

2019

N° d'ordre 05/87-M/PH

N° SIRC

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE

HOUARI BOUMEDIENE (U.S.T.H.B) ALGER

3638

INSTITUT DE PHYSIQUE

THESE

Présentée à l'U.S.T.H.B pour l'obtention du grade de

M A G I S T E R

Spécialité : Electronique des systèmes

par

Abderrezak HADJ LARBI

SIMULATION NUMERIQUE DE PROCESSUS PHYSIQUE ET METHODE MONO ET MULTI PAS : APPLICATION A UN ALGORITHME DE COMMANDE ADAPTATIVE D'UN ROBOT

Soutenue publiquement le 14 Juin 1987

devant le jury composé de :

DAHEL	Abdelhamid	PROFESSEUR U.S.T.H.B	PRESIDENT
TOUMI	Redouane	M.de CONFERENCE U.S.T.H.B	RAPPORTEUR
SANSAL	Boualem	PROFESSEUR U.S.T.H.B	EXAMINATEUR
BENDALI	Abderahmane	M.de CONFERENCE U.S.T.H.B	"
KACIMI	Messaoud	CHARGE de COURS U.S.T.H.B	"

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE

HOUARI BOUMEDIENE (U.S.T.H.B) ALGER

INSTITUT DE PHYSIQUE

THESE

Présentée à l'U.S.T.H.B pour l'obtention du grade de

M A G I S T E R

Spécialité : Electronique des systèmes

par

HAD / 1430
1

Abderrezak HADJ LARBI

SIMULATION NUMERIQUE DE PROCESSUS PHYSIQUE ET METHODE MONO
ET MULTI PAS : APPLICATION A UN ALGORITHME DE COMMANDE ADAPTATIVE
D'UN ROBOT

Soutenue publiquement le 14 Juin 1987

devant le jury composé de :

DAHEL	Abdelhamid	PROFESSEUR U.S.T.H.B	PRESIDENT
TOUMI	Redouane	M.de CONFERENCE U.S.T.H.B	RAPPORTEUR
SANSAL	Boualem	PROFESSEUR U.S.T.H.B	EXAMINATEUR
BENDALI	Abderahmane	M.de CONFERENCE U.S.T.H.B	"
KACIMI	Messaoud	CHARGE de COURS U.S.T.H.B	"

REMERCIEMENTS

Je tiens tout d'abord à remercier le professeur SANSAL pour m'avoir accueilli dans son laboratoire, et de m'avoir permis de concrétiser ce travail.

Je remercie vivement le professeur A. DAHEL de m'avoir fait l'honneur d'assumer la présidence du jury.

Que Monsieur le professeur R. TOUMI trouve ici l'expression de ma profonde gratitude pour ses conseils, et l'intérêt qu'il a toujours porté à ce travail.

Mes profonds remerciements et ma gratitude vont aussi à Monsieur le professeur A. MIGNOT de l'université de RENNES 1 (FRANCE). pour ses enseignements, et ses précieux conseils en analyse numérique.

J'adresse mes vifs remerciements à Monsieur le professeur BENDALI pour avoir accepté de faire partie de ce jury.

Que Monsieur KACIMI MESSAOUD trouve ici l'expression de ma reconnaissance pour avoir accepté d'être membre du jury.

Enfin je tiens à remercier tous ceux qui de près ou de loin ont contribué à l'aboutissement de ce travail, notamment les gens du centre de calcul, ainsi que GUETTACHE et ABDERRAHMANE pour la qualité du tirage.

TABLE DES MATIERES

<u>INTRODUCTION GENERALE</u>	<u>Page</u>
<u>PARTIE A:</u>	4
<u>INTRODUCTION</u>	Idem
<u>CHAPITRE I: MISE EN OEUVRE DES METHODES DE RUNGE-KUTTA A PAS VARIABLES:</u>	6
I.1 <u>Généralités</u>	Idem
I.1.1 Consistance du schéma	7
I.1.2 Stabilité du schéma	Idem
I.1.3 Convergence du schéma	8
I.2 <u>Schéma général des méthodes de RUNGE-KUTTA à q pas intermédiaires</u>	Idem
I.3 <u>Comportement asymptotique de l'erreur</u>	10
I.3.1 Expression de l'erreur	11
I.3.2 Choix optimal du pas h_n	Idem
I.4 <u>Contrôle du pas</u>	13
I.4.1 Estimation de l'erreur	Idem
I.4.2 Contrôle du pas	14
I.5 <u>Schéma final de RUNGE-KUTTA adopté</u>	15
I.5.1 Calcul des coefficients de la méthode	Idem
I.5.2 Estimation de $\tilde{\epsilon}_n$	16

I.6	<u>Différents types de contrôle de l'erreur</u>	16
I.6.1	Premier type de contrôle d'erreur	17
I.6.2	Deuxième type de contrôle d'erreur	18
I.6.3	Choix $\sigma_0(t_n)$	Ide
I.6.4	Troisième type de contrôle d'erreur	19
I.7	<u>ORGANIGRAMME</u>	21

<u>CHAPITRE II:</u>	<u>MISE EN OEUVRE D'UN SCHEMA D'ADAMS A PAS ET ORDRE VARIABLES, EXPRIMES A L'AIDE DES DIFFERENCES DIVISEES</u>	22
II.1	<u>Méthode d'ADAMS d'ordre (r+1) à pas variables</u>	Ide
II.2	<u>Méthode d'ADAMS employée sous forme Prédicteur-Correcteur</u>	24
II.2.1	Définition du schéma	25
II.2.2	Stabilité et ordre du schéma	27
II.3	<u>Formulation du schéma Prédicteur-correcteur à l'aide des différences divisées</u>	28
II.3.1	<u>Expression du schéma et choix des variables</u>	Ide
II.3.1.1	Schéma prédicteur d'ordre K	29
II.3.1.2	Schéma correcteur d'ordre K	31
II.3.1.3	Expression du schéma P & EC & E et résumé des opérations à une étape	33
II.3.2	Calcul des coefficients du schéma	34

	P	
II.4	<u>Contrôle du pas et de l'ordre</u>	
II.4.1	Expression des erreurs de consistance	
II.4.2	Algorithme en phase normale de fonctionnement	
II.4.3	Algorithme en phase de démarrage, choix du 1er pas	
<u>CHAPITRE III:</u>	<u>COMPORTEMENT EFFECTIF DES METHODES:</u>	
a)	<u>Comportement de la méthode P _k EC _k E</u>	
b)	<u>Comportement de la méthode de RUNGE-KUTTA à pas variable comparée avec la méthode à pas constant</u>	
<u>PARTIE B:</u>	<u>SIMULATION NUMERIQUE DU COMPORTEMENT DYNAMIQUE D'UN MANIPULATEUR RIGIDE ET ALGORITHME DE COMMANDE ADAPTATIVE</u>	
<u>INTRODUCTION:</u>		
<u>CHAPITRE I:</u>	<u>COMMANDE DYNAMIQUE DE MANIPULATEUR</u>	
I.	<u>Obtention du modèle dynamique d'un robot manipulateur</u>	
1)	Forme générale du modèle dynamique	
2)	Algorithme	
II.	<u>Commande adaptative</u>	
III.	<u>Schéma MRAC continu</u>	
IV.	Commande échantillonnée	

<u>CHAPITRE II: SIMULATION NUMERIQUE DU COMPORTEMENT DYNAMIQUE D'UN MANIPULATEUR RIGIDE ET COMMANDE ADAPTATIVE</u>	68
I. <u>Synoptique du schéma MRAC continu</u>	69
1. Commande en consigne	Id
2. Stabilité du schéma MRAC	71
3. Algorithme d'adaptation	Id
II. <u>Résultats de simulation</u>	72
III. <u>Schéma de commande avec adaptation parfaite</u>	76
III.1 Résultats de simulation	Id
III.2 CONCLUSION	77
<u>CONCLUSION GENERALE:</u>	78

INTRODUCTION GENERALE:

Les simulations numériques de processus physiques continus constituent un moyen très efficace pour la connaissance de l'évolution de certains phénomènes physiques très complexes, tels que le traitement d'image; le domaine nucléaire; l'aérodynamique et les systèmes automatiques.

Devant un tel problème, en l'occurrence la simulation d'un phénomène (processus) physique, trois points doivent constituer le centre de nos préoccupations, il s'agit: de la contrainte de temps réel, de la précision du calcul et enfin, le modèle du processus physique à simuler lui-même.

En ce qui concerne le temps réel, il s'agit en fait de se doter d'un ordinateur assez puissant en vitesse de calcul, et en taille mémoire, capable de suivre l'évolution de notre processus physique dans la mesure où ce dernier évolue rapidement.

Dans le cas contraire, une différence notable apparaîtra entre le processus physique et le modèle; le développement de la technologie des micro-processus a permis d'associer ces derniers, obtenant ainsi des architectures de calculateurs dont les performances répondent aux exigences de la contrainte du temps réel.

Le modèle mathématique étant régi par un système d'équations différentielles qu'il s'agit de résoudre; pour cela, plusieurs méthodes d'intégration numériques ont vu le jour, suivant la nature du problème à intégrer (équations différentielles ordinaires, raides, implicites différentielles... etc.). Partant de là, suivant la nature du problème à résoudre, il s'agit de choisir une méthode qui lui soit adéquate.

Les méthodes qui permettent d'obtenir des solutions avec une grande précision, seront privilégiées afin d'atteindre l'objectif d'une simulation qui permet une bonne analyse du processus.

Par ailleurs, le modèle étant stable (les systèmes à modèles instables, présentant peu d'intérêt), la stabilité de méthodes d'intégration est très importante, elle permet à la méthode de suivre la solution réelle du système même en présence de perturbations externes, pourvu que celles-ci soient de valeurs bornées au cours du temps.

Enfin, au cours de la simulation, on peut toujours (ce qui est d'ailleurs le but d'une simulation), étudier l'effet de la possible modification de certains paramètres intervenant dans le modèle. Cela permet une meilleure connaissance du processus réel. Le problème de la conception d'une architecture pour s'affranchir de la contrainte de temps réel a été étudié par de nombreux auteurs entre-autres (G.B.). Dans notre travail, nous nous sommes attelés à contribuer à la résolution de deux derniers problèmes (précision du calcul et modification de paramètres). Lors d'une simulation d'une commande adaptative appliquée à un robot manipulateur rigide, dont le modèle mathématique est régi par un système d'équations différentielles ordinaires de la forme:

$$\left\{ \begin{array}{l} \frac{d}{dt} Y(t) = F(t, y(t)) \\ Y(t_0) = Y_0 \text{ donnée dans } R^m \end{array} \right.$$

La solution de ce système traduit en fait le comportement dynamique de notre manipulateur rigide sous l'effet de la commande adaptative.

Ce travail à priori simple, ne peut être mené à bien que si on s'assure trois conditions principales:

- le modèle mathématique représente assez fidèlement le processus physique;
- les algorithmes de résolution doivent être très performants du point de vue précision et stabilité;
- les algorithmes doivent être compatibles avec la nature du problème mathématique à résoudre.

Notre travail sera présenté en deux grandes parties:

- Dans la première partie, on étudiera des algorithmes de résolution à base de méthodes numériques classiques permettant d'approcher la solution d'un problème différentiel bien conditionné. Notre choix s'est porté sur deux méthodes considérées comme les plus efficaces, et les plus utilisées pour ce genre de problèmes l'une multi-pas, c'est la méthode d'ADAMS BASHFORTH-MOULTON; l'autre, mono-pas, c'est la méthode de RUNGE KUTTA, classique. Ces méthodes sont écrites avec un pas et un ordre variables; ceci correspond à un double but:

- 1) Minimiser le coût de la résolution pour une précision donnée;
- 2) Adapter la méthode à la régularité de la solution.

Enfin, un chapitre est consacré à l'étude du comportement effectif de ces méthodes, en traitant certains exemples importants et en montrant l'avantage acquis par la méthode de RUNGE-KUTTA à pas variable sur celle à pas constant.

- Dans la deuxième partie, on abordera tout d'abord, la modélisation du processus physique, soit donc le calcul du modèle dynamique du manipulateur. Ensuite, on passera à la simulation du comportement dynamique de ce dernier au cours d'un transport de charge variable, sous l'effet d'une commande adaptative.

Nous montrerons qu'en fait, la commande n'est pas améliorée par l'identification en ligne, car celle-ci est très mauvaise lors d'un déplacement rapide du manipulateur. Mais qu'une commande robuste vis-à-vis des phénomènes qui apparaissent en réaction au mouvement peut être synthétisée en utilisant de grand-gains permettant de réduire les effets des erreurs de modélisation.

Pour finir, nous présenterons une conclusion relative à l'ensemble de nos travaux.

PARTIE A: ALGORITHME D'INTEGRATION NUMERIQUE:

INTRODUCTION:

Notre travail consiste en la résolution numérique du problème différentiel suivant:

$$\begin{cases} \frac{d}{dt} Y(t) = F(t, Y(t)) & t \in I = [0, T] \\ Y(0) = Y_0 \text{ donne dans } \mathbb{R}^m \end{cases} \quad (1)$$

ou: $F(t, Y(t))$ est continue, définie sur $[0, T] \times \mathbb{R}^m$ à valeur dans \mathbb{R}^m .

Nous dirons que ce problème est "BIEN CONDITIONNE" si la fonction F est uniformément lipschitzienne, c'est-à-dire, qu'il existe une constante L telle que:

$$\forall Z \in \mathbb{R}^m \quad \| F(t, Y) - F(t, Z) \| \leq L \| Y - Z \| \quad (2)$$

on sait alors que le problème (1) admet une solution unique, de même que le problème perturbé:

$$\begin{cases} \frac{d}{dt} Y_\epsilon(t) = F(t, Y_\epsilon(t)) + \epsilon(t) \\ Y_\epsilon(0) = Y_0 + \epsilon_0 \end{cases}$$

on obtient alors la majoration suivante:

$$\| Y(t) - Y_\epsilon(t) \| \leq e^{LT} \|\epsilon_0\| + \int_0^t e^{L(t-s)} \|\epsilon(s)\| ds$$

qui permet d'affirmer que Y_ϵ tend vers Y uniformément, dès que les termes de perturbation $\epsilon_0, \epsilon(\cdot)$ tendent vers zéro.

Actuellement, les méthodes numériques les plus efficaces pour approcher la solution du problème différentiel (1), sont les méthodes de RUNGE-KUTTA et les méthodes d'ADAMS.

Ces méthodes marchent bien tant que l'hypothèse (2) est vérifiée même localement [1].

Elles sont écrites avec un nombre et une taille de pas variables pour mieux adapter la méthode à la régularité de la solution, contrairement aux méthodes classiques qui utilisent des pas constants en nombre et de taille fixe.

CHAPITRE I: MISE EN OEUVRE DES METHODES DE RUNGE-KUTTA A PAS VARIABLES:

I.1 Généralités:

Pour approcher la solution du problème différentiel (1), choisissons une subdivision de $I = [0, T]$:

$$0 < t_1 < t_2 \dots \dots < t_n < t_{n+1} < \dots \dots < t_N = T$$

tel que: $h_n = t_{n+1} - t_n$ et $h = \max_{0 \leq n \leq N} h_n$

Le principe consiste, partant d'une approximation Y_n de $Y(t_n)$ prise à l'instant t_n , à calculer une approximation Y_{n+1} de $Y(t_{n+1})$ à l'instant t_{n+1} au moyen d'une relation de récurrence d'ordre 1 de la forme:

$$\begin{cases} Y_{n+1} = Y_n + h_n \cdot \phi(t_n, Y_n, h_n) \\ Y_0 = \eta \text{ donnée dans } R^m \end{cases} \quad (I.1)$$

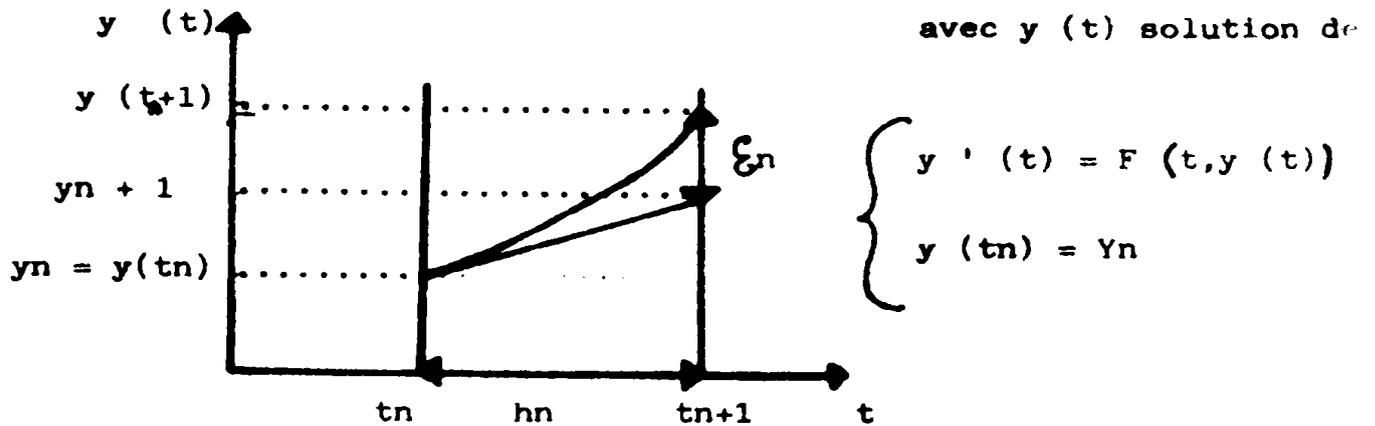
ou ϕ est une fonction continue de $[0, T] \times R^m \times [0, h^*]$ tel que $h^* > 0$.

Cependant, remplacer le système (1) par le schéma, ne va pas sans introduire un terme d'erreur appelée erreur de discrétisation. L'étude de cette dernière de par son effet sur l'approximation de la solution du problème différentiel (1), nous amène à définir les notions de:

- consistance,
 - stabilité,
 - convergence
- du schéma (I.1).

I.1.1 Consistance:

L'erreur de consistance ou de troncature est l'erreur qu'on commet en remplaçant le système (1) par le schéma (I.1)



Soit donc:

$$Y(t_{n+1}) = Y(t_n) + h_n \cdot \phi(t_n, y(t_n), h_n) + \epsilon_n \quad (I.2)$$

$$N-1 \qquad \qquad \qquad N-1$$

d'où:
$$\sum_{n=0}^{N-1} |y(t_{n+1}) - y(t_n) - h_n \cdot \phi(t_n, y(t_n), h_n)| = \sum_{n=0}^{N-1} |\epsilon_n|$$

dire que le schéma (I.1) est consistant avec le système différentiel (1), revient à dire que:

$$\sum_{n=0}^{N-1} |\epsilon_n| \longrightarrow 0 \quad \text{dès que} \quad h \longrightarrow 0$$

I.1.2 Stabilité:

Cette notion est intrinsèque au schéma de résolution numérique, définie comme étant la propriété qui fait que toute petite perturbation sur les données du schéma n'entraîne qu'une petite perturbation sur la solution. Ceci nous permet de pallier à l'effet des erreurs d'arrondis et leur propagation au niveau de la résolution.

Soit donc:

$$\begin{cases} Y_n + 1 = Y_n + h_n \cdot \phi (t_n, Y_n, h_n) & Y_0 \text{ fixée} \\ Z_n + 1 = Z_n + h_n \cdot (\phi (t_n, Z_n, h_n) + \epsilon_n) & Z_0 \text{ fixée} \end{cases}$$

La stabilité de la méthode est affirmée si pour tout:

$h \leq h_0$, (h_0 pas maximum), il existe deux constantes M_1 et M_2 indépendantes de h , Z_0 , Y_0 et ϵ_n tel que:

$$\text{Max } | Y_n - Z_n | \leq M_1 | Y_0 - Z_0 | + M_2 \text{ Max } | \epsilon_n |$$

ce résultat a été établi par CROUZEIX, et MINGNOT en [1], GEAR [2].

I.1.3 Convergence:

Ecrivons que:

$Y(t_n)$ est la solution réelle

et Y_n est la solution calculée à partir du schéma (I.1)

Alors l'erreur sera définie comme suit:

$$e_n = y(t_n) - Y_n$$

La convergence est alors assurée si:

$$\begin{aligned} \text{Max } | e_n | &= \text{Max } | Y(t_n) - Y_n | \longrightarrow 0 \\ h &= \text{Max } h_n \longrightarrow 0 \\ &0 \leq n \leq N \end{aligned}$$

I.2 Schéma général des méthodes de RUNGE-KUTTA à q pas intermédiaires:

Le schéma général de RUNGE-KUTTA à q pas intermédiaires est:

$$\text{I.3 } \begin{cases} Y_{n,i} = Y_n + h_n \cdot \sum_{j=1}^q a_{ij} \cdot f(t_n, j; Y_n, j) & i = 1, q \quad (\text{I.3 a}) \\ Y_{n+1} = Y_n + h_n \cdot \sum_{j=1}^q b_j f(t_n, j; Y_n, j) & (\text{I.3 b}) \end{cases}$$

Y_n étant supposée connue, le calcul de q quantités auxiliaires $Y_{n,i}$ définies comme solutions de q équations (I.3.a) lorsque après permutation éventuelle des indices, la matrice \underline{a} des éléments a_{ij} devient strictement triangulaire inférieure (i.e. $a_{ij} = 0$ si $i \leq j$), le calcul de $Y_{n,i}$ ne demande que des évaluations de f , il en est de même pour Y_{n+1} d'après (I.3 b) la méthode est alors explicite.

Il est plus aisé de montrer que le schéma (I.3) vérifie les propriétés précédemment annoncées.

- partons de [1], [2]:

$$\|\epsilon_n\| \leq Kh^p \quad (I.4)$$

et comme la méthode de RUNGE-KUTTA est d'ordre $p > 1$, alors:

$$\|\epsilon_n\| \rightarrow 0 \quad \text{dès que} \quad h_n \rightarrow 0$$

- Etudions la sensibilité du schéma face aux perturbations (dont les valeurs sont bornées dans l'espace et le temps):

$$\begin{cases} Z_{n,i} = Z_n + h_n \cdot \sum_{j=1}^q a_{ij} \cdot f(t_{n,j}, Z_{n,j}) \\ Z_{n+1} = Z_n + h_n \cdot \sum_{j=1}^q b_j \cdot f(t_{n,j}, Z_{n,j}) + h_n \epsilon_n \end{cases} \quad i = 1, \dots, q$$

On démontre (voir annexe B1), le résultat suivant:

$$\|Y_n - Z_n\| \leq e^{CT} \|Y_0 - Z_0\| + \frac{e^{CT} - 1}{c} \max_n \|\epsilon_n\| \quad (I.5)$$

Ce qui montre la stabilité du schéma avec:

$$M_1 = e^{CT} \quad \text{et} \quad M_2 = \frac{e^{CT} - 1}{c}$$

La convergence est déduite directement de (I.4) et (1.5)

pour cela, il suffit de poser:

$$Y(t_0) = Z_0 = Y_0 \quad \text{et} \quad Y(t_n) = Z_n$$

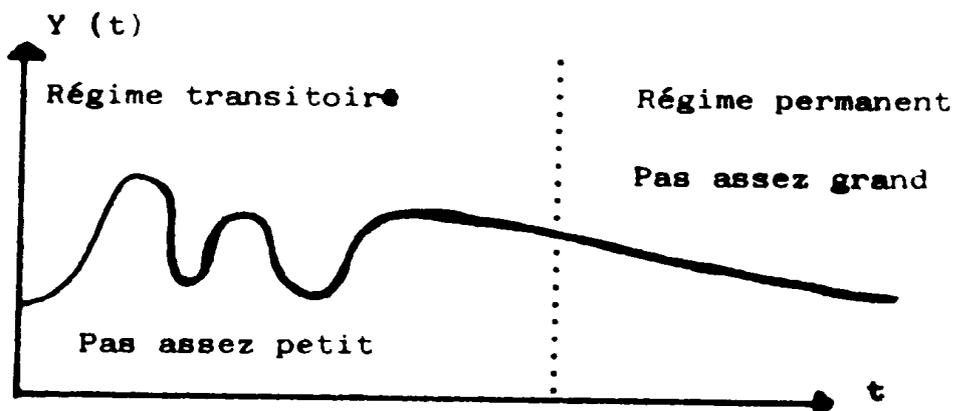
et comme le schéma est consistant, on obtient alors le résultat recherché, à savoir, la convergence du schéma (I.1)

$$\|e_n\| = \|Y_n - Y(t_n)\| \leq \frac{e^{CT} - 1}{c} \max_n \|E_n\| \longrightarrow 0 \quad (I.6)$$

$$\text{qd } h \longrightarrow 0$$

I.3 Comportement asymptotique de l'erreur:

Pour adapter la méthode à la régularité de la solution, minimiser son coût calcul, et détecter les éventuelles "SINGULARITES" du problème différentiel. La mise en oeuvre des méthodes de RUNGE-KUTTA, se fait avec contrôle de pas. Les singularités du problème différentiel sont les points où le schéma a du mal à approcher la solution de ce dernier. Le contrôle du pas permet alors de choisir au voisinage de ces points un pas h_n suffisamment petit pour atteindre la précision exigée ou de constater l'impossibilité de l'obtenir (la précision demandée). Le changement du pas obéit à une technique explicite basée sur le comportement asymptotique de l'erreur, donc sur son estimation locale.



Le schéma permet de montrer comment le pas doit évoluer compte-tenu des évolutions de Y (t).

N.B.: La relation (I.6) nous permet de constater que contrôler l'erreur locale revient donc essentiellement à contrôler l'erreur de consistance E_n .

I.3.1 Expression de l'erreur:

Pour connaître le comportement de l'erreur, il faut d'abord trouver son expression, pour cela, donnons-nous une fonction $\Theta(t)$ continue, lipschitzienne sur I telle que:

$$0 < \Theta(t) \leq 1 \quad \text{et} \quad h_n = h (\Theta(t) + o(h))$$

sachant que pour une méthode d'ordre p , l'erreur de discrétisation obéit à la relation suivante:

$$e_n = Y(t_n) - Y_n = h^p \sigma_1(t_n) + (N - Nh) \sigma_0(t_n) + o(h^{p+1}) \quad (I.7)$$

ou:

σ_0 et σ_1 sont solutions des équations différentielles suivantes.

$$\left\{ \begin{array}{l} \sigma_0'(t) = \frac{\partial f}{\partial y}(t, y(t)) \sigma_0(t) \\ \sigma_0(t_0) = 1 \end{array} \right. \quad (I.7.a) \quad \left\{ \begin{array}{l} \sigma_1'(t) = \frac{\partial f}{\partial y}(t, y(t)) \sigma_1(t) + \Theta(t) \psi_p(t, y(t)) \\ \sigma_1(t_0) = 0 \end{array} \right. \quad (I.7.8)$$

Ce résultat est établi dans CROUZEIX, MIGNOT [1] avec:

- $h^p \sigma_1(t_n)$ erreur due à la méthode
- $(N - Nh) \sigma_0(t_n)$ due à l'imprécision sur les conditions initiales.

Notre préoccupation réside dans le choix du pas h_n , de sorte que l'erreur soit aussi faible que possible.

I.3.2 Choix optimal du Pas h_n :

Le pas optimal est celui qui nous permet d'avoir une erreur:

$$e_n = Y(t_n) - Y_n$$

voisine d'un certain seuil de tolérance.

Le choix du pas donc, celui de $\Theta(t)$ n'influe pas sur le terme $(N - Nh) \sigma_0(t_n)$, par contre, il joue un rôle prépondérant dans l'erreur de méthode.

Sachant que le coût de la méthode en nombre du pas est:

$$N = \frac{h_0}{h_0} + \frac{h_1}{h_1} + \dots + \frac{h_{n-1}}{h_{n-1}} = \frac{1}{h} \left[\frac{h_0}{\theta(t_0)} + \frac{h_1}{\theta(t_1)} + \dots + \frac{h_{n-1}}{\theta(t_{n-1})} \right] = \frac{1}{h} \int_{t_0}^{t_0+T} \frac{dS}{\theta(S)} + o(1) \quad (1)$$

On se demande alors pour un coût donné N comment choisir les points t_n , de sorte que l'erreur finale $|Y(t_n) - Y_n|$ soit la plus faible possible.

Ce qui revient à dire:

$$\left\{ \begin{array}{l} \text{Pour: } \frac{1}{h} \int_{t_0}^{t_0+T} \frac{dS}{\theta(S)} \text{ fixé trouver } h \text{ et } \theta \text{ tel que:} \\ \\ h^p \sigma_1(t_n) \text{ soit minimum} \end{array} \right. \quad (I.8)$$

Posons $\psi(t) = h \theta(t)$ et cherchons d'abord l'expression de $\sigma_1(t_n)$.
à partir de (I.7 a) et (I.7b), on démontre (voir Annexe B).

$$\sigma_1(t) = \sigma_0(t) \int_0^t \frac{\theta^p(\delta) \psi_p(\delta, y(\delta))}{\sigma_0(\delta)} d\delta$$

(I.8) devient alors:

$$\left\{ \begin{array}{l} \int_{t_0}^{t_0+T} \frac{d\delta}{\psi(\delta)} = N \text{ fixé} \\ \\ \left| \int_{t_0}^{t_0+T} \frac{\theta^p(\delta) \psi_p(\delta, y(\delta)) d\delta}{\sigma_0(\delta)} \right| \text{ Minimum} \end{array} \right.$$

Sous l'hypothèse que $\psi_p(\delta, y(\delta))$ ne s'annule pas et garde un signe constant sur I on aura:

$$\varphi(\delta) = h \sigma(\delta) = \lambda \left[\frac{\sigma_0(\delta)}{\Psi_p(\delta, y(\delta))} \right]^{\frac{1}{p+1}}$$

résultat établi

Avec:

$$\lambda = \frac{1}{M} \int_{t_0}^{t_0+T} \left| \frac{\Psi_p(\delta, y(\delta))}{\sigma_0(\delta)} \right|^{\frac{1}{p+1}} d\delta$$

dans [1]

d'autre part, la formule de TAYLOR nous donne:

$$\varepsilon_n = y(t_{n+1}) - y(t_n) - hn \phi(t_n, Y_n, hn) = hn \Psi_p(t_n, Y(t_n)) + o(hn)^{p+2}$$

d'où on tire:

$$\varepsilon_n = \pm \lambda^{p+1} \sigma_0(t_n) + o(hn)^{p+2} \longrightarrow \frac{\varepsilon_n}{\sigma_0(t_n)} = \pm \lambda^{p+1} + o(hn)^{p+2}$$

On peut donc en conclure que le choix optimal du pas conduit au résultat suivant:

$$\frac{\varepsilon_n}{\sigma_0(t_n)} \text{ reste asymptotiquement constant.}$$

I.4 Contrôle du pas:

La connaissance de l'erreur locale permet le contrôle du pas. Ce dernier nécessite le calcul d'une valeur approchée $\tilde{\varepsilon}_n$ de cette erreur qui sera utilisée dans le programme numérique sur calculateur.

I.4.1 Estimation de l'erreur:

Nous avons vu que la formule de TAYLOR nous donnait:

$$\varepsilon_n = hn \Psi_p(t_n, y(t_n)) + o(hn)^{p+2}$$

Ce qui nous amène à prendre:

$$\tilde{\varepsilon}_n = hn \Psi_p(t_n, Y_n)$$

La différence entre l'erreur et l'erreur approchée nous donne:

$$\varepsilon_n - \tilde{\varepsilon}_n = h_n^{P+1} \left[\Psi_p(t_n, Y(t_n)) - \Psi_p(t_n, Y_n) \right] + O(h_n^{P+2})$$

Le théorème des accroissements finis nous ramène à:

$$\varepsilon_n - \tilde{\varepsilon}_n = h_n^{P+1} \frac{\partial \Psi_p}{\partial y}(t_n, Y_n) \left[Y(t_n) - Y_n \right] + O(h_n^{P+2})$$

Or: $y(t_n) - y_n = O(h_n^P)$

On obtient alors:

$$\varepsilon_n - \tilde{\varepsilon}_n = h_n^2 O(h_n^P) \implies \varepsilon_n = \tilde{\varepsilon}_n + h_n^2 O(h_n^P)$$

d'où $\tilde{\varepsilon}_n$ est une bonne approximation de ε_n .

I.4.2 Contrôle du pas:

Une fois la valeur approchée $\tilde{\varepsilon}_n$ de l'erreur ε_n estimée, posons:

$$\tilde{\varepsilon}_n = h_n^{P+1} \Psi_p(t_n, Y_n) = \beta$$

Comparons cette valeur à la précision exigée, deux cas peuvent se présenter:

- $\tilde{\varepsilon}_n$ supérieure à la précision exigée, alors on diminue le pas.
- $\tilde{\varepsilon}_n$ inférieure à la précision exigée, alors on augmente le pas, pour cela, nous noterons:

$$\tilde{\varepsilon}_n = h_n^* \Psi_p(t_n, Y_n) = \beta$$

L'erreur obtenue avec le pas h_n^* et qui répond à la précision exigée. Le changement du pas se fera grâce à la relation suivante:

$$h_n^* = h_n \left(\frac{\beta}{\alpha} \right)^{\frac{1}{P+1}}$$

si $\beta > \alpha$ alors $h_n^* < h_n$

si $\beta \leq \alpha$ alors $h_n^* \geq h_n$

I.5 Schéma final de RUNGE-KUTTA adopté:

Le schéma final adopté est basé sur deux méthodes de RUNGE-KUTTA emboîtées, l'une d'ordre \underline{P} , l'autre d'ordre au moins $P' = P + 1$. Ce choix est dû au fait qu'il permet le calcul de $\tilde{\xi}_n$ à un coût moindre.

Dans le programme que nous avons mis en oeuvre sur calculateur, nous avons utilisé RKpp' avec $p = 3$ et $p' = 4$, la méthode d'ordre 3 permet le calcul de \underline{Y}_n solution approchée de $y(t_n)$, celle d'ordre 4 permet l'estimation de l'erreur. Pour faciliter la programmation de la méthode, nous la résumons grâce aux équations suivantes:

$$\left\{ \begin{array}{l} K_{n,i} = f(t_{n,i}; Y_n + h_n \cdot \sum_{j=1}^q a_{ij} \cdot K_{n,j}) \quad i = 1, \dots, q \\ Y_{n+1} = Y_n + h_n \cdot \sum_{j=1}^q b_j \cdot K_{n,j} \\ t_{n,i} = t_n + C_i h_n \quad i = 1, \dots, q \end{array} \right.$$

ou $\underline{K}_{n,i}$ est vue comme approximation de $y'(t_{n,i})$.

I.5.1 Calcul des coefficients de la méthode:

Les coefficients suivants ont été calculés par [1], pour la méthode d'ordre 4, disposés sous forme de tableau, les coefficients sont:

c_i	c_i	a_{ij}					
$\left\{ \begin{array}{l} 0 \\ 2/7 \\ 4/7 \\ 6/7 \end{array} \right.$	0	0	0	0	0	$\left. \begin{array}{l} 0 \\ 0 \\ 0 \\ 0 \end{array} \right\} \rightarrow$	a'_{ij}
	2/7	2/7	0	0	0		0
	4/7	- 8/35	4/5	0	0		0
	6/7	29/42	- 2/3	5/6	0		0

	1	1/6	1/6	5/12	1/4	$\left. \begin{array}{l} 0 \\ 0 \end{array} \right\} \rightarrow$	b'_j

		$\frac{11}{96}$	$\frac{7}{24}$	$\frac{35}{96}$	$\frac{7}{48}$	$\left. \begin{array}{l} 0 \\ 0 \end{array} \right\} \rightarrow$	b_j
							$\frac{1}{12}$

Pour la méthode d'ordre 3, il suffit de supprimer la dernière ligne et la dernière colonne, soit donc: (a'_{ij} , b'_j , c'_i).

I.5.2 Estimation de $\bar{\epsilon}_n$:

- Soit: $\epsilon_n = Y(t_{n+1}) - Y(t_n) - h_n \cdot \phi(t_n, y(t_n), h_n)$

Pour la méthode d'ordre 3.

- et soit: $\epsilon_n^* = Y(t_{n+1}) - Y(t_n) - h_n \cdot \phi^*(t_n, y(t_n), h_n)$

Pour la méthode d'ordre 4.

En faisant la différence des deux expressions de l'erreur, on obtient [1]:

$$\epsilon_n = \epsilon_n^* + h_n^2 O(h_n)$$

avec: $\epsilon_n^* = \phi^*(t_n, y(t_n), h_n) - \phi(t_n, y(t_n), h_n)$

I.6 Différents types de contrôle d'erreur:

Nous allons décrire les différents types de contrôle d'erreur employés dans le programme de résolution implanté sur calculateur.

I.6.1 1er type de contrôle d'erreur [1]:

Basé sur le fait que $\left| \frac{\tilde{\epsilon}_n}{\tilde{\sigma}_0(t_n)} \right|$ soit minimum constant. Donnons-nous un paramètre μ qui tend vers zéro, et calculons à chaque pas $\tilde{\epsilon}_n$ et $\tilde{\sigma}_0(t_n)$.

Valeur approchée de $\tilde{\sigma}_0(t_n)$ et de la sorte, on ajuste h de telle manière que:

$$\left| \frac{\tilde{\epsilon}_n}{\tilde{\sigma}_0(t_n)} \right| = \mu$$

Nous avons d'autre part:

$$\tilde{\epsilon}_n = h^{P+1} \Psi_p(t_n, y_n) \Rightarrow h = \left| \frac{\tilde{\sigma}_0(t_n) \mu}{\Psi_p(t_n, y_n)} \right|^{\frac{1}{P+1}}$$

mais alors si $\Psi_p(t_n, y_n)$ vient à s'annuler, on aura un pas très grand et de là la relation:

$$\epsilon_n = \tilde{\epsilon}_n + h^2 O(h^P)$$

n'est plus valable du moment que le terme $h^2 O(h^P)$ ne peut plus être négligé, afin d'éviter cela, on impose:

$$\left| \frac{\tilde{\epsilon}_n}{\tilde{\sigma}_0(t_n)} \right| \leq \mu \quad \text{et} \quad h \leq 10 \left(\mu \tilde{\sigma}_0(t_n) \right)^{\frac{1}{P+1}}$$

d'un point de vue pratique, l'algorithme est décrit comme suit:

On part avec le pas $h_0 = \left(\mu \right)^{\frac{1}{P+1}}$ et on passe de t_n à t_{n+1} ainsi:

h_n étant connu pour le pas précédent, on calcule:

$$y_{n+1}, \quad \tilde{\epsilon}_n, \quad \tilde{\sigma}_0(t_n)$$

- Si: $0,9 \mu \leq \frac{\tilde{\epsilon}_n}{\sigma_0(t_n)} \leq 1,1 \mu$ alors $h_{n+1} = h_n$ on conserve le pas.

- Si: $\frac{\tilde{\epsilon}_n}{\sigma_0(t_n)} < 0,9 \mu$ alors $h_{n+1} = \inf \left(h_n \left(\frac{\mu \sigma_0(t_n)}{|\tilde{\epsilon}_n|} \right)^{\frac{1}{p+1}}, 10(\mu \sigma_0(t_n))^{\frac{1}{p+1}} \right)$

- Si: $1,1 \mu < \left| \frac{\tilde{\epsilon}_n}{\sigma_0(t_n)} \right| \leq 1,2 \mu$ alors $h_{n+1} = h_n \left(\mu \frac{|\sigma_0(t_n)|}{|\tilde{\epsilon}_n|} \right)^{\frac{1}{p+1}}$

- Si: $1,2 \mu < \left| \frac{\tilde{\epsilon}_n}{\sigma_0(t_n)} \right|$ alors on rejette le pas h_n , et la valeur Y_{n+1} et on démarre avec:

$$h_n^* = h_n \left(\frac{0,9 \mu |\sigma_0(t_n)|}{|\tilde{\epsilon}_n|} \right)^{\frac{1}{p+1}}$$

REMARQUE:

L'utilisateur peut choisir les coefficients (0,9 - 1,1 - 1,2) en fonction de la précision souhaitée.

I.6.2 Deuxième type de contrôle de l'erreur [1]:

Cette fois le contrôle d'erreur est relatif sur l'intervalle d'intégration $[t_{n-1}, t_n]$, on cherche à rendre la quantité

$$\left| \frac{\tilde{\epsilon}_n}{h_n \sigma_0(t_n)} \right| \leq \nu$$

ou ν est un paramètre qui tend vers zéro, ensuite, on procède de la même manière que dans le cas précédent, en remplaçant:

$$\frac{1}{p+1} \sqrt{\quad} \quad \text{par} \quad \frac{1}{p} \sqrt{\quad}$$

I.6.3 Choix de $\tilde{\sigma}_0(t_n)$: on va donner les choix possibles de $\tilde{\sigma}_0(t_n)$ valeur approchée de $\sigma_0(t_n)$:

- $\tilde{\sigma}_0(t_n) = 1$ Il s'agit de rendre l'erreur de consistance, absolue constante.

- $\tilde{\sigma}_0(t_n) = |Y_n|$ Il s'agit de rendre l'erreur de consistance, relative constante; pour cela, il faut d'abord s'assurer au préalable que $|Y_n|$ ne s'annule pas sur l'intervalle $I = [0, T]$.

- $\sigma_0(t_n)$ étant définie comme la solution de:

$$\begin{cases} \sigma_0'(t_n) = \frac{\partial f}{\partial y}(t_n, y_n) \cdot \sigma_0(t_n) \\ \sigma_0(t_0) = 1 \end{cases}$$

Or: $\frac{\partial f}{\partial y} = a_{ij}$ alors $\sigma_0(t_n) = \sum_i a_i e^{\lambda_i t_n}$

On peut prendre par exemple pour $\sigma_0(t_n)$, la valeur:

$$\tilde{\sigma}_0(t_n) = e^{\lambda_{\max} t_n}$$

Cela est justifié par le rôle de λ_{\max} .

I.6.4 Troisième type de contrôle d'erreur:

L'évolution de la solution se traduit par celle de sa dérivée, on va exploiter ceci pour concevoir une méthode du réajustement du pas, elle est basée essentiellement sur des considérations expérimentales:

Supposons que la solution $y(t)$ évolue suivant la figure (2).

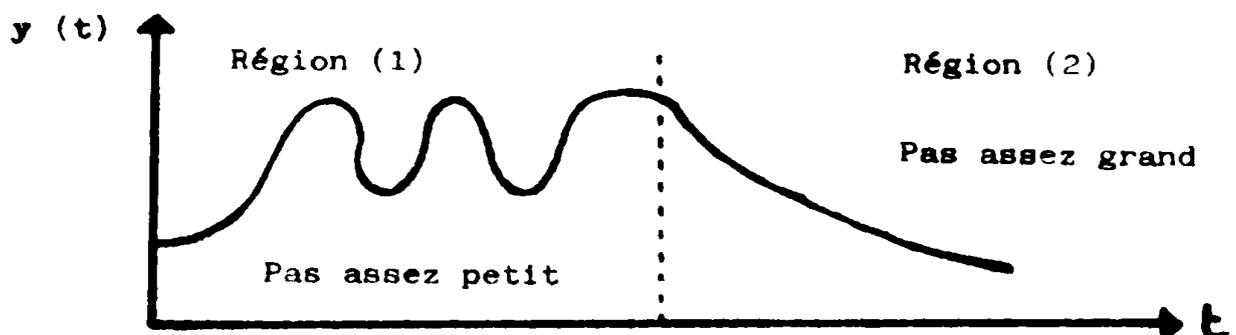


Fig. 2

Région (1): Evolution rapide qui peut traduire un régime transitoire par exemple, la dérivée est alors très grande en valeur absolue.

Région (2): Evolution lente (régime permanent) qui traduit une solution assez régulière, d'où une dérivée très faible en valeur absolue.

Soit ϕ cette dérivée; et h_0 le pas maximum autorisé, l'expression du réajustement du pas est la suivante:

$$h_n = \frac{h_0}{1 + |\phi|}$$

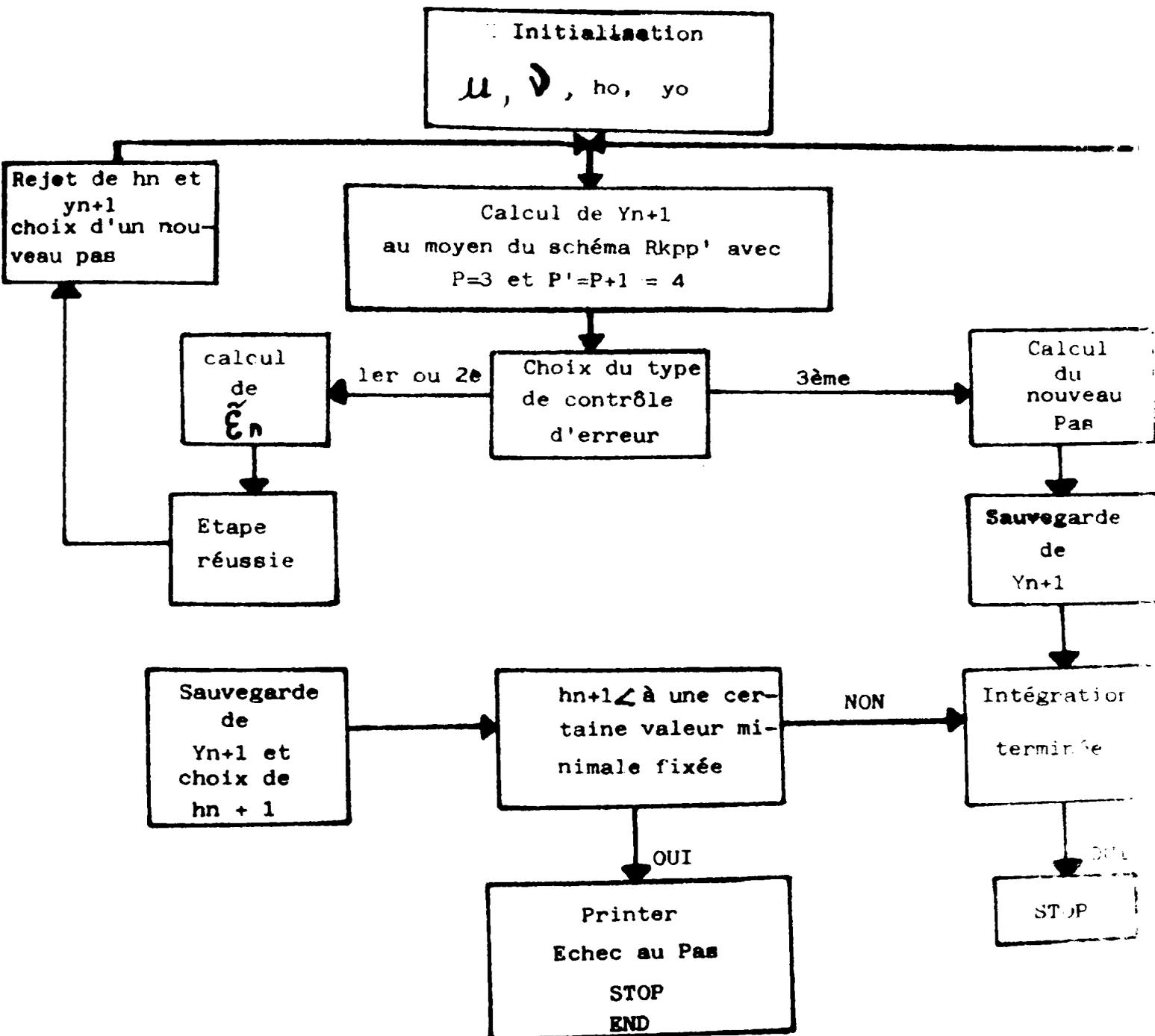
Si: $|\phi| \gg$ alors $h_n \ll h_0$

Si: $|\phi| \ll$ alors $h_n \simeq h_0$

Si: $|\phi| = 0$ alors $h_n = h_0$

I,7 ORGANIGRAMME:

L'organigramme suivant résume les différents types de contrôle de l'erreur:



CHAPITRE II: MISE EN OEUVRE D'UN SCHEMA D'ADAMS A PAS ET ORDRE VARIABLES, EXPRIME A L'AIDE DES DIFFERENCES DIVISEES:

Les méthodes multi-pas d'ordre r permettent de calculer une valeur approchée Y_{n+i} de $y(t_{n+1})$ en utilisant les valeurs approchées $Y_n, Y_{n-1}, \dots, Y_{n-r+1}$ contrairement aux méthodes mono-pas (telle que la méthode de RUNGE-KUTTA), qui n'utilisent que la valeur approchée Y_n pour pouvoir calculer Y_{n+1} . Pour une même précision fixée, les premières permettent une résolution des équations différentielles ordinaires avec un coût plus faible.

Parmi ces méthodes, celles d'ADAMS sont les plus connues pour leur précision et leur efficacité dans l'intégration d'un système d'équations différentielles ordinaires non raides. Pour les exprimer, on choisit la formulation de KRÖGH basée sur la décomposition des polynômes à l'aide des différences divisées. Le choix de cette formulation est motivé par le fait qu'elle diminue le nombre de calculs à faire, SHAMPINE et GORDON [3] ont décrit cette formulation pour le schéma prédicteur-correcteur $P_{kE}C_{k+1E}$, tel que le schéma prédicteur d'ordre k est celui d'ADAMS-BASHFORTH et le schéma correcteur d'ordre $(k+1)$ est celui d'ADAMS-MOULTON avec k variant de 1 à 12.

Dans ce chapitre, nous définissons un programme correspondant au schéma $P_{kE}C_{k+1E}$ k variant de 1 à 12, avec une procédure assez fine pour le démarrage de l'intégration, nécessaire pour la précision dans cette phase, surtout dans l'intégration des systèmes différentiels instables.

II.1 Méthode d'ADAMS d'ordre $(r+1)$ à pas variable:

Afin de résoudre le problème différentiel (1), on utilise le schéma suivant:

Une subdivision:

$T_0 = 0 < t_1 < t_2 < \dots < t_n < t_{n+1} < \dots < t_n = T$ de l'intervalle
 $I = [0, T]$ est choisie;

on note alors $h_n = t_{n+1} - t_n \quad \forall n \in [0, N-1]$

les quantités f_0, \dots, f_n ($n \geq r$) étant des valeurs approchées de $Y'(t_0, \dots, t_n)$ que l'on suppose connues, on calcule alors une valeur approchée Y_{n+1} de $Y(t_{n+1})$ au moyen de la relation:

$$Y_{n+1} = Y_n + \int_{t_n}^{t_{n+1}} R(t) dt \quad (II.1)$$

où $R(t)$ est l'un des polynômes $P_{r,n}(t)$ ou $Q_{r,n}(t)$ de degrés inférieur ou égal à r tel que:

$$P_{r,n}(t_n - i) = f_{n-i} \quad i = 0, 1, \dots, r$$

et $Q_{r,n}(t_n - i) = f_{n-i} \quad i = -1, \dots, r-1$

Ces polynômes sont définis par les relations:

$$P_{r,n}(t) = \sum_{i=0}^r L_{n,r,i}(t) \cdot f_{n-i}$$

$$Q_{r,n}(t) = \sum_{i=-1}^{r-1} L_{n+1,r,i}(t) \cdot f_{n-i}$$

où $L_{n,r,i}(t)$ est le polynôme de LAGRANGE tel que:

$$L_{n,r,i}(t) = \prod_{\substack{j=0 \\ i \neq j}}^r (t - t_{n-j}) / (t_{n-i} - t_{n-j})$$

Deux schémas sont alors obtenus à partir de (II.1)

- Dans le premier cas où $R(t) = P_{r,n}(t)$, le schéma est celui d'ADAMS-BASHFORTH:

$$Y_{n+1} = Y_n + h_n \cdot \sum_{i=0}^r b_{n,i} \cdot f_{n-i} \quad (II.2)$$

$$f_{n+1} = f(t_{n+1}, y_{n+1})$$

avec:

$$b_{n,i} = \frac{1}{h_n} \int_{t_n}^{t_{n+1}} L_{n,r,i}(t) dt$$

c'est donc un schéma explicite.

- Dans le deuxième cas ou $R(t) = Q_{r,n}(t)$, le schéma est celui d'ADAMS-MOULTON:

$$y_{n+1} = Y_n + h_n \cdot \sum_{i=-1}^{r-1} b_{n,i}^* \cdot f_{n-i}$$

$$f_{n+1} = f(t_{n+1}, y_{n+1})$$

avec:

$$b_{n,i}^* = \frac{1}{h_n} \int_{t_n}^{t_{n+1}} L_{n+1,r,i}(t) dt$$

et:

$$b_n^* = b_{n,-1}^* = \frac{1}{h_n} \int_{t_n}^{t_{n+1}} \prod_{j=0}^{r-1} \frac{(t-t_{n-j})}{(t_{n+1}-t_{n-j})} dt$$

on aura alors:

$$Y_{n+1} - h_n \cdot b_n^* \cdot f(t_{n+1}, y_{n+1}) = Y_n + \sum_{i=0}^{r-1} h_n \cdot b_{n,i}^* \cdot f_{n-i} \quad (Q_{r,n}^{t_{n+1}} \text{ connue})$$

la résolution de l'équation en Y_{n+1} n'est pas immédiate, c'est donc un schéma implicite.

II.2 Méthode d'ADAMS employée sous forme Prédicteur-Correcteur:

La méthode d'ADAMS-MOULTON est plus précise que la méthode d'ADAMS-BASHFORTH [1], [3], mais son caractère implicite rend son emploi plus délicat: il s'agit en fait de résoudre l'équation en Y_{n+1} suivante:

$$Y_{n+1} + h_n \cdot b_n^* \cdot f(t_{n+1}, y_{n+1}) = Y_n + \sum_{i=0}^{r-1} h_n \cdot b_{n,i}^* \cdot f_{n-i} \quad (Q_{r,n}^{t_{n+1}} \text{ connue}).$$

Or, dans la majorité des cas, cette résolution nécessitera l'emploi d'une méthode itérative.

Pour initialiser cette dernière, on calcule $Y_{n+1}^{(P)}$, une valeur approchée au moyen d'un schéma explicite appelé schéma prédicteur, puis on obtient $f^{(P)}(t_{n+1}, y_{n+1})$ par une évaluation de f .

Enfin, les valeurs approchées de Y_{n+1} et f_{n+1} , sont obtenues au moyen d'un schéma implicite appelé schéma correcteur.

Dans notre cas, les schémas prédicteur-correcteur sont respectivement les méthodes d'ADAMS-BASHFORTH et d'ADAMS-MOULTON de même ordre, cet ordre peut maintenant varier à chaque pas.

II.2.1 Définition du schéma:

Soit $k_n = r_n + 1$ l'ordre à l'instant t_{n+1} et $K = r + 1$ l'ordre maximal.

* schéma Prédicteur d'ordre (r+1)

La valeur prédite de la solution est donnée par le schéma explicite d'ADAMS-BASHFORTH.

$$(Prédiction) \quad Y_{n+1}^{(P)} = Y_n + h_n \sum_{i=0}^{r_n} b_{n,i} \cdot f_{n-i}$$

Après une évaluation de f , on obtient:

$$(Evaluation) \quad f_{n+1}^{(P)} = f(t_{n+1}, Y_{n+1}^{(P)})$$

* schéma correcteur d'ordre (r+1)

Les approximations suivantes de Y_{n+1} et f_{n+1} sont obtenues à l'aide du schéma implicite d'ADAMS-MOULTON:

$$(Correction) \quad Y_{n+1} = Y_n + h_n b_n^* f_{n+1}^{(P)} + h_n \sum_{i=0}^{r_n-1} b_{n,i}^* \cdot f_{n-i}$$

après une évaluation de f on obtient:

(Evaluation) $f_{n+1} = f(t_{n+1}, y_{n+1})$

II.2.2 Ordre et stabilité du schéma:

On définit l'erreur de consistance du schéma par:

$$\mathcal{E}_n = y(t_{n+1}) - y(t_n) - h_n \cdot b_n^* f(t_{n+1}, Z_{n+1}) - h_n \sum_{i=0}^{r_n-1} b_{n,i}^* \cdot f_{n,i} \quad (P)$$

avec:

$$Z_{n+1}^P = y(t_n) + h_n \sum_{i=0}^{r_n} b_{n,i} \cdot y'(t_n - i)$$

Soit: \mathcal{E}_n^P et \mathcal{E}_n^C les erreurs de consistance correspondant respectivement au schéma prédicteur et au schéma correcteur d'ordre $k_n = r_n$.

$$\mathcal{E}_n^P = y(t_{n+1}) - y(t_n) - h_n \sum_{i=0}^{r_n} b_{n,i} \cdot y'(t_n - i) = y(t_{n+1}) - Z_{n+1}^P$$

$$\mathcal{E}_n^C = y(t_{n+1}) - y(t_n) - h_n \cdot b_n^* \cdot y'(t_{n+1}) - h_n \sum_{i=0}^{r_n-1} b_{n,i}^* \cdot y'(t_n - i)$$

En écrivant que:

$$\mathcal{E}_n = y(t_{n+1}) - y(t_n) - h_n \cdot b_n^* \cdot f(t_{n+1}, Z_{n+1}^P) - h_n \sum_{i=0}^{r_n-1} b_{n,i}^* \cdot y'(t_n - i) + h_n \cdot b_n^* \cdot y'(t_n - i) - h_n \cdot b_n^* \cdot y'(t_n - i)$$

On obtient alors:

$$\mathcal{E}_n = \mathcal{E}_n^C + h_n \cdot b_n^* (f(t_{n+1}, y(t_{n+1})) - f(t_{n+1}, Z_{n+1}^P))$$

Or:

$$f(b) - f(a) = (b - a) f'(c)$$

Ainsi:

$$\varepsilon_n = \varepsilon_n^c + h_n \left[\frac{\partial}{\partial y} f(t_{n+1}, y_{n+1}) \varepsilon_n^p + o((\varepsilon_n^p)^2) \right] b_n^*$$

$$\varepsilon_n = \varepsilon_n^c + h_n b_n^* \frac{\partial}{\partial y} f(t_{n+1}, y_{n+1}) \varepsilon_n^p + o(h_n (\varepsilon_n^p)^2)$$

or, les schémas prédicteur-correcteur sont tous deux d'ordre $k_n = r_n + 1$ on a:

$$\varepsilon_n^c = o(h_n^{k_n+1}) \quad ; \quad \varepsilon_n^p = o(h_n^{k_n+1})$$

résultats établis dans [1] et [3]

on obtient alors:

$$\varepsilon_n = \varepsilon_n^c + o(h_n^{k_n+2})$$

on remarque alors que le schéma prédicteur-correcteur conserve l'ordre k_n et que sa précision est celle du schéma correcteur, en l'occurrence celui d'ADAMS-MOULTON.

Pour la stabilité, on a le résultat suivant [1]:

on suppose qu'on intègre avec le schéma PKECKE (Prédicteur-Correcteur tous deux d'ordre k), depuis t_0 et soit $r = k - 1$, si l'hypothèse suivante est vérifiée:

$$\left\{ \begin{array}{l} \text{il existe trois (3) constantes } \vartheta, \beta, \gamma \text{ tel que:} \\ \text{Pour } n \geq 0 \quad \sum_{i=0}^r |b_{n,i}| \leq \vartheta ; \quad \sum_{i=-1}^{r-1} |b_{n,i}^*| \leq \beta \text{ et } h_n |b_n^*| \leq h \gamma \end{array} \right. \quad (\text{II.4})$$

ou h est la valeur maximale du Pas, on obtient dans ce cas:

$$|Y(t_n) - Y_n| \leq e^{K(h)(t_n - t_0)} \max |Y(t_0) - Y_0| + c h^k \int_{t_0}^{t_n} e^{K(h)(t_n - \tau)} |Y(\tau)| d\tau$$

avec $K(h) = \beta L + h \gamma L^2$ constante de stabilité ou L est la constante de LIPSCHITZ.

REMARQUE:

L'hypothèse (II.4) n'est vérifiée que si l'augmentation du pas est bornée, c'est-à-dire:

$$\frac{h_{n+1}}{h_n} \leq S, \forall n = 0, \dots, N-1$$

II.3 Formulation du schéma prédicteur-correcteur à l'aide des différences divisées:

La formulation de \hat{KROGH} permet d'écrire les méthodes d'ADAMS sous une forme adaptée à l'implantation sur ordinateur, surtout lorsqu'on souhaite contrôler l'ordre et le pas.

Cette formulation a été entièrement décrite par SHAMPINE et GORDON [3] pour le schéma $P_{kE}C_{k+1}E$, on reprend ici la même démarche pour le schéma $P_{kE}C_kE$, du fait que les erreurs estimées sont celles à l'ordre k .

Dans la formulation naturelle du schéma (décomposition des polynômes sur les polynômes de LAGRANGE vu précédemment), les coefficients (b_n, i) et (\hat{b}_n, i) resteront constants, d'un pas à l'autre, tant que l'ordre k_n est conservé, et si le pas est constant depuis k_n étapes. Dans le cas contraire, tous les coefficients sont à calculer. Par contre, la formulation de \hat{KROGH} réduit le nombre des calculs à faire puisque un changement d'ordre n'affecte pas les coefficients et l'utilisation de pas constants dans les dernières étapes réduit notablement leurs calculs.

II.3.1 Expression du schéma et choix des variables:

Introduisons quelques définitions utiles pour la suite:

$$h_i = t_i - t_{i-1}$$

$$\begin{aligned} \Psi_{i, n+1} &= t_{n+1} - t_{n+1-i} & i \geq 1 \\ &= h_{n+1} + h_n + \dots + h_{n+2-i} \end{aligned}$$

$$\beta_{1, n+1} = 1$$

$$\beta_{i, n+1} = \frac{\Psi_{1, n+1} \dots \Psi_{i-1, n+1}}{\Psi_{1, n} \dots \Psi_{i-1, n}} = \frac{\prod_{i, n} (t_{n+1})}{\prod_{i, n-1} (t_n)} \quad i \geq 2 \quad (\text{II.5})$$

$$\phi_{1, n} = f [t_n]$$

$$\begin{aligned} \phi_{i, n} &= \Psi_{1, n} \dots \Psi_{i-1, n} f [t_n, \dots, t_{n-i+1}] = \prod_{i, n-1} (t_n) f [t_n, \dots, t_{n-i+1}] \\ & i \geq 2 \end{aligned}$$

ET:

$$f [t_0, t_1, \dots, t_r] = \frac{f [t_1, \dots, t_r] - f [t_0, \dots, t_{r-1}]}{t_r - t_0}$$

différence divisée d'ordre r au point t_0, t_1, \dots, t_r qui va nous permettre d'écrire le polynôme $P_{r, n}(t)$ représenté par les Quantités $(f_{n-i})_{i=0, r}$ aux points $(t_{n-i})_{i=0, r}$.

II.3.1.1 Schéma prédicteur d'ordre $K = r + 1$

Nous avons vu que cette méthode se traduit par les équations suivantes:

$$P_{n+1} = Y_n + \int_{t_n}^{t_{n+1}} P_{r, n}(t) \quad \text{avec} \quad P_{n+1} = Y_{n+1} \quad (\text{II.6})$$

Exprimé sous forme de différence divisée $P_{r, n}(t)$ devient:

$$P_{r, n}(t) = \sum_{i=1}^{r+1} \prod_{n, i}(t) f [t_n, t_{n-1}, \dots, t_{n-i+1}] \quad (\text{II.7})$$

avec:
$$\begin{cases} \pi_{n,i}(t) = (t-t_n)(t-t_{n-1})\dots(t-t_{n-i+2}) = \prod_{j=0}^{i-2} (t-t_{n-j}) & i \geq 2 \\ \pi_{n,1}(t) = 1 \end{cases}$$

à partir des définitions précédentes (II.7) devient:

$$Pr,n(t) = \sum_{i=1}^{r+1} \frac{\pi_{n,i}(t)}{\pi_{n-1,i}(t_n)} \phi_{i,n} \cdot \frac{\pi_{n,i}(t_{n+1})}{\pi_{n,i}(t_{n+1})}$$

Soit donc:

$$Pr,n(t) = \sum_{i=1}^{r+1} Ci,n(t) \phi_{i,n}^* \quad (II.8)$$

Avec:

$$\begin{cases} Ci,n(t) = \frac{\pi_{n,i}(t)}{\pi_{n,i}(t_{n+1})} \\ \phi_{i,n}^* = \beta_{i,n+1} \phi_{i,n} \end{cases}$$

Portons: (II.8) dans (II.6), on aura le schéma suivant:

$$P_{n+1} = Y_n + \sum_{i=1}^{r+1} \phi_{i,n}^* \int_{t_n}^{t_{n+1}} Ci,n(\theta) d\theta$$

Finalement on obtient:

$$P_{n+1} = Y_n + h_{n+1} \cdot \sum_{i=1}^{r+1} gi,n \cdot \phi_{i,n}^* \quad (II.9)$$

avec
$$h_n \cdot gi,n = \int_{t_n}^{t_{n+1}} Ci,n(\theta) d\theta$$

Nous verrons plus loin, comment calculer les coefficients gi,n .

II,3.1.2 Schéma correcteur d'ordre $K = r + 1$:

Dans ce cas le schéma est le suivant:

$$Y_{n+1} = Y_n + \int_{t_n}^{t_{n+1}} Q_{r,n}(t) dt$$

or: $Q_{r,n}(t) = \sum_{i=1}^r \prod_{n,i}(t) f(t_n, \dots, t_{n=i+1}) + \prod_{r+1,n}(t) f^P(t_{n+1}, \dots, t_{n-r})$

de la même manière que précédemment, on obtient:

$$Q_{r,n}(t) = P_{r,n}(t) + C_{r+1,n}(t) \phi_{r+2,n+1}^P$$

Soit donc:

$$Y_{n+1} = Y_{n+1} + h_{n+1} \cdot g_{r+1,n} \phi_{r+2,n+1}^P$$

$$h_n \cdot g_{r+1,n} = \int_{t_n}^{t_{n+1}} C_{r+1,n}(\delta) d\delta$$

(II.10)

Il faut maintenant trouver le moyen de calculer $\phi_{i,n+1}^P$ et $\phi_{i,n+1}$ pour compléter le pas.

à partir de:

$$\phi_{i+1,n+1}^P = \prod_{i+1,n+1}(t) f^P[t_{n+1}, \dots, t_{n-i+1}]$$

et en utilisant les définitions (II.5) précédentes, on aura:

$$\phi_{i+1,n+1}^P = \phi_{i,n+1}^P - \phi_{i,n}^* \tag{II.11}$$

de la même façon on aboutit à:

$$\phi_{i+1,n+1} = \phi_{i,n+1} - \phi_{i,n}^* \tag{II.12}$$

Connaissant les Quantités $\phi_{i,n+1}^i = f_{n+1}$ et $\phi_{1,n+1} = f_{n+1}$, on déterminera les autres termes, pour tout $i = 1, \dots, r+1$.

En écrivant que:

$$\phi_{i,n+1}^P = \phi_{i+1,n+1}^P + \phi_{i,n}^*$$

Il nous faut trouver $\phi_{r+2,n+1}^P$ qui découle directement de (II.11)

$$\phi_{r+2,n+1}^P = \phi_{1,n+1}^P - \sum_{i=1}^{r+1} \phi_{i,n}^*$$

La différence entre (II.11) et (II.12), nous amène à:

$$\phi_{i+1,n+1} - \phi_{i+1,n+1}^P \Rightarrow \phi_{i,n+1} = \phi_{i,n+1}^P + (f_{n+1} - \phi_{1,n+1}^P)$$

Il reste maintenant à déterminer la Quantité:

$$\sum_{i=1}^{r+1} \phi_{i,n}^*$$

Pour cela, introduisons une nouvelle variable $\phi_{i,n+1}^e$ différence divisée de $P_{r+1,n}(t)$ aux instants t_{n+1}, t_n, \dots utilisant f_{n+1}^P à t_{n+1} , ce qui va nous permettre de passer des:

$$\phi_{i,n+1}^* \text{ aux } \phi_{i,n+1}^e \text{ au moyen de la relation}$$

$$\phi_{i+1,n+1}^e = \phi_{i,n+1}^e - \phi_{i,n}^*$$

Homologue de la relation de (II.11) qui devient:

$$\phi_{i,n+1}^e = \phi_{i+1,n+1}^* + \phi_{i,n}^* \quad i=r+1, r, \dots, 1$$

comme $P_{r+1,n}(t)$ est un polynôme de degrés $\leq r$ soit donc:

$$\phi_{r+2,n+1}^e = 0$$

On obtient alors à partir de (Z):

$$\phi_{i,n+1}^e = \sum_{i=1}^{r+1} \phi_{i,n}^*$$

Finalemment on obtient:

$$\left\{ \begin{array}{l} \phi_{r+2,n+1}^P = f_{n+1}^P - \phi_{1,N+1}^e \\ \phi_{i,n+1} = \phi_{i,n+1}^e + (f_{n+1} - \phi_{1,n+1}^e) \end{array} \right.$$

Les différentes étapes que nous venons de décrire seront résumées dans le paragraphe suivant.

II.3.1.. Expression du schéma PKECKE et résumé des opérations à une ETAPE:

En utilisant les nouvelles variables pour exprimer le schéma et en posant $K = r + 1$, on obtient:

CALCUL DES COEFFICIENTS: $g_{i,n} \quad i = 1, 2, \dots, K + 1$

P: (PREDICTION):

$$\phi_{i,n}^* = \beta_{i,n+1} \cdot \phi_{i,n} \quad i = 1, 2, \dots, K$$

$$P_{n+1} = Y_n + h_{n+1} \cdot \sum_{i=1}^K g_{i,n} \cdot \phi_{i,n}^*$$

$$\phi_{k+1,n+1}^e = 0$$

$$\phi_{i,n+1}^e = \phi_{i+1,n+1}^e + \phi_{i,n}^* \quad i = K, K-1, \dots, 1$$

E: EVALUATION:

$$f_{n+j}^P = f(t_{n+1}, P_{n+1})$$

C: (CORRECTION):

$$Y_{n+1} = P_{n+1} + h_{n+1} g_{k,n} (f_{n+1} - \phi_{1,n+1}^e)$$

E: (EVALUATION) : $f_{n+1} = f(t_{n+1}, y_{n+1})$

$$\phi_{k+1,n+1} = f_{n+1} - \phi_{1,n+1}^e$$

$$\phi_{i,n+1} = \phi_{i,n+1}^e + \phi_{k+1,n+1} \quad i = k, k-1, \dots, 1$$

On remarque que les calculs sont réalisés de manière à ce que les Quantités:

$\phi_{i,n}; \phi_{i,n}^*; \phi_{i,n+1}^e; \phi_{i,n+1}$ puissent être successivement stockées dans le même emplacement mémoire, ce qui permet une optimisation de l'espace mémoire.

II.3.2: CALCUL DES COEFFICIENTS ($g_{i,n}$) $i \geq 1$:

Nous avons établi (Relation (II.9) ; (II.10)) que:

$$h_{n+1} g_{i,n} = \int_{t_n}^{t_{n+1}} C_{i,n}(\delta) d\delta \quad \text{avec} \quad C_{i,n}(t) = \frac{\prod_{n,i}(t)}{\prod_{n,i}(t_{n+1})}$$

Soit donc:

$$g_{i,n} = \frac{1}{h_{n+1}} \int_{t_n}^{t_{n+1}} C_{i,n}(\delta) d\delta$$

En posant:

$${}^{(1)} C_{i,n}(t) = \int_{t_n}^{t_{n+1}} C_{i,n}(\delta) d\delta \quad (II.13)$$

$$\text{On a:} \quad g_{i,n} = \frac{1}{h_{n+1}} {}^{(1)} C_{i,n}(t_{n+1})$$

ET:

$$C_{i,n}^{(q)}(t_n + 1) = \int_{t_n}^{t_{n+1}} C_{i,n}^{(q-1)}(\delta) d\delta \quad (II.14)$$

de là on tire:

$$g_{i,q} = (q-1)! \int_{t_n}^{t_{n+1}} C_{i,n}^{(q)}(\delta) \frac{1}{h^{q+1}} d\delta \quad (II.15)$$

Il suffit de voir que pour $q = 1$, on retombe sur (II.14), alors la relation de récurrence suivante est établie dans [3].

$$g_{i,q} = g_{i-1,q} - \psi_{i-1,n+1} g_{i-1,q+1} \text{ avec } \psi_{i-1,n+1} = \frac{h^{n+1}}{\psi_{i-1,n+1}} \quad (II.1)$$

Nous allons d'abord calculer $g_{1,q}$ et $g_{2,q}$ qui sont indépendants du pas choisi.

D'après (II.15), on aura:

$$g_{i,q} = (q-1)! \int_{t_n}^{t_{n+1}} C_{i,n}^{(q)}(\delta) \frac{1}{h^{q+1}} d\delta$$

$$\text{OR: } C_{1,n}^{(q)}(t) = 1 \implies C_{1,n}^{(q)}(t_n + 1) = (t_n + 1 - t_n)^q \frac{1}{q!}$$

en reportant dans (II.15), on obtient:

$$g_{1,q} = \frac{1}{q}$$

pour calculer $g_{2,q}$, on reporte dans (II.16) en remarquant que:

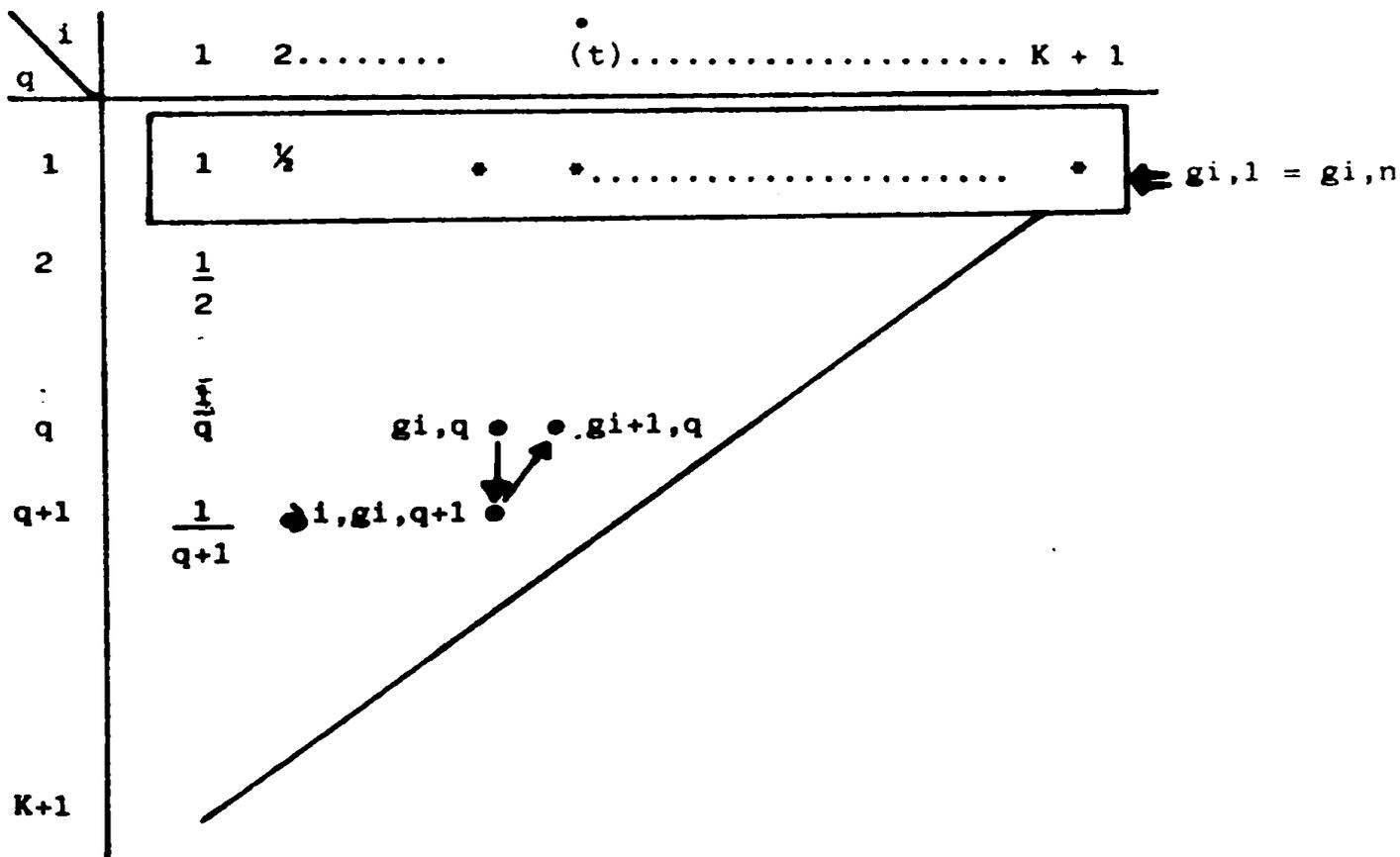
$$\psi_{1,n+1} = 1 \implies g_{2,q} = \frac{1}{q(q+1)}$$

Comme coefficients $g_{i,q}$, on aura donc:

$$g_{i,q} \begin{cases} \frac{1}{q} & i = 1 \\ \frac{1}{(q+1)q} & i = 2 \\ g_{i-1,q} - \frac{1}{q} g_{i-1,q+1} & i \geq 3 \end{cases}$$

D'après (II.14), nous n'aurons besoin que des $g_{i,1}$ ($g_{i,n} = g_{i,1}$).

En disposant les coefficients $g_{i,q}$ sous forme de tableau, on a la règle de construction suivante:



ALGORITHME DE CALCUL DES COEFFICIENTS:

Introduisons deux vecteurs de coefficients utiles pour la suite $[3], [1]$:

$$g_{i,n} = \frac{h_n}{\psi_{i,n}} \quad \text{et} \quad G_{i,n} = (i-1)! h_n^{i-1} / \prod_{j=1}^{i-1} (t_{j,n}) \quad i \geq 1$$

Il s'agit donc de construire:

$$(\Psi_{i,n})_{i=1,K} ; (\Theta_{i,n})_{i=1,K} ; (\beta_{i,n})_{i=1,K} ; (\sigma_{i,n})_{i=1,K+1} ; (g_{i,n})_{i=1,K+1}$$

Avant de décrire les algorithmes permettant la construction de tels coefficients, on calcule leur valeur lorsque le pas est constant depuis quelques étapes, puisque la formulation choisie (formulation de KROGH) l'a justement été pour réduire ces calculs.

On suppose que $h_n = h_{n-1} = \dots = h_{n+1} = h$

ou n est un entier supérieur à 1 est inférieur ou égal à $K+1$, dans cette situation on a:

$$\Psi_{i,n} = ih_n ; \Theta_{i,n} = 1/i ; \beta_{i,n} = 1 \quad \text{Pour } i = 1, \dots, n$$

ET:

$$\sigma_{i,n} = 1 \quad \text{Pour } i = 1, \dots, n+1$$

La relations (II.16) devient:

$$g_{i,q} = g_{i-1,q} - \frac{1}{i-1} g_{i-1,q+1} \quad \text{Pour } i = 1, \dots, n \quad \text{et } q \geq 1$$

Alors l'algorithme devient comme suit:

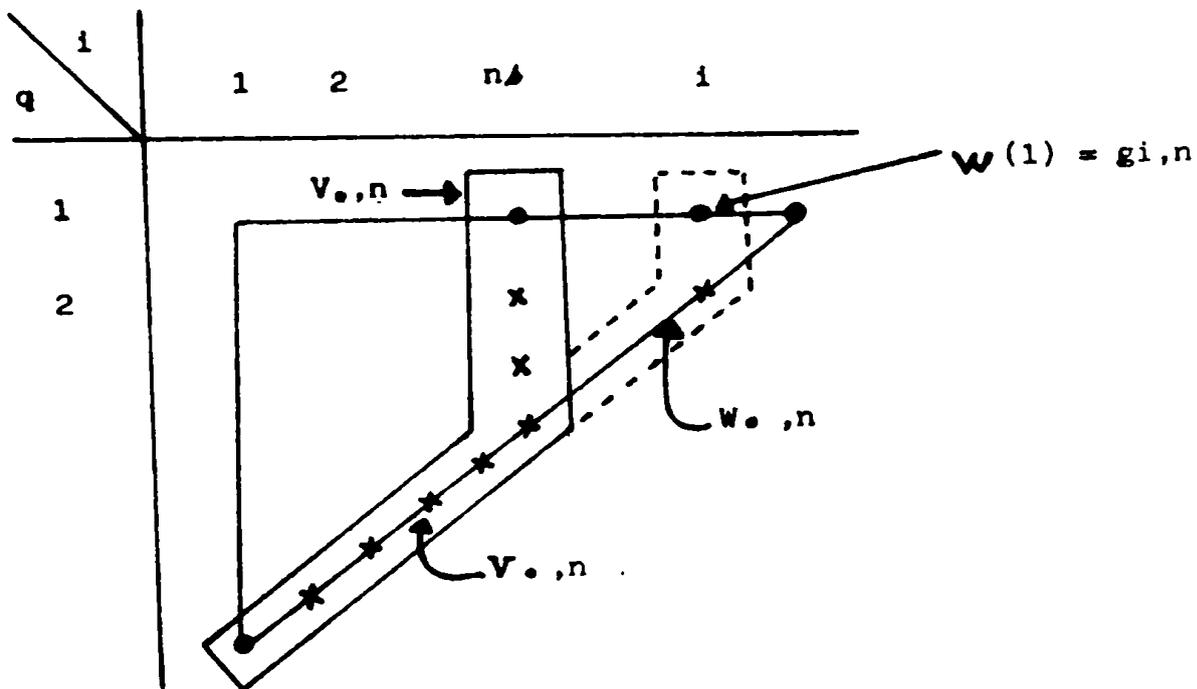
$\beta_{i,n} = 1$ $\sigma_{i,n} = 1$ $\Psi_{i,n} = ih_n$ $\Theta_{i,n} = \frac{1}{i}$

$\beta_{i+1,n} = \beta_{i,n} - \frac{\Psi_{i,n}}{\Psi_{i,n-1}}$ $\sigma_{i+1,n} = \sigma_{i,n} \Theta_{i,n}$ $\Psi_{i+1,n} = \Psi_{i,n-1} + h_n$ $\Theta_{i+1,n} = h_n / \Psi_{i+1,n}$
--

Dans la construction du tableau $(g_{i,q})$, il nous reste à générer les colonnes de $i = ns + 1$ jusqu'à $i = K + 1$, il ne sera donc pas nécessaire de stocker tout le tableau, mais seulement sa bordure (elle restera invariante si le prochain pas est égal au précédent). Cette bordure sera stockée dans un vecteur $V_{\bullet,n}$ de la manière suivante:

$$V_{q,n} = g_{n\Delta, q} \quad \text{Pour } q = 1, \dots, K+1 - n\Delta \quad \text{"PARTIE VERTICALE"}$$

$$V_{q,n} = g_{k+1 - n, q} \quad \text{Pour } q = K+1 - n\Delta, \dots, K \quad \text{"PARTIE OBLIQUE"}$$



Le vecteur W est un vecteur de travail de même dimension que $V_{\bullet,n}$ et qui permet de calculer $g_{i,q}$ de $i = ns + 1, \dots, K + 1$ à partir des valeurs stockées dans W par l'algorithme.

Pour: $q = 1, K$ Faire $W_q = V_{q,n} - 1$

Pour: $i = n\Delta + 1, K + 1$

Faire: Pour: $q = K + 2 + i, 1, - 1$

Faire $W_q = W_q - \delta_{i-1, n} \cdot W_{q+1}$

Quatre (4) situations peuvent se présenter:

- * si l'ordre et le pas sont conservés, alors la bordure du vecteur $V_{\bullet, n}$ correspond au vecteur \underline{W} obtenu pour $i = n\Delta + 1$
- * si le pas a été conservé mais l'ordre a augmenté d'une Unité, on doit décaler la partie oblique par l'algorithme:

$$K = K + 1 \quad (\text{Augmentation d'1 Unité})$$

$$V_{k,n} = 1/K (k+1) \quad \left(\text{homologue à } g_{2,q} = \frac{1}{q(q+1)} \right)$$

Pour: $q = K - 1, K + 2 - n\Delta, - 1$

On a: $V_{q,n} = V_{q,n-1} - \frac{1}{K+1-q} \cdot V_{q+1,n}$

- * si le pas est resté constant et si l'ordre a diminué d'une unité, alors on conserve la partie oblique du vecteur \underline{V} .
- * le pas est changé alors $V_{\bullet, n} = g_{1,q}$

II.4 CONTROLE DE L'ORDRE ET DU PAS:

Le contrôle du pas et de l'ordre sont nécessaires pour permettre au schéma de suivre la régularité de la solution.

II.4.1 Valeurs approchées des erreurs de consistance:

(dans la formulation de KRÔGH)

Il n'existe pas actuellement de méthode pour contrôler l'ordre et le pas à partir d'une expression de l'erreur globale. Cependant, une méthode efficace de choix du pas, consiste à rendre l'erreur de troncature locale $|E_n| \leq \text{EPS}$ sur l'intervalle d'intégration.

D'autre part, pour choisir l'ordre, il est nécessaire de connaître des estimations des erreurs locales (troncature) à l'ordre courant \underline{K} et aux ordres $(K + 1)$ et $(K - 1)$.

On note respectivement ces erreurs \mathcal{E}_n , \mathcal{E}_n^+ , \mathcal{E}_n^- , on montre dans [3] les résultats suivants:

$$\mathcal{E}_n(K) = h_n (g_{K+1,1} - g_{K,1}) \cdot \phi_{K+1,n}^P$$

$$\mathcal{E}_n^-(K-1) = h_n (g_{K,1} - g_{K-1,1}) \cdot \phi_{K,n}^P$$

$$\mathcal{E}_n^+(K+1) = h_n \delta_{K+1}^* [\nabla_{f_{n+1}}^K - \nabla_{f_n}^K]$$

Où: $\nabla^k f_n$ est la différence divisée d'ordre \underline{K} à l'instant $\underline{t_n}$

Lorsque le pas est constant, on a:

$$g_{i,1} - g_{j,1} = \delta_j^*$$

Soit donc:

$$ERK = h_n \cdot \sigma_{K+1} \cdot \delta_K^* \cdot \phi_{K+1,n}^P$$

$$ERKM1 = h_n \cdot \sigma_K \cdot \delta_{K-1}^* \cdot \phi_{K,n}^P$$

Le principe est donc d'accepter ou de refuser le pas si $|\mathcal{E}_n| \leq EPS$
nous avons donc deux possibilités:

- le pas est un échec ($|\mathcal{E}_n| > EPS$), il faut donc savoir quelle erreur on peut avoir si on répète l'intégration avec un autre pas.
- le pas est pris ($|\mathcal{E}_n| < EPS$), et là aussi il faut savoir quelle erreur on obtient avec un autre pas à l'étape $\underline{t_{n+1}}$ avec un pas $\underline{h_{n+1}}$ et un ordre \underline{K}

L'idée émise par SCHAMPINE et GORDON [3] est la suivante:

$$\varepsilon_{n+1}(K) = h_{n+1} \cdot (g_{k+1,1} - g_{k,1}) \cdot \phi_{K+1, n+1}^P$$

Avec:

$$\phi_{K+1, n+1}^P = \Psi_{1, n+1} \cdot \dots \cdot \Psi_{K, n+1} f^P(t_{n+1}, \dots, t_{n+1} - k)$$

Si: $h_{n+1} = rh$ et $h_n = h$ on aura:

$$\begin{aligned} \phi_{K+1, n+1}^P &= (rh) (2rh) \dots (krh) f^P [t_{n+1}, \dots, t_{n+1} - k] \\ &= r^K \cdot \sigma_{K+1, n+1} \phi_{K+1, n+1}^P \end{aligned}$$

Or: quant le pas est constant on a $g_{k+1,1} - g_{k,1} = \sigma_K^*$

$$\Rightarrow \varepsilon_{n+1}(K) = r^{k+1} h \sigma_K^* \sigma_{K+1, n+1} \phi_{K+1, n+1}^P$$

de là l'erreur à l'étape t_{n+1} devient:

$$\varepsilon_{n+1}(K) = ERK \cdot r^{k+1} \text{ avec } ERK = |h_n \sigma_{K+1, n+1} \sigma_K^* \phi_{K+1, n+1}^P|$$

Il en serait de même pour le cas où il y a ECHEC au pas.

II.4.2 ALGORITHME EN PHASE NORMALE DE FONCTIONNEMENT:

Après une phase de démarrage décrite au paragraphe suivant, l'intégration atteint un régime de croisière, où le pas et l'ordre sont ajustés en fonction des variations de la solution.

* Contrôle du pas:

On a vu qu'un choix efficace du pas consiste pour une valeur de tolérance EPS fixée à imposer la condition suivante pour accepter l'étape en t_n :

$$|\varepsilon_n| \leq \text{EPS} \quad (\text{ou } |\text{ERK}| \leq \text{EPS} \text{ si le pas est constant})$$

on recherche ensuite un pas $h_{n+1} = r h_n$ pour l'étape suivante, tel que:

$$\varepsilon_{n+1} = r^{k+1} \text{ERK} \leq 0.5 \text{EPS} \Rightarrow r = \left(\frac{0.5 \text{EPS}}{\text{ERK}} \right)^{\frac{1}{k+1}}$$

Ce principe admis, il faut maintenant en préciser les limites d'application. En effet, pour assurer la stabilité du schéma lorsque l'ordre est supérieur à 1, on impose un maximum au rapport h_{n+1}/h_n , pratiquement le pas sera au plus doublé d'une étape à l'autre.

D'autre part, les séquences de pas constant étant particulièrement recherchées pour l'allègement qu'elles engendrent dans le calcul des coefficients, on augmentera le pas que lorsqu'il est possible de le doubler.

Soit donc: $0.5 \text{EPS} \geq 2^{k+1} \text{ERK}$

Par contre, il n'y a aucune restriction sur les réductions du pas, sauf celle qu'impose le calculateur ($r \geq$ à la précision de la machine).

* Traitement d'un échec:

Si $|\varepsilon_n| > \text{EPS}$ (ou $\text{ERK} > \text{EPS}$ pour un pas constant), on restitue les informations et on procède comme suit:

- si c'est le 1er ou le 2ème ECHEC, le pas est réduit de moitié et l'ordre conservé;
- si c'est le 3ème ECHEC, le pas est réduit de moitié et l'ordre est réduit à 1.

* Contrôle de l'ordre:

L'ordre étant limité à 12

il nous faut trouver un ordre pour lequel:

$$|\varepsilon_n| \leq \text{MIN} (|\varepsilon_n^-|, |\varepsilon_n^+|)$$

En fait, on recherche localement le meilleur ordre à partir d'un ordre donné à plus ou moins une unité, mais rien n'assure qu'un ordre supérieur ne soit pas meilleur. Cette considération nous incite à favoriser la montée en ordre qui en général accroît la précision; mais d'un autre côté, on doit éviter de passer trop facilement sur des "ACCIDENTS" de la solution (point anguleux, asymptote verticale... etc), car la suite des calculs en serait très altérée.

De plus, lorsque l'ordre et le pas augmentent rapidement (surtout en phase de démarrage), il vaut mieux parfois réduire l'ordre pour partir sur des valeurs très fiables.

On est ainsi devant le dilemme suivant:

- * la montée en ordre est favorisée et les résultats obtenus sont très précis et très efficaces dans le cas régulier.
- * pour mieux suivre le cas à "ACCIDENTS", la descente en ordre est souhaitée, et cela devient souvent une politique très coûteuse.

Partant de ce raisonnement, on ne peut changer l'ordre que si l'erreur prédite est réduite; à chaque étape, on considère alors la réduction de l'ordre $K \rightarrow (K-1)$.

$$\text{Si } K > 2 \quad |\varepsilon_n^-| < |\varepsilon_n|$$

$$\text{Si } K = 2 \quad |\varepsilon_n^-| < 0.5 |\varepsilon_n|$$

Dans le cas où il n'y a pas d'ECHEC et le pas est constant depuis K étape, on possède alors une estimation de $|\varepsilon_n^+|$

$$|\varepsilon_n^-| < \text{MIN} (|\varepsilon_n|, |\varepsilon_n^+|)$$

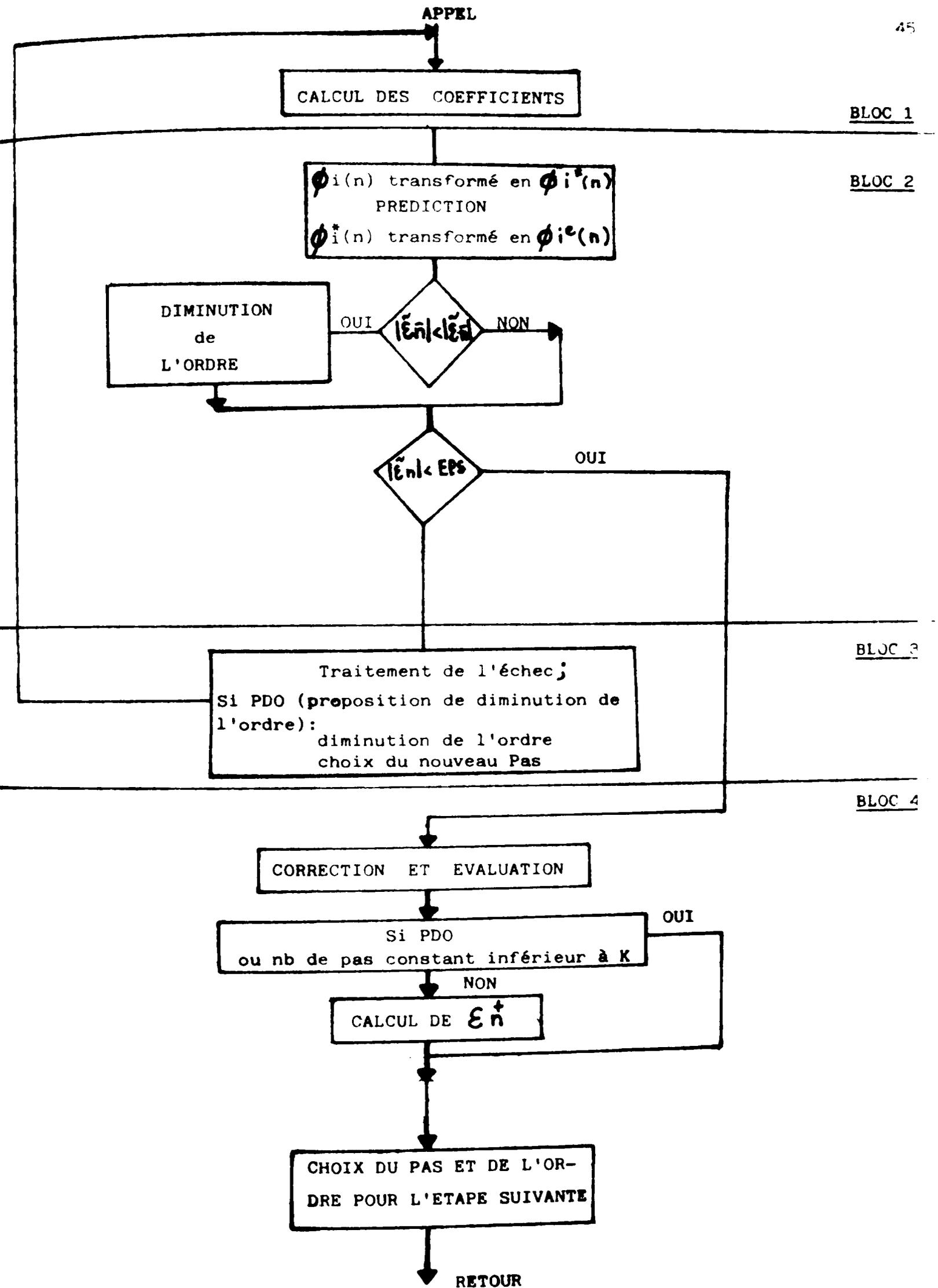
- Augmentation de l'ordre:

$$\text{Si } |\varepsilon_n^+| < |\varepsilon_n| < |\varepsilon_n^-| \quad 1 \leq K \leq 12$$

$$\text{Si } |\varepsilon_n^+| < 0,5 |\varepsilon_n| \quad K = 1$$

* ALGORITHME:

L'algorithme de la phase normale de fonctionnement est alors le suivant:



II.4.3 ALGORITHME EN PHASE DE DEMARRAGE - CHOIX DU 1er PAS:

Le schéma est auto-démarrant si on choisit à la première étape l'ordre 1 (on a besoin que de la valeur initiale y₀ pour démarrer l'intégration). Dans ce cas, il faut choisir un pas h₀ tel que:

$$\frac{h_0^2}{2} \left| Y^{(2)}(\Delta_1) \right| \leq \text{EPS}/2 \quad \text{où } \Delta_1 \in [t_0, t_1]$$

en démarrant à l'ordre 2, on aurait à choisir h₀ tel que:

$$h_0^3 \left| Y^{(3)}(\Delta) \right| / 12 \leq \text{EPS}/2 \quad \text{où } \Delta \in [t_0, t_1]$$

Ainsi, dans un cas on aura choisi:

$$h_0 = \left(\text{EPS} / \left| Y^{(2)}(\Delta_1) \right| \right)^{\frac{1}{2}}$$

et dans l'autre cas:

$$h_0 = \left(6 \text{EPS} / \left| Y^{(3)}(\Delta_2) \right| \right)^{\frac{1}{3}}$$

Si on suppose que $\left| Y^{(2)}(\Delta_1) \right|$ et $\left| Y^{(3)}(\Delta_2) \right|$ sont tous deux non nulles et de même ordre de grandeur, on constate le gain réalisé sur la taille du 1er pas en démarrant à l'ordre 2 surtout pour EPS très petit.

Pour pouvoir démarrer à l'ordre 2, on choisit la méthode P1 (EC₂)²E qui possède la même erreur de consistance que le schéma correcteur, il nous reste la difficulté de l'estimation de l'erreur du schéma:

Celle-ci s'exprime en fonction de la dérivée d'ordre 3 de la solution.

Pour cela, on compare les résultats du schéma à un autre du même ordre; le schéma du point-milieu défini par [4]

$$Y_{n+1} = Y_n + h_n \left(f\left(t_n + \frac{h_n}{2}, y_n + h_n \frac{f_n}{2}\right) \right)$$

On note:

(1)

Y_{n+1} et \tilde{Y}_{n+1} les solutions après la 1ère et la 2ème correction

du schéma P1 (EC₂)²E et par \tilde{y}_{n+1} la solution du schéma du point-milieu dont l'erreur de consistance est donnée en [4]

$$\varepsilon_n = (y_{n+1}^{(1)} + y_{n+1} - 2\tilde{y}_{n+1}) / 6 + O(h_n^4) \quad (\text{II.17})$$

* Choix du 1er pas:

Il faut maintenant choisir le 1er pas pour pouvoir démarrer l'intégration); c'est-à-dire, la valeur du pas h₀ qui nous donnerait:

$$|\varepsilon_0| \leq \frac{\text{EPS}}{2}$$

On a le résultat suivant établi par (1):

$$\varepsilon_n^{(e)} = \gamma_{k,n}^* h_n^{k+1} y^{(k+1)}(t_n) + O(h_n^{k+2})$$

La relation: $\varepsilon_n = \varepsilon_n^e + O(h_n^{k+2})$ nous donne:

$$\varepsilon_n = \gamma_{k,n}^* h_n^{k+1} |y^{(k+1)}(t_n)| + O(h_n^{k+2})$$

Avec un autre pas h'n constant on aurait:

$$\varepsilon_n^l = \gamma_k^* h_n^{k+1} |y^{(k+1)}(t_n)| + O(h_n^{k+2})$$

En négligeant le terme d'ordre (K + 2), on obtient:

$$h'n = (\delta_{K,n}^* \text{EPS}/2 \delta_{K,n}^* |\epsilon_n|)^{\frac{1}{k+1}} h_n \quad (\text{II.18})$$

Tel que:

$$|\epsilon_n| \leq \frac{\text{EPS}}{2}$$

Finalement, on procède comme suit:

- On estime ϵ_0 par la relation (II.17) pour accepter ou refuser le pas et avoir une estimation correcte de $|Y^{(3)}(t_0)|$.
- Cette estimation doit permettre de choisir à nouveau un premier pas pour la formule:

$$2 h_0' = \left(\text{EPS}/2 |\epsilon_0| \right)^{\frac{1}{3}} \cdot h_0$$

Cette formule découle directement de (II.18). Or, à l'étape suivante, l'ordre est maintenu à 2 et le pas doublé; on ne doit donc prendre que la moitié de cette valeur ($2 h_0'/2$).

La correction sur le choix du 1er pas aura lieu si h_0 est trop grand ou trop petit. On a donc deux cas.

1er CAS - h_0 est trop grand:

Si avec le pas $2h_0$ il y a échec au pas, cela se traduit par:

$$\left. \begin{aligned} \epsilon_0' &= (2 h_0)^3 \delta_{K,n}^1 Y^{(3)}(t_0) \\ \epsilon_0 &= \delta_{K,n}^1 h_0^3 Y^{(3)}(t_0) \end{aligned} \right\} \Rightarrow \epsilon_0' = (2 h_0)^3 \frac{|\epsilon_0|}{h_0^3} > \text{EPS}$$

D'où: $|\epsilon_0| > \frac{\text{EPS}}{8}$

2ème CAS - ho est trop petit:

Si avec le pas (4 ho), il n'y a pas eu échec, cela se traduit par:

$$\varepsilon'_0 = (4 \text{ ho})^3 \frac{|\varepsilon_0|}{\text{ho}^3} \leq \text{EPS}$$

D'où: $|\varepsilon_0| \left\langle \begin{array}{l} \text{EPS} \\ 64 \end{array} \right.$

OR: $\delta_{K,n}^* = \delta_K^*$ si h reste constant d'où le 1er pas est accepté.

SI: $\frac{\text{EPS}}{64} \leq |\varepsilon_0| \leq \frac{\text{EPS}}{8}$

OR: $|\varepsilon_0| = \frac{\text{ho}^3}{12} Y^{(3)}(t_0)$ en négligeant le terme d'ordre ho⁴

$$\Rightarrow \left(\frac{16}{3} \text{EPS} / |Y^{(3)}(t_0)| \right)^{\frac{1}{3}} \leq \text{ho} \leq \left(\frac{3}{2} \text{EPS} / |Y^{(3)}(t_0)| \right)^{\frac{1}{3}} \quad (\text{II.19})$$

Si on affecte à priori la valeur 1 à $|Y^{(3)}(t_0)|$ alors le 1er pas est:

$$\text{ho} = \left(\frac{3}{2} \text{EPS} \right)^{\frac{1}{3}}$$

REMARQUE:

Si la dérivée d'ordre 3 vient à s'annuler en t₀, le pas serait infini (Relation II.19).

Pour cela, l'utilisateur du programme doit définir une borne supérieure pour le choix du 1er pas.

* DESCRIPTION DE LA PHASE DE DEMARRAGE:

Dans cette phase on double le pas et on augmente l'ordre d'une unité à chaque étape. L'opération se poursuivra jusqu'à ce que l'erreur de consistance que l'on obtiendrait avec le pas constant h (égale au double du pas actuel) et à l'ordre supérieur soit plus grande que $\underline{EPS/2}$.

Cette erreur ξ_n^+ ne peut être calculée que sous l'hypothèse que

$|Y^{(k+1)}(t_n)| \approx |Y^{(k+2)}(t_n)|$. De ce fait, cette valeur ne sera

pas calculée par contre à chaque étape. La diminution de l'ordre est envisagée, elle prend effet dès qu'elle est retenue, c'est-à-dire, dès que:

$$|\xi_{n-1}^-| < |\xi_n^-|$$

Ceci permet d'améliorer la précision des calculs.

CHAPITRE III: COMPORTEMENT EFFECTIF DES METHODES:

Les deux méthodes décrites dans les CHAPITRES précédents ont été implantées sur ordinateur, grâce à deux programmes rédigés en FORTRAN.

Le programme contenant la méthode de RUNGE-KUTTA à pas variable a pour nom SIMUL. Il est appelé à chaque pas avec un pas proposé, et il rend la main avec de nouvelles valeurs pour le pas suivant.

Par contre, la méthode P_{REC}E a été implantée dans un sous-programme appelé ITER. Après chaque pas d'intégration, il donne de nouvelles valeurs de l'ordre et du pas pour la prochaine intégration (pas et ordre variables).

L'évaluation de la dérivée est très coûteuse, elle constitue une grande partie du travail effectué lors d'une intégration. De ce fait, le coût d'une méthode est estimé grâce au nombre d'évaluations de la dérivée lors de l'intégration. Dans notre cas, nous avons au minimum deux (2) évaluations par pas d'intégration.

1) EXEMPLES TRAITES:

Dans les tableaux suivants, on donne pour chaque valeur de EPS fixe (erreur de tolérance), le nombre d'évaluations nécessaires pour l'intégration, la solution, et la valeur maximale de l'erreur absolue ou relative commise durant l'intégration.

a) Comportement de la méthode P_{REC}E

a.1 On désire intégrer le système suivant:

$$(1) \quad \begin{cases} Y_1'(t) = Y_2(t) + Y_3(t) & Y_1(0) = 0.0 \\ Y_2'(t) = -Y_1(t) \cdot Y_3(t) & \text{avec } Y_2(0) = 1.0 \\ Y_3'(t) = -0.51 Y_1(t) Y_2(t) & Y_3(0) = 1.0 \end{cases}$$

Sur une période T dont le quart de la période est:

$\frac{T}{4} = 1,862640802332739$ avec une précision EPS = 10^{-5}
 les solutions du système (1) sont:

$$\begin{cases} Y_1(t) = 0 \\ Y_2(t) = 1 \\ Y_3(t) = 1 \end{cases}$$

L'intégration sur calculateur du système (1) sur 4,40, 400 et 4000 pas fournit les résultats suivants:

Nbre. de Pas		Solution calculée	Erreurs commises	Nbre. d'évaluations
Pour 4,40 et 400 Pas	$Y_1(t) =$ $Y_2(t) =$ $Y_3(t) =$	$- 0.5120307 E - 04$ $0.1001184 E + 01$ $0.1000293 E + 01$	0.51210^{-4} 11810^{-5} 2910^{-5}	113
Pour 4000 Pas	$Y_1(t) =$ $Y_2(t) =$ $Y_3(t) =$	$0.1802303 E + 07$ $0.1000002 E + 01$ $0.1000000 E + 01$	$0.18 10^{-7}$ 210^{-6} $< 10^{-6}$	881

a.2 Intégration du système (2) sur [0.5 - 1[avec contrôle d'erreur relative:

$$(2) \begin{cases} Y'(t) = 2 Y(t) / (1-t) \\ \text{avec } Y(0.5) = 4 \end{cases}$$

la solution du système (2) est $Y(t) = 1/(1-t)^2$ elle admet donc une asymptote verticale en $t = 1$.

Pour $EPS = 10^{-5}$ l'intégration a progressé jusqu'à $t = 0.9999999999995035$

avec comme dernier Pas réussi $hn = 0.9443 E - 13$

Les résultats obtenus sont:

t	Solution calculée	Erreurs commises	Nombre d'évaluations
0.5	4	0	0
0.6	0.6244567 E + 01	0.0054	30
0.7	0.1103490 E + 02	0.075	42
0.8	0.2498775 E + 02	0.012	52
0.9	0.9998045 E + 02	0.019	62
1	EHEC AU PAS AVEC T = 0.1000000 E + 01		

a.3 Intégrons le système (3) sur [0.4] avec contrôle d'erreur absolue:

$$(3) \begin{cases} Y'(t) = 1 - 2(t-4)(y-t) \\ Y(0) = 0 + 10 e^{-16} \end{cases}$$

La solution est alors $Y(t) = t + 10 \exp(-(t-4)^2)$ et $Y(4) = 14$

Les résultats fournis par l'intégration sont:

EPS	Nombre d'évaluations	Solution calculée	Erreur Absolue
10^{-3}	42	0.8475248 E + 01	5.525
10^{-5}	87	0.1318956 E + 02	0.82
10^{-7}	149	0.1388510 E + 02	0.12
10^{-16}	506	0.1400049 E + 02	$0.49 \cdot 10^{-3}$

Si on avait annulé la condition initiale (i.e. $Y(0) = 0$), la solution serait $Y(t) = t$, en perturbant cette dernière de $\epsilon = 10 e^{16} \approx 10^{-6}$, la solution est alors $Y(t) = t + 10 \exp [-(t-4)^2]$, soit donc à $t = 4$ on passe de $Y(4) = 4$ à $Y\epsilon(4) = 14$, le système (3) est donc un système instable.

D'après le tableau précédent, la solution du système instable ne peut être correctement suivie que si l'erreur de tolérance EPS est suffisante pour prendre en compte la valeur initiale. De ce fait, la précision au début de l'intégration est primordiale.

a.4 Intégration sur $[0 - 0.810^{-3}]$ avec contrôle d'erreur relative du système (4) suivant:

$$(4) \begin{cases} Y'(t) = -10^5 Y(t) \\ Y(0) = 1 \end{cases} \quad \begin{array}{l} \text{solution } Y(t) = e^{-10^5 t} \\ Y(0.810^{-3}) = 0.0000 \end{array}$$

EPS	Nombre d'évaluations	Solution calculée
10^{-3}	468	0.1814498 E - 34
10^{-5}	685	0.1760527 E - 34

Le système (4) est raide (STIFF), la solution décroît pendant une phase transitoire très rapide jusqu'à la valeur 0.4510^{-4} , puis elle atteint son régime permanent avec une valeur pratiquement nulle.

Bien que le programme ne résoud pas les systèmes STIFF, la résolution du système différentiel précédent revient à ce que la raideur (STIFFNESS) de ce dernier ne dure qu'un très court instant pendant lequel la précision est plus importante que la stabilité pour la poursuite de l'intégration.

b) Comportement de la méthode de RUNGE-KUTTA à pas variable comparée avec la méthode à pas constant:

Dans ce paragraphe, nous allons comparer les deux méthodes de RUNGE-KUTTA: l'une à pas constant, et l'autre, à pas variable, tant du point de vue coût calcul que précision des résultats en intégrant les systèmes suivants:

COMPORTEMENT EFFECTIF DE LA METHODE COMPAREE A CELLE DE RUNGE-KUTTA

CLASSIQUE:

Résolution de $dy/dt = - Y$ avec $Y(0) = 1$

En utilisant le 1 contrôle d'erreur

<u>hn</u>	<u>Y(n,1)</u>	<u>Z1</u>	<u>I</u>	<u>Neval</u>
0.2208679 E + 00	0.4488230 E - 04	0.4483642 E - 04	0.1001249 E + 02	230

ERREUR ABSOLUE = 0.458813929 E - 07

Résolution de $dy/dt = - Y$ avec $Y(0) = 1$

En utilisant le 2 contrôle d'erreur

<u>hn</u>	<u>Y(n,1)</u>	<u>Z1</u>	<u>I</u>	<u>Neval</u>
0.1175846 E + 01	0.4071327 E - 04	0.2598865 E - 04	0.1055785 E + 02	65

ERREUR ABSOLUE = 0.147248210 E - 04

Résolution de $dy/dt = - Y$ avec $Y(0) = 1$

En utilisant le 3 contrôle d'erreur

<u>hn</u>	<u>Y(n,1)</u>	<u>Z1</u>	<u>I</u>	<u>Neval</u>
0.9999487 E - 01	0.4410748 E - 04	0.4410470 E - 04	0.1002894 E + 02	550

ERREUR ABSOLUE = 0.277697776 E - 08

Résolution avec la méthode de RUNGE-KUTTA classique:

<u>hn</u>	<u>Y(1)</u>	<u>Z1</u>	<u>T</u>	<u>Neval</u>
0.1000000 E + 00	0.4540034 E - 04	0.4539984 E - 04	0.1000000 E + 02	400
<u>ERREUR ABSOLUE</u> = 0.4973594 E - 09				

L'importance du 1er contrôle d'erreur est mise en évidence sur les solutions très régulières, il permet d'approcher la solution avec une grande précision pratiquement la même que celle (précision) obtenue avec la méthode de RUNGE-KUTTA classique. Mais l'avantage réside dans le moindre coût où nous avons 230 contre 400 évaluations. La valeur de hn dans le premier contrôle d'erreur est plus que le double de celle de hn obtenue avec la méthode de RUNGE-KUTTA classique, donc ce contrôle d'erreur adapte la méthode à la régularité de la solution.

Résolution de $dy/dt = -10 * Y$ avec $Y(0) = 1$

En utilisant le 1er contrôle d'erreur

<u>hn</u>	<u>Y(n,1)</u>	<u>Z1</u>	<u>T</u>	<u>Neval</u>
0.2650886 E + 00	0.3832454 E - 04	0.4217010 E - 18	0.4230999 E + 01	240
<u>ERREUR ABSOLUE</u> = 0.383245388 E - 04				

Résolution de $dy/dt = -10 * Y$ avec $Y(0) = 1$

En utilisant le 2ème contrôle d'erreur

<u>hn</u>	<u>Y(n,1)</u>	<u>Z1</u>	<u>T</u>	<u>Neval</u>
Echec au contrôle du pas. hn = 0.2132				

Résolution de $dy/dt = - 10 * Y$ avec $Y(0) = 1$

En utilisant le 3ème contrôle d'erreur

<u>hn</u>	<u>Y(n,1)</u>	<u>Z1</u>	<u>T</u>	<u>Neval</u>
0.1000000 E + 00	0.9888700 E - 17	0.2774812 E - 17	0.4042595 E + 01	230

ERREUR ABSOLUE = 0.711388810 E - 17

Résolution avec la méthode de RUNGE-KUTTA classique

<u>hn</u>	<u>Y(1)</u>	<u>Z1</u>	<u>T</u>	<u>Neval</u>
0.5000000 E - 01	0.4385062 E - 17	0.4248468 E - 17	0.3999997 E + 01	320

ERREUR ABSOLUE = 0. 1365941 E - 18

Dans ce cas, l'équation est légèrement raide, on remarque alors l'importance du 3ème contrôle d'erreur qui permet d'obtenir un résultat avec une très grande précision pour un coût très faible, soit 230 évaluations de la dérivée contre 320 évaluations avec la méthode de RUNGE-KUTTA classique.

PARTIE B: SIMULATION NUMERIQUE DU COMPORTEMENT DYNAMIQUE D'UN MANIPULATEUR RIGIDE ET ALGORITHME DE COMMANDE ADAPTATIVE:

INTRODUCTION:

Nous nous intéressons au problème de la commande d'un robot manipulateur rigide. Ce problème semble à priori difficile, du fait de la non-linéarité des relations entrées-sorties de ce dernier.

Les stratégies simples de commande peuvent toujours être appliquées tant que la non-linéarité du problème peut être négligée (manipulateur transportant un objet connu à faible vitesse). Dans le cas contraire, l'étude de commandes plus élaborées est justifiée.

Ces commandes se feront en boucle fermée, du fait que la solution par retour d'état est très pratique en raison de l'insensibilité vis-à-vis des perturbations externes qu'elle apporte.

Nous allons simuler le comportement dynamique d'un manipulateur en exploitant un algorithme de commande adaptative proposé par TOMIZUKA et HOROWITZ [1].

Nous montrerons que les résultats obtenus sont la conséquence de l'utilisation de grand-gains, soit donc les bons résultats enregistrés sont plutôt dûs à la robustesse de la commande qu'à l'adaptativité.

Pour pouvoir mener à bien le travail que nous nous sommes proposés d'effectuer, nous avons eu besoin du modèle dynamique du manipulateur.

CHAPITRE I: COMMANDE DYNAMIQUE DE MANIPULATEUR:

La simulation d'une commande dynamique d'un manipulateur exige la résolution des équations du modèle mathématique de ce dernier.

I. Obtention du modèle dynamique d'un robot manipulateur:

Le modèle dynamique des robots manipulateurs permet l'étude de leur comportement sous l'action des couples ou des forces développés par les moteurs, grâce aux moyens de la simulation. La mise en oeuvre d'un algorithme de commande est nécessaire dans ce cadre. Ce modèle peut être déduit à partir des équations de LAGRANGE, qui sont les plus usitées, du fait de leur manipulation aisée.

1. Forme générale du modèle dynamique [2] :

Tout manipulateur rigide est un système mécanique invariant dans le temps, son énergie cinétique est une forme quadratique définie positive par rapport aux vitesses généralisées (vitesse articulaire).

Si $M(\underline{q})$ est la matrice d'une telle forme ou matrice d'inertie du manipulateur est $T(\underline{q}, \dot{\underline{q}})$ l'énergie cinétique alors:

$$T(\underline{q}, \dot{\underline{q}}) = \frac{1}{2} \dot{\underline{q}}^T M(\underline{q}) \dot{\underline{q}} \quad (B.1)$$

en posant $Q = [Q_i]$ $i = 1, N$ vecteur de forces généralisées (couples ou forces s'exerçant sur les articulations du manipulateur exercées par les moteurs, dûs à la gravité, aux frottements, ... etc).

Et en appliquant les équations de LAGRANGE:

$$Q_i = \frac{d}{dt} \left[\frac{\partial T}{\partial \dot{q}_i} \right] - \frac{\partial T}{\partial q_i} \quad i = 1 \text{ à } N$$

à l'équation (B.1), après avoir développé et regroupé les termes, on retrouve la forme générale du modèle dynamique:

$$\underline{\ddot{q}} = \underline{M}(\underline{q}) \underline{\ddot{q}} + \underline{N}(\underline{q}, \underline{\dot{q}})$$

Où:

* $\underline{M}(\underline{q})$: matrice symétrique définie positive et pour raison de structure d'un manipulateur, aucune de ses valeurs propres ne tend vers zéro.

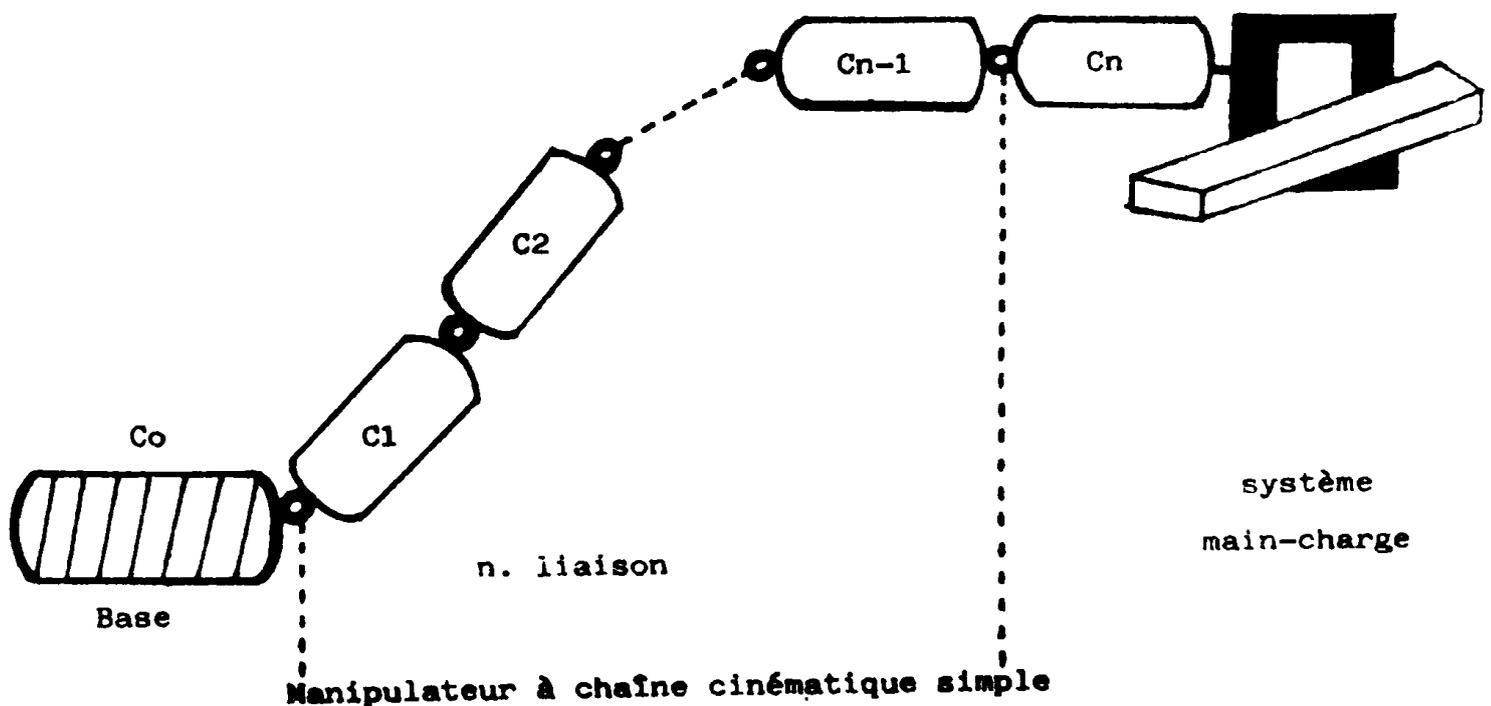
* $\underline{N}(\underline{q}, \underline{\dot{q}}) = \frac{1}{2} \underline{\dot{q}}^T \underline{B} \underline{\dot{q}}$: vecteur des forces de coriolis et centrifuges, à composantes réelles, continu en $\underline{q}, \underline{\dot{q}}$ et bornées uniformément.

REMARQUE:

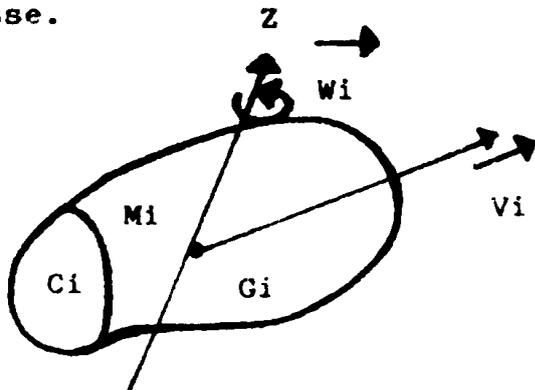
Les coefficients des matrice \underline{M} et \underline{B} dépendent des q_i d'une façon fortement non-linéaire.

2. ALGORITHME [3]:

Le manipulateur est considéré comme un ensemble de corps C_i ou segments articulés (FIGURE 1).



L'Énergie cinétique T_i d'un corps C_i peut être vue comme étant la somme de son énergie cinétique de translation (masse ponctuelle) et d'une énergie cinétique de rotation autour d'un axe Z passant par le centre de masse.



$$T_i = \frac{1}{2} m_i \underline{V}_{Gi}^2 + \underline{w}_i \mathbf{I}^i \underline{w}_i$$

- REMARQUE:
- Les vitesses \underline{V}_{Gi} sont exprimées dans le repère R_i lié au corps C_i , pris par rapport à R_0
 - I^i représente le tenseur d'Inertie.

L'énergie cinétique totale est donc:

$$T = \sum_{i=1}^N \left(\frac{1}{2} m_i \underline{V}_{Gi}^2 + \underline{w}_i \mathbf{I}^i \underline{w}_i \right)$$

Une fois cette énergie définie, l'algorithme est le suivant:

- * Les matrices de transformation des repères (passage de $R_i (O_i, X_i, Y_i, Z_i)$ lié au corps C_i à $R_{i+1} (O_{i+1}, X_{i+1}, Y_{i+1}, Z_{i+1})$ lié au corps C_{i+1}), dans le cas d'une rotation sont:

$$M_i^{i+1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos q_i & -\sin q_i \\ 0 & \sin q_i & \cos q_i \end{pmatrix}$$

à une permutation des lignes et des colonnes près, suivant l'axe de rotation.

* transformation des vitesses de translation lorsque les degrés de liberté sont des rotations:

$$\underline{V}_{O_{i+1}}^{(o)}(R_i) = \underline{V}_{O_i}^{(o)} + \underline{W}_i^{(o)}(R_i) \wedge \vec{O_i O_{i+1}}(R_i)$$

$$\underline{V}_{O_{i+1}}^{(o)}(R_i) = M_i^{(i+1)} \underline{V}_{O_{i+1}}^{(o)}(R_{i+1})$$

* transformation de la vitesse de rotation d'un repère:

$$\underline{W}_i^{(o)}(R_i) = \underline{W}_i^{+i-1}(R_{i-1}) + (M_{i-1})^T \underline{W}_{i-1}^{(o)}(R_{i-1})$$

Où: $\underline{W}_i^{+i-1}(R_{i-1})$ est la vitesse de rotation du repère R_i/R_{i-1} ,
exprimée dans le repère R_{i-1} .

N.B.: Nous avons omis de parler des translations du fait que tous les degrés de liberté de notre manipulateur sont des rotations.

II. COMMANDE ADAPTATIVE:

Nous avons vu que le modèle mathématique du manipulateur est de la forme:

$$M(\underline{X}_p) \ddot{\underline{X}}_p + N(\underline{X}_p, \underline{X}_v) = \underline{Q}(t) \quad (B.2)$$

OU:

\underline{X}_p : vecteur position

\underline{X}_v : vecteur vitesse

Les commandes adaptatives sont utilisées entre-autres pour deux raisons:

- 1) L'évolution de la structure du robot, intervenant lors de la prise d'objet, conduit à une inadéquation du modèle aussi complexe soit-il à la réalité physique.
- 2) L'importance de l'identification approchée de la structure afin de déterminer les caractéristiques essentielles du manipulateur indépendamment de la configuration prise par ce dernier.

Leur principe consiste en l'identification en ligne des matrices M et N . On utilise les matrices identifiées \hat{M} et \hat{N} dans le schéma de commande pour obtenir une commande dite indirecte: il s'agit d'identifier le modèle du système tout en commandant le système.

Malheureusement, la complexité et les difficultés de l'identification ne favorisent pas l'application d'une méthodologie générale.

Cependant, une technique consiste à estimer récursivement (estimation implicite) un vecteur θ de paramètre constant ou lentement variable au moyen d'une relation de la forme:

$$\theta(t) = \theta^T(t) Y(t) \quad (B.3)$$

Ou: $Y(t)$: vecteur de "mesure" connu
 $\Phi^T(t)$: matrice d'observation connue de dimension compatible avec elle de $y(t)$ et $\Theta(t)$.

L'équation (B.3) est obtenue après paramétrisation de l'équation (B.2). TOMIZUKA et HOROWITZ [1] proposèrent en 1980, un schéma de commande adaptative indirecte appelé schéma MRAC (commande adaptative avec modèle de référence).

Le problème est donc de générer une commande $U(X_p(t), X_v(t), t)$ qui permette aux trajectoires $(X_p(t), X_v(t))$ de "suivre" les trajectoires $(X_{pr}(t), X_{vr}(t))$ élaborées à partir d'un modèle de référence d'équation:

$$\ddot{X}_{pr}(t) = U_r(t)$$

III. SCHEMA MRAC CONTINU [1] :

C'est un schéma de commande adaptative avec modèle de référence. Il assure un découplage entre les différents joints d'articulations (les degrés de liberté), et une compensation non-linéaire.

1) Compensation non linéaire et découplage:

En négligeant les forces de frottements difficilement modélisables et de gravité souvent compensées par contre-poids, l'équation (B.2) devient:

$$M(\underline{X}_p) \ddot{\underline{X}}_p + \underline{V}(\underline{X}_p, \underline{X}_v) = \underline{Q}(t) \quad (B.4)$$

La loi de commande permettant d'atteindre cet objectif est:

$$\underline{Q}(t) = M(\underline{X}_p) \underline{U}(t) + \underline{V}(\underline{X}_p, \underline{X}_v) \quad (B.5)$$

Tel que:

$$\begin{cases} \dot{\underline{X}}_p(t) = \underline{X}_v(t) \\ \dot{\underline{X}}_v(t) = \underline{U}(t) \end{cases} \quad (B.6)$$

L'équation (B.6) représente trois double intégrateurs découplés. De ce fait, le premier terme du second membre de l'équation (B.5) sert à découpler les interactions entre les différents degrés de liberté, alors que le second terme sert à compenser le terme $\underline{V}(\underline{X}_p, \underline{X}_v)$ dans l'équation (B.4) du manipulateur (compensation des forces de coriolis et centrifuges).

Pour utiliser la loi de commande (B.5), il faut connaître au préalable, les valeurs de $M(\underline{X}_p)$ et $\underline{V}(\underline{X}_p, \underline{X}_v)$ qui seront calculées et stockées pour chaque valeur de X_p et X_v , ce qui représente un travail très coûteux en temps calcul et place mémoire. Un moyen de contourner la difficulté et d'utiliser une commande adaptative (il s'agit d'ajuster les paramètres dans la loi de commande jusqu'à ce que les dynamiques du manipulateur convergent vers celles décrites par l'équation (B.6), utilisant le schéma MRAC continu qui se résume comme suit:

- MODELE DE REFERENCE

Il est défini par les équations suivantes:

$$\left\{ \begin{array}{l} \dot{\underline{X}}_{pr}(t) = \underline{X}_{vr}(t) \\ \dot{\underline{X}}_{vr}(t) = \underline{U}_r(t) \end{array} \right. \quad (B.7)$$

Il sert surtout au découplage adaptatif, dans ce cas, le couple d'entrée est alors:

$$\underline{Q}(t) = \hat{M}(t) \underline{U}_r(t) + \hat{V}(t, \underline{X}_v) - \underline{F}_p \underline{\xi}_p(t) - \underline{F}_v \underline{\xi}_v(t) \quad (B.8)$$

AVEC:

$\underline{\xi}_p(t) = X_p(t) - X_{pr}(t)$ l'erreur de poursuite en position

$\underline{\xi}_v(t) = X_v(t) - X_{vr}(t)$ l'erreur de poursuite en vitesse

$\hat{M}(t)$ et $\hat{V}(t, \underline{X}_v)$ seront ajustées par un algorithme d'adaptation, \underline{F}_p et \underline{F}_v , deux matrices qui garantissent la stabilité du schéma rebouclé.

L'utilisation d'un ordinateur pour le traitement et le stockage de l'information nous impose une commande numérique, de ce fait $\underline{U}_r(t)$ sera une commande échantillonnée.

IV. COMMANDE ECHANTILLONNEE:

Quand on simule le comportement d'un système physique, on a accès à toutes les grandeurs intervenant dans le modèle du système.

Cet accès aisé aux divers paramètres et variables, est d'ailleurs une source d'erreurs, si on ne prend pas la précaution de bien séparer les variables internes du modèle des variables accessibles aux mesures. l'écriture du modèle sous forme d'équation d'états, et d'équations de mesures permet d'éviter cet écueil, soit donc:

$$\left\{ \begin{array}{l} \dot{\underline{X}}(t) = \underline{F}(\underline{X}(t), \underline{U}(t)) \\ \underline{Y} = \underline{C}\underline{X} \end{array} \right. \text{ avec } \underline{C}=\underline{I} \text{ dans notre cas. Où } \underline{I} \text{ est la ma-} \\ \text{trice identité.}$$

L'équation d'état étant:

$$\dot{X}(t) = f(X(t), U(t))$$

La commande $U(t)$ varie d'une façon discontinue et reste constante pendant la période d'échantillonnage T_E , les mesures ne sont effectuées qu'aux instants $k T_E$ ($k \in \mathbb{N}$), dans ce cas, l'équation de mesure est alors discrète.

$$Y_k = g(X(k T_E))$$

La valeur de la commande entre l'instant $k T_E$ et l'instant $(k+1) T_E$ est calculée à partir des mesures disponibles à l'instant $k T_E$ soit donc:

$$U_k = h(Y_k)$$

h pouvant contenir des termes décrivant un mécanisme adaptatif. Connaissant l'état X_k à l'instant $k T_E$ pour le déterminer à l'instant $(k+1) T_E$, on intègre l'équation différentielle suivante:

$$\left\{ \begin{array}{l} \dot{X}(t) = f(X(t), U(t)) \\ X(k T_E) = X_k \\ t \in [k T_E, (k+1) T_E] \end{array} \right.$$

CHAPITRE II: SIMULATION NUMERIQUE DU COMPORTEMENT DYNAMIQUE D'UN MANIPULATEUR RIGIDE PAR LE BIAIS D'UNE COMMANDE ADAPTATIVE:

Nous allons simuler le comportement dynamique d'un manipulateur rigide à trois degrés de liberté (FIGURE I), transportant trois charges différentes 5 kg, 10kg et 20kg, en utilisant une loi de commande adaptative.

Soit donc:

$$M(\underline{X}_p) \ddot{\underline{X}}_p + \underline{V}(\underline{X}_p, \underline{X}_v) = \underline{Q}(t)$$

Où:

$$\underline{Q}(t) = \hat{M}(t) \underline{U}_r(t) + \hat{V}(t, \underline{X}_v) - \underline{F}_p \underline{\xi}_p(t) - \underline{F}_v \underline{\xi}_v(t)$$

D'où:

$$\ddot{\underline{X}}_p(t) = M^{-1}(\underline{X}_p) \left[\underline{Q}(t) - \underline{V}(\underline{X}_p, \underline{X}_v) \right] \quad (B.9)$$

$$\underline{X}_p = \iint M^{-1}(\underline{X}_p) \left[\underline{Q}(t) - \underline{V}(\underline{X}_p, \underline{X}_v) \right] dt \quad (B.10)$$

Au cours de plusieurs études de simulation, il s'est avéré [1] que le terme non-linéaire $\underline{V}(\underline{X}_p, \underline{X}_v)$ est secondaire, de ce fait, négliger le terme $\hat{V}(t, \underline{X}_v)$ ne cause aucun changement essentiel dans la réponse du manipulateur.

De là, l'équation (B.10) devient:

$$\begin{cases} \underline{X}_p = \iint M^{-1}(\underline{X}_p) \left[\underline{Q}(t) \right] dt \\ \underline{Q}(t) = \hat{M}(t) \underline{U}_r(t) - \underline{F}_p \underline{\xi}_p(t) - \underline{F}_v \underline{\xi}_v(t) \end{cases} \quad (B.11)$$

Le modèle retenu est donc de la forme:

$$M(\underline{X}_p) \ddot{\underline{X}}_p = \underline{Q}(t) \quad (B.12)$$

Où:

$$\ddot{\underline{X}}_p(t) = M^{-1}(\underline{X}_p) \cdot \underline{Q}(t)$$

La transformation canonique suivante permet d'exprimer l'équation (B.12) sous forme d'équation d'état:

En posant:

$$\begin{cases} \dot{X}_p(t) = X_v(t) \\ \dot{X}_v(t) = U(t) \end{cases}$$

On obtient alors le système suivant:

$$\begin{bmatrix} \dot{X}_p \\ \dot{X}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} X_p \\ X_v \end{bmatrix} + \begin{bmatrix} \mathbf{0}_3 \\ \dots \\ \mathbf{M}^{-1}(X_p) \end{bmatrix} \underline{Q}(t)$$

de la même manière, on écrit le modèle de référence sous forme:

$$\begin{bmatrix} \dot{X}_{pr} \\ \dot{X}_{vr} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} X_{pr} \\ X_{vr} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{I}_3 \end{bmatrix} \underline{U}_r(t)$$

I. SYNOPTIQUE DU SCHEMA MRAC CONTINU:

Le schéma synoptique sur lequel nous avons travaillé est extrait de celui proposé par TOMIZUKA et HOROWITZ [1] avec de légères modifications, notamment la suppression de l'algorithme d'adaptation donnant le vecteur $\hat{V}(t, X_v)$.

1) Commande en consigne:

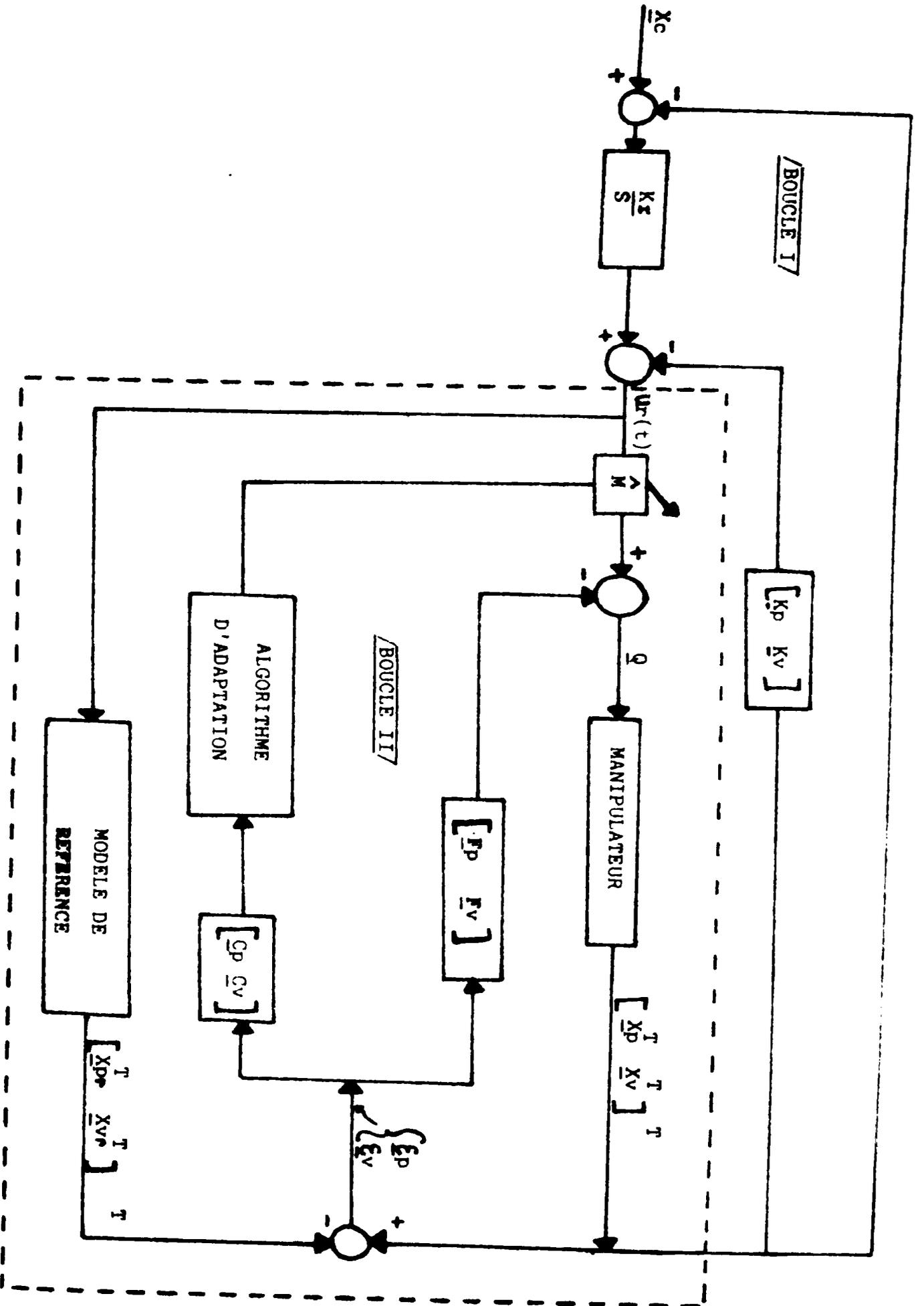
Cette commande est définie par l'équation suivante:

$$U_r(t) = \lambda (X_c - X_p) - \beta X_v$$

Résultat établi dans [4].

Dans notre cas, on a:

$$\begin{aligned} U_r(\delta) &= \frac{K_I}{\delta} (X_c(\delta) - X_p(\delta)) - K_p X_p(\delta) - K_v X_v(\delta) = \ddot{X}_p(\delta) \\ \Rightarrow \delta^3 X_p(\delta) &= K_I (X_c - X_p(\delta)) - K_p \delta X_p(\delta) - K_v \delta^2 X_p(\delta) \end{aligned}$$



SCHEMA SYNOPTIQUE DE LA COMMANDE ADAPTATIVE

N.B. : L'encadre constitue le schéma MRAC continu

Après regroupement, on trouve:

$$X_p(S) = \frac{K_I}{S^3 + K_v S^2 + k_p S + K_I} \quad X_c = \frac{K_I}{(a-s)(s^2 + b s + c)} \quad X_c$$

La constante de temps et le dépassement de la réponse indicielle se réglant par le choix des constantes K_I , K_p , K_v . donc, ces dernières sont déterminées entièrement par la dynamique du ralliement de la consigne.

2) Stabilité du schéma MRAC:

La stabilité du schéma rebouclé (Boucle II) est asymptotique, c'est-à-dire que:

$$\lim_{t \rightarrow \infty} \epsilon_p(t) \rightarrow 0$$

Si la fonction de transfert $G(S)$ définie par:

$$G(S) = \frac{C_p + C_v S}{M S^2 + F_v S + F_p}$$

est strictement réelle positive (résultat établi dans [1]) ceci suppose que la matrice $M(\underline{X}_p)$ reste constante pendant l'adaptation.

Les matrices F_p , F_v , C_p , C_v qui doivent satisfaire la condition de positivité de $G(S)$, sont obtenues à partir du lemme de stabilité de:

YAKUBOVITCH - KALMAN - POPOV

3. Algorithme d'adaptation:

L'algorithme d'adaptation permettant d'ajuster la matrice $\hat{M}(t)$ est donnée par TOMIZUKA et HOROWITZ [1] :

$$\frac{d}{dt} [\hat{m}_{ij}(t)] = - K_{ij} [y_i - x_{vrj}] \quad i = 1, 2, 3 \text{ et } j = 1, 2, 3 \quad (B.13)$$

Avec: $\underline{Y}(t) = C_p \underline{E}_p(t) + C_v \underline{E}_v(t)$ et $K_{ij} > 0$

Tel que:

$$K_{11} = K_{23} = K_{33} = 5 \quad ; \quad K_{42} = K_{46} = 0.01$$

Et $K_{32} = 7$

II. RESULTATS DE SIMULATION:

Avant de donner les résultats de simulation, nous allons d'abord préciser le schéma de la commande $U_r(t)$ ainsi que celui de l'algorithme d'adaptation que nous avons implantés sur ordinateur.

N'ayant aucun moyen analogique pour parfaire l'intégration continue, nous sommes passés au système discret.

- Schéma de commande $U_r(t)$

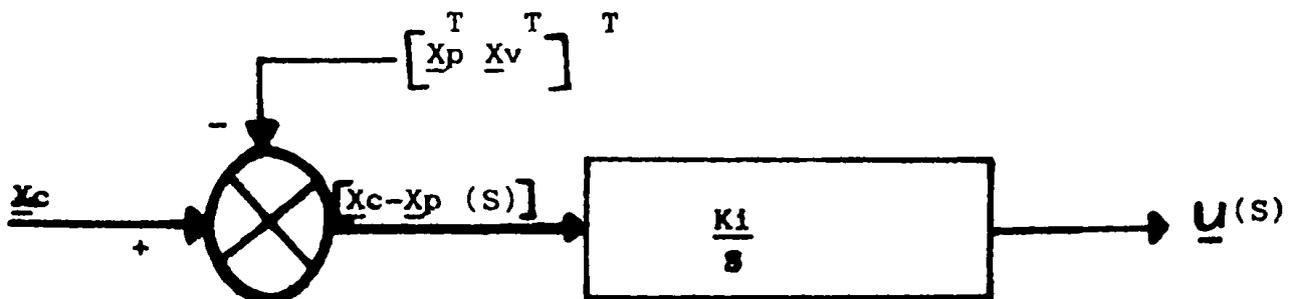
En posant:

$$U_r(S) = U(S) - V(S)$$

Avec:

$$\begin{cases} V(S) = K_p X_p(S) + K_v X_v(S) \\ \text{et} \\ U(S) = \frac{K_I}{S} (X_c - X_p(S)) \end{cases} \quad (B.14)$$

L'équation (B.14) correspond au schéma suivant:



On a alors:

$$S \underline{U}(S) = K_I (\underline{X}_c(S) - \underline{X}_p(S))$$

En passant à l'équation temporelle, on aura:

$$\underline{U}'(t) = Kz (\underline{X}_c - \underline{X}_p(t)) = \frac{\underline{U}(t_{n+1}) - \underline{U}(t_n)}{t_{n+1} - t_n}$$

Pour: $\underline{U}_n (t_{n+1} - t_n)$ très petit

$$\Rightarrow \begin{cases} \underline{U}(t_{n+1}) = \underline{U}(t_n) + \Delta t Kz (\underline{X}_c - \underline{X}_p(t)) \\ \underline{U}(t_0) = \underline{0} \quad \text{et} \quad \Delta t = t_{n+1} - t_n \end{cases}$$

En fait:

$$\Delta t = t_{n+1} - t_n = (K+1)T_E - KT_E = T_E$$

période d'échantillonnage qui sera très petite pour assurer la constance des coefficients dynamiques sur une période (pour raison de stabilité).

- ALGORITHME D'ADAPTATION

Partant de l'équation:

$$\frac{d}{dt} [\hat{m}_{ij}(t)] = - K_{ij} [Y_i X_{vrj}]$$

la dérivée d'une fonction F au point \underline{x}_0 est:

$$\frac{F(\underline{x}_0 + h) - F(\underline{x}_0)}{(\underline{x}_0 + h) - \underline{x}_0} = F'(\underline{x}_0)$$

Appliquée à l'équation précédente, cela permet d'obtenir:

$$\frac{d}{dt} [\hat{m}_{ij}(t)] = \frac{\hat{m}_{ij}(t_\beta) - \hat{m}_{ij}(t_\alpha)}{t_\beta - t_\alpha} \quad \text{avec } t_\beta = t_\alpha + h_\beta$$

$$\Rightarrow h_\beta \frac{d}{dt} [\hat{m}_{ij}(t_\alpha)] = \hat{m}_{ij}(t_\beta) - \hat{m}_{ij}(t_\alpha)$$

$$\Rightarrow \hat{m}_{ij}(t_\beta) = \hat{m}_{ij}(t_\alpha) - h_\beta K_{ij} [Y_i X_{vrj}] \quad (B.15)$$

Avec:

$t_{\beta} \in [kT_E, (k+1)T_E]$ et h_{β} : pas d'intégration, pour initialiser l'équation (B.15), nous avons utilisé les valeurs suivantes:

$$\hat{M}(t_0) = M(0) \text{ avec } [X_{p1}, X_{p2}, X_{p3}]^T = [0, 0, 0]^T \text{ rd}$$

dans le cas où la charge est de 5 kg.

Les éléments de la matrice d'inertie $\underline{M}(X_p)$ du modèle du manipulateur que nous avons implanté sur machine sont donnés en Annexe (B.3). La matrice d'inertie est fonction du vecteur position X_p ainsi que de la masse variable de la charge transportée et qui vaut 5kg, 10kg, et 20kg.

REMARQUE:

Comme \underline{Y} et \underline{X}_{vr} ne sont connues qu'aux instants kT_E (KEN) alors l'équation (B.15) s'écrira pour $t_{\beta} \in [kT_E, (k+1)T_E]$

$$\hat{m}_{ij}(t_{\beta}) = \hat{m}_{ij}(t_{\alpha}) - h_{\beta} K_{ij} [Y_i(kT_E), X_{vrj}(kT_E)]$$

- RESULTATS:

La simulation que nous avons effectuée sur l'ordinateur VAX 785 du centre de calcul a permis d'obtenir les courbes de la FIGURE (1) avec les coefficients de simulation suivants (ce sont des Matrices Constantes):

$$\begin{aligned} K_z &= 3000 I \Delta^{-3} & K_p &= 600 I \Delta^{-2} & K_v &= 40 I \Delta^{-1} \\ F_p = F_v &= 20 I & C_p &= I & C_v &= 15 I \end{aligned}$$

Où I est la matrice identité.

La FIGURE (2) montre la réponse du manipulateur pour un vecteur consigne échelon d'entrée de valeur $[0.5, 1.5, 1]^T$ rd quand la charge transportée est 5kg, 10kg et 20kg.

Ces courbes sont pratiquement confondues.

- INTERPRETATION

L'efficacité d'une commande adaptative doit être évaluée d'après deux critères:

- la stabilité du système rebouclé,
- la convergence du modèle identifié vers le vrai modèle, malheureusement, dans l'exemple présenté par TOMIZUKA et HOROWITZ [1], le deuxième critère n'est manifestement pas respecté; nous pouvons expliquer cela par deux causes principales:

- i) Au vu des courbes précédentes, les paramètres que l'on cherche à identifier varient trop rapidement; or, tous les algorithmes d'identification sont conçus pour identifier des paramètres constants ou lentement variables. En aucun cas on ne peut "SUIVRE" des paramètres variables rapidement.

Si on est en présence de tels paramètres (qui varient rapidement), il faut modéliser cette variation et identifier les coefficients de ce modèle.

- ii) Ces paramètres dépendent de la commande par l'intermédiaire de X_p et X_v et la commande adaptative dépend elle-même des paramètres estimés. De ce fait, le problème d'identification de la paramétrisation présenté par TOMIZUKA et HOROWITZ [1] est mal conditionné.

Malgré les performances médiocres de l'identification, les commandes adaptatives avec modèle de référence gardent de bonnes performances lorsque les gains K_I , K_p et K_v sont grands, la trajectoire de référence est bien suivie; cela est dû à la robustesse de la commande à grand gain, théorie développée entre-autres par Claude SAMSON [3].

Pour étayer ce résultat, nous allons effectuer une autre simulation en supposant que l'identification a été parfaite.

III. SCHEMA DE COMMANDE AVEC ADAPTATION PARFAITE:

Nous avons vu que l'équation du modèle dynamique implantée sur calculateur, est la suivante:

$$M(\underline{x}_p) \ddot{\underline{x}}_p = \underline{Q}(t)$$

Avec:

$$\underline{Q}(t) = \hat{M}(t) \underline{u}_r(t) - F_p \underline{\epsilon}_p(t) - F_v \underline{\epsilon}_v(t)$$

Soit donc:

$$\ddot{\underline{x}}_p(t) = M^{-1}(\underline{x}_p) \hat{M}(t) \underline{u}_r(t) - M^{-1}(\underline{x}_p) (F_p \underline{\epsilon}_p(t) + F_v \underline{\epsilon}_v(t))$$

Supposer que l'adaptation a été parfaite (de \hat{M}), cela revient à dire que:

$$M^{-1}(\underline{x}_p) \hat{M}(t) = I$$

Donc:

$$\ddot{\underline{x}}_p(t) = I \underline{u}_r(t) - M^{-1}(\underline{x}_p) (F_p \underline{\epsilon}_p(t) + F_v \underline{\epsilon}_v(t))$$

Et:

$$\underline{x}_p(t) = \int \ddot{\underline{x}}_p(t) dt$$

III.1 Résultats de simulation:

Les résultats obtenus sont présentés sous forme de courbes dans la FIGURE (3).

Ces résultats ont été obtenus pour les coefficients de simulation suivante:

$$K_z = 2000I \Delta^{-3} \quad K_p = 400I \quad K_v = 25I \Delta^{-1}$$

$$F_p = 30I \quad F_v = 13I \quad C_p = 0I \quad C_v = 0I$$

Où I est la matrice d'identité.

III.2 CONCLUSION:

Au vu des deux simulations effectuées, on peut déjà citer deux résultats importants:

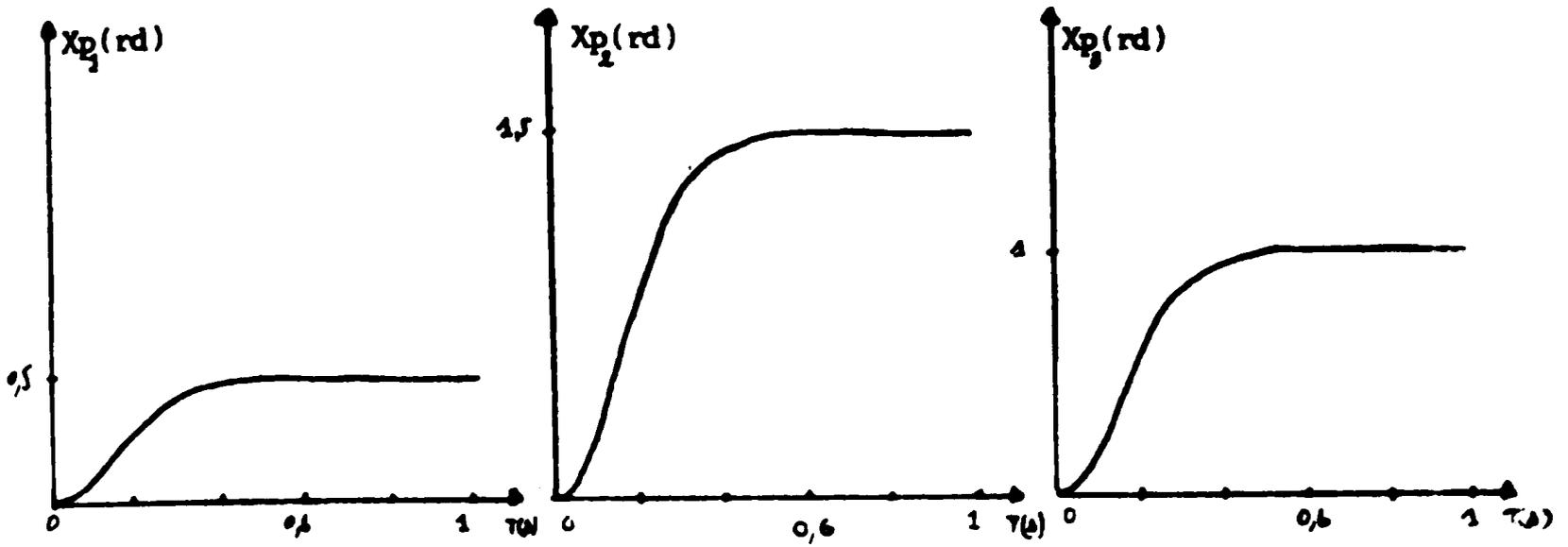
- Les gains K_i , K_p et K_v sont beaucoup plus grands dans le cas où l'identification est mauvaise que dans le cas contraire;
- Dans le cas d'une identification parfaite, les résultats sont beaucoup plus performants (l'erreur statique autour de la position terminale est éliminée, on ne remarque aucun dépassement de la réponse indicielle.

D'où on peut en conclure que:

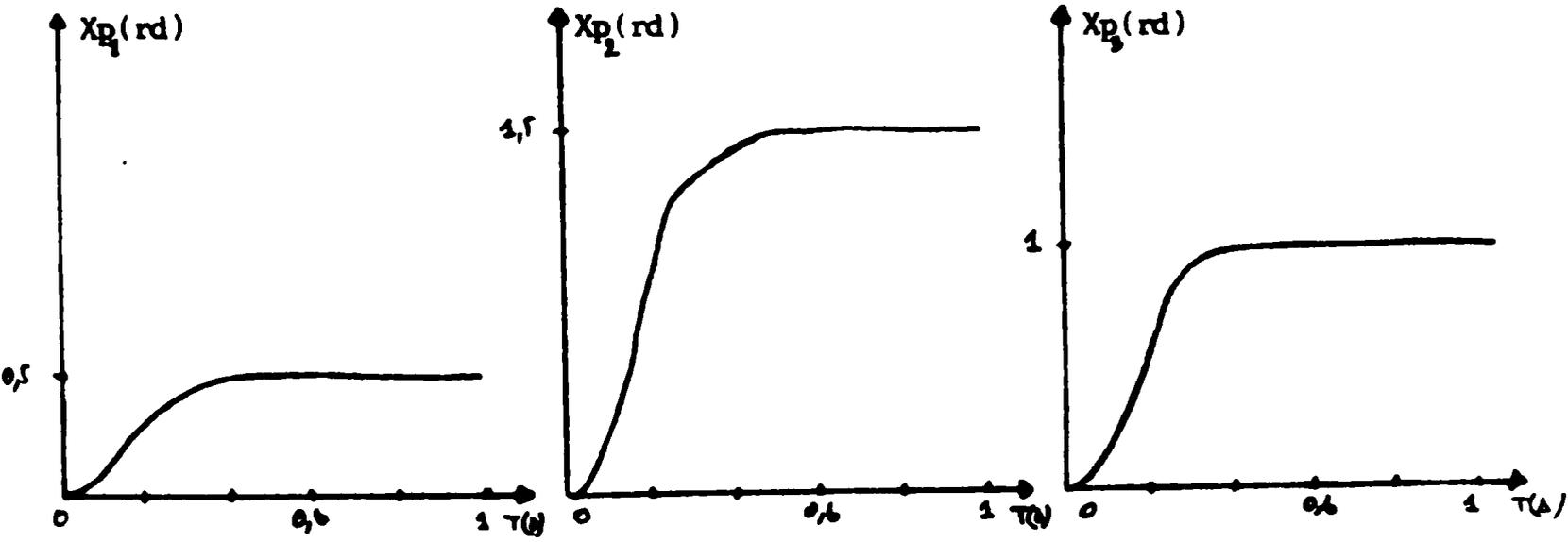
- 1) Il est très difficile d'affirmer que la commande est améliorée par l'identification (par l'adaptation).
- 2) Les grand-gains permettent un bon rattrapage de l'erreur par rapport au modèle de référence.

Ce sont donc les qualités de robustesse de la commande qui sont la cause des bons résultats obtenus plutôt que l'adaptativité. En ce qui concerne l'outil d'intégration pour notre simulation, nous avons utilisé la méthode de RUNGE-KUTTA à pas variable, vu que le système à résoudre était de faible ordre (6 équations). Cette méthode est très stable et très précise, sa stabilité nous permet de dire que nous avons fait une bonne intégration même en présence de perturbations externes (quand elles sont bornées dans le temps). Sa précision nous facilite l'analyse des résultats; en effet, à ce moment là, les résultats seront jugés bons ou mauvais que du côté de la commande, du moment que nous nous sommes assurés de leur précision.

cas ou $M=5\text{kg}$



cas ou $M=10\text{kg}$



cas ou $M=20\text{kg}$

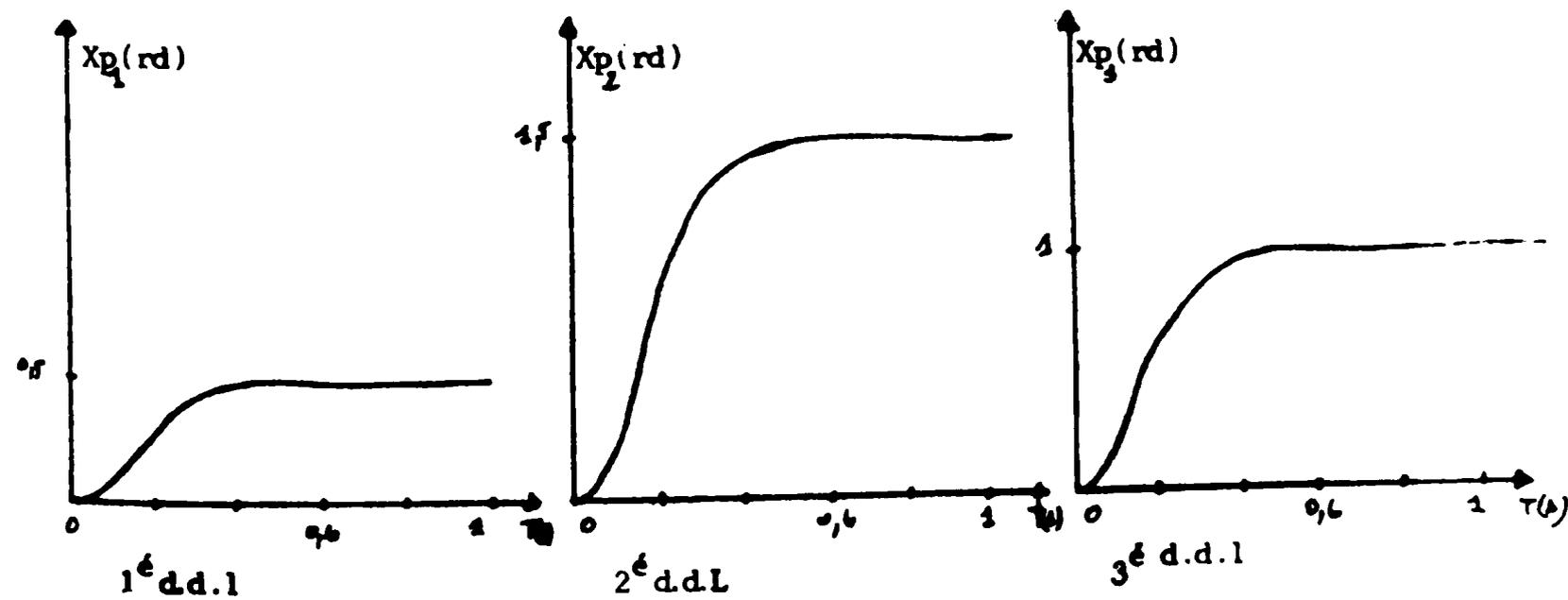
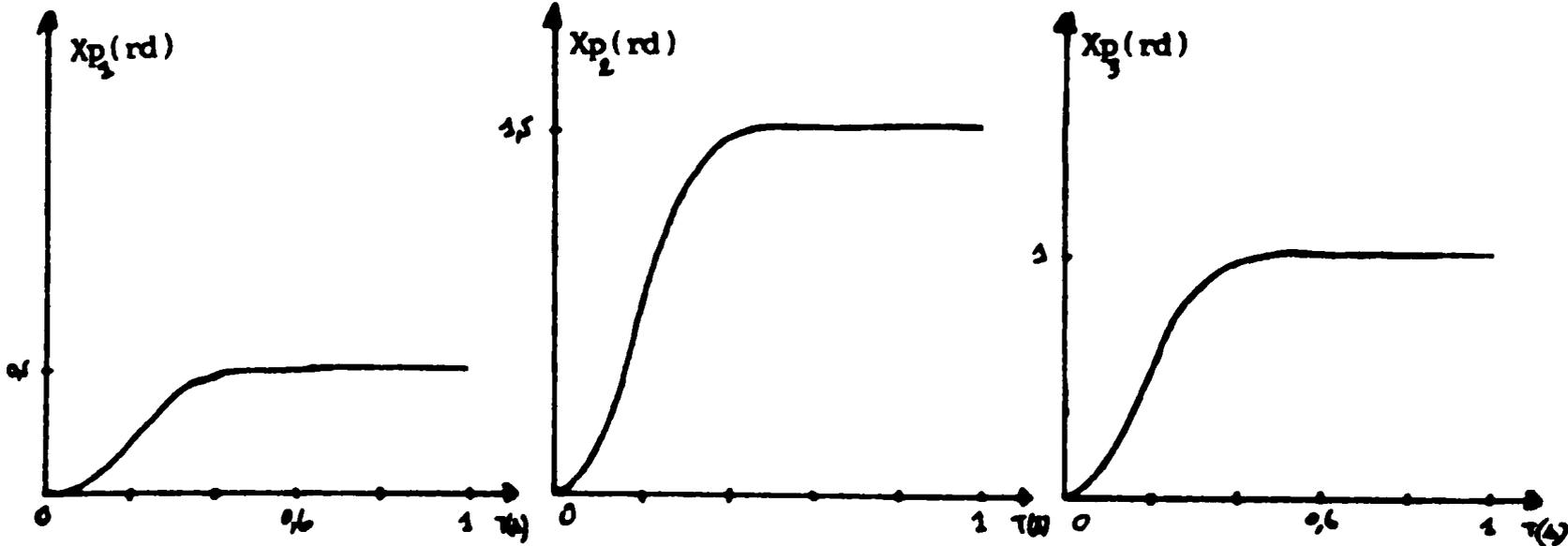
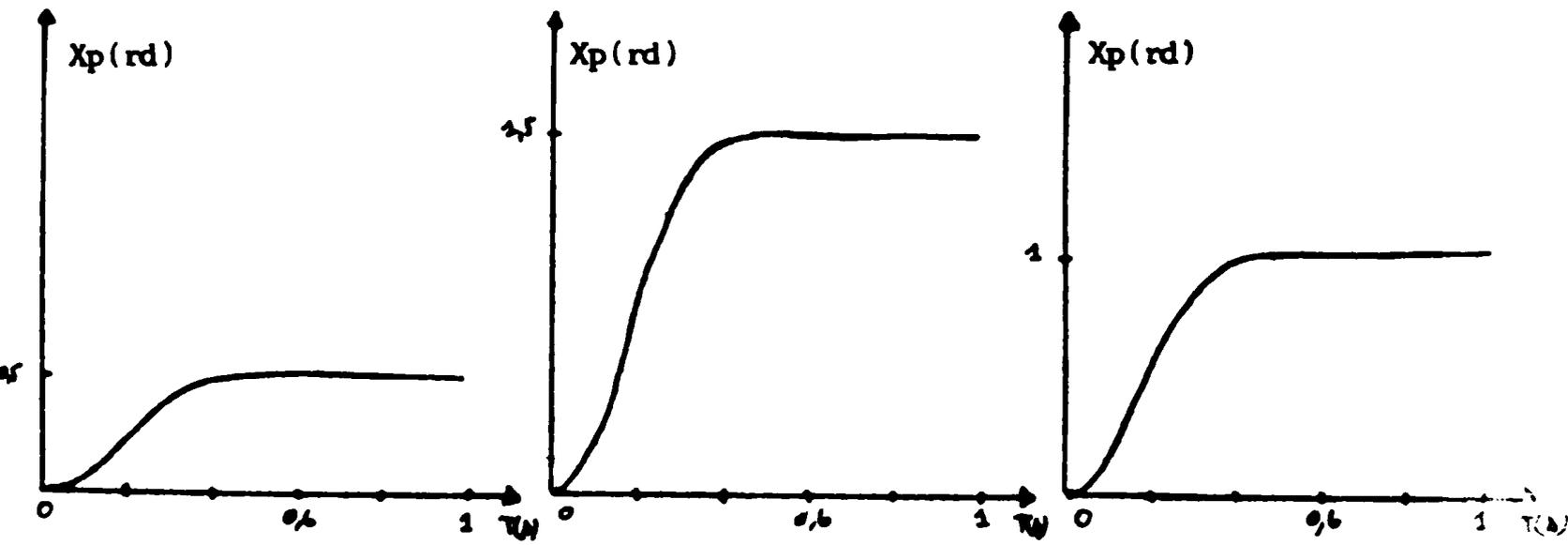


FIGURE (II)

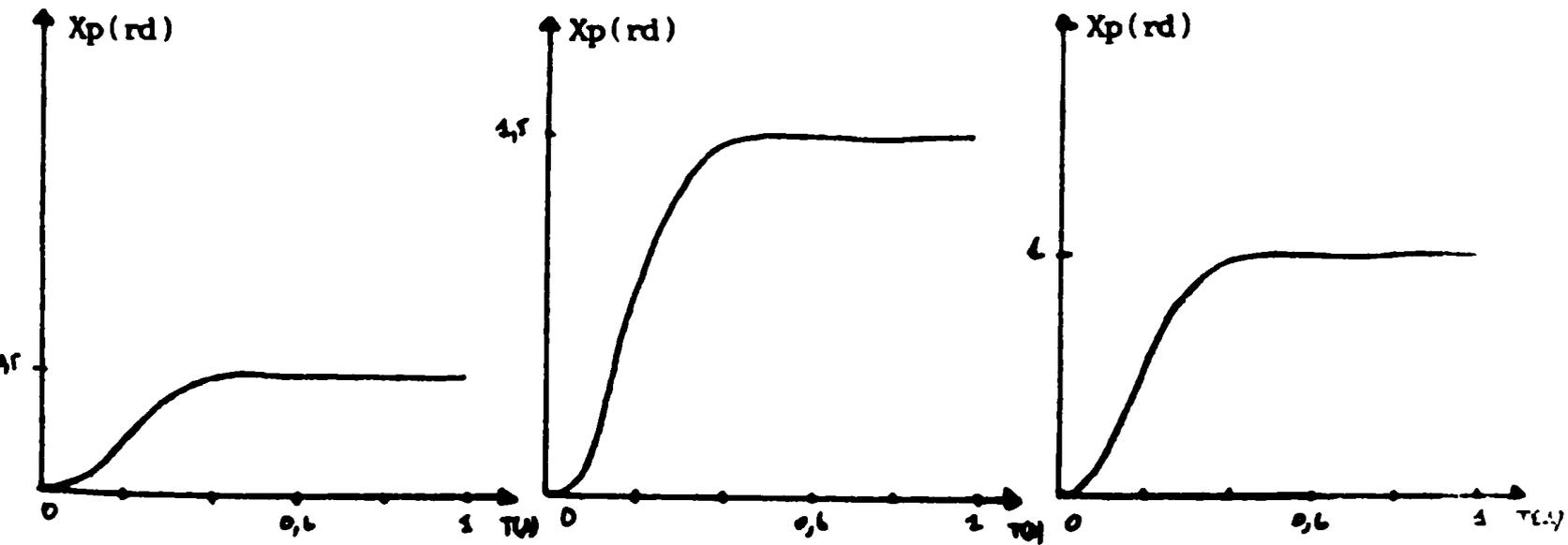
cas ou M=5kg



cas ou M=10kg



cas ou M=20kg



1^e d.d.1

2^e d.d.1

3^e d.d.1

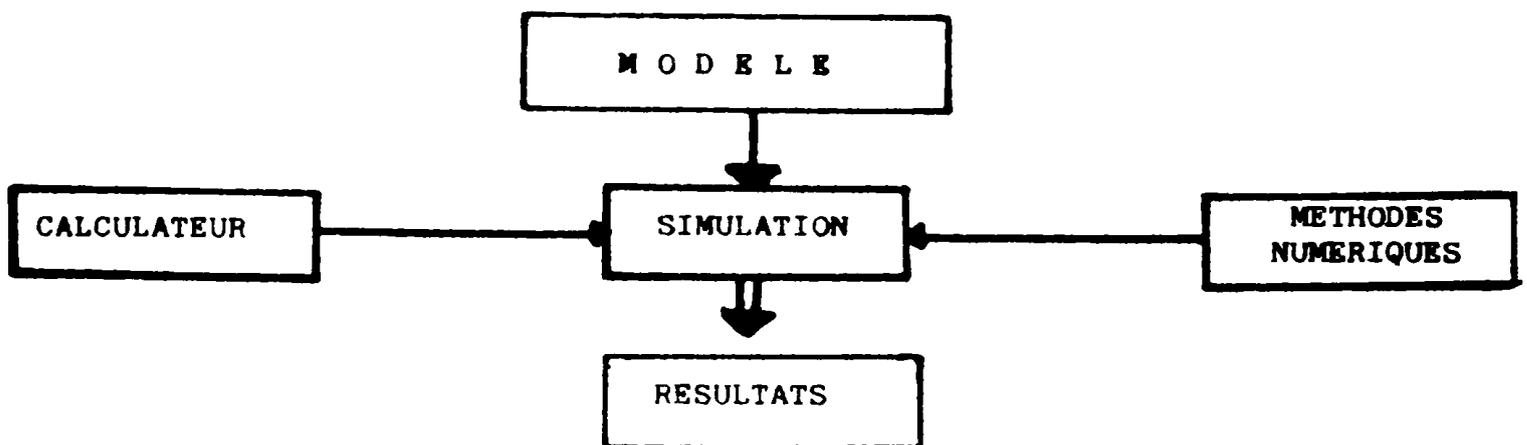
FIGURE (III)

CONCLUSION GENERALE:

La simulation numérique de processus physiques continus conduit à la résolution des modèles mathématiques les représentants. Pour mener à bien ce travail, on doit s'assurer trois conditions:

- La validité du modèle dans le domaine requis pour la simulation;
- Un calculateur assez rapide et puissant (notamment pour une simulation en temps réel);
- Des méthodes d'intégration numériques très stables et très précises.

On peut donc schématiser cela comme suit:



Les résultats et les propriétés d'une simulation sont donc liés aux trois facteurs cités précédemment:

1) Calculateur:

Nous avons utilisé un calculateur très puissant et très rapide, c'est le VAX 785 du centre de calcul.

2) Le modèle mathématique:

On a simulé le comportement dynamique d'un robot manipulateur en utilisant un algorithme de commande adaptatif. La modélisation d'un robot est un problème très complexe, en raison de l'existence de certains termes non modélisables (frottement sec - frictions, aspect aléatoire des usures de pièces dans le temps... etc).

Ce problème peut être rendu moins crucial si on arrive à synthétiser une commande robuste peu sensible vis-à-vis des erreurs de modélisation.

Notre attention s'est portée sur une commande adaptative permettant d'identifier ce modèle en ligne, malheureusement, cette identification ne peut être assurée parfaitement que dans certaines conditions, notamment les paramètres à identifier doivent être constants, ou tout au plus, lentement variables. Dans le cas contraire, il faut modéliser les variations rapides, ce qui nous ramène toujours donc à la modélisation; dans ce cas, il est possible de faire appel à une commande robuste. Nous avons alors montré qu'il suffit d'utiliser de grands gains pour contrebalancer les effets d'erreurs de modélisation (ou d'identification). L'utilisation de grands-gains peut conduire à l'augmentation de la bande passante du système, d'où des mesures très bruitées; pour cela, CLAUDE SAMSON [4] propose des gains non-linéaires variables qui prennent des valeurs importantes dans les zones névralgiques de la trajectoire.

METHODES NUMERIQUES:

Les problèmes que nous avons eus à résoudre sont des problèmes bien conditionnés (les fonctions $f(t, Y(t))$ sont uniformément lipschitziennes). Or, seules les méthodes classiques peuvent résoudre ce genre de problèmes, nous avons alors vu l'avantage acquis tant du point de vue précision que temps de calcul en utilisant ces méthodes avec un nombre et une dimension des pas variables pour mieux suivre l'évolution de la solution.

L'efficacité de la méthode multi-pas (METHODE D'ADAMS-BASHFORTH-MOULTON) à résoudre les exemples instables, est due à la phase de démarrage qui permet d'avoir une très grande précision au début de l'intégration, permettant de partir sur des valeurs très fiables.

L'avantage de la méthode de RUNGE-KUTTA à pas variable relativement à celle à pas constant, est due essentiellement à l'estimation de l'erreur et à son contrôle très performant.

Les exemples traités dans le CHAPITRE II de la PARTIE A, montrent clairement l'avantage de l'une sur l'autre. Dans notre simulation, nous avons utilisé la méthode de RUNGE-KUTTA à pas variables, les méthodes d'ADAMS (méthodes multi-pas), assurent pour la même précision fixée que la méthode de RUNGE-KUTTA, un temps d'exécution relativement court. De ce fait, elles pourront être utilisées chaque fois que le système à résoudre est d'ordre très élevé, soit entre 10 et 20 équations.

Nous avons donc essayé de limiter les conséquences des erreurs de modélisation, tout en assurant la précision, la stabilité et le moindre coût à l'outil d'intégration, le choix d'un calculateur rapide et puissant. Compte tenu de ceci, on pourra s'assurer de bonnes performances de la simulation.

En conclusion, nous pensons qu'il faut peut être orienter les recherches vers d'autres aspects importants. En effet, en dehors des travaux qui consistent à appliquer sur analyse des méthodes de commandes, mises au point dans un tout autre contexte, aux robots manipulateurs, les études sur le plan théorique de nouvelles commandes sont rares et difficiles.

Ce qui serait intéressant, ce serait de passer à des applications pratiques, mais cela demande de gros moyens et nécessite que soit résolu un certain nombre de problèmes, notamment au niveau de la programmation, des processus de calcul, de l'intégration des modules extérieurement réceptifs. Il y a là tout un domaine de recherche qui fait intervenir essentiellement des techniques liées à l'informatique, aux réseaux, aux architectures spécialisées, aux langages et à l'intelligence artificielle.

Dans le domaine mécanique, les recherches sont aussi nécessaires pour améliorer les performances des manipulateurs.

ANNEXE (B1)

Soit le schéma:

$$(1) \quad \begin{cases} Y_{n,i} = Y_n + h_n \cdot \sum_{j=1}^q a_{ij} \cdot f(t_{n,j}; Y_{n,j}) \\ Y_{n+1} = Y_n + h_n \cdot \sum_{j=1}^q b_j \cdot f(t_{n,j}; Y_{n,j}) \end{cases} \quad i = 1, \dots, q$$

et soit le schéma perturbé associé au schéma (1)

$$(2) \quad \begin{cases} Z_{n,i} = Z_n + h_n \cdot \sum_{j=1}^q a_{ij} \cdot f(t_{n,j}; Z_{n,j}) \\ Z_{n+1} = Z_n + h_n \cdot \sum_{j=1}^q b_j \cdot f(t_{n,j}; Z_{n,j}) + h_n \cdot \epsilon_n \end{cases} \quad i = 1, \dots, q$$

Posons:

$$Y_n = \begin{bmatrix} Y_{n,1} \\ \vdots \\ Y_{n,q} \end{bmatrix} ; \quad Z_n = \begin{bmatrix} Z_{n,1} \\ \vdots \\ Z_{n,q} \end{bmatrix} ; \quad e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Nous avons alors:

$$\| Y_{n,i} - Z_{n,i} \| \leq \| Y_n - Z_n \| + L h_n \sum_{j=1}^q a_{ij} \cdot \| Y_{n,j} - Z_{n,j} \|$$

Soit donc:

$$| Y_n - Z_n | \leq \| Y_n - Z_n \| + L \cdot h_n |a| | Y_n - Z_n |$$

On en déduit par récurrence:

$$| Y_n - Z_n | \leq \| Y_n - Z_n \| (1 + L h_n |a| + \dots + (L h_n)^P |a|^P) e + (L h_n)^{P+1} |a|^{P+1} | Y_n - Z_n |$$

d'où pour $P \rightarrow +\infty$ (passage à la limite)

$$(3) \quad | Y_n - Z_n | \leq \| Y_n - Z_n \| (1 - L h_0 |a|)^{-1} \quad \text{où } h_0 \text{ est le pas maximum}$$

d'où en vertu de (3)

$$\| Y_{n+1} - Z_{n+1} \| \leq \| Y_n - Z_n \| (1 + h_n L \| b^T \| (I - h_n L |a|)^{-1} e) + h_n \| \epsilon_n \|$$

En posant $C = L(b^T) (I - h_n L |a|)^{-1} e$

On obtient:

$$\| Y_{n+1} - Z_{n+1} \| \leq (1 + C h_n) \| Y_n - Z_n \| + h_n \| \epsilon_n \|$$

d'où en vertu du lemme cité dans [1], [2], [3],

$$\| Y_n - Z_n \| \leq e^{CT} \| Y_0 - Z_0 \| + \frac{e^{CT} - 1}{C} \max_n \| \epsilon_n \|$$

ANNEXE (B₂)

Détermination de l'expression de $Z_1(t)$:

Soit:

$$\begin{cases} z_1'(t) = D_y f(t, y(t)) z_1(t) + \theta^p(t) \Psi_p(t, y(t)) \\ z_1(t_0) = 0 \end{cases}$$

$$\text{et } \begin{cases} z_0'(t) = D_y f(t, y(t)) z_0(t) \\ z_0(t_0) = 1 \end{cases}$$

$$\int_{t_0}^t (z_1'(\Delta) - D_y f(\Delta, y(\Delta)) z_1(\Delta)) d\Delta = \int_{t_0}^t \theta^p(\Delta) \Psi_p(\Delta, y(\Delta)) d\Delta$$

$$\int_{t_0}^t (z_1'(\Delta) - \frac{z_1'(\Delta) z_0(\Delta)}{z_0(\Delta)}) d\Delta = \int_{t_0}^t \theta^p(\Delta) \Psi_p(\Delta, y(\Delta)) d\Delta$$

$$\int_{t_0}^t \left(\frac{z_1'(\Delta) z_0(\Delta) - z_0'(\Delta) z_1(\Delta)}{z_0^2(\Delta)} \right) d\Delta = \int_{t_0}^t \frac{\theta^p(\Delta) \Psi_p(\Delta, y(\Delta))}{z_0(\Delta)} d\Delta$$

$$\int_{t_0}^t \left(\frac{z_1(\Delta)}{z_0(\Delta)} \right)' d\Delta = \int_{t_0}^t \frac{\theta^p(\Delta) \Psi_p(\Delta, y(\Delta))}{z_0(\Delta)} d\Delta = \frac{z_1(t)}{z_0(t)} - \frac{z_1(t_0)}{z_0(t_0)}$$

$$\Rightarrow \boxed{z_1(t) = z_0(t) \int_{t_0}^t \frac{\theta^p(\Delta) \Psi_p(\Delta, y(\Delta))}{z_0(\Delta)} d\Delta.}$$

ANNEXE (B3)

calcul du modele dynamique du manipulateur
rigide articule

L4=0.914

M3=6.71

MP=5 ,10,20 kg

A=M3+MP

A2=((MP*L4)**2)/A

* CALCUL DE LA MATRICE *

C232=(COS(Q1(2)+Q1(3)))**2

C22=(COS(Q1(2)))**2

S22=(SIN(Q1(2)))**2

C2C23=COS(Q1(2))*COS(Q1(2)+Q1(3))

S232=(SIN(Q1(2)+Q1(3)))**2

S23C23=SIN(Q1(2)+Q1(3))*COS(Q1(2)+Q1(3))

C23=COS(Q1(2)+Q1(3))

S23=SIN(Q1(2)+Q1(3))

C2S2=COS(Q1(2))*SIN(Q1(2))

C2=COS(Q1(2))

S2=SIN(Q1(2))

C3=COS(Q1(3))

c32=(COS(Q1(3)))**2

S32=(SIN(Q1(3)))**2

A1=(6.368+0.792*A)*C22+0.1*S2C2

A3=(0.974+A2)*C232+0.1*S232+0.04*S23C23

A4=1.627*MP*C2C23+0.1*S22

* CALCUL DE MATRICE M *

M1(1,1)=A1+A3+A4

M1(1,2)=0.05*S2+0.5*C2+0.02*S23+0.1*C23

M1(1,3)=0.02*S23+0.1*C23

M1(2,2)=6.368+A2+0.792*A*(S32+C32)+1.627*MP*C3+0.924

M1(2,3)=0.924+A2+0.6135*MP*C3+2.3

M1(3,3)=0.924+A2+2.3

M1(2,1)=M1(1,2)

M1(3,1)=M1(1,3)

M1(3,2)=M1(2,3)

M1(1,4)=0.

M1(2,4)=0.

M1(3,4)=0.

BIBLIOGRAPHIE

1^{ère} PARTIE

- [1] CROUZEIX M.; MIGNOT A.-"Analyse numerique des equations differentielles."
edition MASSON (1983)
- [2] GEAR C.W.-"Numerical initial value problems in ordinary differential equations."(1971) PRENTICE HALL INC
- [3] SHAMPINE L.F; GORDON M.K.-"computer solution of ordinary differential equations the initial value problem."
W.H FREEMAN AND COMPANY
- [4] PHILIPPE BERNARD-33"Resolution numerique des equations differentielles par deux methodes lineaires multiples à pas et ordre variables.
THESE DE 3^{ème} CYCLE (1980) UNIVERSITE DE RENNES 1
- [5] HENRICI P.-"Discrete variable methods in ordinary differential equations."
JOHN WILEY, NEW YORK, LONDON
- [6] CROUZEIX M.; PIERRE A.R- "Approximation des problemes d'evolution
COURS D.E.A ANALYSE NUMERIQUES
universite de rennes 1

2^{ème} PARTIE

- [1] TOMIZUKA M.; HOROWITZ R.-"Application of modele reference adaptative controle techniques to mechanical manipulators:" universite of california
BERKELEY paper No 80 (1980)
- [2] MICHELE LE BORNE-"Modelisation des robots manipulateurs rigides."
publication interne irisa (1985)
- [3] PAUL COIFFET- "Les robots: modelisation et commande." TOME1
EDITION HERMES
- [4] CLAUDE SAMSON- "Commande non lineaire robuste des robots manipulateurs.
publication interne IRISA(1985)
- [5] RICHARD D.; CLAUDE S.- "Robuste non lineaire control of robotics manipulators:
implementation aspects and simulations." IRISA/INRIA RENNES(1985)
- [6] EDSON DE PAULA FERREIRA.- "Contribution à l'identification de parametres et à
la commande adaptative des manipulateurs."
these de docteur ingenieur(1984) universite de PAUL-SABATIER
TOULOUSE