}} = \varepsilon_h^{\text{rot}} c$$

$$e_b^{\text{rot}} = \varepsilon_b^{\text{rot}} c$$

Introduisant les valeurs des excentricités dans les équations (2.19) et (2.20), les excentricités adimensionnelles du haut et du bas de la jupe s'écrivent :

$$\varepsilon_h^{\text{rot}} = \alpha \frac{a}{c} \quad (2.21)$$

$$\varepsilon_b^{\text{rot}} = -\frac{(L-a)\alpha}{c} = -\left(\frac{L}{a} - 1\right) \varepsilon_h^{\text{rot}} \quad (2.22)$$

Les zones de pression active sont montrées en gras dans la figure 2.8.

2.3.4 Le mouvement hydrodynamique global

Le mouvement hydrodynamique global, défini par le couple d'excentricités e_h et e_b , est obtenu par la superposition des mouvements introduits ci-dessus

$$e_h = e_h^a + e_h^{\text{Lat}} + e_h^{\text{rot}} \quad (2.23)$$

$$e_b = e_b^a + e_b^{Lat} + e_b^{rot} \tag{2.24}$$

Le mécanisme de superposition est donné par la figure 2.7.

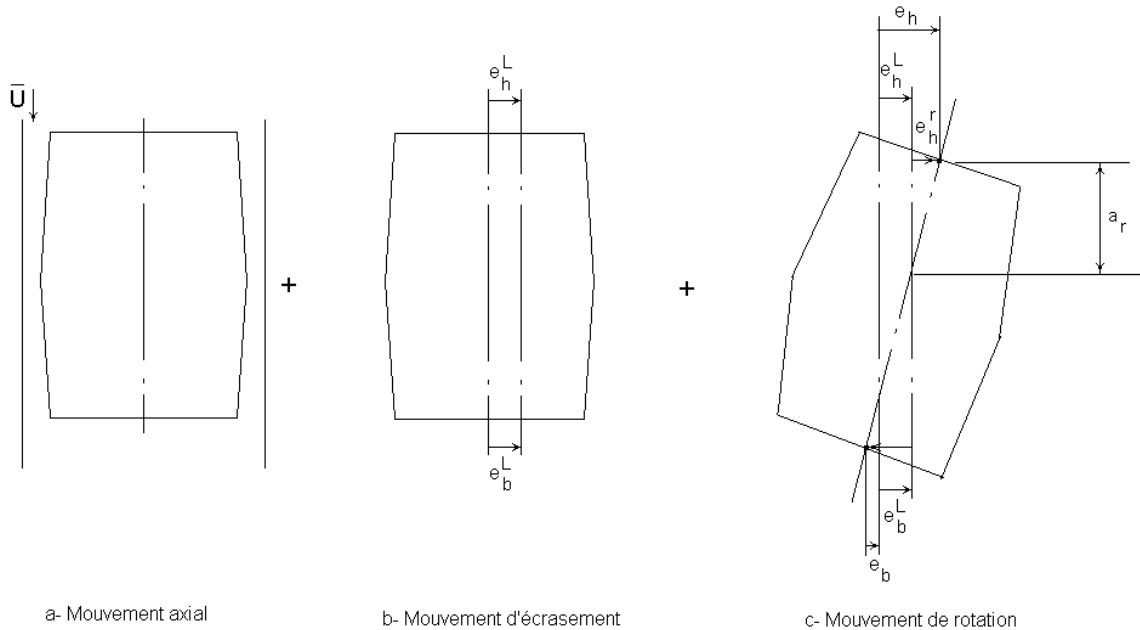


Figure 2.7 : Le mouvement hydrodynamique global

Introduisant les relations 2.21 et 2.22 dans les relations ci-dessus le mouvement composé a pour équations :

$$\varepsilon_h = \varepsilon_h^a + \varepsilon_h^{Lat} + \varepsilon_h^{rot} \tag{2.25}$$

$$\varepsilon_b = \varepsilon_b^a + \varepsilon_b^{Lat} - \left(\frac{L}{a} - 1 \right) \varepsilon_h^{rot} \tag{2.26}$$

Les relations inverses des équations 2.25 et 2.26 permettent d’obtenir les mouvements composés en fonction du mouvement global

a) Mouvement axial

Les équations restent inchangées et sont donnés par l’équation (2.15)

b) Mouvement latéral de translation

$$\varepsilon_h^{Lat} = \varepsilon_b^{Lat} = \varepsilon_h \left(1 - \frac{a}{L} \right) + \varepsilon_b \frac{a}{L} \tag{2.27}$$

où $cte = \frac{a}{L}$

c) Mouvement de rotation

$$\varepsilon_h^{\text{rot}} = \frac{a}{L}(\varepsilon_h - \varepsilon_b) \quad (2.28)$$

$$\varepsilon_b^{\text{rot}} = \left(\frac{a}{L} - 1\right)(\varepsilon_h - \varepsilon_b) \quad (2.29)$$

2.3.5 Equations des charges hydrodynamiques

2.3.5.1 Charges hydrodynamiques des mouvements composés

Les charges sont calculées instantanément séparément pour chaque modèle hydrodynamique. La charge totale est obtenue par la superposition des différents cas de chargement.

Les différentes pressions \bar{p}_u , \bar{p}_{1c} et \bar{p}_{2c} sont déterminées par la résolution des équations de Reynolds 2.12-2.14. Elles sont ensuite intégrées pour trouver les forces adimensionnées \bar{F}_u , \bar{F}_{1c} et \bar{F}_{2c} et les moments \bar{M}_u , \bar{M}_{1c} et \bar{M}_{2c} calculées par rapport à l'axe du piston . Pour un mouvement composé les charges hydrodynamiques ont pour expression

$$\bar{F}_\ell = 2 \int_0^{\bar{L}} \int_0^\pi \bar{p}_\ell \cos\theta d\theta d\bar{y} \quad (2.30)$$

$$\bar{M}_\ell = 2 \int_0^{\bar{L}} \int_0^\pi \bar{p}_\ell (\bar{a} - \bar{y}) \cos\theta d\theta d\bar{y} \quad (2.31)$$

où, l'indice ℓ prend les valeurs 1, 2 et 3 qui correspondent respectivement aux mouvement axial, latéral de translation et de rotation respectivement

- **Charges hydrodynamiques du mouvement axial**

Les forces hydrodynamiques dépendent de la position de la zone active de pression. Cette dernière dépend du sens de déplacement axial du piston ou encore de la vitesse. Dans le cas de la vitesse axiale positive (figure. 2.8), le piston se déplace du PMH vers le PMB, on a les forces hydrodynamiques :

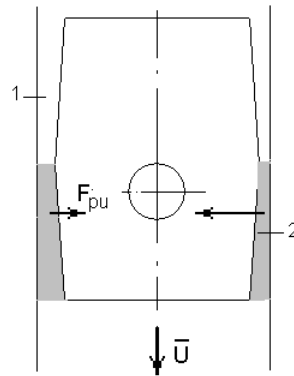


Figure 2.8 : Forces hydrodynamiques du mouvement axial

c) zone inactive 1 : $F_{pu}=0$

d) zone active 2 : les forces et moments sont données par les équations 2.30 et 2.31 (avec $\bar{p}_\ell = \bar{p}_u$ et $\bar{F}_\ell = \bar{F}_u$) dans chacun des deux domaines ayant pour section les angles fluides de pression $(-\theta_1, \theta_1)$ et $(\theta_2, 2\pi-\theta_2)$

Si la vitesse axiale est négative, les forces hydrodynamiques sont situées dans la zone 1.

• Charges hydrodynamiques du mouvement de translation latéral

a) cas $d < 0$

Dans ce cas les excentricités sont négatives (figure 2.9). La force hydrodynamique est nulle dans la zone 2. Dans la zone de pression 1, les charges hydrodynamiques F_{Lat} et M_{Lat} sont données par les relations 2.30 et 2.31 (avec $\bar{p}_\ell = \bar{p}_{lc}$ et $\bar{F}_\ell = \bar{F}_{lat}$).

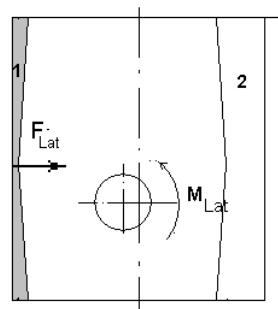


Figure 2.9 : Forces hydrodynamiques du mouvement latéral

b) $d > 0$

Les excentricités sont positives, les charges de pression sont situées dans la zone active 2 et sont données par les équations 2.30 et 2.31.

- Charges hydrodynamiques du mouvement de rotation

Dans ce cas les forces hydrodynamiques existent dans les deux zones de pression 1 et 2 définies, comme précédemment, leurs valeurs sont données par les expressions 2.30 et 2.31 avec $\bar{p}_\ell = \bar{p}_{2c}$ et $\bar{F}_\ell = \bar{F}_{rot}$

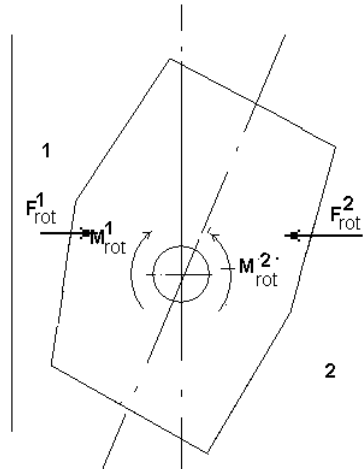


Figure 2.10: Forces hydrodynamiques du mouvement de rotation

- Charges hydrodynamiques globales

Les charges hydrodynamiques de chacun des mouvements étant déterminées, les charges hydrodynamiques globales F et M sont données par :

a) en fonction des vitesses latérales adimensionnées

$$F = -F_u \bar{V} + \dot{\epsilon}_h(\bar{t}) F_{1C} + [\dot{\epsilon}_b(\bar{t}) - \dot{\epsilon}_h(\bar{t})] F_{2C} \tag{2.32}$$

$$M = -M_u \bar{V} + \dot{\epsilon}_h(\bar{t}) M_{1C} + [\dot{\epsilon}_b(\bar{t}) - \dot{\epsilon}_h(\bar{t})] M_{2C} \tag{2.33}$$

ou en fonction des vitesses latérales dimensionnées

$$F = -F_u \bar{V} + \frac{\dot{\epsilon}_h(\bar{t})}{c} F_{1C} + \frac{[\dot{\epsilon}_b(\bar{t}) - \dot{\epsilon}_h(\bar{t})]}{c} F_{2C} \tag{2.34}$$

$$M = -M_u \bar{V} + \frac{\dot{\epsilon}_h(\bar{t})}{c} M_{1C} + \frac{[\dot{\epsilon}_b(\bar{t}) - \dot{\epsilon}_h(\bar{t})]}{c} M_{2C} \tag{2.35}$$

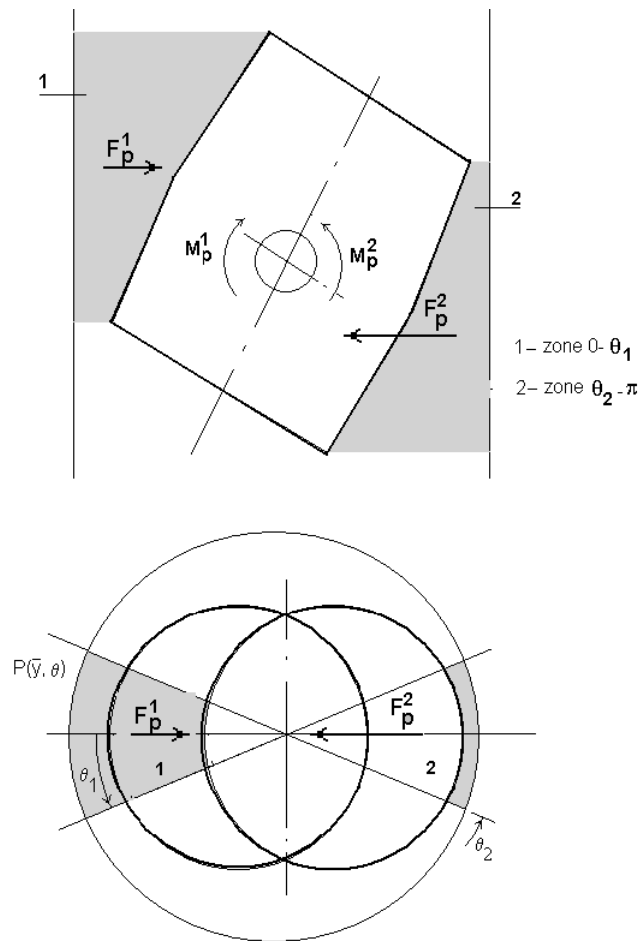


Figure 2.11: Forces hydrodynamiques globales

Chapitre 3

LES SYSTEMES MULTI AGENTS

3.1 Système Multi-agents

Il existe plusieurs définitions de système multi agent :

- un système multi agents est une communauté d'agents autonomes travaillant en commun, selon des modes parfois complexes de coopération, conflit, concurrence, pour aboutir à un objectif global : la résolution d'un problème, l'établissement d'un diagnostic ou d'une autre manière un SMA est un système artificiel constitué d'une population d'agents autonomes interactifs qui coopèrent, avec un but commun, même si chaque agent peut ne poursuivre qu'un but individuel.
- Un système distribué composé d'un ensemble d'agents, qui simulent dans une certaine mesure les capacités du raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence.
- On appelle Système Multi-Agents, un système composé des éléments suivants (**figure 3.1**) : un ensemble d'agents, un ensemble de tâches à réaliser et un ensemble d'objets associés à l'environnement [33].

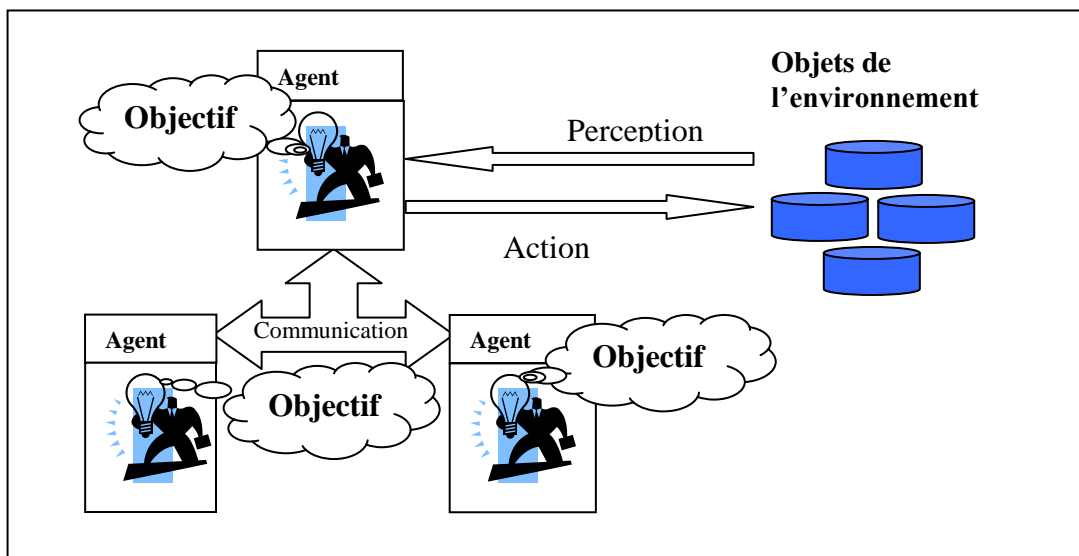


Figure 3.1 : système multi-agents

Un SMA a des principales caractéristiques qui se résument dans la décentralisation des données, l'asynchronisation des traitements, le contrôle réparti du système (c'est-à-dire aucun contrôle global du système), ainsi que chaque agent a des informations ou des capacités de résolution de problèmes limitées. Ainsi chaque agent a un point de vue partiel.

Un SMA peut résoudre un problème de manière distribuée, simuler un phénomène complexe, la plupart des exemples utilisent des agents réactifs, ou créer et maintenir un environnement de travail, nous allons examiner ces trois domaines, montrer la nature des agents nécessaires pour les mettre en place et la nature de leurs interactions.

Les principaux intérêts des systèmes multi-agents se résument dans l'utilisation de nombreux agents résolvant le même problème de façon différente produit généralement des solutions de meilleure qualité en terme de complétude et de précision, ils s'adaptent bien à la réalité dans la mesure où de nombreux problèmes sont de nature distribuée, ils permettent d'intégrer des connaissances diverses et complexes adoptant les formalismes hétérogènes, Ils permettent de résoudre des problèmes de taille et de complexité telle qu'il n'est pas réaliste d'essayer de les résoudre avec un seul agent ,et les connaissances insuffisantes peuvent être complétées suivant la résolution.

Il existe différentes domaines d'applications des systèmes multi-agents comme la simulation des phénomènes complexes (physiques, biologiques, économiques, sociologiques...), les réseaux d'experts (agent cognitif), l'Eco résolution (agent réactif), la modélisation de système physiquement distribué, l'agent d'interface, et les jeux vidéo.

Et d'après [33], Il existe cinq catégories d'application des systèmes multi-agents qui sont :

- La résolution distribuée de problèmes au sens large
- La robotique distribuée
- La simulation multi-agents
- La construction de mondes synthétiques
- La conception de programme à travers les concepts d'agents et d'interaction.

Et notre modèle étudié qui est l'analyse hydrodynamique d'un piston de Moteur à Combustion Interne par Simulation Systèmes Multi-agents (SMA).

3.2 L'agent

Le terme agent pourrait s'appliquer à n'importe quel programme informatique qui produit des résultats. Puisqu'il est difficile de donner une définition précise du terme agent, En ce qui suit, on présente les définitions les plus importantes :

Un agent est une entité qui perçoit son environnement et agit sur celui-ci.

Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs (buts) pour lesquels il a été conçu [34].

Les agents intelligents sont des entités logicielles qui réalisent des opérations à la place d'un utilisateur ou d'un autre programme, avec une sorte d'indépendance ou d'autonomie, et pour faire cela ils utilisent une sorte de connaissance ou de représentation des buts ou des désires de l'utilisateur, (L'agent IBM).

Un agent est une entité qui fonctionne continuellement et de manière autonome dans un environnement où d'autres processus se déroulent et d'autres agents existent [35].

Un agent peut être défini comme une entité active (physique; capteurs, processeurs...ou abstraite; tâches à réaliser, déplacement,...) composant le système, capable d'agir sur elle même et son environnement, disposant de capacité d'une représentation partielle de cet environnement, pouvant communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de sa connaissance et des interactions avec d'autres agents. [36].

Le terme d'entité active sert à signaler le fait qu'un agent ne correspond pas à la notion de module statique, mais à une notion d'objet concurrent et autonome ayant le propre contrôle de son activité.

Une autre définition est donnée par Ferber [33], on appelle agent une entité physique ou virtuelle :

1. qui est capable d'agir dans un environnement,
2. qui peut communiquer directement avec d'autres agents,
3. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
4. qui possède des ressources propres
5. qui est capable de percevoir (mais de manière limitée) son environnement,
6. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),

7. qui possède des compétences et offre de services,
 8. qui peut éventuellement se reproduire
 9. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et fonction de sa perception, de ses représentations et des communications qu'elle reçoit.
- On peut le définir aussi par le schéma suivant:

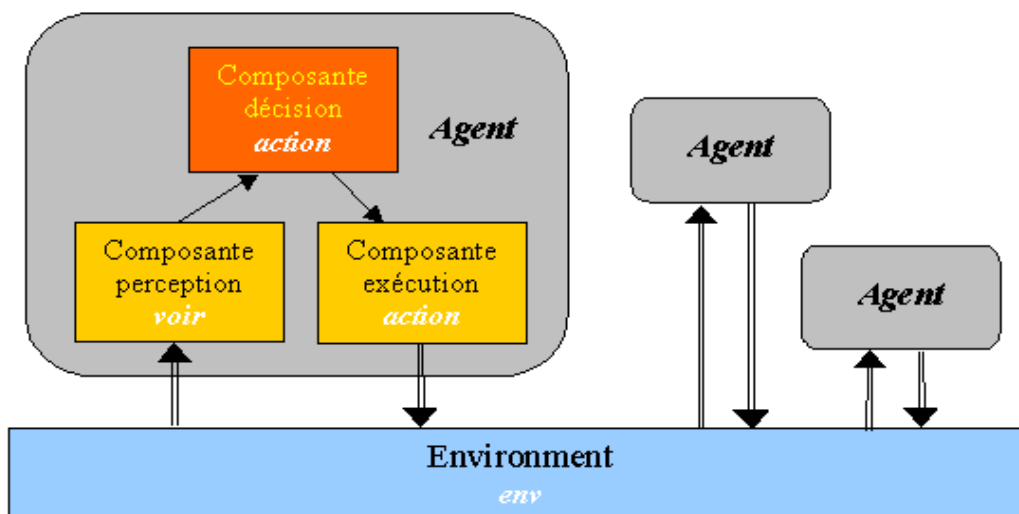


Figure 3. 2 : Structure générale d'un agent dans un système multi-agents

3.2.1 Caractéristique d'un agent

En partant de [34], et des définitions citées, on peut identifier les caractéristiques suivantes pour la notion d'agent:

Situé : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement.

Autonome : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.

Proactif : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment.

Capable de répondre à temps : l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis.

Social : l'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.

Intelligence : c'est la capacité de l'agent à accepter les demandes de l'utilisateur et de mener à bien la tâche que lui est déléguée. A un degré d'intelligence plus élevé, l'agent devrait comprendre ce que l'utilisateur veut, et planifier les moyens à mettre en œuvre pour atteindre ce but. A un degré supérieur l'agent non seulement planifie ses propres actions mais aussi tient compte de celle des autres.

Réactivité : l'agent doit pouvoir faire face aux modifications de l'environnement quelque que ce soit la modification des objets de l'utilisateur ou des ressources disponible.

3.2.2 Typologie des agents

La typologie d'un agent est une description de son organisation interne, les données et les connaissances de l'agent, les opérations qui peuvent être effectuées sur ces composantes et le flux de contrôle des opérations.

On peut citer essentiellement les typologies suivantes :

- **Agents réactifs**

Les agents réactifs sont souvent qualifiés de ne pas être "intelligents" par eux-mêmes, ils sont considérés intelligents au niveau du groupe, du système. Ils sont des composantes très simples qui réagissent uniquement à leurs perception et qui agissent en fonction de cette perception.. Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent ni croyances, ni mécanismes d'envoi de messages. Leurs capacités répondent uniquement au mode stimulus/action qui peut être considéré comme une forme de communication. Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents et présente un comportement global intelligent.

- **Agents cognitifs**

Les agents cognitifs possèdent une représentation partielle de l'environnement, des buts explicites, ils sont capables de planifier leur comportement, mémoriser leurs actions passées, communiquer par envoi de messages, négocier, etc. Un SMA constitué d'agents cognitifs possède communément peu d'agents.

L'agent est constitué d'une base de connaissances qui contient toutes les connaissances sur son fonctionnement (sa tâche) et sur son environnement. L'agent acquiert de nouvelles connaissances soit par l'échange d'informations avec les autres agents ou par le biais de ses perceptions sur son environnement. L'agent cognitif a la capacité de raisonner sur sa base de connaissances, ses intentions et ses buts pour prendre des décisions suivant une planification.

Ils font souvent appel à des modes de communication plus complexes qu'une simple perception. Ils communiquent généralement grâce à des structures de données partagées ou par des communications directes.

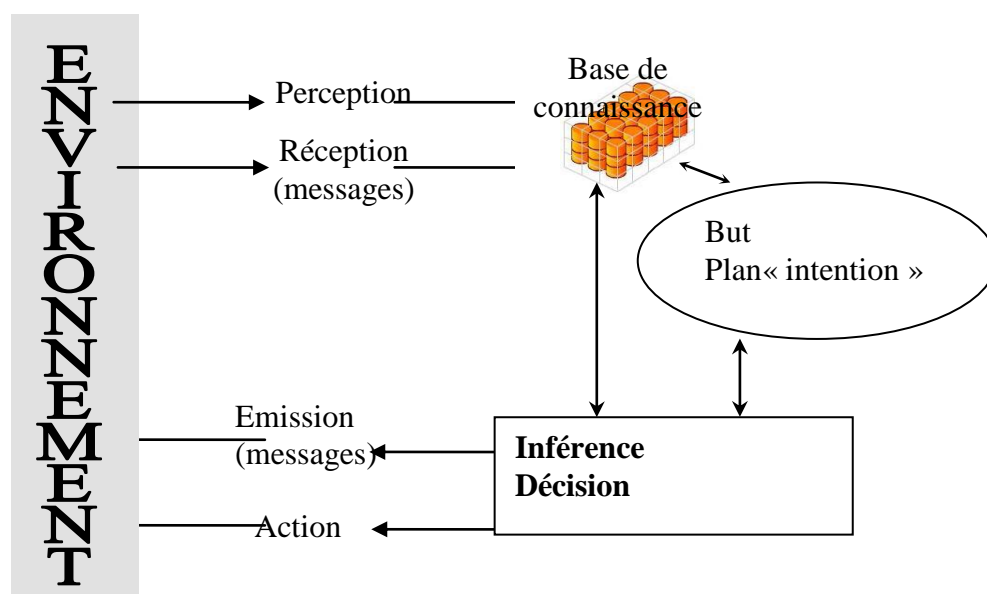


Figure 3.3 : Architecture d'un agent cognitif

- **Agents hybrides**

Une architecture hybride d'un agent intelligent est une architecture composée d'un ensemble de modules organisés dans une hiérarchie, chaque module étant soit une composante cognitive avec représentation symbolique des connaissances et capacités de raisonnement, soit une composante réactive. De cette manière, on combine le comportement pro-actif de l'agent, dirigé par les buts, avec un comportement réactif aux changements de l'environnement. En plus, on espère obtenir simultanément les avantages des architectures cognitives et réactives, tout en éliminant leurs limitations.

On peut présenter l'architecture hybride par le schéma suivant :

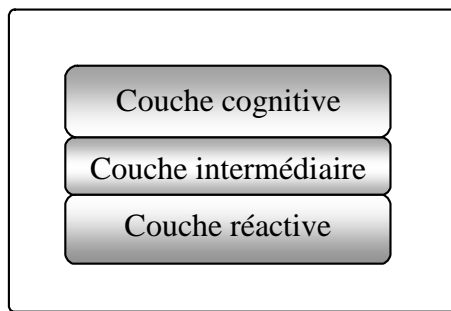


Figure 3.4 : Architecture hybride en couches verticales

3.2.3 Comportement d'un agent

Le comportement d'un agent s'intéresse aux notions suivantes :

- Environnement.
- interaction.
- contrôle.
- organisation.

3.2.3.1 Environnement

Du point de vue d'un agent, l'environnement correspond à «tout ce qui lui est extérieur » : les autres agents font aussi a priori partie de l'environnement. Cependant, l'environnement est souvent considéré comme une structure centralisée dynamique mais non autonome, qui contient l'ensemble des entités avec lesquelles interagissent les agents mais qui ne sont pas des agents.

L'évolution de l'environnement se modélise différemment selon les caractéristiques que l'on prend en compte, et les simplifications que l'on s'autorise. Les principales distinctions à faire sur les types d'environnements sont :

Un environnement est accessible à l'agent s'il perçoit entièrement l'état de l'environnement, Sinon, l'environnement est inaccessible.

Si l'état suivant de l'environnement est déterminé d'une manière unique par l'état courant et l'action de l'agent, alors l'environnement est déterministe. Sinon, l'environnement est non déterministe.

L'environnement est dynamique si son état peut se modifier sans l'action de l'agent dans l'intervalle de temps entre deux perceptions de l'agent. Sinon, on est dans le cas statique.

Si tout passage d'un état de l'environnement à un autre nécessite le passage par une séquence d'états intermédiaires, alors on a un environnement continu ; sinon, l'environnement est discret.

Les caractéristiques de l'environnement influencent la façon dont on conçoit un agent car il faut tenir compte de l'évolution de l'environnement, de la capacité de l'agent de saisir cette évolution et de sa capacité à décider en conséquence.

3.2.3.2 Interaction

Le concept d'interaction est l'un des aspects les plus importants des SMA, dans la mesure où il permet de relier les agents qui peuvent échanger des informations et s'influencer afin de constituer un système. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent. Elles permettent aux agents de participer à la satisfaction d'un but global. Cette participation permet au système d'évoluer vers un de ses objectifs et d'avoir un comportement intelligent indépendamment du degré de complexité des agents qui le composent.

En général, les interactions sont mises en œuvre par un transfert d'informations entre agents ou entre l'environnement et les agents, soit par perception, soit par communication. Par la perception, les agents ont connaissance d'un changement de comportement d'un tiers au travers du milieu. Par la communication, un agent fait un acte délibéré de transfert d'informations vers un ou plusieurs autres agents. L'interaction peut être décomposée en trois phases non nécessairement séquentielles :

- La réception d'informations ou la perception d'un changement.
- Le raisonnement sur les autres agents à partir des informations acquises.
- Une émission de message(s) ou plusieurs actions (plan d'actions) modifiant l'environnement. Cette phase est le résultat d'un raisonnement de l'agent sur son propre savoir-faire et celui des autres agents.

Les protocoles d'interaction

L'interaction entre les agents est plus qu'un simple échange de messages [37] : un des aspects d'interaction montre une conversation basée sur un échange partagé et conventionné de messages.

Les conversations entre agents dans les SMA sont souvent structurées selon des schémas typiques appelés protocoles d'interaction.

Les protocoles d'interaction permettent de décrire explicitement des enchaînements conversationnels lors des communications entre les agents. Ils représentent un schéma commun de conversation utilisé pour exécuter une tâche, une stratégie de haut niveau gouvernant les interactions entre les agents tout en permettant de faciliter leur dialogue. Un protocole précise qui peut dire quoi à qui et les réactions possibles à ce qui est dit.

Les protocoles d'interaction permettent de définir des séquences causales des messages communiqués entre les agents et décrivent comment les agents doivent réagir aux messages reçus durant les interactions. Les aspects de ces protocoles d'interaction diffèrent selon les types des agents (agents concurrents ou agents à des buts communs).

Il existe différents types de protocoles d'interaction, on cite :

- Les protocoles de coordination.
- Les protocoles de coopération.
- Les protocoles de négociation, ainsi que.
- Les mécanismes du commerce électronique.

Les protocoles d'interaction peuvent être modélisés en utilisant différents formalismes tels que :

- Les graphes état transition.
- Les graphes de raisonnement.
- Les réseaux de pétri.
- Les langages formels.
- Les langages de description de protocoles.

Les systèmes multi agents s'intéressent aux comportements collectifs produits par ces interactions selon plusieurs schémas d'interactions dont :

3.2.3.2.1 Communication

Les communications, dans les SMA comme chez les humains, sont à la base des interactions et de l'organisation. La communication est le concept de base qui permet de relier les agents d'un système pour qu'ils constituent un tout. Les agents communiquent dans le but de mieux accomplir leurs buts ou les buts

de la société dans laquelle ils existent. Elle permet aux agents de coordonner leurs actions et leurs comportements pour obtenir un système cohérent.

La communication, peut être définie, comme une action locale d'un agent vers les autres qui peut prendre deux modes : elle peut être indirecte ou directe.

La communication est l'ensemble des processus physique et psychologique par lesquels s'effectue l'opération de mise en relation d'un ou plusieurs agents (émetteur) avec un ou plusieurs agents (récepteur), en vue d'atteindre certains objectifs [33].

Pour que la communication entre les agents puisse avoir lieu, il faut définir une infrastructure de communication et un protocole de communication. L'infrastructure de communication assure la production et la réception des messages via un média physique (son, gestes, traces graphiques). Le protocole de communication assure la compréhension des messages. Il définit qui peut parler à qui, quand et comment. Il décrit les champs du message et leur domaine de valeurs, la syntaxe et la sémantique des énoncés.

On peut distinguer deux formes de communication : la communication synchrone et la communication asynchrone. Dans la communication synchrone, un message est reçu en même temps qu'il est émis (dialogue oral ou gestuel, téléphone), alors que dans la communication asynchrone, le message est émis hors de la présence de l'allocutaire.

Modes de communication

Les modes et protocoles de communication établissent les moyens et les styles de communication entre les agents. Il existe principalement deux modèles de communication : communication par envoi de messages (communication directe) et la communication par partage d'information (communication indirecte). Ces deux modèles ont été combinés pour donner naissance à un hybride [36].

1. Communication directe :

Ce type de communication utilise une technique d'envoi de message pour transmettre directement les informations. Les agents sont en liaison directe et envoient leurs messages directement et explicitement aux destinataires.

Dans les SMA, les communications sont supposées être fiables (aucun message n'est perdu ou modifié par le mécanisme d'envoi).

Cette technique de communication peut prendre plusieurs formes :

a. Communication par envoi de messages point à point :

Dans ce cas, l'agent émetteur (celui qui réalise l'action de communication) connaît nominativement l'agent destinataire du message et lui adresse alors directement l'information. Cet agent destinataire est le seul à recevoir l'information qui est alors privée.

b. Communication par diffusion généralisée

Il s'agit, dans ce cas, d'un envoi simultané d'un même message à tous les agents du système. L'agent émetteur ne connaît pas nécessairement nominativement les destinataires du message, ou du moins ne le désigne pas explicitement lors de l'envoi. L'agent suppose également que parmi tous les agents du système, il en existera bien un capable de traiter ce message.

c. Communication par diffusion restreinte

Il s'agit là aussi d'un envoi simultané d'un même message mais seulement à un sous-ensemble des agents du système. L'agent émetteur ne connaît pas nécessairement nominativement tous les agents destinataires, mais doit être capable de les décrire par les caractéristiques discriminantes, ou de les atteindre grâce à une notion de groupe présente dans le système.

Elle est généralement séparée des capacités d'interaction avec l'environnement. Il s'agit d'une conversation entre agents cognitifs, capable d'interpréter des actes de langage et de raisonner sur des messages ambigus. Un acte de langage est une action intentionnelle effectuée au cours d'une communication. Une conversation est un échange de messages, généralement structuré selon **un protocole de communication**.

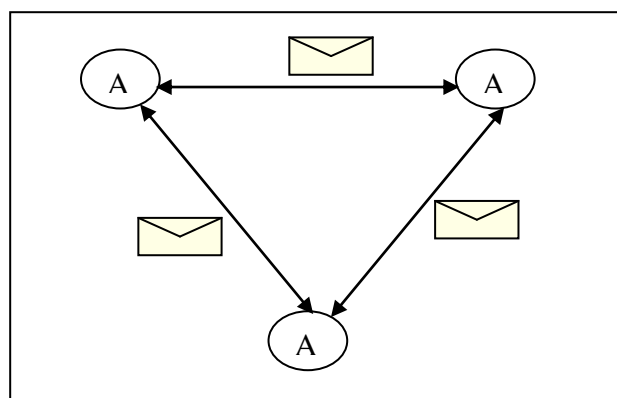


Figure 3.5 : communication par envoi de messages

2. Communication indirecte :

Les communications indirectes (ou implicites) supposent l'existence d'un support de communication commun aux agents (un environnement ou une structure de donnée partagée) qui mémorise et partage l'information associée aux signaux déclenchés par la réalisation de l'action. Ce type de communication distingue deux techniques différentes, selon le support de communication, qui sont la communication par tableau noir qui est une technique utilise une structure de donnée commune aux agents dont la quelle les agents déposent, consultent et modifient les informations, et la communication via l'environnement dans cette dernière technique, l'environnement est une structure commune aux agents. L'environnement possède des lois de comportements et contient un ensemble d'objets passifs qui peuvent être perçus, créer, détruit ou modifier par l'ensemble des agents. Les agents réagissent à leurs perceptions partielles de l'état de l'environnement et combinent leurs actions dans celui-ci, afin de modifier l'état de l'environnement et ainsi influencer la résolution collective.

L'étude comparative effectuée sur les deux modèles de communication, cités ci-dessus, a montré l'efficacité de l'approche basée sur l'échange de message.

Aussi lorsque le nombre des communications, autour du tableau noir, augmente ceci crée un encombrement gênant, limitant ainsi de l'efficacité du système .Dans le cas ou le tableau est distribué, l'approche revient alors à une approche par envoi de message entre tableaux noirs.

Cependant, l'inconvénient major de la communication par envoi de message est qu'elle nécessite l'élaboration d'un langage et des protocoles de communication complexes, ce qui rend cette approche plus délicate à mettre en œuvre.

Des travaux ont été conçus pour essayer de combiner la supervision de la communication à base de tableau noir avec la souplesse des approches à base d'envoi de message ; ils ont été menés à distribuer le tableau noir c-à-d : créer des agents figuratifs (agent données, agent faits, agents résultats..). Cette approche a été nommée : communication hybride.

Théorie des actes de langage

La théorie des actes de langage est une théorie de la communication fondée dans les années 1960 par le philosophe J.L. Austin [38]. Elle fut développée sous l'instigation de L. Searle [39]. Le postulat à la base de la théorie affirme que la production d'un énoncé s'assimile à l'accomplissement d'actions : si agir c'est transformer l'état des choses, parler c'est, également, transformer l'état mental des interlocuteurs. Une

conséquence de ce postulat est qu'il doit être possible d'identifier les constituants élémentaires de la communication. Ceux-ci sont appelés **actes de langage**.

J.L. Austin s'est intéressé aux différents types d'actes de langage que constituent les actions intentionnelles effectuées au cours d'une communication. Il distingue trois types d'actes de langage : actes locutoires, actes illocutoires et actes perlocutoires.

- **Les actes locutoires** se rapportent à la formulation d'un énoncé. Ces actes sont satisfaits lorsque l'énoncé est correctement formulé.
- **Les actes illocutoires** désignent l'action effectuée sur l'auditeur lors de la formulation de l'énoncé (par exemple donner un ordre, poser une question). Ils sont accomplis avec succès lorsque l'effet attendu sur auditeur est obtenu (un ordre exécuté, réponse à une question).
- **Les actes perlocutoires** rapportent aux conséquences indirectes visées par les actes locutoires et illocutoires. Ils ne sont pas codés dans l'énoncé et dépendent du contexte dans lequel s'effectue la communication. Ils sont satisfaits lorsqu'ils sont reconnus par l'auditeur (convaincre, faire croire, etc.).

En tant qu'objet d'étude, la théorie des actes de langage se présente en trois points:

- répertorier et proposer une taxonomie des différentes formes d'actes,
- définir un formalisme d'expression et de manipulation des actes (langage de communication),
- étudier les liens entre les actes et leurs sémantiques en terme de modification d'état mental.

Langage de communication

Pour que les agents puissent interagir de manière efficace, ils doivent disposer d'un langage commun pour pouvoir échanger des informations et des connaissances et coopérer pour la résolution d'un problème. Ce langage est un ensemble de primitives communs par chaque entité constitue un protocole de communication qui permet de structurer et d'uniformiser les interactions.

Tout langage de communication multi-agents est représenté par une structure de donnée comprenant les champs suivants :

- Emetteur
- Récepteur
- Langage utilisé

- L'ontologie : est utilisé pour identifier la source du vocabulaire, en plus est un moyen pour le receveur d'interpréter le contenu du message.
- Contenu du message

a) KQML (Knowledge Query and Manipulation Language)

Défini aux USA par la DARPA-KSE3 (American knowledge Sharing (KSE) un projet de l'université de Maryland.), c'est une approche basée sur les actes de langage visant à résoudre l'interopération entre des agents en prenant en compte une diversité des langages de communication.

La communication est considérée comme un ensemble d'échanges de messages de type déclaratif et assertifs. La force illocutoire appliquée au contenu du message est prise en compte dans cette approche. Chaque message comprend trois couches :

- Couche de contenu : qui spécifie le contenu réel du message d'un agent.
- Couche de communication : qui décrit tous les paramètres de communication de bas niveau.
Par exemple : l'identificateur de l'agent émetteur et celui de l'agent récepteur, l'identificateur de la communication...etc.
- Couche de message : qui est considéré comme le noyau de KQML. Sa fonction principale s'agit d'identifier le protocole de réseau, utilisé pour envoyer le message, et de déterminer le performatif, indiquant le type de ce message (exemple :une affirmation, une requête, une commande,...etc.).

KQML est un protocole de transmission d'informations et de connaissances. La souplesse qu'il offre est que toute information de compréhension du message est incluse dans le message lui-même. Le protocole de base est défini par la structure suivante :

(KQML-performative

```

: Sender < name >
: Receiver < name >
: Language < text >
: Ontology < text >
: Content < expression >
)
```

Où KQML-performative : représente le performatif de l'acte du langage

Le nom de l'agent : l'émetteur du message.

Le nom de l'agent : le récepteur du message.

- Sender : Le nom de l'agent émetteur du message.
- Receiver : Le nom de l'agent récepteur du message.
- Language : permet de préciser le langage dans lequel le message est exprimé.
- Ontology : précise le vocabulaire des « Mots » utilisés dans le message.
- Content : Le contenu réel du message.

Les termes Content, language et ontology permettent de délimiter la sémantique du message.

Une expression KQML est composée d'un message interne spécifié en langage KIF (Knowledge Interface Format) basé sur la logique du premier ordre. Le langage du message dans KQML n'est pas restreint à KIF, d'autre langage peuvent être utilisé tel que PROLOG, LISP, SQL... .

Dans KQML, l'émetteur et le récepteur doivent comprendre le langage de communication utilisé, l'ontologie doit être accessible par tous les agents qui communiquent.

Bien qu'il présente un grand intérêt pour les utilisateurs, KQML montre quelques lacunes [40]:

- La signification floue de certains performatifs (exemple : le cas d'un performatif « dénier ».
- Le manque des performatifs promissifs, exprimant l'engagement, auprès d'un tiers, d'accomplir une action.
- L'utilisation de ce langage que pour des communications isolées.

b) FIPA ACL (Foundation for Intelligent physical Agents)

Récemment, la collaboration internationale des membres d'organisation universitaires et industrielles regroupés au sein de FIPA a permis de spécifier des standards dans la technologie agent et vise à favoriser l'interopérabilité des applications, des services et des équipements informatiques basés sur le Paradigme agent. Ils ont défini un certain nombre de spécifications principales d'agents.

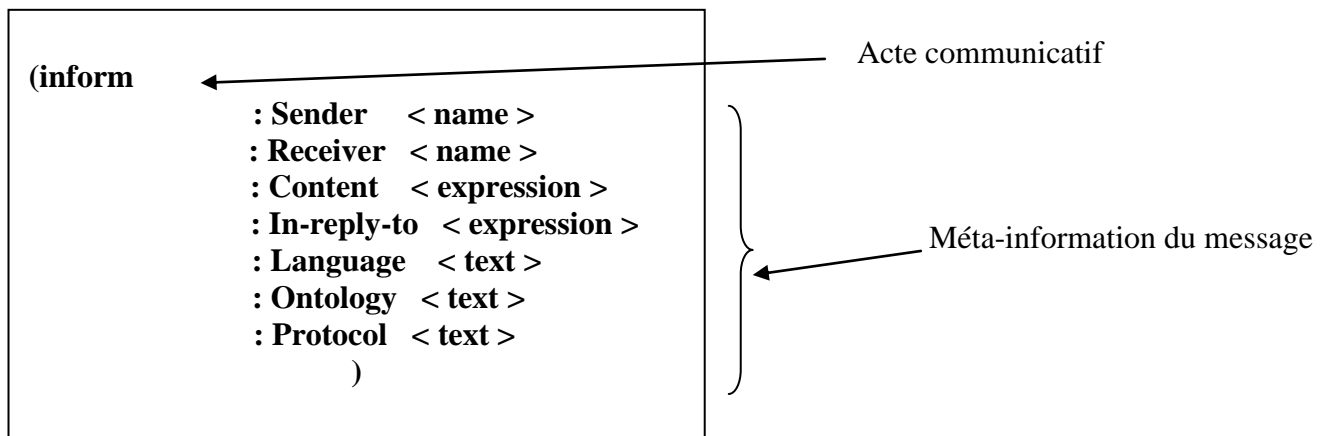
Notamment, un standard de langage de communication agent ACL (Agent Communication Language) [41], a été proposé et spécifié.

Comme KQML ce dernier est basé sur la théorie des actes de langage : les messages sont des actions ou des actes communicatif car ils sont prévus pour effectuer une certaine action en vertu de l'envoi. Les spécifications de FIPA-ACL se composent d'un ensemble de types de message, Et de la

description de leur pragmatique qu'est des effets sur les attitudes mentales des agents (expéditeur et récepteur). Les spécifications décrivent chaque acte communicatif avec une forme narrative et une sémantique formelle basées sur la logique modal. Elles fournissent également la description normative d'un ensemble de protocoles d'interaction de haut niveau, y compris la demande d'action, l'établissement de contrat (contract net) et plusieurs genres de ventes aux enchères.

La sémantique formelle de FIPA ACL se compose en cinq niveaux :

- ✓ Le protocole qui décrit les règles sociales des dialogues entre les agents.
- ✓ Les actes communicatifs (AC) qui définit le type de communication entre les agents (e.g. la demande, la confirmation...etc.).
- ✓ Le Méta-information : concernant le message (l'identificateurs d'un agent émetteur et d'un agent récepteur, le contexte...etc.).
- ✓ Le langage du contenu : qui décrit la grammaire et la sémantique associée, utilisées pour exprimer le contenu d'un message.
- ✓ L'ontologie : qui définit le vocabulaire et les significations des termes et des concepts, employés dans le contenu.



Cette structure d'un message FIPA ACL se représente par :

Un Méta-information du message :

- ✓ Sender : Le nom de l'agent émetteur du message.
- ✓ Receiver : Le nom de l'agent récepteur du message.
- ✓ Content : Le contexte du message.
- ✓ In-reply-to : Le contenu réel du message.
- ✓ Language : permet de préciser le langage dans lequel le message est exprimé.

- ✓ Ontology : précise le vocabulaire des « Mots » utilisés dans le message.
- ✓ Protocol : le protocole du message.

Et un acte communicatif (inform).

FIPA-ACL est superficiellement semblable à KQML. Sa syntaxe est identique à celle de KQML excepté différents noms pour quelques primitifs réservés. Le message FIPA ACL est représenté sous la forme d'une liste, contenant le type de l'acte communicatif (INFORM, REQUEST ...etc.). Ainsi, il maintient l'approche de KQML de distinguer le langage externe du langage interne. Le langage externe définit la signification prévue du message; le langage interne ou le contenu, dénote l'expression à laquelle s'appliquent les croyances, les désirs, et les intentions des interlocuteurs.

3.2.3.2 Négociation

Dans le cas des agents intelligents et dans les systèmes multi-agents, la négociation est une composante de base de l'interaction surtout parce que les agents sont autonomes [42]; il n'y a pas de solution imposée à l'avance et les agents doivent arriver à trouver des solutions dynamiquement, pendant qu'ils résolvent les problèmes.

On peut citer plusieurs définitions pour la négociation. [43] disent que:

- "Par négociation, on entend une discussion dans laquelle des individus intéressés échangent des informations et arrivent à un accord en commun."
- Une autre définition qui s'appuie sur des considérations psychologiques et pour laquelle le conflit est l'élément de base.
- La négociation est caractérisée par [36] :
 - Un nombre faible d'agents impliqués dans le processus,
 - Un protocole minimal d'actions : proposer, évaluer, modifier et accepter ou refuser une solution.
- "La négociation est le processus par lequel plusieurs individus prennent une décision commune. Les participants expriment d'abord des demandes contradictoires, puis ils essaient de trouver un accord par concession ou par la recherche de nouvelles alternatives."

Modélisation de la négociation

Pour modéliser la négociation dans un logiciel multi-agents il faut alors prendre en compte les aspects suivants:

- **Le langage de négociation** : le langage utilisé par les agents pour échanger des informations pendant la négociation; le langage de négociation est composé d'un ensemble de primitives de communication.
- **Le protocole de négociation** : il s'agit de proposer des modèles généraux de comportement des agents en situation de négociation.
- **L'objet de négociation** : un objet abstrait qui comprend les attributs qu'on veut négocier;(le prix, le temps nécessaire pour satisfaire une commande, la qualité des produits, etc.)
- **Le processus de décision** : des études sur les algorithmes de comparaison des sujets de négociation, mais aussi sur les fonctions d'utilité ou encore sur les caractéristiques des préférences des agents (positions, concessions, et critères de compromis).

Techniques de négociation

De nombreux travaux ont proposé des techniques de négociation. Elles sont Soit centrées sur l'environnement c'est à dire voir comment on peut agir sur l'environnement, elle doivent avoir les propriétés suivantes : efficacité, stabilité, simplicité, répartition, symétrie.

Soit sur les agents : Ici le problème n'est plus d'adapter le contexte à la négociation, mais le comportement de l'agent compte tenu des propriétés du contexte donné .La plupart des stratégies de négociation de ce type ont été faites pour des problèmes spécifiques.

3.2.3.2.3 Coopération

La coopération est une caractéristique très importante dans les systèmes multi agents. En effet, une résolution distribuée de problème est le résultat d'interaction coopérative entre les différents agents.

La coopération consiste à faire en sorte que des agents travaillent à la satisfaction d'un but commun, ou tout de moins à la satisfaction de tous les buts individuels.

La coopération dans un système multi-agents a pour but de permettre à l'agent de :

- Déléguer la tâche qu'il ne sait pas résoudre à un autre agent dont il connaît les compétences.
- Interrompre son plan d'exécution pour aider les autres agents.
- Prendre en considération les informations venant d'autres agents.
- Mettre à jour les connaissances globales du système.

Un agent coopératif, peut coopérer suivant les modèles suivants :

- Coopération par partage de tâche et de résultats, avec la possibilité de prendre en compte localement les plans des autres
- Commande : un agent supérieur A décompose le problème en sous-problèmes qu'il répartit entre les autres agents X. Ceux-ci le résolvent et renvoient les solutions partielles à A.
- Appel d'offre : A décompose le problème en sous-problèmes dont il diffuse la liste. Chaque agents Xi qui le souhaite envoie une offre, A choisit parmi celles-ci et distribue les sous-problèmes. Le système travaille ensuite en mode commande.
- Compétition : Dans ce mode, A décompose et diffuse la liste des sous-problèmes comme dans le mode appel d'offres, chaque agent Xi résout un ou plusieurs sous-problèmes et envoie les résultats correspondants à A qui à son tour fait le tri.

3.2.3.2.4 Planification

Le principe de cette technique repose sur le découpage de l'action en deux phases : dans la première, on réfléchit sur l'ensemble des actions à effectuer pour atteindre un but en produisant un ensemble de plans, dans la seconde, on choisit l'un de ces plans que l'on exécute. La planification est basée sur des conventions préétablies, elle détermine quelles sont les tâches à effectuer, quand et par qui, et s'applique à des situations stables et prévisibles.

3.2.3.3 Contrôle

Le contrôle dans un système multi agents concerne la façon dont les divers agents vont agir pour enchaîner leurs raisonnements et choisir les meilleures actions à entreprendre parmi les diverses actions possibles. Ainsi le contrôle dirige et guide le processus de résolution et de construction de la solution globale.

Le contrôle dans un système multi-agents est distribué de manière interne dans l'agent lui-même. Il consiste à déclencher une tâche de manière asynchrone ou périodique. Celui-ci permet à un agent de consulter les nouveaux messages et effectuer les actions appropriées. Ainsi, l'état de l'agent évoluera avec l'évolution de la résolution globale.

3.2.3.4 Organisation

La notion d'organisation joue un rôle fondamental au sein des systèmes multi-agents : c'est un moyen permettant d'organiser logiquement les agents, c'est un réseau de communication par défaut et c'est un media de recherche d'agent, de rôle ou de compétence.

Il existe deux types d'organisation : l'organisation statique et l'organisation dynamique :

a- Organisation statique d'un système multi-agents

De nombreux modèles organisationnels se sont inspirés de la théorie des organisations économiques. Généralement, les systèmes de ce type laissent à leur concepteurs le soin de définir eux-mêmes a priori l'organisation du système multi-agents. Les architectures multi-agents hiérarchiques sont des exemples d'organisation statiques de SMA. Dans cette approche, les liens relationnels entre les agents sont tout à fait préétablis. Ils n'émergent pas de l'évolution du système. Le contrôle du système est globalement.

b- Organisation dynamique d'un système multi-agents :

Dans ce modèle, la structure organisationnelle du système multi-agents est créée dynamiquement par les agents eux-mêmes. La réalisation des buts d'un agent stimule le Phénomène de réorganisation de société multi-agents. Ce phénomène est géré par un processus collectif ou individuel mettant en jeu des mécanismes évolution. Il existe différentes approches qui traitent des problèmes d'adaptation dans les systèmes multi-agents. Une de ces approches est l'apprentissage individuel ou collectif dans les systèmes multi-agents par des techniques symboliques, neuronales, des algorithmes génétiques, etc. [33] cette approche se fonde sur l'apprentissage des comportements organisationnels en se servant des résultats observés durant les précédentes interactions positives et négatives entre agents. Les liens relationnels entre les agents émergent de ces interactions.

Chapitre 4

METHODE DE RESOLUTION MECANIQUE DU MODELE HYDRODYNAMIQUE

4. Le modèle hydrodynamique

4.1 Méthode de résolution

Le mouvement dynamique du piston, régi par le système d'équations non linéaires du second ordre (2.1), se présente comme un problème de valeurs initiales. Les excentricités recherchées sont cycliques et se répètent identiquement sur une période de temps qui correspond à un cycle moteur, c'est à dire une rotation de 4π du vilebrequin.

Au lieu de résoudre ce système par des méthodes d'intégration directe, on utilise la méthode numérique itérative suivante:

1^{ère} étape: Déterminer la solution sans effet inertiel

Dans une première étape la solution est recherchée dans le cas où l'on suppose que les forces d'inertie sont nulles, c'est à dire que l'on a

$$\ddot{e}_h = \ddot{e}_b = 0$$

Le système d'équations du second ordre (2.1) devient alors un système du premier ordre non linéaire en e_h , e_b , \dot{e}_h et \dot{e}_b .

- **Le système d'équations du premier ordre**

Si on élimine les termes d'inertie, le système d'équations (2.1) s'écrit

$$\begin{Bmatrix} F + F_S \\ M + M_S \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (4.1)$$

F et M , forces hydrodynamiques données par les équations 2.32 à 2.35, sont fonctions de e_h , e_b , \dot{e}_h et \dot{e}_b tandis que F_S et M_S sont les forces latérales de paroi qui ne dépendent que des données du mouvement primaire ou axial. Ces forces sont donc indépendantes de e_h , e_b , \dot{e}_h et \dot{e}_b , caractéristiques du mouvement secondaire.

Ici la position du piston (e_h , e_b) au début de chaque étape de temps est connue. La vitesse latérale correspondante (\dot{e}_h , \dot{e}_b) est reliée à la position du piston sous la forme fonctionnelle suivante:

$$\begin{Bmatrix} F[\dot{e}_h(\bar{t}), \dot{e}_b(\bar{t}), e_h(\bar{t}), e_b(\bar{t})] \\ M[\dot{e}_h(\bar{t}), \dot{e}_b(\bar{t}), e_h(\bar{t}), e_b(\bar{t})] \end{Bmatrix} - \begin{Bmatrix} F_S(\bar{t}) \\ M_S(\bar{t}) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (4.2)$$

où F et M sont des fonctionnelles données par les équations 2.20-2.23.

- **Méthode de résolution du système d'équations du premier ordre**

L'équation 4.2 est non linéaire et doit être résolue à chaque pas de temps (correspondant au pas de la rotation du vilebrequin) par la méthode itérative de Newton-Raphson. A la $i^{\text{ème}}$ itération une nouvelle vitesse du piston est donnée par ;

$$\begin{Bmatrix} \dot{e}_h \\ \dot{e}_b \end{Bmatrix}^{(i+1)} = \begin{Bmatrix} \dot{e}_h \\ \dot{e}_b \end{Bmatrix}^{(i)} - \begin{bmatrix} \frac{\partial F}{\partial \dot{e}_h} & \frac{\partial F}{\partial \dot{e}_b} \\ \frac{\partial M}{\partial \dot{e}_h} & \frac{\partial M}{\partial \dot{e}_b} \end{bmatrix}^{-1} \begin{Bmatrix} F^{(i)} - F_S \\ M^{(i)} - M_S \end{Bmatrix} \quad (4.3)$$

Les dérivées partielles sont obtenues par dérivation des équations 2.32-2.33

Une fois déterminée la vitesse du piston, les excentricités, donnant la position transversale du piston à la fin de l'étape de temps actuelle, sont données par les formules d'Euler

$$e_h(\bar{t} + \Delta\bar{t}) = e_h(\bar{t}) + \Delta\bar{t} \cdot \dot{e}_h(\bar{t}) \quad (4.4)$$

$$e_b(\bar{t} + \Delta\bar{t}) = e_b(\bar{t}) + \Delta\bar{t} \cdot \dot{e}_b(\bar{t}) \quad (4.5)$$

2^{ème} étape: Résolution avec effet inertiel

Une fois la solution $(e_h(\bar{t}), e_b(\bar{t}))$, obtenue dans le cas où les forces d'inertie ne sont pas prise en compte, est cyclique c'est-à-dire

$$e_h(\bar{t}) = e_h(\bar{t} + 4\pi) \quad (4.6)$$

$$e_b(\bar{t}) = e_b(\bar{t} + 4\pi) \quad (4.7)$$

on forme de nouveau un système d'équation différentielles du premier ordre semblable au système d'équations 4.2. L'idée est d'incorporer les termes d'inertie comme forces extérieures dans le système 4.2. Les accélérations latérales $\ddot{e}_h(\bar{t})$ et $\ddot{e}_b(\bar{t})$ sont calculées par la méthode des différences finies à partir de la solution $\dot{e}_h(\bar{t})$, $\dot{e}_b(\bar{t})$, $e_h(\bar{t})$ et $e_b(\bar{t})$. Ainsi les forces d'inertie sont connues à chaque pas de l'itération de Newton.

La solution est alors recalculée avec les forces additionnelles d'inertie. Cette procédure est répétée jusqu'à ce que

- a) une précision sur $\ddot{e}_h(\bar{t})$ et $\ddot{e}_b(\bar{t})$ soit satisfaite ou
- b) une précision sur l'équation (2.1) soit atteinte.

Dans ce dernier cas on utilise la méthode qui consiste à déterminer le résidu de l'équation (2.1) en introduisant à chaque pas de temps la solution dans cette équation puis à comparer la valeur obtenue avec la précision donnée.

Dans la procédure d'obtention de chaque solution cyclique l'équation du premier ordre provenant de l'équation (4.2) se met sous la forme :

$$\begin{Bmatrix} F[\dot{e}_h(\bar{t}), \dot{e}_b(\bar{t}), e_h(\bar{t}), e_b(\bar{t})] \\ M[\dot{e}_h(\bar{t}), \dot{e}_b(\bar{t}), e_h(\bar{t}), e_b(\bar{t})] \end{Bmatrix} - \begin{Bmatrix} F_S(\bar{t}) \\ M_S(\bar{t}) \end{Bmatrix} - \begin{Bmatrix} F_{inert} \\ M_{inert} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (4.8)$$

où $\begin{Bmatrix} F_{inert} \\ M_{inert} \end{Bmatrix} = \begin{Bmatrix} F_{IP} + F_{IC} \\ M_{IC} \end{Bmatrix}$ est la charge d'inertie dont les différents termes F_{IP} , F_{IC} et M_{IC} valeur est

donnée par les équations :

$$F_{IP} = -m_{axe} \cdot \left[\ddot{e}_h + \frac{a}{L} (\ddot{e}_b - \ddot{e}_h) \right], \quad F_{IC} = -m_{pist} \cdot \left[\ddot{e}_h + \frac{b}{L} (\ddot{e}_b - \ddot{e}_h) \right], \quad M_{IC} = \frac{-I_{pist} \cdot (\ddot{e}_h - \ddot{e}_b)}{L}.$$

qui peuvent encore s'écrire :

$$\begin{Bmatrix} F_{inert} \\ M_{inert} \end{Bmatrix} = [M] \begin{Bmatrix} \ddot{e}_h \\ \ddot{e}_b \end{Bmatrix}$$

de même ici la position du piston (e_h , e_b) au début d'une étape de temps est connue. La vitesse latérale correspondante (\dot{e}_h , \dot{e}_b) est reliée à la position du piston sous la forme fonctionnelle de l'équation 4.8, qui s'écrit sous la forme :

$$\begin{Bmatrix} F[\dot{e}_h(\bar{t}), \dot{e}_b(\bar{t}), e_h(\bar{t}), e_b(\bar{t})] \\ M[\dot{e}_h(\bar{t}), \dot{e}_b(\bar{t}), e_h(\bar{t}), e_b(\bar{t})] \end{Bmatrix} - \begin{Bmatrix} F_e(\bar{t}) \\ M_e(\bar{t}) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (4.9)$$

où

$F_e(\bar{t})$ et $M_e(\bar{t})$ représentent la somme des forces et des moments d'inertie et latérales de paroi.

Le reste du calcul se fait alors de manière identique au cas de la solution sans effet inertielle.

On peut maintenant établir les organigrammes du code de calcul et en premier lieu l'organigramme principal.

4.2 Algorithme de résolution du modèle hydrodynamique

L'algorithme principal de résolution du modèle hydrodynamique est alors donné par :

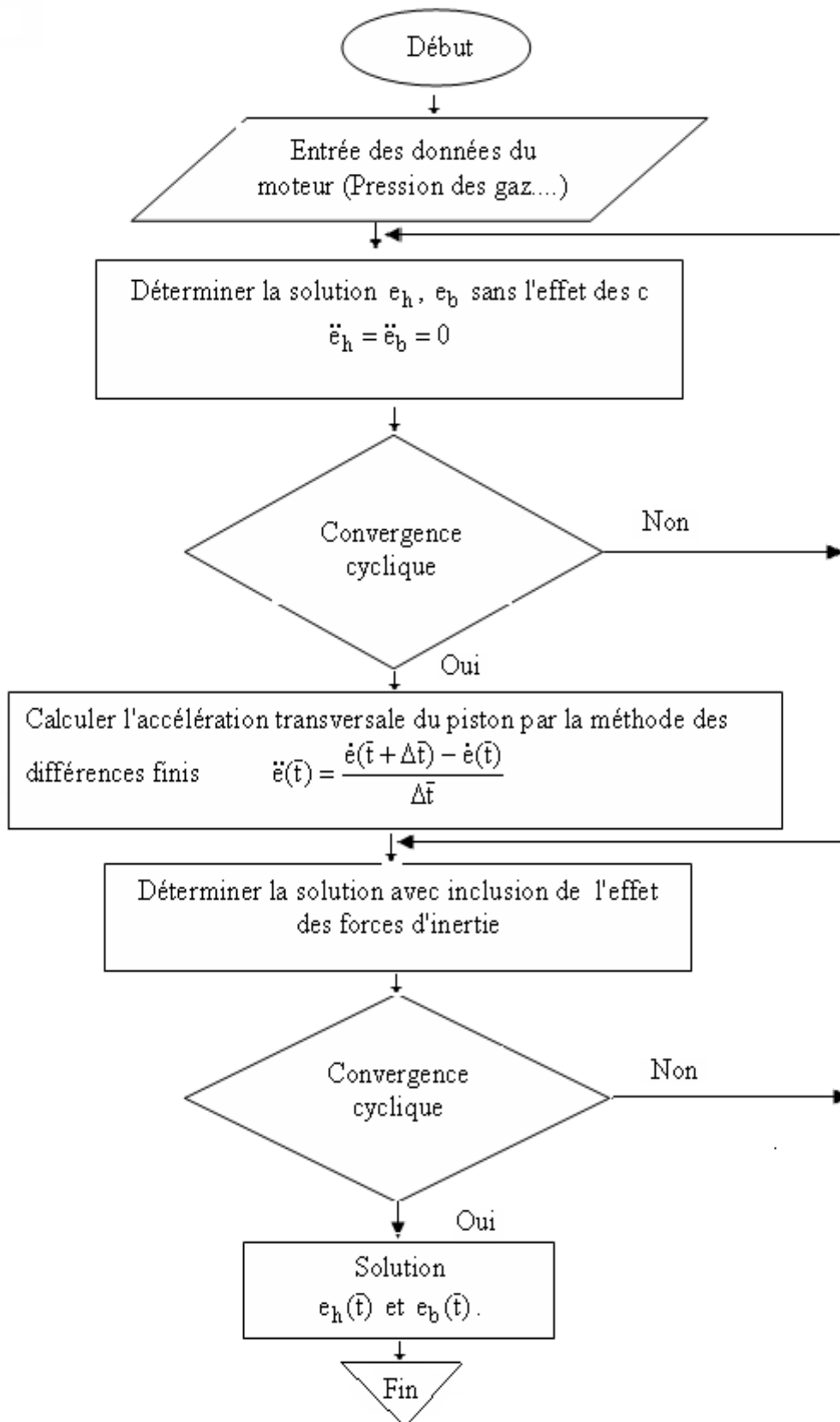


Figure 4.1: Organigramme principal de résolution

La procédure d'obtention de la solution cyclique est donnée par l'organigramme suivant:

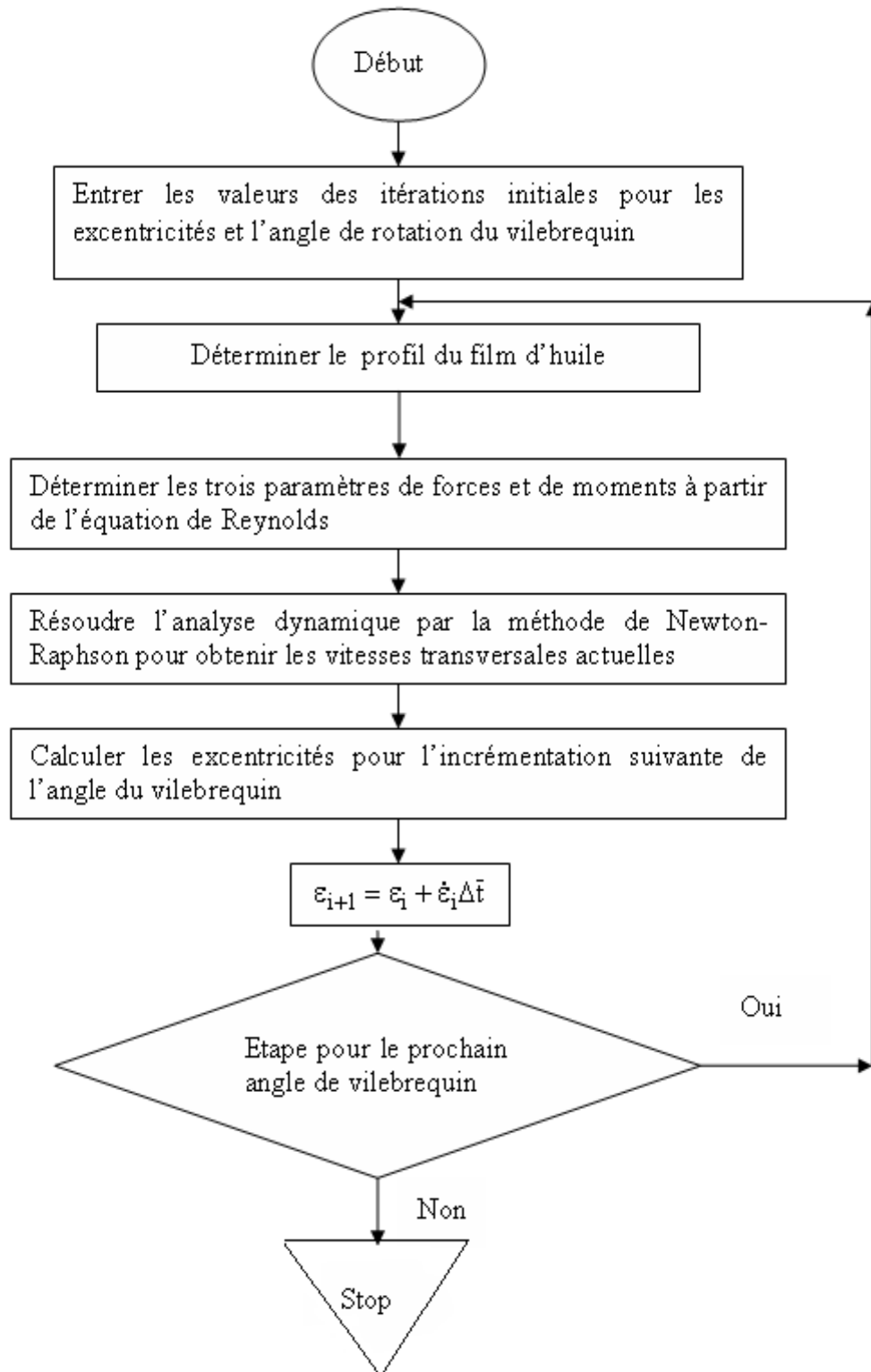


Figure 4.2 : Organigramme schématisant la solution cyclique

La procédure d'obtention de la solution cyclique se base principalement sur la performance du mouvement hydrodynamique. On a vu que le mouvement hydrodynamique peut être décomposé en trois

mouvements simples : axial ou glissement, de translation latéral et de rotation. Chacun des mouvements est obtenu par les sous programmes suivants :

4.3 Algorithme de Newton-Raphson modifié

L'algorithme de Newton-Raphson modifié est utilisé pour résoudre le système d'équations non linéaires 4.9. Les inconnues sont les vitesses transversales $\dot{e}_h(\bar{t})$, $\dot{e}_b(\bar{t})$. Comme l'indique l'organigramme principal, la méthode de Newton-Raphson est appliquée à chaque itération angulaire ϕ du vilebrequin. Chaque pas angulaire correspond à un pas de temps adimensionné $\bar{t} = t\omega$, ω étant la vitesse de rotation du moteur.

L'utilisation de la méthode de Newton-Raphson est essentielle à cette étape, elle permet de corriger dans l'équation 4.9 les vitesses actuelles par des nouvelles vitesses qui correspondent aux excentricités obtenues à la fin de l'étape précédente. Dans l'équation 4.3, on a vu que les quantités $F^{(i)}$ et $M^{(i)}$ sont données à la $i^{\text{ème}}$ itération par :

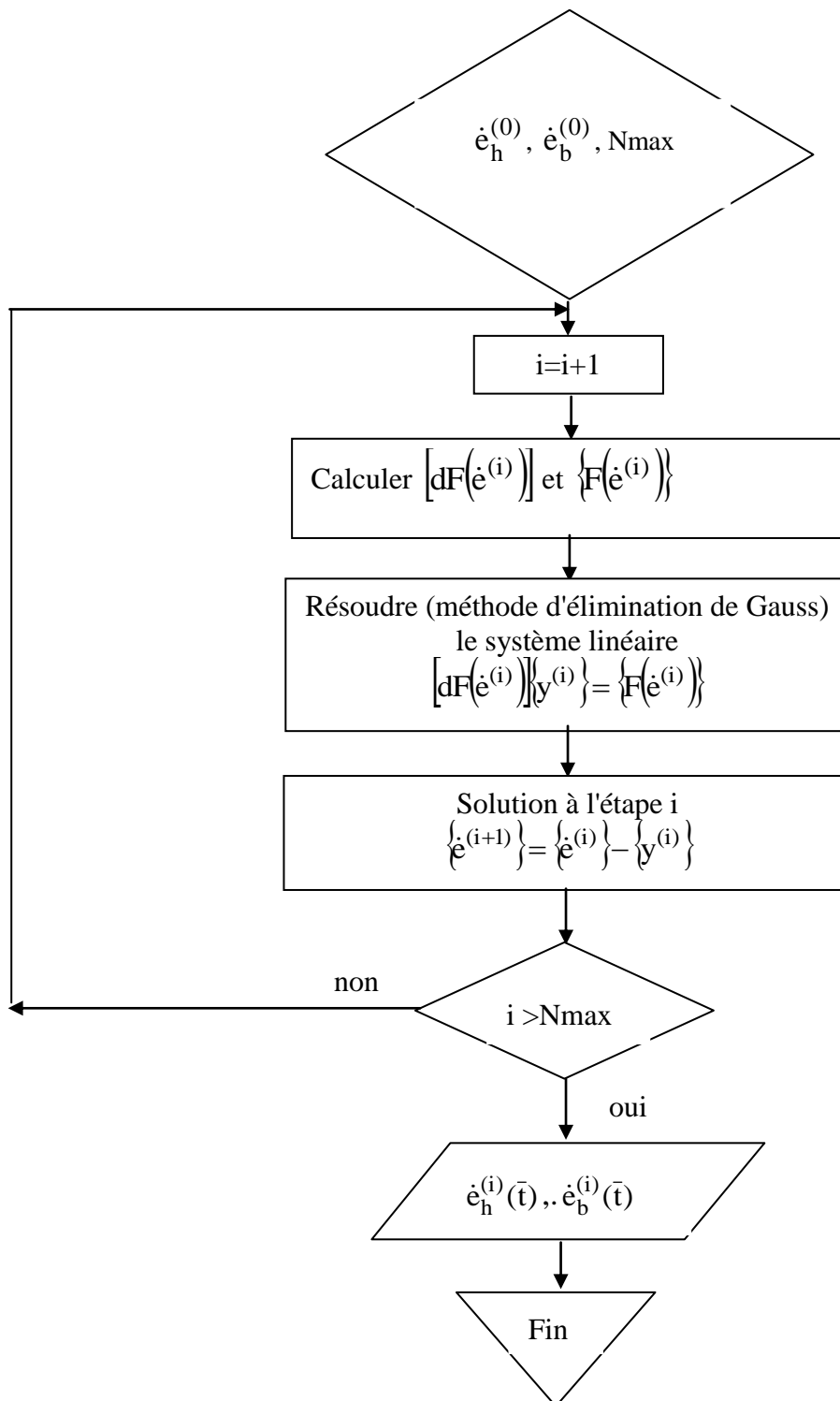
$$F^{(i)} = -F_u \bar{V} + \dot{\varepsilon}_h^{(i)}(\bar{t})F_{1C}^{(i)} + \left[\dot{\varepsilon}_b^{(i)}(\bar{t}) - \dot{\varepsilon}_h^{(i)}(\bar{t}) \right] F_{2C}^{(i)} \quad (4.10)$$

$$M^{(i)} = -M_u \bar{V} + \dot{\varepsilon}_h^{(i)}(\bar{t})M_{1C}^{(i)} + \left[\dot{\varepsilon}_b^{(i)}(\bar{t}) - \dot{\varepsilon}_h^{(i)}(\bar{t}) \right] M_{2C}^{(i)} \quad (4.11)$$

Les termes (F_u, M_u) , $(F_{1C}^{(i)}, M_{1C}^{(i)})$ et $(F_{2C}^{(i)}, M_{2C}^{(i)})$ sont les charges obtenues par les mouvements axial, d'écrasement et de rotation respectivement.

- **Le procédé itératif**

Présenté sous la forme de l'équation 4.9 le processus de calcul n'offre guère d'intérêt puisqu'on doit inverser la matrice jacobienne à chaque itération i . Une procédure équivalente est la méthode de Newton-Raphson modifiée. L'algorithme est le suivant :



$$\text{où } \{F(\dot{e}^{(i)})\} = \begin{Bmatrix} F^{(i)} - F_e \\ M^{(i)} - M_e \end{Bmatrix}$$

Figure 4.3 : Organigramme de Newton-Raphson modifié

4.4 Organigramme des mouvements composés

a) Méthode de résolution de l'équation de Reynolds

Le but de la résolution de l'équation de Reynolds est de déterminer les forces hydrodynamiques \bar{F} et \bar{M} . Ces forces, une fois déterminées, seront introduites dans l'équation dynamique (2.1). On rappelle que l'équation dynamique est résolue à chaque pas angulaire du vilebrequin selon le processus itératif donné par l'algorithme principal 4.1 du modèle hydrodynamique et ce en vue d'avoir les excentricités e_h et e_b . Pour obtenir les forces hydrodynamiques il faut obtenir la pression dans le film d'huile en résolvant l'équation de Reynolds. La pression est obtenue si on connaît l'épaisseur du film lubrifiant. Le processus de résolution est donc itératif. La résolution de l'équation se fait à l'aide de la méthode des différences finies.

b) Résolution de l'équation de Reynolds

Utilisant un schéma 2D de différences finies centrées, les différents termes de l'équation de Reynolds, relatives aux variables \bar{y} et θ , auxquelles correspondent les indices de discrétisation i et j respectifs, peuvent être obtenus par le calcul en chaîne suivant :

- On a

$$\frac{\partial}{\partial \theta} \left(\bar{h}^3 \frac{\partial \bar{p}}{\partial \theta} \right) + \frac{\partial}{\partial \bar{y}} \left(\bar{h}^3 \frac{\partial \bar{p}}{\partial \bar{y}} \right) = -\bar{U} \left(\frac{(\varepsilon_b - \varepsilon_h) \cos \theta}{\bar{L}} + \frac{\partial \bar{f}}{\partial \bar{y}} \right) + \beta \left(\dot{\varepsilon}_h \cos \theta + (\dot{\varepsilon}_b - \dot{\varepsilon}_h) \frac{\bar{y}}{\bar{L}} \cos \theta \right) \quad (2.1)$$

- en dérivant par rapport à la variable axiale \bar{y} (indice i)

$$\frac{\partial}{\partial \bar{y}} \left(\bar{h}^3 \frac{\partial \bar{p}}{\partial \bar{y}} \right) = 3\bar{h}^2 \frac{\partial \bar{h}}{\partial \bar{y}} \frac{\partial \bar{p}}{\partial \bar{y}} + \bar{h}^3 \frac{\partial^2 \bar{p}}{\partial \bar{y}^2}$$

- puis en discrétisant les termes du second membre par différences finis

$$\frac{\partial \bar{p}}{\partial \bar{y}} = \frac{\bar{p}_{i+1,j} - \bar{p}_{i-1,j}}{2\Delta \bar{y}}$$

$$\frac{\partial^2 \bar{p}}{\partial \bar{y}^2} = \frac{\bar{p}_{i+1,j} - 2\bar{p}_{i,j} + \bar{p}_{i-1,j}}{\Delta \bar{y}^2}$$

le premier terme de l'équation s'écrit alors

$$\frac{\partial}{\partial \bar{y}} \left(\bar{h}^3 \frac{\partial \bar{p}}{\partial \bar{y}} \right) = 3\bar{h}^2 \frac{\partial \bar{h}}{\partial \bar{y}} \left(\frac{\bar{p}_{i+1,j}}{2\Delta \bar{y}} - \frac{\bar{p}_{i-1,j}}{2\Delta \bar{y}} \right) + \frac{\bar{h}^3}{\Delta \bar{y}^2} \bar{p}_{i+1,j} + \frac{\bar{h}^3}{\Delta \bar{y}^2} \bar{p}_{i-1,j} - \frac{2\bar{h}^3}{\Delta \bar{y}^2} \bar{p}_{i,j}$$

$$= \bar{p}_{i+1,j} \left(\frac{3\bar{h}^2}{2\Delta\bar{y}} \frac{\partial\bar{h}}{\partial\bar{y}} + \frac{\bar{h}^3}{\Delta\bar{y}^2} \right) + \bar{p}_{i-1,j} \left(\frac{\bar{h}^3}{\Delta\bar{y}^2} - \frac{3\bar{h}^2}{2\Delta\bar{y}} \frac{\partial\bar{h}}{\partial\bar{y}} \right) - \frac{2\bar{h}^3}{\Delta\bar{y}^2} \bar{p}_{i,j}$$

de même en changeant la variable θ en \bar{y} le deuxième terme de l'équation s'écrit:

$$\frac{\partial}{\partial\theta} \left(\bar{h}^3 \frac{\partial\bar{p}}{\partial\theta} \right) = \bar{p}_{i,j+1} \left(\frac{3\bar{h}^2}{2\Delta\theta} \frac{\partial\bar{h}}{\partial\theta} + \frac{\bar{h}^3}{\Delta\theta^2} \right) + \bar{p}_{i,j-1} \left(\frac{\bar{h}^3}{\Delta\theta^2} - \frac{3\bar{h}^2}{2\Delta\theta} \frac{\partial\bar{h}}{\partial\theta} \right) - \frac{2\bar{h}^3}{\Delta\theta^2} \bar{p}_{i,j}$$

Finalement l'équation s'écrit sous la forme discrète condensée

$$\bar{p}_{i,j} A_{i,j} + \bar{p}_{i+1,j} B_{i,j} + \bar{p}_{i-1,j} C_{i,j} + \bar{p}_{i,j+1} D_{i,j} + \bar{p}_{i,j-1} E_{i,j-1} = F_{i,j}$$

où les coefficients introduits sont donnés par

$$A_{i,j} = -2\bar{h}^3 \left(\frac{1}{\Delta y^2} + \frac{1}{\Delta\theta^2} \right)$$

$$B_{i,j} = \frac{3\bar{h}^2}{2\Delta y} \frac{\partial\bar{h}}{\partial y} + \frac{\bar{h}^3}{\Delta y^2}$$

$$C_{i,j} = \frac{\bar{h}^3}{\Delta y^2} - \frac{3\bar{h}^2}{2\Delta y} \frac{\partial\bar{h}}{\partial y}$$

$$D_{i,j} = \frac{3\bar{h}^2}{2\Delta\theta} \frac{\partial\bar{h}}{\partial\theta} + \frac{\bar{h}^3}{\Delta\theta^2}$$

$$E_{i,j} = \frac{\bar{h}^3}{\Delta\theta^2} - \frac{3\bar{h}^2}{2\Delta\theta} \frac{\partial\bar{h}}{\partial\theta}$$

Dans le cas du mouvement axial le second membre s'écrit

$$F_{i,j}^{\text{ax}} = \frac{\varepsilon_b - \varepsilon_h}{L} \cos\theta + \frac{\partial\bar{f}}{\partial\bar{y}}$$

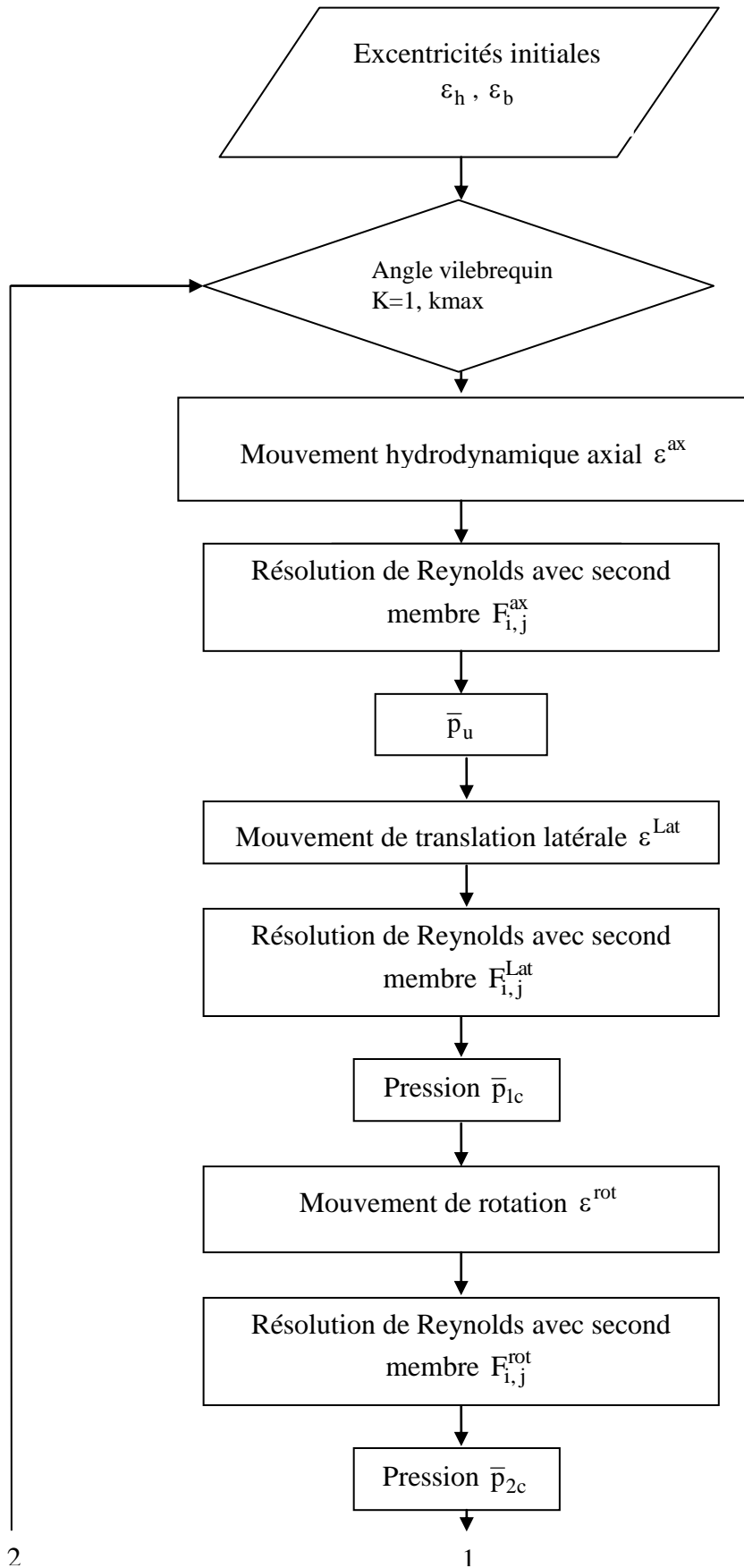
Pour le mouvement latéral le second membre est :

$$F_{i,j}^{\text{Lat}} = \beta \cos\theta$$

et enfin pour le mouvement de rotation le second membre s'écrit :

$$F_{i,j}^{\text{rot}} = \beta \frac{\bar{y}}{L} \cos\theta$$

Les différentes pressions \bar{p}_u , \bar{p}_{1c} et \bar{p}_{2c} étant déterminées par la résolution des différentes équations de Reynolds, la pression totale \bar{p} est alors déterminée par l'équation 2.11, d'où l'organigramme



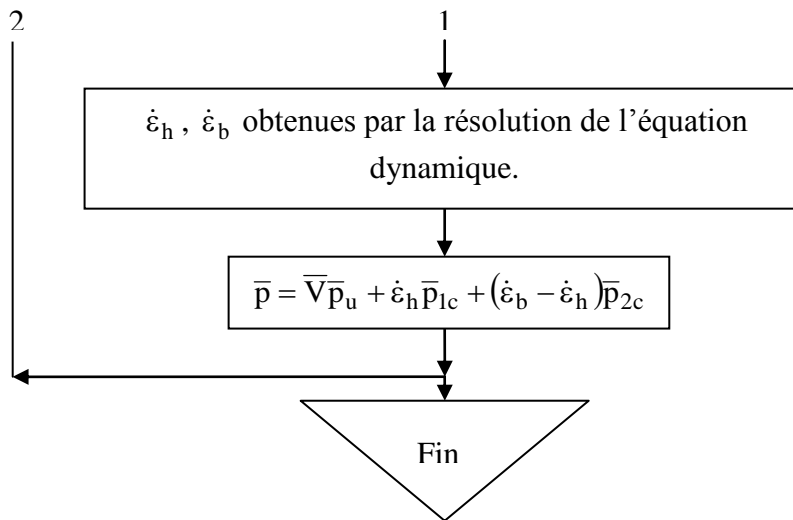


Figure 4.4 : Organigramme de calcul de la pression globale

L'organigramme du mouvement axial est donné par :

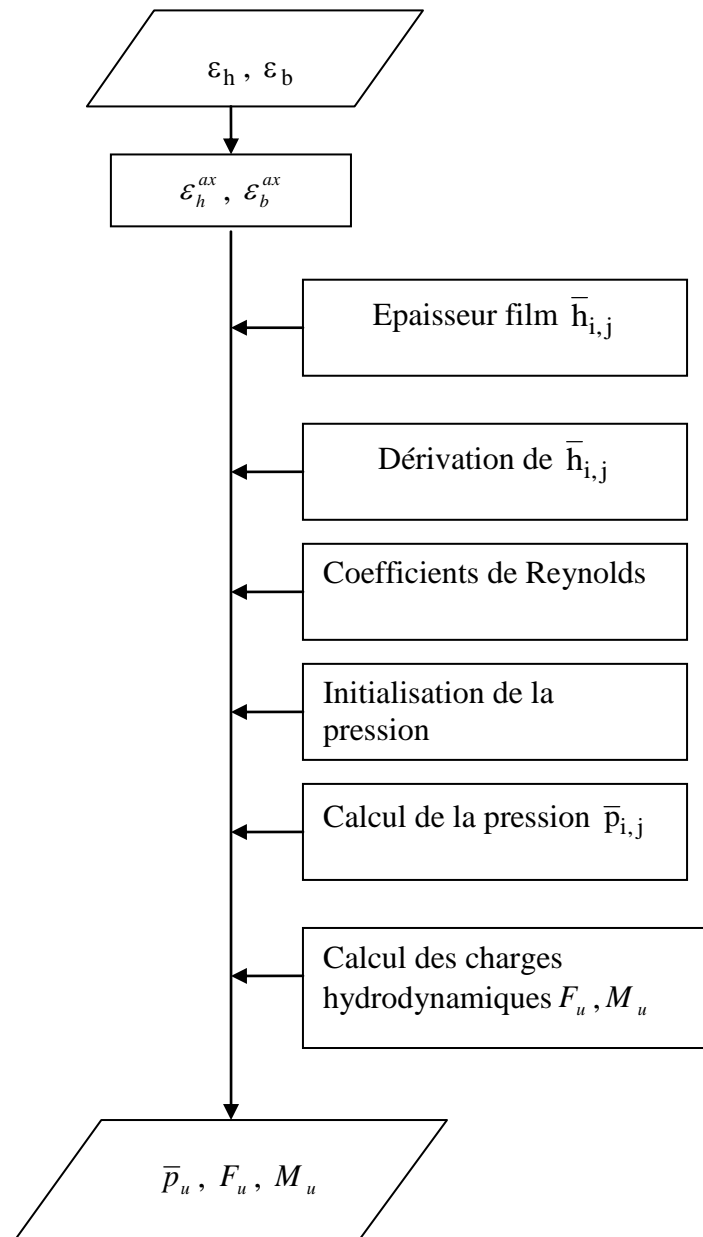


Figure 4.5: Organigramme du mouvement hydrodynamique axial

Les mouvements de translation latéral et de rotation sont obtenus selon la même procédure.

4.5 Organigramme de calcul de la pression

La résolution de l'équation de Reynolds permet de trouver la pression $\bar{p}(i, j)$ en chaque point du maillage du film d'huile. Un système d'équation linéaire est obtenu. Sa résolution est faite à l'aide de la méthode itérative de Gauss-Seidel avec accélération de sur-relaxation (SOR) selon l'organigramme

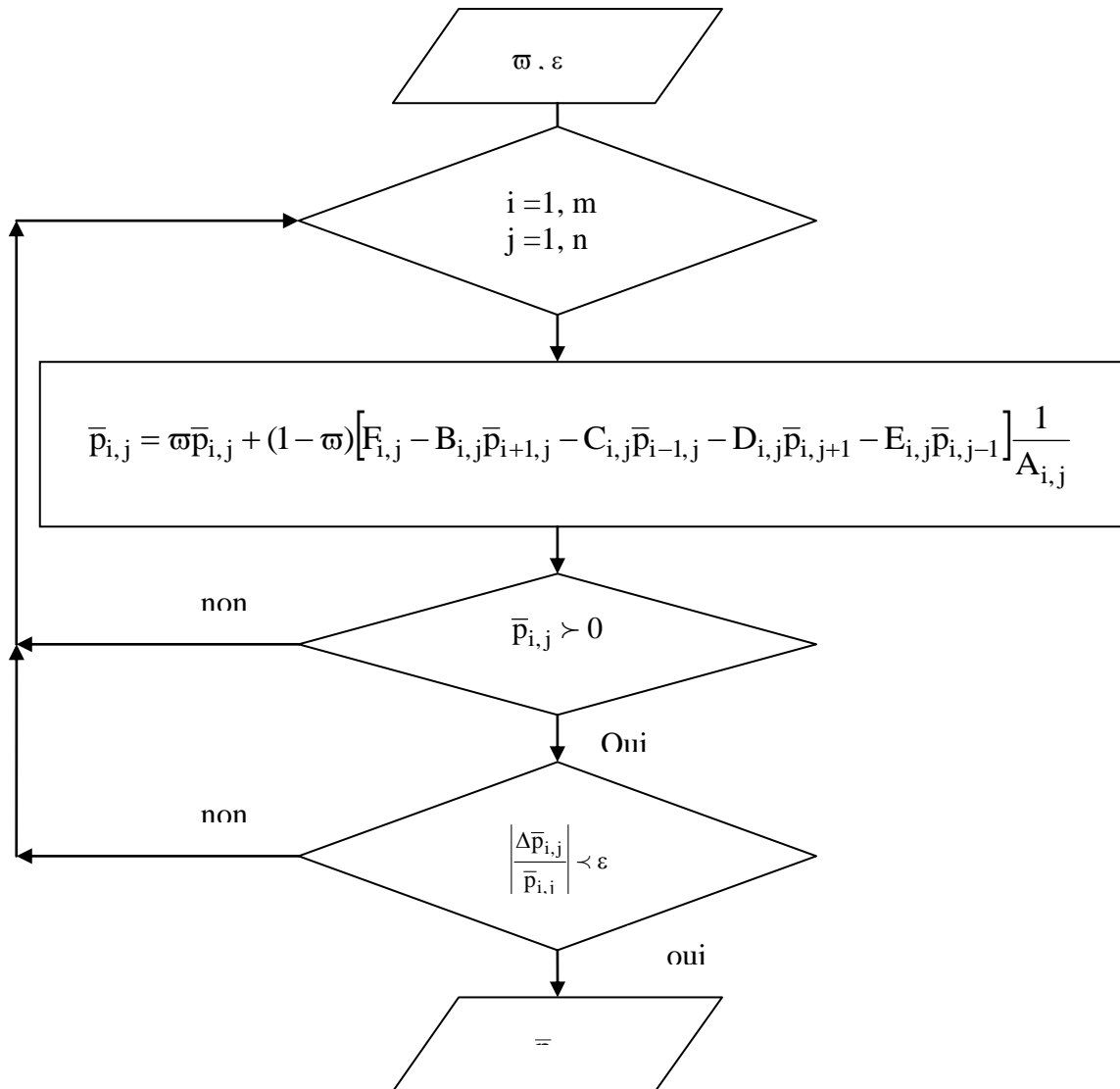


Figure 4.6: Méthode SOR de calcul de la pression hydrodynamique globale

La pression obtenue, on détermine alors numériquement (intégration numérique de la méthode des trapèzes) les charges hydrodynamiques \bar{F} et \bar{M} données par les intégrales 2.30 et 2.31.

4.6 Calcul des charges hydrodynamiques

Les charges hydrodynamiques sont calculées à l'aide des équations 2.30 et 2.31 pour chaque modèle hydrodynamique. Le calcul se fait à chaque pas de temps c'est à dire à chaque pas angulaire du vilebrequin. Les charges globales \bar{F} et \bar{M} sont données par les équations 2.29-2.32. L'organigramme qui donne le couple de charge hydrodynamique est le suivant :

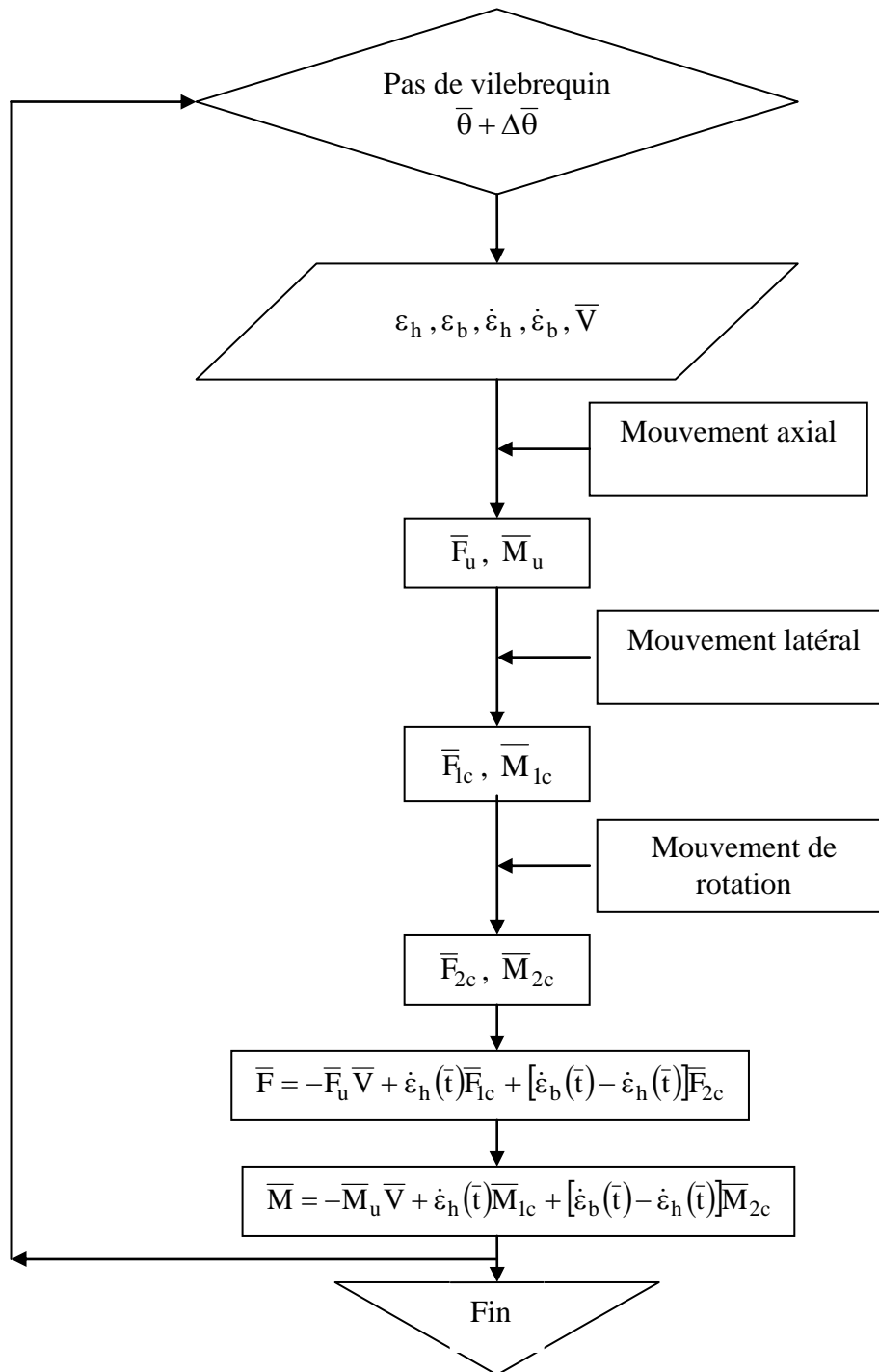
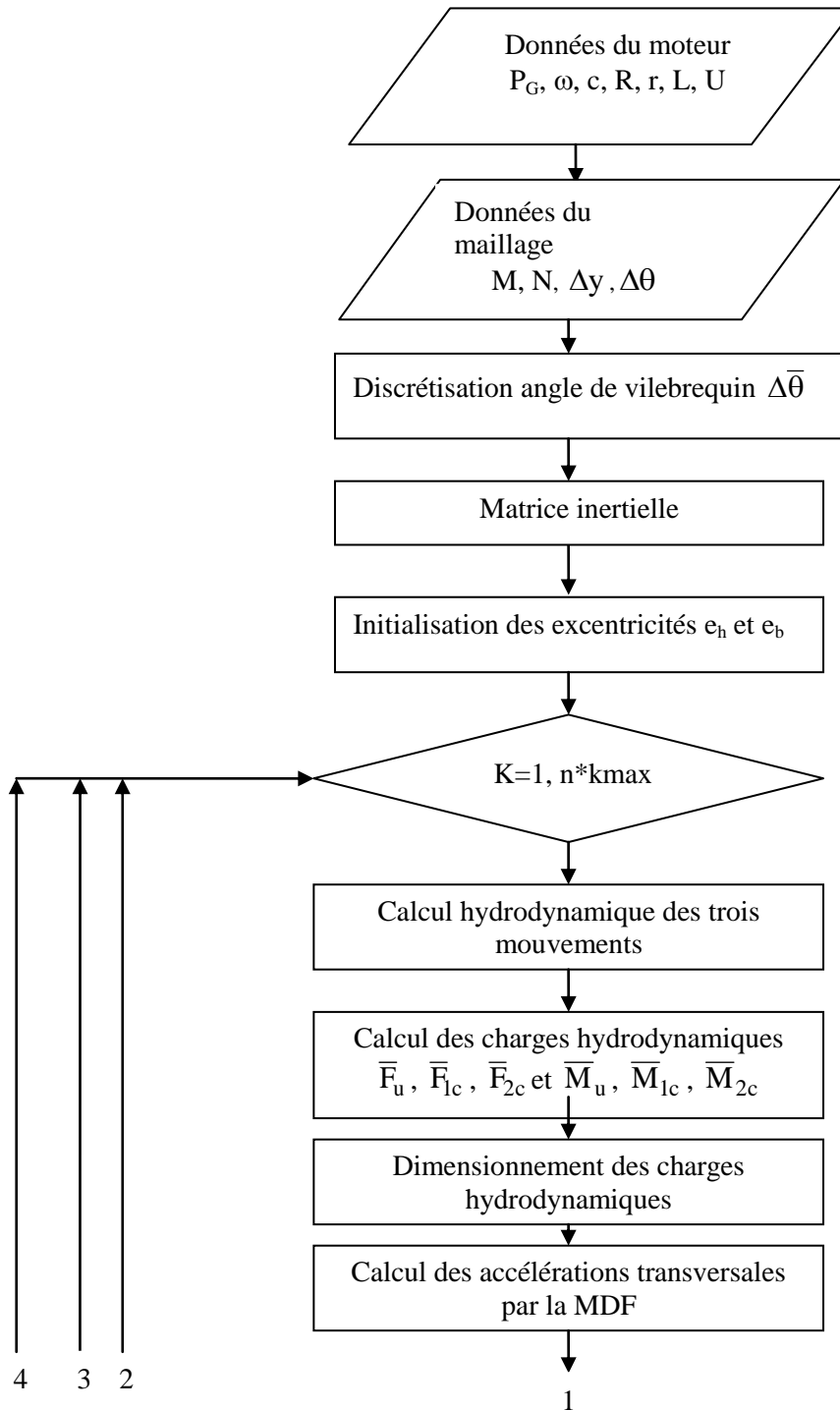
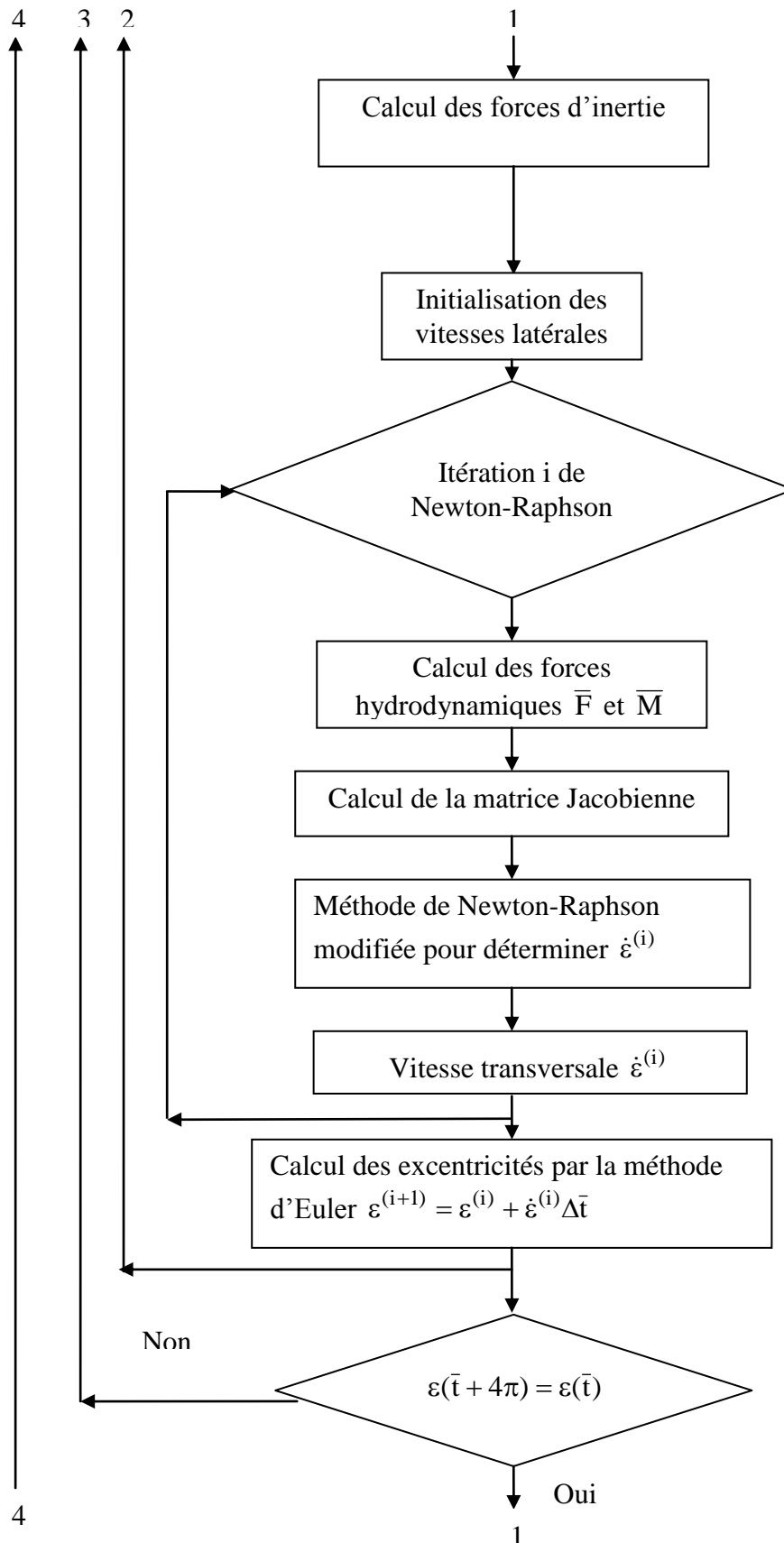


Figure 4.7 : Organigramme de calcul des charges \bar{F} et \bar{M}

4.7 Organigramme de la solution avec effet des forces d'inertie

Maintenant il est possible de donner l'organigramme global détaillé de la résolution du problème hydrodynamique avec effet inertiel du piston. L'organigramme ci-dessous en donne les principales étapes.





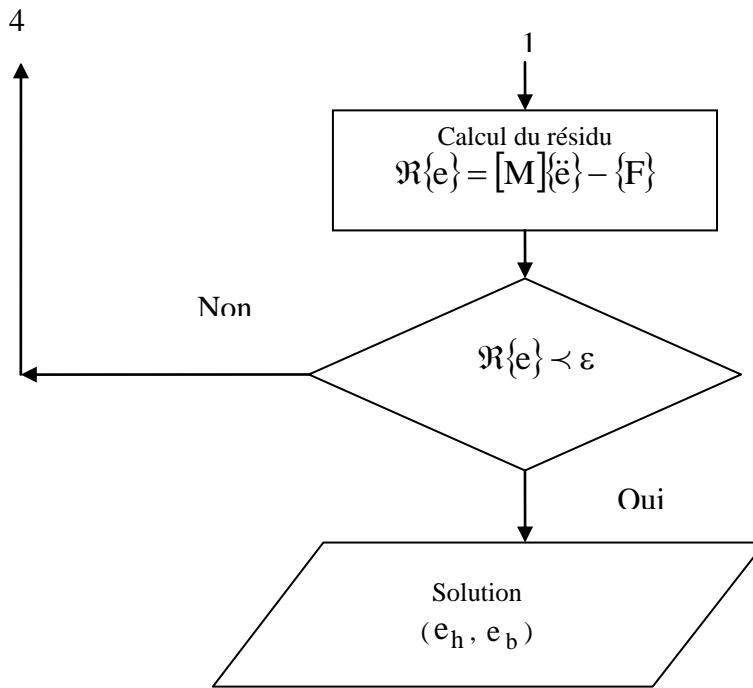


Figure 4.8 : Organigramme de la solution avec effet d'inertie

Chapitre 5

CHOIX DE METHODOLOGIE ET CONCEPTION DU SMA

5.1. Choix de la méthodologie des SMA

5.1.1 Introduction

Une méthodologie est une démarche progressive qui commence par la définition des besoins préalables de l'utilisateur pour arriver à une implémentation d'un système capable de satisfaire les besoins initiaux. Plusieurs travaux [44, 45] ont étudié et classifié selon plusieurs critères les méthodologies multi-agents, dans le but de les améliorer ou pour aller à une unification (ou une standardisation) de la démarche de développement des systèmes multi-agents. Dans [46], Cernuzzi et ses collègues se sont intéressés aux modèles de développement adoptés par les méthodologies multi-agents. Ils ont donné aussi pour chaque méthodologie les phases de développement qu'elle intègre dans sa démarche.

Plusieurs recherches dans le domaine de l'ingénierie orientée agent se sont focalisées sur la proposition de méthodologies pour les systèmes multi-agents; en identifiant les démarches à suivre durant les différentes phases de développement et les abstractions exploitées durant ces phases. Les méthodologies orientées agent propose une approche claire permettant d'analyser, de concevoir et de développer un SMA en utilisant des méthodes et des techniques spécifiques. Pour cela, les méthodologies orientées agent commence par un méta-modèle identifiant les abstractions de base à exploiter durant le développement (par exemple : agent, rôle, but, tâche, ressource, environnement, structure organisationnelle).

Un méta-modèle est un modèle des concepts qui peuvent être utilisés pour concevoir et décrire un système. Les modèles décrivant le système sont des instances du méta-modèle (entités du modèle sont des instances des entités du méta-modèle). Les méta-modèles permettent de guider le processus de développement. Dans le cadre des SMA, un méta-modèle spécifie les types d'entités que le développeur doit prendre en considération. Il permet aussi d'établir les relations et les rapports entre ces entités.

Il existe plusieurs méthodologies de conception de SMA considérant comme point de départ une liste de buts, fonctionnalités, comportements, entités ou structures liés aux objectifs de l'application finale, citant quelques unes parmi les méthodologies orientées agents les plus représentatives : AAIL, CASSIOPEE, DESIRE, AALAADIN, GAIA, ROADMAP, MESSAGE / UML, INGENIAS, ADELFE, PASSI, PROMETHEUS, TROPOS. On étudié le modèle par la méthodologie AALAADIN (AGR) [47] qui va être définis par la suite, cette méthodologie qui est le premier travail qui propose un méta-modèle pour l'organisationnel en multi-agents, et qui nous a choisi pour notre étude car elle convient par sa simplicité a notre modèle.

5.1.2 La Méthodologie Aalaadin

La méthodologie Aalaadin [48] s'intéresse à la conception et au développement des systèmes multi-agents dans une perspective organisationnelle. Aalaadin a introduit le premier méta-modèle générique de systèmes multi-agents basé sur les concepts d'agent, de groupe et de rôle (AGR). Le modèle proposé décompose l'analyse des structures collectives en deux niveaux:

- a. Le niveau descriptif: correspond aux concepts centraux d'agent, de groupe et de rôle.
- b. Le niveau méthodologique: définit l'ensemble des rôles possibles, spécifie les interactions et décrit les structures abstraites de groupes et d'organisations.

5.1.2.1 Concepts centraux

Aalaadin est basé sur trois concepts: agent, groupe et rôle (AGR).

a. Agent: un agent est une entité autonome communicante qui joue des rôles au sein de différents groupes. Aucune contrainte n'est posée sur l'architecture interne de l'agent et ne suppose pas de formalisme ou modèle pour décrire le comportement de l'agent.

b. Groupe: un groupe est un ensemble d'agents ayant des points communs et les interactions sont toujours locales à un groupe. Un système multi-agent est vu comme un ensemble de groupes d'agents. Un groupe peut être fondé par n'importe quel agent.

c. Rôle: le rôle est une représentation abstraite d'une fonction, d'un service ou d'une identification d'un agent au sein d'un groupe particulier. Chaque agent peut avoir plusieurs rôles distincts au sein de plusieurs groupes. Un même rôle peut être tenu par plusieurs agents, les rôles sont locaux aux groupes. La tenue d'un rôle dans un groupe doit être demandée par l'agent et elle n'est pas forcément accordée. Un rôle important et particulier dans un groupe est celui du gestionnaire du groupe (initialement le fondateur), ce rôle gère les demandes d'admission dans le groupe et la tenue de rôle ainsi que leurs éventuelles réservations.

5.1.2.2 Niveau méthodologique

De nombreux concepts sont ajoutés au niveau conceptuel à savoir les structures de groupe et les structures organisationnelles. A la différence des concepts de base, ces nouveaux concepts sont des représentations abstraites des groupes et de l'organisation. Le but est de pouvoir spécifier une structure organisationnelle à partir de la quelle un système multi-agents puisse être développé et exprimé dans les concepts de base.

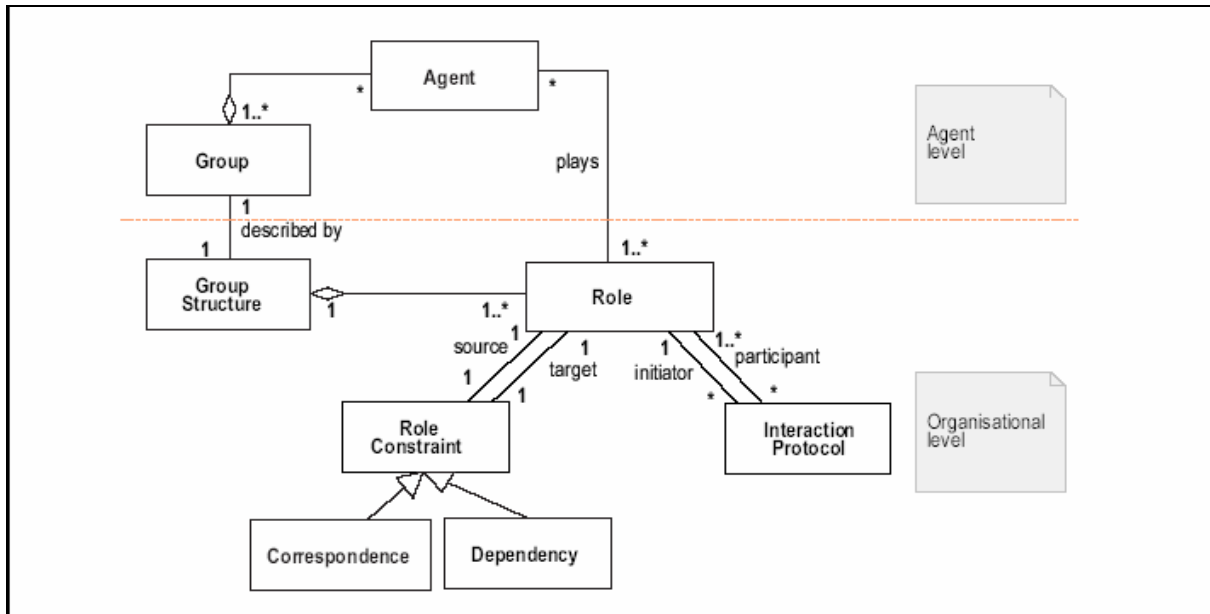
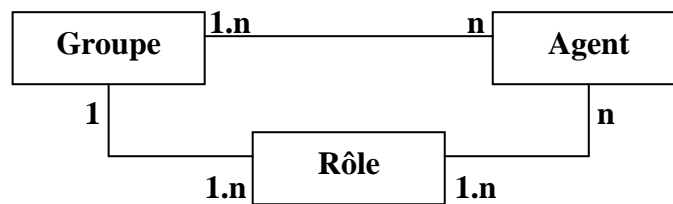


Figure 5.1 : Le méta-modèle de la méthodologie Aalaadin.

Se méta modèle peut se résumer en :



Avec 1 et n sont les cardinalités, ou n signifie plusieurs.

La modélisation graphique d'une organisation AGR peut passer par trois types de diagrammes différents :

- Le diagramme de classe.
- Les diagrammes d'activités.
- Les diagrammes de comportements.

Remarque: Aalaadin n'exploite que l'aspect organisationnel d'un système multi-agents. Il n'y a aucun moyen de spécifier le comportement interne des agents.

5.1.2.3 Processus de développement

Le processus de développement de méthodologie Aalaadin peut se résumer en:

1. L'analyse qui permet d'identifier les fonctions du système et les dépendances au sein de communautés identifiées, il convient de définir quels sont les mécanismes de coordination et d'interaction entre les entités d'analyse.
2. La conception est la phase la mieux décrite. Elle contient l'identification des groupes, des agents et des rôles dans les diagrammes cités ci-dessus.
3. La réalisation commence par le choix de l'architecture de l'agent. Puis, la gestion des entités du domaine permet d'implanter le système à partir d'organisations concrètes.

5.2 Conception du Système Multi Agents

Notre système est constitué essentiellement d'un groupe qui se compose de quatre (04) agents qui se définissent par leurs rôles, buts et comportements.

Dans cette partie on a essayé de concevoir les diagrammes appropriés à cette méthodologie de conception du système multi agents cité ci-dessus.

5.2.1 Diagramme de classe

Tout d'abord il faut savoir que notre diagramme de classe se base sur le Méta-modèle de la méthodologie **Aalaadin** basé sur l'**AGR** qui se représente dans la Figure suivante. Dans notre application nous avons un seul diagramme de classe pour notre étude.

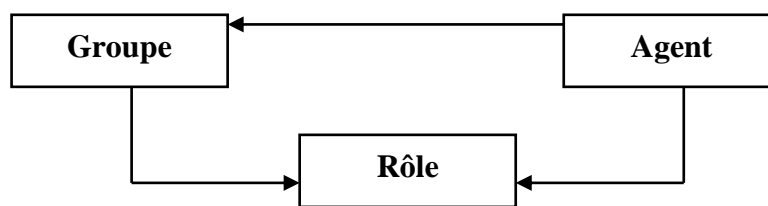


Figure 5.2 : modèle AGR

Les classes qui constituent notre diagramme de classe sont :

- La classe de Groupe :(Groupe analyse hydrodynamique).
- Les classes d'Agent :(Amvt_global, Amvt_axi, Amvt_lat, Amvt_rot).
- Les classes de Rôle (les rôles) :(Calculer de Pression, Calculer de charg_hydro).

Notre Système Multi-Agents est composé d'un seul groupe nommé **Groupe analyse hydrodynamique** qui se compose lui-même de quatre (04) agents nommé successivement par :

- **Amvt_global** : qui définit le mouvement hydrodynamique global du piston.
- **Amvt_axi** : qui définit le mouvement hydrodynamique axial du piston.
- **Amvt_lat** : qui définit le mouvement hydrodynamique latéral du piston.
- **Amvt_rot** : qui définit le mouvement hydrodynamique de rotation du piston

Chacun de ces agents a ses propres tâches à réaliser et en plus ont des rôles communs entre eux.

Concernant l'Agent Amvt_global on peut dire qu'il joue le rôle d'un synthétiseur c-à-d qu'il a ses propres tâches (leurs méthodes à calculer) et ses buts qui sont :

- Calculer la pression global \bar{P} (total) pour cela il doit communiquer avec les autres agents à travers des messages dans le but qu'après ces communications chaque agent de ces trois (03) calcule leur pression partielle ($\bar{P}_u, \bar{P}_{1c}, \bar{P}_{2c}$) en activant son rôle et suit leur comportement d'obtention du résultat de pression pour donner enfin le résultat global.

- Calculer les charges hydrodynamique globales **F** et **M** en décomposant le calcul sur les 3 Agents (mouvement) pour obtenir les charges partielles (**Fu, F1c, F2c** et **Mu, M1c, M2c**) de chaque mouvement et cela se fait en suivant les mêmes étapes du but précédent.

Et après cet agent **Amvt_global** complète leur calcul interne pour réaliser leur but final qui est l'obtention du résultat des excentricités **eh** et **eb**.

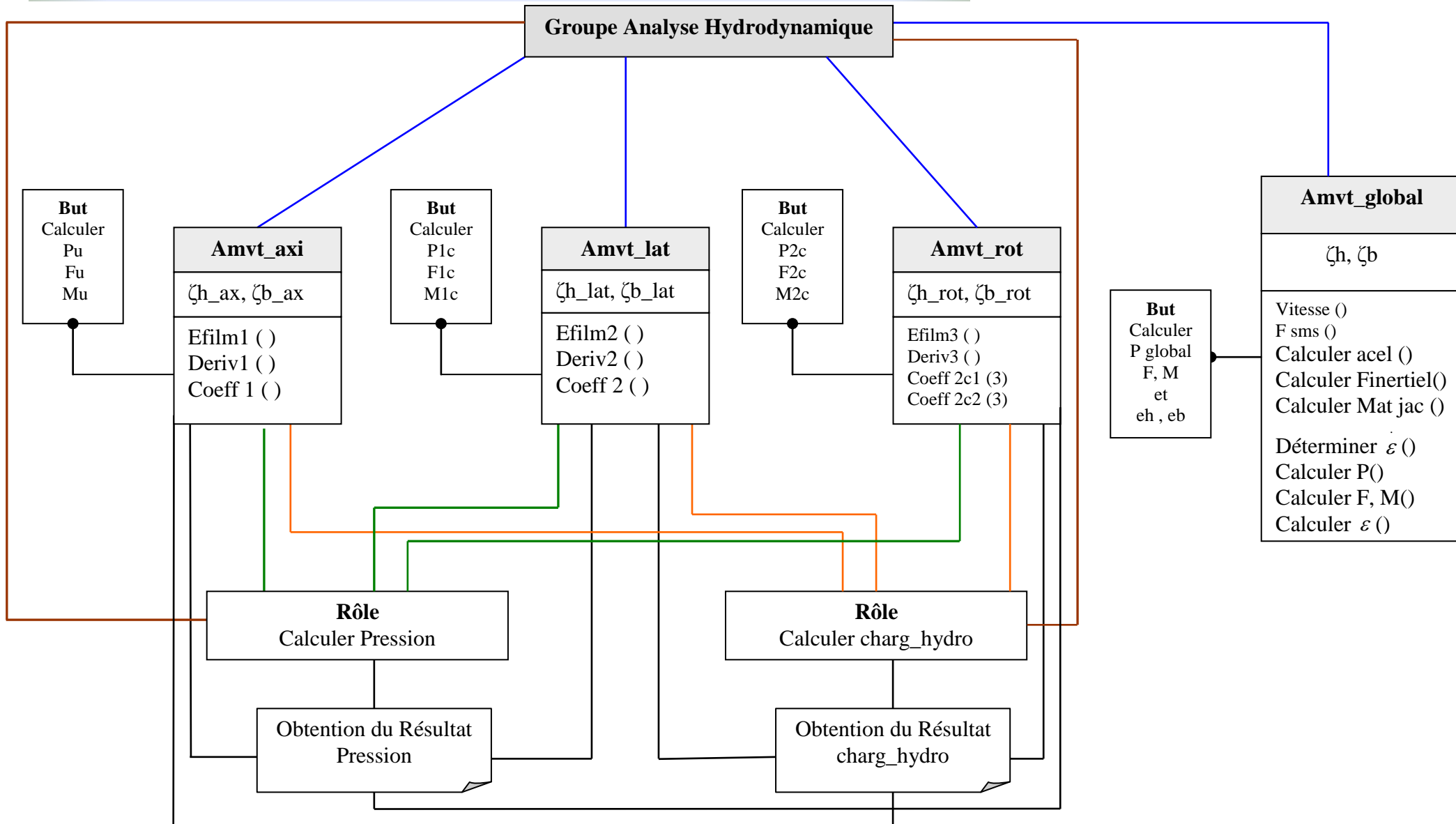
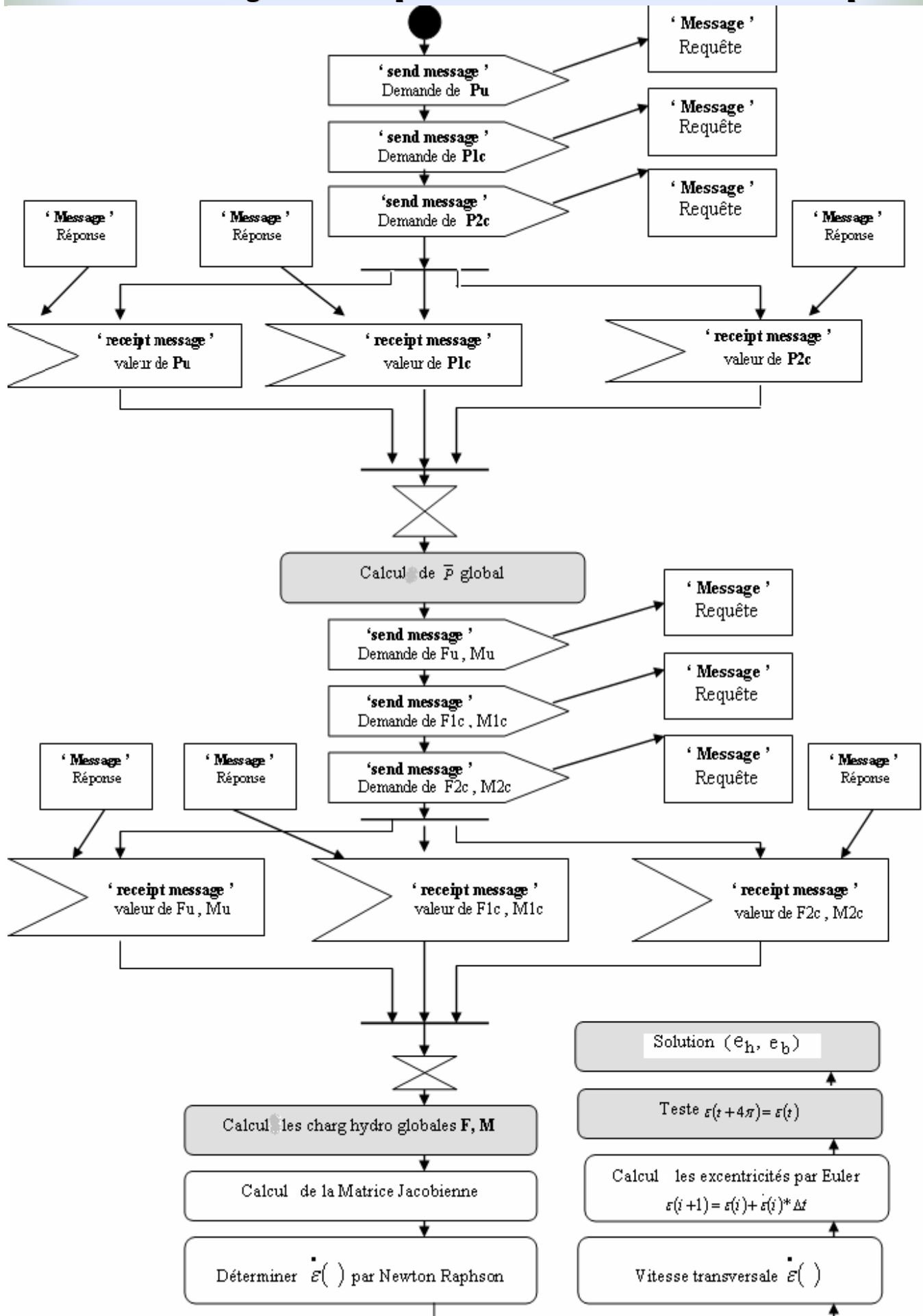


Figure 5.3 : Diagramme de classe

5.2.2 Diagramme d'activités de l'Agent Mouvement_Global

L'agent Amvt_global envoie trois (03) messages au trois agents (Amvt_axi, Amvt_lat, Amvt_rot) en donnant la valeur de vitesse v_{bar} et des excentricités initiales (ε_h , ε_b) et demandant les résultats des pressions partielles (**Pu, P1c, P2c**) et attend la réponse de la part de ces trois (03) agents, après que ces trois agents obtenir ses buts ils envoient leurs messages de réponse c à d que l'agent Amvt_global reçoit les messages de réponse, en suite il réalise leur premier but en calculant la pression global **Pu**, l'agent renvoie trois (03) messages au trois agents (Amvt_axi, Amvt_lat, Amvt_rot) en demandant aussi les résultats des charges hydrodynamiques partielles (**Fu, Mu** et **F1c, M1c** et **F2c, M2c**) et attend la réponse de la part de ces trois (03) agents, après que ces trois agents obtenir ses buts ils renvoient leurs messages de réponse c à d que l'agent Amvt_global reçoit les trois (03) messages de réponse, pour réaliser leur second but en calculant les charges hydrodynamiques global **F** et **M**, en suite de ça l'agent complète leurs tâches (leurs calculs) jusqu'à la réalisation du dernier but pour l'obtention du résultats des excentricités **eh** et **eb**.

Figure 5.4 : Diagramme d'activités de l'Agent Mouvement_Global



5.2.3 Diagramme d'activités de l'Agent Mouvement_Axial

L'agent Amvt_axi reçoit un message de l'agent Amvt_global en donnant les valeurs initiales des excentricités (ε_h , ε_b) et la résultat de calcul de Vbar (sa valeur), et demandant la valeur de Pu. Lorsque le message est reçu, alors l'agent Amvt_axi complète ses calculs (tâches) : donc il appelle ces propres méthodes pour calculer l'épaisseur du film d'huile Efilm1(), calculer les dérivées de l'épaisseur deriv1() et ensuite calculer les coefficients de Reynolds coeff1(), et après l'agent Amvt_axi active leur premier rôle pour débiter le calcul de la pression **Pu** et suit le comportement d'obtention du résultat de pression **Pu** jusqu'à la fin du calcul c à d la fin du rôle, et envoie un message réponse contenant la résultat de **Pu** à l'agent Amvt_global, ensuite il reçoit un 2^{ième} message de l'agent Amvt_global qui demande de calculer **Fu**, **Mu**, pour cela il active du second rôle pour débiter le calcul des charges hydrodynamiques **Fu**, **Mu** et suit le comportement d'obtention du résultat des charges hydrodynamiques **Fu**, **Mu** jusqu'à la fin du calcul c à d la fin du rôle, et après il envoie un 2^{ième} message réponse contenant la résultat de **Fu**, **Mu** à l'agent Amvt_global et termine la procédure.

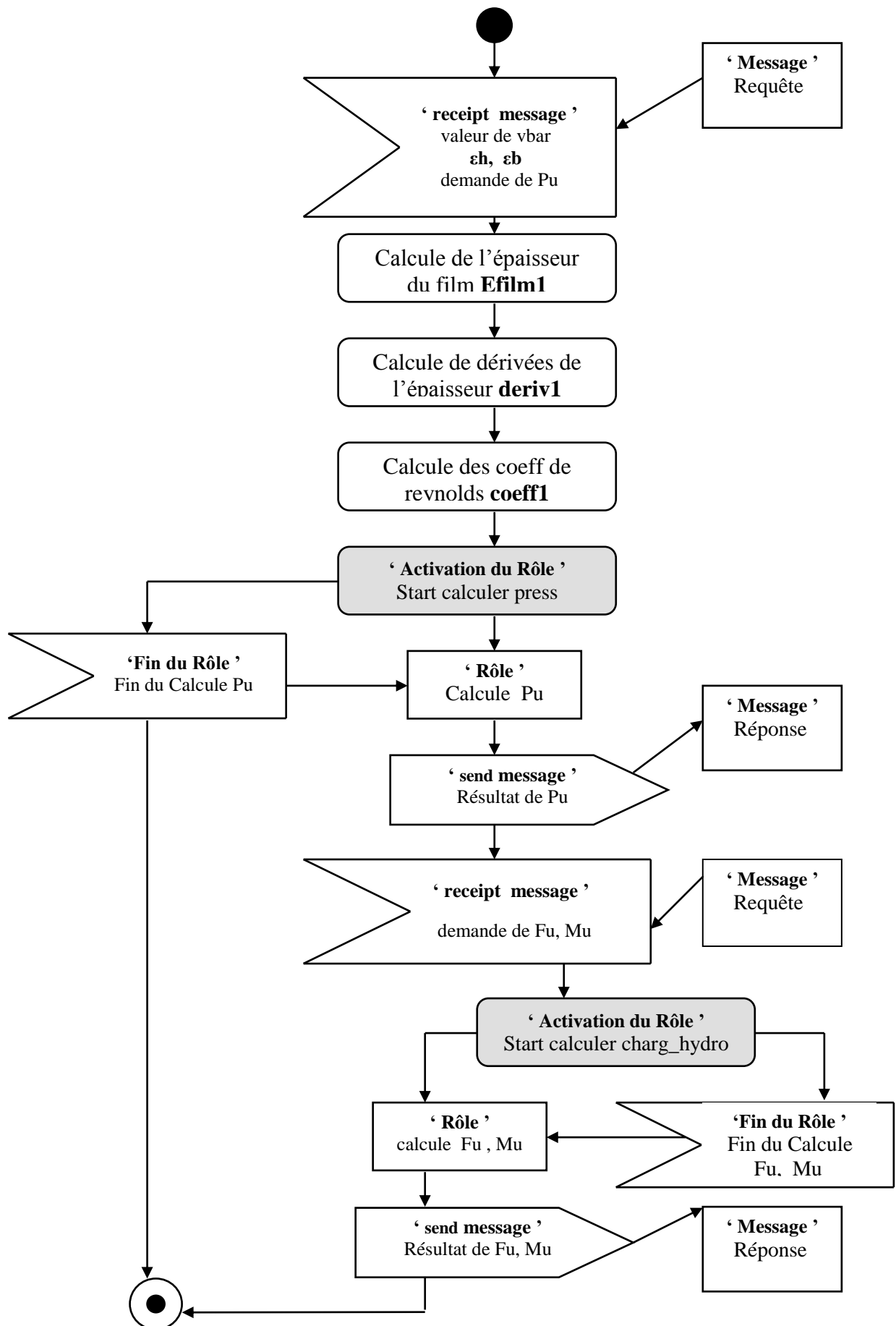


Figure 5.5 : Diagramme d'activités de l'Agent Mouvement_Axial

5.2.4 Diagramme d'activités de l'Agent Mouvement_Latéral

L'agent Amvt_lat reçoit un message de l'agent Amvt_global en donnant les valeurs initiales des excentricités (ε_h , ε_b) la résultat de calcul de Vbar (sa valeur), et demandant la valeur de P1c. Lorsque le message est reçu, alors l'agent Amvt_lat complète ses calculs (tâches) : donc il appelle ces propres méthodes pour calculer l'épaisseur du film d'huile Efilm2(), calculer les dérivées de l'épaisseur deriv2() et ensuite calculer les coefficients de Reynolds coeff2(), et après l'agent Amvt_lat active leur premier rôle pour débiter le calcul de la pression **P1c** et suit le comportement d'obtention du résultat de pression **P1c** jusqu'à la fin du calcul c à d la fin du rôle, et envoie un message réponse contenant la résultat de **P1c** à l'agent Amvt_global, ensuite il reçoit un 2^{ième} message de l'agent Amvt_global qui demande de calculer **Fu**, **Mu**, pour cela il active du second rôle pour débiter le calcul des charges hydrodynamiques **F1c**, **M1c** et suit le comportement d'obtention du résultat des charges hydrodynamiques **F1c**, **M1c** jusqu'à la fin du calcul c à d la fin du rôle, et après il envoie un 2^{ième} message réponse contenant la résultat de **F1c**, **M1c** à l'agent Amvt_global et termine la procédure.

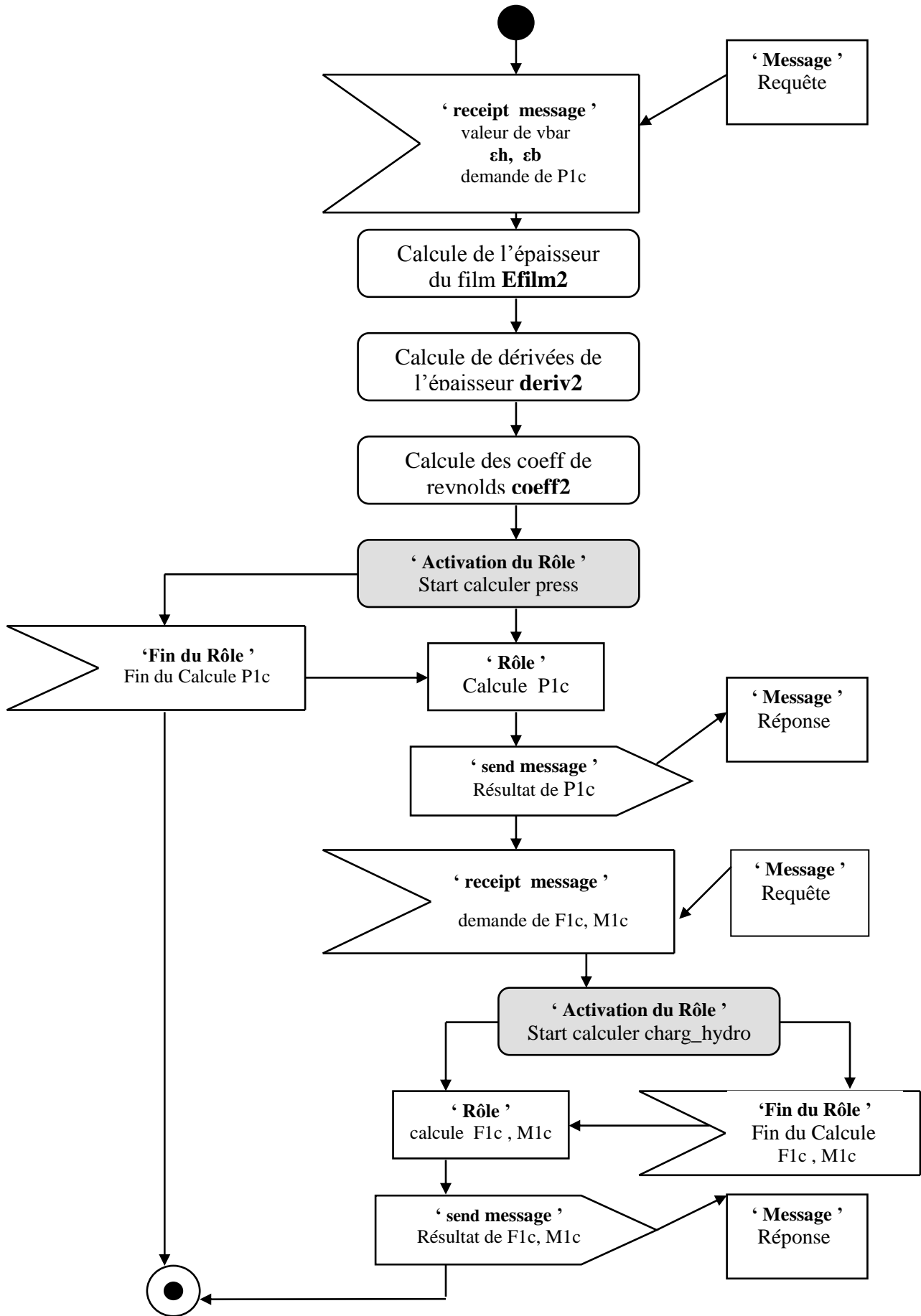


Figure 5.6 : Diagramme d'activités de l'Agent Mouvement_Latéral

5.2.5 Diagramme d'activités de l'Agent Mouvement_rotation

L'agent Amvt_rot reçoit un 1^{ier} message de l'agent Amvt_global en donnant les valeurs initiales des excentricités ε_{h_rot} et ε_{b_rot} , la résultat de calcul de Vbar (sa valeur), et demandant la valeur de P2c, et un 2^{ieme} message demandant les valeurs des charges hydrodynamiques F2c, M2c. lorsque les deux (02) messages sont reçus, alors à ce moment l'agent à deux choix selon le teste de la valeur d'excentricité ε_{h_rot} , il doit suivre un des deux choix selon cette valeur sil est supérieur ou inférieur de 0 donc ou bien il choisit (1) : l'agent Amvt_rot complète ses calculs (tâches) : calcule de l'épaisseur du film d'huile Efilm31(), calcule des dérivées de l'épaisseur deriv31() et ensuite calcule les coefficients de Reynolds coeff2c2(), et après l'agent Amvt_rot active leur premier rôle pour débiter le calcul de la pression $\bar{P}2c2$ de la zone 1 et suit le comportement d'obtention du résultat de pression $\bar{P}2c2$ jusqu'à la fin du calcul c à d la fin du rôle, et après l'activation du second rôle pour débiter le calcul des charges hydrodynamiques Fdim_0, M1dim_0 de la zone 1 et suit le comportement d'obtention du résultat des charges hydrodynamiques Fdim_0, M1dim_0 jusqu'à la fin du calcul c à d la fin du rôle et ensuite complète ses calculs (tâches) : calcule de l'épaisseur du film d'huile Efilm32(), calcule des dérivées de l'épaisseur deriv32() et ensuite calcule les coefficients de Reynolds coeff2c1(), et après l'agent Amvt_rot active leur premier rôle pour débiter le calcul de la pression $\bar{P}2c1$ de la zone 2 et suit le comportement d'obtention du résultat de pression $\bar{P}2c1$ jusqu'à la fin du calcul c à d la fin du rôle, et après l'activation du second rôle pour débiter le calcul des charges hydrodynamiques Fdim_pi, M1dim_pi et suit le comportement d'obtention du résultat des charges hydrodynamiques Fdim_pi, M1dim_pi jusqu'à la fin du calcul c à d la fin du rôle, enfin il fait la somme des deux résultats de pression (des deux zones) pour l'obtention du premier but $\bar{P}2c$ et envoie un message réponse contenant la résultat de **P2c** à l'agent Amvt_global, et la somme des deux autres résultats des charges hydrodynamiques pour l'obtention du second but F2c et M2c et envoie un message réponse contenant la résultat de **F2c, M2c** à l'agent Amvt_global, et termine la procédure.

sinon il choisit le (2) c à d : l'agent Amvt_rot complète ses calculs (tâches) : calcule de l'épaisseur du film d'huile Efilm31(), calcule des dérivées de l'épaisseur deriv31() et ensuite calcule les coefficients de Reynolds coeff2c1(), et après l'agent Amvt_rot active leur premier rôle pour débiter le calcul de la pression $\bar{P}2c1$ de la zone 2 et suit le comportement d'obtention du résultat de pression $\bar{P}2c1$ jusqu'à la fin du calcul c à d la fin du rôle, et après l'activation du second rôle pour débiter le calcul des charges hydrodynamiques Fdim_pi, M1dim_pi de la zone 2 et suit le comportement d'obtention du résultat des charges hydrodynamiques Fdim_pi, M1dim_pi jusqu'à la fin du calcul c à d la fin du rôle et ensuite complète ses calculs (tâches) : calcule de l'épaisseur du film d'huile Efilm32(), calcule des dérivées de l'épaisseur deriv32() et ensuite calcule les coefficients de Reynolds coeff2c2(), et après l'agent Amvt_rot

active leur premier rôle pour débiter le calcul de la pression \overline{P}_{2c2} de la zone 2 et suit le comportement d'obtention du résultat de pression \overline{P}_{2c2} jusqu'à la fin du calcul c à d la fin du rôle, et après l'activation du second rôle pour débiter le calcul des charges hydrodynamiques F_{dim_0} , M_{dim_0} et suit le comportement d'obtention du résultat des charges hydrodynamiques F_{dim_0} , M_{dim_0} jusqu'à la fin du calcul c à d la fin du rôle, enfin il fait la somme des deux résultats de pression (des deux zones) pour l'obtention du premier but \overline{P}_{2c} et envoie un message réponse contenant la résultat de **P2c** à l'agent *Amvt_global*, et la somme des deux autres résultats des charges hydrodynamiques pour l'obtention du second but F_{2c} et M_{2c} et envoie un message réponse contenant la résultat de **F2c, M2c** à l'agent *Amvt_global*, et terminer la procédure.

Remarque : dans ce cas (mouvement de rotation) la pression et les charges hydrodynamiques sont décomposées en deux parties (zone 1 et zone 2 selon l'angle θ) pour les rassembler à la fin de la procédure.

$$\overline{P}_{2c} = \overline{P}_{2c1} + \overline{P}_{2c2} ;$$

$$F_{2c} = F_{dim_0} + F_{dim_pi} ;$$

$$\text{Et } M_{2c} = M_{dim_0} + M_{dim_pi}$$

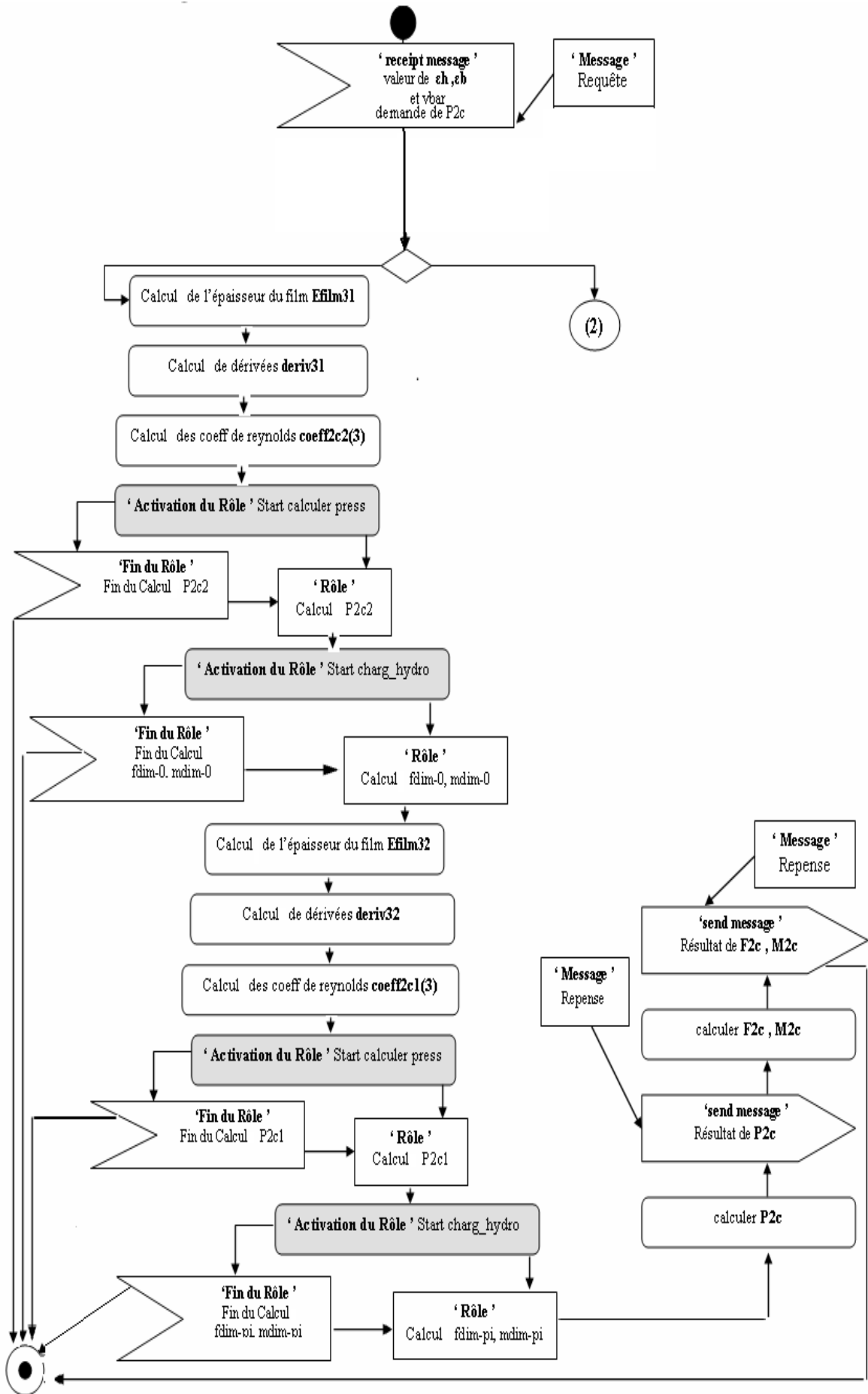
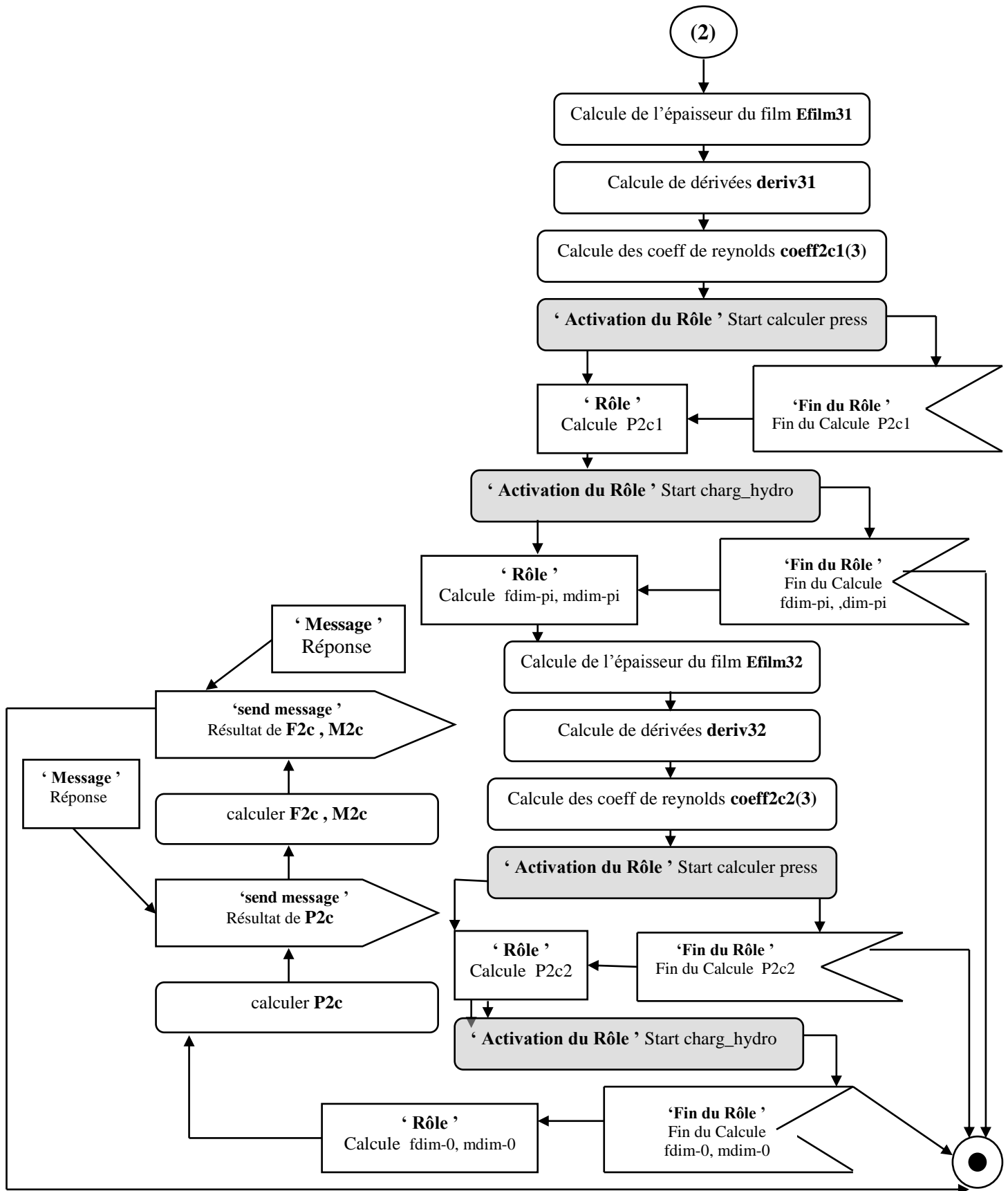


Figure 5.7 : Diagramme d'activités de l'Agent Mouvement_Rotation



Suite du Diagramme d'activités de l'Agent Mouvement_Rotation

5.2.6 Diagramme d'activité de rôle

5.2.6.1 Diagramme d'activité du rôle calcul de pression

Ce diagramme résume les actions du « rôle calcul de pression » pour obtenir **Pu**, **P1c** et **P2c** pour les trois Agents successivement **Agent mvt_axial**, **Agent mvt_latéral** et **Agent mvt_rotation**.

Une fois l'agent active le rôle de calcul de pression, il vient de lancer sa 1^{ière} action qui est l'initialisation de la pression $P(i, j)$, ensuite il calcule la pression dans une 2^{ième} action suivant un traitement (P_u , P_{1c} , P_{2c} selon l'agent qui active ce rôle $Amvt_axi$, $Amvt_lat$ et $Amvt_rot$ successivement) et après qu'il obtient la résultat il revoie en fin la résultat à son agent responsable.

Ce rôle est commun entre les trois (03) agents, alors à n'importe quelle moment donné un de ces trois (03) active ce rôle peut activé ce rôle.

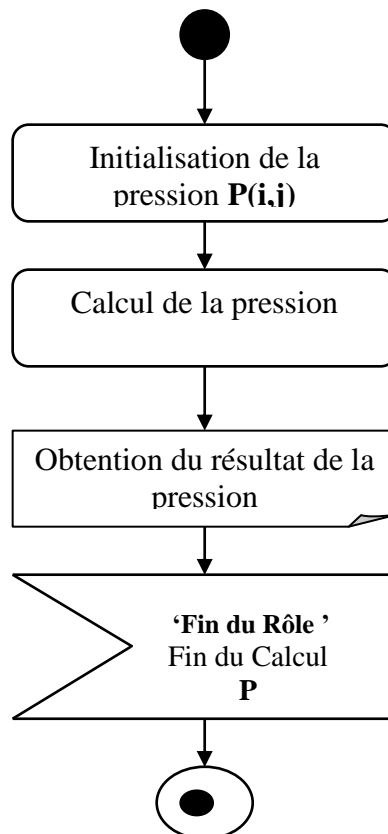


Figure 5.8 : Diagramme d'activité du rôle calcul de pression

5.2.6.2 Diagramme d'activité du rôle calcul des charges hydrodynamiques :

Ce diagramme résume les actions du « rôle calcul de charge hydrodynamique » pour obtenir F_u , M_u et F_{1c} , M_{1c} et F_{2c} , M_{2c} pour les trois Agents successivement **Agent mvt_axial**, **Agent mvt_latéral** et **Agent mvt_rotation**.

Une fois l'agent active le rôle de calcul charges hydrodynamiques, il vient de lancer son action qui est l'initialisation pour calculer la les charges hydrodynamiques F et M suivant un traitement (F_u , F_{1c} , F_{2c} et M_u , M_{1c} , M_{2c}) selon l'agent qui active ce rôle $Amvt_axi$, $Amvt_lat$ et $Amvt_rot$ successivement) et après qu'il obtient la résultat il revoie en fin la résultat à son agent responsable.

Ce rôle est commun entre les trois (03) agents, alors à n'importe quelle moment donné un de ces trois (03) active ce rôle peut activé ce rôle.

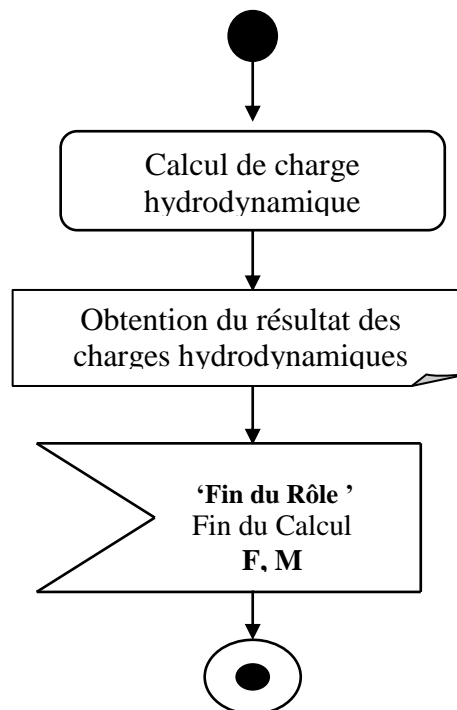


Figure 5.9 : Diagramme d'activité du rôle calcul des charges hydrodynamiques

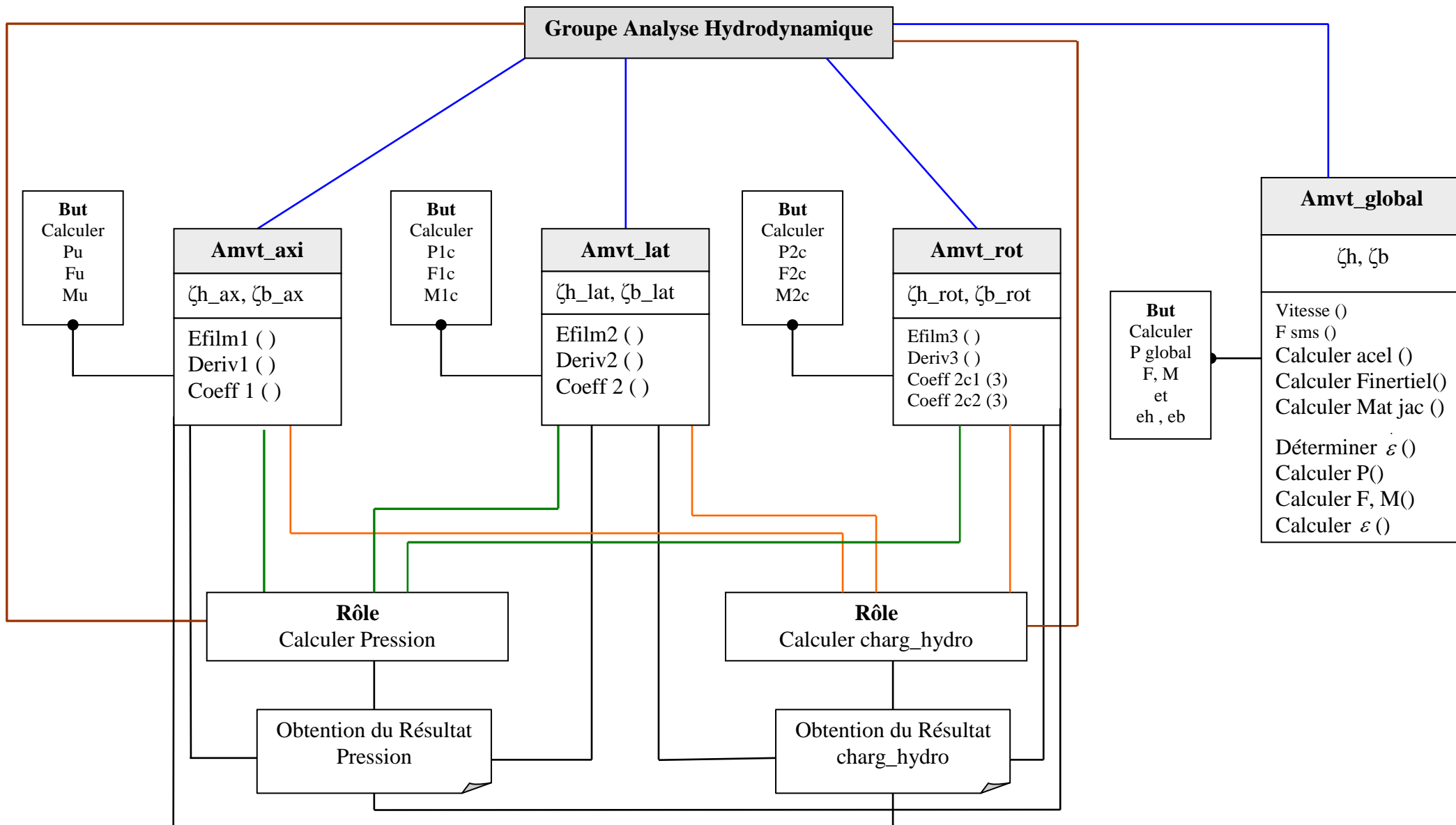


Figure 5.3 : Diagramme de classe

Chapitre 6

IMPLEMENTATION DE METHODE MECANIQUE ET SMA : RESULTATS ET DISCUSSION

6.1 Introduction

Il est naturellement plus commode d'utiliser un langage de programmation multi-agents pour la simulation d'un SMA. Cependant, comme nous l'avons déjà signalé, il n'y a pas, jusqu'à maintenant, de langage qui a reçu un consensus de la part de la communauté des SMA. Il faut alors choisir un des langages de programmation existants ayant des potentialités de prendre en charge le maximum des aspects des agents.

Le langage à utiliser pour la simulation des SMA doit posséder les caractéristiques suivantes :

- être orienté objet parce que ceux-ci sont, actuellement, les plus puissants sémantiquement.
- supporter la communication par messages orientés objet pour simplifier l'implémentation de la communication entre agents.
- être multi-threading pour supporter l'autonomie des agents.

Différents paradigmes de communication existent dans les technologies actuelles : communication classique de données, RPC (Remote Procedure Call), RMI (Remote Method Invocation) de JAVA, CORBA, Se sont les paradigmes RMI, principalement, qui supportent la communication dans le modèle orienté objet.

Pour ces raisons, nous pensons que JAVA est un bon candidat pour l'implémentation des agents. Cependant, un langage supportant plusieurs paradigmes, incluant les aspects de JAVA, serait meilleur.

Il est portable car le byte code JAVA peut être exécuté sur différentes machines virtuelles. En plus est un très fiable, car il permet la détection d'erreurs au moment de la compilation et de l'exécution. JAVA permet aussi le développement d'application dans le contexte d'environnements distribués et hétérogènes, est un langage de programmation adapté à Internet.

6.2 Simulation des agents

Un agent est décrit comme un objet thread. Un objet thread est un objet dont les méthodes s'exécutent sur un thread indépendant. Ce qui permet à un objet de disposer des informations de contrôle indépendamment des autres thread du système.

Dans un modèle de communication, les messages sont des méthodes et l'envoi d'un message se fait par invocation d'une méthode. La communication entre agents se fait par les actes de langage, auxquels nous associons des méthodes des objets, représentant les agents.

6.3 Programme principal

Notre système est constitué d'un :

- 1) **Agent mouvement Global** : ce programme appelle au début la méthode de calcul initial `methodcalcul()`, ensuite initialise les données d'entrées comme `eh` et `eb`, et calcule la vitesse en appelant la méthode `vitesse()`, lance leur thread `threadReception()`, crée les trois (03) messages à envoyer au agents (`AgentmvtAxial`, `AgentmvtLatéral`, `AgentmvtRotation`) en demandant les pressions partielles, ensuite fait appel aux méthodes d'envoies de ces messages `sendmsgdemande (String nomfichier, String ad)` avec leurs paramètres spécifiés, à ce moment l'agent va attendre la réception des trois(03) messages résultats réalisé par le thread de réception, ensuite appelle la deuxième méthode de calcul `methodcalcul1()` qui permet d'obtenir la pression totale, et juste après crée trois (03) autres messages pour renvoyer au agents (`AgentmvtAxial`, `AgentmvtLatéral`, `AgentmvtRotation`) en demandant les charges hydrodynamiques partielles et en appellent aussi les méthodes d'envoies de ces messages `sendmsgdemande (String nomfichier, String ad)` avec leurs paramètres spécifiés, en attendant la réception de ces trois(03) messages résultats réalisé par le même thread de réception cité avant, ensuite fait appel à la troisième méthode de calcul `methodcalcul2()` qui permet d'obtenir les charges hydrodynamiques totales et les excentricités `eh` et `eb`.

Les deux parties suivantes du code source permettent de structurer un message selon la forme d'un message Fipa-acl mais un peu simplifiée basant sur le besoin de notre modèle.

Création du message type 1

```
fe21= new PrintStream (new FileOutputStream ("c:/Execution/MSGdemande1.dat"));

rechercheNomService ("calcul_pu");
rechercheAdrNom ("Agent_Global");
AgentmvtGlobal.fe21.println ("Sender : "+ "Agent_Global"+" "+adr);
rechercheAdrNom (Receiver);
AgentmvtGlobal.fe21.println ("Receiver : "+ Receiver+" "+adr);
AgentmvtGlobal.fe21.println ("Content : ");
AgentmvtGlobal.fe21.println ("eh = "+ VariableCommun.epsh);
AgentmvtGlobal.fe21.println ("eb = "+ VariableCommun.epsb);
AgentmvtGlobal.fe21.println ("vitesse = "+ VariableCommun.vbar);
AgentmvtGlobal.fe21.println ("service : "+ "calcul_pu");
AgentmvtGlobal.fe21.close();
```

Création du message type 2

```

fe24= new PrintStream (new FileOutputStream ("c:/Execution/MSGdemande4.dat"));
rechercheNomService ("calcul_FuMu");
rechercheAdrNom ("Agent_Global");
fe24.println ("Sender : "+ "Agent_Global"+" "+adr);
rechercheAdrNom (Receiver);
fe24.println ("Receiver : "+ Receiver+" "+adr);
fe24.println("Content : ");
fe24.println("service : "+"calcul_FuMu");
fe24.close();

```

Méthode d'envoi du message

```

public void sendmsgdemande (String nomfichier, String ad) throws Exception,UnknownHostException, IOException
{
    File f = new File(nomfichier);
    if(f.exists())
    {
        // System.out.println("\nenvoie le MSG demande ");
        Socket soc = new Socket(ad, 3000);
        OutputStream fluxsortie = soc.getOutputStream();
        InputStream in = new BufferedInputStream(new FileInputStream(f));
        ByteArrayOutputStream tableaubytes = new ByteArrayOutputStream();
        BufferedOutputStream tampon = new BufferedOutputStream(tableaubytes);
        int lu = in.read();
        int[] aecrire = new int[4097];
        int compteur = 0;
        //Tant qu'on est pas à la fin fichier
        while(lu > -1)
        {
            aecrire[compteur] = lu;//On lit les données du fichier
            lu = in.read();
            compteur++;
        }
        //Quand on a rempli le tableau, on envoie un paquet de 4096 octets
        if(compteur == 4096)
        {
            compteur=0;
            for(int x=0;x<4096;x++) //On remplit le tampon
            tampon.write(aecrire[x]);
            fluxsortie.write(tableaubytes.toByteArray());//Et on l'envoie
            tableaubytes.reset();
        }
    }
    //while
    //On envoie le dernier paquet, qui ne fait pas forcément 4096
}

```

```

for(int x=0;x<compteur;x++) //On remplit le tampon
tampon.write(aecrire[x]);
tampon.flush();//Et on l'envoie
fluxsortie.write(tableaubytes.toByteArray());
fluxsortie.flush();
in.close();
tampon.close();
soc.close();
} //if exist
else
{
    System.out.println("Le fichier "+f+" est introuvable");
}
f.deleteOnExit();
VariableCommun.mutex_0.signal();
}
    
```

Fonctionnement sur un exemple :

1- envoi msg1 du AgentmvtGlobal au AgentmvtAxial en donnant la valeur de vbar, epsh et epsb et demandant la résultat de pu.

```

try {
    sendmsgdemande ("c:/Execution/MSGdemande1.dat","192.168.1.2");
} catch (IOException e) {}
    
```

2- envoi msg4 du AgentmvtGlobal au AgentmvtAxial en demandant la résultat de Fu et Mu.

```

try {
    sendmsgdemande ("c:/Execution/MSGdemande4.dat","192.168.1.2");
} catch (IOException e) {}
    
```

Thread de reception

```

package analysehydrodynamicpiston;

import java.math.*;
import java.io.*;
import java.net.UnknownHostException;
import java.lang.*; // for Threads
import java.net.*; // sockets
import java.util.*;

public class thread_reception extends Thread{

    Socket amaxiSocket =null;
    Socket Soc =null;
    ServerSocket ssoc=null;
    int port ; // port
    int cpt=0, cp=0, cpt1=0, cpt2=0, cpt3=0 ;
    boolean tt=true;
    thread_reception (int port) {
        this.port=port;
    }
}
    
```

```

public void run () {
    try {
        ssoc = new ServerSocket(port);
        while (tt==true){
            Soc = ssoc.accept();
            BufferedInputStream inBuffer1=new BufferedInputStream(Soc.getInputStream());
            OutputStream sortie =new FileOutputStream ("c:/Execution/indic.dat");
            BufferedOutputStream sorteBuffer=new BufferedOutputStream(sortie);
            int lir =inBuffer1.read(); // prise le String de client
            sorteBuffer.write(lir);
            sorteBuffer.flush();
            sorteBuffer.close();
            sortie.flush();
            sortie.close();
            BufferedReader entrer = new BufferedReader(new FileReader("c:/Execution/indic.dat"));
            String st = new String();
            StringTokenizer tokenn;
            String lign=new String();
            int indicate;
            lign = entrer.readLine();
            tokenn = new StringTokenizer(lign, " ");
            st = tokenn.nextToken();
            indicate= Integer.parseInt(st);
            entrer.close();
            if (indicate==1) //reçu de l'agent 1
            // if (Soc.getInetAddress().getHostAddress()=="192.168.1.2"){//reçu de l'agent 1
                if (cpt1==0){

                    AgentmvtGlobal.receivevmsg ( "c:/Execution/MSGpu.dat", inBuffer1);

                    try {
                        AgentmvtGlobal.liremsgreçu1 ("c:/Execution/MSGpu.dat");

                    } catch (Exception e) {}
                    cpt++;
                    cpt1++;
                }
                else if (cpt1==1) {

                    cpt1=0;
                    AgentmvtGlobal.receivevmsg ( "c:/Execution/MSGchargaxi.dat", inBuffer1);

                    try {
                        AgentmvtGlobal.liremsgreçu4 ("c:/Execution/MSGchargaxi.dat");

                    } catch (Exception e) {}
                    cp++;

                }
            }
            if (indicate==2){//reçu de l'agent 2
                if (cpt2==0){
                    AgentmvtGlobal.receivevmsg ( "c:/Execution/MSGp1c.dat", inBuffer1);

                    try {
                        AgentmvtGlobal.liremsgreçu2 ("c:/Execution/MSGp1c.dat");

                    } catch (Exception e) {}
                    cpt++;
                    cpt2++;
                }
                else if (cpt2==1) {
                    cpt2=0;
                    AgentmvtGlobal.receivevmsg ( "c:/Execution/MSGcharglat.dat", inBuffer1);

                    try {
                        AgentmvtGlobal.liremsgreçu5 ("c:/Execution/MSGcharglat.dat");
                    }
                }
            }
        }
    }
}

```



```

        } catch (Exception e) {}
        cp++;
    }
}
if (indicate==3){//reçu de l'agent 3
    if (cpt3==0){
        AgentmvtGlobal.receivemsg ( "c:/Execution/MSGp2c.dat", inBuffer1);

        try {
            AgentmvtGlobal.liremsgreçu3 ("c:/Execution/MSGp2c.dat");

            }catch (Exception e) {}
            cpt++;
            cpt3++;
        }
    else if (cpt3==1) {
        cpt3=0;
        AgentmvtGlobal.receivemsg ( "c:/Execution/MSGchargrot.dat", inBuffer1);

        try {
            AgentmvtGlobal.liremsgreçu6 ("c:/Execution/MSGchargrot.dat");

            } catch (Exception e) {}
            cp++;
        }
    }
}
try {
    if (cpt==3) {VariableCommun.mutex_10.signal(); cpt=0;} // tester
    } catch (Exception e) {}
}
try {
    if (cp==3) {VariableCommun.mutex_11.signal(); cp=0;} // tester
    } catch (Exception e) {}
}
} catch (Exception e) {}
}
}
}

```

Méthode de réception

```

public static void receivemsg ( String nomfichier,BufferedInputStream inBuffer1) throws
Exception,UnknownHostException, IOException
{
    int lu1;
    long taille = 0;
    OutputStream out1 =new FileOutputStream (nomfichier);
    BufferedOutputStream outBuffer1=new BufferedOutputStream(out1);
    try {
        lu1 =inBuffer1.read(); // prise le String de client
        int compteur1=0;
        while (lu1>-1){
            outBuffer1.write(lu1);
            lu1=inBuffer1.read();
            compteur1=compteur1+1;
        }
    }
}

```

```

    }
    outBuffer1.flush();
    outBuffer1.close();
    inBuffer1.close();
    out1.flush();
    out1.close();
}
catch (IOException e) {
    System.out.println("N'a pas pu obtenir I/O pour la connection à :");
    System.exit(1);
}
}

```

Méthode de lecture du message

```

public static void liremsgreçu1 (String chemainfichier ) throws Exception, IOException
{
    BufferedReader entre11 = new BufferedReader(new FileReader(chemainfichier));
    String s11 = new String();
    StringTokenizer token11;
    String ligne11=new String();
    double d11;
    ligne11 = entre11.readLine();
    // System.out.print(" \ncontenu de msg :" );
    ligne11 = entre11.readLine();
    token11 = new StringTokenizer(ligne11, " ");
    s11 = token11.nextToken();
    Sender=s11;// System.out.print(" \nSender :"+ s );
    ligne11 = entre11.readLine();
    token11 = new StringTokenizer(ligne11, " ");
    s11 = token11.nextToken();
    Receiver=s11;// System.out.print(" \nReceiver :"+ s );
    ligne11 = entre11.readLine();
    token11 = new StringTokenizer(ligne11, " ");
    s11 = token11.nextToken();
    //System.out.print(" \ncontent :" );
    ligne11 = entre11.readLine();
    token11 = new StringTokenizer(ligne11, " ");
    s11 = token11.nextToken();
    d11 = Double.parseDouble(s11);
    VariableCommun.epsh_ax=d11;
}

```

```

ligne11 = entre11.readLine();
token11 = new StringTokenizer(ligne11, " ");
s11 = token11.nextToken();
d11 = Double.parseDouble(s11);
VariableCommun.epsb_ax=d11;
VariableCommun.i=1;
for (VariableCommun.i=1; VariableCommun.i<=VariableCommun.m; VariableCommun.i++) {
    for (VariableCommun.j=1; VariableCommun.j<=VariableCommun.n ; VariableCommun.j++) {
        ligne11 = entre11.readLine();
        token11 = new StringTokenizer(ligne11, " ");
        s11 = token11.nextToken();
        if ((s11.startsWith("+"))||(s11.startsWith("-"))||(s11.startsWith("0"))||(s11.startsWith("1"))||(s11.startsWith("2"))||(s11.startsWith("3"))||(s11.startsWith("4"))||(s11.startsWith("5"))||(s11.startsWith("6"))||(s11.startsWith("7"))||(s11.startsWith("8"))||(s11.startsWith("9"))) {
            d11 = Double.parseDouble(s11);
            VariableCommun.pu[VariableCommun.i][VariableCommun.j]=d11;
        }
    }
}
entre11.close();
}

```

2) Agent mouvement Axial : est un programme qui dispose d'un thread de réception qui contient une méthode de réception et une méthode de lecture du message reçu de l'agent mouvement Global ; et des méthodes de calculs de pression Pu et des charges hydrodynamiques Fu et Mu .

Ensuite il dispose d'un thread d'envoi qui se compose d'une méthode d'envoi permettant de réaliser la tâche d'envoi des messages résultats à l'agent mouvement Global.

Thread de réception

```

package analysehydrodynamicpiston;
import java.math.*;
import java.io.*;
import java.net.UnknownHostException;
import java.lang.*; // for Threads
import java.net.*; // sockets
import java.util.*;
public class threadax extends Thread{
    Socket amaxiSocket =null;
    Socket Soc =null;
}

```

```

ServerSocket ssoc=null;
int port ; // port
int cpt=0;
boolean tt=true;
threadax(int port) {
    this.port=port;
}
public void run () {
    try {
        ssoc = new ServerSocket(port,11);
        while (tt==true){
            Soc = ssoc.accept();
            BufferedInputStream inBuffer1=new BufferedInputStream(Soc.getInputStream());
            if (cpt==0){
                AgentmvtAxial.receivemsg ( "c:/Execution/MSGdemandePUreçu.dat",inBuffer1);
                try {
                    AgentmvtAxial.liremsgreçu1 ("c:/Execution/MSGdemandePUreçu.dat");
                catch (Exception e) {}
            if ( AgentmvtAxial.vbar>0){
                // iflag=-1;
                AgentmvtAxial.iflag=1;
            }
            else if ( AgentmvtAxial.vbar<0) {
                //iflag=1;
                AgentmvtAxial.iflag=-1;
            }
            for ( AgentmvtAxial.i=1; AgentmvtAxial.i<= AgentmvtAxial.m; AgentmvtAxial.i++) {
                for ( AgentmvtAxial.j=1; AgentmvtAxial.j<= AgentmvtAxial.n ; AgentmvtAxial.j++) {
                    AgentmvtAxial.pu[ AgentmvtAxial.i][ AgentmvtAxial.j]=0;
                }
            }
            // System.out.println("appelle les méthodes");
            AgentmvtAxial.efilm1 ();
            AgentmvtAxial.deriv1 ();
            AgentmvtAxial.coeff1 ();
            saveaxi svaxi=new saveaxi();
            for ( AgentmvtAxial.i=1; AgentmvtAxial.i<= AgentmvtAxial.m; AgentmvtAxial.i++) {
                for ( AgentmvtAxial.j=1; AgentmvtAxial.j<=(1+( AgentmvtAxial.n-1)/2); AgentmvtAxial.j++) {
                    svaxi.AA[ AgentmvtAxial.i][ AgentmvtAxial.j]= AgentmvtAxial.A[ AgentmvtAxial.i][ AgentmvtAxial.j];
                    svaxi.BB[ AgentmvtAxial.i][ AgentmvtAxial.j]= AgentmvtAxial.B[ AgentmvtAxial.i][ AgentmvtAxial.j];
                    svaxi.CC[ AgentmvtAxial.i][ AgentmvtAxial.j]= AgentmvtAxial.C[ AgentmvtAxial.i][ AgentmvtAxial.j];
                }
            }
        }
    }

```

```

svaxi.DD[ AgentmvtAxial.i][ AgentmvtAxial.j]= AgentmvtAxial.D[ AgentmvtAxial.i][ AgentmvtAxial.j];
svaxi.EE[ AgentmvtAxial.i][ AgentmvtAxial.j]= AgentmvtAxial.E[ AgentmvtAxial.i][ AgentmvtAxial.j];
svaxi.FF[ AgentmvtAxial.i][ AgentmvtAxial.j]= AgentmvtAxial.F[ AgentmvtAxial.i][ AgentmvtAxial.j];
    }
}

Pression pressionaxi= new Pression();
pressionaxi.calculpression(1);
for ( AgentmvtAxial.i=1; AgentmvtAxial.i<= AgentmvtAxial.m; AgentmvtAxial.i++) {
    for ( AgentmvtAxial.j=1; AgentmvtAxial.j<=1+( AgentmvtAxial.n-1)/2; AgentmvtAxial.j++) {
        AgentmvtAxial.pu[ AgentmvtAxial.i][ AgentmvtAxial.j]= AgentmvtAxial.pu[ AgentmvtAxial.i][
AgentmvtAxial.j]+pressionaxi.p[ AgentmvtAxial.i][ AgentmvtAxial.j];
    }
}

for ( AgentmvtAxial.i=1; AgentmvtAxial.i<= AgentmvtAxial.m; AgentmvtAxial.i++) {
    for ( AgentmvtAxial.j=1; AgentmvtAxial.j<=1+( AgentmvtAxial.n-1)/2; AgentmvtAxial.j++) {
        AgentmvtAxial.fe6.print( AgentmvtAxial.i);
        AgentmvtAxial.fe6.print(" ");
        AgentmvtAxial.fe6.print( AgentmvtAxial.j);
        AgentmvtAxial.fe6.print(" ");
        AgentmvtAxial.fe6.println( AgentmvtAxial.pu[ AgentmvtAxial.i][ AgentmvtAxial.j]);
    }

for ( AgentmvtAxial.j=2+( AgentmvtAxial.n-1)/2; AgentmvtAxial.j<= AgentmvtAxial.n; AgentmvtAxial.j++) {
    AgentmvtAxial.kk= AgentmvtAxial.n- AgentmvtAxial.j+1;
    AgentmvtAxial.fe6.print( AgentmvtAxial.i);
    AgentmvtAxial.fe6.print(" ");
    AgentmvtAxial.fe6.print( AgentmvtAxial.j);
    AgentmvtAxial.fe6.print(" ");
    AgentmvtAxial.fe6.println( AgentmvtAxial.pu[ AgentmvtAxial.i][ AgentmvtAxial.kk]);
}
}
AgentmvtAxial.fe6.close();
    VariableCommun.mutex_2.signal();
cpt++;
}
else if (cpt==1) {
    cpt=0;
    AgentmvtAxial.receiveMsg ( "c:/Execution/MSGdemandeFMreçu.dat",inBuffer1);
    try {
        AgentmvtAxial.liremsgreçu4 ("c:/Execution/MSGdemandeFMreçu.dat");
    } catch (Exception e) {}
ChargeHydro chargeaxi = new ChargeHydro ();

```

```

chargeaxi.calculchargefm(1);
AgentmvtAxial.Fudim= AgentmvtAxial.rapp1*(chargeaxi.Fubar)* AgentmvtAxial.vbar;
AgentmvtAxial.Fudim1= AgentmvtAxial.rapp1*(chargeaxi.Fubar1)* AgentmvtAxial.vbar;
AgentmvtAxial.Fudim2= AgentmvtAxial.rapp1*(chargeaxi.Fubar2)* AgentmvtAxial.vbar;
AgentmvtAxial.fe5.print("charge hydro Force movt axial : ");
AgentmvtAxial.fe5.println( AgentmvtAxial.Fudim);
System.out.println("charge hydro Force movt axial Fudim : "+
AgentmvtAxial.Fudim+"Fudim1"+AgentmvtAxial.Fudim1 +"Fudim2"+AgentmvtAxial.Fudim2 );
AgentmvtAxial.FMudim= AgentmvtAxial.rapp2*(chargeaxi.FMubar)* AgentmvtAxial.vbar;
AgentmvtAxial.FMudim1= AgentmvtAxial.rapp2*(chargeaxi.FMubar1)* AgentmvtAxial.vbar;
AgentmvtAxial.FMudim2= AgentmvtAxial.rapp2*(chargeaxi.FMubar2)* AgentmvtAxial.vbar;
AgentmvtAxial.fe5.print("charge hydro Momment movt axial : ");
AgentmvtAxial.fe5.println( AgentmvtAxial.FMudim); //afficher les charges hydrodynamiques
System.out.println("charge hydro Momment movt axial 1 et 2 FMudim : "+ AgentmvtAxial.FMudim+"FMudim1"+
AgentmvtAxial.FMudim1 +"FMudim2" +AgentmvtAxial.FMudim2);
// System.out.println("charge hydro Momment movt axial : "+ AgentmvtAxial.FMudim);
AgentmvtAxial.fe5.close();
VariableCommun.mutex_3.signal();
}
} //while()
ssoc.close();}
catch (Exception e) {}
}
}

```

Création du message type 1

```

// création du message de résultat 1
AgentmvtAxial.fe15.println("1");
AgentmvtAxial.fe15.println("Sender : "+AgentmvtAxial.Receiver + " "+AgentmvtAxial.adrr2);
AgentmvtAxial.fe15.println("Receiver : "+AgentmvtAxial.Sender+" " +AgentmvtAxial.adrr1);
AgentmvtAxial.fe15.println("Content : ");
AgentmvtAxial.fe15.println(AgentmvtAxial.epsh_ax );
AgentmvtAxial.fe15.println(AgentmvtAxial.epsb_ax );
for ( AgentmvtAxial.i=1; AgentmvtAxial.i<= AgentmvtAxial.m; AgentmvtAxial.i++) {
for ( AgentmvtAxial.j=1; AgentmvtAxial.j<=1+( AgentmvtAxial.n-1)/2; AgentmvtAxial.j++) {
AgentmvtAxial.fe15.println( AgentmvtAxial.pu[ AgentmvtAxial.i][ AgentmvtAxial.j]);
}
for ( AgentmvtAxial.j=2+( AgentmvtAxial.n-1)/2; AgentmvtAxial.j<= AgentmvtAxial.n; AgentmvtAxial.j++) {
AgentmvtAxial.kk= AgentmvtAxial.n- AgentmvtAxial.j+1;
AgentmvtAxial.fe15.println( AgentmvtAxial.pu[ AgentmvtAxial.i][ AgentmvtAxial.kk]);
}
}
}

```

```

    }
}
AgentmvtAxial.fe15.close();

```

Création du message type 2

```

// création du message de resultats 2
AgentmvtAxial.fe16.println("1");
AgentmvtAxial.fe16.println("Sender : "+AgentmvtAxial.Receiver +" "+AgentmvtAxial.adrr2);
AgentmvtAxial.fe16.println("Receiver : "+AgentmvtAxial.Sender+" " +AgentmvtAxial.adrr1);
AgentmvtAxial.fe16.println("Content : ");
AgentmvtAxial.fe16.println( AgentmvtAxial.Fudim);
AgentmvtAxial.fe16.println( AgentmvtAxial.FMudim);
AgentmvtAxial.fe16.close();

```

Méthode de réception

```

public static void receivemsg ( String nomfichier, BufferedInputStream inBuffer1) throws
Exception,UnknownHostException, IOException

```

```

{ int lu1;
  long taille = 0;
  OutputStream out1 =new FileOutputStream (nomfichier);
  BufferedOutputStream outBuffer1=new BufferedOutputStream(out1);
  try {
    lu1 =inBuffer1.read(); // prise le String de client
    int compteur1=0;
    while (lu1>-1){
      outBuffer1.write(lu1);
      lu1=inBuffer1.read();
      compteur1=compteur1+1;
    }
    outBuffer1.flush();
    outBuffer1.close();
    inBuffer1.close();
    out1.flush();
    out1.close();
  }
  catch (IOException e) {
    System.out.println("N'a pas pu obtenir I/O pour la connection à : de receive ");
    System.exit(1);    }    }

```

Méthode de lecture

```

public static void liremsgreçu1 (String chemainfichier1) throws Exception, IOException

```

```

{
BufferedReader entre1 = new BufferedReader(new FileReader(chemainfichier1));
String s1 = new String();
String s2 = new String();
String s3 = new String();
StringTokenizer token1;
String ligne1=new String();
double d1;
int nb1=0;
while ( (ligne1=entre1.readLine())!=null) {
nb1=nb1+1;
token1 = new StringTokenizer(ligne1, " ");
s1 = token1.nextToken();
s2 = token1.nextToken();
s2 = token1.nextToken();
s3 = token1.nextToken();
if (nb1==1) { Sender=s2;adrr1=s3; }
if (nb1==2) { Receiver=s2;adrr2=s3; }
if (nb1==3) { }
if ((s1.startsWith("+"))||(s1.startsWith("-"))||(s1.startsWith("0"))||(s1.startsWith("1"))||(s1.startsWith("2"))||(s1.startsWith("3"))||(s1.startsWith("4"))||(s1.startsWith("5"))||(s1.startsWith("6"))||(s1.startsWith("7"))||(s1.startsWith("8"))||(s1.startsWith("9"))) {
d1 = Double.parseDouble(s1);
if (nb1==4) { epsH=d1;
}
if (nb1==5) { epsB=d1;
}
if (nb1==6) { vbar=d1;
}
if (nb1==7) { demande=s1;
}
} entre1.close();
}

```

Thread d'envoi


```

package analysehydrodynamicpiston;
import java.math.*;
import java.io.*;
import java.net.UnknownHostException;
import java.lang.*; // for Threads
import java.net.*; // sockets
import java.util.*;
public class AgentmvtAxial extends Thread {
    //déclaration des données
    //initialisation des éxcentricités
    AgentmvtAxial() { }
public void run () {
    try {
        fe5= new PrintStream (new FileOutputStream ("c:/Execution/ch_hy.dat"));
        fe6= new PrintStream (new FileOutputStream ("c:/Execution/pu.dat"));
        fe15= new PrintStream (new FileOutputStream ("c:/Execution/putemp.dat"));
        fe16= new PrintStream (new FileOutputStream ("c:/Execution/chargaxitemp.dat"));
        pi=Math.acos(-1);
        cst=( bb/ alongjupe);
        alongcirc=2* pi* ray;
        along= alongjupe/ ray;
        pascirc=2* pi/( n-1);
        pasaxial= along/( m-1);
        omega= w*2* pi/60;
        den=6* r* amu* omega* ray;
        rapp1=6* amu* r* omega*Math.pow( ray,3)/( cc*2);
        rapp2= rapp1* ray;
        abar= aa/ ray;
        deltateta= deltae* pi/180;
        deltat= deltateta/ omega;
        delt2=(1/Math.pow(( deltateta),2));
        final double cste= aa/ alongjupe;
        cst1=1/cste;
        System.out.println("/n debut mouvement axial : ");
        threadax threadax1 = new threadax (3000);
        threadax1.start();
    } catch (IOException e) {}
    try {VariableCommun.mutex_2.Wait();
        sendmsgaxi ("c:/Execution/putemp.dat", 3000);
    } catch (IOException e) {}
    try {
        VariableCommun.mutex_3.Wait();
    }
}
}

```

```

    sendmsgaxi ("c:/Execution/chargaxitemp.dat",3000);
} catch (IOException e) {}
System.out.println("fin mouvement axial : "+ VariableCommun.k);
VariableCommun.k++;
VariableCommun.mutex_13.signal();
VariableCommun.mutex_2 = new semaphore(0);
VariableCommun.mutex_3 = new semaphore(0);
VariableCommun.mutex_13 = new semaphore(0);
VariableCommun.mutex_12 = new semaphore(0);
} catch (Exception e) {}
} //run

```

Méthode d'envoi

```

public static void sendmsgaxi (String nomfichier, int port) throws Exception,UnknownHostException, IOException
{
    File f = new File(nomfichier);
    if(f.exists())
    {
        amaxiSocket = new Socket("192.168.1.5", port);
        OutputStream fluxsortie = amaxiSocket.getOutputStream();
        InputStream in = new BufferedInputStream(new FileInputStream(f));
        ByteArrayOutputStream tableaubytes = new ByteArrayOutputStream();
        BufferedOutputStream tampon = new BufferedOutputStream(tableaubytes);
        int lu = in.read();
        int[] aecrire = new int[4097];
        int compteur = 0;
        //Tant qu'on est pas à la fin fichier
        while(lu > -1)
        {
            aecrire[compteur] = lu;//On lit les données du fichier
            lu = in.read();
            compteur++;
            //Quand on a rempli le tableau, on envoie un paquet de 4096 octets
            if(compteur == 4096)
            {
                compteur=0;
                for(int x=0;x<4096;x++){ //On remplit le tampon
                    tampon.write(aecrire[x]);}
                fluxsortie.write(tableaubytes.toByteArray());//Et on l'envoie
                tableaubytes.reset();
            }
        }
    }
}

```

```

    }
    //while
    //On envoie le dernier paquet, qui ne fait pas forcément 4096
    for(int x=0;x<compteur;x++) //On remplit le tampon
    tampon.write(aecrire[x]);
    tampon.flush();//Et on l'envoie
    fluxsortie.write(tableaubytes.toByteArray());
    fluxsortie.flush();
    in.close();
    tampon.close();
    amaxiSocket.close();
    }//if exist
    else
    {
    System.out.println("Le fichier "+f+" est introuvable");
    }
    f.deleteOnExit();
    }

```

3) Agent mouvement Latéral : est un programme qui dispose d'un thread de réception qui contient une méthode de réception et une méthode de lecture du message reçu de l'agent mouvement Global ; et des méthodes de calculs de pression P1c et des charges hydrodynamiques F1c et M1c . Ensuite il dispose d'un thread d'envoi qui se compose d'une méthode d'envoi permettant de réaliser la tâche d'envoi des messages résultats à l'agent mouvement Global.

4) Agent mouvement Rotation : c'est un programme dispose aussi d'un thread de réception qui à en lui même une méthode de réception et une méthode de lecture du message reçu de l'agent mouvement Global ; ses propres méthodes de calculs de pression P2c et des charges hydrodynamiques F2c et M2c. Ensuite il dispose d'un thread d'envoi qui permet de réaliser la tâche d'envoi des messages par sa méthode d'envoi des messages résultats à l'agent mouvement Global;

5) Agent AMS : "Agent Management System"

Il supervise l'enregistrement des agents et leurs services fournis.

```

package analysehydrodynamicpiston;
public class AMS {
public static String Table_ams[][]=new String [20][3];
public AMS() {
    Table_ams[1][1]="Agent_Axial";
    Table_ams[2][1]="Agent_Axial";
    Table_ams[3][1]="Agent_Latéral";

```

```

Table_ams[4][1]="Agent_Latéral";
Table_ams[5][1]="Agent_Rotation";
Table_ams[6][1]="Agent_Rotation";
Table_ams[7][1]="Agent_Global";
Table_ams[8][1]="Agent_Global";
Table_ams[9][1]="Agent_Global";
Table_ams[1][2]="calcul_pu";
Table_ams[2][2]="calcul_FuMu";
Table_ams[3][2]="calcul_p1c";
Table_ams[4][2]="calcul_F1cM1c";
Table_ams[5][2]="calcul_p2c";
Table_ams[6][2]="calcul_F2cM2c";
Table_ams[7][2]="calcul_ptot";
Table_ams[8][2]="calcul_FMtot";
Table_ams[9][2]="calcul_Eh,Eb";
}}

```

Méthode de recherche

```

public void rechercheNomService (String Demande){
for (VariableCommun.i=1;VariableCommun.i<20;VariableCommun.i++){
if (Table_ams[VariableCommun.i][2]==Demande)
{
Receiver=Table_ams[VariableCommun.i][1];
}}}

```

6) Agent DF : "Director Facilitateur"

Il fournit un service de pages jaunes au système dont il contient les noms des Agents inclus dans le système avec leurs adresses IP.

```

package analysehydrodynamicpiston;
import java.net.*; // sockets
public class DF {
public static String Table_df[][]=new String [10][3];
public static Socket Table_df2[][]=new Socket [10][2];
public static Socket amgSocket, amaxiSocket, amlatSocket, amrotSocket;
public DF() {
Table_df[1][1]="Agent_Global";
Table_df[2][1]="Agent_Axial";
Table_df[3][1]="Agent_Latéral";
Table_df[4][1]="Agent_Rotation";
Table_df[1][2]="192.168.1.5";

```

```

Table_df[2][2]="192.168.1.2";
Table_df[3][2]="192.168.1.3";
Table_df[4][2]="192.168.1.4";
Table_df2[1][1]=amgSocket;
Table_df2[2][1]=amaxiSocket;
Table_df2[3][1]=amlatSocket;
Table_df2[4][1]=amrotSocket; } }
    
```

Méthode de recherche

```

public void rechercheAdrNom (String nomAgent){
    DF df1=new DF();
    for (VariableCommun.i=1;VariableCommun.i<10;VariableCommun.i++){
        if (df1.Table_df[VariableCommun.i][1]==nomAgent)
        {
            adr=df1.Table_df[VariableCommun.i][2];
            socke=df1.Table_df2[VariableCommun.i][1];
        } } }
    
```

6.4 Communication du Système Multi Agents

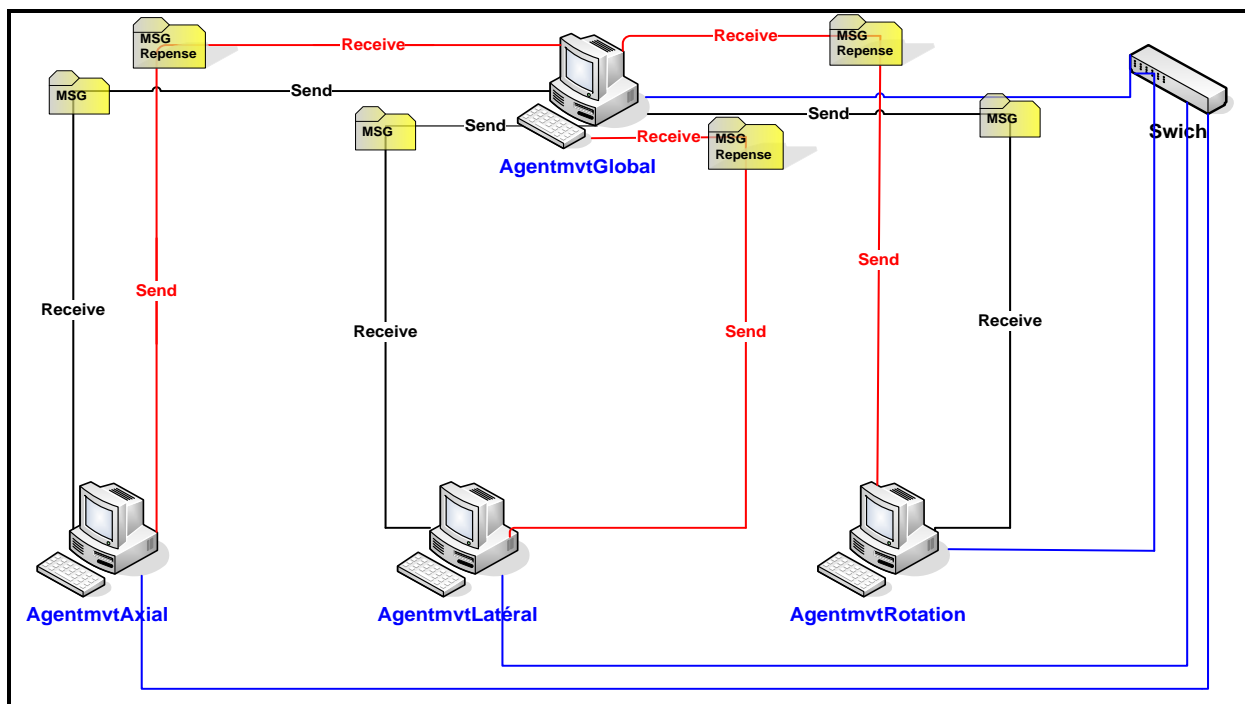


Figure 6.1 : Schéma du Système Multi Agents du modèle mécanique sous réseau

Cette figure illustre la communication de notre système multi agents, elle résume le travail qui se déroule entre les agents de base du système. Notre système forme un réseau local contenant quatre ordinateurs "poste" dont chaque agent est installé dans un poste, ces ordinateurs sont reliés entre eux avec des câbles

et un swieth pour communiquer entre eux selon le protocole TCP/IP en se basant sur l'architecture client/serveur où chaque agent joue les deux rôles en même temps client et serveur. Notre agent mouvement global joue le rôle d'un synthétiseur dont il reçoit les résultats partiels de ces trois agents figurants dans le schéma précédant qui font leurs propres calculs en parallèle; pour faire ses calculs globaux.

6.5 Déroulement de la communication

Dans cette partie en détaillant le déroulement de la communication entre l'agent du mouvement global et un des trois agents, prenant l'exemple de l'agent mouvement axial sachant que pour les deux autres agents on passe par les mêmes étapes.

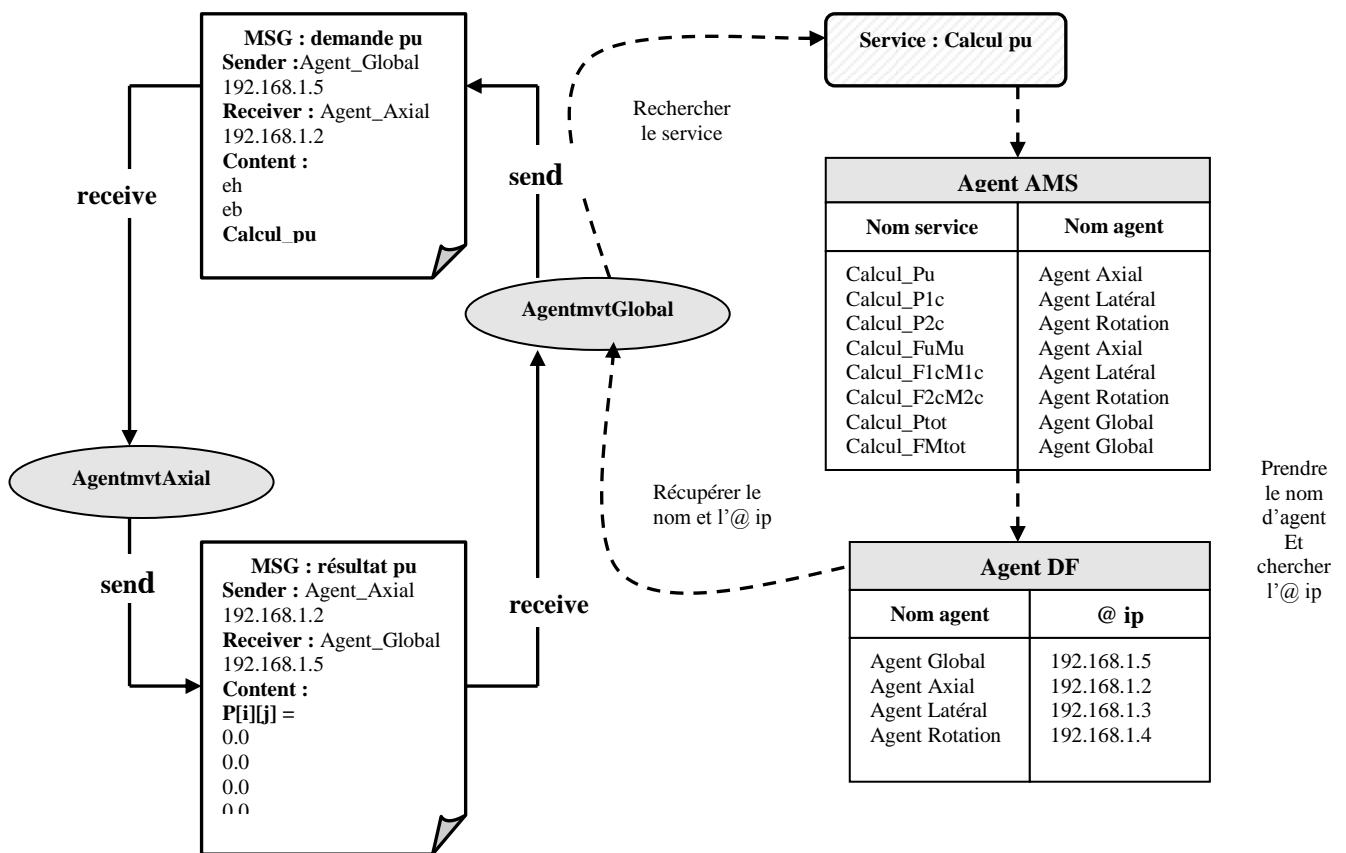


Figure 6.2 : Schéma du déroulement de la communication entre les agents

Lors de l'envoi d'un message de l'agent mouvement global à n'importe quel agent du système selon le service demandé, par exemple demande **calcul pu**, il consulte l'agent **AMS** pour chercher l'existence du service si oui, il renvoie le nom de leur agent responsable, quand il récupère le nom d'agent il consulte l'agent **DF** pour qu'il récupère l'adresse IP de cet agent.

A ce moment l'agent mouvement global peut envoyer son message à l'agent intéressé par exemple l'Agent mouvement axial.

MSG : demande pu
Sender :Agent_Global 192.168.1.5

Receiver : Agent_Axial 192.168.1.2

Quand l'agent mouvement axial reçoit le message demande, il enregistre chez lui le nom et l'adresse ip du Sender pour qu'il puisse renvoyer le message résultat de calcul au destinataire.

MSG : résultat pu

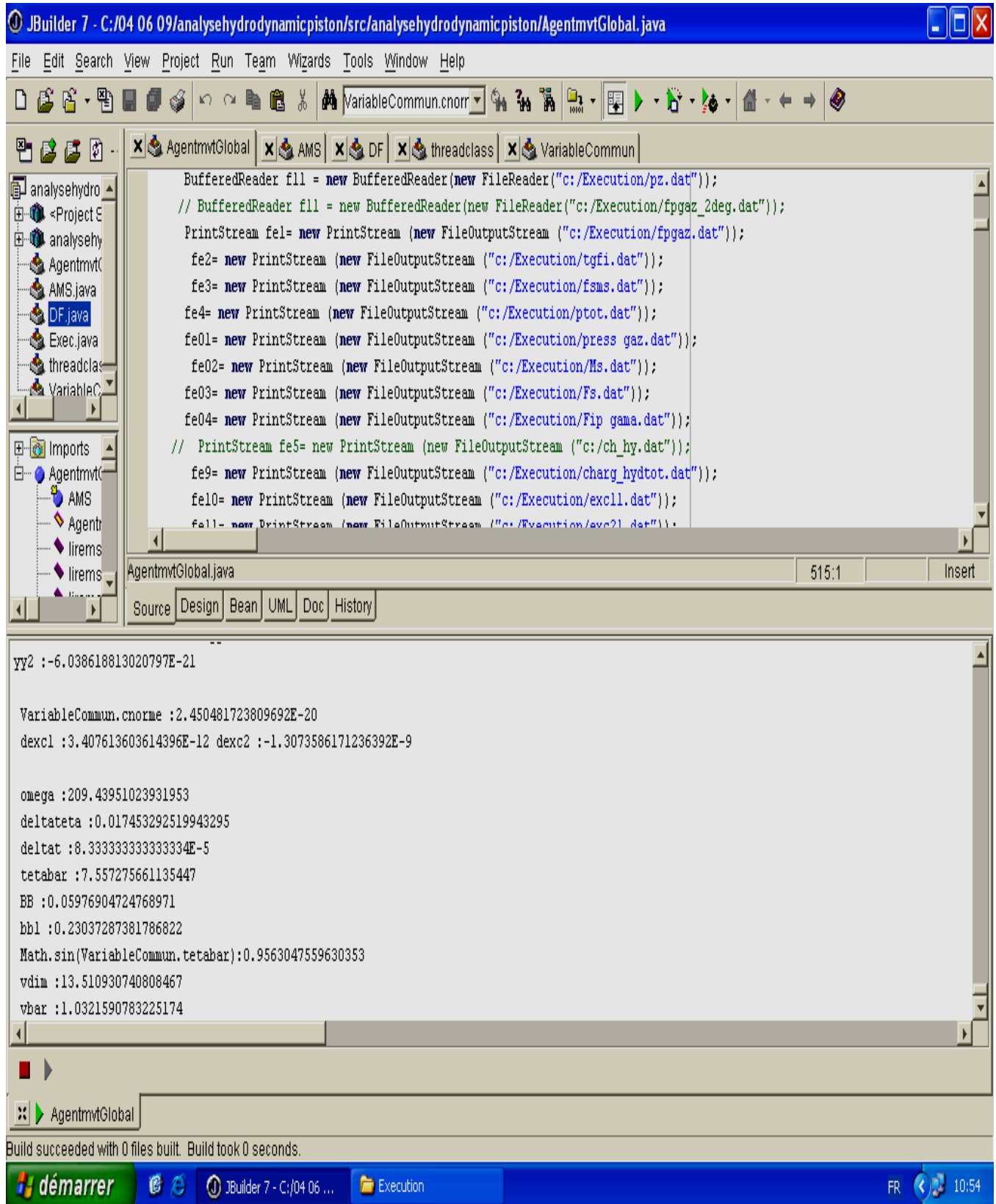
Sender : Agent_Axial 192.168.1.2

Receiver : Agent_Global 192.168.1.5

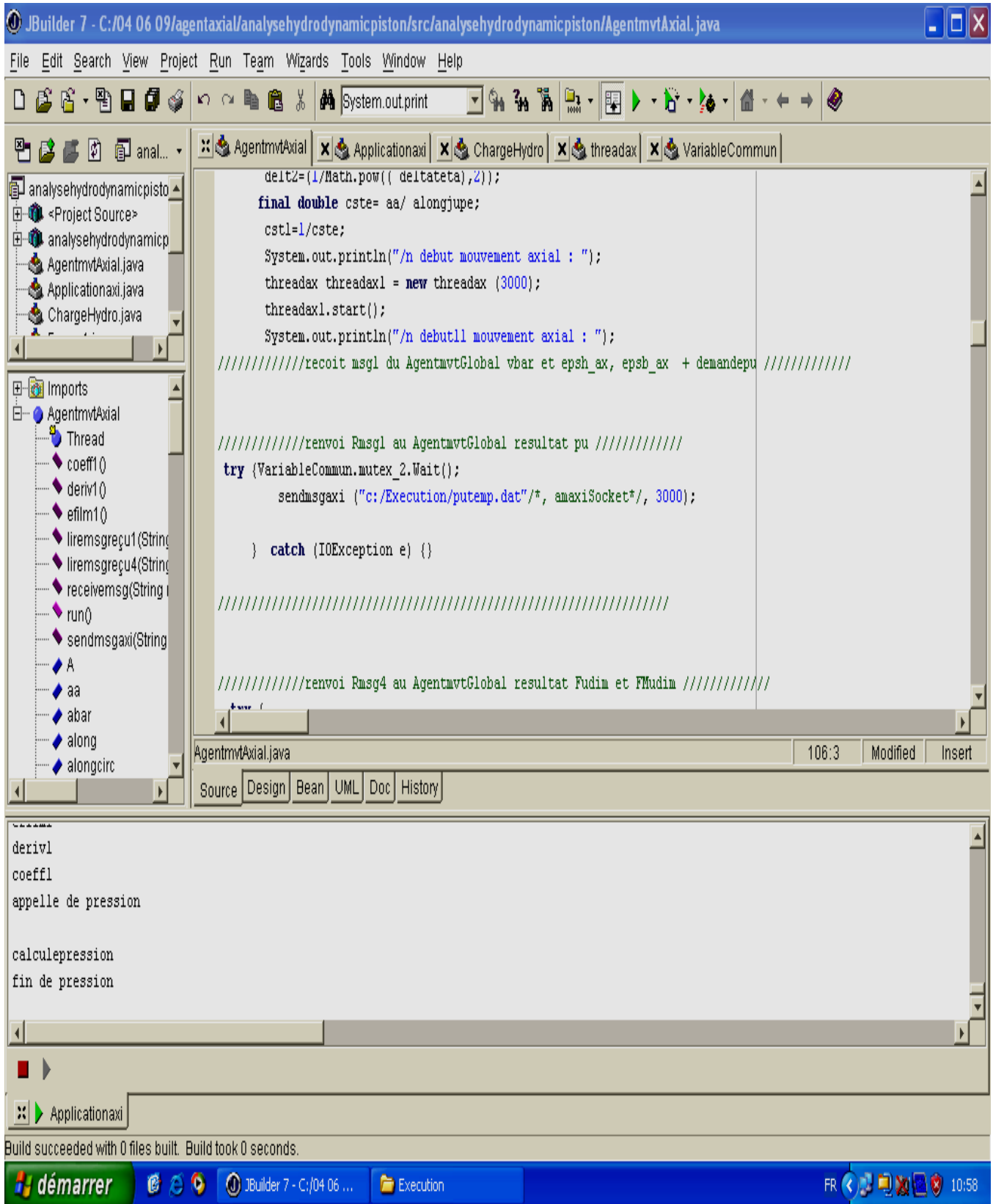
6.6 Exemple d'application de notre Système Multi Agents

Notre application a été exécuté sous un réseau local avec une connexion de 100 Mbit/s, il faut installer les quatre (04) sous application chacune dans un ordinateur ou poste

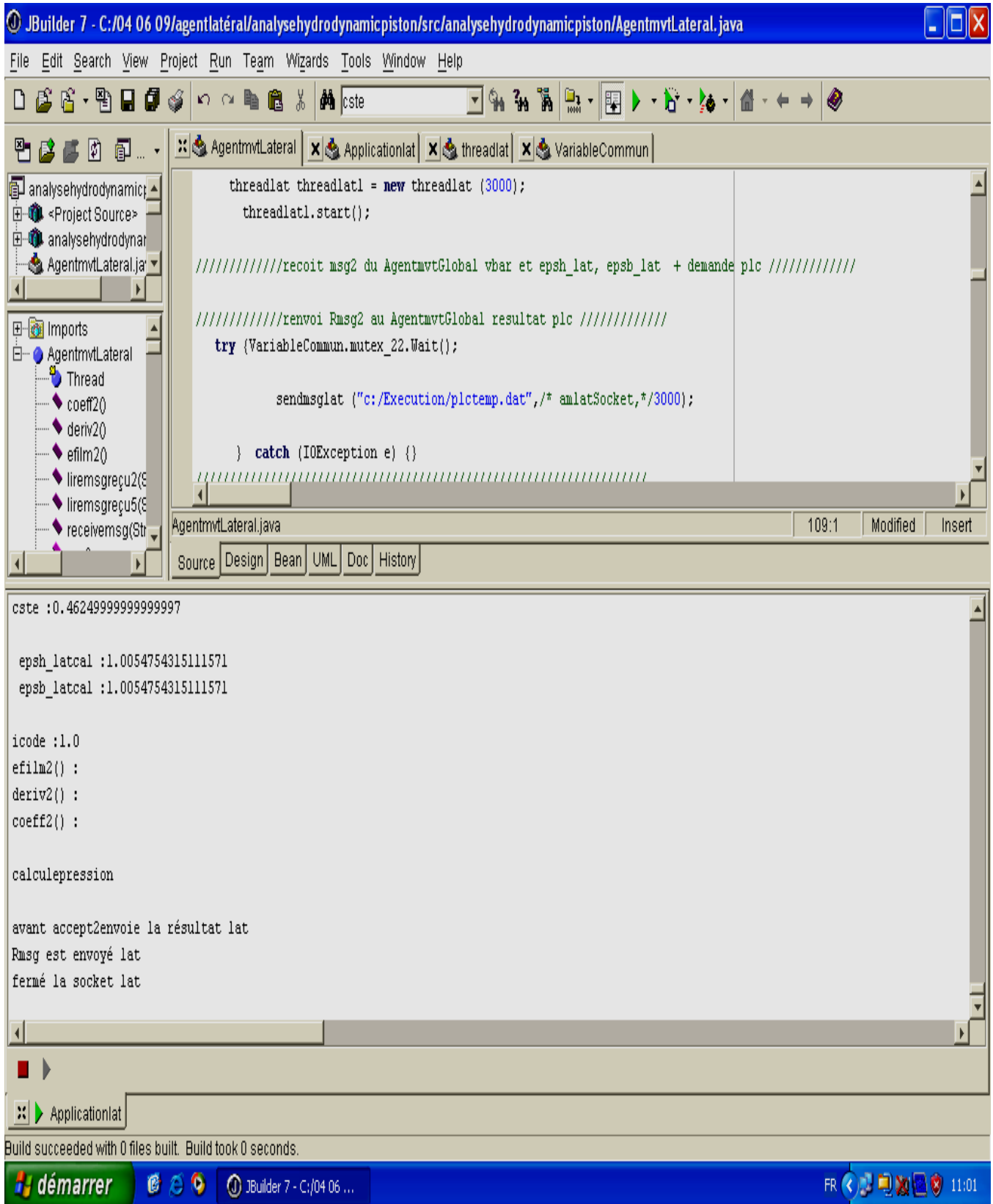
Lancer l'application d'Agent mouvement Global dans le poste 1



Lancer l'application d'Agent mouvement Axial dans un poste 2



Lancer l'application d'Agent mouvement Latéral dans un poste 3



Lancer l'application d'Agent mouvement Rotation dans un poste 4

The screenshot displays the JBuilder 7 IDE interface. The title bar shows the file path: `C:/04 06 09/agentrotation/analysehydrodynamicpiston/src/analysehydrodynamicpiston/AgentmvtRotation.java`. The menu bar includes File, Edit, Search, View, Project, Run, Team, Wizards, Tools, Window, and Help. The toolbar contains various icons for file operations and development tools.

The left sidebar shows a project tree for `analysehydrodynamicpiston.jk` with files like `AgentmvtRotation.java`, `Applicationrot.java`, `ChargeHydro.java`, and `Frame1.java`. Below it is an 'Imports' section listing classes such as `Thread`, `coeff2c1()`, `coeff2c2()`, `deriv31()`, `deriv32()`, `efilm31()`, `efilm32()`, `liremsgreçu3(String ch)`, `liremsgreçu6(String ch)`, and `receivemsg(String non)`.

The main editor window shows the source code for `AgentmvtRotation.java`. The code includes thread creation, message handling, and a try-catch block for an IOException. Comments in French describe the flow of messages and data between the thread and the global agent.

```

threadrot threadrot1 = new threadrot (3000);
threadrot1.start();
//////////recoit msg3 du AgentmvtGlobal vbar et epsb_rot, epsb_rot + demande p2c //////////

//////////renvoi Rmsg3 au AgentmvtGlobal resultat p2c //////////
try {VariableCommun.mutex_222.Wait();
    sendmsgrot ("c:/Execution/p2ctemp.dat"/*, amrotSocket*,3000);

} catch (IOException e) {}
//////////renvoi Rmsg6 au AgentmvtGlobal resultat F2cdim et FM2cdim //////////
//////////recoit msg6 du AgentmvtGlobal demande F2cdim et FM2cdim //////////

//forces en teta= pi

//////////renvoi Rmsg6 au AgentmvtGlobal resultat F2cdim et FM2cdim //////////
try {

```

The bottom console window shows the execution output:

```

-5.95767440106365
9.097818544337391
-0.2315839306027688
calcul_p2c
fin de receiiiiiiiiiiiiivePPPPPPPPepsh_rot :-6.963161299489602
epsb_rot :8.092322591298728vbar :-0.2315839306027688
epsb_rotcal :-6.963165487247982
epsb_rotcal :8.092327458153061
calculepression

```

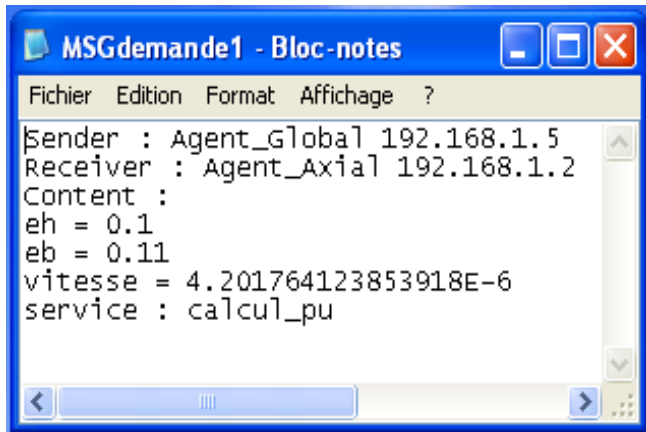
At the bottom, a status bar indicates 'Build succeeded with 0 files built. Build took 0 seconds.' The Windows taskbar at the very bottom shows the 'démarrer' button, the JBuilder 7 application icon, and the system clock at 11:04.

A ce moment le programme est en cours d'exécution dont en observant les échanges des messages entre nos agents c'est-à-dire entre les différentes postes du réseau.

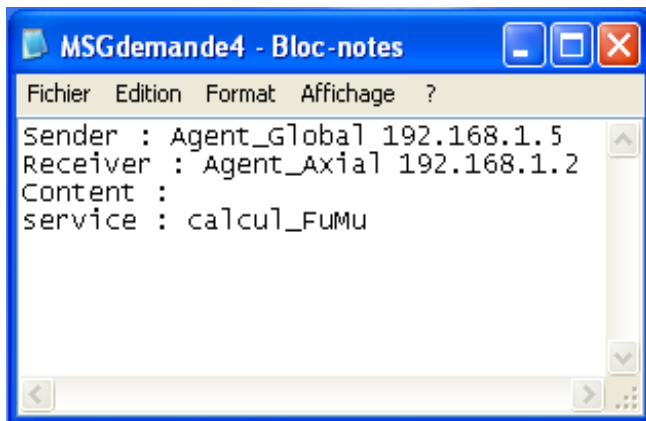
Par la suite on va voir un exemple des différents messages échangés dans notre réseau local.

1-Messages envoyés par l'AgentmvtGlobal

a) message demande de pression



b) message demande des charges hydrodynamiques



2-Messages reçus par l'AgentmvtGlobal

a) message résultats de pression

```

MSGpu - Bloc-notes
Fichier Edition Format Affichage ?
Sender : Agent_Axial 192.168.1.2
Receiver : Agent_Global 192.168.1.5
Content :
Pu[i][j] =
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
    
```

a) message résultats des charges hydrodynamiques

```

MSGchargaxi - Bloc-notes
Fichier Edition Format Affichage ?
Sender : Agent_Axial 192.168.1.2
Receiver : Agent_Global 192.168.1.5
Content :
Fu = -3.0471695254648867E-24
Mu = 0.0
    
```

6.7 Affichage des résultats

6.7.1 Le mouvement global

Le mouvement global est défini par les équations 2.23-2.24. La procédure de calcul est donnée par l’organigramme de la figure 4.4 qui consiste à décomposer le mouvement, en utilisant les équations des mouvements composés pour avoir les différents champs de pression ensuite le champ de pression global $\bar{p}(\bar{t})$ instantané est donné par l’équation 2.11.

Le tableau suivant montre les données utilisées dans la simulation du mouvement hydrodynamique.

Désignation	Variable	Valeur
Longueur de la jupe du piston	L	80 mm
Rayon du piston	R	60 mm
Rayon du vilebrequin	R	62.5 mm
Longueur de bielle	l	238 mm
Masse du piston	m_{pist}	1.9 Kg
Masse de l’axe	m_{axe}	0.9 Kg
Distance de l’axe au haut du piston	A	37 mm
Jeu radial de l’ensemble piston-cylindre (Température ambiante)	C	0.035 mm
Désaxage (piston-axe)	C_p	1.5 mm
Désaxage du centre de gravité	C_g	0
Vitesse de rotation	ω	2000t/mn
Viscosité dynamique de l’huile	μ	0.0069 Pa.s
Angle fluide assumé	α	15°

Tableau 6.1 : Données de base utilisées dans la simulation

Pour obtenir les résultats suivantes on a besoin de connaître les vitesses latérales et la vitesse axiale pour une position angulaire donnée du vilebrequin.

- donné de profil : $h=1$
- discrétisation de la rotation du vilebrequin : $\Delta\bar{\theta} = \frac{\pi}{90}$
- vitesse axiale : $\bar{U}(\bar{t}_i) = -1$
- vitesse transversale : $\dot{e}_h(\bar{t}_i) = 1.1$, $\dot{e}_b(\bar{t}_i) = 1$
- excentricités initiales : $\varepsilon_h = 0.5$, $\varepsilon_b = 0.2$

a) le mouvement axial

$$\varepsilon_h^{ax} = \varepsilon_b^{ax} = 0$$

La pression correspondante est donnée par les deux figures suivantes :

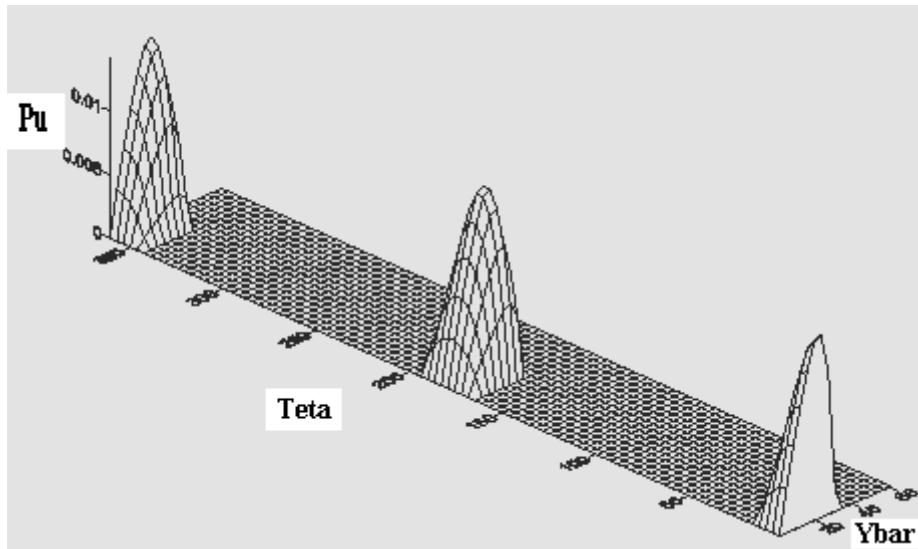


Figure 6.3 : Champ de pression P_u ($P_{max}=0.0145$)

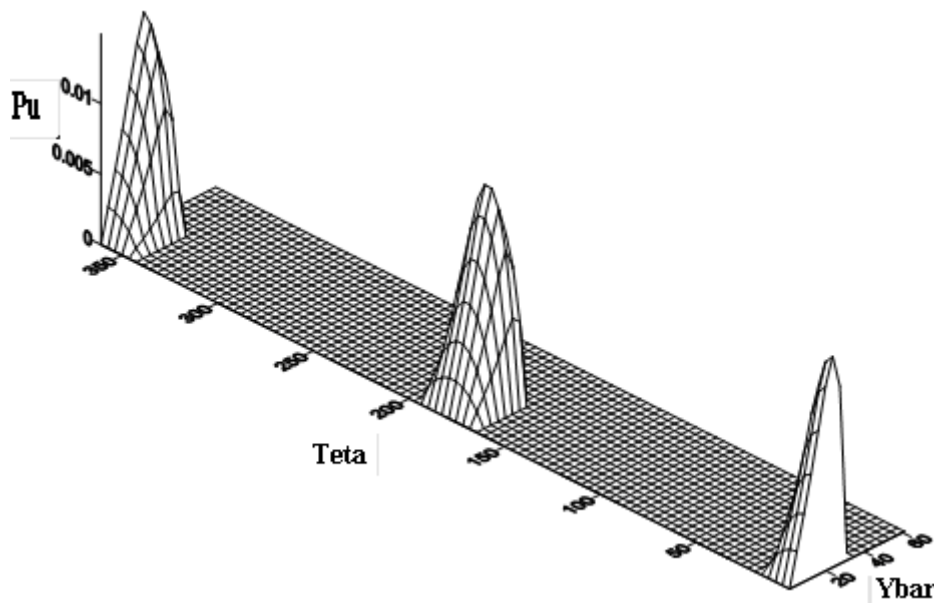


Figure 6.4 : Champ de pression P_u ($P_{max}=0.0145$), [10]

Les charges hydrodynamiques

On obtient les résultats suivants :

Mouvement axial : $\varepsilon_h^{ax} = \varepsilon_b^{ax} = 0$

Angle fluide de pression $(-\theta_1 \quad \theta_1)$: $F_u = 625.277 \text{ N}, \quad M_u = -10.068 \text{ N.m}$

Angle de pression $(\theta_2, \quad 2\pi - \theta_2)$: $F_u = -625.277 \text{ N}, \quad M_u = 10.068 \text{ N.m}$

b) le mouvement latéral

Le mouvement latéral, défini par les équations 2.27 est donné par $\epsilon_h^{Lat} = \epsilon_b^{Lat} = 0.36125$

Le champ de pression résultant est montré par les deux figures suivantes :

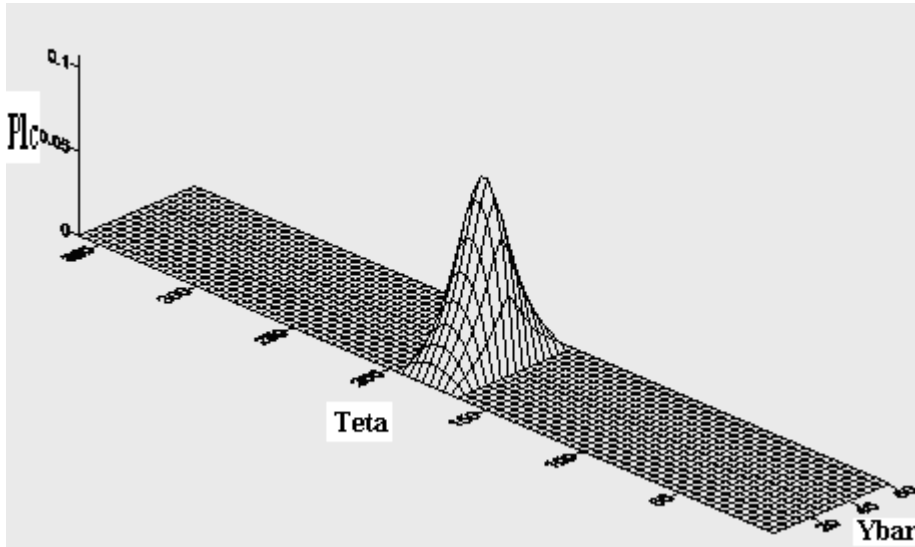


Figure 6.5 : Pression P_{1c} ($P_{max}=0.109$)

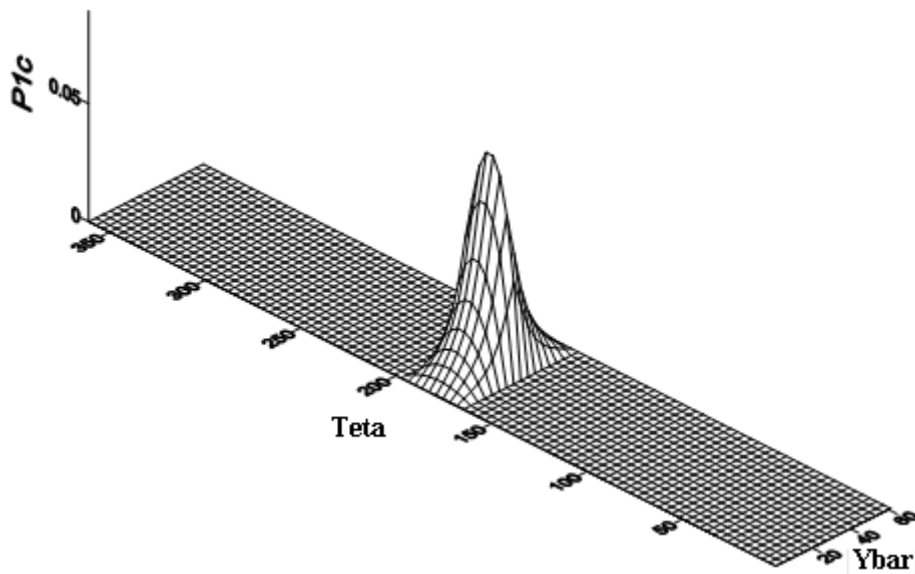


Figure 6.6 : Pression P_{1c} ($P_{max}=0.09$) , [10]

Les charges hydrodynamiques

Mouvement de déplacement latéral : $\epsilon_h^{Lat} = \epsilon_b^{Lat} = d = -0.08499$

$d < 0$, c'est un mouvement de déplacement vers la gauche.

Angle fluide de pression ($-\theta_1 \quad \theta_1$) : $F_{1c} = 3173.59 \text{ N}$, $M_{1c} = -9.474 \text{ Nm}$

Angle de pression ($\theta_2, \quad 2\pi - \theta_2$) : $F_{1c} = 0$, $M_{1c} = 0$

c) le mouvement rotation

Les équations de la rotation 2.28-2.29 donnent:

$$\varepsilon_h^{rot} = 0.13875$$

$$\varepsilon_b^{rot} = -0.16125$$

La pression est donnée par les deux figures suivantes :

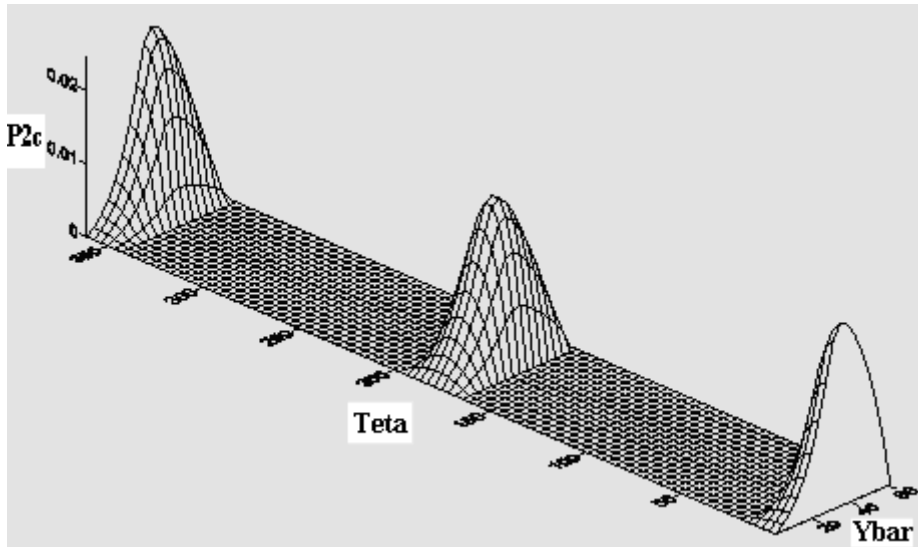


Figure 6.7 : Pression P_{2c} ($P_{max}= 0.024$)

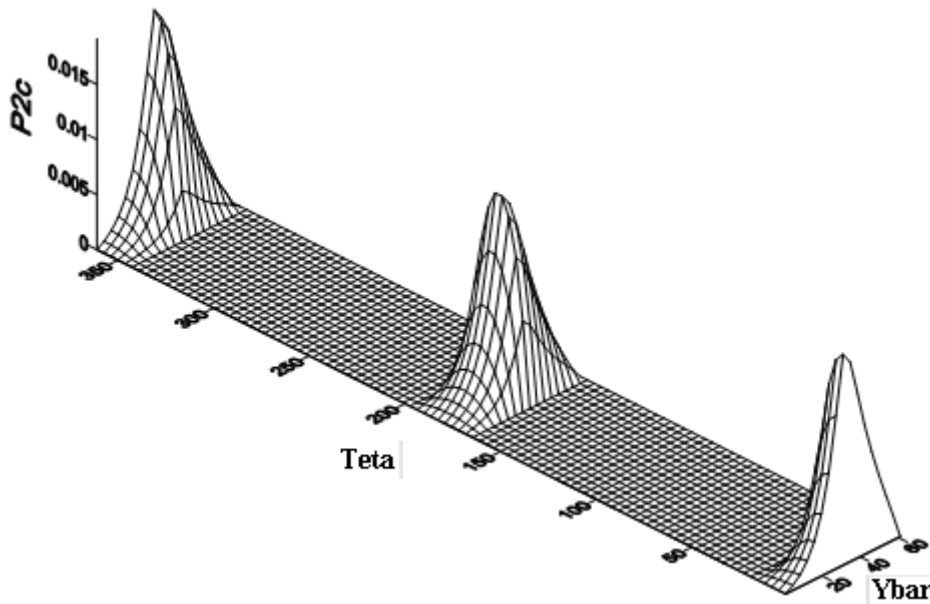


Figure 6.8 : Pression P_{2c} ($P_{max}= 0.019$) , [10]

Les charges hydrodynamiques

Mouvement de rotation latéral : $\varepsilon_h^{rot} = 0.185$, $\varepsilon_b^{rot} = -0.215$

Forces hydrodynamiques (calculées sur toute la hauteur de la jupe) :

Angle de pression ($-\theta_1 \quad \theta_1$): $F_{2c} = 1476.638 \text{ N}$, $M_{2c} = -14.488 \text{ Nm}$

Angle de pression ($\theta_2, 2\pi - \theta_2$) : $F_{2c} = -1381.82 \text{ N}$, $M_{2c} = -13.58 \text{ Nm}$

d) Le mouvement global

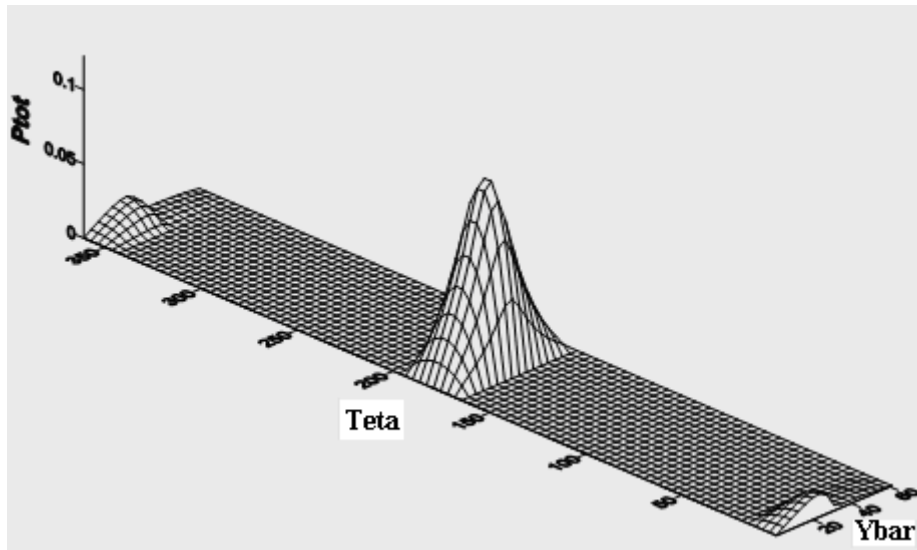


Figure 6.9 : Pression globale \bar{p} ($P_{\max}=0.125$)

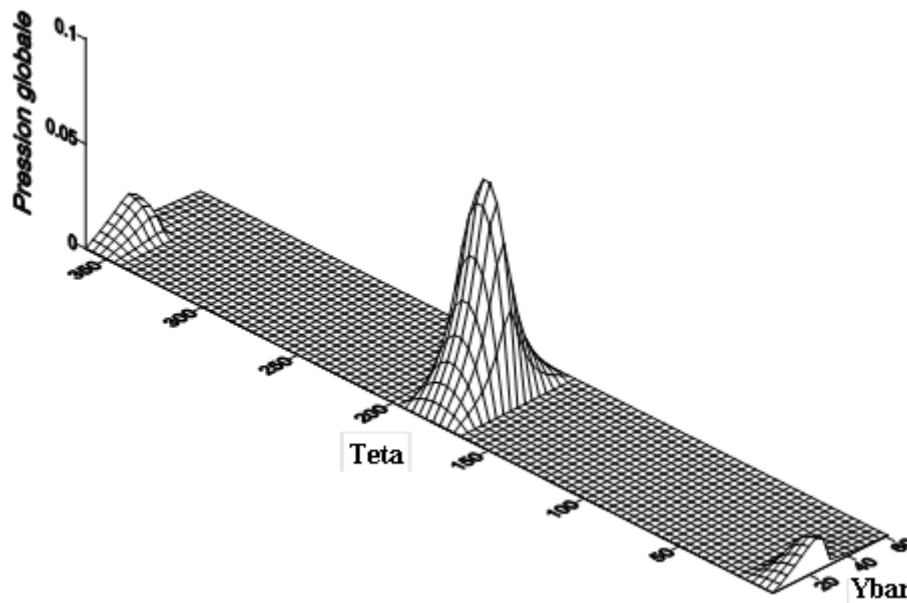


Figure 6.10 : Pression globale \bar{p} ($P_{\max}=0.104639$) , [10]

Les charges hydrodynamiques

Les charges hydrodynamiques sont donnée par la procédure de calcul suivant l’organigramme 4.7.

Les forces hydrodynamiques sont obtenues selon la configuration de cette figure ci-dessous, où le piston est libre de se mouvoir dans le cylindre dont cette configuration a été donnée par :

- Vitesses en début du pas de rotation du vilebrequin (quelconques) : $\dot{\epsilon}_h(\bar{t})=0$, $\dot{\epsilon}_b(\bar{t})=0.1$
- Vitesse axiale adimensionnée : $\bar{V} = 0.047$

- excentricités initiales : $\varepsilon_h(\bar{t})=1D-1$, $\varepsilon_b(\bar{t})=-3D-1$

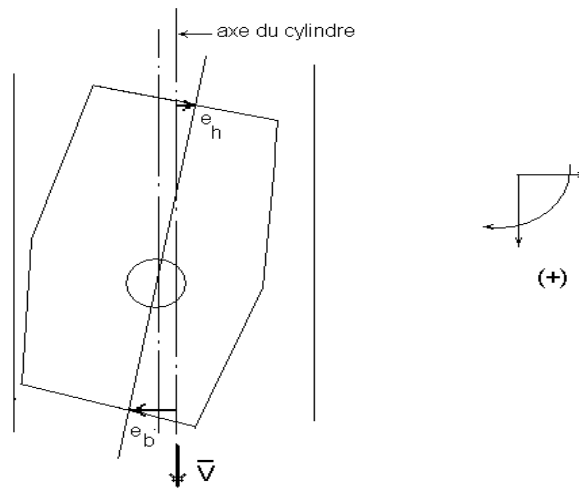


Figure 6.11 : Configuration du mouvement adopté dans l'exemple

$F_{tot} = 793.071N$, $M_{tot} = -47.94Nm$

Vitesses latérales corrigées par la méthode de Newton-Raphson : $\dot{\varepsilon}_h(\bar{t}) = 3.01E-6$, $\dot{\varepsilon}_b(\bar{t}) = 2.76E-5$

Cette figure résume les résultats des charges hydrodynamiques du mouvement global résultant ainsi que ceux des différents mouvements.

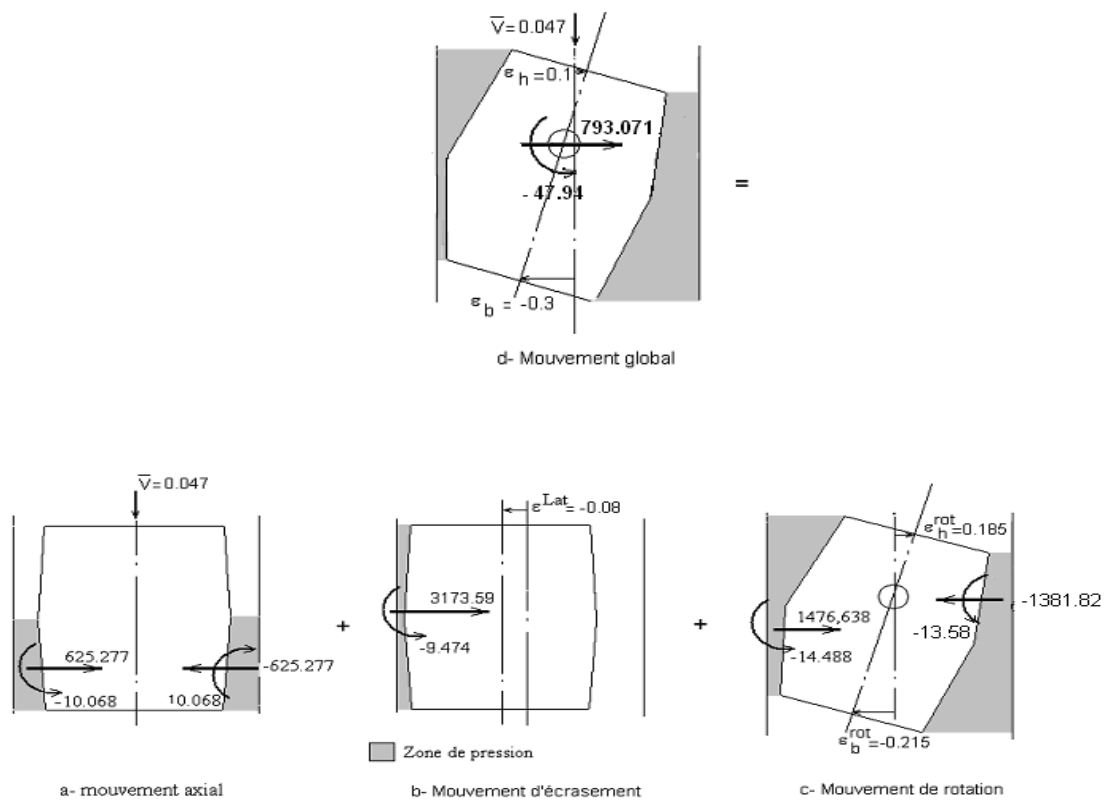
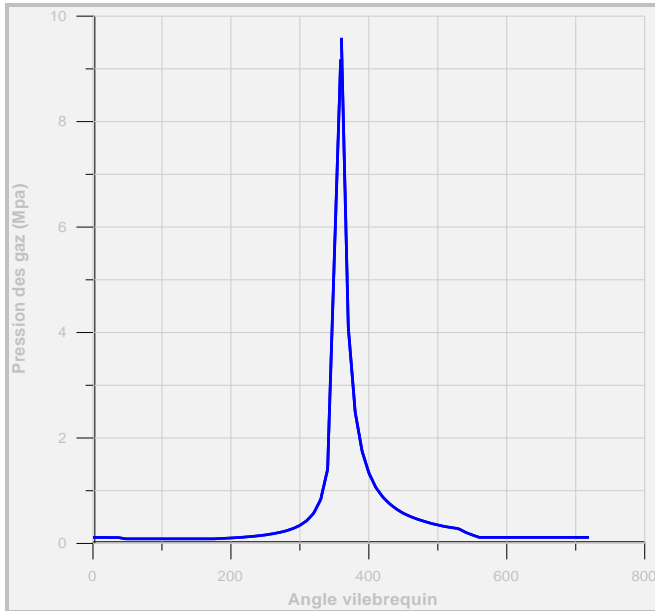


Figure 6.12 : Forces hydrodynamiques de la configuration de la figure 6.11

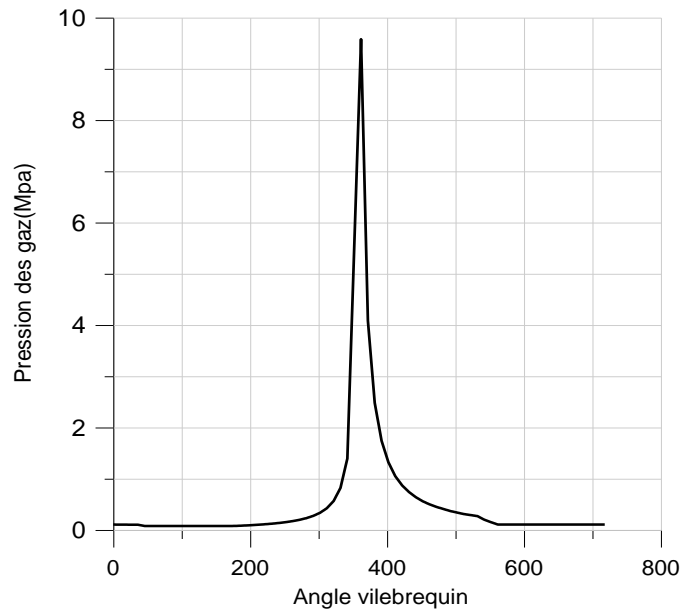
6.7.2 Calcul des excentricités

a) les charges dynamiques :

L'organigramme 4.2 pour obtenir la solution cyclique. La pression des gaz de combustion est donnée par la figure suivante. La pression maximale obtenue est $P_{max}=9577.5$ kPa. L'origine de cette courbe peut être pris au PMB, début de la phase d'aspiration comme le montre la figure 6.13, ou encore au PMH, début de la phase de détente figure 6.17.

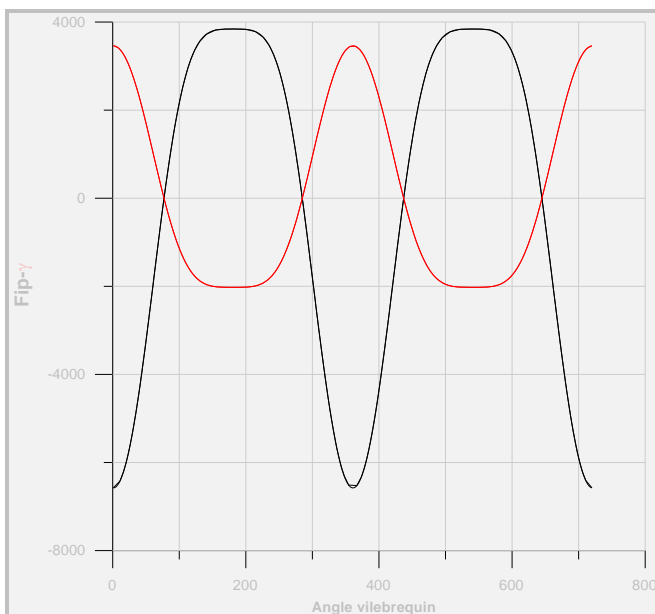


a) résultats avec SMA

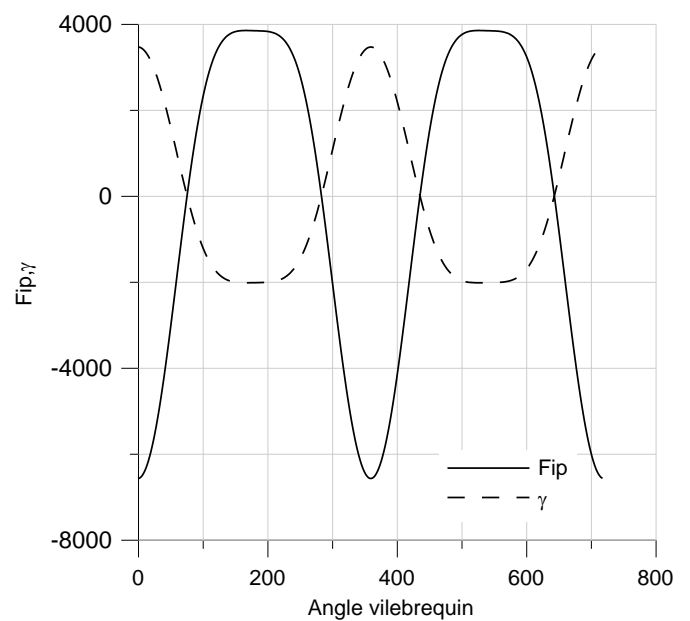


b) résultats , [10]

Figure 6.13 : Pression des gaz de combustion sur un cycle moteur



a) résultats avec SMA



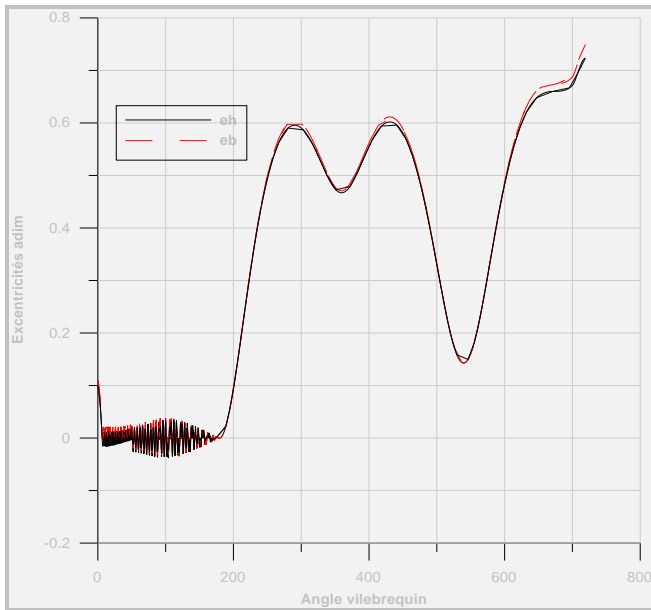
b) résultats, [10]

Figure 6.14 : Force d'inertie axiale \bar{F}_{IP} et accélération axiale du piston

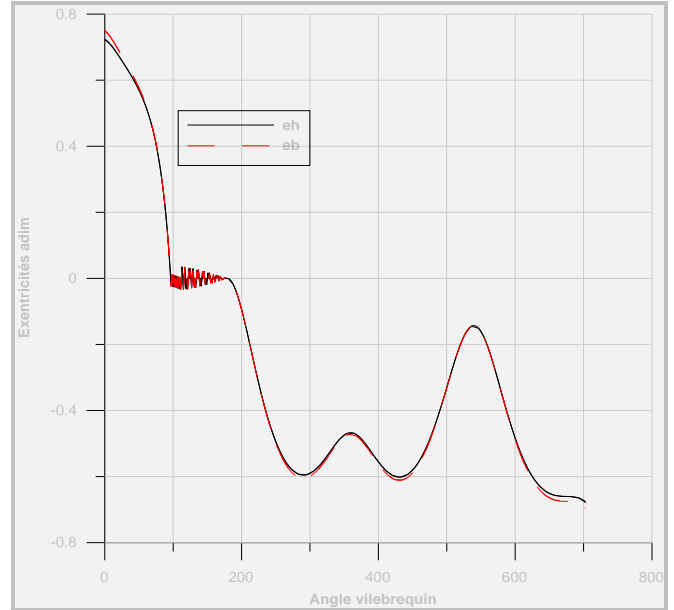
La convergence sera obtenue quand la condition de cyclicité : $\varepsilon_h(\bar{t} + 4\pi) = \varepsilon_h(\bar{t})$, $\varepsilon_b(\bar{t} + 4\pi) = \varepsilon_b(\bar{t})$ est satisfaite.

b) l'axe du piston occupe le milieu de la jupe : Pour obtenir les résultats suivants, on à :

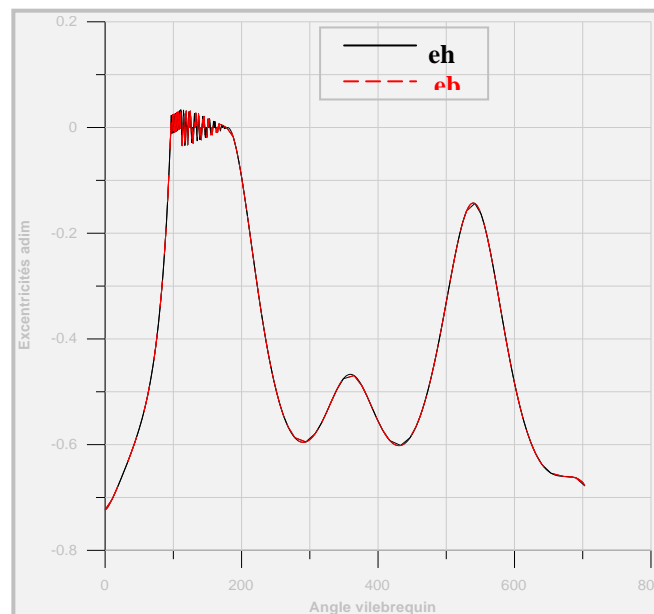
- Longueur de la jupe du piston : $L=90$ mm
- Distance de l'axe au haut du piston : $a=45$ mm
- Désaxage (piston-axe) : $Cp=0$ mm
- donné de profil : $h=1$
- vitesse transversale : $\dot{e}_h(\bar{t}_i) = 0$, $\dot{e}_b(\bar{t}_i) = 0$
- excentricités initiales : $\varepsilon_h = 0.1$, $\varepsilon_b = 0.11$



a) 1^{ère} itération, $\bar{t} \in [0 \ 4\pi]$



b) 2^{ème} itération $\bar{t} + 4\pi$



c) 3^{ème} itération $\bar{t} + 8\pi$

Figure 6.15 : Progression de la convergence de la solution dans le cas où $Cp=0$ mm

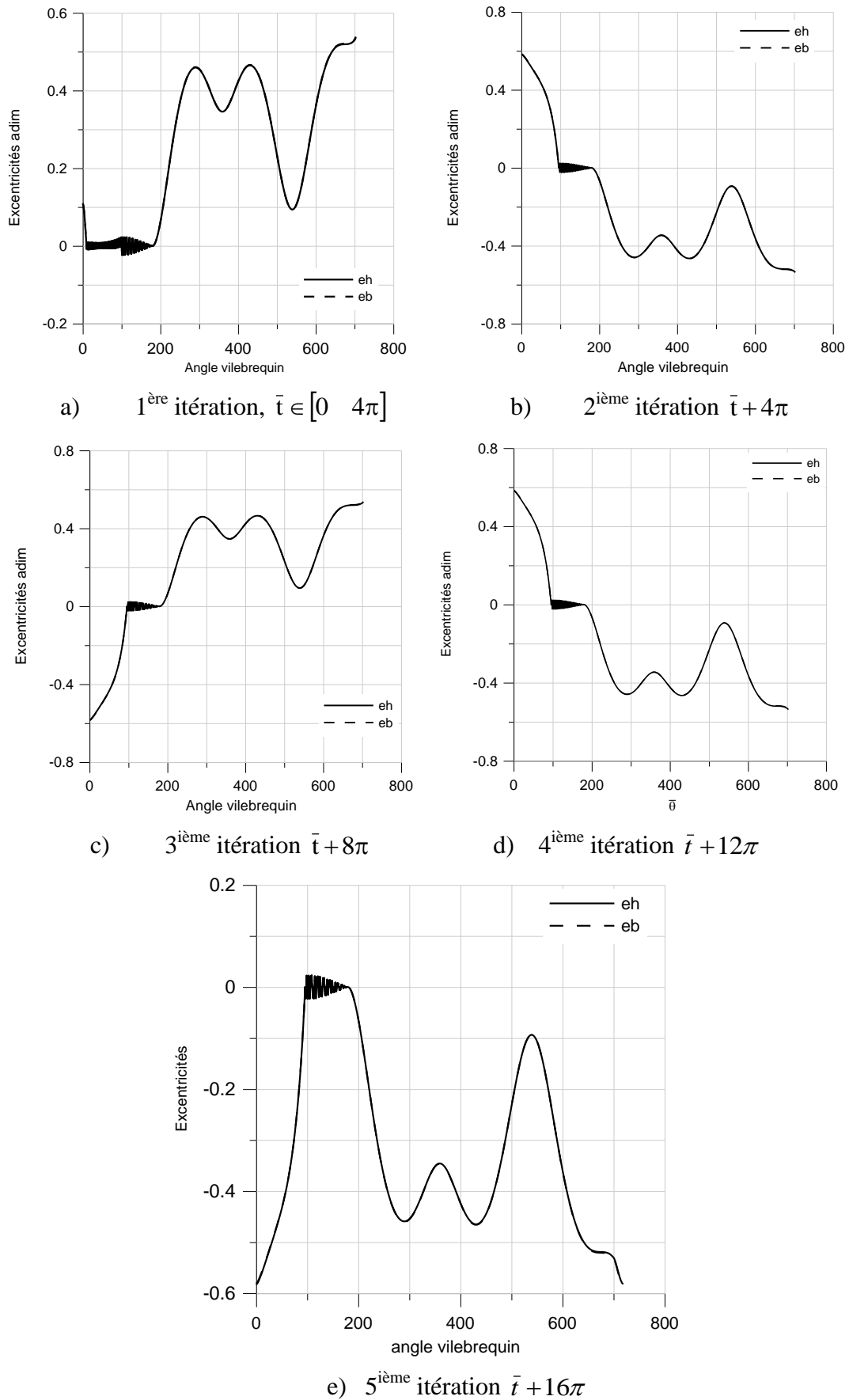


Figure 6.16 : Progression de la convergence de la solution dans le cas où $C_p=0\text{mm}$, [10]

c) *Résolution sans inertie* : Cette figure montre la pression des gaz. L'origine de cette courbe est prise au PMH, début de la phase de détente.

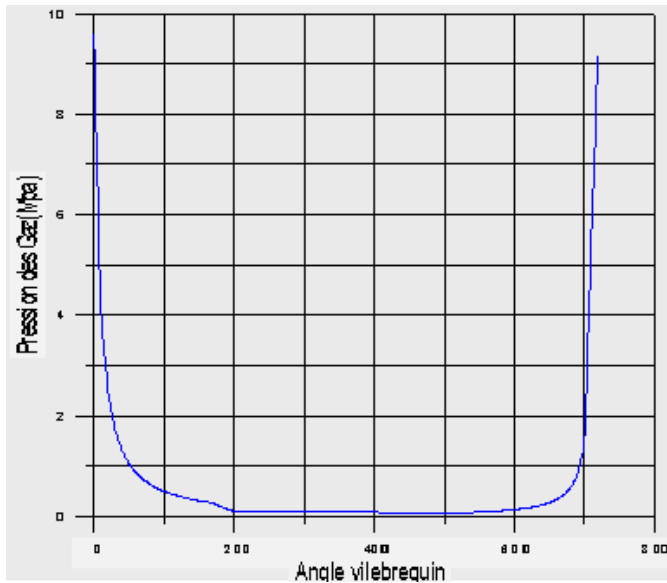
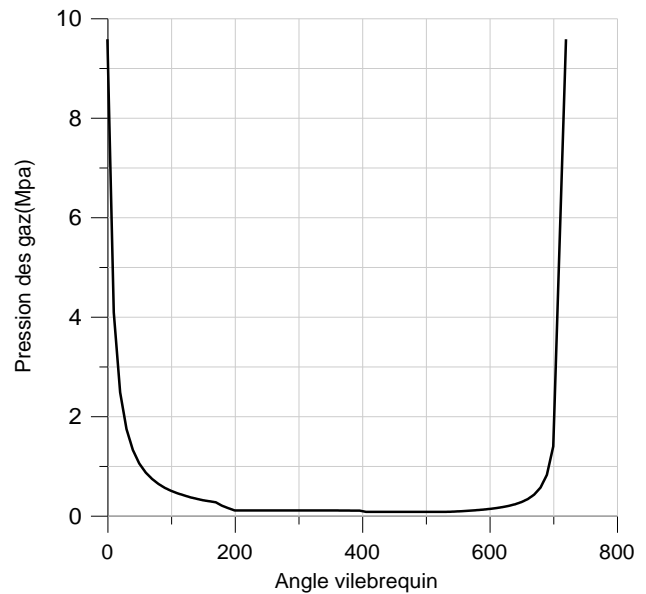


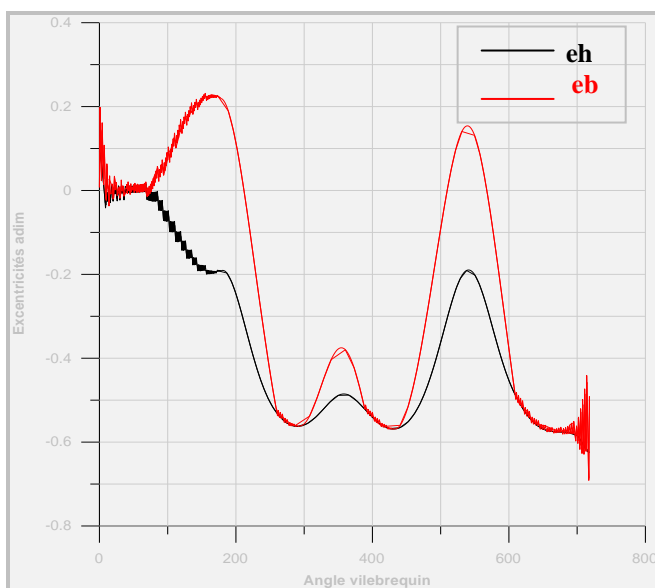
Figure 6.17 : a) Pression des gaz



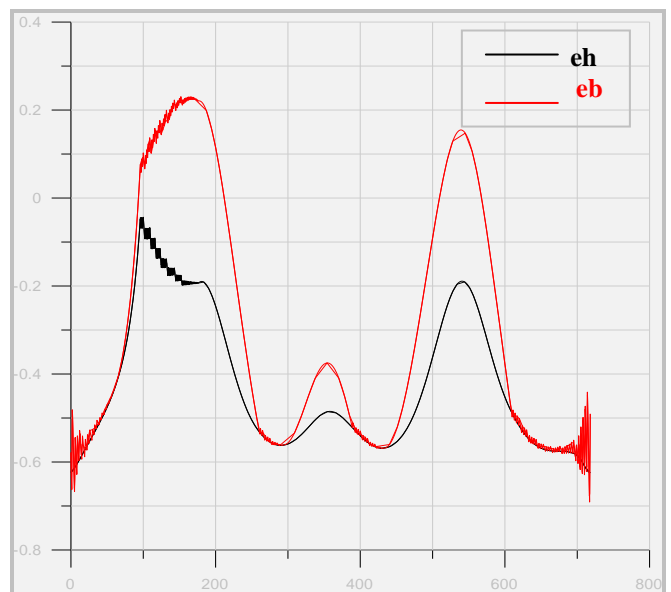
b) Pression des gaz, [10]

Pour obtenir les résultats suivants, on à :

- Longueur de la jupe du piston : $L=90$ mm
- Distance de l'axe au haut du piston : $a=37$ mm
- Désaxage (piston-axe) : $Cp=1.5$ mm
- donné de profil : $h=0.5$
- vitesse transversale : $\dot{e}_h(\bar{t}_i) = 0$, $\dot{e}_b(\bar{t}_i) = 0$
- excentricités initiales : $\varepsilon_h = 0.1$, $\varepsilon_b = 0.11$
- Jeu radial de l'ensemble piston-cylindre : $c=0.035$ m
-



a) 1^{ère} itération, $\bar{t} \in [0 \quad 4\pi]$



b) 2^{ème} itération $\bar{t} + 4\pi$

Figure 6.18 : Progression de la solution dans le cas où $Cp=1.5$ mm

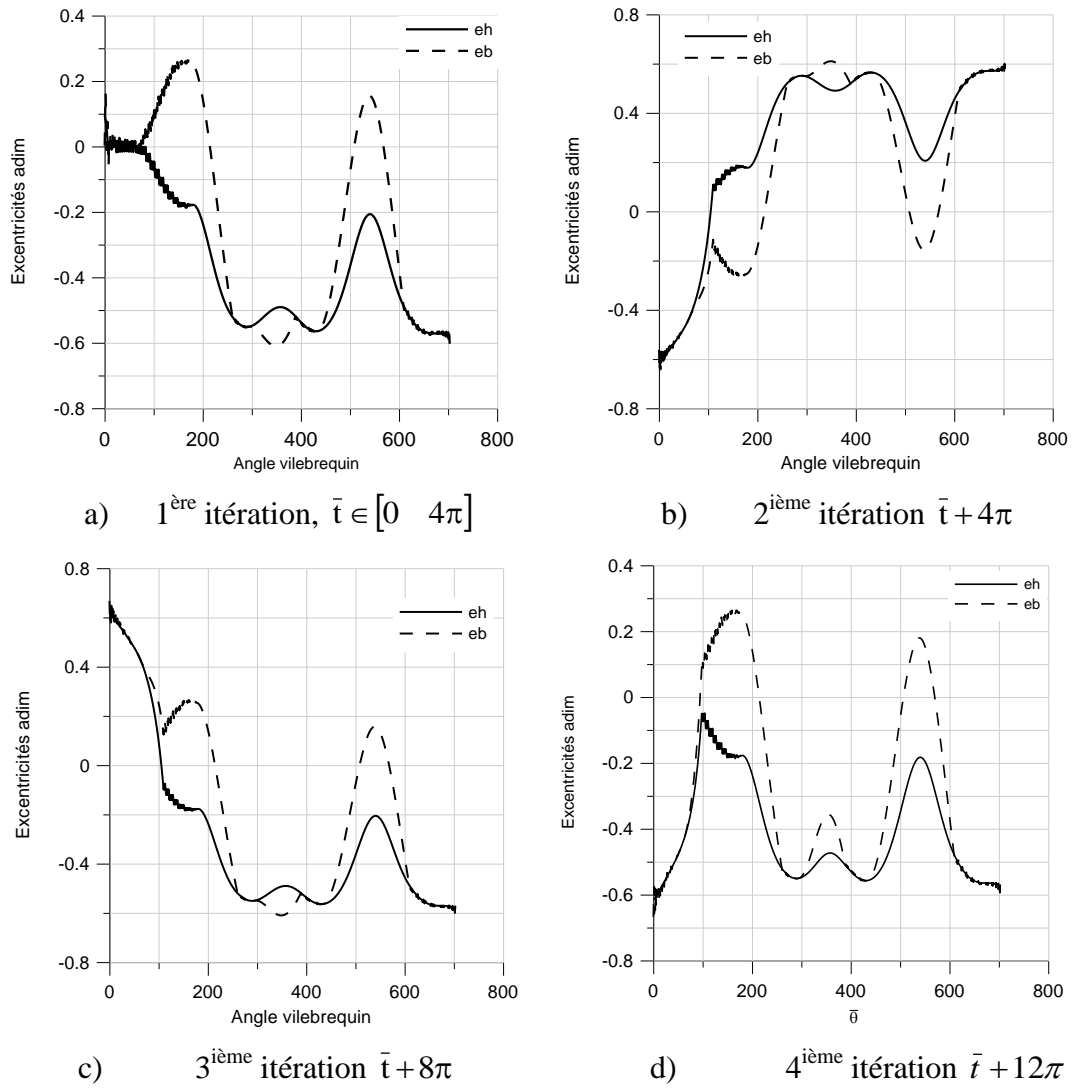


Figure 6.19 : Progression de la solution dans le cas où $C_p=1.5\text{mm}$, [10]

La cyclicité de la solution est obtenue au bout de l'itération 2. On a :

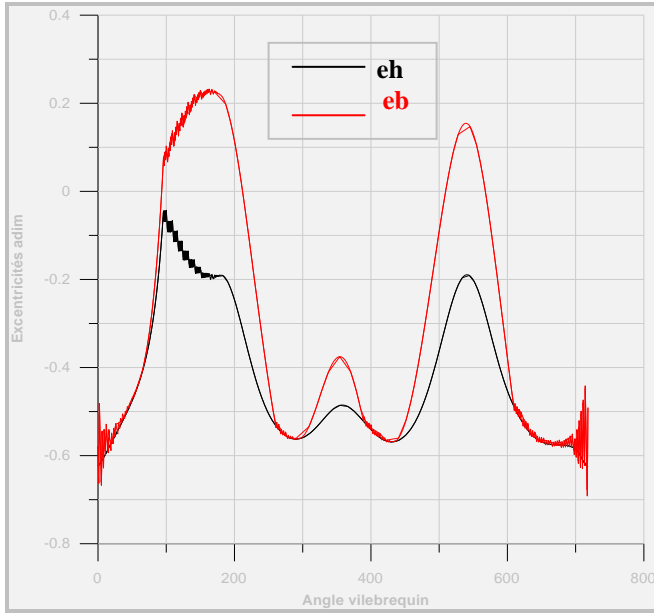
Solution au début du cycle :

$$\bar{\theta} = 0^\circ \quad \varepsilon_h = -0.62 \quad \varepsilon_b = -0.58$$

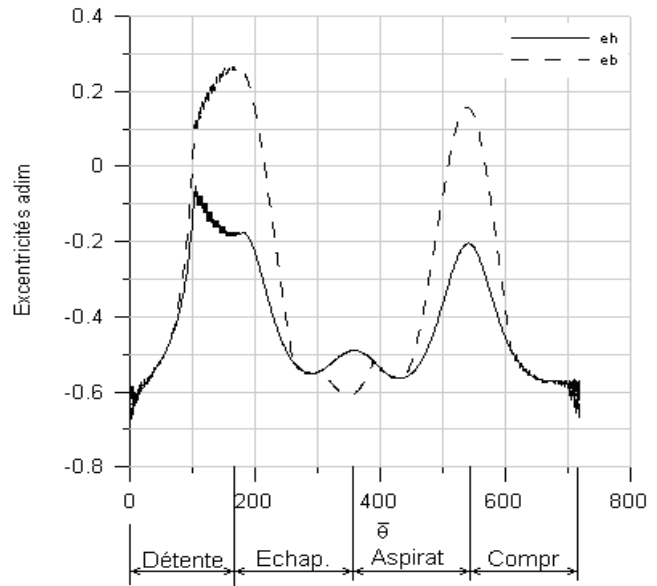
Solution en fin de cycle:

$$\bar{\theta} = 720^\circ \quad \varepsilon_h = -0.62 \quad \varepsilon_b = -0.57$$

d) Résolution avec inertie :

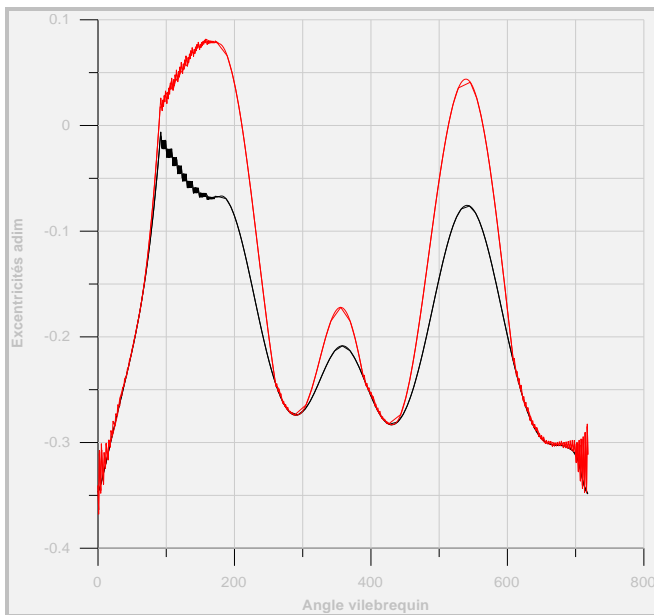


a1) résultats avec SMA

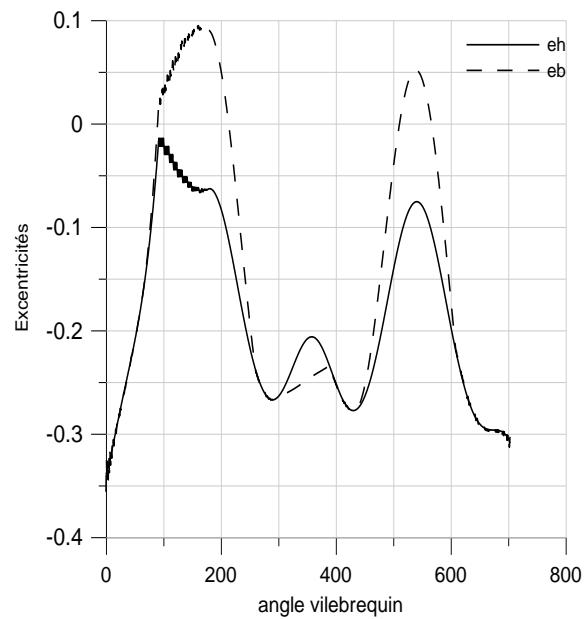


a2) résultats , [10]

Figure 6.20 : Mouvement secondaire avec prise en considération de l'inertie du piston ($c = 35\mu$)

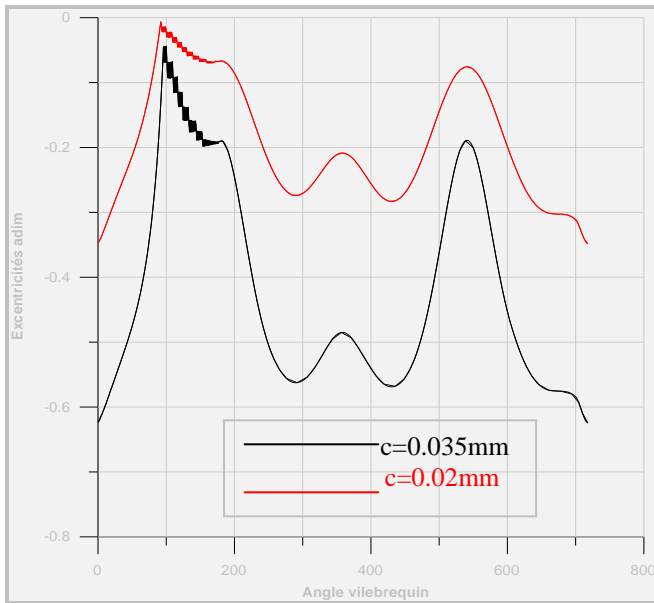


b1) résultats avec SMA

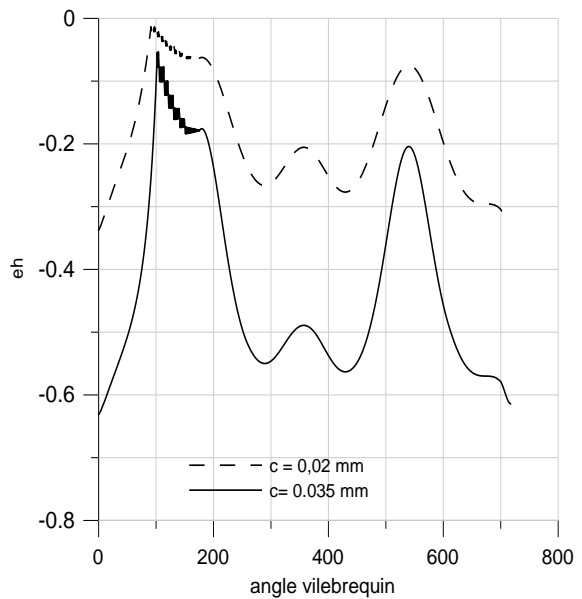


b2) résultats , [10]

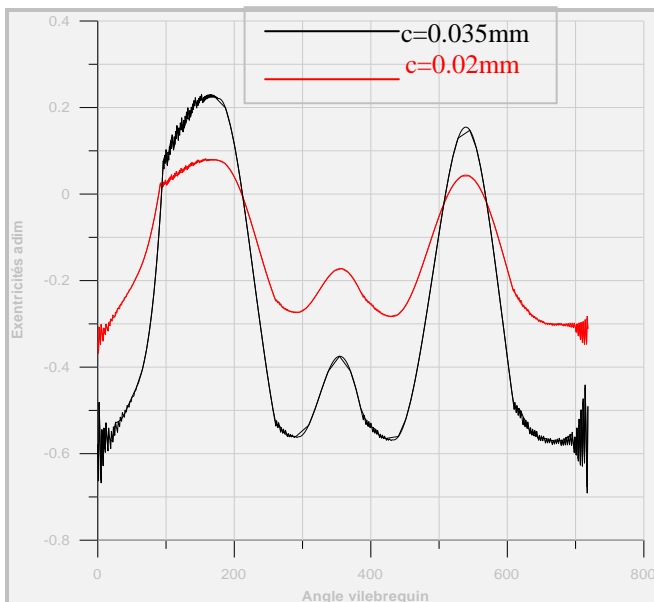
Figure 6.21 : Mouvement secondaire avec prise en considération de l'inertie du piston ($c = 20\mu$)



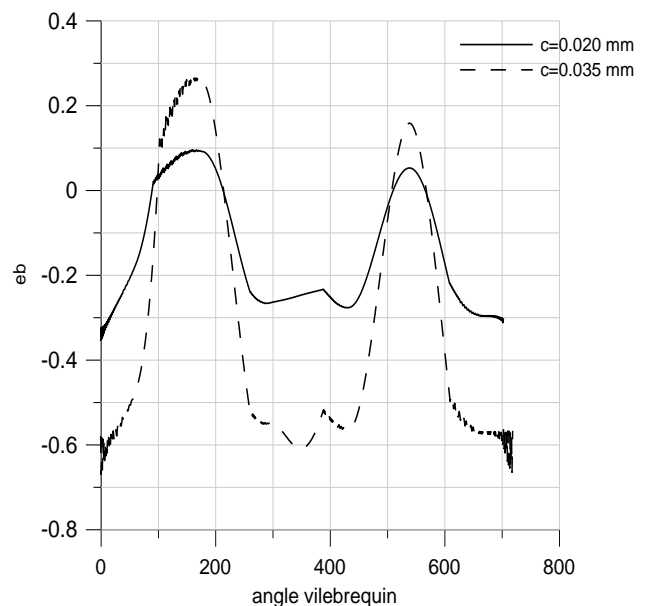
c1) résultats avec SMA



c2) résultats , [10]



d1) résultats avec SMA



d2) résultats , [10]

Figure 6.22 : Effet du jeu radial sur le mouvement secondaire.

6.8 Résultats et discussion

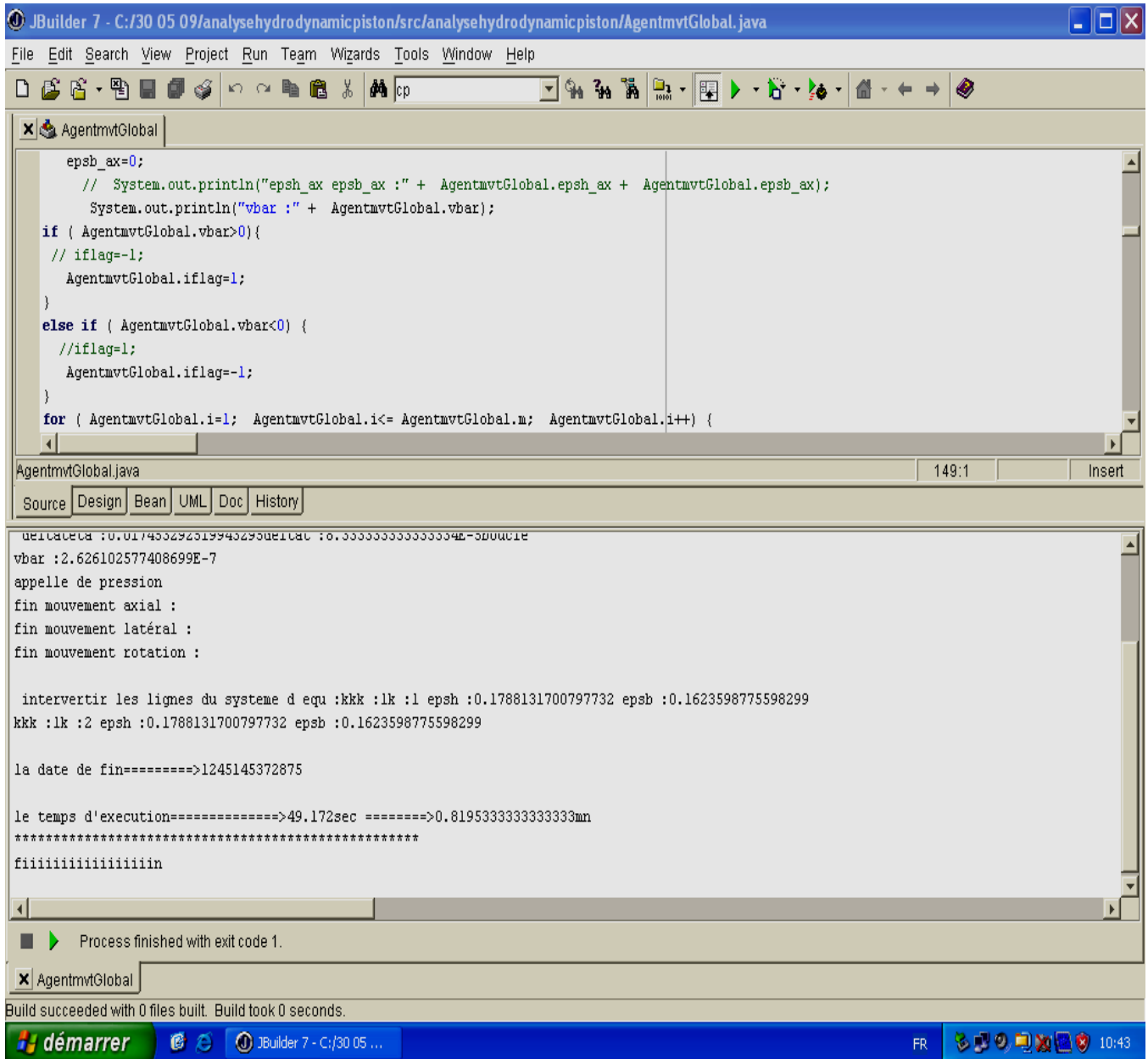
Notre exécution a été faite sous la configuration suivante :

Des ordinateurs Pentium 4 : Processeur 3.60 GHz, 3.60 GHz et 512 Mo de RAM.

On a fait trois types de simulation pour comparer les différents résultats obtenus.

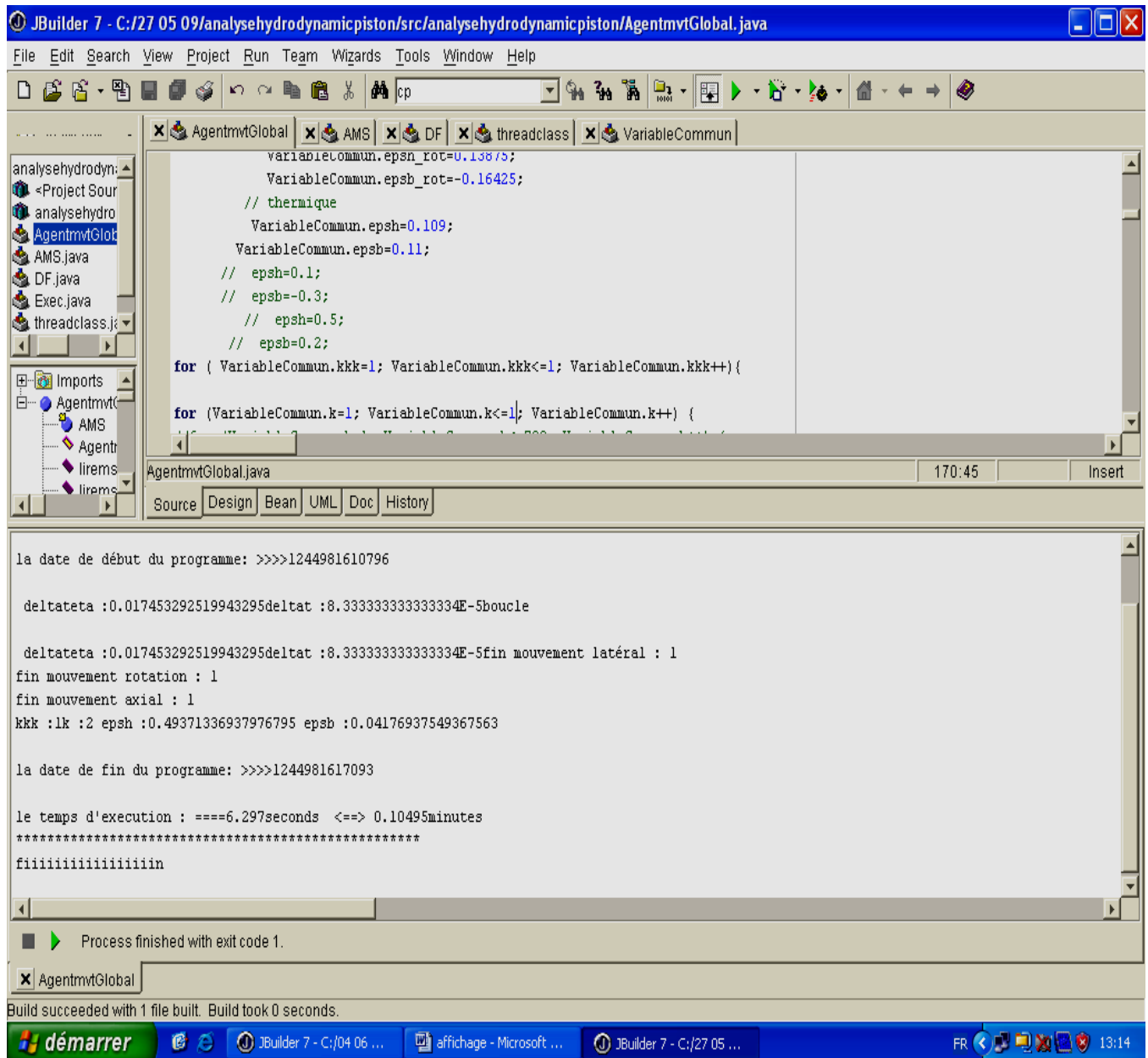
Le langage de programmation des trois (03) simulations est Java.

1- Simulation 1 : c'est une exécution du programme classique où le traitement de calcul des mouvements se fait séquentiellement, l'exemple d'application suivant se fait pour une seule itération $k=1$.



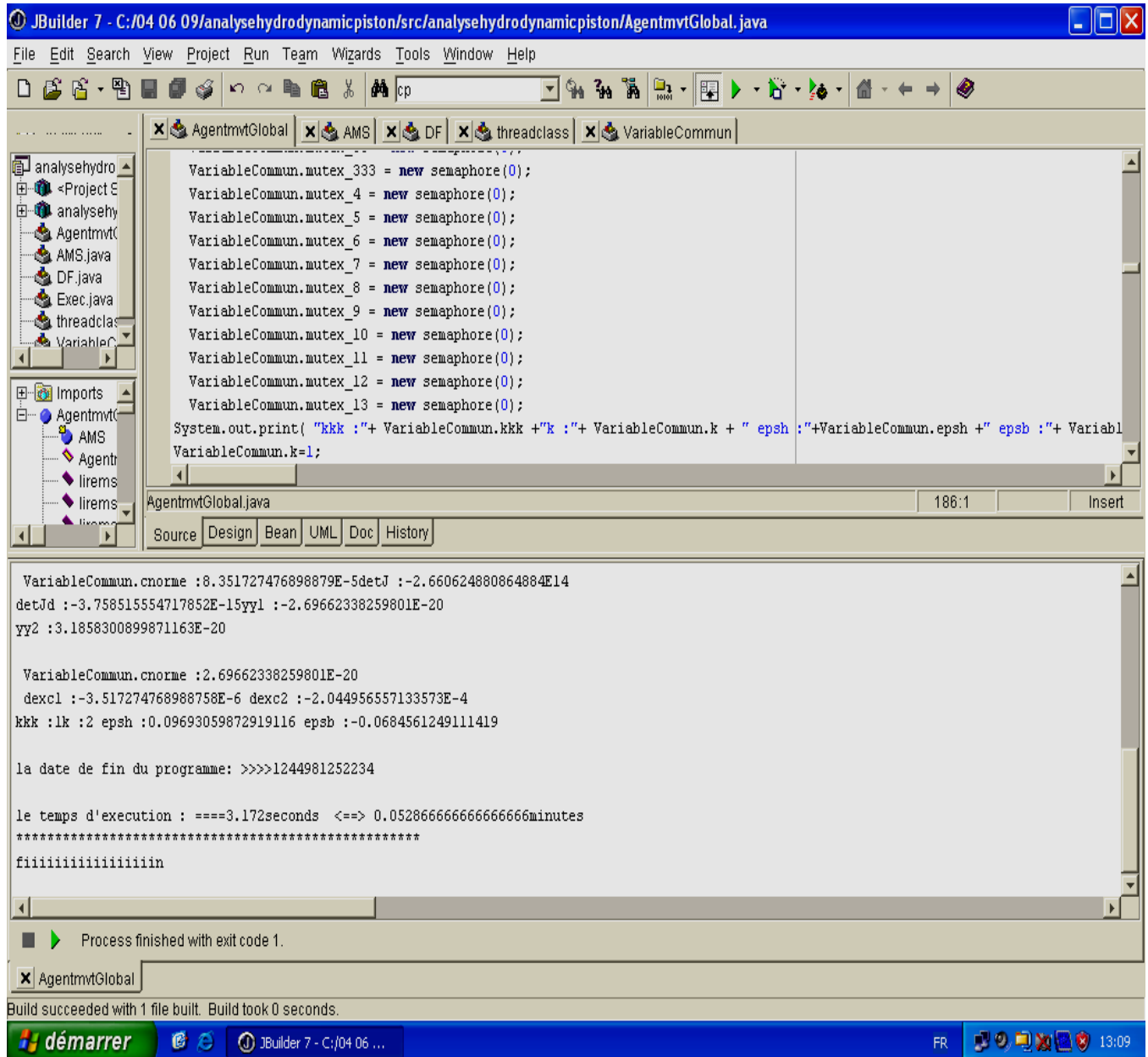
On distingue que le temps d'exécution d'après la console de cette simulation pour un $k=1$, est égal à : 49.172 seconds.

2- **Simulation 2** : c'est une exécution du programme centralisé c'est à dire programme qui utilise le système multi agents dont en installant tous les agents dans un seul ordinateur, l'exemple d'application suivant se fait aussi pour une seule itération $k=1$.



On distingue que le temps d'exécution d'après la console de cette simulation pour un $k=1$, est égal à : 6.297 seconds.

3- **Simulation 3** : c'est une exécution du programme distribué, c'est à dire un programme qui utilise le système multi agents dont en installant les applications sous un réseau de 4 ordinateurs où chaque agent est installé dans un ordinateur, l'exemple d'application suivant se fait aussi pour une seule itération k=1.



On distingue que le temps d'exécution d'après la console de cette simulation pour un k=1, est égal à : 3.172 seconds.

D'après ces différents résultats de simulation on arrive à les interpréter dans les graphes suivants :

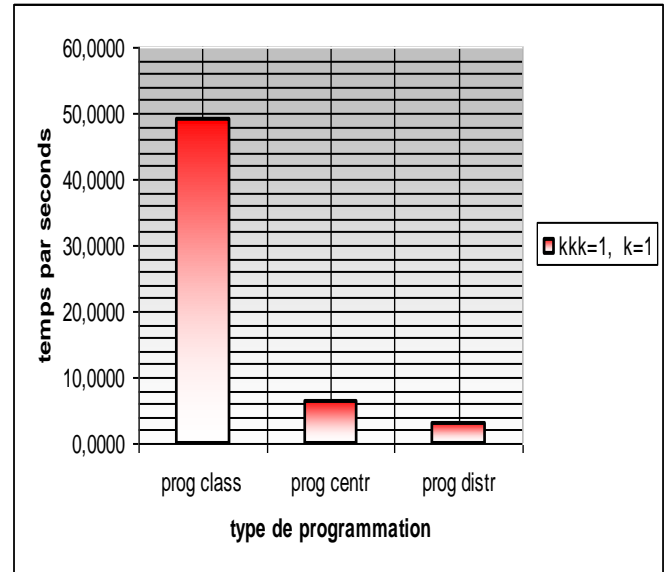
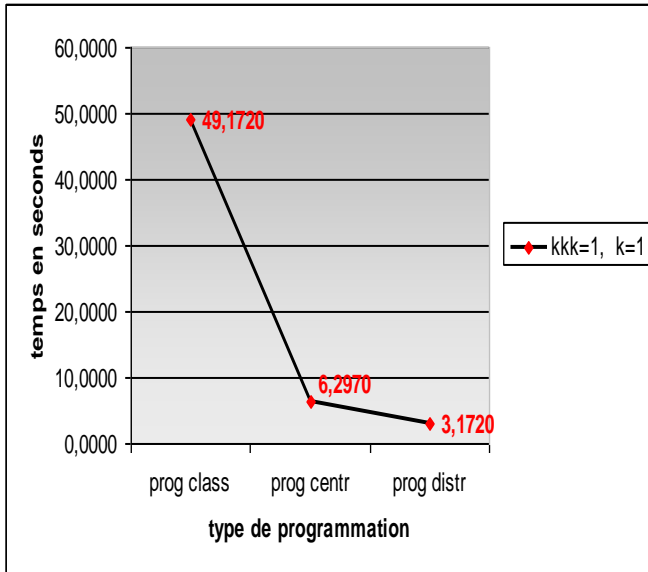


Figure 6.23 : courbe du résultats du temps/type programmation, k=1 Figure 6.24 : Histogramme du temps d'exécution, k=1

D'après la courbe Figure 6.23 et l'histogramme Figure 6.24 on peut dire que le temps d'exécution le plus élevé est celui de la programmation classique, et celui de la centralisée est moins élevée, tandis que la programmation distribuée son temps est le moins élevé c'est-à-dire fait le temps le plus court. Donc la meilleure simulation est celle qui utilise la programmation distribué c'est à dire les SMA fait améliorer le temps d'exécution pour une seule itération, en prenant en compte qu'il y à 12 messages échangés entre les agents.

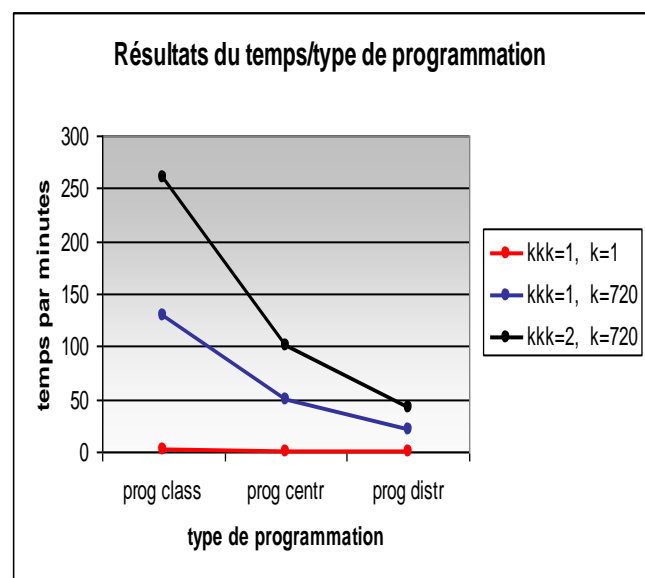
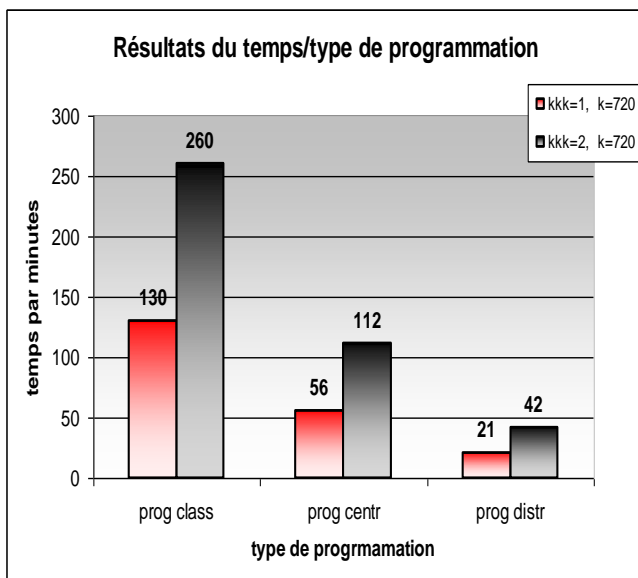


Figure 6.25 : courbe du résultats du temps/type programmation, k=720 Figure 6.26 : Histogramme du temps d'exécution, k=720

D'après cet histogramme Figure 6.25 et cette courbe Figure 6.26 on distingue que tant qu'en amplifiant le nombre d'ordinateurs c'est à dire utilisé la programmation distribuée au lieu du centralisée, le temps d'exécution diminue d'une façon remarquable toute en augmentant le nombre d'itération et sachant que le nombre de messages échangés augmente d'une façon remarquable aussi.

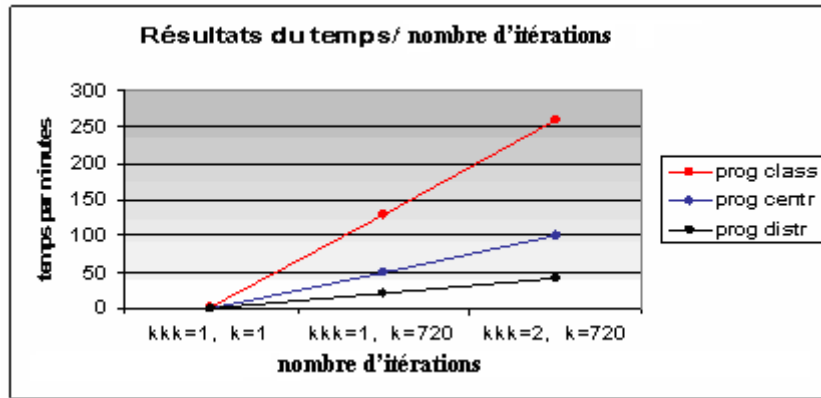


Figure 6.27 : courbe des résultats de temps d'exécution/ nombre d'itérations

D'après cette courbe on distingue que tant que le nombre d'itération reste à augmenter, le temps d'exécution augmente pour chaque type de programmation, mais on observe que la programmation distribuée donne toujours le meilleur temps d'exécution en rapidité.

6.9 L'apport de la simulation des Systèmes Multi Agents au modèle mécanique

D'après ces différents simulations on peut résumer l'apport du système multi agents au modèle mécanique comme suit :

- Le gain de temps

Avec un ordinateur traditionnel à un processeur unique, une tâche donnée est divisée en multiples opérations qui sont effectuées l'une après l'autre : c'est le traitement séquentiel. Notre système multi agents permet en revanche d'effectuer plusieurs opérations en même temps. On parle de calcul parallèle ou distribué. Comment ? Grâce à la présence de plusieurs processeurs dans un ensemble de machines « **réseau** ». On accède ainsi à une puissance de calcul et un gain de vitesse d'exécution considérables par rapport au traitement séquentiel. Cependant, le gain de temps apporté n'est pas toujours proportionnel au nombre de processeurs, Mais dépend aussi de la proportion d'activités parallélisables. Le challenge pour les informaticiens et mathématiciens est donc de paralléliser la plus grande part possible des traitements. D'où le développement de nouveaux algorithmes, de nouvelles formes de codage parallèle...etc. Plus la tâche est partitionnée en tâches indépendantes et plus elle peut être simultanément répartie entre les différents processeurs.

- La tolérance aux pannes

Dans le cas centralisé, si la machine tombe en panne ou se déconnecte pour une raison ou une autre, tout le travail (calcul) peut être perdue. Ce qui peut se refléter sur d'éventuelles pertes de données, voire perte de temps de calcul. En utilisant un système multi agents, on peut réduire le coût d'un projet. Ceci est possible par l'utilisation des ressources inutilisées disponibles sur le réseau.

Le premier type sera utilisé pour distribuer des applications collaboratives sur les différents ordinateurs. Il peut influencer directement les temps de calcul, la fiabilité du réseau et la robustesse du système. Prenant en considération les caractéristiques suivantes :

- Les ressources matérielles (charge du processeur libre, mémoire libre, espace disque libre).
- Bande passante élevée.
- Le temps de propagation en boucle.
- Disponibilité et durée de fonctionnement des machines.

Le deuxième type sera utilisé pour donner plus de fiabilité. A chaque pas de temps prédéfini, recopié dans des machines. Cette redondance des données peut améliorer la fiabilité du système. Les caractéristiques les plus importantes à prendre en compte sont :

- Bande passante élevée.
- Ressources matérielles (espace disque libre).
- Disponibilité et durée de fonctionnement des machines.

- La charge de machine

La répartition de la charge sur différentes machines est un avantage énorme par rapport au cas centralisé où l'ordinateur devient rapidement un goulot d'étranglement.

La charge d'une machine dépend des différents types de ressources locales pouvant être utilisées par un agent. Elle focalise sur deux, le processeur et la mémoire vive.

Alors nous constatons que dans le cas centralisé l'occupation de mémoire est plus élevée et le processeur est surchargé par rapport au cas distribué. Ce qui permet de dégrader les performances de la machine. Donc les SMA vient pour palier cet inconvénient et en évitant améliore la rapidité de calcul.

Conclusion Générale

L'hydrodynamique d'un piston de moteur à combustion a été résolue grâce à la simulation par des systèmes multi agents (SMA).

L'introduction des agents SMA constitue une originalité dans le domaine de simulation hydrodynamique d'un piston. Les passerelles mécaniques-informatiques ont pu avoir lieu grâce à la maîtrise multidisciplinaire des facteurs intervenant dans la résolution du modèle principalement les agents informatiques, le langage java, les threads, le réseau et l'adaptation des algorithmes du modèle mécanique vers le modèle informatique principalement.

L'objectif des SMA est dans une première étape le développement d'une alternative aux codes classiques par l'utilisation de réseau où chaque module mécanique est pris en charge par un PC du réseau.

Cette façon de faire constitue un apport très appréciable sur le contrôle du module mécanique séparé par l'utilisateur une fois la programmation développée. Ce qui entraîne une vérification plus simple en intervenant sur le PC spécifique.

La première partie est résolue à l'aide d'un code classique écrit en Java. Le mouvement est décomposé en trois mouvements simples : axial, latéral et de rotation. On a pu obtenir par superposition des trois mouvements les résultats du modèle global, principalement le champ de pression hydrodynamique et la trajectoire du piston.

Dans la deuxième partie on a utilisé les agents SMA pour chacun des mouvements. Les résultats obtenus sont les mêmes que ceux du modèle mécanique.

Ainsi un problème aussi complexe que celui de l'hydrodynamique du piston vient d'être résolu par cette nouvelle méthode. La méthode pourrait s'étendre à d'autres applications.

A l'issue de ces résultats, il s'avère que l'application des multi agents apporte les avantages suivants sur les codes classiques :

- 1°) Fiabilité : Etant donné que de nombreux résultats d'analyse par éléments finis ne sont pas vérifiés par des essais, et qu'ils sont souvent utilisés directement dans la conception d'un produit, la fiabilité du système d'analyse doit être absolument garantie. De plus, les modèles utilisés deviennent de plus en plus complexes, et les résultats doivent être produits de plus en plus vite. C'est pourquoi, la détection des erreurs de modélisation et leur élimination constituent une caractéristique essentielle du système multi agents.
- 2°) La robustesse de la méthode qui permet de prendre en charge des problèmes complexes et à large champ d'application.
- 3°) Le gain de temps relatif à la configuration du réseau utilisé.
- 4°) La tolérance aux pannes : application des techniques de fiabilisation pour les SMA reposent sur l'existence de services redondants et la re-exécution.
- 5°) L'utilisation du réseau réduit la charge (occupation de mémoire, processeur) sur chacun des PC ce qui améliore la vitesse et le temps d'exécution, les performances d'un pc d'une façon générale.

Concluant que l'un des objectifs de la simulation multi agents n'est pas de coller expressément à la réalité mais d'étudier comment à partir d'interactions simples on peut observer l'émergence d'un comportement global du système.

L'utilisation des systèmes multi agents ne modifie pas la séquence des opérations numériques. Ainsi les résultats d'analyses du même modèle sur la même machine, obtenus en exécution séquentielle et parallèle, sont rigoureusement identiques (si tous les autres paramètres restent inchangés).

La simulation multi agents est un besoin essentiel d'une meilleure qualité de service.

Perspectives

- **L'utilisation de la plate forme JADE :** (*Java Agent DEvelopment Framework*) est une plateforme java de développement et d'exécution de Système Multi Agents. Elle repose donc sur une communication indirecte par envois de messages asynchrones, facilite aussi la création d'un agent simple, elle est évolutive car on peut commencer par quelques agents simples puis faire évoluer l'architecture avec l'application, permettant aussi l'utilisation avancée de ACL.
- **L'utilisation des agents mobiles :** Une application construite à base d'agents mobiles peut se redéployer dynamiquement suivant un plan pré-établi ou en réaction à une situation particulière, afin par exemple d'améliorer la performance ou de satisfaire la tolérance aux pannes.
- **L'utilisation des grilles de calcul :** qui permet de faire du calcul distribué car elle exploite la puissance de calcul (processeurs, mémoires, ...) de milliers d'ordinateurs afin de donner l'illusion d'un ordinateur virtuel très puissant. Ce modèle permet de résoudre d'importants problèmes de calcul nécessitant des temps d'exécutions très longs en environnement "classique".

En dehors de son utilité dans le domaine scientifique, la grille de calcul aura sans aucun doute des retombées industrielles et commerciales : industriels impliqués dans le développement des logiciels, entreprises ayant des besoins de calcul importants, fourniture à la demande de ressources informatiques à des sociétés...

Références bibliographiques

[1]	Burrell, G. and Butler, F. G. , 'A Study in Applied Physics: Locating the Piston Pin to Minimise Piston Slap', General Motor Engineering Journal,38, November -December, pp 38-43, 1953.
[2]	Laws, A. M., Parker, D. A. and Turner, B. , 'Piston Movement in Diesel Engine', 10th International Congress on Combustion Engines, CIMAC, Washington, D.C., paper n°33, pp. 5-9, 1973.
[3]	Forbes, J. E. and Taylors , E. S. , 'A Method For Studying Piston Friction', NACA Wartime Report W-37, 1943.
[4]	Furuhama, S. and Takiguchi M. , 'Mesurement of Piston Friction Force in Actuel Operating Diesel Engine', SAE paper 790855, 1979.
[5]	Furuhama S. Takiguchi M. and Tomizawa K. , 'Effect of Piston and Piston Ring Design on the Piston Friction Forces in Diesel Engine', SAE paper 810977, 1981.
[6]	Furuhama S., Asahi C., Hiruma M. , 'Measurement of Piston Ring Oil Film Thikness in an Operating Engine', ASLE Trans., Vol. 26, pp. 325-332, 1982.
[7]	Takiguchi M., Machida K., Furuhama S. , 'Piston Friction Force of a Small High Speed Gosoline Engine', Journal Of Tribology, Vol. 110, pp. 112-118, Jan. 1988,
[8]	Uras H. M., and Paterson D. J. , 'Measurement of Piston and Ring Assembly Friction: Instantaneous IMEP Method', SAE Paper 8304416, 1983.
[9]	Hosch M. and Baba Y. 'A Study of piston friction force in an internal combustion engine, ASLE Transaction, 1986.
[10]	Tahar Abbes M., thèse 'Etude du piston de moteur à combustion interne sous chargement dynamique, 2004.
[11]	Cameron A. , 'The principles of Lubrication', Longmans, 1966.
[12]	Frene J. , 'La Lubrification hydrodynamique', bulletin de l'Union des Physiciens n° 689, pp.1531-1560, 1986.
[13]	Frene J., Nicolas D., Degueurce B., Berthe D. et Godet M. , 'Lubrification hydrodynamique : paliers et butées', Collection de la direction des études et Recherche d'électricité de France, n° 72, Eyrolles, 1990.
[14]	Fantino B., Frene J. et Godet M. 'Condition d'utilisation de l'équation de Reynolds rn mécanique des films minces visqueux', C.R.Ac.Sc. Paris t. 272, pp. 691-693, Mars 1971.
[15]	Hoult D. P. , 'Lubrication and Support of a Single Piston Ring', ASLE Trans. Vol. 28, pp. 139-149, 1984
[16]	Miltsios G. K., Patterson D.J., Papanastasiou T.C. , 'Solution of the Lubrication Problem and Calculation of the Friction Force on the Piston Rings', J. of Tribology Vol. 111, pp.635-640, 1989.
[17]	Jeng Y.R. , 'Theoretical Analysis of Piston-Ring Lubrication, Part I-Fully Flooded Lubrication' Tribolgy Trans. Vol. 35, pp. 696-706, 1992.

[18]	Jeng Y.R. , 'Theoretical Analysis of Piston-Ring Lubrication, Part II-Starved Lubrication and its Application to a complete Ring Pack', Tribology Trans. Vol. 35, pp. 707-714, 1992.
[19]	Michail S.K., Barber G. C. , 'The Effect of Roughness on Piston Ring Lubrication, Part I: Model Development', Tribology Trans. Vol.38, pp. 19-26, 1992.
[20]	Ma A T., Smith E.H., Sherrington I. , 'A three-Dimensional of Piston Ring Lubrication Part 1: Modelling', Proc. Instn. Mech. Engrs. Vol. 209, 1995.
[21]	Yang Q., Theo G., Keith J.R. , 'Two-Dimensional Piston Ring Lubrication-Part I: Rigid Ring and Liner Solution', Tribology Trans., Vol. 39, pp.757-768, 1996.
[22]	Yang Q., Theo G., Keith J.R. , 'Two-Dimensional Piston Ring Lubrication-Part II: Elastic Ring Consideration', Tribology Trans., Vol. 39, pp.870-880, 1996.
[23]	Ma A T., Sherrington I and Smith E.H. , 'Analysis of Lubrication and Friction for a Complete Piston-Ring Pack with an Improved Oil Availability Model, Part1: Circumferentially Uniform Film', Proc. Instn. Mech. Engrs. Vol. 211 Part J, 1997.
[24]	Shaws, M. C. and Nussdorfer, T. 'A Visual and Photographic study of Cylindre Lubrication', National Advisory Committe for Aeronautics, Report N° 850, pp 247-265, 1945.
[25]	Green A. B. , 'Initial Visual Studies of Piston-Cylinder Dynamic Oil Film Behavior', Wear, 13 pp 345-569, 1969.
[26]	Laws, A. M., Parker, D. A. and Turner, B. , 'Piston Movement in Diesel Engine', 10th International Congress on Combustion Engines, CIMAC, Washington, D.C., paper n°33, pp. 5-9, 1973.
[27]	Knoll G. D. and Peekin H. J. , 'Hydrodynamic lubrication of piston skirt', Trans of the ASME, Vol.104, (oct. 1982), pp.504-509.
[28]	Li D. F., Rhode S. M. and Ezzat H. A. , 'An automotive piston lubrication model', ASLE Trans. vol. 26, (feb. 1983), pp. 151-160.
[29]	Mourelatos Z. P. , 'Gas Lubrication of a Ringless Piston in an Internal Combustion Engine Under Dynamic Loading', ASME JOURNAL OF TRIBOLOGY, Vol. 110 Oct. 1988.
[30]	Gommed K. and Etsion I. "Dynamic Analysis of Gas Lubricated Reciprocating Ringless Pistons- Basic Modeling", Journal Of Tribology, Vol.115, pp. 207-213, Apr. 1993.
[31]	Liu K.,Xie Y.B. and Gui C. L. , 'A comprehensive Study of the Friction and Dynamic motion of the Piston Assembly', Proc. Instn. Mech. Engrs., Vol. 212 Part J, pp.221-226, 1998.
[32]	Siyoul J. , 'Analysis of the effects of Bore Clearance Due to Skirt Profile Change on the Piston Secondary Movments', KSTLE International Journal, Vol. 3, N°2, Dec. 2002, pp. 00-00.
[33]	J. Ferber « Les Systèmes Multi-Agents : Vers une intelligence collective », InterEdition, 1995.
[34]	Nicholas Jennings, Katia Sycara, Michael Wooldridge: A Roadmap of Agent Research and Development. Dans les actes de <i>International Conference on Autonomous Agents</i> (AA), Minneapolis, MN, USA, Mai 1998. ACM

[35]	Y. Shoham « Agent-oriented programming », <i>Artificial Life</i> 60, 1993, pp 139..159.
[36]	Yannis Labrou, Tim Finin: A Semantics Approach for Kqml - A General Purpose Communication Language for Software Agents. Dans les actes de <i>Conference on Information and Knowledge Management (CIKM)</i> , Gaithersburg, MD, USA, Décembre 1993.
[37]	Mihai Barbuceanu, Mark Fox: COOL: A Language for Describing Coordination in Multi Agent Systems. Dans les actes de <i>International Conference on Multi-Agent Systems (ICMAS)</i> , San Francisco, CA, USA, Juin 1995. V. Lesser, L. Gasser (éd.), AAAI Press
[38]	John Austin: <i>How to Do Things with Words</i> . Oxford Press, 1962, Londres, Angleterre
[39]	John Searle: <i>Speech Acts: an Essay in the Philosophy of Language</i> . Cambridge Press, 1969
[40]	Philip Cohen, Hector Levesque: Communicative Actions for Artificial Agents. Dans <i>International Conference on Multi-Agent Systems (ICMAS)</i> , (), Juin 1995. AAAI Press
[41]	FIPA97 Specification Part 2 Agent Communication, Issue 1 (voir http://www.iit.upco.es/~luisf/programacion/fipa/Spec.www/fipa97/fipa8a22.doc)
[42]	Jennings, N. R. On agent-based software engineering. <i>Artificial Intelligence</i> , Vol. 117, No.2, 2000, p.277-296.
[43]	Smith, R. G. “The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver”. Dans <i>IEEE Transactions on Computers</i> , 29 (12): 1104-1113, Décembre 1980.
[44]	C. Iglesias, M. Garrijo, and J. Gonzalez. A survey of agent-oriented methodologies. In Jörg Müller, Munindar P. Singh, and Anand S. Rao, editors, <i>Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)</i> , volume 1555, pages 317–330. Springer-Verlag : Heidelberg, Germany, 1999.
[45]	M. Wooldridge and P. Ciancarini. Agent-Oriented Software Engineering : The State of the Art. In P. Ciancarini and M. Wooldridge, editors, <i>First Int. Workshop on Agent-Oriented Software Engineering</i> , volume Lecture Notes in AI Volume 1957, pages 1–28. Springer-Verlag, Berlin, 2001.
[46]	L. Cernuzzi, M. Cossentino, and F. Zambonelli. Process models for agent-based development. <i>Journal of Engineering Applications of Artificial Intelligence</i> , Elsevier. Vol.18(2) :205–222, March 2005.
[47]	P. FELBER, R. GUERRAOUI et A. SCHIPER. « The Implementation of a CORBA Object Group Service ». <i>Theory and Practive of Object Systems</i> , volume 4, numéro 2, pages 93–105, 1998.
[48]	J. Ferber, O. Gutknecht, “Aalaadin: A Meta-model for the Analysis and Design of Organizations in Multi-agent System”, ICMAS’98, 1998.