

REPUBLIQUE DU SENEGAL

UNIVERSITE CHEIKH ANTA DIOP

ECOLE POLYTECHNIQUE DE THIES

DEPARTEMENT DE GENIE ELECTROMECHANIQUE

Gm. 0183

# PROJET DE FIN D'ETUDES

en vue de l'obtention du Diplôme d'Ingénieur de Conception

ANALYSE DYNAMIQUE DES MECANISMES

ASSISTEE PAR ORDINATEUR

AUTEUR: Yvon Polycarpe DOSSA

DIRECTEUR: Mohamadou L. DIALLO Ing, MScA

DATE: JUILLET 92

## DÉDICACE

A tous ceux qui, par leur amour, ont été sur ma route le reflet  
de l'unique AMOUR;

à ma future Épouse parée de ses bijoux;

à l'humanité entière.

## REMERCIEMENTS

Nous tenons tout d'abord à remercier très sincèrement Monsieur Mohamadou Lamine DIALLO -directeur du projet- pour son attention particulière, pour son franc encadrement et pour la documentation mise à notre portée.

Notre profonde gratitude va également aux élèves ingénieurs Georges CHABI, Wilfred HOUETO et à Vladimir Serge OKEY pour la précieuse aide qu'ils nous ont apporté dans l'édition du texte.

Que tous ceux qui, bien qu'ayant voulu nous aider, ne l'ont pas pu faute de temps soient assurés de notre sincère reconnaissance.

Nous n'oublieront évidemment pas tous ceux qui dans l'ombre ont contribué par leurs prières -déjà exaucées- à l'aboutissement du présent travail.

## SOMMAIRE

L'objectif de ce projet de fin d'études, est de réaliser un outil informatique d'aide à l'analyse dynamique des mécanismes plans à quatre membrures. Le travail comprend deux volets.

Dans un premier temps, nous avons effectué l'étude théorique du sujet. Dans cette analyse dynamique des mécanismes à quatre membrures, l'analyse cinéto-statique tient une place importante. En lui faisant appel et à d'autres notions, nous avons déterminé:

- les forces aux joints et le couple d'entrée à partir des caractéristiques cinétiques;
- l'accélération de la membrure d'entrée connaissant le couple d'entrée: réponse temporelle;
- les actions à entreprendre pour réaliser l'équilibrage des mécanismes.

Notons que l'équilibrage des rotors a également fait l'objet de notre étude.

Une fois la théorie bien établie, il ne restait plus qu'à programmer. Ne prétendant aucunement établir une relation d'ordre entre les langages de programmation -débat très épineux-, nous remarquons simplement que "Pascal" a été suffisamment adapté à notre sujet.

Pour des raisons d'efficacité et de simplicité, l'analyse cinéto-statique a été faite par la méthode la méthode matricielle; la représentation complexe -des vecteurs- a été à l'honneur.

## TABLE DES MATIERES

	Page
REMERCIEMENTS.....	I
SOMMAIRE.....	II
INTRODUCTION.....	1
CHAPITRE 1 FORCES AUX JOINTS ET COUPLE D'ENTRÉE: analyse cinéto-statique.....	3
1.1 MÉCANISMES A QUATRE MEMBRURES DE DIMENSIONS FINIES...4	
1.1.1 Positions des membrures 3 et 4.....4	
1.1.2 Vitesses et accélérations angulaires.....6	
1.1.3 Vitesses et accélérations linéaires.....8	
1.1.4 Forces et Moments d'inertie; Forces aux joints et Couple d'entrée.....9	
1.2 MÉCANISMES A QUATRE MEMBRURES MANIVELLE-COULISSEAU..13	
1.2.1 Position angulaire de la membrures 3.....13	
1.2.2 Vitesses et accélérations.....14	
1.2.3 Forces et Moments d'inertie; Forces aux joints et Couple d'entrée.....16	
CHAPITRE 2 ANALYSE DE LA RÉPONSE TEMPORELLE.....	18
2.1 Mécanismes à quatre membrures de dimensions finies...19	
2.2 Mécanismes à quatre membrures: Manivelle-Coulisseau..25	
CHAPITRE 3 ÉQUILIBRAGE DES ROTORS.....	28
3.1 ÉQUILIBRAGE STATIQUE.....	28
3.2 ÉQUILIBRAGE DYNAMIQUE.....	30
CHAPITRE 4 ÉQUILIBRAGE DES MÉCANISMES A QUATRE MEMBRURES.....	34
4.1 ÉQUILIBRAGE DES FORCES.....	34
4.1.1 Mécanismes à quatre membrures de longueurs finies.....	34
4.1.2 Mécanismes à quatre membrures: Manivelle-coulisseau.....	38

4.2	ÉQUILIBRAGE DES MOMENTS.....	40
4.2.1	La membrure In-line.....	41
4.2.2	Le Pendule physique.....	42
4.2.2.1	Cas général: une quelconque membrure in-line.....	44
4.2.2.2	1 <sup>er</sup> cas particulier: barre rectangulaire.....	45
4.2.2.3	2 <sup>ème</sup> Cas particulier: Augmented link.....	46
4.2.3	Équilibrage d'un mécanisme dont le coupleur est une in-line et un pendule physique.....	47
	RECOMMANDATIONS.....	49
	CONCLUSION.....	50
	BIBLIOGRAPHIE.....	51
	ANNEXES.....	54
A-	Organigrammes des différents modules.....	55
A-1	Analyse cinéto-statique.....	56
A-2	Analyse de la réponse temporelle.....	57
A-3	Équilibrage dynamique des rotors.....	58
A-4	Équilibrage des forces des mécanismes à quatre membrures.....	59
A-5	Équilibrage complet (forces et moments) des mécanismes à quatre membrures.....	60
B-	Exemples d'exécution du logiciel.....	61
B-1	Exemple 1: mécanisme à quatre membrures de longueurs finies: analyse cinéto-statique et de la réponse temporelle.....	62
B-2	Exemple 2: mécanisme à quatre membrures manivelle-coulisseau: analyse cinéto-statique et équilibrage des forces.....	69
B-3	Exemple 3: mécanisme à quatre membrures de longueurs finies: équilibrage des forces et des moments.....	75
B-4	Exemple 4: équilibrage dynamique d'un rotor.....	81
C-	Listing du programme informatique.....	85

## LISTE DES FIGURES.

	page
Figure 1.1 Mécanisme à quatre membrures: positions angulaires des membrures intermédiaire et de sortie.....	4
Figure 1.2 Mécanisme à quatre membrures: calcul des vitesses.....	6
Figure 1.3 Calcul des forces aux joints: diagramme du corps libre de la membrure d'entrée.....	9
Figure 1.4 Calcul des forces aux joints: diagramme du corps libre de la membrure intermédiaire.....	10
Figure 1.5 Calcul des forces aux joints: diagramme du corps libre de la membrure de sortie.....	10
Figure 1.6 Calcul des forces aux joints: système d'équations.....	12
Figure 1.7 Mécanisme à quatre membrures Manivelle-coulisseau: position angulaire de la membrures intermédiaire.....	13
Figure 1.8 Mécanisme à quatre membrures Manivelle-coulisseau: calcul des vitesses.....	14
Figure 1.9 Calcul des forces aux joints: diagramme du corps du coulisseau.....	16
Figure 1.10 Calcul des forces aux joints du mécanisme à quatre membrures manivelle-coulisseau: système d'équations..	17
Figure 2.1 Analyse de la réponse temporelle: diagramme du corps libre de la membrure d'entrée.....	19
Figure 2.2 Analyse de la réponse temporelle: diagramme du corps libre de la membrure intermédiaire.....	20

Figure 2.3	Analyse de la réponse temporelle: diagramme du corps libre de la membrure de sortie.....	22
Figure 2.4	Système d'équations de l'analyse de la réponse temporelle d'un mécanisme à quatre membrures de longueurs finies.....	24
Figure 2.5	Analyse de la réponse temporelle: diagramme du corps libre du coulisseau.....	26
Figure 2.6	Système d'équations de l'analyse de la réponse temporelle d'un mécanisme à quatre membrures manivelle-coulisseau.....	27
Figure 3.1	Excentricités d'un rotor.....	29
Figure 3.2	Équilibrage statique des excentricités d'un rotor.....	29
Figure 3.3	Déséquilibre dynamique.....	30
Figure 3.4	Équilibrage dynamique: plans de correction.....	31
Figure 4.1	Équilibrage d'un mécanisme à quatre membrures de longueurs finies.....	35
Figure 4.2	Équilibrage d'un mécanisme à quatre membrures manivelle-coulisseau.....	38
Figure 4.3	Membrure in-line.....	41
Figure 4.4	Addition d'un contre-poids pour rendre une membrure in-line.....	41
Figure 4.5	Pendule simple.....	44
Figure 4.6	Pendule physique: cas général.....	44
Figure 4.7	Pendule physique: barre rectangulaire.....	45
Figure 4.8	Pendule physique: augmented link.....	46



Figure B-1	Exemple 1, mécanisme à quatre membrures de longueurs finies: analyse cinéto-statique.....	62
Figure B-2	Exemple 2, mécanisme manivelle-coulisseau: analyse cinéto-statique et équilibrage des forces.....	69
Figure B-3	Exemple 3, mécanisme à quatre membrures de longueurs finies: équilibrage complet.....	75
Figure B-4	Exemple 3, représentation des résultats de l'équilibrage complet du mécanisme.....	80
Figure B-5	Exemple 4, rotor: équilibrage dynamique.....	81
Figure B-6	Exemple 4, représentation des résultats de l'équilibrage dynamique du rotor.....	84

## INTRODUCTION

L'homme dans sa perpétuelle recherche des moyens de se faciliter le labeur n'a jamais été à court d'idées ni de réalisations. Ainsi les machines au début utilisées pour aider l'homme dans ses tâches les plus pénibles, envahissent de plus en plus notre vie de manière à faire aujourd'hui concurrence à l'homme. Il est tout de même incontestable que tout développement économique passe aujourd'hui par la maîtrise de la mécanique des machines.

La mécanique des machines peut être subdivisée en deux composantes: la "théorie des mécanismes" et la "théorie des machines". La théorie des machines étudie l'ensemble des mécanismes associés pour constituer la machine.

Quant à la théorie des mécanismes, elle étudie les propriétés des différents mécanismes usuels employés dans les machines sous toute forme. Elle étudie également la cinématique et la dynamique des mécanismes en fonction de leurs paramètres géométriques et des forces y agissant.

Deux genres de problèmes peuvent être traités par la théorie des mécanismes: la "synthèse" et l'"analyse". Cette dernière peut être subdivisée en deux parties:

- l'analyse structurale et cinématique et
- l'analyse dynamique.

L'analyse structurale et cinématique comme son nom l'indique, s'attache à faire une étude purement géométrique de l'organisation

de principe des mécanismes et des mouvements des corps qui forment ces mécanismes. On n'y tient pas compte des forces produisant le mouvement des corps.

L'analyse dynamique consiste en la détermination des forces et moments agissant sur les corps -formant le mécanisme- au cours de son mouvement, ainsi qu'en l'étude des relations existant entre les forces sollicitant les corps, les mouvements et les masses des corps. Ce domaine est l'objet de notre présent travail. Comme on le voit, il est assez vaste, nous nous efforceront de le couvrir dans la mesure du possible.

Nous ferons successivement:

- l'analyse cinéto-statique des mécanismes à quatre membrures en vue de la détermination des forces aux joints et du couple d'entrée;
- l'analyse de la réponse temporelle des mêmes mécanismes;
- l'étude de l'équilibrage des rotors ainsi que
- l'étude de l'équilibrage des mécanismes à quatre membrures.

Aventure assez laborieuse mais captivante dans laquelle nous allons nous lancer.

## CHAPITRE 1

### FORCES AUX JOINTS ET COUPLE D'ENTREE : analyse cinéto-statique

Un mécanisme fait l'objet de deux différentes sortes de sollicitations: les statiques et les dynamiques. Les sollicitations dynamiques comme leurs nom l'indique apparaissent uniquement lors du mouvement des mécanismes. Elles sont donc directement liées à la vitesse et à l'accélération des mécanismes.

Nous utiliseront l'analyse cinéto-statique dans le but de déterminer les charges de roulements (forces aux joints) dans une position donnée du mécanisme. Nous utiliseront le concept de l'équilibre dynamique<sup>(1)</sup> du mécanisme sous l'action des forces et des moments d'inertie. Le mécanisme est considéré comme en équilibre sous l'action de ces charges d'inertie qui sont en fait des charges cinétiques: d'où la désignation cinéto-statique. De nombreuses méthodes sont utilisées, mais nous estimons celle des nombres complexes comme étant la plus adaptées à la programmation. Pour atteindre cet objectif, nous procéderons comme suite:

- A partir des caractéristiques de la membrure d'entrée  
déterminer les positions, les accélérations linéaires et  
angulaires des deux autres membrures mobiles,

(1) En fait sous l'action des charges d'inertie et des charges extérieures, le corps est considéré comme en équilibre dit équilibre dynamique. Mais comme il s'agit ici du calcul de la seule contribution des charges d'inertie, nous faisons abstraction des charges externes.

- Déterminer les forces et moments d'inertie de toutes les membrures mobiles,
- Déterminer les charges aux différents joints.

Nous suivrons cette procédure pour chacun des deux cas suivants étudiés.

### 1.1- MÉCANISMES A QUATRE MEMBRURES DE DIMENSIONS FINIES

#### 1.1.1 Positions des membrures 3 et 4

Considérons un mécanisme à quatre membrures dont les dimensions sont connues.  $S, \beta, \delta, \Gamma$  et  $z$  sont définies comme montrer sur la figure. Pour un même angle d'entrée  $\theta_2$ , deux configurations distinctes sont possibles (fig1.1):

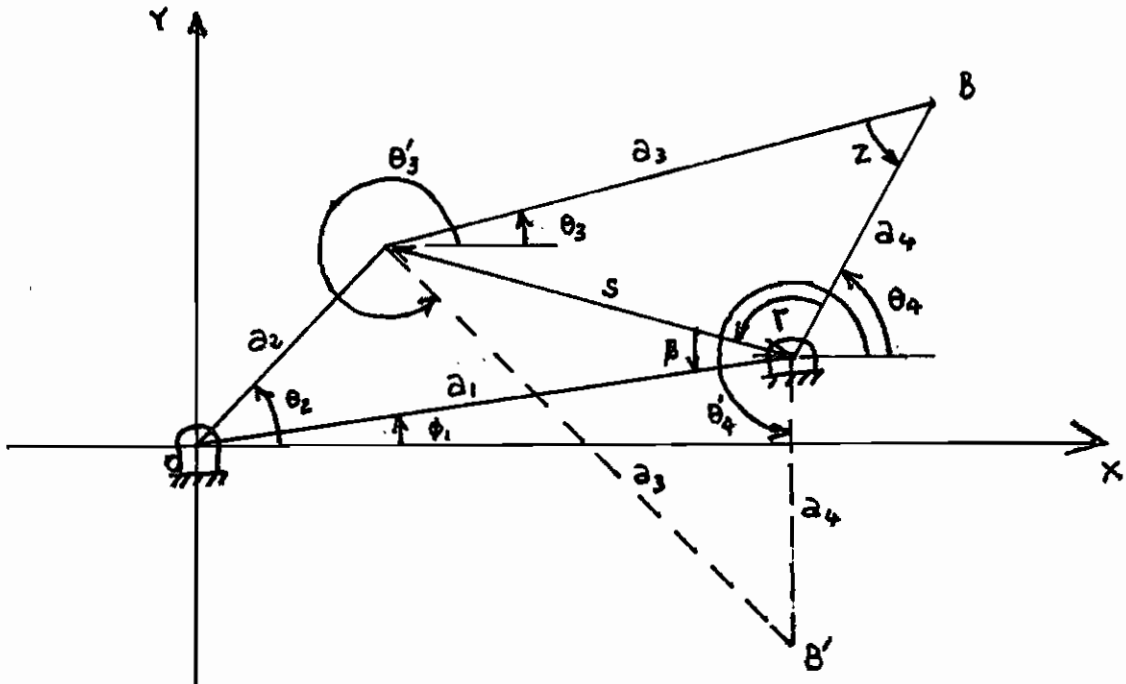


Figure 1.1 Positions angulaires des membrures 3 et 4.

1<sup>er</sup> cas (représenté en trait continu)

En considérant le triangle (OAC) on pose:

$$S^2 = a_1^2 + a_2^2 - 2a_1a_2 \cos(\theta_2 - \phi_1) \quad (1.1)$$

$$\sin\beta/a_2 = \sin(\theta_2 - \phi_1)/S \implies \sin\beta = a_2 \sin(\theta_2 - \phi_1)/S \quad (1.2a)$$

$$a_2^2 = a_1^2 + S^2 - 2a_1S \cos\beta \implies \cos\beta = (-a_2^2 + a_1^2 + S^2)/(2a_1S) \quad (1.2b)$$

$$\text{Nous pouvons alors déterminer } \beta = \arctan(\sin\beta/\cos\beta) \quad (1.2).$$

Considérant le triangle (ABC) nous avons:

$$a_4^2 = a_3^2 + S^2 - 2a_3S \cos\delta \implies \cos\delta = (-a_4^2 + a_3^2 + S^2)/(2a_3S) \quad (1.3a)$$

$$\text{Pour le premier cas, } \sin\delta = [1 - \cos^2\delta]^{1/2} \quad (1.3b)$$

$$\text{On a alors } \delta = \arctan(\sin\delta/\cos\delta) \quad (1.3)$$

$$\text{On établit aisément la relation: } \theta_3 = \phi_1 - \beta + \delta. \quad (1.4)$$

Considérons de nouveau le triangle (ABC):

$$\sin\Gamma/a_3 = \sin\delta/a_4 \implies \sin\Gamma = a_3 \sin\delta/a_4 \quad (1.5a)$$

$$a_3^2 = a_4^2 + S^2 - 2a_4S \cos\Gamma \implies \cos\Gamma = (-a_3^2 + a_4^2 + S^2)/(2a_4S) \quad (1.5b)$$

$$\text{On a alors } \Gamma = \arctan(\sin\Gamma/\cos\Gamma) \quad (1.5)$$

$$\text{On établit aisément: } \theta_4 = \theta_3 + (\pi - \delta - \Gamma) = \pi - \beta + \phi_1 - \Gamma \quad (1.6)$$

2<sup>ème</sup> cas (représenté en trait discontinu)

En observant la figure précédente et en faisant de simples considérations angulaires on trouve:

$$\theta'_3 = -\delta + \phi_1 - \beta \quad (1.7)$$

$$\theta'_4 = \pi + \Gamma + \phi_1 - \beta \quad (1.8).$$

**1.1.2 Vitesses et accélérations angulaires.**

Considérons le mécanisme à quatre membrures de la figure (fig1.2). Suivant la notation complexe, les vecteur-positions  $a_1, a_2, a_3, a_4$  sont exprimés comme suite:

$$a_1 = a_1 e^{i\theta_1} \quad (1.9a)$$

$$a_2 = a_2 e^{i\theta_2} \quad (1.9b)$$

$$a_3 = a_3 e^{i\theta_3} \quad (1.9c)$$

$$a_4 = a_4 e^{i\theta_4} \quad (1.9d)$$

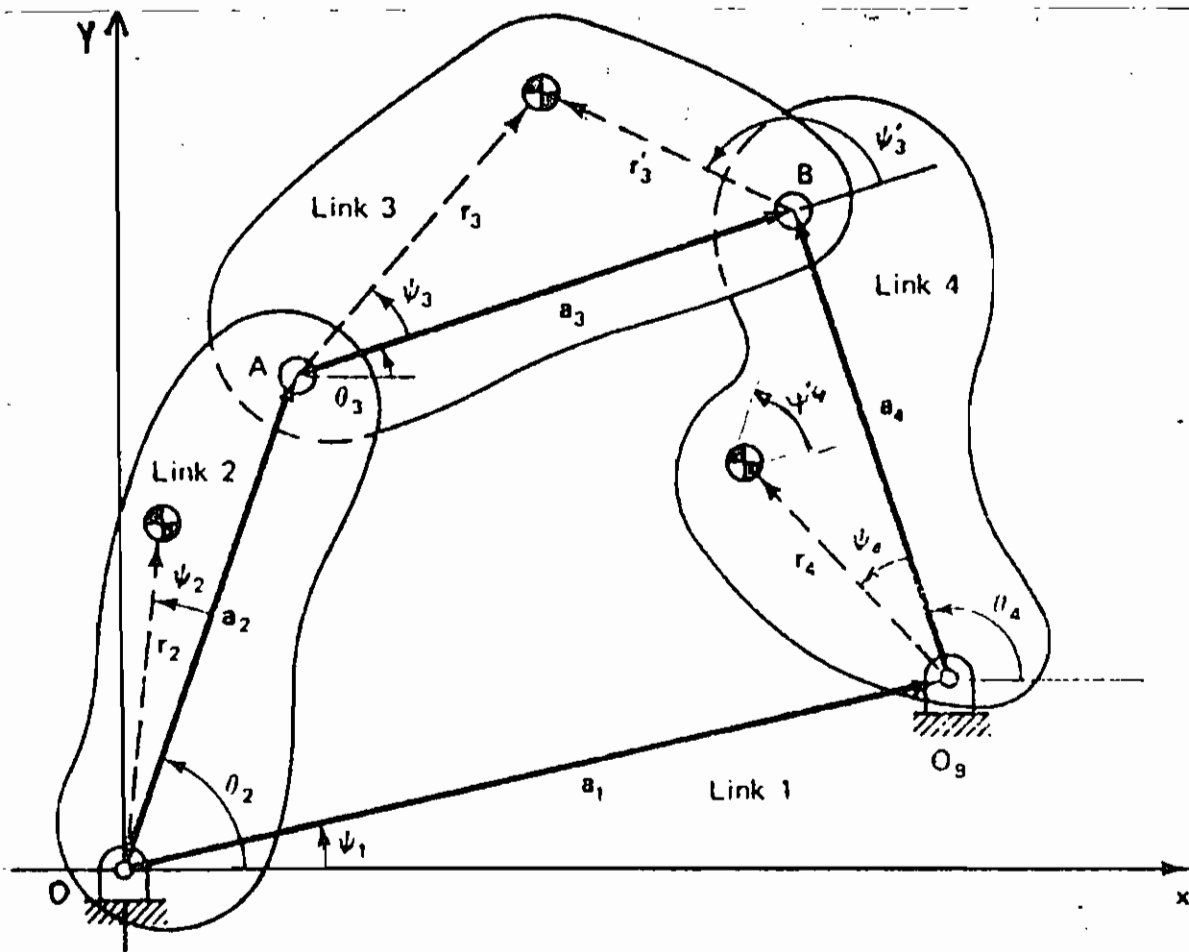


Figure 1.2 Mecanisme à quatre membrures: calcul des vitesses.

Les quatre membrures formant un polygone fermé, d'après les relations précédentes nous avons:

$$a_1 e^{i\theta_1} + a_4 e^{i\theta_4} = a_2 e^{i\theta_2} + a_3 e^{i\theta_3} \quad (1.10)$$

En dérivant (1.10), on obtient:

$$a_4 w_4 e^{i\theta_4} = a_2 w_2 e^{i\theta_2} + a_3 w_3 e^{i\theta_3} \quad (1.11)$$

$$\text{Partie imaginaire: } a_4 w_4 \sin\theta_4 = a_2 w_2 \sin\theta_2 + a_3 w_3 \sin\theta_3 \quad (1.12a)$$

$$\text{Partie réelle: } a_4 w_4 \cos\theta_4 = a_2 w_2 \cos\theta_2 + a_3 w_3 \cos\theta_3 \quad (1.12b)$$

En résolvant (1.12a) et (1.12b) on obtient:

$$w_3 = w_2 [-a_2 \sin(\theta_2 - \theta_4)] / [a_3 \sin(\theta_3 - \theta_4)] \quad (1.13)$$

$$w_4 = w_2 [+a_2 \sin(\theta_2 - \theta_3)] / [a_4 \sin(\theta_3 - \theta_4)] \quad (1.14)$$

La dérivation de (1.11) donne:

$$a_4 e^{i\theta_4} (\alpha_4 + i w_4^2) = a_2 e^{i\theta_2} (\alpha_2 + i w_2^2) + a_3 e^{i\theta_3} (\alpha_3 + i w_3^2) \quad (1.15)$$

Les parties imaginaire et réelle donnent respectivement:

$$a_4 (\alpha_4 \sin\theta_4 + w_4^2 \cos\theta_4) = a_2 (\alpha_2 \sin\theta_2 + w_2^2 \cos\theta_2) + a_3 (\alpha_3 \sin\theta_3 + w_3^2 \cos\theta_3) \quad (1.16a)$$

$$a_4 (\alpha_4 \cos\theta_4 - w_4^2 \sin\theta_4) = a_2 (\alpha_2 \cos\theta_2 - w_2^2 \sin\theta_2) + a_3 (\alpha_3 \cos\theta_3 - w_3^2 \sin\theta_3) \quad (1.16b)$$

En résolvant ces deux équations on obtient:

$$\alpha_4 = \frac{w_4 \alpha_2}{w_2} + \frac{a_2 w_2^2 \cos(\theta_3 - \theta_2) - a_4 w_4^2 \cos(\theta_3 - \theta_4) + a_3 w_3^2}{a_4 \sin(\theta_4 - \theta_3)} \quad (1.17)$$

$$\alpha_3 = \frac{w_3 \alpha_2}{w_2} + \frac{a_2 w_2^2 \cos(\theta_4 - \theta_2) + a_3 w_3^2 \cos(\theta_3 - \theta_4) - a_4 w_4^2}{-a_3 \sin(\theta_3 - \theta_4)} \quad (1.18)$$



### 1.1.3 Vitesses et accélérations linéaires.

Par la méthode des nombres complexes, les positions des centres de gravité des membrures mobiles sont données par:

$$\mathbf{r}_{g2} = r_{g2} e^{i(\theta_2 + \phi_2)} \quad (1.19a)$$

$$\mathbf{r}_{g3} = r_{g3} e^{i(\theta_3 + \phi_3)} + a_2 e^{i\theta_2} \quad (1.19b)$$

$$\mathbf{r}_{g4} = r_{g4} e^{i(\theta_4 + \phi_4)} + a_1 e^{i\theta_1} \quad (1.19c)$$

En les dérivant on obtient les vitesses:

$$\mathbf{V}_{g2} = i r_{g2} \omega_2 e^{i(\theta_2 + \phi_2)} \quad (1.20a)$$

$$\mathbf{V}_{g3} = i r_{g3} \omega_3 e^{i(\theta_3 + \phi_3)} + i a_2 \omega_2 e^{i\theta_2} \quad (1.20b)$$

$$\mathbf{V}_{g4} = i r_{g4} \omega_4 e^{i(\theta_4 + \phi_4)} \quad (1.20c)$$

En les dérivant on obtient les accélérations:

$$\mathbf{A}_{g2} = i r_{g2} \alpha_2 e^{i(\theta_2 + \phi_2)} - r_{g2} \omega_2^2 e^{i(\theta_2 + \phi_2)} \quad (1.21a)$$

$$\mathbf{A}_{g3} = i r_{g3} \alpha_3 e^{i(\theta_3 + \phi_3)} - r_{g3} \omega_3^2 e^{i(\theta_3 + \phi_3)} + i a_2 \alpha_2 e^{i\theta_2} - a_2 \omega_2^2 e^{i\theta_2} \quad (1.21b)$$

$$\mathbf{A}_{g4} = i r_{g4} \alpha_4 e^{i(\theta_4 + \phi_4)} - r_{g4} \omega_4^2 e^{i(\theta_4 + \phi_4)} \quad (1.21c)$$

En les développant et en les décomposant on obtient:

$$A_{g2x} = r_{g2} [ -\alpha_2 \sin(\theta_2 + \phi_2) - \omega_2^2 \cos(\theta_2 + \phi_2) ] \quad (1.22a)$$

$$A_{g2y} = r_{g2} [ +\alpha_2 \cos(\theta_2 + \phi_2) - \omega_2^2 \sin(\theta_2 + \phi_2) ] \quad (1.22b)$$

$$A_{g3x} = r_{g3} [ -\alpha_3 \sin(\theta_3 + \phi_3) - \omega_3^2 \cos(\theta_3 + \phi_3) ] + a_2 [ -\alpha_2 \sin\theta_2 - \omega_2^2 \cos\theta_2 ] \quad (1.23a)$$

$$A_{g3y} = r_{g3} [ +\alpha_3 \cos(\theta_3 + \phi_3) - \omega_3^2 \sin(\theta_3 + \phi_3) ] + a_2 [ +\alpha_2 \cos\theta_2 - \omega_2^2 \sin\theta_2 ] \quad (1.23b)$$

$$A_{g4x} = r_{g4} [ -\alpha_4 \sin(\theta_4 + \phi_4) - \omega_4^2 \cos(\theta_4 + \phi_4) ] \quad (1.24a)$$

$$A_{g4y} = r_{g4} [ +\alpha_4 \cos(\theta_4 + \phi_4) - \omega_4^2 \sin(\theta_4 + \phi_4) ] \quad (1.24b)$$

**1.1.4 Forces et Moments d'inertie; Forces aux joints et Couple d'entrée.**

Ayant déterminé au paragraphe précédent les accélération linéaires et angulaires, l'équation générale de la dynamique nous permet d'exprimer les forces et moments d'inertie comme suite:

$$\begin{aligned}
 F_{o2x} &= -m_2 A_{g2x} & F_{o3x} &= -m_3 A_{g3x} & F_{o4x} &= -m_4 A_{g4x} \\
 F_{o2y} &= -m_2 A_{g2y} & F_{o3y} &= -m_3 A_{g3y} & F_{o4y} &= -m_4 A_{g4y} & (1.25) \\
 T_{o2} &= -I_{g2} \alpha_2 & T_{o3} &= -I_{g3} \alpha_3 & T_{o4} &= -I_{g4} \alpha_4
 \end{aligned}$$

Faisons le digramme du corps rendu libre de chacune des trois membrures et pour chacune, appliquons les équations d'équilibre.

Membrure 2

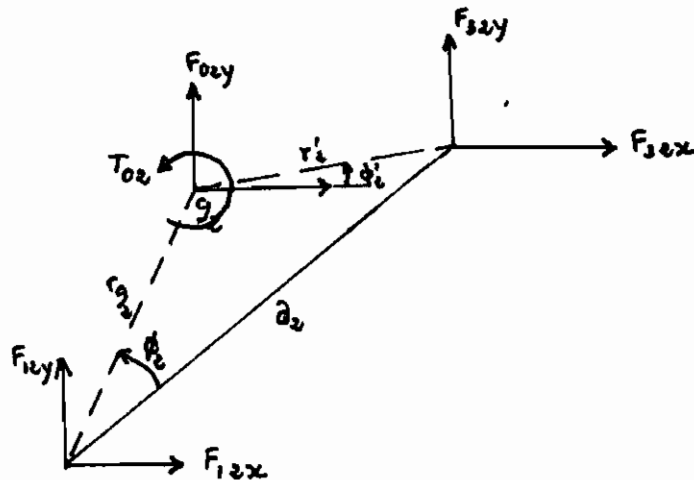


Figure 1.3 Diagramme du corps rendu libre de la membrure 2: forces aux joints.

$$\Sigma F_x = 0, \quad F_{12x} + F_{32x} + F_{o2x} = 0 \quad (1.26a)$$

$$\Sigma F_y = 0, \quad F_{12y} + F_{32y} + F_{o2y} = 0 \quad (1.26b)$$

$$\begin{aligned}
 \Sigma M = 0, \quad T_{o2} + T_s + F_{12x} r_{g2} \sin(\theta_2 + \phi_2) - F_{12y} r_{g2} \cos(\theta_2 + \phi_2) \\
 + F_{32y} r'_2 \cos \phi'_2 - F_{32x} r'_2 \sin \phi'_2 = 0 \quad (1.26c)
 \end{aligned}$$

Membrure 3

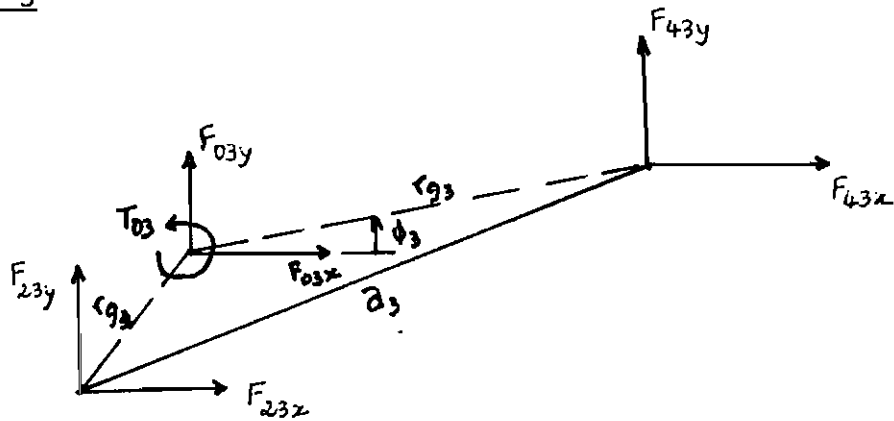


Figure 1.4 Diagramme du corps rendu libre de la membrure 3: forces aux joints.

$$\Sigma F_x=0, F_{23x}+F_{43x}+F_{03x} = 0 \quad (1.27a)$$

$$\Sigma F_y=0, F_{23y}+F_{43y}+F_{03y} = 0 \quad (1.27b)$$

$$\Sigma M=0, T_{03}+F_{23x}r_{g3}\sin(\theta_3+\phi_3)-F_{23y}r_{g3}\cos(\theta_3+\phi_3) \\ +F_{43y}r'_3\cos\phi'_3-F_{43x}r'_3\sin\phi'_3 = 0 \quad (1.27c)$$

Membrure 4

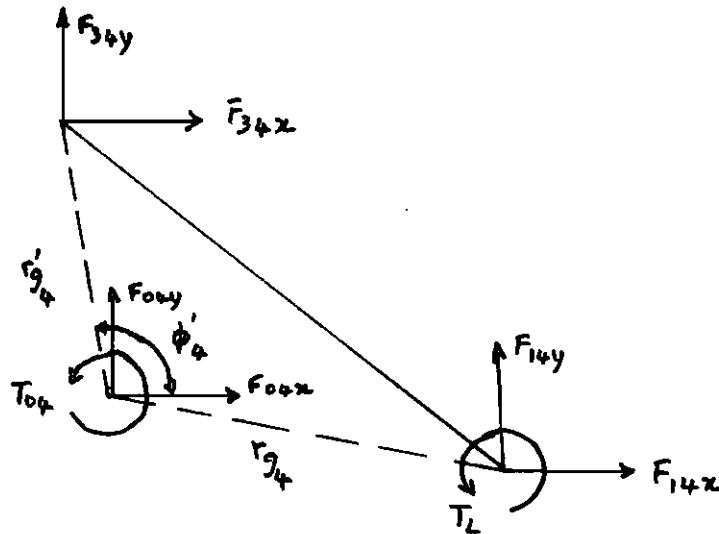


Figure 1.5 Diagramme du corps rendu libre de la membrure 4: forces aux joints.

$$\Sigma F_x=0, F_{14x}+F_{34x}+F_{o4x} = 0 \quad (1.28a)$$

$$\Sigma F_y=0, F_{14y}+F_{34y}+F_{o4y} = 0 \quad (1.28b)$$

$$\Sigma M=0, T_{o4}+T_L+F_{14x}r_{g4}\sin(\theta_4+\phi_4)-F_{14y}r_{g4}\cos(\theta_4+\phi_4) \\ +F_{34y}r'_4\cos\phi'_4-F_{34x}r'_4\sin\phi'_4 = 0 \quad (1.28c)$$

Notons qu'en vertu du principe de l'action et de la réaction:

$$F_{ij}=-F_{ji} \quad (1.29)$$

En y tenant compte et en organisant les neuf équations obtenues sous forme matricielle, on parvient à la représentation de la figure (fig1.6). Un tel système d'équations peut être résolu par plusieurs méthodes dont l'inversion matricielle et la méthode de gauss. Dans le cadre de ce projet, nous avons opté pour la méthode de gauss car elle est plus rapide, moins laborieuse et plus précise sur ordinateurs.

$F_{02x}$	-1	0	0	1	0	0	0	0	0	0	$F_{12x}$
$F_{02y}$	0	-1	0	0	1	0	0	0	0	0	$F_{12y}$
$T_{02}$	$-rg_2 \sin(\theta_2 + \phi_2)$	$rg_2 \cos(\theta_2 + \phi_2)$	-1	$-rg'_2 \sin \phi'_2$	$rg'_2 \cos \phi'_2$	0	0	0	0	0	$T_6$
$F_{03x}$	0	0	0	-1	0	1	1	0	0	0	$F_{23x}$
$F_{03y}$	0	0	0	0	-1	0	0	0	0	0	$F_{23y}$
$T_{03}$	0	0	0	$-rg_3 \sin(\theta_3 + \phi_3)$	$rg_3 \cos(\theta_3 + \phi_3)$	$-rg'_3 \sin \phi'_3$	$rg'_3 \cos \phi'_3$	0	0	0	$F_{34x}$
$F_{04x}$	0	0	0	0	0	-1	0	-1	0	0	$F_{34y}$
$F_{04y}$	0	0	0	0	0	0	-1	0	0	-1	$F_{14x}$
$T_{04} + T_L$	0	0	0	0	0	$rg'_4 \sin \phi'_4$	$-rg'_4 \cos \phi'_4$	$-rg_4 \sin(\theta_4 + \phi_4)$	$rg_4 \cos(\theta_4 + \phi_4)$	0	$F_{14y}$

Fig 1.6 Matrice donnant les forces aux joints et le couple d'entrée  $T_6$ .

## 1.2- MÉCANISMES A QUATRE MEMBRURES MANIVELLE-COULISSEAU

Ce mécanisme diffère de celui étudié au paragraphe précédent par le fait que la quatrième membrure (le coulisseau) a une longueur infinie. Cette particularité rend impossible l'application intégrale de toutes les relations établies ci-haut. Certains ajustements s'avèrent donc nécessaires. Pour ce, nous allons suivre la même procédure que précédemment.

### 1.2.1 Position angulaire de la membrures 3

Considérons un mécanisme à quatre membrures manivelle-coulisseau dont les dimensions et le décalage sont connus. Pour un même angle d'entrée  $\theta_2$ , deux différentes configurations sont possibles (figure 1.7):

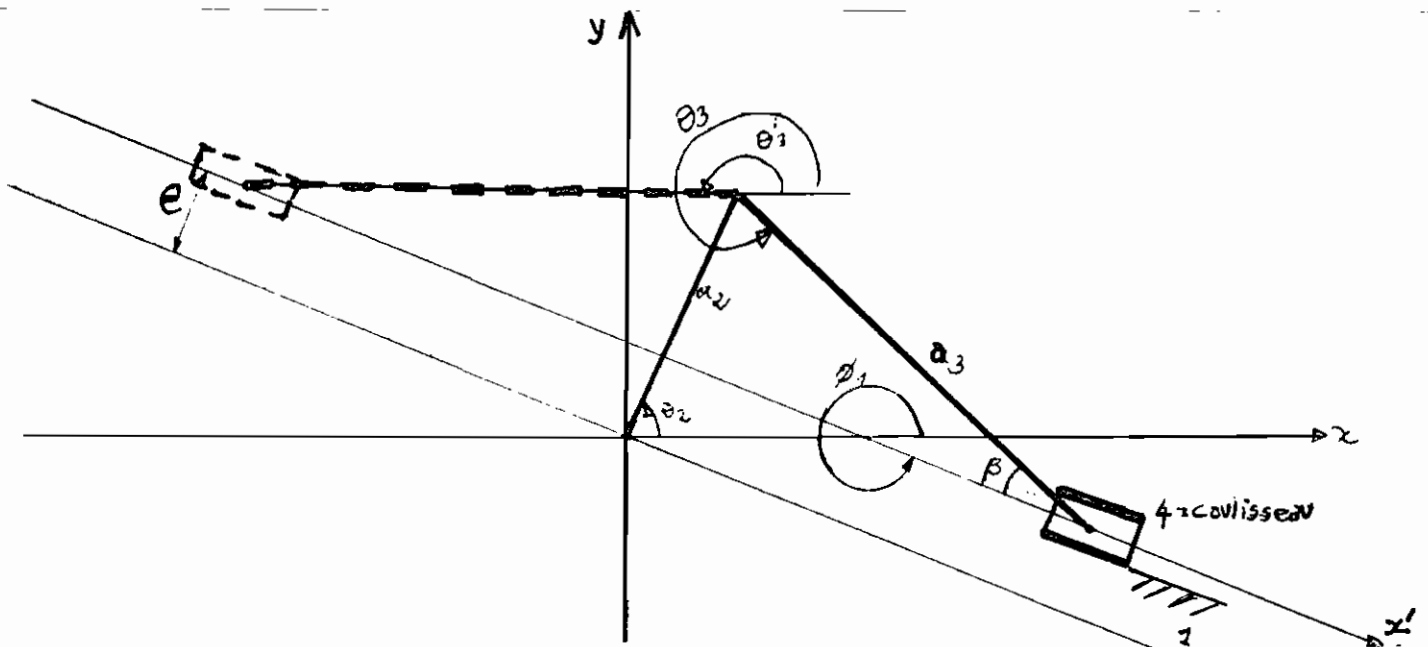


Figure 1.7 Mécanisme Manivelle-coulisseau: position angulaire de s.

$\phi_1$  et  $\beta$  sont définies comme montrées sur la figure précédente.

Dans les deux cas,  $\sin\beta = (-a_2 \sin(\theta_2 - \phi_1) - e) / a_3$  (1.30a)

1<sup>er</sup> cas (représenté en trait continu)

$\cos\beta = [1 - \sin^2\beta]^{1/2}$  (1.30b)

De (1.30a) et de (1.30b) on détermine  $\beta = \arctan(\sin\beta / \cos\beta)$  (1.30)

Les relations entre les angles nous conduisent à:  $\theta_3 = \phi_1 - \beta - \pi$ . (1.31)

2<sup>ème</sup> cas (représenté en trait discontinu)

En observant la figure précédente et en faisant de simples considérations angulaires on trouve:

$\theta'_3 = \phi_1 + \beta$  (1.32)

**1.2.2 Vitesses et accélérations.**

Considérons le mécanisme à quatre membrures de la figure 1.8.

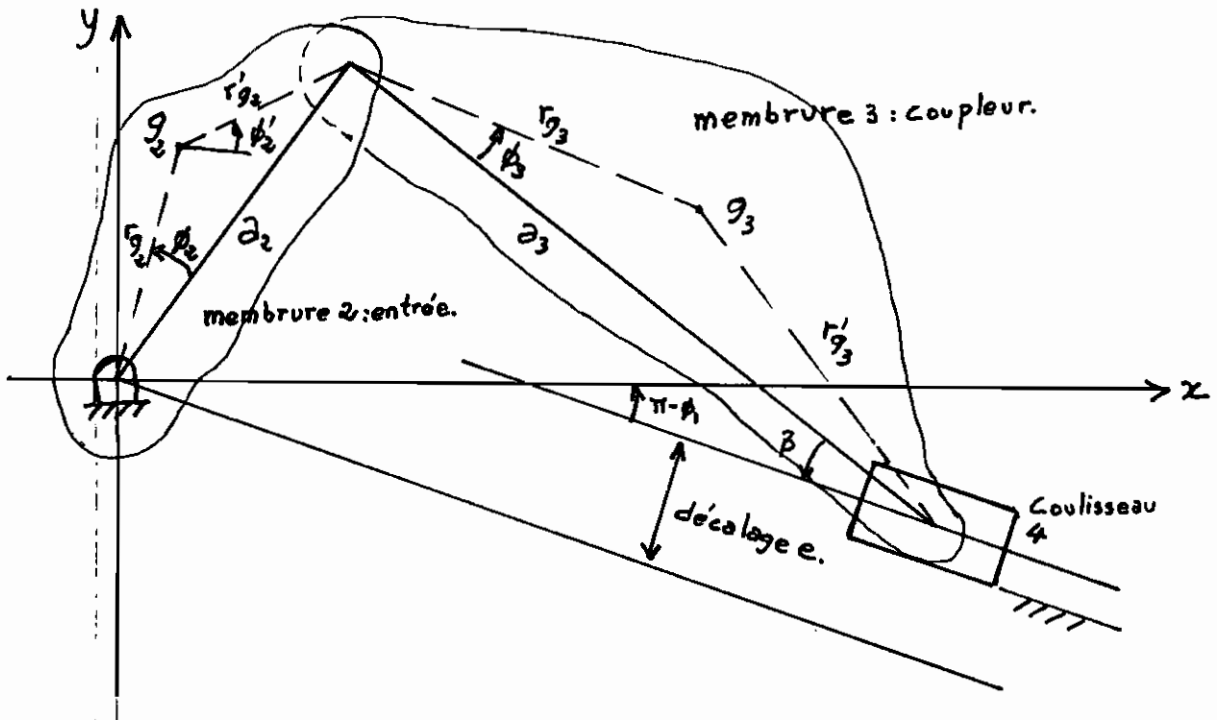


Figure 1.8 Mécanisme manivelle-coulisseau: calcul des vitesses.

Exprimons le vecteur position  $OB$  sous forme complexe dans le repère  $(x', y')$ , tel que défini sur la figure précédente:

$$OB = a_2 e^{i(\theta_2 + \tau - \phi_1)} + a_3 e^{i(\theta_3 + \tau - \phi_1)}$$

$$OB = -a_2 e^{i(\theta_2 - \phi_1)} - a_3 e^{i(\theta_3 - \phi_1)} \quad (1.33)$$

$$dOB/dt = -ia_2 w_2 e^{i(\theta_2 - \phi_1)} - ia_3 w_3 e^{i(\theta_3 - \phi_1)} \quad (1.34)$$

Étant donné que le coulisseau ne fait aucun mouvement dans la direction  $(By')$ ,  $(dOB/dt)_{y'} = 0$  d'où nous tirons la vitesse angulaire de la membrure 3:

$$w_3 = -w_2 [a_2 \cos(\theta_2 - \phi_1) / a_3 \cos(\theta_3 - \phi_1)] \quad (1.35)$$

La vitesse de translation du coulisseau est donnée par:

$$(dOB/dt)_{x'} = V_{Bx'} = a_2 w_2 \sin(\theta_2 - \phi_1) + a_3 w_3 \sin(\theta_3 - \phi_1) \quad (1.36)$$

Dérivons la relation (1.34):

$$\begin{aligned} d^2OB/dt^2 = & -ia_2 \alpha_2 e^{i(\theta_2 - \phi_1)} + a_2 w_2^2 e^{i(\theta_2 - \phi_1)} \\ & - ia_3 \alpha_3 e^{i(\theta_3 - \phi_1)} + a_3 w_3^2 e^{i(\theta_3 - \phi_1)} \end{aligned} \quad (1.37)$$

Pour les mêmes raisons évoquées plus haut,  $(d^2OB/dt^2)_{y'} = 0$  d'où nous tirons l'accélération angulaire de la membrure 3:

$$\alpha_3 = \alpha_2 \frac{w_2}{w_4} + \frac{a_2 w_2^2 \sin(\theta_2 - \phi_1)}{a_3 \cos(\theta_3 - \phi_1)} + w_3^2 \tan(\theta_3 - \phi_1) \quad (1.38)$$

L'accélération de translation du coulisseau est donnée par:

$$\begin{aligned} (d^2OB/dt^2)_{x'} = A_{Bx'} = & a_2 \alpha_2 \sin(\theta_2 - \phi_1) + a_2 w_2^2 \cos(\theta_2 - \phi_1) \\ & + a_3 \alpha_3 \sin(\theta_3 - \phi_1) + a_3 w_3^2 \cos(\theta_3 - \phi_1) \end{aligned} \quad (1.39)$$

Les vitesses et les accélérations linéaires des membrures 2 et 3 sont définies par les mêmes relations qu'au paragraphe A-3.



**1.2.3 Forces et Moments d'inertie; Forces aux joints et Couple d'entrée.**

De la même manière qu'au paragraphe A-4, nous pouvons écrire:

$$\begin{aligned}
 F_{o2x} &= -m_2 A_{g2x} & F_{o3x} &= -m_3 A_{g3x} & T_{o2} &= -I_{g2} \alpha_2 & T_{o3} &= -I_{g3} \alpha_3 \\
 F_{o2y} &= -m_2 A_{g2y} & F_{o3y} &= -m_3 A_{g3y} & F_{o4x'} &= -m_4 A_{B4x'} & & 
 \end{aligned}
 \tag{1.40}$$

Les six équations d'équilibre (1.25) obtenues par application pour les membrures 2 et 3 sont ici encore valables. Pour la membrure 4 faisons le digramme du corps rendu libre et écrivons l'équation d'équilibre suivant la direction Bx'.

Membrure 4

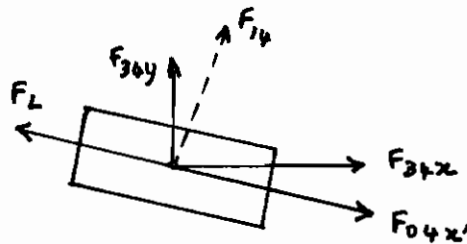


Figure 1.9 Diagramme du corps rendu libre du coulisseau.

$$\begin{aligned}
 \Sigma F_{x'} &= 0, \quad F_{14x'} + F_{34x'} + F_{o4x'} = 0 \\
 F_{o4x'} &= F_{34x} \cos \phi_1 + F_{14x} \sin \phi_1
 \end{aligned}
 \tag{1.41}$$

On obtient ainsi sept équations linéaires qui organisées sous forme matricielle donnent à la représentation de la figure 1.10.

$$\begin{array}{|l}
 F_{O2x} \\
 F_{O2y} \\
 T_{O2} \\
 F_{O3x} \\
 F_{O3y} \\
 T_{O3} \\
 F_{O4x} + F_1
 \end{array}
 =
 \begin{array}{|c|c|c|c|c|c|c|c|}
 \hline
 -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 \hline
 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
 \hline
 -rg_2 \sin(\theta_2 + \phi_2) & rg_2 \cos(\theta_2 + \phi_2) & -1 & -rg'_2 \sin \phi'_2 & rg'_2 \cos \phi'_2 & 0 & 0 & 0 \\
 \hline
 0 & 0 & 0 & -1 & 0 & 1 & 1 & 1 \\
 \hline
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
 \hline
 0 & 0 & 0 & -rg_3 \sin(\theta_3 + \phi_3) & rg_3 \cos(\theta_3 + \phi_3) & -rg'_3 \sin \phi'_3 & rg'_3 \cos \phi'_3 & 0 \\
 \hline
 0 & 0 & 0 & 0 & 0 & \cos \phi_1 & \sin \phi_1 & 0 \\
 \hline
 \end{array}
 \begin{array}{|l}
 F_{12x} \\
 F_{12y} \\
 C_2 \\
 F_{23x} \\
 F_{23y} \\
 F_{34x} \\
 F_{34y}
 \end{array}$$

Fig 1.10 système permettant de déterminer les forces aux joints et le couple d'entrée

## CHAPITRE 2

### ANALYSE DE LA RÉPONSE TEMPORELLE

Dans certains problèmes de conception, l'analyse cinéto-statique étudiée au chapitre précédent s'avère inadéquat. En effet nous avons supposé connues toutes les caractéristiques cinétiques de la membrure d'entrée; ce qui n'est pourtant toujours pas le cas. Il est plutôt fréquent de connaître la position angulaire initiale  $\theta_2$ , la vitesse angulaire initiale  $w_2$  et la loi d'évolution du couple d'entrée  $T_s$ . Ainsi le couple d'entrée qui avait précédemment été considéré comme une inconnue devient une donnée connue; tandis que l'accélération angulaire  $\alpha_2$  fait le chemin inverse. Le problème ainsi posé est appelé: **analyse de la réponse temporelle.**

La résolution d'un tel problème comporte les étapes suivantes:

- établir l'équation différentielle caractérisant le mouvement de la membrure d'entrée;
- déterminer l'accélération angulaire  $\alpha_2$  à partir de la position angulaire  $\theta_2$  et de la vitesse angulaire  $w_2$  initiales, et du couple d'entrée  $T_s$ ;
- déterminer la position et la vitesse angulaire à l'instant  $(t+\Delta t)$  en résolvant l'équation différentielle en tenant compte des conditions de l'instant  $t$ . Et de là reprendre la procédure à partir de l'étape 2.

Étant donné la grande variété des machines entraînant, une infinité d'équations différentielles peuvent être établit. Pour ce, dans le cadre de ce projet, nous nous limiterons à l'étude du

deuxième point. Ainsi il s'agira de déterminer les accélérations angulaires à partir de la position  $\theta_2$ , de la vitesse  $w_2$ , du couple  $T_s$  etc.

La démarche pour établir le système d'équation nous permettant d'y parvenir est la même que celle suivie au chapitre précédent. La différence c'est qu'ici  $\alpha_2$  devient une inconnue alors que  $T_s$  est connu.

### 2.1 -Mécanismes à quatre membrures de dimensions finies

#### Membrure 2:

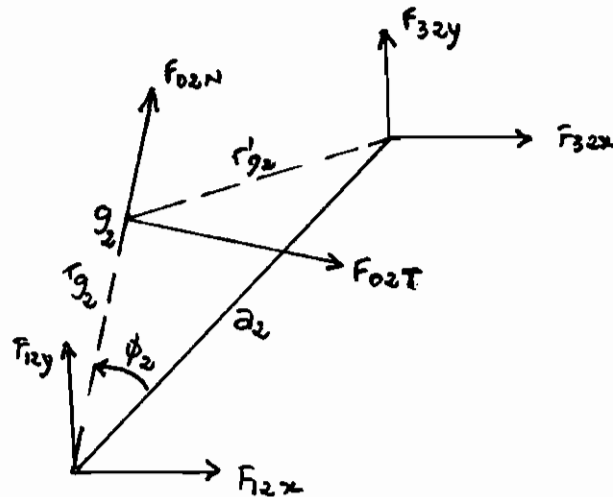


Figure 2.1 Réponse temporelle: diagramme du corps rendu libre de la membrure 2.

L'accélération linéaire du centre de gravité  $g_2$  de la membrure 2 est:

$$A_{g_2} = i r_{g_2} \alpha_2 - r_{g_2} w_2^2 \quad (2.1)$$

$$\begin{aligned} F_{o2} &= F_{o2T} + F_{o2N} = -m_2 A_{g_2} \\ &= -i m_2 r_{g_2} \alpha_2 + m_2 r_{g_2} w_2^2 \\ &= -m_2 (i r_{g_2x} - r_{g_2y}) \alpha_2 + m_2 (r_{g_2x} + i r_{g_2y}) w_2^2 \end{aligned} \quad (2.2)$$

$$\Sigma \mathbf{F} = 0, \quad \mathbf{F}_{12} + \mathbf{F}_{32} + \mathbf{F}_{o2} = 0;$$

$$\Sigma F_x = 0, \quad F_{12x} - F_{23x} + m_2 r_{g2x} w_2^2 + m_2 r_{g2y} \alpha_2 = 0$$

$$m_2 r_{g2x} w_2^2 = -F_{12x} + F_{23x} - m_2 r_{g2y} \alpha_2 \quad (2.3a)$$

$$\Sigma F_y = 0, \quad F_{12y} - F_{23y} + m_2 r_{g2y} w_2^2 - m_2 r_{g2x} \alpha_2 = 0$$

$$m_2 r_{g2y} w_2^2 = -F_{12y} + F_{23y} - m_2 r_{g2x} \alpha_2 \quad (2.3b)$$

$$\Sigma M = 0, \quad T_s = -F_{12x} r_{g2} \sin(\theta_2 + \phi_2) + F_{12y} r_{g2} \cos(\theta_2 + \phi_2) + F_{23y} r'_{g2} \cos \phi'_2$$

$$-F_{23x} r'_{g2} \sin \phi'_2 + I_{g2} \alpha_2 \quad (2.3c)$$

Membrure 3:

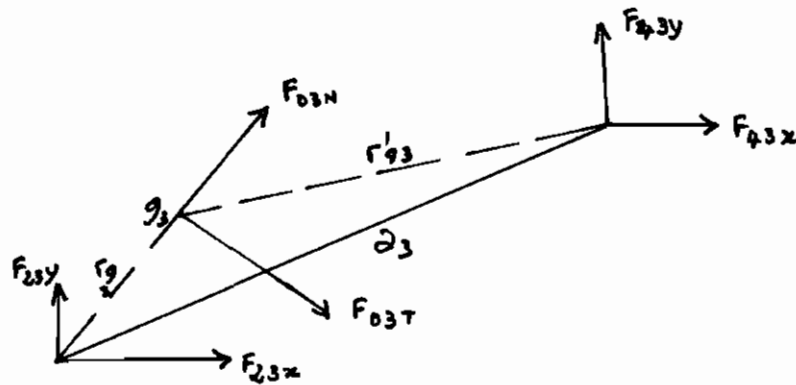


Figure 2.2 Réponse temporelle: diagramme du corps rendu libre de la membrure 3.

Nous avons vu au chapitre I :

$$\alpha_3 = \frac{w_3 \alpha_2}{w_2} + \frac{a_2 w_2^2 \cos(\theta_4 - \theta_2) + a_3 w_3^2 \cos(\theta_3 - \theta_4) - a_4 w_4^2}{-a_3 \sin(\theta_3 - \theta_4)} \quad (1.18)$$

Notons C = 
$$\frac{a_2 w_2^2 \cos(\theta_4 - \theta_2) + a_3 w_3^2 \cos(\theta_3 - \theta_4) - a_4 w_4^2}{-a_3 \sin(\theta_3 - \theta_4)}$$

$$\text{Alors } \alpha_3 = \frac{w_3 \alpha_2}{w_2} + C \quad (2.4)$$

$$A_{g3} = A_A + i r_{g3} \alpha_3 - r_{g3} w_3^2 \text{ or } A_A = i a_2 \alpha_2 - a_2 w_2^2$$

$$\text{d'où } A_{g3} = i a_2 \alpha_2 - a_2 w_2^2 + i r_{g3} \alpha_3 - r_{g3} w_3^2 \quad (2.5)$$

En inserrant (2.4) dans (2.5), on obtient:

$$A_{g3} = i a_2 \alpha_2 - a_2 w_2^2 + i r_{g3} (w_3/w_2) \alpha_2 + i C r_{g3} - r_{g3} w_3^2 \quad (2.6)$$

$$A_{g3x} = -a_{2y} \alpha_2 - a_{2x} w_2^2 - (r_{g3y} w_3/w_2) \alpha_2 - C r_{g3y} - r_{g3x} w_3^2 \quad (2.7a)$$

$$A_{g3y} = +a_{2x} \alpha_2 - a_{2y} w_2^2 + (r_{g3x} w_3/w_2) \alpha_2 + C r_{g3x} - r_{g3y} w_3^2 \quad (2.7b)$$

$$\Sigma F = 0, F_{23} + F_{43} + F_{o3} = 0$$

En tenant compte des relations (1.25), nous poursuivons:

$\Sigma F_x = 0, F_{23x} - F_{34x} - m_3 A_{g3x} = 0$  en y inserrant (2.7a) et en réarrangeant on obtient:

$$m_3 (a_{2x} w_2^2 + C r_{g3y} + r_{g3x} w_3^2) = -F_{23x} + F_{34x} - m_3 (a_{2y} + r_{g3y} (w_3/w_2)) \alpha_2 \quad (2.8a)$$

$\Sigma F_y = 0$ , en procédant de même que ci-haut on obtient:

$$m_3 (a_{2y} w_2^2 - C r_{g3x} + r_{g3y} w_3^2) = -F_{23y} + F_{34y} - m_3 (a_{2x} + r_{g3x} (w_3/w_2)) \alpha_2 \quad (2.8b)$$

Au chapitre 1 nous avons vu:

$$\begin{aligned} \Sigma M = 0, T_{o3} + F_{23x} r_{g3} \sin(\theta_3 + \phi_3) - F_{23y} r_{g3} \cos(\theta_3 + \phi_3) \\ + F_{43y} r'_3 \cos \phi'_3 - F_{43x} r'_3 \sin \phi'_3 = 0 \end{aligned} \quad (1.27c)$$

En tenant compte des relations (1.25) et (2.4) on obtient:

$$\begin{aligned} -I_{g3} C = - F_{23x} r_{g3} \sin(\theta_3 + \phi_3) + F_{23y} r_{g3} \cos(\theta_3 + \phi_3) - F_{34x} r'_3 \sin \phi'_3 \\ + F_{34y} r'_3 \cos \phi'_3 + (I_{g3} w_3/w_2) \alpha_2 \end{aligned} \quad (2.8c)$$

Membrure 4:

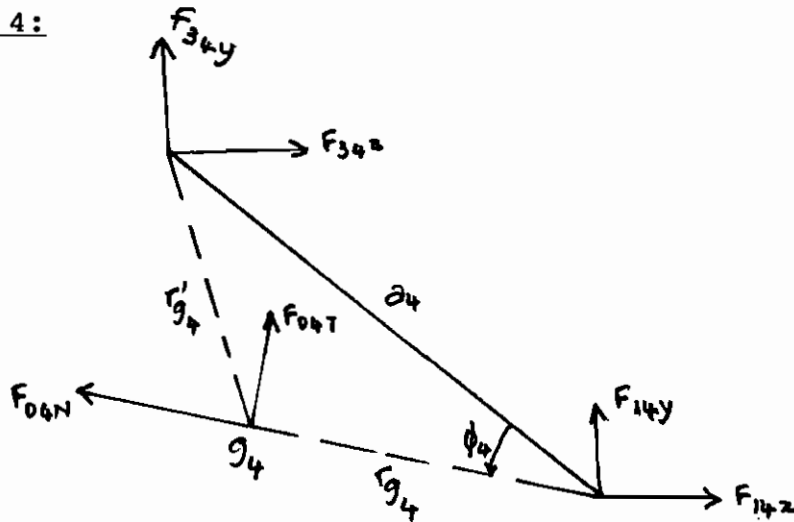


Figure 2.3 Réponse temporelle: diagramme du corps rendu libre de la membrure 4.

Nous avons vu au chapitre I :

$$\alpha_4 = \frac{w_4 \alpha_2}{w_2} + \frac{a_2 w_2^2 \cos(\theta_3 - \theta_2) - a_4 w_4^2 \cos(\theta_3 - \theta_4) + a_3 w_3^2}{a_4 \sin(\theta_4 - \theta_3)} \quad (1.17)$$

Notons  $D = \frac{a_2 w_2^2 \cos(\theta_3 - \theta_2) - a_3 w_4^2 \cos(\theta_3 - \theta_4) + a_3 w_3^2}{a_3 \sin(\theta_4 - \theta_3)}$

Alors  $\alpha_4 = \frac{w_4 \alpha_2}{w_2} + D \quad (2.9)$

$$A_{g4} = +i r_{g4} \alpha_4 - r_{g4} w_4^2 \quad (2.10)$$

En portant (2.9) dans (2.10), on obtient:

$$A_{g4} = +i r_{g4} (w_4/w_2) \alpha_2 + i D r_{g4} - r_{g4} w_4^2 \quad (2.11)$$

$$A_{g4x} = - (r_{g4y} w_4/w_2) \alpha_2 - D r_{g4y} - r_{g4x} w_4^2 \quad (2.12a)$$

$$A_{g4y} = + (r_{g4x} w_4/w_2) \alpha_2 + C r_{g4x} - r_{g4y} w_4^2 \quad (2.12b)$$

$$\Sigma F=0, F_{34x}+F_{14x}+F_{o4}=0$$

En prenant en considération les relations (1.25) et en procédant comme ci-haut, nous poursuivons:

$\Sigma F_x=0, F_{34x}+F_{14x}-m_4 A_{g4x} = 0$  en y inserrant (2.12a) et en réarrangeant on obtient:

$$m_4(Dr_{g4y}+r_{g4x}w_4^2) = -F_{34x} - F_{14x} - m_4 r_{g4y} (w_4/w_2)\alpha_2 \quad (2.13a)$$

$\Sigma F_y=0$ , en procédant de même que ci-haut on obtient:

$$m_4(-Dr_{g4x}+r_{g4y}w_4^2) = -F_{34y} - F_{14y} + m_4 r_{g4x} (w_4/w_2)\alpha_2 \quad (2.13b)$$

Au chapitre 1 nous avons vu:

$$\begin{aligned} \Sigma M=0, T_{o4}+T_L+F_{14x}r_{g4}\sin(\theta_4+\phi_4)-F_{14y}r_{g4}\cos(\theta_4+\phi_4) \\ +F_{34y}r'_{4}\cos\phi'_{4}-F_{34x}r'_{4}\sin\phi'_{4} = 0 \end{aligned} \quad (1.28c)$$

En tenant compte des relations (1.25) et (2.9) nous obtenons:

$$\begin{aligned} -I_{g4}D + T_L = +F_{34x}r'_{4}\sin\phi'_{4} - F_{34y}r'_{4}\cos\phi'_{4} -F_{14x}r_{g4}\sin(\theta_4+\phi_4) \\ +F_{14y}r_{g4}\cos(\theta_4+\phi_4) + I_{g4}(w_4/w_2)\alpha_2 \end{aligned} \quad (2.13c)$$

Les équations ainsi établies forment un système de neuf équations linéaires à neuf inconnues. Sous forme matricielle il est représenté à la figure 2.4.



$y_1$	-1	0	$-m_2 r g_{2y}$	1	0	0	0	0	0	$F_{12x}$
$y_2$	0	-1	$m_2 r g_{2x}$	0	1	0	0	0	0	$F_{12y}$
$y_3 = T_4$	$-r g_{2y}$	$r g_{2x}$	$I_{r2}$	$-r g'_2 \sin \phi'_2$	$r g'_2 \cos \phi'_2$	0	0	0	0	$\alpha_2$
$y_4$	0	0	$a(\alpha_2, y_4)$	-1	0	1	0	0	0	$F_{23x}$
$y_5$	0	0	$a(\alpha_2, y_5)$	0	-1	0	1	0	0	$F_{23y}$
$y_6$	0	0	$a(\alpha_2, y_6)$	$-r g_3 \sin(\theta_3 + \phi_3)$	$-r g_3 \cos(\theta_3 + \phi_3)$	$-r g'_3 \sin \phi'_3$	$r g'_3 \cos \phi'_3$	0	0	$F_{34x}$
$y_7$	0	0	$a(\alpha_2, y_7)$	0	0	-1	0	-1	0	$F_{34y}$
$y_8$	0	0	$a(\alpha_2, y_8)$	0	0	0	-1	0	-1	$F_{14x}$
$y_9$	0	0	$a(\alpha_2, y_9)$	0	0	$r g'_4 \sin \phi'_4$	$-r g'_4 \cos \phi'_4$	$-r g_4 \sin(\theta_4 + \phi_4)$	$r g_4 \cos(\theta_4 + \phi_4)$	$F_{14y}$

$$\begin{aligned}
y_1 &= m_2 r g_{2x} w_2^2 \\
y_2 &= m_2 r g_{2y} w_2^2 \\
y_3 &= T_4 \\
y_4 &= m_3 (a_{r2x} w_2^2 + C r g_{3y} + r g_{3x} w_3^2) \\
y_5 &= m_3 (a_{r2y} w_2^2 - C r g_{3y} + r g_{3y} w_3^2) \\
y_6 &= -I_{r3} C \\
y_7 &= m_4 (D r g_{4y} + r g_{4x} w_4^2) \\
y_8 &= m_4 (-D r g_{4x} + r g_{4y} w_4^2) \\
y_9 &= -I_{r4} D + I_1 \\
a(\alpha_2, y_4) &= -m_3 (a_{2y} + r g_{3y} w_3 / w_2) \\
a(\alpha_2, y_5) &= -m_3 (a_{2x} + r g_{3x} w_3 / w_2) \\
a(\alpha_2, y_6) &= I_{r3} w_3 / w_2 \\
a(\alpha_2, y_7) &= -m_4 r g_{4y} w_4 / w_2 \\
a(\alpha_2, y_8) &= m_4 r g_{4x} w_4 / w_2 \\
a(\alpha_2, y_9) &= I_{r4} w_4 / w_2
\end{aligned}$$

Fig 2.4 Matrice de la réponse temporelle d'un mécanisme à quatre membrures de dimensions finies

## 2.2 Mécanismes à quatre membrures: Manivelle-Coulisseau

### Membrure 2:

On obtient les mêmes équations (2.3a), (2.3b) et (2.3c) qu'au paragraphe précédent:

$$m_2 r_{g2x} w_2^2 = -F_{12x} + F_{23x} - m_2 r_{g2y} \alpha_2 \quad (2.3a)$$

$$m_2 r_{g2y} w_2^2 = -F_{12y} + F_{23y} - m_2 r_{g2x} \alpha_2 \quad (2.3b)$$

$$\begin{aligned} Ts = & -F_{12x} r_{g2} \sin(\theta_2 + \phi_2) + F_{12y} r_{g2} \cos(\theta_2 + \phi_2) + F_{23y} r'_{g2} \cos \phi'_2 \\ & - F_{23x} r'_{g2} \sin \phi'_2 + I_{g2} \alpha_2 \end{aligned} \quad (2.3c)$$

### Membrure 3:

$$\text{En notant } E = + \frac{a_2 w_2^2 \sin(\theta_2 - \phi_1)}{a_3 \cos(\theta_3 - \phi_1)} + w_3^2 \tan(\theta_3 - \phi_1) ,$$

d'après la relation (1.38) nous pouvons écrire:

$$\alpha_3 = \alpha_2 \frac{w_2}{w_4} + E \quad (2.14)$$

En remplaçant C par E dans les relations (2.8a), (2.8b) et (2.8c) on obtient les équations recherchées:

$$m_3 (a_{2x} w_2^2 + E r_{g3y} + r_{g3x} w_3^2) = -F_{23x} + F_{34x} - m_3 (a_{2y} + r_{g3y} (w_3/w_2)) \alpha_2 \quad (2.15a)$$

$$m_3 (a_{2y} w_2^2 - E r_{g3x} + r_{g3y} w_3^2) = -F_{23y} + F_{34y} - m_3 (a_{2x} + r_{g3x} (w_3/w_2)) \alpha_2 \quad (2.15b)$$

$$\begin{aligned} -I_{g3} E = & -F_{23x} r_{g3} \sin(\theta_3 + \phi_3) + F_{23y} r_{g3} \cos(\theta_3 + \phi_3) - F_{34x} r'_3 \sin \phi'_3 \\ & + F_{34y} r'_3 \cos \phi'_3 + (I_{g3} w_3/w_2) \alpha_2 \end{aligned} \quad (2.15c)$$

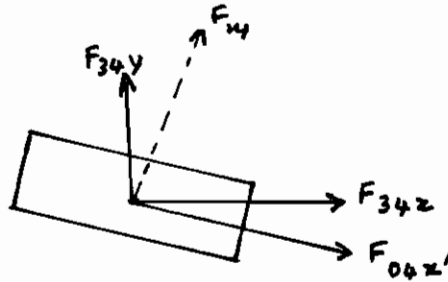
Membrure 4:

Figure 2.5 Réponse temporelle: diagramme du corps rendu libre du coulisseau.

$$A_{Bx'} = a_2 \alpha_2 \sin(\theta_2 - \phi_1) + a_2 w_2^2 \cos(\theta_2 - \phi_1) + a_3 \alpha_3 \sin(\theta_3 - \phi_1) + a_3 w_3^2 \cos(\theta_3 - \phi_1) \quad (1.39)$$

En y inserrant (2.14) on obtient:

$$A_{Bx'} = a_2 \alpha_2 \sin(\theta_2 - \phi_1) + a_2 w_2^2 \cos(\theta_2 - \phi_1) + a_3 w_3^2 \cos(\theta_3 - \phi_1) + a_3 \sin(\theta_3 - \phi_1) (w_3/w_2) \alpha_2 + E a_3 \sin(\theta_3 - \phi_1) \quad (2.16)$$

$$\Sigma F_{x'} = 0, \quad F_{34x} \cos(\pi - \phi_1) - F_{34y} \sin(\pi - \phi_1) + F_{04x'} = 0 \quad (2.17)$$

De la relation  $F_{04x'} = -m_4 A_{Bx'}$ , et de ce qui précède il vient:

$$m_4 [a_2 w_2^2 \cos(\theta_2 - \phi_1) + a_3 w_3^2 \cos(\theta_3 - \phi_1) + E a_3 \sin(\theta_3 - \phi_1)] = F_{34x} \cos \phi_1 + F_{34y} \sin \phi_1 - \alpha_2 m_4 [a_3 \sin(\theta_3 - \phi_1) (w_3/w_2) + a_2 \sin(\theta_2 - \phi_1)] \quad (2.18)$$

Les équations (2.3a,b,c); (2.15a,b,c) et (2.17) obtenues forment un système à sept inconnues (représenté à la figure 2.6 permettant de faire l'analyse de la réponse temporelle des mécanismes à quatre membrures manivelle-coulisseau.

$y_1$	-1	0	$-m_2 r_{r2y}$	1	0	0	0	$F_{12x}$
$y_2$	0	-1	$m_2 r_{r2x}$	0	1	0	0	$F_{12y}$
$y_3 = T_s$	$-r g_{2y}$	$r g_{2x}$	$I_{r2}$	$-r g'_2 \sin \phi'_2$	$r g'_2 \cos \phi'_2$	0	0	$\alpha_2$
$y_4$	0	0	$a(\alpha_2, y_4)$	-1	0	1	0	$F_{23x}$
$y_5$	0	0	$a(\alpha_2, y_5)$	0	-1	0	1	$F_{23y}$
$y_6$	0	0	$a(\alpha_2, y_6)$	$-r g_3 \sin(\theta_3 + \phi_3)$	$-r g_3 \cos(\theta_3 + \phi_3)$	$-r g'_3 \sin \phi'_3$	$r g'_3 \cos \phi'_3$	$F_{34x}$
$y_7$	0	0	$a(\alpha_2, y_7)$	0	0	$\cos \phi_1$	$\sin \phi_1$	$F_{34y}$

$$\begin{aligned}
 y_4 &= m_3(a_{2x} w_2^2 + E r g_{3y} + r g_{3x} w_3^2) \\
 y_5 &= m_3(a_{2y} w_2^2 - E r g_{3x} + r g_{3y} w_3^2) \\
 y_6 &= -I_{r3} E \\
 y_7 &= m_4[a_2 w_2^2 \cos(\theta_2 - \phi_1) + a_3 w_3^2 \cos(\theta_3 - \phi_1) + E a_3 \sin(\theta_3 - \phi_1)] + F_1 \\
 a(\alpha_2, y_4) &= (a_{2y} + r g_{3y} w_3 / w_2)(-m_3) \\
 a(\alpha_2, y_5) &= (a_{2x} + r g_{3x} w_3 / w_2)(-m_3) \\
 a(\alpha_2, y_6) &= I_{r3} w_3 / w_2 \\
 a(\alpha_2, y_7) &= -m_4[a_3 \sin(\theta_3 - \phi_1) * w_3 / w_2 + a_3 \sin(\theta_2 - \phi_1)]
 \end{aligned}$$

Fig 2.6 Matrice de la réponse temporelle ( Manivelle - coulisseau ).

## CHAPITRE 3 ÉQUILIBRAGE DES ROTORS

L'équilibrage des rotors revêt une importance considérable dans la machinerie. Un défaut d'équilibrage est source de vibrations pouvant avoir des conséquences dommageables pour les rotors. On distingue deux types d'équilibrage: l'équilibrage statique et l'équilibrage dynamique.

### 3.1 ÉQUILIBRAGE STATIQUE

Considérons une masse excentrique  $m_i$  situé à la position  $r_i$  de l'axe de rotation du rotor (figure 3.1). La force de gravité de cette excentricité crée un moment égal à :  $r_i \wedge m_i g$  (3.1)

( $g$  étant la constante de la pesanteur).

Ce moment a pour effet de déplacer le rotor de sa position d'équilibre statique. Nous obtiendrons l'équilibrage statique lorsque la somme des moments de toutes les excentricités sera nulle. Il s'agira donc de faire en sorte que le centre de gravité du rotor et de toutes ces excentricités soit sur l'axe de rotation. La notion d'équilibre statique, ne tient compte d'aucune considération cinétique.

Désignons par le thème "désbalancement statique" d'une telle excentricité, le produit scalaire:  $S_i = m_i r_i$  (3.2)

Si le rotor compte plusieurs excentricités, on aura un désbalancement total  $S_M = \sum S_i = \sum m_i r_i$  (3.3)

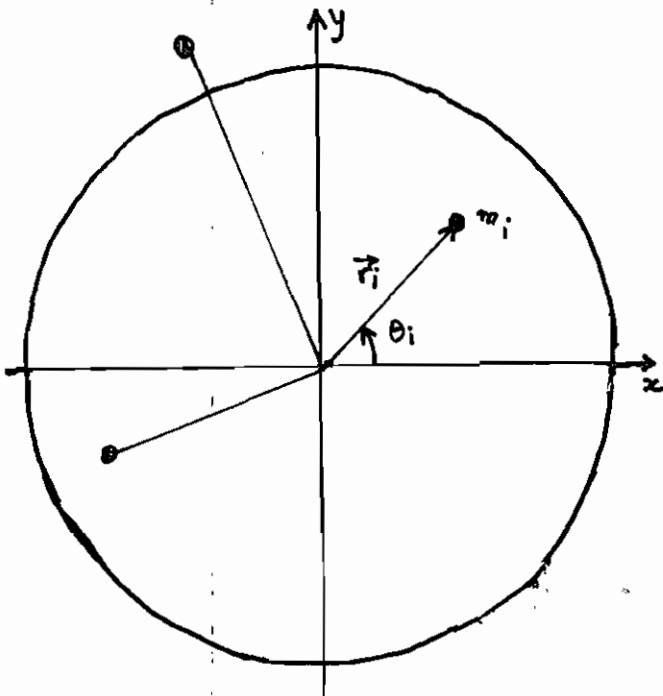


Figure 3.1 Excentricités.

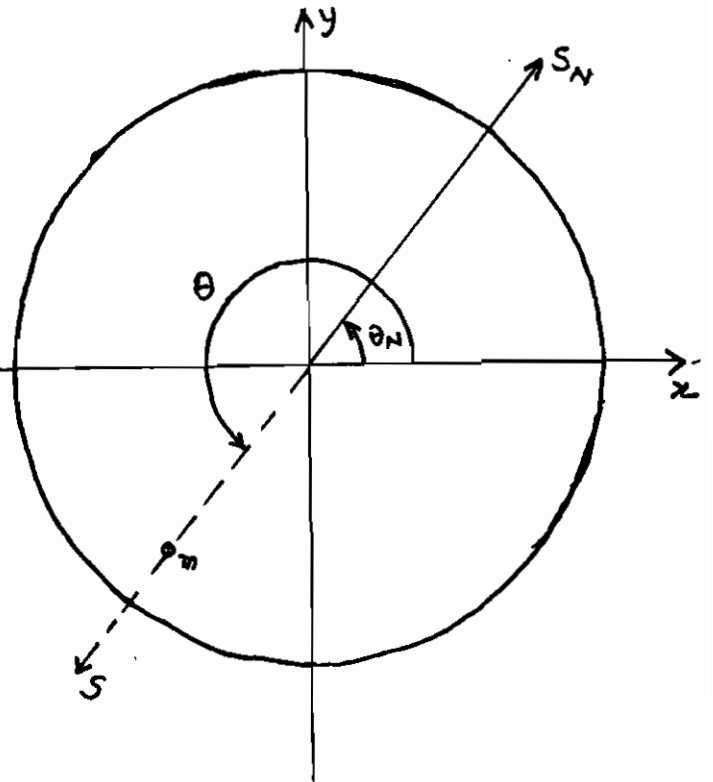


Figure 3.2 Correction.

Dans un repère orthonormé  $(xoy)$ , de l'équation (3.3) on peut écrire:

$$S_x = \sum m_i x_i \quad (3.4)$$

$$S_y = \sum m_i y_i \quad (3.5)$$

$$|S_N| = (S_x^2 + S_y^2)^{1/2} \quad (3.6)$$

$$\theta_n = \arg S_N \quad (3.7)$$

sachant que  $x_i = r_i \cos \theta_i$  et  $y_i = r_i \sin \theta_i$

où  $\theta_i$  est la position angulaire de l'excentricité  $i$ ;

$S_x$  et  $S_y$  les composantes du déséquilibre total suivant les axes  $x$  et  $y$ ;

et  $\theta_n$  la position angulaire du déséquilibre total.

L'équilibrage statique ne sera obtenu que lorsqu'on aura disposer sur le rotor une excentricité de correction (figure 3.2) de masse  $m$ , de position radiale  $r$ , de position angulaire  $\theta$ , de débalancement  $\mathbf{s} = m\mathbf{r}$  telle que:

$$\mathbf{s} = -\mathbf{s}_N \quad (3.8)$$

$$|\mathbf{s}| = |\mathbf{s}_N| \quad (3.9)$$

$$\theta = \theta_N + \pi \quad (3.10)$$

### 3.2 ÉQUILIBRAGE DYNAMIQUE

Le déséquilibre dynamique survient uniquement lorsque, le rotor est en mouvement. La figure 3.3 montre deux excentricités chacune de masse  $m$  située à la position radiale  $r$  sur le rotor.

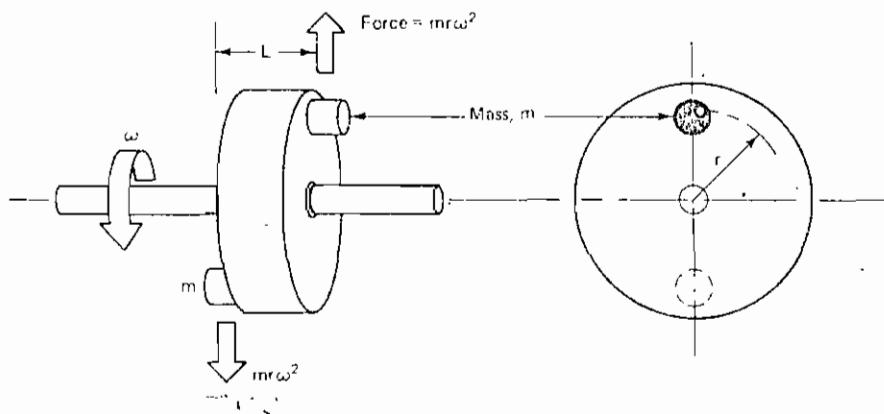


Figure 3.3 Déséquilibre dynamique.

Lorsque le rotor est animé d'un mouvement de rotation à la vitesse  $w$ , chacune des excentricités est soumise à une force centrifuge  $F = mrw^2 = Sw^2$ . Étant donné que ces deux excentricités sont séparées par une distance transversale  $L$ , le rotor subira l'effet d'un couple  $T_i = F_i L_i = m_i r_i w^2 L_i$ . (3.11)

L'équilibrage dynamique n'interviendra que lorsqu'on aura installé des excentricités de corrections permettant d'annuler les couples ainsi engendrés. Dans certains cas particuliers, un seul plan de correction pourrait suffire à réaliser l'équilibrage. Mais dans le cas général, l'utilisation de deux plans s'avère nécessaire. On pourrait à dessein utiliser plus de deux plans mais puisse que deux suffiront dans tous les cas, notre étude s'y limitera. Notons ces deux plans A et B (voir figure 3.4).

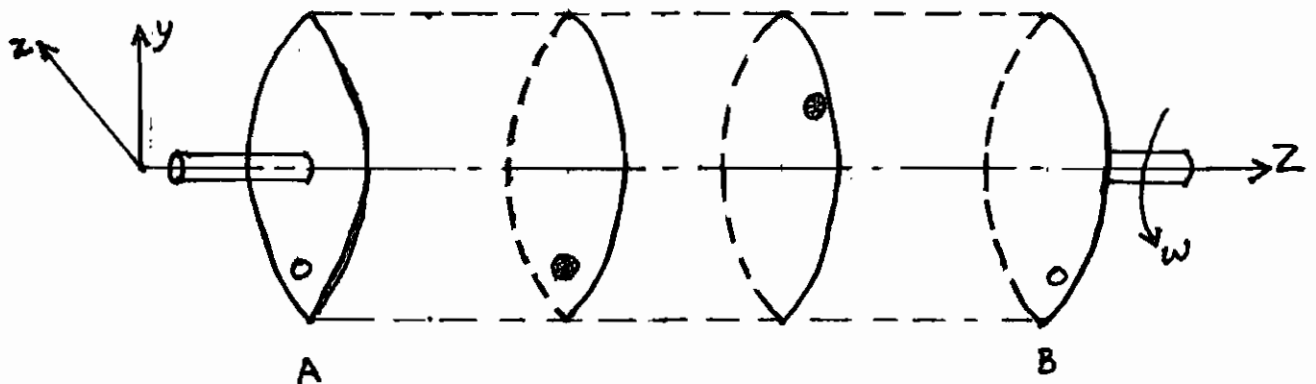


Figure 3.4 Plans de correction A et B.



L'excentricité de correction placée dans le plan B doit être de telle sorte que le moment par rapport à A de la force d'inertie qu'elle subit sera l'opposé de la somme des moments par rapport au même plan A de toutes les excentricités.

En notant  $I_B = m_B r_B (z_B - z_A)$  (3.12) (le module de  $I_B$  a dimension de moment d'inertie), l'énoncé précédent se traduit comme suite:

$$I_B \omega^2 = -\sum S_i \omega^2 (z_i - z_A) \quad (3.13)$$

La mécanique des matériaux nous enseigne que compte tenu des flèches -dues à la flexion-, la vitesse angulaire diffère d'une position de z -fonction de la portée- à une autre. Dans notre cas, nous supposerons que la portée du rotor est assez faible et que le rotor est suffisamment rigide pour ne pas tenir compte de la variation de vitesse angulaire. Ainsi (3.13) devient:

$$I_B = -\sum S_i (z_i - z_A). \quad (3.14)$$

En faisant de même pour l'excentricité de correction placée dans le plan A, on obtient:  $I_A = -\sum S_i (z_B - z_i)$ . (3.15)

$$\text{avec } I_A = m_A r_A (z_A - z_B). \quad (3.16)$$

Comme nous l'avons fait au premier paragraphe, les "vecteurs moment d'inertie" ( $I_A$  et  $I_B$ ) peuvent être aisément décomposés en leurs composantes cartésiennes ( $I_{Ax}, I_{Ay}$  et  $I_{Bx}, I_{By}$ ) et polaires ( $I_A, \theta_A$  et  $I_B, \theta_B$ ).

Une fois que les positions transversales ( $z_A, z_B$ ) des plans de correction A et B sont fixées, nous pouvons déterminer les "vecteurs moments d'inertie"  $I_A$  et  $I_B$  à partir des équations (3.14)

et (3.15). Dès lors que  $I_A$  et  $I_B$  sont connues, pour chacune des masses de correction nous pouvons fixer la masse et déterminer le vecteur position ou vice versa.

## CHAPITRE 4

### ÉQUILIBRAGE DES MÉCANISMES A QUATRE MEMBRURES

L'équilibrage des mécanismes à quatre membrures comporte deux aspects à savoir: le balancement des forces et le balancement des moments. Comme nous l'avons déjà dit dans le chapitre I, le mouvement des éléments d'un mécanisme fait apparaître des charges dynamiques. Ces charges dynamiques sont subies par la membrure fixe qui les transmet à la fondation du mécanisme. Elles sont à l'origine des forces de frottement supplémentaires, des vibrations des éléments et de la fondation, de contraintes supplémentaires et de bruits, etc... L'équilibrage a pour but de réduire ou mieux d'éliminer ces charges dynamiques.

#### 4.1 ÉQUILIBRAGE DES FORCES

On démontre que pour réaliser le balancement des forces d'inertie il suffit de concevoir le mécanisme de manière à ce que lors de son mouvement, le centre de gravité du système formé par les membrures mobiles restent fixe<sup>(1)</sup>.

##### 4.1.1 Mécanismes à quatre membrures de longueurs finies

Considérons le mécanisme à quatre membrures de la figure 4.1. Soient  $M$  la masse totale des membrures mobiles et  $r_g$  la position du centre de gravité du système formé par ces dernières.

<sup>(1)</sup> Pour plus de détail consulter "Théorie des mécanismes et des machines" de I. ARTOBOLEVSKI aux ÉDITIONS MIR MOSCOU. (Page 284-289)

Soient respectivement  $m_i$  et  $r_i$  la masse et la position du centre de gravité de la membrure  $i$ .

$$M = \sum_i m_i \quad (4.1)$$

$$M\mathbf{r}_s = \sum_i m_i \mathbf{r}_i \quad (4.2)$$

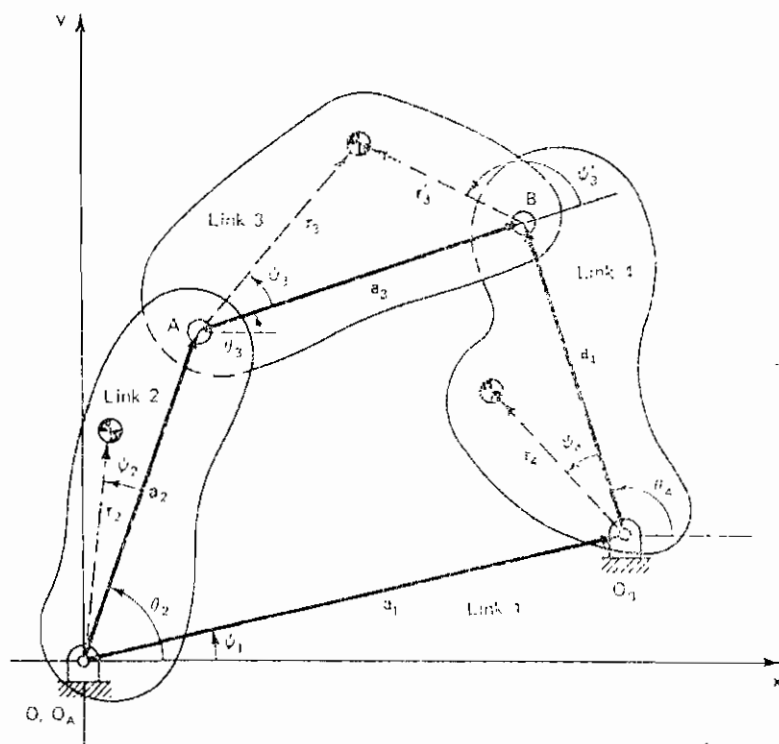


Figure 4.1 Mécanisme à quatre membrures: équilibre.

D'après la figure précédente on a:

$$\mathbf{r}_2 = r_2 e^{i(\theta_2 + \theta_2)} \quad (4.3)$$

$$\mathbf{r}_3 = r_3 e^{i(\theta_3 + \theta_3)} + a_2 e^{i\theta_2} \quad (4.4)$$

$$\text{et } \mathbf{r}_4 = r_4 e^{i(\theta_4 + \theta_4)} + a_1 e^{i\theta_1} \quad (4.5)$$

En les inserrant dans l'équation (4.2) on obtient:

$$M\mathbf{r}_s = m_2 r_2 e^{i(\theta_2 + \theta_2)} + m_3 (r_3 e^{i(\theta_3 + \theta_3)} + a_2 e^{i\theta_2}) + m_4 (r_4 e^{i(\theta_4 + \theta_4)} + a_1 e^{i\theta_1}) \quad (4.6)$$

Les membrures 1,2,3 et 4 formant un polygone fermé nous permettent d'écrire:

$$a_2 e^{i\theta_2} + a_3 e^{i\theta_3} - a_4 e^{i\theta_4} - a_1 e^{i\theta_1} = 0 \quad (4.7)$$

$$\text{De la dernière nous tirons } e^{i\theta_3} = (a_4 e^{i\theta_4} + a_1 e^{i\theta_1} - a_2 e^{i\theta_2}) / a_3 \quad (4.8)$$

En substituant (4.8) dans (4.6) on obtient:

$$\begin{aligned} M\mathbf{r}_s = & [m_2 r_2 e^{i\theta_2} + m_3 a_2 - m_3 r_3 a_2 e^{i\theta_3} / a_3] e^{i\theta_2} + [m_4 r_4 e^{i\theta_4} + m_3 r_3 a_4 e^{i\theta_3} / a_3] e^{i\theta_4} \\ & + [m_4 a_1 + m_3 r_3 a_1 e^{i\theta_3} / a_3] e^{i\theta_1} \end{aligned} \quad (4.9)$$

La relation (4.9) peut être écrite sous la forme:

$$M\mathbf{r}_s = A e^{i\theta_2} + B e^{i\theta_4} + C \quad (4.10)$$

Le centre de gravité du système restera fixe (indépendant du temps donc indépendant de  $\theta_2$  et de  $\theta_4$ ) à condition que:  $A = B = 0$

$$A=0 \iff m_2 r_2 e^{i\theta_2} + m_3 a_2 - m_3 r_3 a_2 e^{i\theta_3} / a_3 = 0 \quad (4.11)$$

$$\text{D'après la (fig4.1), } r_3 e^{i\theta_3} = a_3 + r'_3 e^{i\theta'_3} \quad (4.12)$$

De la dernière (4.11) devient:

$$m_2 r_2 e^{i\theta_2} = m_3 r'_3 a_2 e^{i\theta'_3} / a_3 \quad (4.13)$$

La relation (4.13) équivaut à:

$$m_2 r_2 = m_3 r'_3 a_2 / a_3 \quad \text{et} \quad \theta_2 = \theta'_3 \quad (4.14)$$

En faisant de même pour la condition  $B=0$  on obtient:

$$m_4 r_4 = m_3 r_3 a_4 / a_3 \quad \text{et} \quad \theta_4 = \theta_3 + \pi \quad (4.15)$$

Lorsque les conditions (4.14) et (4.15) sont satisfaites, la position du centre de gravité restera fixe à :

$$\mathbf{r}_s = (1/M)(m_4 a_1 + m_3 r_3 a_1 e^{i\theta_3} / a_3) e^{i\theta_1} = C/M \quad (4.16)$$

Pour satisfaire les conditions (4.14) et (4.15), il faudra choisir deux membrures - parmi les trois mobiles - sur lesquelles on interviendra. L'intervention peut être soit une reconfiguration de la membrure, soit l'addition d'un contre-poids de telle sorte que  $m_i r_i$  et  $\phi_i$  respectent les conditions (4.14) et (4.15). L'addition d'un contre-poids suppose évidemment que l'on ne désire pas modifier les dimensions de la membrure initiale.

Désignons par:

$m_i^\circ, r_i^\circ, \phi_i^\circ$  : les paramètres des membrures du mécanisme débalancé

$m_i^*, r_i^*, \phi_i^*$  : les paramètres des contre-poids.

Le contre-poids doit être choisi de telle sorte que:

$$m_i r_i e^{i\phi_i} = m_i^\circ r_i^\circ e^{i\phi_i^\circ} + m_i^* r_i^* e^{i\phi_i^*} \quad (4.17)$$

En décomposant (4.17) en coordonnées polaires on obtient:

$$m_i^* r_i^* = [(m_i r_i)^2 + (m_i^\circ r_i^\circ)^2 - 2m_i r_i m_i^\circ r_i^\circ \cos(\phi_i - \phi_i^\circ)]^{1/2} \quad (4.18)$$

et

$$\phi_i^* = \arg\{(m_i r_i \cos\phi_i - m_i^\circ r_i^\circ \cos\phi_i^\circ) + i(m_i r_i \sin\phi_i - m_i^\circ r_i^\circ \sin\phi_i^\circ)\} \quad (4.19)$$

4.1.2 Mécanismes à quatre membrures Manivelle-coulisseau

Dans ce paragraphe, nous allons étudier l'équilibrage des forces d'un mécanisme à quatre membrures manivelle-coulisseau tel que représenté à la figure (fig6.2).

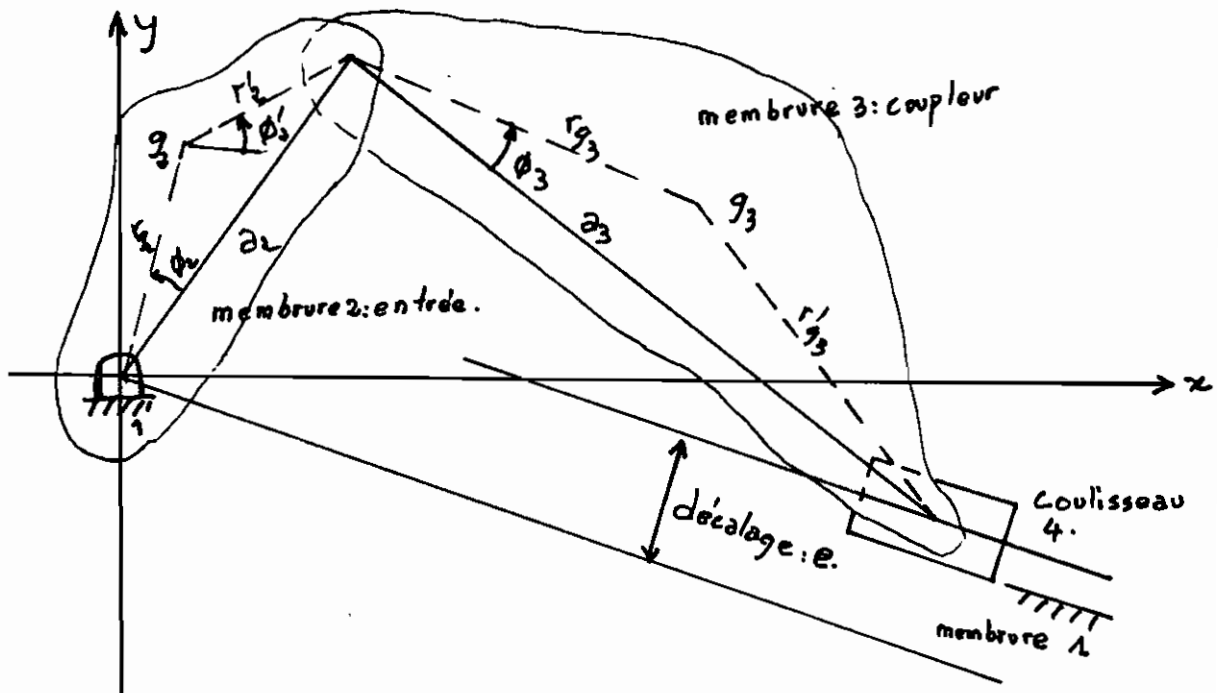


Figure 4.2 Manivelle-coulisseau: équilibrage.

En suivant la même procédure qu'au paragraphe précédent, nous établissons successivement:

$$M = \sum_i m_i \tag{4.1}$$

$$M\mathbf{r}_s = \sum_i m_i \mathbf{r}_i \tag{4.2}$$

$$\mathbf{r}_2 = r_2 e^{i(\theta_2 + \theta_2)} \tag{4.3}$$

$$r_3 = r_3 e^{i(\theta_3 + \phi_3)} + a_2 e^{i\theta_2} \quad (4.4)$$

$$r_4 = x_4 \cdot e^{i0} + e \cdot e^{i\pi/2} \quad (4.20)$$

$$Mr_s = m_2 r_2 e^{i(\theta_2 + \phi_2)} + m_3 [r_3 e^{i(\theta_3 + \phi_3)} + a_2 e^{i\theta_2}] + m_4 [x_4 + e \cdot e^{i\pi/2}] \quad (4.21)$$

$$e^{i\theta_3} = (x_4 + ie - a_2 e^{i\theta_2}) / a_3 \quad (4.22)$$

En substituant (4.22) dans (4.21) on obtient:

$$Mr_s = [m_2 r_2 e^{i\theta_2} + m_3 a_2 - m_3 r_3 a_2 e^{i\theta_3} / a_3] e^{i\theta_2} + [m_4 + m_3 r_3 e^{i\theta_3} / a_3] x_4 + [m_4 + m_3 r_3 e^{i\theta_3} / a_3] e \quad (4.23)$$

La relation (4.23) se présente également sous la forme:

$$Mr_s = A e^{i\theta_2} + B e^{i\theta_4} + C \quad (4.24)$$

Pour les mêmes raisons, il faudra que:  $A = B = 0$

$A=0$  donne les mêmes conditions que ci-dessus:

$$m_2 r_2 = m_3 r_3 a_2 / a_3 \quad \text{et} \quad \phi_2 = \phi_3 \quad (4.14)$$

La condition  $B=0$  équivaut à:

$$m_4 = m_3 r_3 / a_3 \quad \text{et} \quad \phi_3 = 0 \quad (4.25)$$

Lorsque les conditions de balancement (4.14) et (4.25) sont satisfaites, la position du centre de gravité restera fixe à :

$$r_s = (1/M) (m_4 + m_3 r_3 / a_3) e = C/M \quad (4.26)$$

Selon la condition (4.25) la membrure 3 doit avoir obligatoirement un  $\phi=0$  (centre de gravité situé entre les deux pivots sur la ligne des centres).

A quelques différences d'ordre pratique près, toutes les considérations concernant la reconfiguration et l'utilisation des contre-poids sont identiques à ce qui a été vu au paragraphe précédent.



## 4.2 Équilibrage des moments

Nous venons de voir, que l'équilibrage des forces produit une force résultante nulle sur les supports. Il ne garantit pas l'équilibrage complet du mécanisme. L'équilibrage des moments revêt une grande importance dans la mesure où son absence est particulièrement à l'origine des problèmes de vibrations, de bruits et de fatigue.

Considérons le mécanisme à quatre membrures de la figure 4.1<sup>(2)</sup>. On démontre que pour un tel mécanisme, si les forces sont balancées, la somme des moments d'inertie par rapport au centre de gravité du système des membrures mobiles est:

$$M_{W/G} = -\sum m_i (k_i^2 + r_i^2 - a_i r_i \cos \phi_i) \alpha_i - 2m_3 a_2 r_3 \sin \phi_3 (\tau \alpha_2 + \mu w_2) \quad (3) \quad (4.20)$$

avec  $\tau = \sin(\theta_2 - \theta_3)$  et  $\mu = d\tau/dt$ .

L'équilibrage des moments n'aura lieu que lorsque ce moment sera nul. Il n'est pas toujours possible de l'annuler. On s'efforce alors dans ces cas de le réduire le plus possible. Avant d'aller plus loin examinons les notions de pendule physique et de membrure "in-line".

(2) Étant donné qu'un contre-poids ne peut être fixé sur le coulisseau, nous n'étudieront que le balancement des mécanismes à longueurs finies.

(3) Pour plus de détails consulter de la page 446 à 454 le manuel "Dynamics of Mechanisms: Advanced concepts" de SANDOR & ERDMAN aux ÉDITIONS Prentice-hall.

#### 4.2.1 La membrure In-line

Une membrure in-line est une membrure dont le centre de gravité est situé sur la ligne des centres reliant les deux pivots de la membrure.

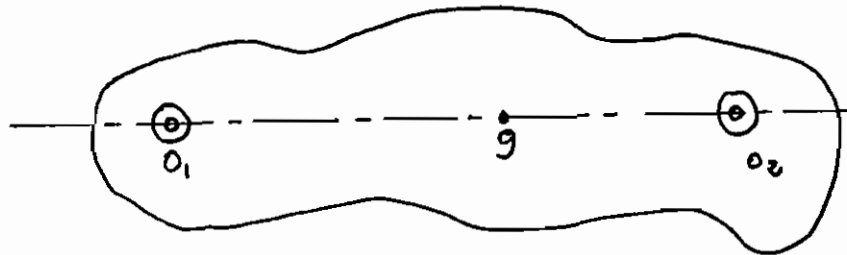


Figure 4.3 Membrure "in-line".

Considérons une membrure binaire quelconque que nous nous proposons de rendre in-line par l'addition d'un contre-poids. (figure 4.4)

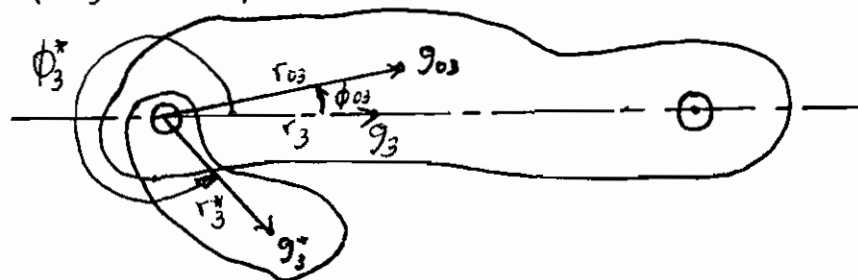


Figure 4.4 Addition d'un contre-poids.

Les paramètres avec astérisque(\*) se rapportent au contre-poids, ceux avec l'indice(o) se rapportent à la membrure initiale et ceux sans indice ni astérisque représentent les paramètres désirés.

Si nous désignons par  $r$  la position du centre de gravité de l'ensemble contre-poids - membrure initiale, on peut écrire :

$$(m^* + m_o) r = m^* r^* + m_o r_o \quad (4.21)$$

$$m^* r^* \sin \phi^* + m_o r_o \sin \phi_o = (m^* + m_o) r \sin \phi \quad (4.22a)$$

$$m^* r^* \cos \phi^* + m_o r_o \cos \phi_o = (m^* + m_o) r \cos \phi \quad (4.22b)$$

Puisque nous désirons un ensemble in-line,  $\phi = 0$  (si  $r > 0$ ) ou  $\phi = \pi$  (si  $r < 0$ ). Dans les deux cas  $\sin \phi = 0$ . La relation (4.22a) devient :

$$m^* r^* = - m_o r_o \sin \phi_o / \sin \phi^* \quad (4.23)$$

En remplaçant (4.23) dans (4.22b), on déduit :

$$m^* = (1/r \cos \phi) [(-m_o r_o \sin \phi_o / \tan \phi^*) + m_o r_o \cos \phi_o - m_o r \cos \phi] \quad (4.24)$$

Si nous fixons donc la position angulaire ( $\phi^*$ ) du contre-poids, la nouvelle position du centre de gravité ( $r$ ;  $\phi = 0$  si  $r > 0$ ,  $\phi = \pi$  si  $r < 0$ ) alors la relation (4.24) donne la masse ( $m^*$ ) du contre-poids. Connaissant alors la masse ( $m^*$ ) on peut grâce à la relation (4.23) déterminer ( $r^*$ ) la position du centre de gravité du contrepoids.

#### 4.2.2 Le Pendule physique:

Une membrure in-line est un pendule physique, si lorsque suspendu à chacun de ses pivots, elle a la même fréquence d'oscillation égale à celle d'un pendule simple (masse concentrée au bout d'un fil) de même masse que la membrure et de longueur égale à l'entraxe (distance entre les pivots).

Notons respectivement  $I$  et  $I'$  les moments d'inertie par

rapport aux pivots A et B. Désignons par M la masse du pendule physique.

$$I = I_g + Mr^2 \quad (4.25a)$$

$$I' = I_g + Mr'^2 \quad (4.25b)$$

or  $I_g = M.k^2$ , (k étant le rayon de giration),

d'où nous tirons:  $I = M.(k^2 + r^2)$

$$\text{et } I' = M.(k^2 + r'^2)$$

En les combinant nous obtenons:  $I/I' = (k^2+r^2)/(k^2+r'^2)$  (4.26)

Les équations différentielles caractérisant les oscillations autour des pivots A et B sont respectivement:

$$\alpha + (Mgr/I)\sin\theta = 0 \quad (4.27a)$$

$$\alpha + (Mgr'/I')\sin\theta = 0 \quad (4.27b)$$

Si la première condition de l'énoncé ci-haut (même fréquence d'oscillation lorsque fixée à chacun de ses pivots) est satisfaite, des relations (4.27a) et (4.27b) nous obtenons:

$$r/I = r'/I' \Rightarrow I/I' = r/r' \quad (4.28)$$

En combinant (4.28) et (4.26) on obtient:  $k^2 = rr'$  (4.29).

La relation (4.29) est caractéristique du pendule physique; nous aurons à l'utiliser dans la suite.

La deuxième condition de l'énoncé susmentionnée (même fréquence d'oscillation que le pendule simple idéal tel que défini) permet de déterminer la nouvelle configuration à donner à une membrure pour laquelle devienne un pendule physique. On peut aussi être amené dans certains cas à ajouter des masses adéquatement calculées à la membrure.

4.2.2.1 Cas général: une membrure in-line quelconque à rendre pendule physique par addition de masses.

Considérons la figure ci-dessous: figure 4.6. Désignons par  $k_{g*}$  et  $k'_{g*}$  les rayons de giration des masses augmentées  $m_*$  et  $m'_{*}$ ;  $r_{g*}$  et  $r'_{g*}$  sont des positions définies telles que montrées sur la figure. Nous appelons  $m$  la masse initiale de la membrure avant toute reconfiguration ou addition de contrepoids.  $M$  désigne la masse finale (c'est à dire la masse du pendule physique).

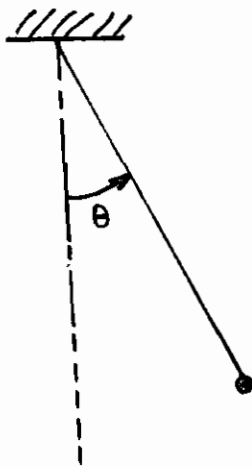


Figure 4.5 Pendule simple

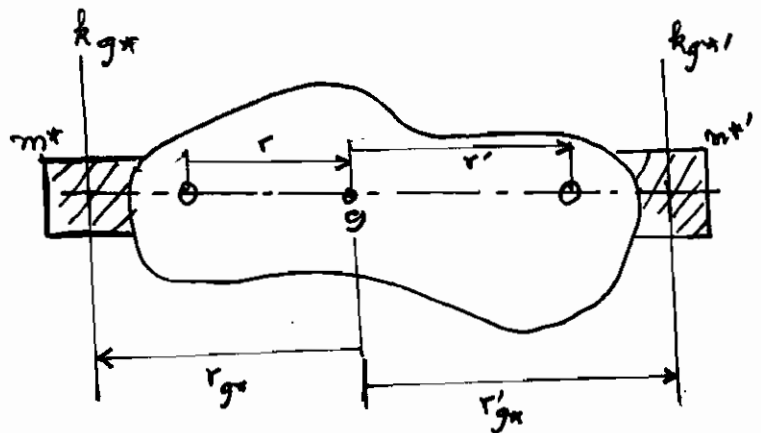


Figure 4.6 Pendule physique: cas général.

$$\text{Posons } A = k_g^2 + r_g^2 \text{ et } B = k'_{g^2} + r'_{g^2} \quad (4.30)$$

Réécrivons les relations (4.25a) et (4.25b) sous la forme:

$$I = I_{go} + (m_* + m'_{*} + m)r^2 + Am_* + Bm'_{*} \quad (4.31a)$$

$$I' = I_{go} + (m_* + m'_{*} + m)r'^2 + Am_* + Bm'_{*} \quad (4.31b)$$

En combinant (4.28), (4.31a) et (4.31b), on obtient l'équation:

$$m_*(r'r - A) + m'_*(r'r - B) = (I_{g_0} - mr'r) \quad (4.32)$$

En convenant que l'ajout des deux masses  $m_*$  et  $m'_*$  ne modifie pas la position initiale du centre de gravité, on obtient l'équation suivante:

$$m_*r_{g_*} - m'_*r'_{g_*} = 0 \quad (4.33)$$

Les équations (4.32) et (4.33) forment un système nous permettant de déterminer  $m_*$  et  $m'_*$  (il faudra évidemment fixer tous les paramètres décrits ci-haut).

#### 4.2.2.2 1<sup>er</sup> cas particulier : barre rectangulaire

La figure 4.7 représente une barre rectangulaire pour laquelle nous allons donner la relation fondamentale devant lier ses caractéristiques pour qu'elle soit un pendule physique.

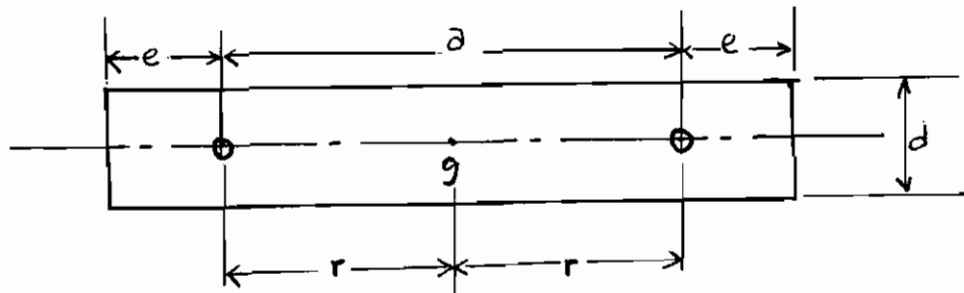


Figure 4.7 Pendule physique: barre rectangulaire.

En suivant la même procédure que dans le cas précédent, on établit aisément la relation suivante définissant les conditions dans lesquelles une barre rectangulaire est un pendule physique:

$$e/d = 0.5[ 3(a/d)^2 - 1 ]^{1/2} - 0.5a/d \quad (4.34)$$

4.2.2.3 2ème Cas particulier: Augmented link

L'objectif est le même qu'au premier cas particulier. Nous avons représenté à la figure 4.8 ce nous que désignons par le thème "augmented link".

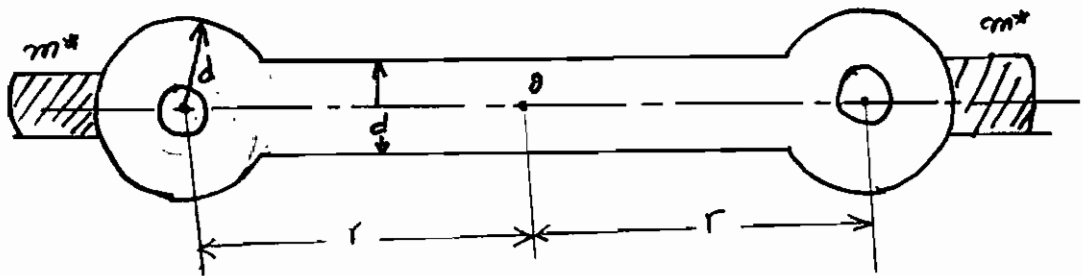


Figure 4.8 Pendule physique: augmented link.

De même on démontre assez aisément la relation fondamentale qui fait d'un augmented link un pendule physique:

$$A(e/d)^3 + B(e/d)^2 + C(e/d) + D = 0 \quad (4.35)$$

où  $A = 8$

$$B = 12(a/d) + 24$$

$$C = 24(a/d) + 26$$

$$D = -2(a/d)^3 + 13(a/d) + 12\pi - 10$$

La relation (4.35) peut aussi être exprimée sous la forme :

$$\alpha(a/d)^3 + B(a/d) + \Gamma = 0 \quad (4.36)$$

où  $\alpha = -2$

$$B = 12(e/d)^2 + 24(e/d) + 13$$

$$\Gamma = 8(e/d)^3 + 24(e/d)^2 + 26(e/d) + 12\pi - 10.$$

Ces considérations sur la membrure in-line et sur le pendule physique nous seront d'une grande utilité pour la suite.

4.2.3 Équilibrage d'un mécanismes à quatre membrures dont le coupleur est une in-line et un pendule physique

Le coupleur (membrure 3) étant une membrure "in line" - i.e  $\phi_3$  égal 0 ou  $\pi$  (modulo  $2\pi$ ) - l'équation (4.20) devient:

$$M_{M/G} = -\sum m_i (k_i^2 + r_i^2 - a_i r_i \cos \phi_i) \alpha_i \quad (4.37)$$

La membrure 3 Étant également un pendule physique on a:

$$k_3^2 = r_3 r'_3 \quad (4.38)$$

$$k_3^2 + r_3^2 - a_3 r_3 \cos \phi_3 = r_3 r'_3 + r_3^2 - (r_3 + r'_3) r_3 = 0 \quad (4.39)$$

La relation (4.37) devient alors:

$$M_{M/G} = -m_2 (k_2^2 + r_2^2 - a_2 r_2 \cos \phi_2) \alpha_2 - m_4 (k_4^2 + r_4^2 - a_4 r_4 \cos \phi_4) \alpha_4 \quad (4.40)$$

$$\text{Posons } K_2 = -m_2 (k_2^2 + r_2^2 - a_2 r_2 \cos \phi_2) \alpha_2 \quad (4.41a)$$

$$\text{et } K_4 = -m_4 (k_4^2 + r_4^2 - a_4 r_4 \cos \phi_4) \alpha_4 \quad (4.41b)$$

Réécrivons donc la relation (4.40):

$$M_{M/G} = K_2 \alpha_2 + K_4 \alpha_4 \quad (4.42)$$

L'équilibrage des moments sera donc obtenu dès qu'on additionnera à la membrure 2 et à la membrure 4 des "contre-poids d'inertie" de moment d'inertie  $I_2^{**} = -K_2$  et  $I_4^{**} = -K_4$  respectivement.

En résumé pour l'équilibrage complet d'un mécanisme à quatre membrures dont toutes les longueurs sont finies, les conditions suivantes doivent être satisfaites:

$$m_2 r_2 = m_3 r'_3 a_2 / a_3 \quad \text{et} \quad \phi_2 = \phi'_3 = \pi \quad (4.43)$$

$$m_4 r_4 = m_3 r_3 a_4 / a_3 \quad \text{et} \quad \phi_4 = \phi_3 + \pi = \pi \quad (4.44)$$



$$k_3^2 = r_3 r'_3 \quad (4.45)$$

$$I_2^{**} = -K_2 = m_2(k_2^2 + r_2^2 + a_2 r_2) \quad (4.46)$$

$$I_4^{**} = -k_4 = m_4(k_4^2 + r_4^2 + a_4 r_4) \quad (4.47)$$

En effet, les quatre chapitres que nous venons d'étudier constituent le socle -la base- de notre projet. Nous nous y sommes à dessein attardés puisque que, sans une bonne maîtrise de la théorie du projet à informatiser, rien de sérieux ne peut être fait. L'on trouvera présenté dans l'annexe A les organigrammes des modules du mini-logiciel, dans l'annexe B quelques exemples d'exécution et dans l'annexe C le listing du programme informatique.

## RECOMMANDATIONS

Dans le premier, deuxième et quatrième chapitre, nous nous étions limités à l'étude de deux types de mécanismes plans à quatre membrures: ceux dont toutes les membrures ont une longueur finie et ceux à manivelle-coulisseau. Quoique ces deux types soient les plus utilisés, il serait profitable d'envisager pour l'avenir une extension du travail à un éventail plus large de mécanismes plans à quatre membrures et autres (les mécanismes à six membrures par exemple).

L'on devra également envisager compléter l'analyse dynamique des rotors par la calcul des volants d'inertie.

D'autre part, l'outil informatique réalisé devra être perfectionné pour répondre aux critères d'un véritable logiciel (entre autres: permettre des sorties temporaires, contrôler les opérations d'entrées/sorties) et pour faciliter son utilisation (permettre le choix des unités des paramètres, donner des explications grâce a un menu d'aide, etc).

## CONCLUSION

Comme nous venons de le voir, ce projet comprend:

- l'étape de l'étude théorique de l'analyse dynamique des mécanismes: véritable travail de mécanicien;
- l'étape de l'informatisation: travail de programmeur.

Les deux étapes ne peuvent évidemment être menées séparément car les méthodes utilisées dans l'une sont fonctions des moyens dont l'autre et vice versa.

Nous nous sommes donc efforcés d'allier les deux qualités de mécanicien et de programmeur pour aboutir à ce mini-projet d'aide à l'analyse dynamique des mécanismes. Il est en mesure:

- de déterminer les forces aux joints et le couple d'entrée à partir des données cinétiques;
- de déterminer l'accélération d'entrée à partir du couple d'entrée et,
- d'équilibrer les rotors et les mécanismes à quatre membrures.

Nous voici donc au bout d'un véritable parcours du combattant (fait de nombreuses nuits blanches pour raison de surcharge de l'emploi du temps). Mais en réalité, il s'agit plutôt d'un nouveau départ pour continuer le travail d'une manière ou d'une autre.

## BIBLIOGRAPHIE

- Ferdinand P. Beer & E. Russell Johnston, Jr  
"Mécanique à l'usage des ingénieurs: STATIQUE"  
Mc Graw-hill Editeurs, 1981.
  
- Ferdinand P. Beer & E. Russell Johnston, Jr  
"Vector Mechanics for Engineers: DYNAMICS"  
Mc Graw-hill, 1984.
  
- Hamilton H. Mabie and Fred W. Ocvirk  
"Mechanisms and Dynamics of Machinery"  
John Wiley & Sons, 1978.
  
- Chung Ha SUH & Charles W. RADCUFFE  
"Kinematics & Mechanisms Design"  
R. E. KRIEGER Publishing company, 1983.
  
- I. ARTOBOLEVSKI  
"Théorie des mécanismes et des machines"  
EDITIONS MIR MOSCOU, 1977.

-G. N. SANDOR & A. G. ERDMAN

"Advanced mechanism design: Analysis and Synthesis"

Volume 1 & 2

Prentice- hall, Inc 1984.

-Douglas HERGERT

"Turbo pascal 5 par la pratique"

SYBEX, 1989.

- J-C. BERNARD, R. GUARDO, J. LAVOIE et P. SARVARD

"Notes de cours 3.307: Informatique I"

Ecole Polytechnique de Montréal, 1988.



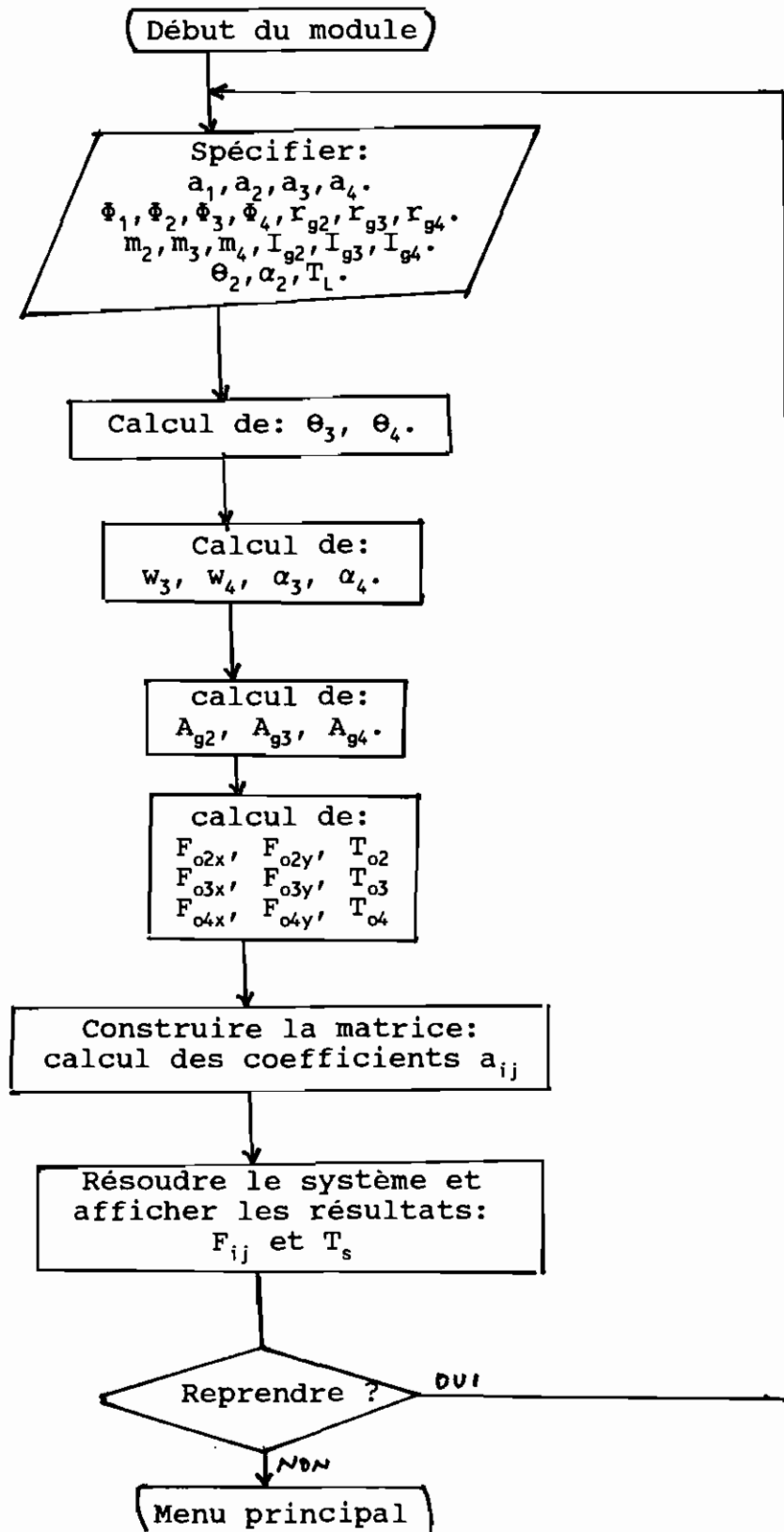
## **ANNEXES**

**ANNEXE A**

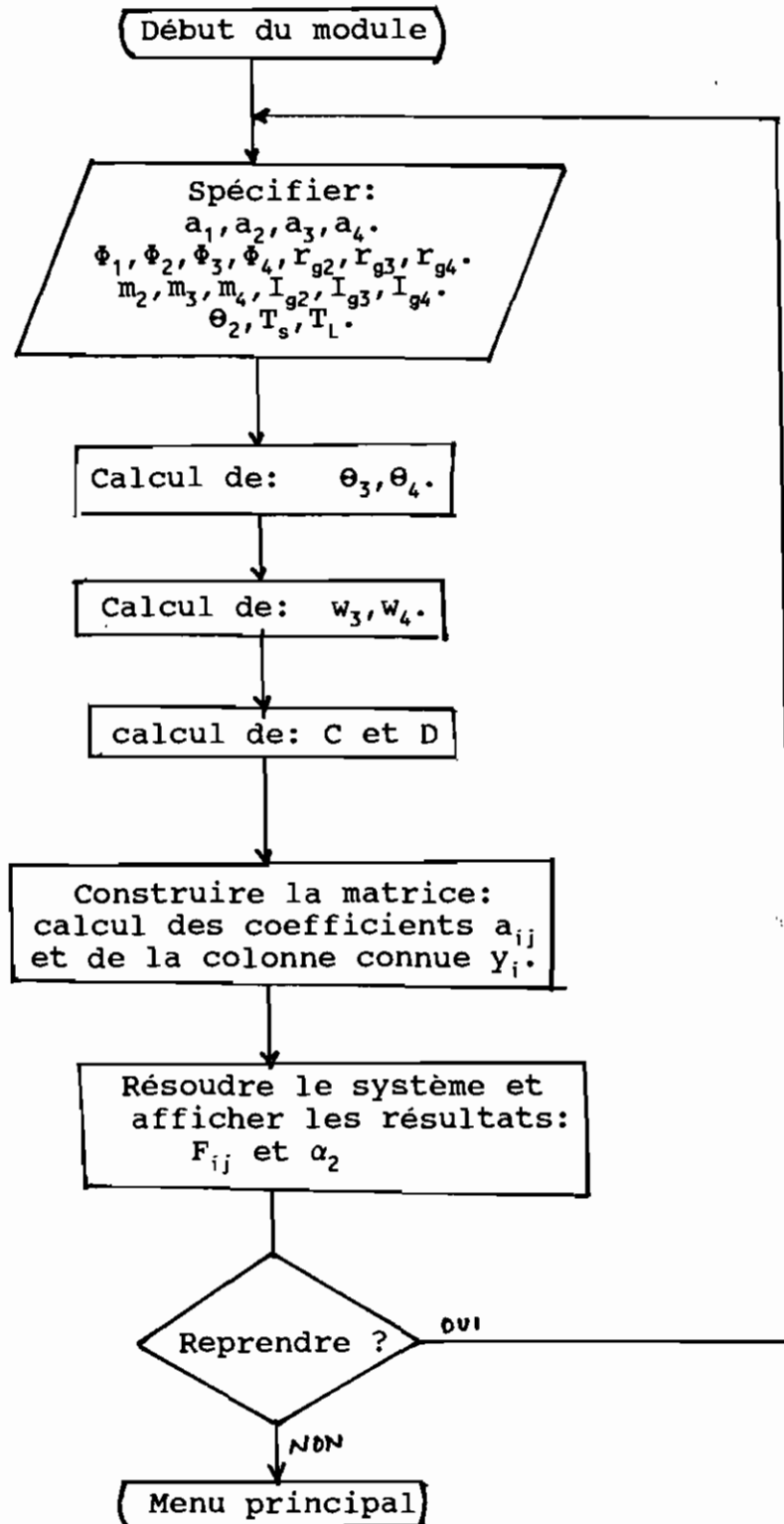
**ORGANIGRAMMES**



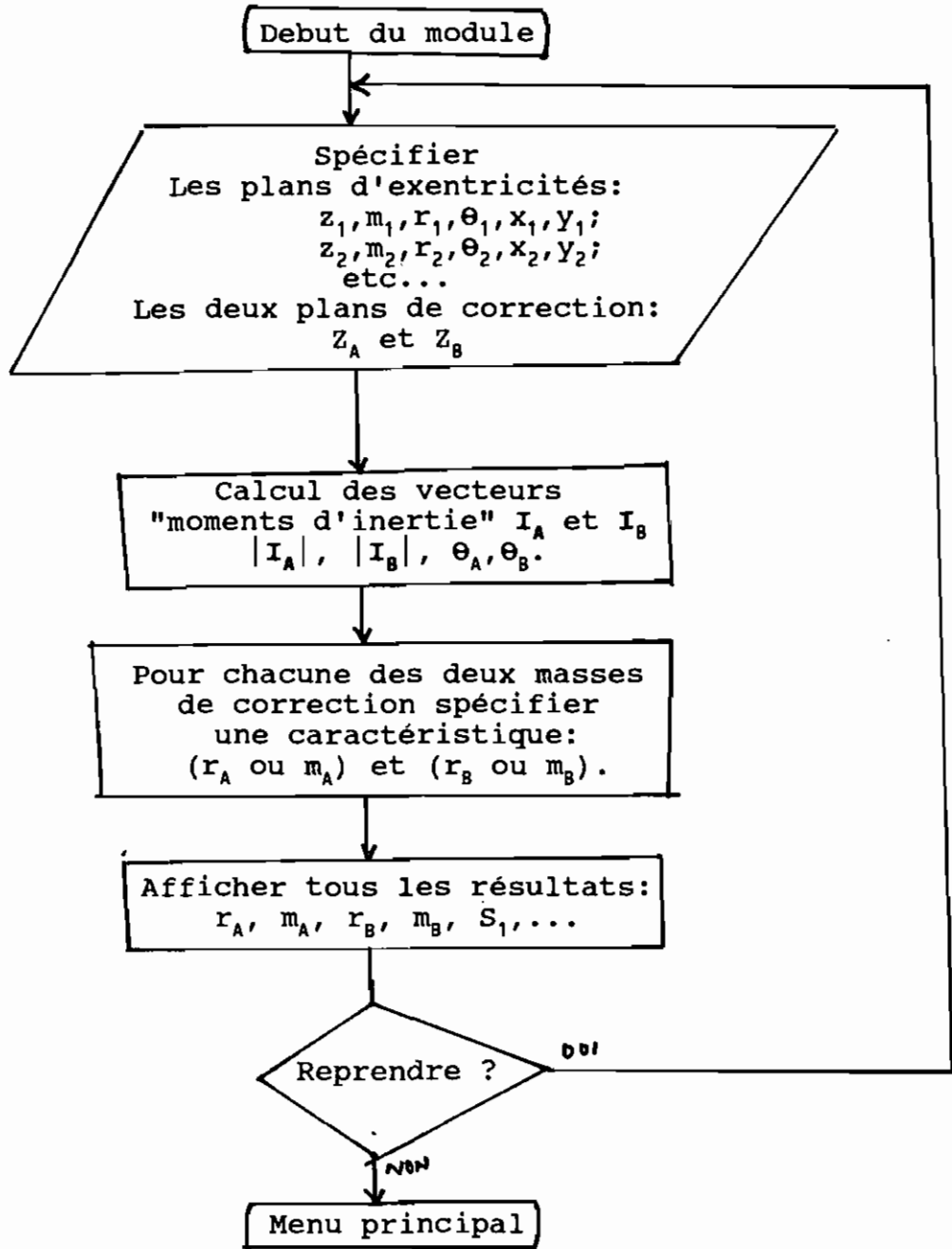
# A-1 ANALYSE CINETO-STATIQUE



## A-2 ANALYSE DE LA REPONSE TEMPORELLE

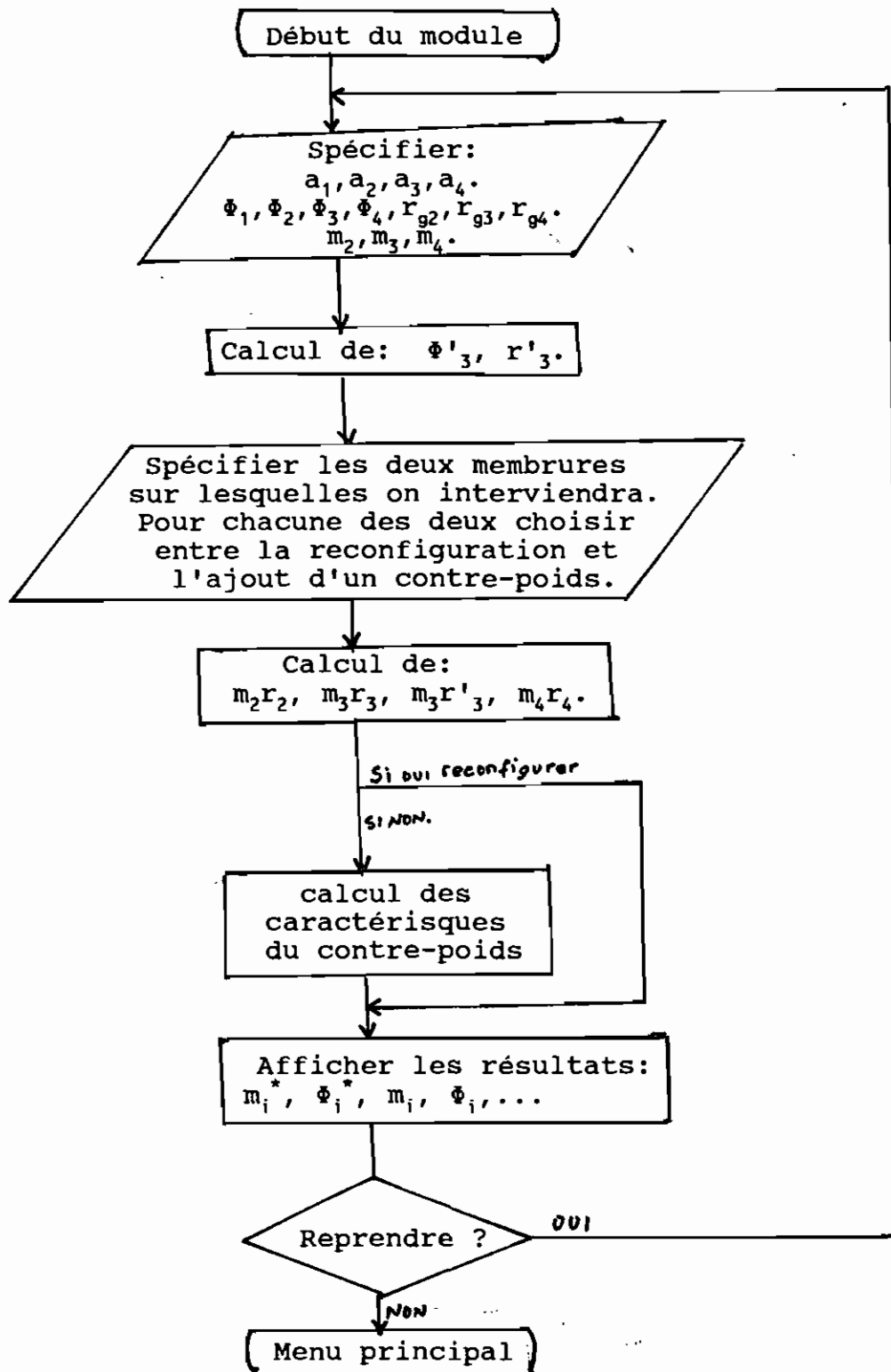


### A-3 Equilibrage dynamique des rotors

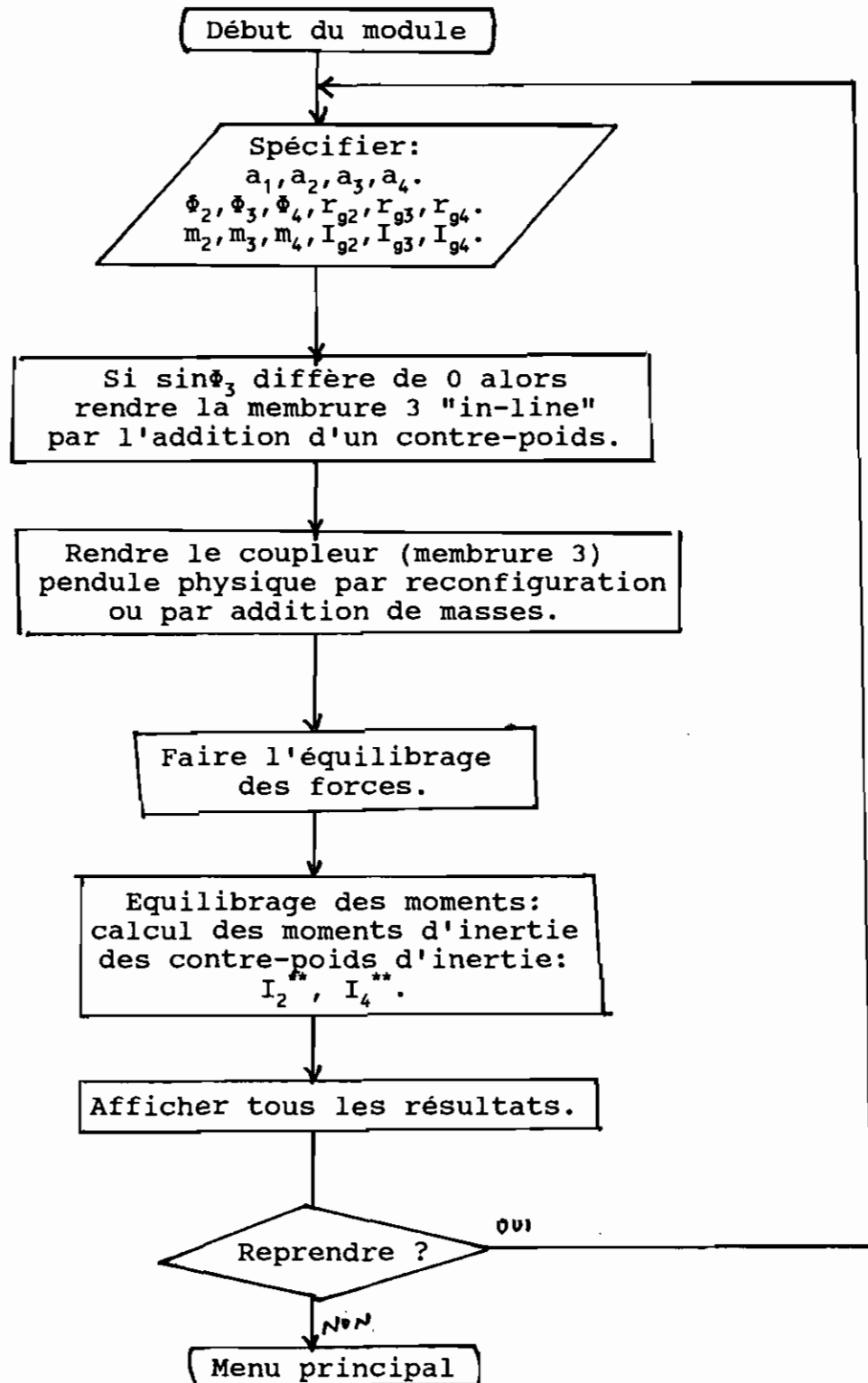


En ce qui concerne l'équilibrage statique, il suffira de supprimer la coordonnée "z" de l'organigramme précédent et de ne considérer qu'une masse de correction.

A-4 Equilibrage des forces des mécanismes à quatre membrures



A-5 Equilibrage des forces et des moments mécanismes à quatre membrures (équilibrage complet)



**ANNEXE B**

**EXEMPLES D'EXÉCUTION DU LOGICIEL**

**B-1 EXEMPLE D'EXÉCUTION 1: mécanisme à quatre membrures de longueurs finies: analyse cinéto-statique.**

Il est donné à la figure B.1 un mécanisme à quatre membrures dont les caractéristiques physiques sont indiquées.

Il est demandé de faire:

- l'analyse cinéto-statique (détermination des forces aux joints et du couple d'entrée) sachant que la membrure d'entrée a une accélération nulle.

- l'analyse de la réponse temporelle (détermination de l'accélération d'entrée et des forces aux joints) sachant cette fois-ci que le couple d'entrée est égal à 10 N.m.

$M_2 = 4.53 \text{ kg}$	$I_2 = 0.023 \text{ kg.m}^2$
$M_3 = 1.81 \text{ kg}$	$I_3 = 0.008 \text{ kg.m}^2$
$M_4 = 3.63 \text{ kg}$	$I_4 = 0.035 \text{ kg.m}^2$

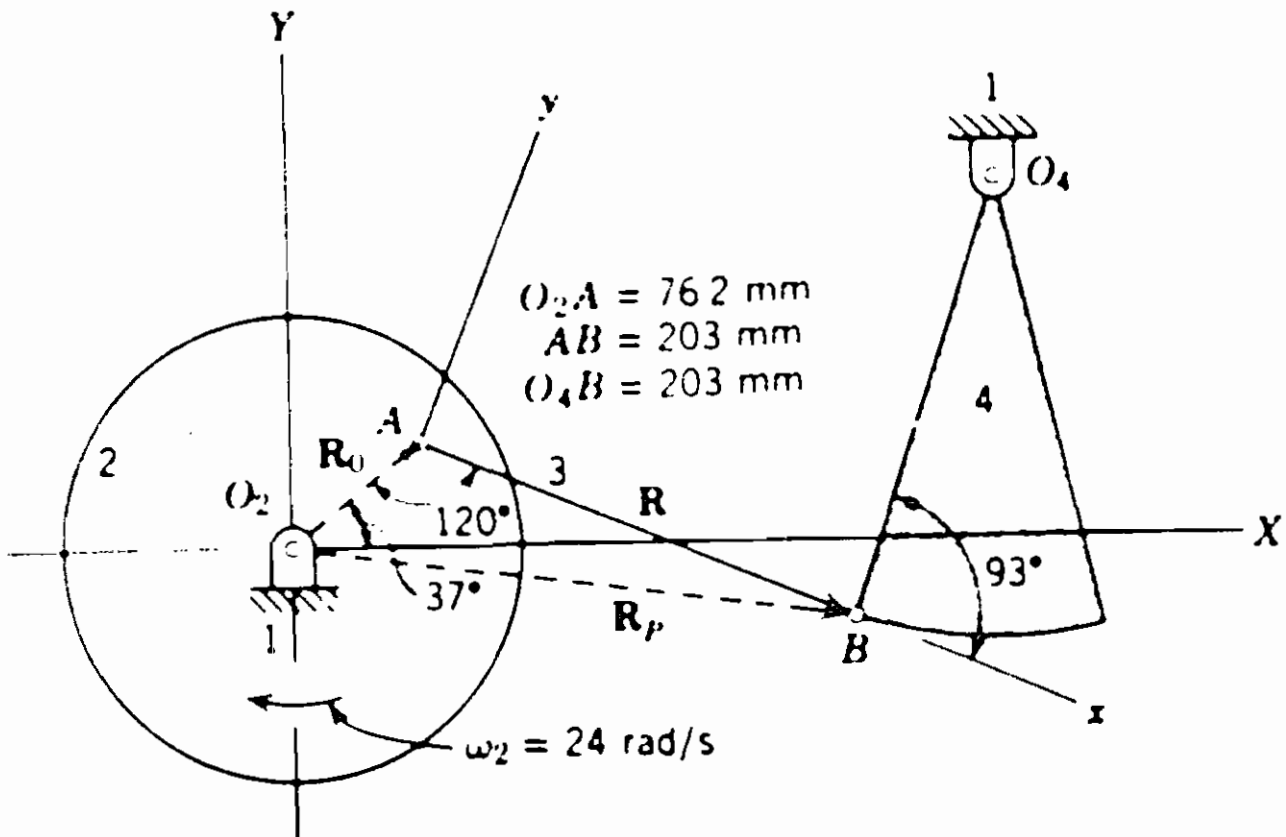


Figure B-1 Exemple 1, mécanisme à quatre membrures de longueurs finies: analyse cinéto-statique.

A:\>ADM4M

Choisir l'option désirée

Analyse cinéto-statique  
Analyse de la réponse temporelle  
Equilibrage des Rotors  
Equilibrage des forces et des moments

Choisir la sous-option désirée

Mécanismes de longueurs finies  
Mécanismes manivelle-coulisseau

connaissez-vous  $\psi_1$  et  $\phi_1$  ? (O,N) :

Entrer les caractéristiques du mécanisme.

a2	0.0762
r3	0.2030
a4	0.2030
$\theta_2$	0.6458
$\theta_3$	5.8818
$\theta_4$	4.3633

R1 = 0.3540

$\psi_1$  = 0.4601

Tapez une touche pour continuer.



Entrer les caractéristiques du mécanisme.

a1	0.3540
a2	0.0762
a3	0.2030
a4	0.2030
$\phi 1$	0.4604
$\phi 2$	0.0000
$\phi 3$	0.0000
$\phi 4$	0.2618
rg2	0.0000
rg3	0.1015
rg4	0.1340
m2	1.5300
m3	1.8100
m4	3.6300
Ig2	0.0230
Ig3	0.0080
Ig4	0.0350

Entrer les caractéristiques du mécanisme.

$\theta 2$	0.6458
$\omega 2$	-24.0000
$\alpha 2$	0.0000
$\tau 1$	0.0000

Choisissez  $\theta 3$  entre : 1.222 5.882

ANALYSE CINETO-STATIQUE: N4N longueurs finies.

Caractéristiques cinétiques.

Paramètres	memb2	memb3	memb4
$\theta_i$	0.646	5.882	4.3633
$w_i$	-24.000	4.913	7.813
$a_i$	0.000	241.446	-129.214
Agix	0.000	-27.731	-16.535
Agiy	0.000	-2.900	9.658
!Agil	0.000	27.882	19.149
$\beta(Ag)$	0.000	3.246	2.613

Forces dynamiques au centre de gravité.

Paramètres	memb2	memb3	memb4
Foix	0.000	50.193	60.0238
Foiy	0.000	5.248	-35.059
Toi	0.000	-1.932	4.522

Forces dynamiques aux joints et couple d'entree.

Voulez-vous imprimer les resultats ? (O,N) :

F12x =	-108.0432
F12y =	22.2507
Ts =	6.3089
F23x =	-108.0432
F23y =	22.2507
F34x =	-57.8501
F34y =	27.4990
F14x =	-2.1737
F14y =	7.5599

Taper ESC pour sortir du se laie ou CR pour retourner changer les valeurs.

Choisir l'option désirée

Analyse cinéto-statique  
Analyse de la réponse temporelle  
Equilibrage des Rotors  
Equilibrage des forces et des moments

Choisir la sous-option désirée

Mécanismes de longueurs finies  
Mécanismes manivelle-coulisseau

connaissez-vous  $\psi_{s1}$  et  $\phi_1$  ? (O,N) :

Entrer les caractéristiques du mécanisme.

a1	0.3540
a2	0.0762
a3	0.2030
a4	0.2030
$\phi_1$	0.4604
$\phi_2$	0.0000
$\phi_3$	0.0000
$\phi_4$	0.2618
rg2	0.0000
rg3	0.1015
rg4	0.1340
m2	4.5300
m3	1.8100
m4	3.6300
Ig2	0.0230
Ig3	0.0080
Ig4	0.0350

Entrer les caractéristiques du mécanisme.

$\theta_2$	0.6458
w2	-24.0000
Is	10.0000
T1	0.0000

Choisissez  $\theta_3$  entre : 1.222 5.882

ANALYSE DE LA REPONSE TENPORELLE: N4M longueurs finies.

Forces dynamiques aux joints et  $\alpha_2$ .

F12x	=	-131.0158
F12y	=	26.9190
$\alpha_2$	=	102.3243
F23x	=	-131.0158
F23y	=	26.9190
F34x	=	-70.8266
F34y	=	39.8966
F14x	=	-5.3350
F14y	=	-3.4268

Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.

Forces dynamiques aux joints et  $a_2$ .

F12x	=	-131.0158
F12y	=	26.9190
$a_2$	=	102.3243
F23x	=	-131.0158
F23y	=	26.9190
F34x	=	-70.8266
F34y	=	39.8966
F14x	=	-5.3350
F14y	=	-3.4268

Taper ESC pour sortir du logiciel  
ou CR pour retourner changer les valeurs.

**B-2 EXEMPLE D'EXÉCUTION 2: mécanisme à quatre membrures manivelle-coulisseau.**

Il est donné à la figure B.2 un mécanisme à quatre membrures manivelle-coulisseau dont les caractéristiques physiques sont indiquées. La membrure entraînante est supposée être la manivelle.

Il est demandé de faire:

- l'analyse cinéto-statique (détermination des forces aux joints et du couple d'entrée) sachant que la membrure d'entrée a une vitesse angulaire de 200 rad/s et une accélération nulle.

- l'équilibrage des forces du mécanisme en disposant sur la manivelle un contrepoids (dont la position du centre de gravité sera  $r_2^* = 0.030$  m) et en modifiant éventuellement la configuration de la bielle (dont la masse sera ramenée à 2 kg).

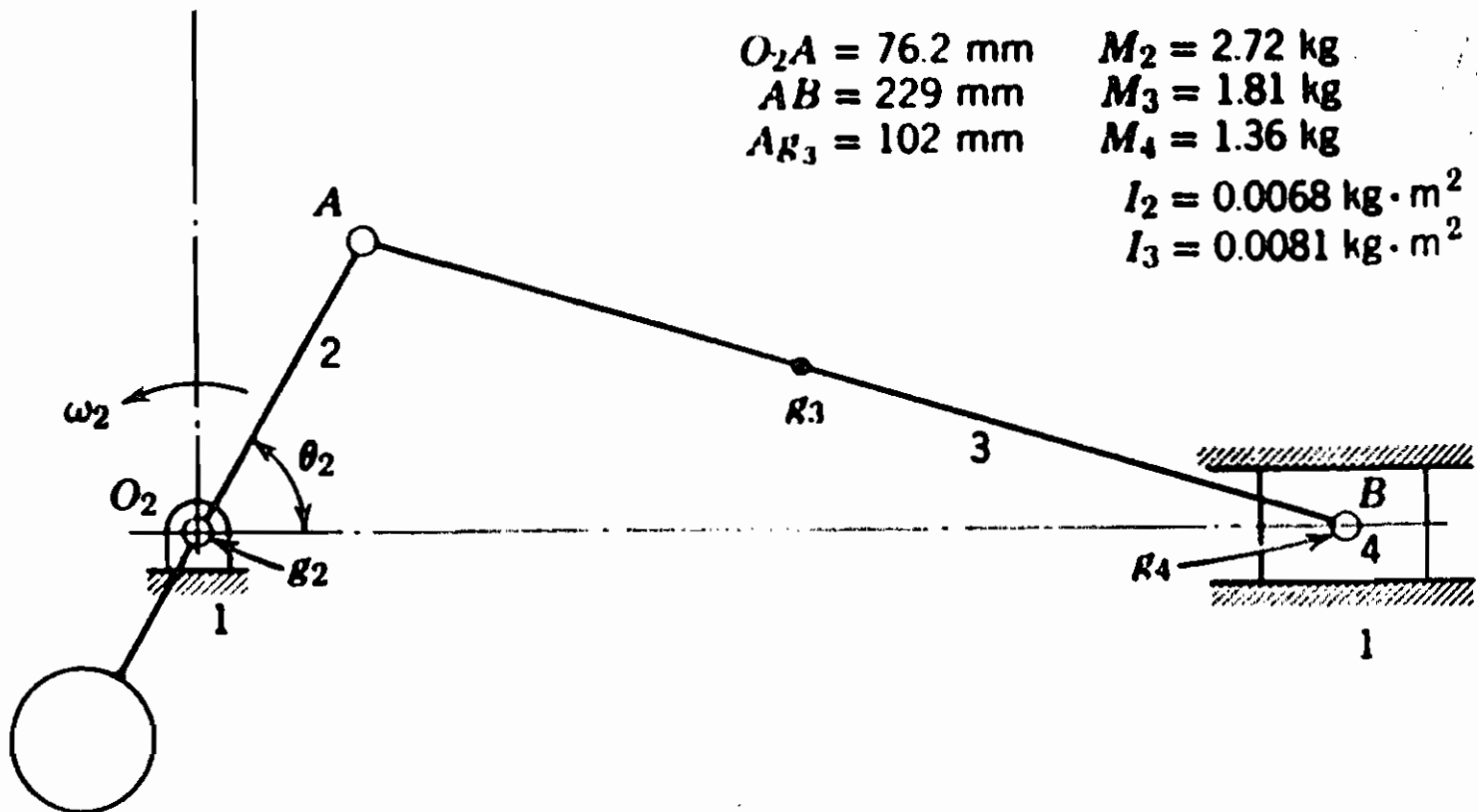


Figure B-2 Exemple 2, mécanisme manivelle-coulisseau: analyse cinéto-statique et équilibrage des forces.

Choisir l'option désirée

Analyse cinéto-statique  
Analyse de la réponse temporelle  
Equilibrage des Rotors  
Equilibrage des forces et des moments

Choisir la sous-option désirée

Mécanismes de longueurs finies  
Mécanismes manivelle-coulisseau

Entrer les caractéristiques du mécanisme.

a2	0.0762
a3	0.2290
upset e	0.0000
$\phi$	0.0000
$\phi 2$	0.0000
$\phi 3$	0.0000
rg2	0.0000
rg3	0.1020
m2	2.7200
m3	1.8100
m4	1.3600
Ig2	0.0068
Ig3	0.0081

Entrer les caractéristiques du mécanisme.

$\theta 2$	1.0472
$\omega 2$	200.0000
$\alpha 2$	0.0000
F1	0.0000

Choisissez  $\theta 3$  entre : -9.132 -0.202

Caractéristiques cinétiques.

Paramètres	memb2	memb3	memb4
$\theta_i$	1.047	-0.292	-1.5708
$w_i$	200.000	-34.749	0
$a_i$	0.000	11674.094	0
VC	-	-	15.491
Agix	0.000	-1298.799	-1018.402
Agiy	0.000	-1463.908	0.000
!Agil	0.000	1957.015	1018.402
$\beta$ (Ag)	0.000	3.987	3.142

Voulez-vous imprimer les résultats ? (O,N) :

Forces dynamiques au centre de gravité.

Paramètres	memb2	memb3	memb4
Foix	0.000	2350.827	1385.0270
Foiy	0.000	2649.674	0.000
Foi	0.000	3542.197	-1385.027
Toi	0.000	-94.560	0

Forces dynamiques aux joints.

F12x =	-3735.8536
F12y =	-1168.7742
Ts =	202.0029
F23x =	-3735.8536
F23y =	-1168.7742
F34x =	-1385.0270
F34y =	1480.8996



Choisir l'option désirée

Analyse cinéto-statique  
Analyse de la réponse temporelle  
Equilibrage des Rotors  
Equilibrage des forces et des moments

Choisir la sous-option désirée

Equilibrage des forces uniquement  
Equilibrage complet (forces et moments)

Choisir le type de mécanisme à équilibrer

Mécanismes de longueurs finies  
Mécanismes manivelle-coulisseau

Entrer les caractéristiques du mécanisme.

a2	0.0762
a3	0.2290
upset e	0.0000
$\phi$	0.0000
$\phi^{\circ}2$	0.0000
$\phi^{\circ}3$	0.0000
$r^{\circ}2$	0.0000
$r^{\circ}3$	0.1020
$m^{\circ}2$	2.7200
$m^{\circ}3$	1.8100
$m^{\circ}4$	1.3600

Donnez la première membrure dont les caractéristiques seront modifiées.

Membrure 2 Membrure 3 Membrure 4

Donnez la deuxième membrure dont les caractéristiques seront modifiées.

Membrure 2 Membrure 3 Membrure 4

Pour la membrure 2, choisissez Reconfiguration Contrepoids

Lequel des paramètres voulez-vous fixer ? : m3 r3 r'3

Entrez m3 [ 0.000 ] : 2

Fixer un paramètre du contrepoids 2 : m\*2 r\*2

Entrez r\*2 [ 0.000 ] : .030

paramètres initiaux

Paramètres	memb1	memb2	memb3	memb4
$\bar{a}$	0.000	0.076	0.229	0.000
$r^{\circ}$	-	0.000	0.102	0.000
$\phi^{\circ}$	0.000	0.000	0.000	0.000
$r^{\circ'}$	-	-	0.127	-
$\phi^{\circ'}$	-	-	3.142	-
$m^{\circ}$	-	2.720	1.810	1.360

Paramètres pour le balancement des forces

Paramètres	memb2	memb3	memb4
$\phi$	3.142	0.000	0.000
$m^r$	0.049	0.311	0.000
$r$	0.011	0.156	0.000
$m$	4.346	2.000	1.360
$\phi'$	-	3.142	-
$m^r'$	-	0.147	-
$r'$	-	0.073	-
$\phi^*$	3.142	0.000	0.000
$m^r^*$	0.049	0.000	0.000
$m^*$	1.626	0.000	0.000
$r^*$	0.030	0.000	0.000
$r'^*$	-	0.000	-

Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.

**B-3 EXEMPLE D'EXÉCUTION 3: mécanisme à quatre membrures de longueurs finies, équilibrage complet.**

Il est donné à la figure B.3 un mécanisme à quatre membrures dont toutes les membrures ont une longueur finie. Leurs caractéristiques physiques sont définies sur la figure.

Il est demandé de réaliser l'équilibrage complet (forces et moments) de ce mécanisme: en additionnant des contre-poids de forces et d'inerties aux membrures 2 et 4.

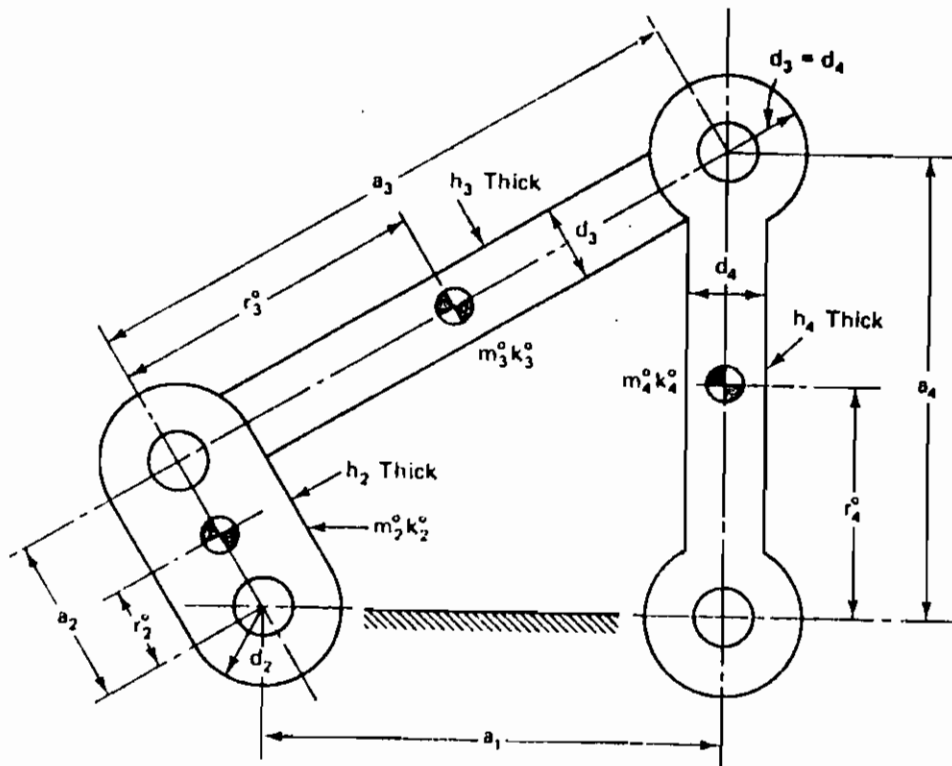


Figure B-3 Exemple 3, mécanisme à quatre membrures: équilibrage complet.

Choisir l'option désirée

Analyse cinéto-statique  
Analyse de la réponse temporelle  
Equilibrage des Rotors  
Equilibrage des forces et des moments

Choisir la sous-option désirée

Equilibrage des forces uniquement  
Equilibrage complet (forces et moments)

Entrer les caractéristiques du mécanisme.

a1	3.0000
a2	1.0000
a3	4.0000
a4	3.0000
$\phi^{\circ}1$	0.0000
$\phi^{\circ}2$	0.0000
$\phi^{\circ}3$	0.0000
$\phi^{\circ}4$	0.0000
$r^{\circ}2$	0.5000
$r^{\circ}3$	2.0000
$r^{\circ}4$	1.5000
$m^{\circ}2$	0.1010
$m^{\circ}3$	0.1740
$m^{\circ}4$	0.1460

Entrer les caractéristiques du mécanisme.

mv2	0.2830
mv3	0.2830
mv4	0.2830
Ig $^{\circ}2$	0.0355
Ig $^{\circ}3$	0.0000
Ig $^{\circ}4$	0.2322

Conversion du coupleur(3) en pendule physique.Choisissez.

Barre rectangulaire  
"Augmented link" telle que définie  
Autre " Augmented link "

Fixez un couple de paramètres.

e3-mo3      e3-h3      d3-mo3      d3-h3      mo3-h3

Entrer la largeur "d3" : [ 0.000 ] : .5

Entrer l'épaisseur "h3" : [ 0.000 ] : .2

Le débordement e3 est : 0.9434

La masse mo3 est : 0.2272

Tapez O si valeurs ok sinon N

Pour la membrure 2, choisissez Reconfiguration Contrepoids

Pour la membrure 4, choisissez Reconfiguration Contrepoids

Fixer un parametre du contrepoids 2 : m\*2      r\*2

Entrez r\*2 [ 0.000 ] : .717

Choisissez le parametre du contrepoids fixé à 4 ? : m\*4      r\*4

Entrez m\*4 [ 0.000 ] : .516

paramètres initiaux

Paramètres	memb1	memb2	memb3	memb4
a	3.000	1.000	4.000	3.000
r <sup>o</sup>	-	0.500	2.000	1.500
φ <sup>o</sup>	0.000	0.000	0.000	0.000
r <sup>o'</sup>	-	-	2.000	-
φ <sup>o'</sup>	-	-	3.142	-
m <sup>o</sup>	-	0.101	0.227	0.146

Paramètres pour le balancement des forces

Paramètres	memb2	memb3	memb4
φ	3.142	0.000	3.142
mr	0.087	0.348	0.261
r	0.297	2.000	0.394
m	0.293	0.174	0.662
φ'	-	3.142	-
mr'	-	0.348	-
r'	-	2.000	-
φ*	3.142	0.000	3.142
m <sup>o</sup> r*	0.138	0.000	0.480
m*	0.192	0.000	0.516
r*	0.717	0.000	0.930
r'*	-	0.000	-

Donner le rayon de giration du C-P fixé à "2" : [ 0.000 ] : .5

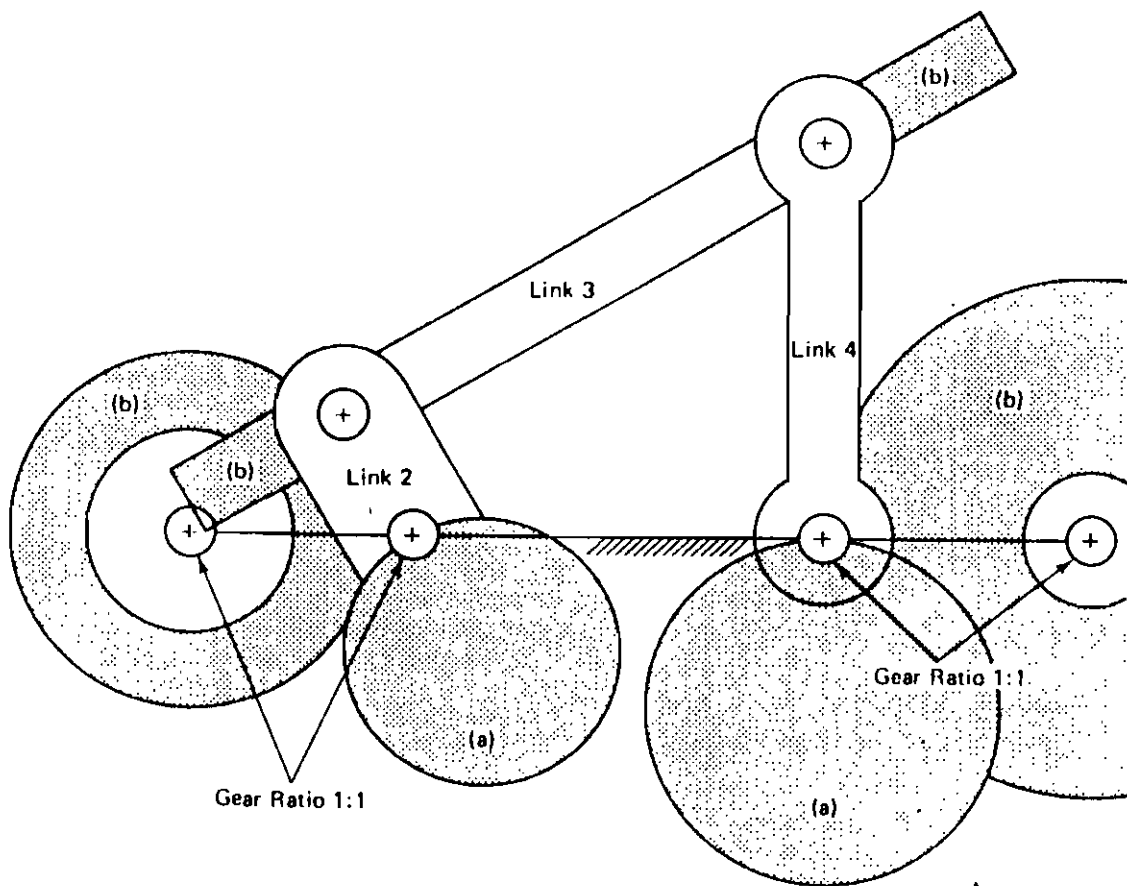
Donner le rayon de giration du C-P fixé à "4" : [ 0.000 ] : .5

Paramètres pour le balancement des moment

Paramètres	memb2	memb3	memb4
I cp-force	0.048	-	0.129
k cp-force	0.500	-	0.500
I memb+cp f	0.112	0.696	0.969
k memb+cp f	0.620	2.000	1.210
I cp-inertie	0.225	-	1.855

Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.





(a): contre-poids de l'équilibrage des forces;  
 (b): contre-poids de l'équilibrage des moments.

Figure B-4 Exemple B-3: représentation des résultats de l'équilibrage complet du mécanisme.

**B-4 EXEMPLE D'EXÉCUTION 4: équilibrage dynamique d'un rotor.**

La figure B.4 représente un rotor possédant des excentricités. Leurs caractéristiques physiques sont indiquées sur la figure ci-dessous.

Il est demandé de réaliser l'équilibrage dynamique de ce rotor: en disposant dans les plans de correction (A et B) des masses (à calculer) situées à la distance de 6 in par rapport au centre de rotation.

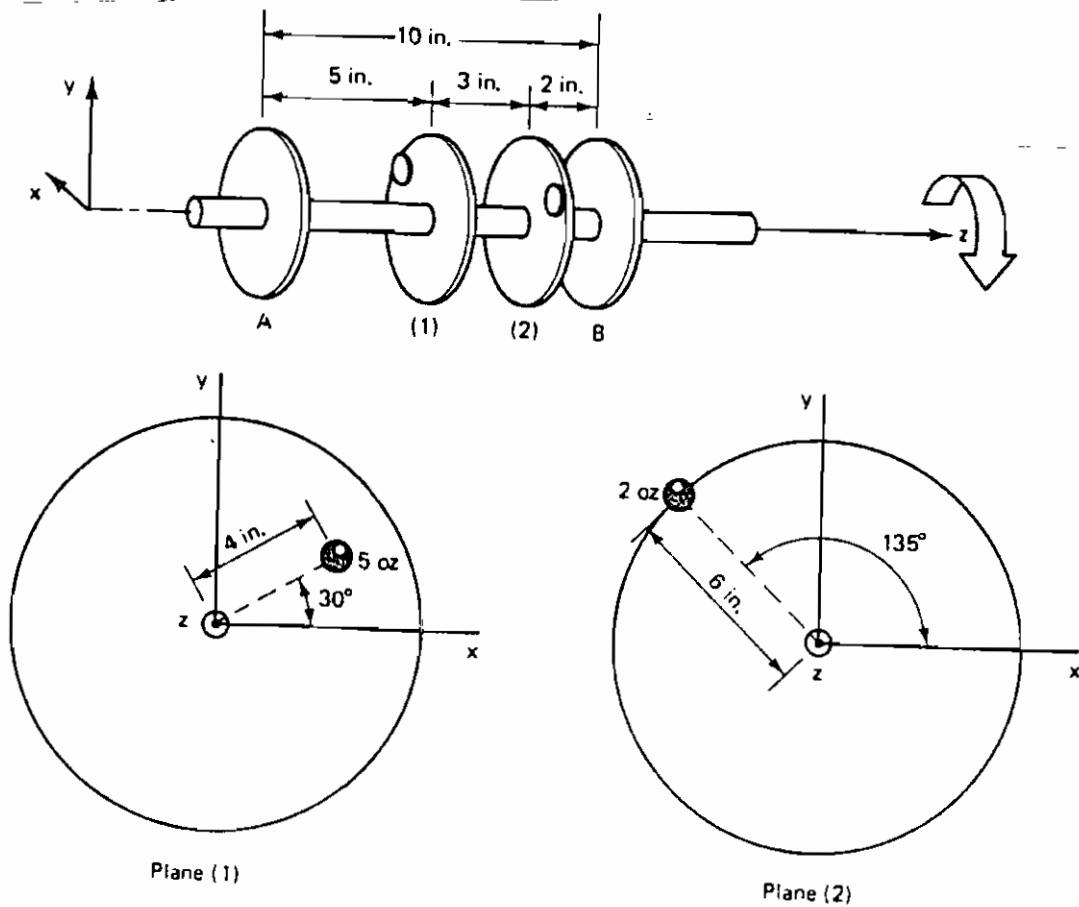


Figure B-5 Exemple 4, rotor: équilibrage dynamique.

Choisir l'option désirée

Analyse cinéto-statique  
Analyse de la réponse temporelle  
Equilibrage des Rotors  
Equilibrage des forces et des moments

Choisir la sous-option désirée

Equilibrage statique  
Equilibrage dynamique

Donnez le nombre de débaleancements < 15 > : 2

Entrez la position (za) du plan A < 0.000 > : 0

Entrez la position (zb) du plan B < 0.000 > : 10

Entrer les caractéristiques du mécanisme.

z1	5.0000
m1	5.0000
r1	4.0000
θ1	0.5236
x1	3.4641
y1	2.0000
z2	8.0000
m2	2.0000
r2	6.0000
θ2	-2.3562
x2	-4.2426
y2	4.2426
z3	0.0000
m3	0.0000
r3	0.0000
θ3	0.0000
x3	0.0000
y3	0.0000

connaissez-vous la position (Rayon) ou la Masse  
 du contrepoids "a" (R, N ou ESC) ? : R  
 Entrez le rayon de "A" < 0.000 > : 6

connaissez-vous la position (Rayon) ou la Masse  
 du contrepoids "b" (R, N ou ESC) ? : R  
 Entrez le rayon de "B" < 0.000 > : 6

	Plan	masse	rayon	Zi	MiRiZi	angle	MiXiZi	MiYiZi
IB	1	5.000	4.000	5.000	100.000	0.524	86.603	50.000
	2	2.000	6.000	8.000	96.000	2.356	-67.882	67.882
Somme							18.720	117.882
IA	1	5.000	4.000	5.000	100.000	0.524	86.603	50.000
	2	2.000	6.000	2.000	24.000	2.356	-16.971	16.971
Somme							69.632	66.971

| IB | = 119.359  
 θb = 1.413

| IA | = 96.611  
 θa = 0.766

Pour une position de rb = 6.000 il nous faudra une masse mb = 1.989  
 Pour une position de ra = 6.000 il nous faudra une masse ma = 1.610

Taper ESC pour sortir du module  
 ou CR pour retourner changer les valeurs.

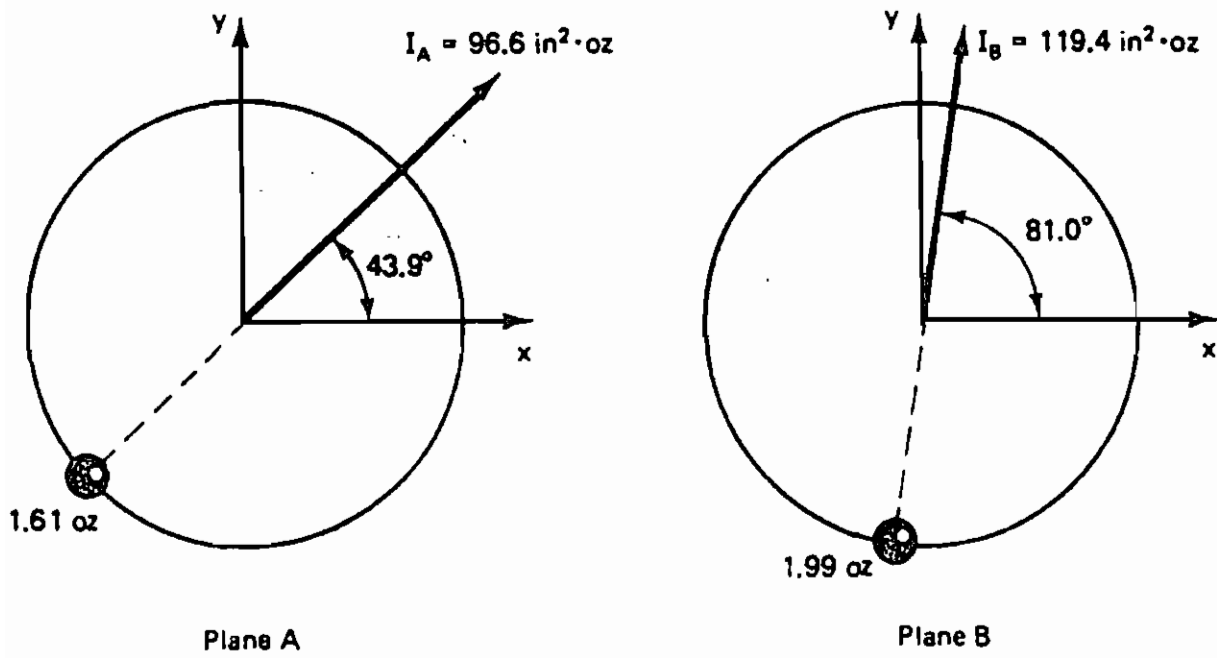


Figure B-6 Exemple B-4: représentation des résultats de l'équilibrage dynamique du rotor.

**ANNEXE C**

**LISTING DU PROGRAMME INFORMATIQUE**

```

UNIT support;
INTERFACE uses crt,printer;
CONST
  rinf=1e-38;
  rsup=1e38;
  imin=6;
  maxdim = 15;
  bp= #7;
  nul=#0;
  cu= #72;  cd= #80; cl=#75; cri=#77; suppr=#83; effg=#8;
  pu= #73;
  pd= #81;
  esc=#27;
  cr= #13;
  ensreel =['0','1','2','3','4','5','6','7','8','9','.','-','+'];
  ensRnegatif =['0','1','2','3','4','5','6','7','8','9','.','-'];
  ensRpositif =['0','1','2','3','4','5','6','7','8','9','.','-'];
  ensentier =['0','1','2','3','4','5','6','7','8','9','.','-'];
  ensZpositif =['0','1','2','3','4','5','6','7','8','9','+'];
  ensZnegatif =['0','1','2','3','4','5','6','7','8','9','-'];

TYPE
  typematrice = array[1..maxdim,1..maxdim] of real;
  typeens = set of char;
  typepage= Record
    taille: byte;
    mot: array[1..25] of string;
    val: array[1..25] of real;
  end;
  champPage=array[1..8] of typepage;
  chtabtype=array[1..5,1..5] of string;
VAR
  i,j:byte;
  car:char;
  page:champPage;
  Rotpart:boolean;
  menux,menuy,menuz:byte;
  chaine:string;
  tabpl:chtabtype;

  PROCEDURE systeme(n:byte;var a:typematrice);
  FUNCTION Arctgt(sina,cosa:real):real;
  Procédure beep;
  PROCEDURE effEcr(debut,fin:byte);
  PROCEDURE vaA(x,y:byte);
  PROCEDURE LireReel(var r:real);
  PROCEDURE LireReelPos(var r:real);
  PROCEDURE LireReelSPos(var r:real);
  PROCEDURE LireReelNeg(var r:real);
  PROCEDURE LireRInter(inf,sup:real; var r:real);
  PROCEDURE LireEntierP(var n:integer);
  PROCEDURE reversevideo(a:boolean);
  PROCEDURE ecrivideo(message:string;x,y:byte;a:boolean);
  PROCEDURE ecrivaleur(valeur:real;x,y:byte;a:boolean);
  PROCEDURE liredefaut(entete:string; var val:real);
  PROCEDURE liredefautPOS(entete:string; var val:real);
  PROCEDURE liredefautSPOS(entete:string; var val:real);
  PROCEDURE liredefauti.nt(entete:string;inf,sup:real; var val:real);
  PROCEDURE plan(entete:string;xe,ye:byte;nl,nc,x,y:byte;tabpl:chtabtype; var l,c:byte);
  PROCEDURE ecripage(page:typepage;k,imin:byte);
  PROCEDURE menu(var page:champPage; debut,fin: byte);

IMPLEMENTATION
PROCEDURE systeme(n:byte;var a:typematrice);

```

```

Var
  i,j,erreur,k: byte;

Begin
  erreur:=0;
  k:=1;
  while (k<=n) and (erreur<>1) do
    if a[k,k]<>0
      then begin for i:=1 to n do
                  if i<k then
                    begin a[n+1,n+1]:=a[i,k];
                          for j:=1 to n+1 do
                            a[i,j]:=a[i,j]-a[k,j]*a[n+1,n+1]/a[k,k];
                          end;
                        k:=k+1
                    end
            else begin i:=k;
                      repeat i:=i+1 until (a[i,k]<>0) or (i=n);
                      if a[i,k]=0 then erreur:=1
                      else begin for j:=1 to n+1 do
                                begin a[n+1,j]:=a[k,j];
                                      a[k,j]:=a[i,j];
                                      a[i,j]:=a[n+1,j]
                                end;
                              end;
                            end;
                      if erreur<>1 then for i:=1 to n do a[n+1,i]:=a[i,n+1]/a[i,i]
                      else writeln('le système n admet pas une solution unique.');
```

```

End;

FUNCTION Arctgt(sina,cosa:real):real;
Begin if cosa=0 then if sina=0 then arctgt:=0
      else if sina<0 then arctgt:=3*pi/2
      else arctgt:=pi/2
      else if cosa>0 then if sina>=0 then arctgt:=arctan(sina/cosa)
      else arctgt:=arctan(sina/cosa)+2*pi
      else arctgt:=pi+arctan(sina/cosa)
End;
```

```

Procedure beep; begin write(bp) end;
```

```

PROCEDURE effEcr(debut,fin:byte);
var k:byte;
begin for k:=debut to fin do begin
      gotoxy(1,k);
      clreol end;
      gotoxy(1,debut);
end;
```

```

PROCEDURE vaA(x,y:byte);
const min=1;
      max=24;
begin if (y<min)or(y>max) then y:=min;
      effecr(y,max);
      gotoxy(x,y)
end;
```

```

PROCEDURE LireReel(var r:real);
Var ens : typeens;
    car2: char;
    code: integer;
    presencepoint:boolean;
    i,j:byte;
    chaine:string;
Begin
  ens:=ensreel;
  i:=1; chaine:=''; presencepoint:=false;
  if car in ens then car2:=car else begin car2:=readkey;
```



```

                                if car2=nul then car2:=readkey
                                end;
repeat
  if ((i>1)and(car2 in ens-['-', '+'])) or ((i=1)and(car2 in ens) )
  then if (car2='.') and presencepoint
    then beep
    else begin insert(car2,chaîne,i);
              for j:=i to length(chaîne) do write(chaîne[j]);
              gotoxy(wherex-j+i,wherey);
              i:=i+1;
              if car2='.' then presencepoint:=true;
            end
  else case car2 of
    cl : if i>1 then begin i:=i-1;
                  gotoxy(wherex-1,wherey) end
          else beep;
    cri : if i<length(chaîne)
          then begin i:=i+1;
                  gotoxy(wherex+1,wherey) end
          else beep;

    effg : if i>1 then begin i:=i-1;
                  if chaîne[i]='.' then presencepoint:=false;
                  delete(chaîne,i,1);
                  gotoxy(wherex+length(chaîne)-i,wherey);
                  clreol;
                  gotoxy(wherex-length(chaîne),wherey);
                  write(chaîne);
                  gotoxy(wherex-length(chaîne)+i-1,wherey) end
          else beep;
    suppr : if i<length(chaîne)
          then begin if chaîne[i]='.' then presencepoint:=false;
                  delete(chaîne,i,1);
                  gotoxy(wherex+length(chaîne)-i,wherey);
                  clreol;
                  gotoxy(wherex-length(chaîne)+1,wherey);
                  write(chaîne);
                  gotoxy(wherex-length(chaîne)+i-1,wherey) end
          else beep;
  end;
  car2:=readkey;
  if car2=nul then car2:=readkey;
until (car2=cu)or(car2=cd)or(car2=cr);
val(chaîne,r,code);
car:=car2;
end;
PROCEDURE LireReelPos(var r:real);
Var ens : typeens;
    car2: char;
    code:integer;
    presencepoint:boolean;
    i,j:byte;
    chaîne:string;
Begin
  ens:=ensRpositif;
  i:=1; chaîne:=''; presencepoint:=false;
  if car in ens then car2:=car
  else begin car2:=readkey;
        if car2=nul then car2:=readkey;
      end;
  repeat
    if ((i>1)and(car2 in ens-['+', '-'])) or ((i=1)and(car2 in ens) )
    then if (car2='.') and presencepoint
      then beep
      else begin insert(car2,chaîne,i);
              for j:=i to length(chaîne) do write(chaîne[j]);

```

```

        gotoxy(wherex-j+i,wherey);
        i:=i+1;
        if car2='.' then presencepoint:=true;
    end
else case car2 of
    cl : if i>1 then begin i:=i-1;
        gotoxy(wherex-1,wherey) end
        else beep;
    cri : if i<=length(chaine)
        then begin i:=i+1;
            gotoxy(wherex+1,wherey) end
        else beep;

    effg : if i>1 then begin i:=i-1;
        if chaine[i]='.' then presencepoint:=false;
        delete(chaine,i,1);
        gotoxy(wherex+length(chaine)-i,wherey);
        clreol;
        gotoxy(wherex-length(chaine),wherey);
        write(chaine);
        gotoxy(wherex-length(chaine)+i-1,wherey) end
        else beep;
    suppr : if i<=length(chaine)
        then begin if chaine[i]='.' then presencepoint:=false;
            delete(chaine,i,1);
            gotoxy(wherex+length(chaine)-i,wherey);
            clreol;
            gotoxy(wherex-length(chaine)+1,wherey);
            write(chaine);
            gotoxy(wherex-length(chaine)+i-1,wherey) end
        else beep
    end;
end;
car2:=readkey;
if car2=nul then car2:=readkey;
until (car2=cu)or(car2=cd)or(car2=cr);
val(chaine,r,code);
car:=car2;
end;
PROCEDURE LireReelSPos(var r:real);
Var ens : typeens;
    car2: char;
    code:integer;
    presencepoint:boolean;
    i,j:byte;
    chaine:string;
Begin
    ens:=ensRpositif;
    i:=1; chaine:=''; presencepoint:=false;
    if car in ens then car2:=car
    else begin car2:=readkey;
        if car2=nul then car2:=readkey;
    end;
repeat
repeat
    if ( (i>1)and(car2 in ens-['+']) ) or ( (i=1)and(car2 in ens) )
    then if (car2='.') and presencepoint
        then beep
        else begin insert(car2,chaine,i);
            for j:=i to length(chaine) do write(chaine[j]);
            gotoxy(wherex-j+i,wherey);
            i:=i+1;
            if car2='.' then presencepoint:=true;
        end
    else case car2 of
        cl : if i>1 then begin i:=i-1;
            gotoxy(wherex-1,wherey) end

```

```

                else beep;
cri : if i<=length(chaine)
    then begin i:=i+1;
            gotoxy(wherex+1,wherey) end
    else beep;

effg : if i>1 then begin i:=i-1;
        if chaine[i]='.' then presencepoint:=false;
        delete(chaine,i,1);
        gotoxy(wherex+length(chaine)-i,wherey);
        clreol;
        gotoxy(wherex-length(chaine),wherey);
        write(chaine);
        gotoxy(wherex-length(chaine)+i-1,wherey) end
    else beep;
suppr : if i<=length(chaine)
    then begin if chaine[i]='.' then presencepoint:=false;
            delete(chaine,i,1);
            gotoxy(wherex+length(chaine)-i,wherey);
            clreol;
            gotoxy(wherex-length(chaine)+1,wherey);
            write(chaine);
            gotoxy(wherex-length(chaine)+i-1,wherey) end
    else beep;
    else beep;
end;
car2:=readkey;
if car2=nul then car2:=readkey;
until (car2=cu)or(car2=cd)or(car2=cr);
val(chaine,r,code);
if r=0 then begin beep; gotoxy(wherex-length(chaine)+1,wherey) end;
until r<>0;
car:=car2;
end;
PROCEDURE LireReelNeg(var r:real);
Var ens : typeens;
car2: char;
code:integer;
presencepoint:boolean;
i,j:byte;
chaine:string;
Begin
ens:=ensRnegatif;
i:=1; chaine=''; presencepoint:=false;
if car in ens then car2:=car
else begin car2:=readkey;
        if car2=nul then car2:=readkey;
        end;
repeat
if ((i>1)and(car2 in ens-['-'])) or ((i=1)and(car2 in ens) )
then if (car2='.') and presencepoint
then beep
else begin insert(car2,chaine,i);
        for j:=i to length(chaine) do write(chaine[j]);
        gotoxy(wherex-j+i,wherey);
        i:=i+1;
        if car2='.' then presencepoint:=true;
        end
else case car2 of
cl : if i>1 then begin i:=i-1;
            gotoxy(wherex-1,wherey) end
        else beep;
cri : if i<=length(chaine)
            then begin i:=i+1;
                    gotoxy(wherex+1,wherey) end
            else beep;

```

```

    effg : if i>1 then begin i:=i-1;
            if chaine[i]='.' then presencepoint:=false;
            delete(chaine,i,1);
            gotoxy(wherex+length(chaine)-i,wherey);
            clreol;
            gotoxy(wherex-length(chaine),wherey);
            write(chaine);
            gotoxy(wherex-length(chaine)+i-1,wherey) end
        else beep;
    suppr : if i<=length(chaine)
            then begin if chaine[i]='.' then presencepoint:=false;
            delete(chaine,i,1);
            gotoxy(wherex+length(chaine)-i,wherey);
            clreol;
            gotoxy(wherex-length(chaine)+1,wherey);
            write(chaine);
            gotoxy(wherex-length(chaine)+i-1,wherey) end
        else beep;
    end;
    car2:=readkey;
    if car2=nul then car2:=readkey;
    until (car2=cu)or(car2=cd)or(car2=cr);
    val(chaine,r,code);
    car:=car2;
end;
PROCEDURE LireInter(Inf,sup:real; var r:real);
Var ens : typeens;
    car2: char;
    code:integer;
    presencepoint:boolean;
    i,j:byte;
    chaine:string;
    rr:real;
Begin
    ens:=ensReel;
    presencepoint:=false;
    if car in ens then car2:=car
    else begin car2:=readkey;
            if car2=nul then car2:=readkey;
        end;
    repeat
        i:=1; chaine:='';
        repeat
            if ( (i>1)and(car2 in ens-['-','+']) ) or ( (i=1)and(car2 in ens) )
            then if (car2='.') and presencepoint
                then beep
                else begin insert(car2,chaine,i);
                        for j:=i to length(chaine) do write(chaine[j]);
                        gotoxy(wherex-j+i,wherey);
                        i:=i+1;
                        if car2='.' then presencepoint:=true;
                    end
            else case car2 of
                cl : if i>1 then begin i:=i-1;
                        gotoxy(wherex-1,wherey) end
                    else beep;
                cri : if i<=length(chaine)
                    then begin i:=i+1;
                        gotoxy(wherex+1,wherey) end
                    else beep;
                effg : if i>1 then begin i:=i-1;
                        if chaine[i]='.' then presencepoint:=false;
                        delete(chaine,i,1);
                        gotoxy(wherex+length(chaine)-i,wherey);
                        clreol;

```

```

                                gotoxy(wherex-length(chaine),wherey);
                                write(chaine);
                                gotoxy(wherex-length(chaine)+i-1,wherey) end
                                else beep;
suppr : if i<=length(chaine)
        then begin if chaine[i]='.' then presencepoint:=false;
                    delete(chaine,i,1);
                    gotoxy(wherex+length(chaine)-i,wherey);
                    clreol;
                    gotoxy(wherex-length(chaine)+1,wherey);
                    write(chaine);
                    gotoxy(wherex-length(chaine)+i-1,wherey) end
                                else beep
                                beep;
                                end;
                                car2:=readkey;
                                if car2=nul then car2:=readkey;
                                until (car2=cu)or(car2=cd)or(car2=cr);
                                val(chaine,rr,code);
                                if (inf>=rr)or(rr>=sup) then begin beep; gotoxy(wherex-length(chaine)+1,wherey) end;
                                until (inf<=rr)and(rr<=sup);
                                r:=rr;
                                car:=car2;
                                end;
PROCEDURE LireEntierP(var n:integer);
Var  ens : typeens;
     car2: char;
     code:integer;
     i,j:byte;
     chaine:string;
Begin
  ens:=ensZpositif;
  i:=1; chaine:='';
  if car in ensZpositif then car2:=car
  else begin car2:=readkey;
         if car2=nul then car2:=readkey
         end;
  repeat
    if ((i>1)and(car2 in ens-['+'])) or ((i=1)and(car2 in ens) )
    then begin insert(car2,chaine,i);
             for j:=i to length(chaine) do write(chaine[j]);
             gotoxy(wherex-j+1,wherey);
             i:=i+1;
            end
    else case car2 of
          cl : if i>1 then begin i:=i-1;
                             gotoxy(wherex-1,wherey) end
                else beep;
          cri : if i<=length(chaine)
                then begin i:=i+1;
                             gotoxy(wherex+1,wherey) end
                else beep;
          effg : if i>1 then begin i:=i-1;
                                delete(chaine,i,1);
                                gotoxy(wherex+length(chaine)-i,wherey);
                                clreol;
                                gotoxy(wherex-length(chaine),wherey);
                                write(chaine);
                                gotoxy(wherex-length(chaine)+i-1,wherey) end
                else beep;
          suppr : if i<=length(chaine)
                 then begin delete(chaine,i,1);
                             gotoxy(wherex+length(chaine)-i,wherey);
                             clreol;
                             gotoxy(wherex-length(chaine)+1,wherey);
                             write(chaine);
        end
    end
  end
end

```

```

                                gotoxy(wherex-length(chaine)+i-1,wherey) end
                                else beep
                                beep;
                                end;
                                end;
                                car2:=readkey;
                                if car2=nul then car2:=readkey;
                                until (car2=cu)or(car2=cd)or(car2=cr);
                                val(chaine,n,code);
                                car:=car2;
                                end;
PROCEDURE reversevideo(a:boolean);
Begin if a then begin textcolor(black); textbackground(white) end
      else begin textcolor(white); textbackground(black) end
end;
PROCEDURE ecrivideo(message:string;x,y:byte;a:boolean);
Begin reversevideo(a);
      gotoxy(x,y);
      write(message);
      reversevideo(false);
end;
PROCEDURE ecrivaleur(valeur:real;x,y:byte;a:boolean);
Begin reversevideo(a);
      gotoxy(x,y);
      write(valeur:10:4);
      reversevideo(false);
end;
PROCEDURE liredefaut(entete:string; var val:real);
begin ecrivideo(entete+' [ ',wherex,wherey,true);
      str(val:8:3,entete);
      ecrivideo(entete,wherex,wherey,true);
      ecrivideo(' ] : ',wherex,wherey,true);
      car:=readkey;
      if car=nul then car:=readkey;
      clreol;
      if (car<>esc)and(car<>cr) then lirereel(val);
end;
PROCEDURE liredefautPOS(entete:string; var val:real);
begin ecrivideo(entete+' [ ',wherex,wherey,true);
      str(val:8:3,entete);
      ecrivideo(entete,wherex,wherey,true);
      ecrivideo(' ] : ',wherex,wherey,true);
      car:=readkey;
      if car=nul then car:=readkey;
      clreol;
      if (car<>esc)and(car<>cr) then lirereelpos(val);
end;
PROCEDURE liredefautSPOS(entete:string; var val:real);
begin ecrivideo(entete+' [ ',wherex,wherey,true);
      str(val:8:3,entete);
      ecrivideo(entete,wherex,wherey,true);
      ecrivideo(' ] : ',wherex,wherey,true);
      car:=readkey;
      if car=nul then car:=readkey;
      clreol;
      if (car<>esc)and(car<>cr) then lirereelspos(val);
end;
PROCEDURE liredefautint(entete:string;inf,sup:real; var val:real);
begin ecrivideo(entete+' [ ',wherex,wherey,true);
      str(val:8:3,entete);
      ecrivideo(entete,wherex,wherey,true);
      ecrivideo(' ] : ',wherex,wherey,true);
      car:=readkey;
      if car=nul then car:=readkey;
      clreol;
      if (car<>esc)and(car<>cr) then lireRinter(inf,sup,val);
end;

```

```

PROCEDURE plan(entete:string;xe,ye:byte;nl,nc,x,y:byte;tabpl:chtabyte; var l,c:byte);
const longy=1;
var i,j,long :byte;
begin long:=length(tabpl[1,1]);
  ecrivideo(entete,xe,ye,true);
  for j:=1 to nl do
    for i:=1 to nc do
      if (i=1)and(j=1) then ecrivideo(tabpl[i,j],x+(i-1)*long,y+(j-1)*longy,true)
        else ecrivideo(tabpl[i,j],x+(i-1)*long,y+(j-1)*longy,false);
    l:=1; c:=1;
    gotoxy(x,y);
    repeat
      car:=readkey;
      if car=nul then car:=readkey;
      case car of
        cri : if c=nc then beep
              else begin ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,false);
                        c:=c+1;
                        ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,true);
                        gotoxy(wherex-long,wherey);
                        end;
        cl : if c=1 then beep
              else begin ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,false);
                        c:=c-1;
                        ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,true);
                        gotoxy(wherex-long,wherey);
                        end;
        cu : if l=1 then beep
              else begin ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,false);
                        l:=l-1;
                        ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,true);
                        gotoxy(wherex-long,wherey);
                        end;
        cd : if l=nl then beep
              else begin ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,false);
                        l:=l+1;
                        ecrivideo(tabpl[c,l],x+(c-1)*long,y+(l-1)*longy,true);
                        gotoxy(wherex-long,wherey);
                        end;
        cr :
          else beep;
          end;
      until car=cr;
    end;
PROCEDURE ecripage(page:typepage;k,imin:byte);
var ii:byte;
begin effecr(3,24);
  ecrivideo('Entrer les caractéristiques du mécanisme.',1,1,4,true);
  with page do
    begin
      ecrivideo(mot[1],1,imin,true);
      ecrivaleur(val[1],11,imin,true);
      for ii:=2 to taille do ecrivideo(mot[ii],1,ii+imin-1,false);
      for ii:=2 to taille do ecrivaleur(val[ii],11,ii+imin-1,false);
      end;
      gotoxy(11,imin);
    end;
PROCEDURE menu(var page:champPage; debut,fin: byte);

Var i,k,actuellu,dernierlu:byte;
    precedent,actuel:string;
    rlu,xlu,olu,ytu:boolean;
Procedure roxy;
begin with page[k] do begin
  dernierlu:=actuellu;
  if i-imin+1 in [3,9,15)

```

```

then begin rlu:=true;
|   actuellu:=1;
|   if (dernierlu=2) then begin
|       val [i-imin+3]:=val [i-imin+1]*cos(val [i-imin+2]);
|       val [i-imin+4]:=val [i-imin+1]*sin(val [i-imin+2]);
|       ecrivaleur(val [i-imin+3],11,i+2,false);
|       ecrivaleur(val [i-imin+4],11,i+3,false); end
|   else if (dernierlu=3)and (val [i-imin+1]>=val [i-imin+3]) then begin
val [i-imin+2]:=arctgt(sqrt(1-sqr(val [i-imin+3]/val [i-imin+1])),val [i-imin+3]/val [i-imin+1]);
|       val [i-imin+4]:=val [i-imin+1]*sin(val [i-imin+2]);
|       ecrivaleur(val [i-imin+2],11,i+1,false);
|       ecrivaleur(val [i-imin+4],11,i+3,false); end
|   else if (dernierlu=4)and (val [i-imin+1]>=val [i-imin+4]) then begin
val [i-imin+2]:=arctgt(val [i-imin+4]/val [i-imin+1],sqrt(1-sqr(val [i-imin+4]/val [i-imin+1])));
|       val [i-imin+3]:=val [i-imin+1]*cos(val [i-imin+2]);
|       ecrivaleur(val [i-imin+2],11,i+1,false);
|       ecrivaleur(val [i-imin+3],11,i+2,false); end;
end;
if i-imin+1 in [4,10,16] then begin
|   actuellu:=2;
|   olu:=true;
|   if (dernierlu=1)then begin
|       val [i-imin+2]:=val [i-imin]*cos(val [i-imin+1]);
|       val [i-imin+3]:=val [i-imin]*sin(val [i-imin+1]);
|       ecrivaleur(val [i-imin+2],11,i+1,false);
|       ecrivaleur(val [i-imin+3],11,i+2,false); end
|   else if (dernierlu=3)then begin
|       val [i-imin]:=val [i-imin+2]/cos(val [i-imin+1]);
|       val [i-imin+3]:=val [i-imin]*sin(val [i-imin+1]);
|       ecrivaleur(val [i-imin],11,i-1,false);
|       ecrivaleur(val [i-imin+3],11,i+2,false); end
|   else if (dernierlu=4)then begin
|       val [i-imin]:=val [i-imin+3]/sin(val [i-imin+1]);
|       val [i-imin+2]:=val [i-imin]*cos(val [i-imin+1]);
|       ecrivaleur(val [i-imin],11,i-1,false);
|       ecrivaleur(val [i-imin+2],11,i+1,false); end;
end;
if i-imin+1 in [5,11,17] then begin
|   xlu:=true;
|   actuellu:=3;
|   if (dernierlu=4)then begin
|       val [i-imin-1]:=sqrt(sqr(val [i-imin+1])+sqr(val [i-imin+2]));
|       val [i-imin]:=arctgt(val [i-imin+2],val [i-imin+1]);
|       ecrivaleur(val [i-imin-1],11,i-2,false);
|       ecrivaleur(val [i-imin],11,i-1,false); end
|   else if (dernierlu=2)then begin
|       val [i-imin-1]:=val [i-imin+1]/cos(val [i-imin]);
|       val [i-imin+2]:=val [i-imin-1]*sin(val [i-imin]);
|       ecrivaleur(val [i-imin-1],11,i-2,false);
|       ecrivaleur(val [i-imin+2],11,i+1,false); end
|   else if (dernierlu=1)and(val [i-imin+1]<=val [i-imin-1])then begin
val [i-imin]:=arctgt(sqrt(1-sqr(val [i-imin+1]/val [i-imin-1])),val [i-imin+1]/val [i-imin-1]);
|       val [i-imin+2]:=val [i-imin-1]*sin(val [i-imin]);
|       ecrivaleur(val [i-imin],11,i-1,false);
|       ecrivaleur(val [i-imin+2],11,i+1,false); end;
end;
if i-imin+1 in [6,12,18] then begin
|   ylu:=true;
|   actuellu:=4;
|   if (dernierlu=3)and(val [i-imin+1]<=val [i-imin-2]) then begin
|       val [i-imin-2]:=sqrt(sqr(val [i-imin])+sqr(val [i-imin+1]));
|       val [i-imin-1]:=arctgt(val [i-imin+1],val [i-imin]);
|       ecrivaleur(val [i-imin-2],11,i-3,false);
|       ecrivaleur(val [i-imin-1],11,i-2,false); end

```



```

    else if dernierlu=2 then begin
        val[i-imin-2]:=val[i-imin+1]/sin(val[i-imin-1]);
        val[i-imin]:=val[i-imin-2]*cos(val[i-imin-1]);
        ecrivaleur(val[i-imin-2],11,i-3,false);
        ecrivaleur(val[i-imin],11,i-1,false); end
    else if dernierlu=1 then begin
val[i-imin-1]:=arctgt(val[i-imin+1]/val[i-imin-2],sqrt(1-sqr(val[i-imin+1]/val[i-imin-2])));
        val[i-imin]:=val[i-imin-2]*cos(val[i-imin-1]);
        ecrivaleur(val[i-imin-1],11,i-2,false);
        ecrivaleur(val[i-imin],11,i-1,false); end;
    end; end;

end;

Begin
xlu:=false; ylu:=false; olu:=false; rlu:=false;
actuellu:=1; dernierlu:=1;
k:=debut; i:=imin;
ecripage(page[k],k,i);
car:=readkey;
if car=nul then car:=readkey;

Repeat
Case car of
cu : begin if i=imin then beep
        else begin precedent:=page[k].mot[i-imin+1];
                ecrivideo(precedent,1,i,false);
                ecrivaleur(page[k].val[i-imin+1],11,i,false);
                i:=i-1;
                actuel:=page[k].mot[i-imin+1];
                ecrivideo(actuel,1,i,true);
                ecrivaleur(page[k].val[i-imin+1],11,i,true);
                gotoxy(11,i) end;
        car:=readkey;
        if car=nul then car:=readkey;
    end;
cd : begin if i=page[k].taille+imin-1 then beep
        else begin precedent:=page[k].mot[i-imin+1];
                ecrivideo(precedent,1,i,false);
                ecrivaleur(page[k].val[i-imin+1],11,i,false);
                i:=i+1;
                actuel:=page[k].mot[i-imin+1];
                ecrivideo(actuel,1,i,true);
                ecrivaleur(page[k].val[i-imin+1],11,i,true);
                gotoxy(11,i) end;
        car:=readkey;
        if car=nul then car:=readkey;
    end;
pu : begin if k=debut then beep
        else begin k:=k-1; i:=imin;
                ecripage(page[k],k,i) end;
        car:=readkey;
        if car=nul then car:=readkey;
    end;
pd : begin if k=fin then beep
        else begin k:=k+1; i:=imin;
                ecripage(page[k],k,i) end;
        car:=readkey;
        if car=nul then car:=readkey;
    end;
esc:
else if car in ensreel
    then begin
        lirereel(page[k].val[i-imin+1]);
        ecrivaleur(page[k].val[i-imin+1],11,i,true);
        if menux=3 then roxy end
    end;
end;

```

```

      else if car=cr then car:=cd
        else begin beep;
              car:=readkey;
              if car=nul then car:=readkey;
              end;
        end;
    Until car=esc; effecr(2,24);
End;
END.
```

UNIT chap12;

```
INTERFACE uses crt,printer,support;
PROCEDURE reptempolf;
PROCEDURE reptempomc;
PROCEDURE ForcesJointsM4M;
PROCEDURE ForcesJointsMC;
```

IMPLEMENTATION

PROCEDURE reptempolf;

```
Var
a1,a2,a3,a4,
q1,q2,q3,q4,rg2,rg3,rg4,
qq,qq2,qq3,qq4,rrg2,rrg3,rrg4,
m2,m3,m4,lg2,lg3,lg4,
s,bd,b,d,c,
o2,o3,o3p,o4,w2,w3,w4,ts,
tl,cosa,sina,ccc,ddd      :real;
x                          :array[1..maxdim] of real;
a                          :typematrice;
tabplfb                   :chtabyte;
entete                    :string;
l,cc                      :byte;
```

Procedure rtlfinitMes;

```
begin
with page[1] do
begin taille:=17;
mot[1]:=' a1      '; val[1]:=a1; mot[2]:=' a2      '; val[2]:=a2;
mot[3]:=' a3      '; val[3]:=a3; mot[4]:=' a4      '; val[4]:=a4;
mot[5]:=' b1      '; val[5]:=q1; mot[6]:=' b2      '; val[6]:=q2;
mot[7]:=' b3      '; val[7]:=q3; mot[8]:=' b4      '; val[8]:=q4;
mot[9]:=' rg2     '; val[9]:=rg2; mot[10]:=' rg3     '; val[10]:=rg3;
mot[11]:=' rg4     '; val[11]:=rg4; mot[12]:=' m2      '; val[12]:=m2;
mot[13]:=' m3      '; val[13]:=m3; mot[14]:=' m4      '; val[14]:=m4;
mot[15]:=' lg2     '; val[15]:=lg2; mot[16]:=' lg3     '; val[16]:=lg3;
mot[17]:=' lg4     '; val[17]:=lg4
end;

with page[2] do
begin taille:=4;
mot[1]:=' Ú2      '; mot[2]:=' w2      ';
mot[3]:=' Ts      '; val[1]:=o2; val[2]:=w2; val[3]:=ts;
mot[4]:=' Tl      '; val[4]:=tl
end;

with page[3] do
begin taille:=6;
mot[1]:=' a2      '; val[1]:=a2; mot[2]:=' a3      '; val[2]:=a3;
mot[3]:=' a4      '; val[3]:=a4; mot[4]:=' Ú2      '; val[4]:=o2;
mot[5]:=' Ú3      '; val[5]:=o3; mot[6]:=' Ú4      '; val[6]:=o4
end; end;
```

Procedure retournevalrtlf(debut,fin: byte);

```
var k:byte;
begin
for k:=debut to fin do
case k of
1: with page[1] do begin
a1:=val[1]; a2:=val[2]; a3:=val[3]; a4:=val[4];
q1:=val[5]; q2:=val[6]; q3:=val[7]; q4:=val[8];
rg2:=val[9]; rg3:=val[10]; rg4:=val[11]; m2:=val[12];
m3:=val[13]; m4:=val[14]; ig2:=val[15]; ig3:=val[16];
ig4:=val[17]; end;
2: with page[2] do begin
o2:=val[1]; w2:=val[2]; ts:=val[3]; tl:=val[4] end;
```

```

3: with page[3] do
    a2:=val [1];    a3:=val [2];    a4:=val [3];    begin
    o2:=val [4];    o3:=val [5];    o4:=val [6];    end;
end; end;

Procedure CalculR1PSY1;
var chaine:string;
Begin
    menu(page,3,3);
    retournevalrtlf(3,3);
    cosa:=a2*cos(o2)+a3*cos(o3)-a4*cos(o4);
    sina:=a2*sin(o2)+a3*sin(o3)-a4*sin(o4);
    q1:=arctgt(sina,cosa);
    if cos(q1)<>0 then a1:=cosa/cos(q1)
        else a1:=sina/sin(q1);
    clrscr;
    ecrivideo('R1 = ',1,10,true);
    str(a1:8:4,chaine);
    ecrivideo(chaine,8,10,false);
    ecrivideo('psy1 = ',1,12,true);
    str(q1:8:4,chaine);
    ecrivideo(chaine,8,12,false);
    ecrivideo('Tapez une touche pour continuer.',1,15,true);
    repeat until keypressed
End;

Procedure CalculTETA34;
Begin s:=sqrt(a1*a1+a2*a2-2*a1*a2*cos(o2-o1));
    cosa:=(a2*a2-a1*a1-s*s)/(-2*a1*s);
    sina:=a2*sin(o2-q1)/s;
    b:=arctgt(sina,cosa);
    cosa:=(s*s+a3*a3-a4*a4)/(2*a3*s);
    if cosa=0 then d:=pi/2
        else d:=arctan(sqrt(1-cosa*cosa)/cosa);
    if cosa<0 then d:=d+pi;
    sina:=a3*sin(d)/a4;
    cosa:=(a3*a3-a4*a4-s*s)/(-2*a4*s);
    c:=arctgt(sina,cosa);
    o3:=d-b+q1;
    o4:=pi-b+q1-c;
    o3p:=o3+2*pi-2*d;
    str(o3:8:3,tabplfb[1,1]);
    str(o3p:8:3,tabplfb[2,1]);
    entete:='Choisissez U3 entre : ';
    plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,cc);
    if cc=2 then begin o3:=o3p; o4:=o4+2*c end;
End;

Procedure calculs;
Procedure rrg0q(a,q,o,rg:real; var rrg,qq:real);
begin rrg:=sqrt(rg*rg+a*a-2*rg*a*cos(q));
    if rg=0 then qq:=o;
    if (rg=a)and(q=0) then qq:=0;
    if (rg<>0) and ((rg<>a)or(q<>0)) then
        begin sina:=a*sin(q)/rrg;
            cosa:=(a*a-rrg*rrg-rg*rg)/(-2*rrg*rg);
            qq:=q+o-pi+arctgt(sina,cosa);
        end;
end;

Begin
w3:=(-a2*sin(o2-o4)*w2)/(a3*sin(o3-o4));
w4:=(-a2*sin(o2-o3)*w2)/(a4*sin(o3-o4));
ccc:=(a2*w2*w2*cos(o4-o2)+a3*w3*w3*cos(o3-o4)-a4*w4*w4)/(-a3*sin(o3-o4));
ddd:=(a2*w2*w2*cos(o3-o2)-a3*w4*w4*cos(o3-o4)+a3*w3*w3)/(a3*sin(o4-o3));
rrg0q(a2,q2,o2,rg2,rrg2,qq2);
rrg0q(a3,q3,o3,rg3,rrg3,qq3);
rrg0q(a4,q4,o4,rg4,rrg4,qq4);

```

```

(*constitution de la matrice*)
for i:=1 to 10 do for j:=1 to 10 do a[i,j]:=0;
for i:=1 to 5 do a[i,i]:=-1; a[3,3]:=ig2;
a[1,3]:=-m2*rg2*sin(o2+q2); a[1,4]:=1;
a[2,3]:=m2*rg2*cos(o2+q2); a[2,5]:=1;
a[3,1]:=-rg2*cos(o2+q2); a[3,2]:=rg2*sin(o2+q2);
a[3,4]:=-rrg2*sin(qq2); a[3,5]:=rrg2*cos(qq2);
a[4,3]:=-m3*(a2*sin(o2)+rg3*w3*sin(o3+q3)/w2); a[4,6]:=1;
a[5,3]:=-m3*(a2*cos(o2)+rg3*w3*cos(o3+q3)/w2); a[5,7]:=1;
a[6,4]:=-rg3*sin(o3+q3); a[6,5]:=rg3*cos(o3+q3); a[6,3]:=ig3*w3/w2;
a[6,6]:=-rrg3*sin(qq3); a[6,7]:=rrg3*cos(qq3);
a[7,3]:=-m4*rg4*w4*sin(o4+q4)/w2; a[7,6]:=-1; a[7,8]:=-1;
a[8,3]:=-m4*rg4*w4*cos(o4+q4)/w2; a[8,7]:=-1; a[8,9]:=-1;
a[9,6]:=rrg4*sin(qq4); a[9,7]:=-rrg4*cos(qq4); a[9,3]:=ig4*w4/w2;
a[9,8]:=-rg4*sin(o4+q4); a[9,9]:=rg4*cos(o4+q4);
a[1,10]:=m2*rg2*cos(o2+q2)*w2*w2; a[2,10]:=m2*rg2*sin(o2+q2)*w2*w2;
a[3,10]:=ts; a[6,10]:=-ig3*ccc; a[9,10]:=tl-ig4*ddd;
a[4,10]:=m3*(a2*cos(o2)*w2*w2+ccc*rg3*sin(o3+q3)+rg3*cos(o3+q3)*w3*w3);
a[5,10]:=m3*(a2*sin(o2)*w2*w2-ccc*rg3*cos(o3+q3)+rg3*sin(o3+q3)*w3*w3);
a[7,10]:=m4*(ddd*rg4*sin(o4+q4)+rg4*cos(o4+q4)*w4*w4);
a[8,10]:=m4*(-ddd*rg4*cos(o4+q4)+rg4*sin(o4+q4)*w4*w4);
End;

Procedure initrtlf;
Begin a1:=0; a2:=0; a3:=0; a4:=0;
q1:=0; q2:=0; q3:=0; q4:=0; rg2:=0; rg3:=0; rg4:=0;
m2:=0; m3:=0; m4:=0; lg2:=0; lg3:=0; lg4:=0;
o2:=0; o3:=0; o4:=0; w2:=0; tl:=0;
end;

BEGIN
initrtlf;
rtlfinitMes;
clrscr;
ecrivideo('connaissez-vous psy1 et o1 ? (0,N) : ',1,10,true);
repeat
car:=upcase(readkey);
if (car<>'0')and(car<>'N') then beep
until (car='0')or(car='N');
if car='N' then calculr1psy1;
Repeat
menu(page,1,2); clrscr;
retournevalrtlf(1,2);
calculteta34;
calculs; clrscr;
systeme(9,a);
clrscr;
writeln('ANALYSE DE LA REPONSE TEMPORELLE: M4M longueurs finies. ');
writeln;
writeln('Forces dynamiques aux joints et Ó2. ');
writeln;
writeln(' F12x = ',a[10,1]:15:4);
writeln(' F12y = ',a[10,2]:15:4);
writeln(' Ó2 = ',a[10,3]:15:4);
writeln(' F23x = ',a[10,4]:15:4);
writeln(' F23y = ',a[10,5]:15:4);
writeln(' F34x = ',a[10,6]:15:4);
writeln(' F34y = ',a[10,7]:15:4);
writeln(' F14x = ',a[10,8]:15:4);
writeln(' F14y = ',a[10,9]:15:4);
ecrivideo('Voulez-vous imprimer les resultats ? (0,N) : ',1,1,true);
repeat
car:=upcase(readkey);
if (car<>'0')and(car<>'N') then beep
until (car='0')or(car='N');
if car='0' then begin
writeln(lst,'ANALYSE DE LA REPONSE TEMPORELLE: M4M longueurs finies. ');

```

```

writeln(lst);
writeln(lst,'Forces dynamiques aux joints et Ó2. ');
writeln(lst);
writeln(lst,' F12x = ',a[10,1]:15:4);
writeln(lst,' F12y = ',a[10,2]:15:4);
writeln(lst,' Ó2 = ',a[10,3]:15:4);
writeln(lst,' F23x = ',a[10,4]:15:4);
writeln(lst,' F23y = ',a[10,5]:15:4);
writeln(lst,' F34x = ',a[10,6]:15:4);
writeln(lst,' F34y = ',a[10,7]:15:4);
writeln(lst,' F14x = ',a[10,8]:15:4);
writeln(lst,' F14y = ',a[10,9]:15:4); end;
ecrivideo('Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.',1,1,true);
repeat car:=readkey;
  if (car<>esc)and(car<>cr) then beep
until (car=esc) or (car=cr);
Until car=esc;
End;
PROCEDURE reptempmc;
Var
a2,a3,a4,e,
q1,q2,q3,q4,rg2,rg3,
qq,qq2,qq3,qq4,rrg2,rrg3,rrg4,
m2,m3,m4,lg2,lg3,lg4,
s,bd,b,d,c,
o2,o3,o3p,o4,w2,w3,w4,
fl,cosa,sina,eee,ts :real;
x :array[1..maxdim] of real;
a :typematrice;
tabplfb :chtabyte;
entete :string;
l,cc :byte;
Procedure rtMCinitMes;
begin
with page[1] do
begin taille:=13;
mot[1]:= ' a2 ' ; val[1]:=a2; mot[2]:= ' a3 ' ; val[2]:=a3;
mot[3]:= 'upset e ' ; val[3]:=e; mot[4]:= ' Y ' ; val[4]:=q1;
mot[5]:= ' b2 ' ; val[5]:=q2; mot[6]:= ' b3 ' ; val[6]:=q3;
mot[7]:= ' rg2 ' ; val[7]:=rg2; mot[8]:= ' rg3 ' ; val[8]:=rg3;
mot[9]:= ' m2 ' ; val[9]:=m2; mot[10]:= ' m3 ' ; val[10]:=m3;
mot[11]:= ' m4 ' ; val[11]:=m4; mot[12]:= ' lg2 ' ; val[12]:=lg2;
mot[13]:= ' lg3 ' ; val[13]:=lg3;
end;
with page[2] do
begin taille:=4; mot[1]:= ' Ú2 ' ; mot[2]:= ' w2 ' ;
mot[3]:= ' Ts ' ; val[1]:=o2; val[2]:=w2; val[3]:=ts;
mot[4]:= ' Fl ' ; val[4]:=fl
end;
end;
Procedure retournevalrtmc(debut,fin: byte);
var k:byte;
begin
for k:=debut to fin do
case k of
1: with page[1] do begin
a2:=val[1]; a3:=val[2]; e:=val[3]; q1:=val[4];
q2:=val[5]; q3:=val[6]; rg2:=val[7]; rg3:=val[8];
m2:=val[9]; m3:=val[10]; m4:=val[11]; lg2:=val[12];
lg3:=val[13]; end;
2: with page[2] do begin
o2:=val[1]; w2:=val[2]; ts:=val[3]; fl:=val[4] end;

```

```

end;
end;

Procedure Calculo3b;
Begin sina:=(-a2*sin(o2-q1)-e)/a3;
      cosa:=sqrt(1-sqr(sina));
      b:=arctgt(sina,cosa);
      o3:=q1-b-pi;
      o3p:=q1+b-2*pi;
      str(o3:8:3,tabplfb[1,1]);
      str(o3p:8:3,tabplfb[2,1]);
      entete:='Choisissez U3 entre : ';
      plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,cc);
      if cc=2 then begin o3:=o3p; b:=pi-b end;
      o4:=q1-0.5*pi;
End;

Procedure calculs;
Procedure rrgQq(a,q,o,rg:real; var rrg,qq:real);
begin rrg:=sqrt(rg*rg+a*a-2*rg*a*cos(q));
      if rg=0 then qq:=o;
      if (rg=a)and(q=0) then qq:=0;
      if (rg<>0) and ((rg<>a)or(q<>0)) then
        begin sina:=a*sin(q)/rrg;
              cosa:=(a*a-rrg*rrg-rg*rg)/(-2*rrg*rg);
              qq:=q+o-pi+arctgt(sina,cosa);
        end;
      end;

Begin
w3:=(-a2*cos(o2-q1)*w2)/(a3*cos(o3-q1));
sina:=sin(o3-q1);
cosa:=cos(o3-q1);
eee:=(a2*w2*w2*sin(o2-q1))/(a3*cos(o3-q1))+w3*w3*arctgt(sina,cosa);

rrgQq(a2,q2,o2,rg2,rrg2,qq2);
rrgQq(a3,q3,o3,rg3,rrg3,qq3);

(*constitution de la matrice*)
for i:=1 to 8 do for j:=1 to 8 do a[i,j]:=0;
for i:=1 to 5 do a[i,i]:=-1; a[3,3]:=ig2;
a[1,3]:=-m2*rg2*sin(o2+q2); a[1,4]:=1;
a[2,3]:=m2*rg2*cos(o2+q2); a[2,5]:=1;
a[3,1]:=-rg2*sin(o2+q2); a[3,2]:=rg2*cos(o2+q2);
a[3,4]:=-rrg2*sin(qq2); a[3,5]:=rrg2*cos(qq2);
a[4,3]:=-m3*(a2*sin(o2)+rg3*sin(o3+q3)*w3/w2); a[4,6]:=1;
a[5,3]:=-m3*(a2*cos(o2)+rg3*cos(o3+q3)*w3/w2); a[5,7]:=1;
a[6,4]:=-rg3*sin(o3+q3); a[6,5]:=rg3*cos(o3+q3); a[6,3]:=ig3*w3/w2;
a[6,6]:=-rrg3*sin(qq3); a[6,7]:=rrg3*cos(qq3);
a[7,6]:=cos(q1); a[7,7]:=sin(q1);
a[7,3]:=-m4*(a3*sin(o3-q1)*(w3/w2)+a2*sin(o2-q1));
a[1,8]:=m2*rg2*cos(o2+q2)*w2*w2; a[2,10]:=m2*rg2*sin(o2+q2)*w2*w2;
a[3,8]:=ts; a[6,8]:=-ig3*eee;
a[4,8]:=m3*(a2*cos(o2)*w2*w2+eee*rg3*sin(o3+q3)+rg3*cos(o3+q3)*w3*w3);
a[5,8]:=m3*(a2*sin(o2)*w2*w2-eee*rg3*cos(o3+q3)+rg3*sin(o3+q3)*w3*w3);
a[7,8]:=m4*(eee*a3*sin(o3-q1)+a3*w3*w3*cos(o3-q1)+a2*w2*w2*cos(o2-q1))+fl;
End;

Procedure initrtMC;
Begin e:=0; a2:=0; a3:=0; a4:=0;
      q1:=0; q2:=0; q3:=0; q4:=0; rg2:=0; rg3:=0;
      m2:=0; m3:=0; m4:=0; lg2:=0; lg3:=0;
      o2:=0; o3:=0; o4:=0; w2:=0; fl:=0;
end;

BEGIN
initrtMC;
rtMCinitMes;

```

```

clrscr;
Repeat
  menu(page,1,2); clrscr;
  retournevalrtmc(1,2);
  calculo3b;
  calculs; clrscr;
  systeme(7,a);
  clrscr;
  writeln('Forces dynamiques aux joints et 02. '); writeln;
  writeln(' F12x = ',a[8,1]:15:4);
  writeln(' F12y = ',a[8,2]:15:4);
  writeln(' 02 = ',a[8,3]:15:4);
  writeln(' F23x = ',a[8,4]:15:4);
  writeln(' F23y = ',a[8,5]:15:4);
  writeln(' F34x = ',a[8,6]:15:4);
  writeln(' F34y = ',a[8,7]:15:4);
  writeln;
  ecrivideo('Voulez-vous imprimer les resultats ? (0,N) : ',1,23,true);
repeat
  car:=upcase(readkey);
  if (car<>'0')and(car<>'N') then beep
until (car='0')or(car='N');
if car='0' then begin
  writeln(lst,'Forces dynamiques aux joints et 02. '); writeln;
  writeln(lst,' F12x = ',a[8,1]:15:4);
  writeln(lst,' F12y = ',a[8,2]:15:4);
  writeln(lst,' 02 = ',a[8,3]:15:4);
  writeln(lst,' F23x = ',a[8,4]:15:4);
  writeln(lst,' F23y = ',a[8,5]:15:4);
  writeln(lst,' F34x = ',a[8,6]:15:4);
  writeln(lst,' F34y = ',a[8,7]:15:4);
  end;
  ecrivideo('Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.',1,24,true);
  repeat car:=readkey;
    if (car<>'esc')and(car<>'cr') then beep
  until (car='esc') or (car='cr');
Until car='esc';
End;

PROCEDURE ForcesJointsM4m;

Var
  a1,a2,a3,a4,
  q1,q2,q3,q4,rg2,rg3,rg4,
  qq,qq2,qq3,qq4,rrg2,rrg3,rrg4,
  m2,m3,m4,lg2,lg3,lg4,
  s,bd,b,d,c,
  o2,o3,o3p,o4,w2,w3,w4,aa2,aa3,aa4,
  ag2x,ag2y,ag3x,ag3y,ag4x,ag4y,
  ag2,b2,ag3,b3,ag4,b4,
  Fo2x,Fo3x,Fo4x,
  Fo2y,Fo3y,Fo4y,
  To2,To3,To4,tl,cosa,sina
  x
  a
  tabplfb
  entete
  l,cc
  :real;
  :array[1..maxdim] of real;
  :typematrice;
  :chtabtype;
  :string;
  :byte;

Procedure M4MinitMes;
begin
  with page[1] do
  begin taille:=17;
    mot[1]:= ' a1 ' ; val[1]:=a1; mot[2]:= ' a2 ' ; val[2]:=a2;
    mot[3]:= ' a3 ' ; val[3]:=a3; mot[4]:= ' a4 ' ; val[4]:=a4;
    mot[5]:= ' b1 ' ; val[5]:=q1; mot[6]:= ' b2 ' ; val[6]:=q2;
    mot[7]:= ' b3 ' ; val[7]:=q3; mot[8]:= ' b4 ' ; val[8]:=q4;
    mot[9]:= ' rg2 ' ; val[9]:=rg2; mot[10]:= ' rg3 ' ; val[10]:=rg3;

```



```

mot[11]:= ' rg4      '; val[11]:=rg4; mot[12]:= ' m2      '; val[12]:=m2;
mot[13]:= ' m3      '; val[13]:=m3; mot[14]:= ' m4      '; val[14]:=m4;
mot[15]:= ' lg2      '; val[15]:=lg2; mot[16]:= ' lg3      '; val[16]:=lg3;
mot[17]:= ' lg4      '; val[17]:=lg4
end;

with page[2] do
begin taille:=4;          mot[1]:= ' Ú2      '; mot[2]:= ' w2      ';
  mot[3]:= ' Ó2      '; val[1]:=o2;   val[2]:=w2;   val[3]:=aa2;
  mot[4]:= ' Tl      '; val[4]:=tl
end;

with page[3] do
begin taille:=6;
  mot[1]:= ' a2      '; val[1]:=a2; mot[2]:= ' r3      '; val[2]:=a3;
  mot[3]:= ' a4      '; val[3]:=a4; mot[4]:= ' Ú2      '; val[4]:=o2;
  mot[5]:= ' Ú3      '; val[5]:=o3; mot[6]:= ' Ú4      '; val[6]:=o4
end; end;

Procedure retournevalm4m(debut,fin: byte);
var k:byte;
begin
for k:=debut to fin do
case k of
1: with page[1] do begin
a1:=val[1]; a2:=val[2]; a3:=val[3]; a4:=val[4];
q1:=val[5]; q2:=val[6]; q3:=val[7]; q4:=val[8];
rg2:=val[9]; rg3:=val[10]; rg4:=val[11]; m2:=val[12];
m3:=val[13]; m4:=val[14]; ig2:=val[15]; ig3:=val[16];
ig4:=val[17]; end;
2: with page[2] do begin
o2:=val[1]; w2:=val[2]; aa2:=val[3]; tl:=val[4] end;
3: with page[3] do begin
a2:=val[1]; a3:=val[2]; a4:=val[3];
o2:=val[4]; o3:=val[5]; o4:=val[6]; end;
end; end;

Procedure CalculR1PSY1;
var chaine:string;
Begin
menu(page,3,3);
retournevalm4m(3,3);
cosa:=a2*cos(o2)+a3*cos(o3)-a4*cos(o4);
sina:=a2*sin(o2)+a3*sin(o3)-a4*sin(o4);
q1:=arctgt(sina,cosa);
if cos(q1)<>0 then a1:=cosa/cos(q1)
else a1:=sina/sin(q1);

clrscr;
ecrivideo('R1 = ',1,10,true);
str(a1:8:4,chaine);
ecrivideo(chaine,8,10,false);
ecrivideo('psyl = ',1,12,true);
str(q1:8:4,chaine);
ecrivideo(chaine,8,12,false);
ecrivideo('Tapez une touche pour continuer.',1,15,true);
repeat until keypressed
End;

Procedure CalculTETA34;
Begin s:=sqrt(a1*a1+a2*a2-2*a1*a2*cos(O2-Q1));
cosa:=(a2*a2-a1*a1-s*s)/(-2*a1*s);
sina:=a2*sin(o2-q1)/s;
b:=arctgt(sina,cosa);
cosa:=(s*s+a3*a3-a4*a4)/(2*a3*s);
if cosa=0 then d:=pi/2
else d:=arctan(sqrt(1-cosa*cosa)/cosa);
if cosa<0 then d:=d+pi;

```

```

sina:=a3*sin(d)/a4;
cosa:=(a3*a3-a4*a4-s*s)/(-2*a4*s);
c:=arctgt(sina,cosa);
o3:=d-b+q1;
o4:=pi-b+q1-c;
o3p:=o3+2*pi-2*d;
str(o3:8:3,tabplfb[1,1]);
str(o3p:8:3,tabplfb[2,1]);
entete:='Choisissez U3 entre : ';
plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,cc);
if cc=2 then begin o3:=o3p; o4:=o4+2*c end;
End;

Procedure calculs;
Procedure rrg0q(a,q,o,rg:real; var rrg,qq:real);
begin rrg:=sqrt(rg*rg+a*a-2*rg*a*cos(q));
if rg=0 then qq:=o;
if (rg=a)and(q=0) then qq:=0;
if (rg<>0) and ((rg<a)or(q<0)) then
begin sina:=a*sin(q)/rrg;
cosa:=(a*a-rrg*rrg-rg*rg)/(-2*rrg*rg);
qq:=q+o-pi+arctgt(sina,cosa);
end; end;

Begin
w3:=(-a2*sin(o2-o4)*w2)/(a3*sin(o3-o4));
w4:=(-a2*sin(o2-o3)*w2)/(a4*sin(o3-o4));
aa3:=aa2*w3/w2+(a2*w2*w2*cos(o4-o2)+a3*w3*w3*cos(o4-o3)-a4*w4*w4)/(-a3*sin(o3-o4));
aa4:=aa2*w4/w2+(a2*w2*w2*cos(o3-o2)-a4*w4*w4*cos(o4-o3)+a3*w3*w3)/(-a4*sin(o3-o4));
ag2x:=rg2*(-aa2*sin(o2+q2)-w2*w2*cos(o2+q2));
ag2y:=rg2*(aa2*cos(o2+q2)-w2*w2*sin(o2+q2));
ag2:=sqrt(ag2x*ag2x+ag2y*ag2y);
b2:=arctgt(ag2y,ag2x);
ag3x:=-rg3*aa3*sin(o3+q3)-rg3*w3*w3*cos(o3+q3)-a2*w2*w2*cos(o2)-a2*aa2*sin(o2);
ag3y:=rg3*aa3*cos(o3+q3)-rg3*w3*w3*sin(o3+q3)-a2*w2*w2*sin(o2)+a2*aa2*cos(o2);
ag3:=sqrt(ag3x*ag3x+ag3y*ag3y);
b3:=arctgt(ag3y,ag3x);
ag4x:=rg4*(-aa4*sin(o4+q4)-w4*w4*cos(o4+q4));
ag4y:=rg4*(aa4*cos(o4+q4)-w4*w4*sin(o4+q4));
ag4:=sqrt(ag4x*ag4x+ag4y*ag4y);
b4:=arctgt(ag4y,ag4x);

fo2x:=-m2*ag2x; fo2y:=-m2*ag2y; to2:=-ig2*aa2;
fo3x:=-m3*ag3x; fo3y:=-m3*ag3y; to3:=-ig3*aa3;
fo4x:=-m4*ag4x; fo4y:=-m4*ag4y; to4:=-ig4*aa4;

rrg0q(a2,q2,o2,rg2,rrg2,qq2);
rrg0q(a3,q3,o3,rg3,rrg3,qq3);
rrg0q(a4,q4,o4,rg4,rrg4,qq4);

(*constitution de la matrice*)
for i:=1 to 10 do for j:=1 to 10 do a[i,j]:=0;
for i:=1 to 5 do a[i,i]:=-1;
a[1,4]:=1; a[2,5]:=1;
a[3,1]:=-rg2*sin(o2+q2); a[3,2]:=rg2*cos(o2+q2);
a[3,4]:=-rrg2*sin(qq2); a[3,5]:=rrg2*cos(qq2);
a[4,6]:=1; a[5,7]:=1;
a[6,4]:=-rg3*sin(o3+q3); a[6,5]:=rg3*cos(o3+q3);
a[6,6]:=-rrg3*sin(qq3); a[6,7]:=rrg3*cos(qq3);
a[7,6]:=-1; a[7,8]:=-1;
a[8,7]:=-1; a[8,9]:=-1;
a[9,6]:=-rrg4*sin(qq4); a[9,7]:=-rrg4*cos(qq4);
a[9,8]:=-rg4*sin(o4+q4); a[9,9]:=rg4*cos(o4+q4);
a[1,10]:=fo2x; a[6,10]:=to3;
a[2,10]:=fo2y; a[7,10]:=fo4x;
a[3,10]:=to2; a[8,10]:=fo4y;
a[4,10]:=fo3x; a[9,10]:=to4+tl;

```

```

a[5,10]:=fo3y
End;

Procedure initM4M;
Begin a1:=0; a2:=0; a3:=0; a4:=0;
      q1:=0; q2:=0; q3:=0; q4:=0; rg2:=0; rg3:=0; rg4:=0;
      m2:=0; m3:=0; m4:=0; lg2:=0; lg3:=0; lg4:=0;
      o2:=0; o3:=0; o4:=0; w2:=0; aa2:=0; tl:=0
end;

BEGIN
  initM4M;
  M4MinitMes;
  clrscr;
  ecrivideo('connaissez-vous psy1 et o1 ? (0,N) : ',1,10,true);
  repeat
    car:=upcase(readkey);
    if (car<>'0')and(car<>'N') then beep
  until (car='0')or(car='N');
  if car='N' then calculr1psy1;
  Repeat
    menu(page,1,2); clrscr;
    retournevalm4m(1,2);
    calculteta34;
    calculs; clrscr;
    systeme(9,a);
    clrscr;
    writeln('ANALYSE CINETO-STATIQUE: M4M longueurs finies. ');
    writeln;
    writeln('Caractéristiques cinétiques. '); writeln;
    writeln('Paramètres', 'memb2':10, 'memb3':10, 'memb4':10);
    writeln(' Ui      ', o2:10:3, o3:10:3, o4:10:4);
    writeln(' wi      ', w2:10:3, w3:10:3, w4:10:3);
    writeln(' Oi      ', aa2:10:3, aa3:10:3, aa4:10:3);
    writeln(' Agix   ', ag2x:10:3, ag3x:10:3, ag4x:10:3);
    writeln(' Agiy   ', ag2y:10:3, ag3y:10:3, ag4y:10:3);
    writeln(' |Agi|  ', ag2:10:3, ag3:10:3, ag4:10:3);
    writeln(' B(Ag)  ', b2:10:3, b3:10:3, b4:10:3);
    writeln; writeln;
    writeln('Forces dynamiques au centre de gravité. '); writeln;
    writeln('Paramètres', 'memb2':10, 'memb3':10, 'memb4':10);
    writeln(' Foix   ', fo2x:10:3, fo3x:10:3, fo4x:10:4);
    writeln(' Foiy   ', fo2y:10:3, fo3y:10:3, fo4y:10:3);
    writeln(' Toi    ', to2:10:3, to3:10:3, to4:10:3);
    writeln('Forces dynamiques aux joints et couple d'entree. ');
    repaat until keypressed; effecr(3,25);
    writeln;
    writeln(' F12x = ', a[10,1]:15:4);
    writeln(' F12y = ', a[10,2]:15:4);
    writeln(' Ts    = ', a[10,3]:15:4);
    writeln(' F23x = ', a[10,4]:15:4);
    writeln(' F23y = ', a[10,5]:15:4);
    writeln(' F34x = ', a[10,6]:15:4);
    writeln(' F34y = ', a[10,7]:15:4);
    writeln(' F14x = ', a[10,8]:15:4);
    writeln(' F14y = ', a[10,9]:15:4);
    ecrivideo('Voulez-vous imprimer les resultats ? (0,N) : ',1,1,true);
    repeat
      car:=upcase(readkey);
      if (car<>'0')and(car<>'N') then beep
    until (car='0')or(car='N');
    if car='0' then begin
      writeln(lst, 'ANALYSE CINETO-STATIQUE: M4M longueurs finies. ');
      writeln(lst);
      writeln(lst, 'Caractéristiques cinétiques. '); writeln;
      writeln(lst, 'Paramètres', 'memb2':10, 'memb3':10, 'memb4':10);
      writeln(lst, ' Ui      ', o2:10:3, o3:10:3, o4:10:4);

```

```

writeln(lst,' wi      ',w2:10:3,w3:10:3,w4:10:3);
writeln(lst,' Ói      ',aa2:10:3,aa3:10:3,aa4:10:3);
writeln(lst,' Agix    ',ag2x:10:3,ag3x:10:3,ag4x:10:3);
writeln(lst,' Agiy    ',ag2y:10:3,ag3y:10:3,ag4y:10:3);
writeln(lst,' |Agi|   ',ag2:10:3,ag3:10:3,ag4:10:3);
writeln(lst,' B(Ag)   ',b2:10:3,b3:10:3,b4:10:3);
writeln(lst); writeln(lst);
writeln(lst,'Forces dynamiques au centre de gravité. '); writeln;
writeln(lst,'Paramètres','memb2':10,'memb3':10,'memb4':10);
writeln(lst,' Foix    ',fo2x:10:3,fo3x:10:3,fo4x:10:4);
writeln(lst,' Foiy    ',fo2y:10:3,fo3y:10:3,fo4y:10:3);
writeln(lst,' Toi     ',to2:10:3,to3:10:3,to4:10:3);
writeln(lst,'Forces dynamiques aux joints et couple d'entree. ');
writeln(lst);
writeln(lst,' F12x = ',a[10,1]:15:4);
writeln(lst,' F12y = ',a[10,2]:15:4);
writeln(lst,' Ts     = ',a[10,3]:15:4);
writeln(lst,' F23x = ',a[10,4]:15:4);
writeln(lst,' F23y = ',a[10,5]:15:4);
writeln(lst,' F34x = ',a[10,6]:15:4);
writeln(lst,' F34y = ',a[10,7]:15:4);
writeln(lst,' F14x = ',a[10,8]:15:4);
writeln(lst,' F14y = ',a[10,9]:15:4); end;
ecrivideo('Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.',1,24,true);
  repeat car:=readkey;
    if (car<>esc)and(car<>cr) then beep
  until (car=esc) or (car=cr);
Until car=esc;
End;
PROCEDURE ForcesJointsMC;
Var
  a2,a3,a4,e,
  q1,q2,q3,q4,rg2,rg3,
  qq,qq2,qq3,qq4,rrg2,rrg3,rrg4,
  m2,m3,m4,lg2,lg3,lg4,
  s,bd,b,d,c,
  o2,o3,o3p,o4,w2,w3,w4,aa2,aa3,aa4,
  ag2x,ag2y,ag3x,ag3y,ag4x,ag4y,
  ag2,b2,ag3,b3,ag4,b4,VC,AC,
  Fo2x,Fo3x,Fo4,
  Fo2y,Fo3y,
  To2,To3,To4,fl,cosa,sina
  x
  a
  tabplfb
  entete
  l,cc
  :real;
  :array[1..maxdim] of real;
  :typematrice;
  :chtabtype;
  :string;
  :byte;
Procedure MCinitMes;
begin
  with page[1] do
  begin taille:=13;
  mot[1]:=' a2      '; val[1]:=a2; mot[2]:=' a3      '; val[2]:=a3;
  mot[3]:='upset e  '; val[3]:=e; mot[4]:=' Y       '; val[4]:=q1;
  mot[5]:=' b2      '; val[5]:=q2; mot[6]:=' b3      '; val[6]:=q3;
  mot[7]:=' rg2     '; val[7]:=rg2; mot[8]:=' rg3     '; val[8]:=rg3;
  mot[9]:=' m2      '; val[9]:=m2; mot[10]:=' m3      '; val[10]:=m3;
  mot[11]:=' m4      '; val[11]:=m4; mot[12]:=' lg2     '; val[12]:=lg2;
  mot[13]:=' lg3     '; val[13]:=lg3;
  end;

  with page[2] do
  begin taille:=4;
  mot[1]:=' Ú2      '; mot[2]:=' w2      ';
  mot[3]:=' o2      '; val[1]:=o2; val[2]:=w2; val[3]:=aa2;
  mot[4]:=' Fl      '; val[4]:=fl;
  end;
end;

```

```

end;

Procedure retournevalmc(debut,fin: byte);
var k:byte;
begin
  for k:=debut to fin do
    case k of
      1: with page[1] do      begin
          a2:=val[1];  a3:=val[2];  e:=val[3];  q1:=val[4];
          q2:=val[5];  q3:=val[6];  rg2:=val[7];  rg3:=val[8];
          m2:=val[9];  m3:=val[10]; m4:=val[11]; ig2:=val[12];
          ig3:=val[13];  end;
      2: with page[2] do      begin
          o2:=val[1];  w2:=val[2];  aa2:=val[3];  f1:=val[4]  end;
    end;
  end;
end;

Procedure Calculo3b;
Begin  sina:=(-a2*sin(o2-q1)-e)/a3;
      cosa:=sqrt(1-sqr(sina));
      b:=arctgt(sina,cosa);
      o3:=q1-b-pi;
      o3p:=q1+b-2*pi;
      str(o3:8:3,tabplfb[1,1]);
      str(o3p:8:3,tabplfb[2,1]);
      entete:='Choisissez U3 entre : ';
      plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,cc);
      if cc=2 then begin o3:=o3p; b:=pi-b end;
      o4:=q1-0.5*pi;
End;

Procedure calculs;
Procedure rrgQq(a,q,o,rg:real; var rrg,qq:real);
begin  rrg:=sqrt(rg*rg+a*a-2*rg*a*cos(q));
      if rg=0 then qq:=o;
      if (rg=a)and(q=0) then qq:=0;
      if (rg<>0) and ((rg<>a)or(q<>0)) then
        begin  sina:=a*sin(q)/rrg;
              cosa:=(a*a-rrg*rrg-rg*rg)/(-2*rrg*rg);
              qq:=q+o-pi+arctgt(sina,cosa);
        end;
end;

Begin
w3:=(-a2*cos(o2-q1)*w2)/(a3*cos(o3-q1));
VC:=a2*w2*sin(o2-q1)+a3*w3*sin(o3-q1);
aa3:=aa2*w3/w2 + (a2*w2*w2*sin(o2-q1))/(a3*cos(o3-q1)) + w3*w3*(sin(o3-q1)/cos(o3-q1));
AC:=a2*w2*w2*cos(o2-q1)+a2*aa2*sin(o2-q1)+a3*w3*w3*cos(o3-q1)+a3*aa3*sin(o3-q1);
ag2x:=rg2*(-aa2*sin(o2+q2)-w2*w2*cos(o2+q2));
ag2y:=rg2*(aa2*cos(o2+q2)-w2*w2*sin(o2+q2));
ag2:=sqrt(ag2x*ag2x+ag2y*ag2y);
b2:=arctgt(ag2y,ag2x);
ag3x:=-rg3*aa3*sin(o3+q3)-rg3*w3*w3*cos(o3+q3)-a2*w2*w2*cos(o2)-a2*aa2*sin(o2);
ag3y:=rg3*aa3*cos(o3+q3)-rg3*w3*w3*sin(o3+q3)-a2*w2*w2*sin(o2)+a2*aa2*cos(o2);
ag3:=sqrt(ag3x*ag3x+ag3y*ag3y);
b3:=arctgt(ag3y,ag3x);
ag4x:=AC*cos(pi-q1);
ag4y:=AC*sin(pi-q1);
ag4:=AC;
b4:=arctgt(ag4y,ag4x);

fo2x:=-m2*ag2x;  fo2y:=-m2*ag2y;  to2:=-ig2*aa2;
fo3x:=-m3*ag3x;  fo3y:=-m3*ag3y;  to3:=-ig3*aa3;
fo4:=-m4*ag4;

rrgQq(a2,q2,o2,rg2,rrg2,qq2);
rrgQq(a3,q3,o3,rg3,rrg3,qq3);

```

```

(*constitution de la matrice*)
for i:=1 to 8 do for j:=1 to 8 do a[i,j]:=0;
for i:=1 to 5 do a[i,i]:=-1;
a[1,4]:=1; a[2,5]:=1;
a[3,1]:=-rg2*sin(o2+q2); a[3,2]:=rg2*cos(o2+q2);
a[3,4]:=-rrg2*sin(qq2); a[3,5]:=rrg2*cos(qq2);
a[4,6]:=1; a[5,7]:=1;
a[6,4]:=-rg3*sin(o3+q3); a[6,5]:=rg3*cos(o3+q3);
a[6,6]:=-rrg3*sin(qq3); a[6,7]:=rrg3*cos(qq3);
a[7,6]:=cos(q1); a[7,7]:=sin(q1);
a[1,8]:=fo2x; a[6,8]:=to3;
a[2,8]:=fo2y; a[7,8]:=fo4+f1;
a[3,8]:=to2;
a[4,8]:=fo3x;
a[5,8]:=fo3y
End;

Procedure initMC;
Begin e:=0; a2:=0; a3:=0; a4:=0;
q1:=0; q2:=0; q3:=0; q4:=0; rg2:=0; rg3:=0;
m2:=0; m3:=0; m4:=0; lg2:=0; lg3:=0;
o2:=0; o3:=0; o4:=0; w2:=0; aa2:=0; fl:=0
end;

BEGIN
initMC;
MCinitMes;
clrscr;
Repeat
menu(page,1,2); clrscr;
retournevalmc(1,2);
calculo3b;
calculs; clrscr;
systeme(7,a);
clrscr;
writeln('Caractéristiques cinétiques. '); writeln;
writeln('Paramètres', 'memb2':10, 'memb3':10, 'memb4':10);
writeln(' Ui ', o2:10:3, o3:10:3, o4:10:4);
writeln(' wi ', w2:10:3, w3:10:3, 0:10);
writeln(' Oi ', aa2:10:3, aa3:10:3, 0:10);
writeln(' VC ', '-':10, '-':10, vc:10:3);
writeln(' Agix ', ag2x:10:3, ag3x:10:3, ag4x:10:3);
writeln(' Agiy ', ag2y:10:3, ag3y:10:3, ag4y:10:3);
writeln(' |Agi| ', ag2:10:3, ag3:10:3, ag4:10:3);
writeln(' B(Ag) ', b2:10:3, b3:10:3, b4:10:3);
repeat until keypressed; effecr(3,25);
writeln('Forces dynamiques au centre de gravité. '); writeln;
writeln(' Paramètres', 'memb2':10, 'memb3':10, 'memb4':10);
writeln(' Foix ', fo2x:10:3, fo3x:10:3, fo4*cos(pi-q1):10:4);
writeln(' Foiy ', fo2y:10:3, fo3y:10:3, fo4*sin(pi-q1):10:3);
writeln(' Foi ', sqrt(fo2y*fo2y+fo2x*fo2x):10:3, sqrt(fo3y*fo3y+fo3x*fo3x):10:3, fo4:10:3);
writeln(' Toi ', to2:10:3, to3:10:3, 0:10);
writeln; writeln;
writeln('Forces dynamiques aux joints. '); writeln;
writeln(' F12x = ', a[8,1]:15:4);
writeln(' F12y = ', a[8,2]:15:4);
writeln(' Ts = ', a[8,3]:15:4);
writeln(' F23x = ', a[8,4]:15:4);
writeln(' F23y = ', a[8,5]:15:4);
writeln(' F34x = ', a[8,6]:15:4);
writeln(' F34y = ', a[8,7]:15:4);
ecrivideo('Voulez-vous imprimer les resultats ? (0,N) : ',1,1,true);
repeat
car:=uppercase(readkey);
if (car<>'0')and(car<>'N') then beep
until (car='0')or(car='N');
if car='0' then begin

```

```

writeln(lst,'Caractéristiques cinétiques.');
```

```

writeln(lst,'Paramètres','memb2':10,'memb3':10,'memb4':10);
writeln(lst,'  Úi      ',o2:10:3,o3:10:3,o4:10:4);
writeln(lst,'  wi      ',w2:10:3,w3:10:3,0:10);
writeln(lst,'  Ói      ',aa2:10:3,aa3:10:3,0:10);
writeln(lst,'  VC      ','-':10,'-':10,vc:10:3);
writeln(lst,'  Agix     ',ag2x:10:3,ag3x:10:3,ag4x:10:3);
writeln(lst,'  Agiy     ',ag2y:10:3,ag3y:10:3,ag4y:10:3);
writeln(lst,'  |Agj|     ',ag2:10:3,ag3:10:3,ag4:10:3);
writeln(lst,'  B(Ag)    ',b2:10:3,b3:10:3,b4:10:3);
writeln(lst); writeln(lst);
writeln(lst,'Forces dynamiques au centre de gravité.');
```

```

writeln(lst,'Paramètres','memb2':10,'memb3':10,'memb4':10);
writeln(lst,' Foix     ',fo2x:10:3,fo3x:10:3,fo4*cos(pi-q1):10:4);
writeln(lst,' Foiy     ',fo2y:10:3,fo3y:10:3,fo4*sin(pi-q1):10:3);
writeln(lst,' Foi      ',sqrt(fo2y*fo2y+fo2x*fo2x):10:3,sqrt(fo3y*fo3y+fo3x*fo3x):10:3,fo4:10:3);
writeln(lst,' Toi      ',to2:10:3,to3:10:3,0:10);
writeln(lst);
writeln(lst,'Forces dynamiques aux joints.');
```

```

writeln(lst,' F12x = ',a[8,1]:15:4);
writeln(lst,' F12y = ',a[8,2]:15:4);
writeln(lst,' Ts    = ',a[8,3]:15:4);
writeln(lst,' F23x = ',a[8,4]:15:4);
writeln(lst,' F23y = ',a[8,5]:15:4);
writeln(lst,' F34x = ',a[8,6]:15:4);
writeln(lst,' F34y = ',a[8,7]:15:4); end;
ecrivideo('Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.',1,24,true);
  repeat car:=readkey;
    if (car<>esc)and(car<>cr) then beep
  until (car=esc) or (car=cr);
Until car=esc;
End;
END.
```

Listing programme ADM: chapitres 3 & 4 et programme principal

```

PROGRAM Analyse_dynamique_des_mecanismes;
uses crt,printer,support,chap12;

(***** Ecole polytechnique de Thies      *****)
(***** Projet de fin d'etude: juillet 1992 *****)
(***** Auteur: E.I. Yvon Polycarpe DOSSA *****)
(***** Directeur: M. Mohamadou L. DIALLO *****)

PROCEDURE ROTOR;

Var car1,car2,car3,car4:char;
i:byte;
n:integer;
chaîne:string;
z,o,r,m,x,y: array[1..15] of real;
za,zb,ra,rb,ma,mb,oma,omb,oa,ob,ia,ib,sx,sy,sn,on :real;
iax,iay,ibx,iby : real;

Procédure rotorInitMes;
begin
with page[4] do
begin taille:=18;
mot[1]:= ' z1      '; val[1]:=z[1]; mot[2]:= ' m1      '; val[2]:=m[1];
mot[3]:= ' r1      '; val[3]:=r[1]; mot[4]:= ' Ú1      '; val[4]:=o[1];
mot[5]:= ' x1      '; val[5]:=x[1]; mot[6]:= ' y1      '; val[6]:=y[1];
mot[7]:= ' z2      '; val[7]:=z[2]; mot[8]:= ' m2      '; val[8]:=m[2];
mot[9]:= ' r2      '; val[9]:=r[2]; mot[10]:= ' Ú2      '; val[10]:=o[2];
mot[11]:= ' x2      '; val[11]:=x[2]; mot[12]:= ' y2      '; val[12]:=y[2];
mot[13]:= ' z3      '; val[13]:=z[3]; mot[14]:= ' m3      '; val[14]:=m[3];
mot[15]:= ' r3      '; val[15]:=r[3]; mot[16]:= ' Ú3      '; val[16]:=o[3];
mot[17]:= ' x3      '; val[17]:=x[3]; mot[18]:= ' y3      '; val[18]:=y[3];
end;
with page[5] do
begin taille:=18;
mot[1]:= ' z4      '; val[1]:=z[4]; mot[2]:= ' m4      '; val[2]:=m[4];
mot[3]:= ' r4      '; val[3]:=r[4]; mot[4]:= ' Ú4      '; val[4]:=o[4];
mot[5]:= ' x4      '; val[5]:=x[4]; mot[6]:= ' y4      '; val[6]:=y[4];
mot[7]:= ' z5      '; val[7]:=z[5]; mot[8]:= ' m5      '; val[8]:=m[5];
mot[9]:= ' r5      '; val[9]:=r[5]; mot[10]:= ' Ú5      '; val[10]:=o[5];
mot[11]:= ' x5      '; val[11]:=x[5]; mot[12]:= ' y5      '; val[12]:=y[5];
mot[13]:= ' z6      '; val[13]:=z[6]; mot[14]:= ' m6      '; val[14]:=m[6];
mot[15]:= ' r6      '; val[15]:=r[6]; mot[16]:= ' Ú6      '; val[16]:=o[6];
mot[17]:= ' x6      '; val[17]:=x[6]; mot[18]:= ' y6      '; val[18]:=y[6];
end;
with page[6] do
begin taille:=18;
mot[1]:= ' z7      '; val[1]:=z[7]; mot[2]:= ' m7      '; val[2]:=m[7];
mot[3]:= ' r7      '; val[3]:=r[7]; mot[4]:= ' Ú7      '; val[4]:=o[7];
mot[5]:= ' x7      '; val[5]:=x[7]; mot[6]:= ' y7      '; val[6]:=y[7];
mot[7]:= ' z8      '; val[7]:=z[8]; mot[8]:= ' m8      '; val[8]:=m[8];
mot[9]:= ' r8      '; val[9]:=r[8]; mot[10]:= ' Ú8      '; val[10]:=o[8];
mot[11]:= ' x8      '; val[11]:=x[8]; mot[12]:= ' y8      '; val[12]:=y[8];
mot[13]:= ' z9      '; val[13]:=z[9]; mot[14]:= ' m9      '; val[14]:=m[9];
mot[15]:= ' r9      '; val[15]:=r[9]; mot[16]:= ' Ú9      '; val[16]:=o[9];
mot[17]:= ' x9      '; val[17]:=x[9]; mot[18]:= ' y9      '; val[18]:=y[9];
end;
with page[7] do
begin taille:=18;
mot[1]:= ' z10     '; val[1]:=z[10]; mot[2]:= ' m10     '; val[2]:=m[10];
mot[3]:= ' r10     '; val[3]:=r[10]; mot[4]:= ' Ú10     '; val[4]:=o[10];
mot[5]:= ' x10     '; val[5]:=x[10]; mot[6]:= ' y10     '; val[6]:=y[10];
mot[7]:= ' z11     '; val[7]:=z[11]; mot[8]:= ' m11     '; val[8]:=m[11];
mot[9]:= ' r11     '; val[9]:=r[11]; mot[10]:= ' Ú11     '; val[10]:=o[11];
mot[11]:= ' x11     '; val[11]:=x[11]; mot[12]:= ' y11     '; val[12]:=y[11];
mot[13]:= ' z12     '; val[13]:=z[12]; mot[14]:= ' m12     '; val[14]:=m[12];
mot[15]:= ' r12     '; val[15]:=r[12]; mot[16]:= ' Ú12     '; val[16]:=o[12];
end;

```



Listing programme ADM: chapitres 3 & 4 et programme principal

```

    mot[17]:= ' x12      '; val[17]:=x[12]; mot[18]:= ' y12      '; val[18]:=y[12]
end;
with page[8] do
begin taille:=18;
mot[1]:= ' z13      '; val[1]:=z[13]; mot[2]:= ' m13      '; val[2]:=m[13];
mot[3]:= ' r13      '; val[3]:=r[13]; mot[4]:= ' Ú13      '; val[4]:=o[13];
mot[5]:= ' x13      '; val[5]:=x[13]; mot[6]:= ' y13      '; val[6]:=y[13];
mot[7]:= ' z14      '; val[7]:=z[14]; mot[8]:= ' m14      '; val[8]:=m[14];
mot[9]:= ' r14      '; val[9]:=r[14]; mot[10]:= ' Ú14      '; val[10]:=o[14];
mot[11]:= ' x14     '; val[11]:=x[14]; mot[12]:= ' y14      '; val[12]:=y[14];
mot[13]:= ' z15     '; val[13]:=z[15]; mot[14]:= ' m15      '; val[14]:=m[15];
mot[15]:= ' r15     '; val[15]:=r[15]; mot[16]:= ' Ú15      '; val[16]:=o[15];
mot[17]:= ' x15     '; val[17]:=x[15]; mot[18]:= ' y15      '; val[18]:=y[15]
end;
end;
Procedure retournevalRot(debut,fin:byte);
var k:byte;
begin
for k:=debut to fin do
case k of
4: with page[4] do begin
z[1]:=val[1]; m[1]:=val[2]; r[1]:=val[3]; o[1]:=val[4]; x[1]:=val[5]; y[1]:=val[6];
z[2]:=val[7]; m[2]:=val[8]; r[2]:=val[9]; o[2]:=val[10]; x[2]:=val[11]; y[2]:=val[12];
z[3]:=val[13]; m[3]:=val[14]; r[3]:=val[15]; o[3]:=val[16]; x[3]:=val[17]; y[3]:=val[18]
end;
5: with page[5] do begin
z[4]:=val[1]; m[4]:=val[2]; r[4]:=val[3]; o[4]:=val[4]; x[4]:=val[5]; y[4]:=val[6];
z[5]:=val[7]; m[5]:=val[8]; r[5]:=val[9]; o[5]:=val[10]; x[5]:=val[11]; y[5]:=val[12];
z[6]:=val[13]; m[6]:=val[14]; r[6]:=val[15]; o[6]:=val[16]; x[6]:=val[17]; y[6]:=val[18]
end;
6: with page[6] do begin
z[7]:=val[1]; m[7]:=val[2]; r[7]:=val[3]; o[7]:=val[4]; x[7]:=val[5]; y[7]:=val[6];
z[8]:=val[7]; m[8]:=val[8]; r[8]:=val[9]; o[8]:=val[10]; x[8]:=val[11]; y[8]:=val[12];
z[9]:=val[13]; m[9]:=val[14]; r[9]:=val[15]; o[9]:=val[16]; x[9]:=val[17]; y[9]:=val[18]
end;
7: with page[7] do begin
z[10]:=val[1]; m[10]:=val[2]; r[10]:=val[3]; o[10]:=val[4]; x[10]:=val[5]; y[10]:=val[6];
z[11]:=val[7]; m[11]:=val[8]; r[11]:=val[9]; o[11]:=val[10]; x[11]:=val[11]; y[11]:=val[12];
z[12]:=val[13]; m[12]:=val[14]; r[12]:=val[15]; o[12]:=val[16]; x[12]:=val[17]; y[12]:=val[18]
end;
8: with page[8] do begin
z[13]:=val[1]; m[13]:=val[2]; r[13]:=val[3]; o[13]:=val[4]; x[13]:=val[5]; y[13]:=val[6];
z[14]:=val[7]; m[14]:=val[8]; r[14]:=val[9]; o[14]:=val[10]; x[14]:=val[11]; y[14]:=val[12];
z[15]:=val[13]; m[15]:=val[14]; r[15]:=val[15]; o[15]:=val[16]; x[15]:=val[17]; y[15]:=val[18]
end;
end; end;
procedure imprimerS;
var i:byte;
    mx,my:real;
begin ecrivideo('Debalancement masse rayon angle Xi MiXi Yi MiYi ',1,3,true);
writeln;
for i:=1 to n do
begin mx:=m[i]*x[i];
my:=m[i]*y[i];
writeln(i:8,m[i]:10:3,r[i]:7:3,o[i]:7:3,x[i]:10:3,mx:10:3,y[i]:8:3,my:8:3);
end;
writeln('Somme', ' ':45,sx:10:3,sy:8:3); writeln;
writeln(' | Sn | = ',sn:10:3);
writeln(' Ún = ',on:10:3); writeln;
case car2 of
'R': writeln('Pour une position de r= ',ra:8:3,' il nous faudra une masse m= ',ma:8:3);
'M': writeln('Pour une masse de m= ',ma:8:3,' il nous faudra une position r= ',ra:8:3);
end;
end;
procedure imprimerD;
var i:byte;

```

Listing programme ADM: chapitres 3 & 4 et programme principal

```

mx,my,mr:real;
begin ecrivideo(' Plan masse rayon Zi MiRiZi angle MixiZi MiYiZi ',1,3,true);
writeln;
for i:=1 to n do
begin mx:=m[i]*x[i]*(z[i]-za);
my:=m[i]*y[i]*(z[i]-za);
mr:=m[i]*r[i]*(z[i]-za);
if i=1 then writeln(' IB ',i:4,m[i]:8:3,r[i]:7:3,z[i]:7:3,mr:9:3,o[i]:7:3,mx:10:3,my:10:3)
else writeln('
',i:4,m[i]:8:3,r[i]:7:3,z[i]-za:7:3,mr:9:3,o[i]:7:3,mx:10:3,my:10:3)
end;
writeln(' Somme', ' ':38,lbx:10:3,lby:10:3); writeln;
for i:=1 to n do
begin mx:=m[i]*x[i]*(zb-z[i]);
my:=m[i]*y[i]*(zb-z[i]);
mr:=m[i]*r[i]*(zb-z[i]);
if i=1 then writeln(' IA ',i:4,m[i]:8:3,r[i]:7:3,z[i]:7:3,mr:9:3,o[i]:7:3,mx:10:3,my:10:3)
else writeln('
',i:4,m[i]:8:3,r[i]:7:3,zb-z[i]:7:3,mr:9:3,o[i]:7:3,mx:10:3,my:10:3)
end;
writeln(' Somme', ' ':38,lax:10:3,lay:10:3); writeln;
writeln(' IB | = ',IB:10:3);
writeln(' Ub = ',ob:10:3); writeln;
writeln(' IA | = ',Ia:10:3);
writeln(' Ua = ',oa:10:3); writeln;
case car4 of
'R': writeln('Pour une position de rb = ',rb:8:3,' il nous faudra une masse mb = ',mb:8:3);
'M': writeln('Pour une masse de mb = ',mb:8:3,' il nous faudra une position rb = ',rb:8:3);
end;
case car3 of
'R': writeln('Pour une position de ra = ',ra:8:3,' il nous faudra une masse ma = ',ma:8:3);
'M': writeln('Pour une masse de ma = ',ma:8:3,' il nous faudra une position ra = ',ra:8:3);
end;
end;

procedure initR;
var i:byte;
begin
for i:=1 to 15 do begin z[i]:=0;m[i]:=0;r[i]:=0;o[i]:=0;x[i]:=0;y[i]:=0 end;
iax:=0; iay:=0; ibx:=0; iby:=0; sx:=0; sy:=0; n:=15; za:=0; zb:=0;
ma:=0; ra:=0; mb:=0; rb:=0;
end;

Begin
initR;
Repeat
if menuy=1 then car1:='S';
if menuy=2 then car1:='D';
ecrivideo('Donnez le nombre de débalancements <',1,10,true);
str(n:3,chaine);
ecrivideo(chaine,wherex,10,true);
ecrivideo('> : ',wherex,10,true);
car:=readkey;
if car=nul then car:=readkey;
if (car<>esc)and(car<>cr) then lireEntierP(n);
if car1='D' then begin
ecrivideo('Entrez la position (za) du plan A <',1,12,true);
str(za:8:3,chaine);
ecrivideo(chaine,wherex,12,true);
ecrivideo('> : ',wherex,12,true);
car:=readkey;
if car=nul then car:=readkey;
if (car<>esc)and(car<>cr) then lire reel(za);
ecrivideo('Entrez la position (zb) du plan B <',1,14,true);
str(zb:8:3,chaine);
ecrivideo(chaine,wherex,14,true);
ecrivideo('> : ',wherex,14,true);

```

```

car:=readkey;
if car=nul then car:=readkey;
if (car<>esc)and(car<>cr) then lirereel(zb) end;
clrscr;
rotorInitMes;
menu(page,4,8);
retournevalRot(4,8);
if car1='S' then
begin for i:=1 to n do
begin sx:=sx+m[i]*x[i]; sy:=sy+m[i]*y[i] end;
sn:=sqrt(sqr(sx)+sqr(sy));
on:=arctgt(sy,sx);
clrscr;
ecrivideo('connaissez-vous la position(Rayon) ou ',1,7,true);
ecrivideo('la Masse du contrepoids (R, M ou ESC) ? : ',1,8,true);
repeat
car2:=upcase(readkey);
if (car2<>'R') and (car2<>'M')and (car2<>esc) then beep
until (car2='R') or (car2='M') or (car2=esc);
write(car2); delay(100);
if car2='R' then begin ecrivideo('Entrez le rayon du contrepoids < ',1,10,true);
str(ra:8:3,chaîne);
ecrivideo(chaîne,wherex,10,true);
ecrivideo(' > : ',wherex,10,true);
car:=readkey;
if car=nul then car:=readkey;
if (car<>esc)and(car<>cr) then lirereel(ra);
ma:=sn/ra end;
if car2='M' then begin ecrivideo('Entrez la masse du contrepoids < ',1,10,true);
str(ma:8:3,chaîne);
ecrivideo(chaîne,wherex,10,true);
ecrivideo(' > : ',wherex,10,true);
car:=readkey;
if car=nul then car:=readkey;
if (car<>esc)and(car<>cr) then lirereel(ma);
ra:=sn/ma end;
clrscr;
imprimerS;
end
else begin
for i:=1 to n do
begin ibx:=ibx+m[i]*x[i]*(z[i]-za);
iby:=iby+m[i]*y[i]*(z[i]-za);
iax:=iax+m[i]*x[i]*(zb-z[i]);
iay:=iay+m[i]*y[i]*(zb-z[i]);
end;
ia:=sqrt(sqr(iax)+sqr(iay));
oa:=arctgt(iay,iax);
oma:=oa+pi;
ib:=sqrt(sqr(ibx)+sqr(iby));
ob:=arctgt(iby,ibx);
omb:=ob+pi;
clrscr;
ecrivideo('connaissez-vous la position (Rayon) ou la Masse ',1,7,true);
ecrivideo(' du contrepoids "a" (R, M ou ESC) ? : ',1,8,true);
repeat
car3:=upcase(readkey);
if (car3<>'R') and (car3<>'M')and (car3<>esc) then beep
until (car3='R') or (car3='M') or (car3=esc);
write(car3);
if car3='R' then begin ecrivideo('Entrez le rayon de "A" < ',1,9,true);
str(ra:8:3,chaîne);
ecrivideo(chaîne,wherex,9,true);
ecrivideo(' > : ',wherex,9,true);
car:=readkey;
if car=nul then car:=readkey;
if (car<>esc)and(car<>cr) then lirereel(ra);

```

Listing programme ADM: chapitres 3 & 4 et programme principal

```

        ma:=ia/(ra*(zb-za)) end;
if car3='M' then begin ecrivideo('Entrez la masse de "A" < ',1,9,true);
                      str(ma:8:3,chaîne);
                      ecrivideo(chaîne,wherex,9,true);
                      ecrivideo(' > : ',wherex,9,true);
                      car:=readkey;
                      if car=nul then car:=readkey;
                      if (car<>esc)and(car<>cr) then lirereel(ma);
                      ra:=ia/(ma*(zb-za)) end;
ecrivideo('connaissez-vous la position (Rayon) ou la Masse ',1,12,true);
ecrivideo(' du contrepoids "b" (R, M ou ESC) ? : ',1,13,true);
repeat
  car4:=upcase(readkey);
  if (car4<>'R') and (car4<>'M')and (car4<>esc) then beep
until (car4='R') or (car4='M') or (car4=esc);
write(car4);
if car4='R' then begin ecrivideo('Entrez le rayon de "B" < ',1,14,true);
                      str(rb:8:3,chaîne);
                      ecrivideo(chaîne,wherex,14,true);
                      ecrivideo(' > : ',wherex,14,true);
                      car:=readkey;
                      if car=nul then car:=readkey;
                      if (car<>esc)and(car<>cr) then lirereel(rb);
                      mb:=ib/(rb*(zb-za)) end;
if car4='M' then begin ecrivideo('Entrez la masse de "B" < ',1,14,true);
                      str(mb:8:3,chaîne);
                      ecrivideo(chaîne,wherex,14,true);
                      ecrivideo(' > : ',wherex,14,true);
                      car:=readkey;
                      if car=nul then car:=readkey;
                      if (car<>esc)and(car<>cr) then lirereel(mb);
                      rb:=ib/(mb*(zb-za)) end;

clrscr;
imprimerD;

end;
ecrivideo('Taper ESC pour sortir du module ',1,23,true);
ecrivideo(' ou CR pour retourner changer les valeurs.',1,24,true);
repeat car:=readkey;
  if car=nul then car:=readkey;
  if (car<>esc) and (car<>cr) then beep
until (car=esc) or (car=cr);
reversevideo(false); clrscr;
Until car=esc;
End;

PROCEDURE Balance;

Var
  a1,a2,a3,a4,
  qo1,qo2,qo3,qo4,ro2,ro3,ro4,
  qqo3,rro3,mo2,mo3,mo4,
  qe2,re2,me2,qe3,re3,qqe3,rre3,me3,qe4,re4,me4,
  q2,r2,m2,q3,r3,m3,q4,r4,m4,qq3,rr3,
  mr2,mr3,mr4,mre2,mre4,mrr3,mre3,mrre3,
  r,rx,ry,ggo,gge :real;
chaîne,entête:string;
chr3,chr3,chr3,chr3,chr3 :string[4];
h3,e3,d3 :real;
mv2,mv3,mv4,igo2,igo3,igo4,ig2,ig3,ig4,
k2,k3,k4,ke2,ke4,lee2,lee4,
mi3,qi3,igi3,ri3,rri3,qqi3,qie3,qqie3,mie3,rie3,rrie3,Igie3 :real;
ma3,ma3p,kgp,kgap,rga,rgap :real;
tablfb: chtabtype;
l,c,c1,memb1,memb2: byte;
reconfig1,reconfig2,allbalance:boolean;

Procedure FBinitMes;

```

```

begin
  with page[1] do begin
    if menuz=1 then begin taille:=14;
      mot[1]:= ' a1      '; val[1]:=a1; mot[2]:= ' a2      '; val[2]:=a2;
      mot[3]:= ' a3      '; val[3]:=a3; mot[4]:= ' a4      '; val[4]:=a4;
      mot[5]:= ' b^1     '; val[5]:=qo1; mot[6]:= ' b^2     '; val[6]:=qo2;
      mot[7]:= ' b^3     '; val[7]:=qo3; mot[8]:= ' b^4     '; val[8]:=qo4;
      mot[9]:= ' r^2     '; val[9]:=ro2; mot[10]:= ' r^3     '; val[10]:=ro3;
      mot[11]:= ' r^4     '; val[11]:=ro4; mot[12]:= ' m^2     '; val[12]:=mo2;
      mot[13]:= ' m^3     '; val[13]:=mo3; mot[14]:= ' m^4     '; val[14]:=mo4;
    end;
    if menuz=2 then begin taille:=11;
      mot[1]:= ' a2      '; val[1]:=a2; mot[2]:= ' a3      '; val[2]:=a3;
      mot[3]:= ' upset e  '; val[3]:=a1; mot[4]:= ' b      '; val[4]:=qo1;
      mot[5]:= ' b^2     '; val[5]:=qo2; mot[6]:= ' b^3     '; val[6]:=qo3;
      mot[7]:= ' r^2     '; val[7]:=ro2; mot[8]:= ' r^3     '; val[8]:=ro3;
      mot[9]:= ' m^2     '; val[9]:=mo2; mot[10]:= ' m^3     '; val[10]:=mo3;
      mot[11]:= ' m^4     '; val[11]:=mo4; end;
    end;
  with page[2] do
    begin taille:=6;
      mot[1]:= ' mv2     '; val[1]:=mv2; mot[2]:= ' mv3     '; val[2]:=mv3;
      mot[3]:= ' mv4     '; val[3]:=mv4; mot[4]:= ' lgo2    '; val[4]:=lgo2;
      mot[5]:= ' lgo3    '; val[5]:=lgo3; mot[6]:= ' lgo4    '; val[6]:=lgo4;
    end; end;
  Procedure retournevalFB;
  begin
    with page[1] do
      begin;
        if menuz=1 then begin
          a1:=val[1]; a2:=val[2]; a3:=val[3]; a4:=val[4];
          qo1:=val[5]; qo2:=val[6]; qo3:=val[7]; qo4:=val[8];
          ro2:=val[9]; ro3:=val[10]; ro4:=val[11];
          mo2:=val[12]; mo3:=val[13]; mo4:=val[14];
          end
        else begin
          a2:=val[1]; a3:=val[2]; a1:=val[3]; qo1:=val[4];
          qo2:=val[5]; qo3:=val[6]; ro2:=val[7]; ro3:=val[8];
          mo2:=val[9]; mo3:=val[10]; mo4:=val[11];
          end
        end;
    with page[2] do
      begin
        mv2:=val[1]; mv3:=val[2]; mv4:=val[3];
        lgo2:=val[4]; lgo3:=val[5]; lgo4:=val[6];
      end; end;
  Procedure initFB;
  Begin a1:=0; a2:=0; a3:=0; a4:=0; qo1:=0; qo2:=0; qo3:=0;
  qo4:=0; ro2:=0; ro3:=0; ro4:=0; mo2:=0; mo3:=0; mo4:=0;
  mv2:=0; mv3:=0; mv4:=0; lgo2:=0; lgo3:=0; lgo4:=0;
  mr3:=0; m3:=0; r3:=0; q3:=0; mrr3:=0; rr3:=0; qq3:=0;
  mr2:=0; m2:=0; r2:=0; q2:=0; mr4:=0; m4:=0; r4:=0; q4:=0;
  re2:=0; re3:=0; re4:=0; me3:=0; me2:=0; me4:=0; rre3:=0; qe4:=0;
  h3:=0; d3:=0; e3:=0; ke2:=0; ke4:=0; qe3:=0; mrrre3:=0; qqe3:=0;
  end;
  Procedure Inline3;
  var car5:char;
  begin
    if qo3<>0 then
      repeat
        vaA(1,wherey+2);
        repeat r:=qie3;
          liredefault('Fixer pi*3 différent de 0 et pi',r);
          if sin(r)=0 then beep;
          gotoxy(1,wherey);
        until sin(r)<>0;
      until sin(r)<>0;
  end;

```

```

qie3:=r; vaA(1,wherey+2);
repeat r:=ri3;
  liredefaut('Fixer ri3 la position de G "in-line"',r);
  if r=0 then beep;
  gotoxy(1,wherey);
until r<>0;
ri3:=r;
mie3:=(mo3/ri3)*(ro3*cos(qo3)-ri3-ro3*sin(qo3)*(cos(qie3)/sin(qie3)));
rie3:=-mo3*ro3*sin(qo3)/(mie3*sin(qie3));
vaA(1,wherey+2);
liredefaut('Donner le moment d'inertie Igie3 du C-P ',igie3);
Igi3:=Igo3+mo3*(ri3*ri3+ro3*ro3-2*ri3*ro3*cos(qo3))+
  Igie3+mie3*(rie3*rie3+ri3*ri3-2*rie3*ri3*cos(qie3));
entete:='La masse du contrepois donnant b3=0 est : ';
vaA(1,wherey+2);
ecrivideo(entete,1,wherey,false);
ecrivaleur(mie3,length(entete)+2,wherey,true);
entete:='La position de son centre de gravité est : ';
vaA(1,wherey+2);
ecrivideo(entete,1,wherey,false);
ecrivaleur(rie3,length(entete)+2,wherey,true);
vaA(1,wherey+3);
ecrivideo('Tapez 0 si valeurs ok sinon N ',1,wherey,true);
repeat car5:=upcase(readkey);
  if (car5<>'0')and(car5<>'N') then beep
until (car5='0')or(car5='N');
until car5='0'
else begin ri3:=ro3; qi3:=qo3; qie3:=0; mie3:=0; rie3:=0; Igie3:=0; end;
qqie3:=arctgt(rie3*sin(qie3),rie3*cos(qie3)-a3);
if sin(qqie3)<>0 then rrie3:=rie3*sin(qie3)/sin(qqie3)
  else rrie3:=(rie3*cos(qie3)-a3)/cos(qqie3);
qqi3:=arctgt(ri3*sin(qi3),ri3*cos(qi3)-a3);
if sin(qqi3)<>0 then rri3:=ri3*sin(qi3)/sin(qqi3)
  else rri3:=(ri3*cos(qi3)-a3)/cos(qqi3);
mi3:=mo3+mie3;
k3:=sqrt(ri3*rri3);
if qo3<>0 then begin
writeln(lst); vaA(1,4);
writeln(lst,'Paramètre du contrepois permettant de ramener le centre de gravité ');
writeln(lst,'de la membrure 3 sur l'entraxe'); writeln;
writeln(lst,'Paramètres      memb3  ');
writeln(lst,' b'   ',qie3:15:3);
writeln(lst,' mi'   ',mie3:15:3);
writeln(lst,' ri'   ',rie3:15:3);
writeln(lst,' b''i'  ',qqie3:15:3);
writeln(lst,' r''i'  ',rrie3:15:3);
writeln(lst,' Igi'   ',Igie3:15:3);
writeln(lst,' kgi'   ',sqrt(Igie3/mie3):15:3);
writeln(lst,' ri3   ',ri3:15:3);
writeln(lst,' mi3   ',mi3:15:3);
writeln(lst,' qi3   ',qi3:15:3);
writeln(lst,' Igi3  ',Igi3:15:3);
writeln(lst,' kgi3  ',sqrt(Igi3/mi3):15:3);
repeat until keypressed;
  end;
end;

Procedure EsurD(ad:real; var ed:real);
const k=1e-5;
var x, xp, pas, val, b, c, d:real;
begin pas:=1;
  if ad>10 then pas:=ad/10;
  x:=0;
  b:=12*ad+24;
  c:=24*ad+26;
  d:=-2*ad*ad*ad+13*ad+12*pi-10;
  val:=d;

```

```

repeat
  xp:=x;
  if val>0 then begin pas:=pas/2; x:=x-pas end;
  if val<0 then x:=x+pas;
  val:=8*x*x*x + b*x*x + c*x + d;
until abs(xp-x)<=k;
ed:=x;
end;
Procedure Asurd(ed:real; var ad:real);
const k=1e-5;
var x,xp,pas,val,a,b,c,d:real;
begin pas:=1;
  if ed>10 then pas:=ed/10;
  x:=0;
  val:=d;
  a:=-2;
  b:=0;
  c:=12*ed*ed+24*ed+13;
  d:=8*ed*ed*ed+24*ed*ed+26*ed+12*pi-10;
  repeat
    xp:=x;
    if val<0 then begin pas:=pas/2; x:=x-pas end;
    if val>0 then x:=x+pas;
    val:=a*x*x*x + b*x*x + c*x + d;
  until abs(xp-x)<=k;
  ad:=x;
end;

Procedure Pendul;
var X1,X2,dd:real;
    car5:char;
    ll,c,c1,c2:byte;
procedure calculE1;
const k=1e-5;
var x1,x1p,x2,x2p,pas,val,d:real;
begin
  d:=sqrt(a3*a3-8*(m3/(h3*ro3))*(1-pi));
  d:=0.25*(a3-d)/(1-pi);
  pas:=d/100;
  x1:=0; x2:=0;
  val:=d;
  repeat
    x1p:=x1; x2p:=x2;
    x1:=(0.5/val)*((m3/(h3*ro3))-a3*val+2*val*val-2*pi*val*val);
    Esurd(a3/val,x2);
    x2:=val*x2;
    if (((x1-x2)>0)and((x1p-x2p)<0)) or (((x1-x2)<0)and((x1p-x2p)>0))
    then begin val:=val+pas; pas:=pas/2 end;
    if (((x1-x2)>0)and((x1p-x2p)>0)) or (((x1-x2)<0)and((x1p-x2p)<0))
    then val:=val-pas;
  until (abs(x1-x2)<=k)or(val<0);
  d3:=val;
  e3:=x1;
end;
procedure calculme3;
const k=1e-5;
var A,B,aa,aap,bb,bbp,cc,ccp:real;
begin
  vaA(1,wherey+2);
  liredefautspos('Entrez le rayon de giration de la 1ère [kg*] ',kga);
  vaA(1,wherey+1);
  liredefautspos('Entrez le rayon de giration de la 2ème [kg*] ',kgap);
  vaA(1,wherey+1);
  liredefaut('Entrez la position rg* de la 1ère ',rga);
  vaA(1,wherey+1);
  liredefaut('Entrez la position rg*' de la 2ème ',rgap);
  A:=kga*kga+rga*rga; B:=kgap*kgap+rgap*rgap;

```

Listing programme ADM: chapitres 3 & 4 et programme principal

```

aa:=rri3*ri3-A;   bb:=rri3*ri3-B;   cc:=lgi3-mi3*rri3*ri3;
aap:=rga;        bbp:=-rgap;       ccp:=0;
if bb*aap-aa*bbp=0 then begin
    ma3:=0;   ma3p:=0 end
else begin ma3:=(bb*ccp-bbp*cc)/(bb*aap-aa*bbp);
           ma3p:=(aa*ccp-aap*cc)/(aa*bbp-bb*aap); end;
vaA(1,wherey+1);
ecrivideo('La 1ère masse à augmenter est : ',1,wherey,false);
ecrivaleur(ma3,wherex,wherey,true);
vaA(1,wherey+1);
ecrivideo('La 2ème masse à augmenter est : ',1,wherey,false);
ecrivaleur(ma3p,wherex,wherey,true);
end;
begin
repeat
    tabplfb[1,1]:='Barre rectangulaire   ';
    tabplfb[1,2]:='Augmented link       ';
    tabplfb[1,3]:='Cas général           ';
    entete:='Conversion du coupleur(3) en pendule physique.Choisissez.';
    vaA(1,4);
    plan(entete,1,wherey,3,1,1,wherey+2,tabplfb,ll,c);
    if (ll=1)or(ll=2) then begin
        tabplfb[1,1]:='e3-mo3   ';
        tabplfb[2,1]:='e3-h3   ';
        tabplfb[3,1]:='d3-mo3   ';
        tabplfb[4,1]:='d3-h3   ';
        tabplfb[5,1]:='mo3-h3   ';
        entete:='Fixez un couple de paramètres.';
        plan(entete,1,wherey+3,1,5,1,wherey+5,tabplfb,l,c1);
        ro3:=a3/2;   rro3:=a3/2;
        qo3:=0;     qqo3:=pi; end;
    if (ll=1) then
        case c1 of
            1: begin entete:='Entrer le débordement "e3" :'; vaA(1,wherey+2);
                liredefaultint(entete,-0.5*a3,-2*a3*(1-sqrt(3)),e3);
                entete:='Entrer la masse "mo3" :'; vaA(1,wherey+2);
                liredefaultSpos(entete,mo3);
                d3:=sqrt(2*a3*a3-4*e3*e3-4*a3*e3);
                h3:=mo3/((a3+2*e3)*d3*mv3);
                vaA(1,wherey+2);
                ecrivideo('La largeur d3 est : ',1,wherey,false);
                ecrivaleur(d3,23,wherey,true);
                vaA(1,wherey+2);
                ecrivideo('L''épaisseur h3 est : ',1,wherey+2,false);
                ecrivaleur(h3,23,wherey,true); end;
            2: begin entete:='Entrer le débordement "e3" :'; vaA(1,wherey+2);
                liredefaultint(entete,-0.5*a3,-2*a3*(1-sqrt(3)),e3);
                entete:='Entrer l''épaisseur "h3" :'; vaA(1,wherey+2);
                liredefaultSpos(entete,h3);
                d3:=sqrt(2*a3*a3-4*e3*e3-4*a3*e3);
                mo3:=h3/((a3+2*e3)*d3*mv3);
                vaA(1,wherey+2);
                ecrivideo('La largeur d3 est : ',1,wherey,false);
                ecrivaleur(d3,23,wherey,true);
                vaA(1,wherey+2);
                ecrivideo('La masse mo3 est : ',1,wherey,false);
                ecrivaleur(mo3,23,wherey,true); end;
            3: begin entete:='Entrer la largeur "d3" :'; vaA(1,wherey+2);
                liredefaultint(entete,-rinf,a3*sqrt(3),d3);
                entete:='Entrer la masse "mo3" :'; vaA(1,wherey+2);
                liredefaultSpos(entete,mo3);
                e3:=0.5*d3*( sqrt(3*sqr(a3/d3)-1)-(a3/d3) );
                h3:=mo3/((a3+2*e3)*d3*mv3);
                vaA(1,wherey+2);
                ecrivideo('Le débordement e3 est : ',1,wherey,false);
                ecrivaleur(e3,26,wherey,true);
                vaA(1,wherey+2);
        end;
    end;
until wherey=26;
end;

```



Listing programme ADM: chapitres 3 & 4 et programme principal

```

    ecrivideo('L''épaisseur h3 est : ',1,wherey,false);
    ecrivaleur(h3,26,wherey,true); end;
4: begin entete:='Entrer la largeur "d3" :'; vaA(1,wherey+2);
    liredefautint(entete,-rinf,a3*sqrt(3),d3);
    entete:='Entrer l''épaisseur "h3" :'; vaA(1,wherey+2);
    liredefautSpos(entete,h3);
    e3:=0.5*d3*( sqrt(3*sqr(a3/d3)-1)-(a3/d3) );
    mo3:=h3*((a3+2*e3)*d3*mv3);
    vaA(1,wherey+2);
    ecrivideo('Le débordement e3 est : ',1,wherey,false);
    ecrivaleur(e3,26,wherey,true);
    vaA(1,wherey+2);
    ecrivideo('La masse mo3 est : ',1,wherey,false);
    ecrivaleur(mo3,26,wherey,true); end;
5: begin entete:='Entrer la masse "mo3":'; vaA(1,wherey+2);
    liredefautspos(entete,mo3);
    entete:='Entrer l''épaisseur "h3" :'; vaA(1,wherey+2);
    liredefautSpos(entete,h3);
    dd:=sqrt(9*a3*a3*a3-4*sqr(mo3/(h3*mv3)));
    x1:=0.5*(3*a3*a3-dd);
    x2:=0.5*(3*a3*a3+dd);
    str(x1:8:3,tabplfb[1,1]);
    str(x2:8:3,tabplfb[2,1]);
    entete:='Choisissez "e3" entre les deux valeurs : ';
    plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c2);
    if c2=1 then e3:=x1 else e3:=x2;
    d3:=sqrt(2*a3*a3-4*e3*e3-4*a3*e3);
    vaA(1,wherey+2);
    ecrivideo('Le débordement e3 est : ',1,wherey,false);
    ecrivaleur(e3,26,wherey,true);
    vaA(1,wherey+2);
    ecrivideo('La largeur d3 est : ',1,wherey,false);
    ecrivaleur(d3,26,wherey,true); end;
end;
if ll=2 then
case c1 of
1: begin entete:='Entrer le débordement "e3" :'; vaA(1,wherey+2);
    liredefautSpos(entete,e3);
    entete:='Entrer la masse "mo3":'; vaA(1,wherey+2);
    liredefautSpos(entete,mo3);
    Asurd(e3/d3,d3); d3:=a3/d3;
    h3:=mo3/((a3-2*d3)*d3+2*pi*d3*d3+2*e3*d3) *mv3;
    vaA(1,wherey+2);
    ecrivideo('La largeur d3 est : ',1,wherey,false);
    ecrivaleur(d3,23,wherey,true);
    vaA(1,wherey+2);
    ecrivideo('L''épaisseur h3 est : ',1,wherey,false);
    ecrivaleur(h3,23,wherey,true); end;
2: begin entete:='Entrer le débordement "e3" :'; vaA(1,wherey+2);
    liredefautspos(entete,e3);
    entete:='Entrer l''épaisseur "h3" :'; vaA(1,wherey+2);
    liredefautSpos(entete,h3);
    Asurd(e3/d3,d3); d3:=a3/d3;
    mo3:=((a3-2*d3)*d3+2*pi*d3*d3+2*e3*d3)*mv3*h3;
    vaA(1,wherey+2);
    ecrivideo('La largeur d3 est : ',1,wherey,false);
    ecrivaleur(d3,23,wherey,true);
    vaA(1,wherey+2);
    ecrivideo('La masse mo3 est : ',1,wherey,false);
    ecrivaleur(mo3,23,wherey,true); end;
3: begin entete:='Entrer la largeur "d3" :'; vaA(1,wherey+2);
    liredefautspos(entete,d3);
    entete:='Entrer la masse "mo3":'; vaA(1,wherey+2);
    liredefautSpos(entete,mo3);
    Esurd(a3/d3,e3); e3:=e3*d3;
    h3:=mo3/((a3-2*d3)*d3+2*pi*d3*d3+2*e3*d3) *mv3;
    vaA(1,wherey+2);

```

Listing programme ADM: chapitres 3 & 4 et programme principal

```

    ecrivideo('Le débordement e3 est : ',1,wherey,false);
    ecrivaleur(e3,26,wherey,true);
    vaA(1,wherey+2);
    ecrivideo('L''épaisseur h3 est : ',1,wherey,false);
    ecrivaleur(h3,26,wherey,true); end;
4: begin entete:='Entrer la largeur "d3" :'; vaA(1,wherey+2);
    liredefautint(entete,-rinf,a3*sqrt(3),d3);
    entete:='Entrer l''épaisseur "h3" :'; vaA(1,wherey+2);
    liredefautSpos(entete,h3);
    EsurD(a3/d3,e3); e3:=e3*d3;
    mo3:= ((a3-2*d3)*d3+2*pi*d3*d3+2*e3*d3) *mv3*h3;
    vaA(1,wherey+2);
    ecrivideo('Le débordement e3 est : ',1,wherey,false);
    ecrivaleur(e3,26,wherey,true);
    vaA(1,wherey+2);
    ecrivideo('La masse mo3 est : ',1,wherey,false);
    ecrivaleur(mo3,26,wherey,true); end;
5: begin calculE1;
    vaA(1,wherey+2);
    ecrivideo('Le débordement e3 est : ',1,wherey,false);
    ecrivaleur(e3,26,wherey,true);
    vaA(1,wherey+2);
    ecrivideo('La largeur d3 est : ',1,wherey,false);
    ecrivaleur(d3,26,wherey,true); end;
end;
if ll=3 then calculme3;
ecrivideo('Tapez 0 si valeurs ok sinon N ',1,23,true);
repeat car5:=upcase(readkey);
    if (car5<>'0')and(car5<>'N') then beep
until (car5='0')or(car5='N');
if car5='N' then ecrivideo(' ',1,23,false);
until car5='0';
vaA(1,4);
end;
Procedure reconfigCP3(mr3,q3:real; var r3,rr3,qq3,re3,rre3,qe3,qqe3,mre3,mrre3:real;var reconfig:boolean);
begin
    if reconfig then begin
        me3:=0; re3:=0; rr3:=a3; qe3:=0; qqe3:=pi; mre3:=0; mrre3:=0;
        tabplfb[1,1]:='m3';
        tabplfb[2,1]:=chr3+'';
        tabplfb[3,1]:=chr3+'';
        tabplfb[4,1]:=chr3+'';
        entete:='Lequel des parametres voulez-vous fixer ? :';
        if (cos(q3)<=-1+1e-10)and(cos(q3)>=-1-1e-10) then begin
            qq3:=pi;
            plan(entete,1,wherey+2,1,3,length(entete)+2,wherey+2,tabplfb,l,c);
            if c=1 then begin entete:='Entrez m3';
                vaA(1,wherey+2);
                liredefautSPOS(entete,m3);
                r3:=mr3/m3;
                rr3:=r3+a3 end;
            if c=2 then begin entete:='Entrez '+chr3;
                vaA(1,wherey+2);
                liredefautSPOS(entete,r3);
                rr3:=r3+a3;
                m3:=mr3/r3 end;
            if c=3 then begin entete:='Entrez '+chr3+' > à';
                rr3:=a3;
                liredefautint(entete,a3,rsup,rr3);
                r3:=rr3-a3;
                m3:=mr3/r3; end;
        end;
        if (cos(q3)<=1+1e-10)and(cos(q3)>=1-1e-10) then begin
            plan(entete,1,wherey+2,1,3,length(entete)+2,wherey+2,tabplfb,l,c);
            if c=1 then begin entete:='Entrez m3';
                vaA(1,wherey+2);
                liredefautSPOS(entete,m3);
    
```

Listing programme ADM: chapitres 3 & 4 et programme principal

```

        r3:=mr3/m3;
        if r3<a3 then begin qq3:=pi; rr3:=a3-r3 end
        else begin qq3:=0; rr3:=r3-a3 end;end;
if c=2 then begin entete:='Entrez '+chr3;
vaA(1,wherey+2);
liredefautSPOS(entete,r3);
if r3<a3 then begin qq3:=pi; rr3:=a3-r3 end
else begin qq3:=0; rr3:=r3-a3 end;
m3:=mr3/r3 end;
if c=3 then begin entete:='Choisissez '+chrqq3+' entre: ';
str(0:8,tabplfb[1,1]);
str(pi:8:3,tabplfb[2,1]);
plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c1);
if c1=1 then qq3:=0 else qq3:=pi;
if c1=2 then begin entete:='Entrez '+chr3+' < à ';
rr3:=a3;
vaA(1,wherey+2);
liredefautint(entete,0,a3,rr3);
r3:=a3-rr3 end
else begin entete:='Entrez '+chr3;
vaA(1,wherey+2);
liredefautSPOS(entete,rr3);
r3:=a3+rr3 end;
m3:=mr3/r3; end;
end;
if ((cos(q3)>-1+1e-10)or(cos(q3)<-1-1e-10) )and
((cos(q3)>1+1e-10)or(cos(q3)<1-1e-10) ) then begin
plan(entete,1,wherey+2,1,4,length(entete)+2,wherey+2,tabplfb,l,c);
if c=1 then begin entete:='Entrer m3';
vaA(1,wherey+2);
liredefautSPOS(entete,m3);
r3:=mr3/m3;
qq3:=arctgt(r3*sin(q3),r3*cos(q3)-a3);
rr3:=r3*sin(q3)/sin(qq3) end;
if c=2 then begin entete:='Entrez '+chr3;
vaA(1,wherey+2);
liredefautSPOS(entete,r3);
m3:=mr3/r3;
qq3:=arctgt(r3*sin(q3),r3*cos(q3)-a3);
rr3:=r3*sin(q3)/sin(qq3) end;
if c=3 then begin entete:='Entrez '+chr3+' >= à ';
rr3:=sqrt(a3*a3-sqr(a3*cos(q3)));
m3:=rr3;
vaA(1,wherey+2);
liredefautint(entete,m3-rinf,rsup,rr3);
m3:=sqrt(sqr(a3*cos(q3))-a3*a3+rr3*rr3);
r3:=a3*cos(q3)+sqrt(m3);
if rr3<a3 then begin
qq3:=a3*cos(q3)-sqrt(m3);
entete:='r3 peut prendre deux valeurs;choisissez: ';
str(r3:8:3,tabplfb[1,1]);
str(qq3:8:3,tabplfb[2,1]);
plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c1);
if c1=2 then r3:=qq3 end;
qq3:=arctgt(r3*sin(q3),r3*cos(q3)-a3);
m3:=mr3/r3; end;
if c=4 then begin entete:='Entrez '+chrqq3;
if (0<sin(q3))and(sin(q3)<1) then begin
entete:=entete+' 0<'+chrqq3+'<pi ';
vaA(1,wherey+2);
liredefautint(entete,0-rinf,pi+rsup,qq3); end;
if (-1<sin(q3))and(sin(q3)<0) then begin
entete:=entete+' pi<'+chrqq3+'<2pi ';
vaA(1,wherey+2);
liredefautint(entete,pi-rinf,2*pi+rsup,qq3);end;
rr3:=a3*sin(q3)/(sin(qq3)*cos(q3)-cos(qq3)*sin(q3));
r3:=rr3*sin(qq3)/sin(q3);

```

```

                m3:=mr3/r3;
            end;                end;
mrr3:=m3*rr3;                end
else begin
    mre3:=sqrt(mr3*mr3+mi3*ri3*mi3*ri3-2*mr3*mi3*ri3*cos(q3-qi3));
    qe3:=arctgt(mr3*sin(q3)-mi3*ri3*sin(qi3),mr3*cos(q3)-mi3*ri3*cos(qi3));
    tabplfb[1,1]:='m*3';
    tabplfb[2,1]:='chre3';
    entete:='Choisissez le parametre du contrepoids fixe à 3 ? : ';
    plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
    if c=1 then begin vaA(1,wherey+2);
        liredefautSPOS('Entrez m*3',me3);
        re3:=mre3/me3;
        qqe3:=arctgt(re3*sin(qe3),re3*cos(qe3)-a3);
        if sin(qqe3)<>0 then rre3:=re3*sin(qe3)/sin(qqe3) end
        else rre3:=re3*cos(qe3)/cos(qqe3);
    if c=2 then begin vaA(1,wherey+2);
        liredefautSPOS('Entrez '+chre3,re3);
        me3:=mre3/re3;
        qqe3:=arctgt(re3*sin(qe3),re3*cos(qe3)-a3);
        if sin(qqe3)<>0 then rre3:=re3*sin(qe3)/sin(qqe3)
        else rre3:=(re3*cos(qe3)-a3)/cos(qqe3);
        end;
    mrrre3:=me3*rre3;
    m3:=mie3+me3; r3:=mr3/m3;
    mrr3:=sqrt(mi3*ri3*mi3*ri3+me3*rre3*me3*rre3+2*mi3*ri3*me3*rre3*cos(qqi3-qqe3));
    qq3:=arctgt(mi3*ri3*sin(qqi3)-me3*rre3*sin(qqe3),mi3*ri3*cos(qqi3)-me3*rre3*cos(qqe3));
    rr3:=mrr3/m3;
end; end;

Procedure RconfigCP4(reconfig:boolean);
Begin
    if reconfig then begin
        me4:=0; re4:=0; qe4:=0; mre4:=0;
        tabplfb[1,1]:='m*4';
        tabplfb[2,1]:='r4';
        entete:='Fixer un parametre de la membrure 4 ? : ';
        plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
        if c=1 then begin vaA(1,wherey+2);
            liredefautSPOS('Entrez m4 ',m4);
            r4:=mr4/m4 end;
        if c=2 then begin vaA(1,wherey+2);
            liredefautSPOS('Entrez r4 ',r4);
            m4:=mr4/r4 end;
        end
    else begin
        mre4:=sqrt(mr4*mr4+mo4*ro4*mo4*ro4-2*mr4*mo4*ro4*cos(q4-qo4));
        qe4:=arctgt(mr4*sin(q4)-mo4*ro4*sin(qo4),mr4*cos(q4)-mo4*ro4*cos(qo4));
        tabplfb[1,1]:='m*4';
        tabplfb[2,1]:='r*4';
        entete:='Choisissez le parametre du contrepoids fixe à 4 ? : ';
        plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
        if c=1 then begin vaA(1,wherey+2);
            liredefautSPOS('Entrez m*4',me4);
            re4:=mre4/me4 end;
        if c=2 then begin vaA(1,wherey+2);
            liredefautSPOS('Entrez r*4',re4);
            me4:=mre4/re4 end;
        m4:=mo4+me4;
        r4:=mr4/m4;
    end;
end;

Procedure RconfigCP2(reconfig:boolean);
Begin
    if reconfig then begin
        me2:=0; re2:=0; qe2:=0; mre2:=0;
        tabplfb[1,1]:='m2';
    end;
end;

```

```

tabplfb[2,1]:='r2      ';
entete:='Fixer un parametre de la membrure 2 : ';
plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
if c=1 then begin
  vaA(1,wherey+2);
  liredefautSPOS('Entrez m2',m2);
  r2:=mr2/m2 end;
if c=2 then begin
  vaA(1,wherey+2);
  liredefautSPOS('Entrez r2',r2);
  m2:=mr2/r2 end; end
else begin
  mre2:=sqrt(mr2*mr2+mo2*ro2*mo2*ro2-2*mr2*mo2*ro2*cos(q2-qo2));
  qe2:=arctgt(mr2*sin(q2)-mo2*ro2*sin(qo2),mr2*cos(q2)-mo2*ro2*cos(qo2));
  tabplfb[1,1]:='m*2      ';
  tabplfb[2,1]:='r*2      ';
  entete:='Fixer un parametre du contrepoids 2 : ';
  plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
  if c=1 then begin vaA(1,wherey+2);
    liredefautSPOS('Entrez m*2',me2);
    re2:=mre2/me2 end;
  if c=2 then begin vaA(1,wherey+2);
    liredefautSPOS('Entrez r*2',re2);
    me2:=mre2/re2 end;
  m2:=mo2+me2;
  r2:=mr2/m2;
end;
end;

Procedure forcebalance;
begin
  qqo3:=arctgt(ro3*sin(qo3),ro3*cos(qo3)-a3);
  if sin(qqo3)<>0 then rro3:=ro3*sin(qo3)/sin(qqo3)
    else rro3:=(ro3*cos(qo3)-a3)/cos(qqo3);
  tabplfb[1,1]:='Reconfiguration  ';
  tabplfb[2,1]:='Contrepoids      ';
  if memb1=2 then begin
    entete:='Pour la membrure 2, choisissez!';
    plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
    if c=1 then reconfig1:=true else reconfig1:=false; end
  else if menuz=2 then reconfig1:=true
  else begin
    entete:='Pour la membrure 3, choisissez!';
    plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
    if c=1 then reconfig1:=true else reconfig1:=false; end;
  if memb2=4 then if menuz=2 then reconfig2:=true
  else begin
    entete:='Pour la membrure 4, choisissez!';
    plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
    if c=1 then reconfig2:=true else reconfig2:=false; end
  else if menuz=2 then reconfig2:=true
  else begin
    entete:='Pour la membrure 3, choisissez!';
    plan(entete,1,wherey+2,1,2,length(entete)+2,wherey+2,tabplfb,l,c);
    if c=1 then reconfig2:=true else reconfig2:=false; end;
  if (memb1=2)and(memb2=3) then
  begin
    chr3:='r3  '; chrr3:='r''3'; chq3:='b3  '; chqq3:='b''3'; chre3:='r*3  ';
    q4:=qo4;
    m4:=mo4;
    r4:=ro4; mr4:=m4*r4;
    if menuz=1 then begin mr3:=a3*m4*r4/a4;
      q3:=q4-pi;
      end
    else begin mr3:=m4*a3;
      q3:=0;
      end;
    reconfigCP3(mr3,q3,r3,rr3,qq3,re3,rre3,qe3,qqe3,mre3,mrre3,reconfig2);

```

```

mr2:=m3*rr3*a2/a3;
q2:=qq3;
RconfigCP2(reconfig1);
end;
if (memb1=3)and(memb2=4) then
begin
chr3:='r3 '; chr3:='r''3'; chq3:='b3 '; chq3:='b''3'; chre3:='r''*3';
q2:=qo2;
m2:=mo2;
r2:=ro2; mr2:=m2*r2;
mrr3:=a3*m2*r2/a2;
qq3:=q2;
reconfigCP3(mrr3,qq3,rr3,r3,q3,rre3,re3,qqe3,qe3,mrre3,mre3,reconfig1);
if menuz=1 then begin mr4:=m3*r3*a4/a3;
q4:=q3+pi;
RconfigCP4(reconfig2);
end
else begin m4:=m3*r3/a3; q4:=0; me4:=0; re4:=0 end;
end;
if (memb1=2)and(memb2=4) then
begin
if not allbalance then begin
q3:=qo3; qq3:=qqo3;
m3:=mo3; rr3:=rro3;
r3:=ro3; mr3:=m3*r3; mrr3:=m3*rr3; end
else begin
q3:=qi3; qq3:=qqi3;
m3:=mi3; rr3:=rri3;
r3:=ri3; mr3:=m3*r3; mrr3:=m3*rr3; end;
mr2:=a2*m3*rr3/a3;
q2:=qq3;
RconfigCP2(reconfig1);
if menuz=1 then begin mr4:=m3*r3*a4/a3;
q4:=q3+pi;
RconfigCP4(reconfig2);
end
else begin m4:=m3*r3/a3;
q4:=0
end;
end;
clrscr;
writeln('paramètres initiaux'); writeln;
writeln('Paramètres      memb1      memb2      memb3      memb4 ');
writeln(' a      ',a1:10:3,a2:10:3,a3:10:3,e4:10:3);
writeln(' r°      ',r1:10:3,r2:10:3,r3:10:3,r4:10:3);
writeln(' p°      ',qo1:10:3,qo2:10:3,qo3:10:3,qo4:10:3);
writeln(' r'''      ',r1:10:3,r2:10:3,rro3:10:3,rri:10:3);
writeln(' p'''      ',p1:10:3,p2:10:3,ppo3:10:3,ppi:10:3);
writeln(' m°      ',m1:10:3,m2:10:3,m3:10:3,m4:10:3);
repeat until keypressed; clrscr;
writeln;
writeln('Paramètres pour le balancement des forces'); writeln;
writeln('Paramètres      memb2      memb3      memb4 ');
writeln(' p      ',q2:10:3,q3:10:3,q4:10:3);
writeln(' mr      ',mr2:10:3,mr3:10:3,mr4:10:3);
writeln(' r      ',r2:10:3,r3:10:3,r4:10:3);
writeln(' m      ',m2:10:3,m3:10:3,m4:10:3);
writeln(' p''      ',p2:10:3,p3:10:3,p4:10:3);
writeln(' mr''      ',mrr3:10:3,mri:10:3);
writeln(' r''      ',r2:10:3,rri:10:3);
writeln(' p*      ',qe2:10:3,qe3:10:3,qe4:10:3);
writeln(' m*r*      ',mre2:10:3,mre3:10:3,mre4:10:3);
writeln(' m*      ',me2:10:3,me3:10:3,me4:10:3);
writeln(' r*      ',re2:10:3,re3:10:3,re4:10:3);
writeln(' r''*      ',rre3:10:3,rri:10:3);
repeat until keypressed;
end;

```

Listing programme ADM: chapitres 3 & 4 et programme principal

```

BEGIN
initFB;
FBinitMes;
REPEAT
if menuz=2 then allbalance:=true else allbalance:=false;
IF NOT allbalance then begin
repeat effecr(4,21);
menu(page,1,1);
retournevalFB;
if (menuz=1)and((a1<=0)or(a2<=0)or(a3<=0)or(a4<=0)) then begin beep;
ecrivideo('a1,a2,a3 et a4 doivent être > 0',1,2,true); end;
until (menuz=2) or ((menuz=1)and((a1>0)and(a2>0)and(a3>0)and(a4>0)));
tabplfb[1,1]:='Membreure 2 ';
tabplfb[2,1]:='Membreure 3 ';
tabplfb[3,1]:='Membreure 4 ';
entete:='Donnez la première membreure dont les caractéristiques seront modifiées. ';
plan(entete,1,5,1,3,1,7,tabplfb,l,memb1);
repeat
entete:='Donnez la deuxième membreure dont les caractéristiques seront modifiées. ';
plan(entete,1,10,1,3,1,12,tabplfb,l,memb2);
if memb2=memb1 then beep;
until memb2<>memb1;
memb1:=memb1+1;
memb2:=memb2+1;
if memb1>memb2 then begin l:=memb1; memb1:=memb2; memb2:=l end;
if menuz=2 then inline3;
forcebalance; end
ELSE begin
repeat effecr(5,24);
menu(page,1,2);
retournevalFB;
if (a1<=0)or(a2<=0)or(a3<=0)or(a4<=0) then begin beep;
ecrivideo('a1,a2,a3 et a4 doivent être > 0',1,2,true); end;
until (a1>0)and(a2>0)and(a3>0)and(a4>0);
inline3;
pendul;
memb1:=2;
memb2:=4;
forcebalance;
delay(2000); clrscr;
vaA(1,wherey+2);
liredefautSpos('Donner le rayon de giration du C-P fixé à "2" ',ke2);
vaA(1,wherey+2);
liredefautSpos('Donner le rayon de giration du C-P fixé à "4" ',ke4);
rx:=(m02*r02*cos(qo2)+me2*re2*cos(qe2))/(m02+me2);
ry:=(m02*r02*sin(qo2)+me2*re2*sin(qe2))/(m02+me2);
ggo:=sqr(rx-ro2*cos(qo2))+sqr(ry-ro2*sin(qo2));
gge:=sqr(rx-re2*cos(qe2))+sqr(ry-re2*sin(qe2));
k2:=( (Igo2+m02*ggo)+me2*(ke2*ke2+gge) )/(me2+m02);
rx:=(m04*r04*cos(qo4)+me4*re4*cos(qe4))/(m04+me4);
ry:=(m04*r04*sin(qo4)+me4*re4*sin(qe4))/(m04+me4);
ggo:=sqr(rx-ro4*cos(qo4))+sqr(ry-ro4*sin(qo4));
gge:=sqr(rx-re4*cos(qe4))+sqr(ry-re4*sin(qe4));
k4:=( (Igo4+m04*ggo)+me4*(ke4*ke4+gge) )/(me4+m04);
Iee2:=m2*(k2*k2+r2*r2+a2*r2);
Iee4:=m4*(k4*k4+r4*r4+a4*r4);
writeln;
writeln('Paramètres pour le balancement des moment');writeln;
writeln('Paramètres      memb2      memb3      memb4 ');
writeln('I cp-force      ',m2*ke2*ke2:10:3,'-':10,m4*ke4*ke4:10:3);
writeln('k cp-force      ',ke2:10:3,'-':10,ke4:10:3);
writeln('I memb+cp f      ',m2*k2*k2:10:3,m3*k3*k3:10:3,m4*k4*k4:10:3);
writeln('k memb+cp f      ',k2:10:3,k3:10:3,k4:10:3);
writeln('I cp-inertie',Iee2:10:3,'-':10,Iee4:10:3);
end;
ecrivideo('Taper ESC pour sortir du module ou CR pour retourner changer les valeurs.',1,23,true);

```

```

repeat car:=readkey;
  if car=nul then car:=readkey;
until (car=esc) or (car=cr);
Until car=esc;
End;

BEGIN
repeat
reversevideo(false);
clrscr;
chaîne:='Choisir l'option désirée';
tabpl[1,1]:='Analyse cinéto-statique';
tabpl[1,2]:='Analyse de la réponse temporelle';
tabpl[1,3]:='Equilibrage des Rotors';
tabpl[1,4]:='Equilibrage des forces et des moments';
plan(chaîne,1,3,4,1,1,5,tabpl,menux,i);
clrscr; reversevideo(false); clrscr;
chaîne:='Choisir la sous-option désirée';
case menux of
  1,2 : begin tabpl[1,1]:='Mécanismes de longueurs finies';
           tabpl[1,2]:='Mécanismes manivelle-coulisseau';
           plan(chaîne,1,3,2,1,1,5,tabpl,menuy,i);
         end;
  3: begin tabpl[1,1]:='Equilibrage statique';
           tabpl[1,2]:='Equilibrage dynamique';
           plan(chaîne,1,3,2,1,1,5,tabpl,menuy,i);
         end;
  4: begin tabpl[1,1]:='Equilibrage des forces uniquement';
           tabpl[1,2]:='Equilibrage complet (forces et moments)';
           plan(chaîne,1,3,2,1,1,5,tabpl,menuy,i);
           if menuy=1 then
             begin reversevideo(false); clrscr;
                   chaîne:='Choisir le type de mécanisme à équilibrer';
                   tabpl[1,1]:='Mécanismes de longueurs finies';
                   tabpl[1,2]:='Mécanismes manivelle-coulisseau';
                   plan(chaîne,1,3,2,1,1,5,tabpl,menuz,i);
             end
           else menuz:=1;
         end;
end;
case menux of
  1: case menuy of
      1: forcesjoints4m;
      2: forcesjoints2m;
    end;
  2: case menuy of
      1: reptempof;
      2: reptempomc;
    end;
  3: rotor;
  4: balance;
end;
  car:=1;
  ecrivideo('Taper ESC pour sortir du logiciel ou CR pour retourner changer les valeurs. ',1,24,true);
  repeat car:=readkey;
    if car=nul then car:=readkey;
    if (car<>esc) and (car<>cr) then beep
  until (car=esc) or (car=cr);
  reversevideo(false); clrscr;
Until car=esc;
END.

```