Ministère de l'Enseignement et de la Recherche Scientifique

Université des Sciences et de la Technologie
« Houari BOUMEDIENE »

ECOLE NATIONALE POLYTECHNIQUE D'ALGER
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES
INGENIORAT D'ETAT EN ELECTRONIQUE

Etude de la Fonction signe de matrice Applications à la résolution des équations de Riccati et de Lyapunov

Proposé et suivi par :

Mr. R. TOUMI : Dr. Ingénieur
Melle M. AMINI : Dr. Ingénieur

Etudié par :

Y. LAOUAR
A. HAMZAOUI

JANVIER 1983

Ministère de l'Enseignement et de la Recherche Scientifique

Université des Sciences et de la Technologie
« Houari BOUMEDIENE »

ECOLE NATIONALE POLYTECHNIQUE D'ALGER
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES
INGENIORAT D'ETAT EN ELECTRONIQUE

Etude de la Fonction signe de matrice Applications à la résolution des équations de Riccati et de Lyapunov

Proposé et suivi par :

Mr. R. TOUMI : Dr. Ingénieur
Melle M. AMINI : Dr. Ingénieur

Etudié par :

Y. LAOUAR
A. HAMZAOUI

JANVIER 1983

Ministère de l'Enseignement et de la Recherche Scientifique

Université des Sciences et de la Technologie
« Houari BOUMEDIENE »

ECOLE NATIONALE POLYTECHNIQUE D'ALGER
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

INGENIORAT D'ETAT EN ELECTRONIQUE

Etude de la Fonction signe de matrice Applications à la résolution des équations de Riccati et de Lyapunov

Proposé et suivi par :

Mr. R. TOUMI : Dr. Ingénieur

Melle M. AMINI : Dr. Ingénieur

Etudié par :

Y. LAOUAR

A. HAMZAOUI

JANVIER 1983

Ministère de l'Enseignement et de la Recherche Scientifique

Université des Sciences et de la Technologie
« Houari BOUMEDIENE »

ECOLE NATIONALE POLYTECHNIQUE D'ALGER
DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES
INGENIORAT D'ETAT EN ELECTRONIQUE

Etude de la Fonction signe de matrice Applications à la résolution des équations de Riccati et de Lyapunov

Proposé et suivi par :
Mr. R. TOUMI : Dr. Ingénieur
Melle M. AMINI : Dr. Ingénieur

Etudié par :
Y. LAOUAR
A. HAMZAOUI

JANVIER 1983

بسم الله الرحمن الرحيم

" الله نور السماوات والأرض مثل نوره كمشكاة
فيها مصباح المصباح في زجاجة الزجاجة
كأنها كوكب دري يوقد من شجرة مباركة
زيتونة لاشرقية ولاغربية يكاد زيتها يضيء
ولو لم تمسسه نار نور على نور يهدي الله لنوره
من يشاء ويضرب الله الأمثال للناس والله
بكل شيء عليم "

سورة النور

بسم الله الرحمن الرحيم

" الله نور السماوات والأرض مثل نوره كمشكاة
فيها مصباح المصباح في زجاجة الزجاجة
كأنها كوكب دري يوقد من شجرة مباركة
زيتونة لا شرقية ولا غربية يكاد زيتها يضيء
ولو لم تمسسه نار نور على نور يهدي الله لنوره
من يشاء ويضرب الله الأمثال للناس والله
بكل شيء عليم "

سورة النور

REMERCIEMENTS

Le présent travail a été réalisé au sein de la division .V. du C.E.N.

Pour cela, nous tenons à remercier Mr. SANSAL pour nous avoir admis au sein de sa division.

Nous exprimons toute notre reconnaissance à Mr. TOUMI et Mlle AMINI pour l'encadrement et l'aide considérable qui nous ont été fourni, durant tout le semestre.

Nous tenons également à remercier tout le personnel du centre de calcul du C.E.N. en particulier RACHID et ABDENOUR.

Le Maire et tous les employés de l'A.P.C. de M'toussa pour la frappe.

Nos vifs remerciements aussi pour Mme. BAHLOUL H. et Monsieur BENSLITANE. B . pour le tirage.

REMERCIEMENTS

Le présent travail a été réalisé au sein de la division .V. du C.E.N.

Pour cela, nous tenons à remercier Mr. SANSAL pour nous avoir admis au sein de sa division.

Nous exprimons toute notre reconnaissance à Mr. TOUMI et Mlle AMINI pour l'encadrement et l'aide considérable qui nous ont été fourni, durant tout le semestre.

Nous tenons également à remercier tout le personnel du centre de calcul du C.E.N. en particulier RACHID et ABDENOUR.

Le Maire et tous les employés de l'A.P.C. de M'toussa pour la frappe.

Nos vifs remerciements aussi pour Mme. BAHLOUL H. et Monsieur BENSLITANE. B . pour le tirage.

REMERCIEMENTS

Le présent travail a été réalisé au sein de la division .V. du C.E.N.

Pour cela, nous tenons à remercier Mr. SANSAL pour nous avoir admis au sein de sa division.

Nous exprimons toute notre reconnaissance à Mr. TOUMI et Mlle AMINI pour l'encadrement et l'aide considérable qui nous ont été fourni, durant tout le semestre.

Nous tenons également à remercier tout le personnel du centre de calcul du C.E.N. en particulier RACHID et ABDENOUR.

Le Maire et tous les employés de l'A.P.C. de M'toussa pour la frappe.

Nos vifs remerciements aussi pour Mme. BAHLOUL H. et Monsieur BENSLITANE. B . pour le tirage.

REMERCIEMENTS

Le présent travail a été réalisé au sein de la division .V. du C.E.N.

Pour cela, nous tenons à remercier Mr. SANSAL pour nous avoir admis au sein de sa division.

Nous exprimons toute notre reconnaissance à Mr. TOUMI et Mlle AMINI pour l'encadrement et l'aide considérable qui nous ont été fourni, durant tout le semestre.

Nous tenons également à remercier tout le personnel du centre de calcul du C.E.N. en particulier RACHID et ABDENOUR.

Le Maire et tous les employés de l'A.P.C. de M'toussa pour la frappe.

Nos vifs remerciements aussi pour Mme. BAHLOUL H. et Monsieur BENSLITANE. B . pour le tirage.

DEDICACE

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

A MON GRAND-PERE

A MON PERE ET MA MERE QUI M'ONT TOUT DONNE

A MES FRERES ET SOEURS

A MES COUSINS ET TOUS MES AMIS

Yahia

A LA MEMOIRE DE MON PERE

A MA GRAND-MERE ET MA MERE

A MES FRERES ET SOEURS

A TOUS MES AMIS ET COUSINS

Abdelaziz

DEDICACE

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

A MON GRAND-PERE

A MON PERE ET MA MERE QUI M'ONT TOUT DONNE

A MES FRERES ET SOEURS

A MES COUSINS ET TOUS MES AMIS

Yahia

A LA MEMOIRE DE MON PERE

A MA GRAND-MERE ET MA MERE

A MES FRERES ET SOEURS

A TOUS MES AMIS ET COUSINS

Abdelaziz

TABLE DES MATIERES

TABLE DES MATIERES

CHAPITRE III/ APPLICATIONS

CHAPITRE III/ APPLICATIONS

INTRODUCTION

Ce travail est consacré à l'application d'un concept, la fonction signe d'une matrice, à l'élaboration d'algorithmes résolvant numériquement les équations de Riccati et de Lyapunov que l'on rencontre dans les problemes de commande optimale et de stabilité des systemes.

Ce problème peut paraître démodé à beaucoup de spécialistes. Il ya plus de quinze ans qu'il a été formalisé et que les premières solutions numériques ont été présentées. En fait les solutions connues et largement enseignées jusqu'à une date récente sont inéfficaces dans au moins deux situations:

- -lorsque la dimension du vecteur d'etat est importante.
- -lorsque le système "linéaire" est instable (ou voisin de l'instabilité).

L'évolution actuelle des applications de l'automatique conduit souvent à la résolution de problèmes qui peuvent notamment dépasser le cadre de la commande de systèmes intrinsèquement stables.

A ce renouvellement récent des domaines d'applications doit correspondreun renouveau des méthodes auquelles notre travail contribue. Il présente en effet une méthode efficace de calcul direct des paramètres stationnaires optimaux d'un feedback de commande dans le cas de systèmes non nécessairement stables et de dimension quelconque.

On attire l'attention du lecteur que ces nouvelles méthodes ne sont pas le fruit du hasard, mais sont liées etroitement au problème de la réduction des calculs (temps machine) et la possibilité de commander en temps réel un processus (instalations industrielles et Probleme de la pour suite d'une cible) et ceci grâce à la possibilité de disposer de calculateurs puissants et rapides.

Ce concept (fonction signe) et surtout son application en automatique résulte de la convergence d'idées, d'études de plusieurs auteurs entre autres "Casti en I977" [3] "Anderson" [4] au congrès IEEE I977 et I978

INTRODUCTION

Ce travail est consacré à l'application d'un concept, la fonction signe d'une matrice, à l'élaboration d'algorithmes résolvant numériquement les équations de Riccati et de Lyapunov que l'on rencontre dans les problemes de commande optimale et de stabilité des systemes.

Ce problème peut paraître démodé à beaucoup de spécialistes. Il ya plus de quinze ans qu'il a été formalisé et que les premières solutions numériques ont été présentées. En fait les solutions connues et largement enseignées jusqu'à une date récente sont inéfficaces dans au moins deux situations:

- -lorsque la dimension du vecteur d'etat est importante.
- -lorsque le système "linéaire" est instable (ou voisin de l'instabilité).

L'évolution actuelle des applications de l'automatique conduit souvent à la résolution de problèmes qui peuvent notamment dépasser le cadre de la commande de systèmes intrinsèquement stables.

A ce renouvellement récent des domaines d'applications doit correspondreun renouveau des méthodes auquelles notre travail contribue. Il présente en effet une méthode efficace de calcul direct des paramètres stationnaires optimaux d'un feedback de commande dans le cas de systèmes non nécessairement stables et de dimension quelconque.

On attire l'attention du lecteur que ces nouvelles méthodes ne sont pas le fruit du hasard, mais sont liées etroitement au problème de la réduction des calculs (temps machine) et la possibilité de commander en temps réel un processus (instalations industrielles et Probleme de la pour suite d'une cible) et ceci grâce à la possibilité de disposer de calculateurs puissants et rapides.

Ce concept (fonction signe) et surtout son application en automatique résulte de la convergence d'idées, d'études de plusieurs auteurs entre autres "Casti en I977" [3] "Anderson" [4] au congrès IEEE I977 et I978

et les travaux de "Beavers et Denman"en I976;ainsi que ceux de Baraud.

Par ce travail nous nous proposons dans un premier chapitre d'etudier le concept de fonction signe de matrice. Le deuxième chapitre est relatif à l'application de ce concept à la resolution des équations de Riccati et de Lyapunov.

Quand au dernier chapitre ,il fait l'objet de diverses applications et de tests numériques avec interpretations.

et les travaux de "Beavers et Denman"en 1976;ainsi que ceux de Baraud.

Par ce travail nous nous proposons dans un premier chapitre d'etudier le concept de fonction signe de matrice. Le deuxième chapitre est relatif à l'application de ce concept à la resolution des équations de Riccati et de Lyapunov.

Quand au dernier chapitre ,il fait l'objet de diverses applications et de tests numériques avec interpretations.

CHAPITRE I/

ETUDE DE LA FONCTION SIGNE D'UNE MATRICE

CHAPITRE I/

ETUDE DE LA FONCTION SIGNE D'UNE MATRICE

INTRODUCTION

Nous allons introduire dans ce chapitre, le concept de fonction signe d'une matrice et nous montrerons que l'algorithme de NEWTON permettant de calculer cette fonction, présente d'abord une convergence linéaire à évolution chaotique avant d'aborder sa phase finale du second ordre.

De cette analyse découlent deux nouveaux algorithmes, l'un fini pour les matrices à spectres réels, l'autre relatif au cas général constituant une méthode de NEWTON accélérée.

Quand à l'application de cette technique, elle fera l'objet des parties qui suivent.

I N T R O D U C T I O N

Nous allons introduire dans ce chapitre, le concept de fonction signe d'une matrice et nous montrerons que l'algorithme de NEWTON permettant de calculer cette fonction, présente d'abord une convergence linéaire à évolution chaotique avant d'aborder sa phase finale du second ordre.

De cette analyse découlent deux nouveaux algorithmes, l'un fini pour les matrices à spectres réels, l'autre relatif au cas général constituant une méthode de NEWTON accélérée.

Quand à l'application de cette technique, elle fera l'objet des parties qui suivent.

I-INTRODUCTION A LA FONCTION SIGNE D'UNE MATRICE

On sait par definition qu'une matrice carrée A(n.n) est semblable à sa forme de Jordan:

$$A=MJM^{-1} \qquad (1)$$

où M est la matrice des vec teurs propres de A et la matrice J a la structure:

$$\begin{bmatrix} J & \cdots & 0 \\ & & \\ 0 & & J \end{bmatrix} \quad \text{avec} \quad Jj = \begin{bmatrix} \lambda i & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda i \end{bmatrix}$$

où le nombre de blocs de Jordan Jj associés à la valeur propre λ_i est égal au nombre de vecteurs propres lineairement independants relatifs à λ

Soit maintenant $f(\lambda)$ d'une fonction definie sur le spectre de A.On aura

$$f(A)=Mf(J)M^{-1}$$

avec

$$f(J)=\begin{bmatrix} f(J_1) & & 0 \\ & \ddots & \\ 0 & & f(J_k) \end{bmatrix} ; f(Jj)=\begin{bmatrix} f(\lambda i) & f'(\lambda i) & \cdots & \frac{f(\lambda i)^{(nj-1)}}{(nj-1)!} \\ & \ddots & \ddots & \vdots \\ & & & f'(\lambda i) \\ & & & f(\lambda i) \end{bmatrix}$$

On définit le signe d'un nombre complexe Z=x+iy comme étant

$$\text{signe}(Z)=\frac{x}{|x|}$$

Appliquons maintenant la fonction signe à une matrice; on aura

$$\text{signe}(A)=M.\text{signe}(J).M^{-1}= S \qquad (2)$$

avec

$$\text{signe}(J)=\begin{bmatrix} \text{signe}(J_1) & & 0 \\ & \ddots & \\ 0 & & \text{signe}(J_k) \end{bmatrix}$$

I-INTRODUCTION A LA FONCTION SIGNE D'UNE MATRICE

On sait par definition qu'une matrice carrée A(n.n) est semblable à sa forme de Jordan:

$$A=MJM^{-1} \qquad (1)$$

où M est la matrice des vec teurs propres de A et la matrice J a la structure:

$$\begin{bmatrix} J & \cdots & 0 \\ & & \\ 0 & & J \end{bmatrix} \quad \text{avec} \quad Jj = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix}$$

où le nombre de blocs de Jordan Jj associés à la valeur propre λ_i est égal au nombre de vecteurs propres lineairement independants relatifd à λ

Soit maintenant $f(\lambda)$ d'une fonction definie sur le spectre de A.On aura

$$f(A)=Mf(J)M^{-1}$$

avec

$$f(J)=\begin{bmatrix} f(J_1) & & 0 \\ & \ddots & \\ 0 & & f(J_k) \end{bmatrix} ; f(J_j)=\begin{bmatrix} f(\lambda_i) & f'(\lambda_i) & \cdots & \frac{f(\lambda_i)^{(nj-1)}}{(nj-1)!} \\ & \ddots & \ddots & \vdots \\ & & & f'(\lambda_i) \\ & & & f(\lambda_i) \end{bmatrix}$$

On définit le signe d'un nombre complexe Z=x+iy comme étant

$$\text{signe}(Z)=\frac{x}{|x|}$$

Appliquons maintenant la fonction signe à une matrice; on aura

$$\text{signe}(A)=M.\text{signe}(J).M^{-1}=S \qquad (2)$$

avec

$$\text{signe}(J)=\begin{bmatrix} \text{signe}(J_1) & & 0 \\ & \ddots & \\ 0 & & \text{signe}(J_k) \end{bmatrix}$$

et

$$\text{signe}(J_j)=\begin{bmatrix}\text{signe}(\lambda i) & & 0\\ & \ddots & \\ 0 & & \text{signe}(\lambda i)\end{bmatrix} = \pm I$$

$+I$ si réel$(\lambda i)>0$ $\qquad -I$ si réel$(\lambda i)<0$

Resumé et propriétés

$A=MJM^{-1}$

$S=\text{signe}(A)$

$S=MDM$ $\quad D=\text{diag}(.....\text{signe}(\lambda i).....)$

$S=I$ si réel$(\lambda i)>0 \forall i$

$S=-I$ si réel$(\lambda i)<0 \forall i$

$S=A$ si A orthogonale symetrique

$\text{Signe}(A^{-1})=\text{Signe}(A)$

$\text{Signe}(\alpha A)=\text{Signe}(A)\ \forall \alpha \in R^{+*}$

$\text{Signe}(A^T)=S^T$

$\text{Signe}(B)=VSV^{-1}$ si $B=VAV^{-1}$

2-CONSTRUCTION D'UNE SUITE Z_K TELLE QUE $Z_K \to \text{Signe}(Z_o)$

On a vu ultérieurement que pour connaitre le signe d'une matrice, il suffisait de connaître les signes de ses valeurs propres.Donc il nous a paru utile de voir dans ce paragraphe comment trouver le signe d'un nombre.On demontrera ici que pour trouver la signe d'un nombre(complexe ou réel),on peut utiliser une suite qui à l'infini,nous donne son signe.

Une telle suite peut être introduite en cherchant à determiner les zeros de la fonction complexe

$$f(Z)=Z^2-I$$

par la methode de Newton.Ce qui donne l'algorithme

$$Z_{K+I}=Z_K-\frac{f(Z_K)}{f'(Z)}$$

c'est à dire(annexe):

$$Z_{K+I}=\frac{I}{2}\left(Z_K+\frac{I}{Z_K}\right) \qquad (3)$$

et

$$\text{signe}(J_j)=\begin{bmatrix}\text{signe}(\lambda i) & & 0\\ & \ddots & \\ 0 & & \text{signe}(\lambda i)\end{bmatrix}=\pm I$$

$+I$ si réel$(\lambda i)>0$ $\qquad$ $-I$ si réel$(\lambda i)<0$

Resumé et propriétés

$A=MJM^{-1}$

$S=\text{signe}(A)$

$S=MDM$ $\qquad$ $D=\text{diag}(.....\text{signe}(\lambda i).....)$

$S=I$ $\qquad$ si réel$(\lambda i)>0\ \forall i$

$S=-I$ $\qquad$ si réel$(\lambda i)<0\ \forall i$

$S=A$ $\qquad$ si A orthogonale symetrique

$\text{Signe}(A^{-1})=\text{Signe}(A)$

$\text{Signe}(\alpha A)=\text{Signe}(A)\ \forall \alpha \in R^{+*}$

$\text{Signe}(A^T)=S^T$

$\text{Signe}(B)=VSV^{-1}$ $\qquad$ si $B=VAV^{-1}$

2-CONSTRUCTION D'UNE SUITE Z_K TELLE QUE $Z_K=\text{Signe}(Z_o)$

On a vu ultérieurement que pour connaitre le signe d'une matrice, il suffisait de connaître les signes de ses valeurs propres.Donc il nous a paru utile de voir dans ce paragraphe comment trouver le signe d'un nombre.On demontrera ici que pour trouver la signe d'un nombre(complexe ou réel),on peut utiliser une suite qui à l'infini,nous donne son signe.

Une telle suite peut être introduite en cherchant à determiner les zeros de la fonction complexe

$$f(Z)=Z^2-I$$

par la methode de Newton.Ce qui donne l'algorithme

$$Z_{K+I}=Z_K-\frac{f(Z_K)}{f'(Z)}$$

c'est à dire(annexe):

$$Z_{K+I}=\frac{I}{2}\left(Z_K+\frac{I}{Z_K}\right) \qquad (3)$$

2-I- CAS REEL

Etudions maintenant plus en detail l'evolution de Z_K. Pour cela nous allons supposer dans un premier cas que Zo est réel($Z_o=x_o$).
Notre suite devient:

$$X_{K+I}=\frac{I}{2}(X_K+\frac{I}{X_K}) \qquad (4)$$

on voit clairement que:

$$\text{signe}(X_K)=\text{signe}(X_o) \quad \forall\ K \qquad (5)$$

On peut donc supposer pour la suite de notre raisonnement que $X_o>0$ et par raison de symétrie de X et I/X par rapport àI,il suffit de considérer $X_o>I$

d'où l'on peut écrire

$$I<X_{K+I}<X_K \ \ldots\ldots\ldots < X_o \qquad (6)$$

Si on prend $X_K=I+\varepsilon \Longrightarrow X_{K+I}=I+\frac{I}{2}\cdot\frac{\varepsilon^2}{1+\varepsilon}$ (7)

c'est à dire

$$X_{K+I}=I+\theta(\frac{\varepsilon^2}{2})$$

resultat qui caractérise la convergence d'ordre deux de l'algorithme de Newton.
D'après (4) et (5) on peut écrire

$$\frac{X_K}{2} < X_{K+I}$$

et sachant que :

$$X_K>I \Longrightarrow I/X<I$$

donc

$$I/2(X_K+I)> I/2(X_K+I/X_K)$$

d'ou

$$I/2(X_K+I)>X_{K+I}$$

finalement on aura la relation:

$$X_K/2<X_{K+I}<I/2(I+X_K) \qquad (8)$$

En prenant K+I=n dans l' négalité (5), on aura:

$$X_o/2^n< X_n <(I-I/2^n)+X_o/2^n \qquad (9)$$

2-I- CAS REEL

Etudions maintenant plus en detail l'evolution de Z_K. Pour cela nous allons supposer dans un premier cas que Zo est réel($Z_o=x_o$).
Notre suite devient:

$$X_{K+I}=\frac{I}{2}(X_K+\frac{I}{X_K}) \tag{4}$$

on voit clairement que:

$$\text{signe}(X_K)=\text{signe}(X_o) \quad \forall\ K \tag{5}$$

On peut donc supposer pour la suite de notre raisonnement que $X_o>0$ et par raison de symétrie de X et I/X par rapport àI, il suffit de considérer $X_o>I$

d'où l'on peut écrire

$$I<X_{K+I}<X_K \ \ldots\ldots\ldots < X_o \tag{6}$$

Si on prend $X_K=I+\varepsilon \Longrightarrow X_{K+I}=I+\frac{I}{2}\cdot\frac{\varepsilon^2}{1+\varepsilon}$ (7)

c'est à dire

$$X_{K+I}=I+\theta(\frac{\varepsilon^2}{2})$$

resultat qui caractérise la convergence d'ordre deux de l'algorithme de Newton.
D'après (4) et (5) on peut écrire

$$\frac{X_K}{2} < X_{K+I}$$

et sachant que :

$$X_K>I \Longrightarrow I/X<I$$

donc

$$I/2(X_K+I)> I/2(X_K+I/X_K)$$

d'ou

$$I/2(X_K+I)> X_{K+I}$$

finalement on aura la relation:

$$X_K/2 < X_{K+I} < I/2(I+X_K) \tag{8}$$

En prenant K+I=n dans l' négalité (5), on aura:

$$X_o/2^n < X_n < (I-I/2^n)+X_o/2^n \tag{9}$$

Ainsi si Xo est de l'ordre de 2^n, il nous faut donc n itérations pour le ramener à une valeur comprise entre I et 2; ce qui traduit une convfrgence linéaire dans un premier temps ,après cela la convergence d'ordre 2 intervient.

A l'aide de la relation (6), pour $K > n+I$, on peut connaître le nombre d'itérations à effectuer pour avoir une precision donnée.

En posant $\quad X_{n+p}=I+\xi_p \qquad 0<\xi_0<I$

on aura: $\quad \xi_p = \frac{1}{2}\,\frac{\xi^2_{p-1}}{1+\xi_{p-1}}$

En prenant $\xi_o=I$, qui correspond au cas le plus défavorable, on pourra dresser le tableau suivant:

ξ_0	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7
I	0,25	0,025	3.10^{-4}	$4,6.10^{-8}$	$I,I.10^{-15}$	$5,8.10^{-31}$	$I,7.10^{-61}$

Ainsi le nombre total d'itérés pour obtenir une précision machine vaut

$$N=n+p \qquad (10)$$

2-2-CAS GENERAL: Zo COMPLEXE

Sans restreindre la genéralité du raisonnement, on peut supposer que

$\rho_0 > 1$ et $0 < \theta < \frac{\Pi}{2}$

Par definition on a $\quad Ze=\rho_0(\cos\theta_0+i\sin\theta_0)$

d'où $\quad Z_{K+I} = \rho_{K+I}(\cos\theta_{K+I}+i\sin\theta_{K+I})$

Ainsi en utilisant (3), on obtient

$$\rho^2_{K+I}=I/4(\rho_K-I/\rho_K)^2+\cos^2\theta_Y=I/4(\rho_Y+I/\rho_K)^2-\sin^2\theta_K \qquad (11)$$

et

$$tg\theta_{K+I}=\frac{\rho_K^2-1}{\rho_K^2+1}tg\theta_K \qquad (12)$$

Etant donné que

Ainsi si Xo est de l'ordre de 2^n, il nous faut donc n itérations pour le ramener à une valeur comprise entre I et 2; ce qui traduit une convergence linéaire dans un premier temps ,après cela la convergence d'ordre 2 intervient.

A l'aide de la relation (6), pour $K > n+I$, on peut connaître le nombre d'itérations à effectuer pour avoir une precision donnée.
En posant $\quad X_{n+p}=I+\xi_p \qquad 0<\xi_0<I$

on aura: $\quad \xi_p = \frac{1}{2}\ \frac{\xi^2_{p-1}}{1+\xi_{p-1}}$

En prenant $\xi_o=I$, qui correspond au cas le plus défavorable,on pourra dresser le tableau suivant:

ξ_0	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7
I	0,25	0,025	3.10^{-4}	$4,6.10^{-8}$	$I,I.10^{-15}$	$5,8.10^{-31}$	$I,7.10^{-61}$

Ainsi le nombre total d'itérés pour obtenir une précision machine vaut

$$N=n+p \qquad (10)$$

2-2-CAS GENERAL: Zo COMPLEXE

Sans restreindre la généralité du raisonnement, on peut supposer que

$\rho_o > 1 \qquad$ et $\quad 0<\theta<\frac{\Pi}{2}$

Par definition on a $\quad Z_e=\rho_e(\cos\theta_e+i\sin\theta_e)$

d'où $\quad Z_{K+I}=\rho_{K+I}(\cos\theta_{K+I}+i\sin\theta_{K+I})$

Ainsi en utilisant (3),on obtient

$$\rho^2_{K+I}=I/4(\rho_K-I/\rho_K)^2+\cos^2\theta_K=I/4(\rho_K+I/\rho_K)^2-\sin^2\theta_K \qquad (11)$$

et

$$tg\theta_{K+I}=\frac{\rho_K^2-1}{\rho_K^2+1}tg\theta_K \qquad (12)$$

Etant donné que

$$\varphi < \left|\frac{\rho_K^2 - 1}{\rho_K^2 + 1}\right| < I \quad \Longrightarrow \quad |tg\varphi_{K+1}| < |tg\varphi_K|$$

on a ainsi

$$\varphi < |\varepsilon_{K+I}| < |\varepsilon_K| \tag{I3}$$

ce qui traduit la convergence de ε_K vers zero.

Etudions maintenant la decroissance de ρ en supposant $\rho \gg I$ et si

$$\rho_K > 1$$

d'après (II) on peut écrire

$$\frac{1}{4}\rho_K^2 - \frac{1}{2} < \rho_{K+1}^2 < \frac{1}{4}\rho_K^2 + \frac{3}{4} < \rho_K^2 \tag{14}$$

Si on prend K+I=n et en opérant comme pour le cas réel, on aboutit à

$$\frac{1}{4}\rho_o^2 - \frac{2}{3}(1-1/4^n) < \rho_n^2 < \frac{1}{4^n}\cdot\rho_o^2 + (1-1/4^n) \tag{15}$$

et si à tous les itérés on a

$$1 < \rho_{n-1} < \rho_{n-2} < \cdots < \rho_1 < \rho_o \tag{16}$$

et dans l'hypothèse où

$$2^n \leqslant \rho_o < 2^{n+1} \qquad \rho_o^2 = \alpha 4^n \qquad 1 \leqslant \alpha < 4$$

on pourra donc écrire

$$\sqrt{\frac{I\varphi}{3}} < \rho_n < \sqrt{I7} \quad \text{et} \quad \sqrt{\frac{I}{3}} < \rho_n < \sqrt{5} \tag{I7}$$

ce qui traduit une convergence lineaire de facteur $\frac{I}{2}$ pour ρ .
En fait le module de Z_K decroit légérement plus vite qu'une progression geometrique de raison $\frac{I}{2}$ lorsque φ est grand (φ voisin de $\frac{\pi}{2}$) et un peu moins vite lorsqu'on se raproche de l'axe des réels.

Maintenant voyons la décroissance monotone de φ_k son évolution découle de de celle du gain $f(\rho) = \rho^2 - 1 / \rho^2 + 1$ et de $tg\varphi$, qui à priori ne reléve d'aucun processus simple tel celui mis en évidence pour ρ.

On peut cependant déjà conclure pour un Zo tel que $\rho_o \gg 1$ et $tg\varphi_o \gg 1$
L'evolution de Z_K se décompose en trois phase :

$$0 < \left| \frac{\rho_K^2 - 1}{\rho_K^2 + 1} \right| < 1 \quad \Longrightarrow \quad |\mathrm{tg}\varphi_{K+1}| < |\mathrm{tg}\varphi_K|$$

on a ainsi

$$0 < |\varphi_{K+1}| < |\varphi_K| \tag{13}$$

ce qui traduit la convergence de φ_K vers zero.

Etudions maintenant la decroissance de ρ en supposant $\rho \gg 1$ et si

$$\rho_K > 1$$

d'après (II) on peut écrire

$$\frac{1}{4}\rho_K^2 - \frac{1}{2} < \rho_{K+1}^2 < \frac{1}{4}\rho_K^2 + \frac{3}{4} < \rho_K^2 \tag{14}$$

Si on prend K+1=n et en opérant comme pour le cas réel, on aboutit à

$$\frac{1}{4}\rho_0^2 - \frac{2}{3}(1-1/4^n) < \rho_n^2 < \frac{1}{4^n}\cdot\rho_0^2 + (1-1/4^n) \tag{15}$$

et si à tous les itérés on a

$$1 < \rho_{n-1} < \rho_{n-2} < \cdots < \rho_1 < \rho_0 \tag{16}$$

et dans l'hypothèse où

$$2^n \leqslant \rho_0 < 2^{n+1} \qquad \rho_0^2 = \alpha\, 4^n \qquad 1 \leqslant \alpha < 4$$

on pourra donc écrire

$$\sqrt{\frac{10}{3}} < \rho_n < \sqrt{17} \qquad \text{et} \qquad \sqrt{\frac{1}{3}} < \rho_n < \sqrt{5} \tag{17}$$

ce qui traduit une convergence lineaire de facteur $\frac{1}{2}$ pour ρ.
En fait le module de Z_K decroit légérement plus vite qu'une progression geometrique de raison $\frac{1}{2}$ lorsque φ est grand (φ voisin de $\frac{\pi}{2}$) et un peu moins vite lorsqu'on se rapproche de l'axe des réels.

Maintenant voyons la décroissance monotone de φ_k son évolution découle de de celle du gain $f(\rho) = \rho^2 - 1 / \rho^2 + 1$ et de $\mathrm{tg}\varphi$, qui à priori ne reléve d'aucun processus simple tel celui mis en évidence pour ρ.

On peut cependant déjà conclure pour un Z_0 tel que $\rho_0 \gg 1$ et $\mathrm{tg}\varphi_0 \gg 1$
L'evolution de Z_K se décompose en trois phase :

Iº PHASE/ décroissance de ρ (du type progression geometrique de raison I/2) à ε approximativement constant.

2º PHASE/ décroissance de ε, ρ oscillant autour de I.

3º PHASE/ convergence du second ordre simultané de ε vers zero et ρ vers I.

3-EXTENSION AU CAS MATRICIEL

Soit A une matrice carrée (n.n)dont la partie réelle des valeurs propres non nulle: .Par analogie au cas scalaire deffnissons la suite:

$$A_{K+I} = I/2(A_K + A_K^{-1}) \qquad A_0 = A \tag{I8}$$

Supposons d'abord que A est diagonalisable,c'est à dire qu'on peut ecrire:

$$A_{K+I} = MD_{K+I}M^{-1} = I/2\ M(D_K + D_K^{-1})M^{-1} \quad \text{avec} \quad A_0 = MD_0M^{-1}$$

par identification on aura donc

$$D_{K+I} = I/2(D_K + D_K^{-1}) = \text{diag}(\ldots\ldots I/2(\lambda i + I/\lambda i)\ldots\ldots)$$

de sorte qu'on ait

$$D_\infty = \text{diag}(\ldots\ldots \text{signe}(\lambda i)\ldots\ldots)$$

En vertu de ce qu'on a vu au paragraphe precedent ,on peut écrire:

$$\text{signe}(A) = S = MD_\infty M^{-1} = \lim_{K \longrightarrow \infty} A_K$$

Maintenant supposons que notre matrice ne peut être mise que sous la forme de Jordan.Soit (J) un bloc de Jordan,et définissons la suite:

$$J_{K+I} = I/2(J_K + J_K^{-1}) \qquad \text{avec} \quad J_0 = J$$

$J_K =$ $\begin{bmatrix} a_i^K & a_2^K & \cdots & a_r^K \\ & \ddots & \ddots & \\ & & & a_2^K \\ & & & a_i^K \end{bmatrix}$ $\qquad J_0 =$ $\begin{bmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & & 1 \\ 0 & & & \lambda \end{bmatrix}$

1º PHASE/ décroissance de ρ (du type progression geometrique de raison I/2) à φ approximativement constant.

2º PHASE/ décroissance de φ, ρ oscillant autour de I.

3º PHASE/ convergence du second ordre simultané de φ vers zero et ρ vers I.

3-EXTENSION AU CAS MATRICIEL

Soit A une matrice carrée (n.n)dont la partie réelle des valeurs propres non nulle: .Par analogie au cas scalaire deffnissons la suite:

$$A_{K+I} = I/2(A_K + A_K^{-1}) \qquad A_0 = A \tag{I8}$$

Supposons d'abord que A est diagonalisable,c'est à dire qu'on peut ecrire:

$$A_{K+I} = MD_{K+I}M^{-1} = I/2\ M(D_K + D_K^{-1})M^{-1} \quad \text{avec} \quad A_0 = MD_0M^{-1}$$

par identification on aura donc

$$D_{K+I} = I/2(D_K + D_K^{-1}) = \text{diag}(\ldots\ldots I/2(\lambda i + I/\lambda i)\ldots\ldots)$$

de sorte qu'on ait

$$D_\infty = \text{diag}(\ldots\ldots \text{signe}(\lambda i)\ldots\ldots)$$

En vertu de ce qu'on a vu au paragraphe precedent ,on peut écrire:

$$\text{signe}(A) = S = MD_\infty M^{-1} = \lim_{K \longrightarrow \infty} A_K$$

Maintenant supposons que notre matrice ne peut être mise que sous la forme de Jordan.Soit (J) un bloc de Jordan,et définissons la suite:

$$J_{K+I} = I/2(J_K + J_K^{-1}) \qquad \text{avec} \quad J_0 = J$$

$J_K =$ $\begin{bmatrix} a_i^K & a_2^K & \cdots & a_r^K \\ & \ddots & \ddots & \\ & & & a_2^K \\ & & & a_i^K \end{bmatrix}$ $\qquad J_0 =$ $\begin{bmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & & 1 \\ 0 & & & \lambda \end{bmatrix}$

d'ou

$$J_0^{-1} = \begin{bmatrix} \frac{1}{\lambda} & -\frac{1}{\lambda^2} & \cdots & (-1)^{n-1}\frac{1}{\lambda^n} \\ & \ddots & \ddots & \\ & & & \frac{1}{\lambda} \end{bmatrix}$$

or

$$J_I = I/2(J_0 + J_0^{-1})$$

par identification on aura:

$$a'_1 = I/2(\lambda + I/\lambda) \qquad a'_2 = I/2(I - I/\lambda^2) \qquad a'_i = (-I)^{i-I} \cdot I/2\lambda^2 \quad i=3,\ldots r$$

Posons

$$J_K^{-1} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_r \\ & \ddots & \ddots & \alpha_2 \\ & & & \alpha_1 \end{bmatrix}$$

puisque $J_K J_K^{-1} = I$

$$\Longrightarrow \alpha_1 = I/a_1 \quad \alpha_2 = -(I/a_1)a_2\alpha_1 ; \alpha_r = -I/a_1 \cdot (a_2\alpha_{r-1} + \cdots + a_r\alpha_1)$$

Sachant que $J_{K+I} = I/2(J_K + J_K^{-1})$, il en découle:

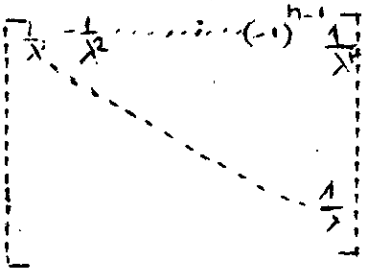
$$a_1^{K+1} = I/2(a_1^K + I/a_1^K) \qquad a_2^{K+1} = I/2 \cdot a_2^K (I - I/(a_1^K)^2)$$

$$a_r^{K+1} = I/2 \cdot a_r^K \left[I - I/(a_1^K)^2 \right] - (I/2a_1^K)(a_2\alpha_{r-1}^K + \cdots + a_{r-1}^K \cdot \alpha_2^K)$$

Cette structure montre clairement que lorsque a_1^K tend vers $\pm I$, alors a_i^K, $i=2,\ldots\ldots r$ tendent vers zero ,ce qui donne:

$$J_\infty = \begin{bmatrix} \text{signe}(\lambda) & & \\ & \ddots & \\ & & \text{signe}(\lambda) \end{bmatrix} = \text{signe}(J)$$

d'ou

$J_o^{-1}=$ 

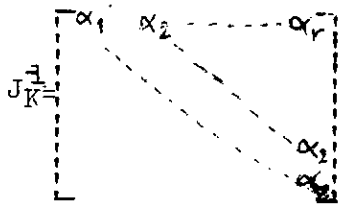
or

$$J_I=I/2(J_o+J_o^{-1})$$

par identification on aura:

$$a'_1=I/2(\lambda+I/\lambda) \qquad a'_2=I/2(I-I/\lambda^2) \qquad a'_i=(-I)^{i-I}\cdot I/2\lambda^2 \quad i=3,\ldots r$$

Posons

$J_K^{-1}=$ 

puisque $J_K J_K^{-1}=I$

$$\Longrightarrow \alpha_1=I/a_1 \quad \alpha_2=-(I/a_1)a_2\alpha_1 ; \alpha_r=-I/a_1\cdot(a_2\alpha_{r-1}+\cdots\cdots+a_r\alpha_1)$$

Sachant que $J_{K+I}=I/2(J_K+J_K^{-1})$, il en découle:

$$a_1^{K+1}=I/2(a_1^K+I/a_1^K) \qquad a_2^{K+1}=I/2\cdot a_2^K(I-I/(a_1^K)^2)$$

$$a_r^{K+1}=I/2\cdot a_r^K\left[I-I/(a_1^K)^2\right]-(I/2a_1^K)(a_2\alpha_{r-1}^K+\cdots\cdots+a_{r-1}^K\cdot\alpha_2^K)$$

Cette structure montre clairement que lorsque a_1^K tend vers $\pm I$, alors a_i^K, $i=2,\ldots\ldots r$ tendent vers zero ,ce qui donne:

$$J_\infty=\begin{bmatrix}\text{signe}(\lambda) & & 0\\ & \ddots & \\ 0 & & \text{signe}(\lambda)\end{bmatrix}=\text{signe}(J)$$

3-I-ALGORITHME FINI POUR LE CALCUL DE S=signe(A)

Nous allons considérer dans ce paragraphe un cas particulier à savoir les matrices régulières à spectre réel ; on aura donc la suite:

$$\lambda_{K+1}=\frac{I}{2}\left(\lambda_K+\frac{I}{\lambda_K}\right) \qquad \text{avec} \quad \lambda_0=\max\left(|\lambda_{max}|,\left|\frac{1}{\lambda_{min}}\right|\right)$$

et le nombre d'itérés pour calculer S sera donné comme pour le cas réel c'est à dire:

$$N=n+r$$

Soit

$$s=|\lambda_{max}| \qquad t=|\lambda_{min}| \qquad \text{et} \quad \alpha=\frac{I}{\sqrt{st}}$$

définissons la transformation

$$A^*=\alpha A$$

de façon à avoir

$$s^*=\left(\frac{s}{t}\right)^{1/2} \; ; \quad t^*=\left(\frac{t}{s}\right)^{1/2} \Longrightarrow \quad s^*=\frac{I}{t^*} \qquad (I9)$$

Appliquons alors une itération de (I8) pour K=0, il vient:

$$A_1^*=\frac{I}{2}\left(A_0^*+(A_0^*)^{-1}\right)$$

d'après (I9), A_0^* a deux valeurs propres de module inverse, alors A_1^* aura deux valeurs propres de même module maximum:

$$\lambda_{max}(A_1^*)=\frac{I}{2}(s^*+t^*)=\frac{I}{2}\alpha(s+t)$$

Plus généralement posons:

$$\left.\begin{aligned} &s_K=\left|\lambda_{max}(A_K^*)\right| \; , \quad t_K=\left|\lambda_{min}(A_K^*)\right| \; , \quad \alpha_K=\frac{I}{\sqrt{s_K t_K}} \\ &\text{et} \\ &A_{K+I}^*=\frac{I}{2}\left[\alpha_K A_K^*+\frac{I}{\alpha_K}(A_K^*)^{-1}\right] \qquad A_0^*=A_0=A \end{aligned}\right\} \qquad (20)$$

En supposant que A_K possède au moins (K+I) valeurs propres de même module maximum. L'axiome de recurrence permet d'affirmer, qu'en faisant une itération (c-à-d A_{K+I}), celle-ci conduit à au moins (K+2) valeurs propres de même module maximum:

3-I-ALGORITHME FINI POUR LE CALCUL DE S=signe(A)

Nous allons considérer dans ce paragraphe un cas particulier à savoir les matrices régulières à spectre réel ; on aura donc la suite:

$$\lambda_{K+1}=\frac{I}{2}\left(\lambda_K+\frac{I}{\lambda_K}\right) \qquad \text{avec} \quad \lambda_0=\max\left(|\lambda_{max}|,\left|\frac{1}{\lambda_{min}}\right|\right)$$

et le nombre d'itérés pour calculer S sera donné comme pour le cas réel c'est à dire:

$$N=n+r$$

Soit

$$s=|\lambda_{max}| \qquad t=|\lambda_{min}| \qquad \text{et} \quad \alpha=\frac{I}{\sqrt{st}}$$

définissons la transformation

$$A^*=\alpha A$$

de façon à avoir

$$s^*=\left(\frac{s}{t}\right)^{\frac{1}{2}} \; ; \quad t^*=\left(\frac{t}{s}\right)^{\frac{1}{2}} \Longrightarrow \quad s^*=\frac{I}{t^*} \tag{I9}$$

Appliquons alors une itération de (I8) pour K=0, il vient:

$$A_1^*=\frac{I}{2}\left(A_0^*+(A_0^*)^{-1}\right)$$

d'après (I9), A_0^* a deux valeurs propres de module inverse, alors A_1^* aura deux valeurs propres de même module maximum:

$$\lambda_{max}(A_1^*)=\frac{I}{2}(s^*+t^*)=\frac{I}{2}\alpha(s+t)$$

Plus généralement nous avons:

$$\left.\begin{aligned} s_K&=\left|\lambda_{max}(A_K^*)\right| \quad , \quad t_K=\left|\lambda_{min}(A_K^*)\right| \quad , \quad \alpha_K=\frac{I}{\sqrt{s_K t_K}} \\ \text{et} \qquad A_{K+I}^*&=\frac{I}{2}\left[\alpha_K A_K^*+\frac{I}{\alpha_K}(A_K^*)^{-1}\right] \qquad A_0^*=\alpha_0 A \end{aligned}\right\} \tag{20}$$

En supposant que A_K possède au moins (K+I) valeurs propres de même module maximum. L'axiome de recurrence permet d'affirmer, qu'en faisant une itération (c-à-d A_{K+I}), celle-ci conduit à au moins (K+2) valeurs propres de même module maximum:

$$\left|\lambda max(\Lambda^*_{K+I})\right| = \frac{I}{2}\cdot\frac{\left|\lambda max(\Lambda^*_K)\right| + \left|\lambda min(\Lambda^*_K)\right|}{\sqrt{\left|\lambda max(\Lambda^*_K).\lambda min(\Lambda^*_K)\right|}}$$

on peut donc affirmer qu'il existe $l \leqslant n-I$ tel que Λ^*_l possède n valeurs propres de même module.

On peut alors écrire:

$$S=signe(\Lambda)=\frac{\Lambda^*_l}{\left|\lambda(\Lambda^*_l)\right|} \qquad (2I)$$

Ainsi il est possible d'énoncer le théorème suivant:

Soit Λ (n.n) régulière à spectre réel, S=signe(Λ) est obtenue par l'algorithme fini:

$$\Lambda^*_{K+I}=\frac{I}{2}\left[\alpha_K \Lambda^*_K+\frac{I}{\alpha_K}(\Lambda^*_K)^{-1}\right] \qquad \Lambda^*_0=\Lambda$$

$$\alpha_K=\frac{I}{\sqrt{\left|\lambda max(\Lambda^*_K).\lambda min(\Lambda^*_K)\right|}} \qquad (22)$$

$$S=\frac{I}{\left|\lambda(\Lambda^*_l)\right|}.\Lambda^*_l$$

3-2-ALGORITHME DE NEWTON ACCELERE

Introduction

La methode qu'on a vue précédement n'est malheuresement utilisable que si Λ est diagonalisable ;à ceci il faut ajouter le fait le plus fondamentale que l'algorithme ne s'étend pas au cas des matrices à spectre complexe.

Il est néanmoins possible d'élaborer à partir duthéorême vu précédement une procédureitérative répondant à ces deux objéctifs et qui constituera ce qu'on appellera une version accélérée de la methòde de Newton.

Algorithme de Newton accéléré

Considérons d'abord le problème des valeurs propres complexes.

Soient Z_1 et Z_2 deux de ces valeurs.Définissons alors la transformation

$$\left|\lambda_{max}(A^*_{K+I})\right| = \frac{I}{2}\cdot\frac{\left|\lambda_{max}(A^*_K)\right| + \left|\lambda_{min}(A^*_K)\right|}{\sqrt{\left|\lambda_{max}(A^*_K).\lambda_{min}(A^*_K)\right|}}$$

on peut donc affirmer qu'il existe $1 \leqslant n-I$ tel que A^*_1 possède n valeurs propres de même module.

On peut alors écrire:

$$S=signe(A)=\frac{A^*_1}{\left|\lambda(A^*_1)\right|} \qquad (2I)$$

Ainsi il est possible d'énoncer le théorème suivant:

> Soit A (n.n) régulière à spectre réel, S=signe(A) est obtenue par l'algorithme fini:

$$A^*_{K+I}=\frac{I}{2}\left[\alpha_K A^*_K+\frac{I}{\alpha_K}(A^*_K)^{-I}\right] \qquad A^*_0=A$$

$$\alpha_K=\frac{I}{\sqrt{\left|\lambda_{max}(A^*_K).\lambda_{min}(A^*_K)\right|}} \qquad (22)$$

$$S=\frac{I}{\left|\lambda(A^*_1)\right|}.A^*_1$$

3-2-ALGORITHME DE NEWTON ACCELERE

Introduction

La methode qu'on a vue précédement n'est malheuresement utilisable que si A est diagonalisable ;à ceci il faut ajouter le fait le plus fondamentale que l'algorithme ne s'étend pas au cas des matrices à spectre complexe.

Il est néanmoins possible d'élaborer à partir duthéorême vu précédement une procédureitérative répondant à ces deux objectifs et qui constituera ce qu'on appellera une version accélérée de la methode de Newton.

Algorithme de Newton accéléré

Considérons d'abord le problème des valeurs propres complexes.

Soient Z_1 et Z_2 deux de ces valeurs.Définissons alors la transformation

$$\overset{*}{Z}i = \alpha Z \qquad i=1,2$$

puis

$$Z' = \frac{1}{2}\left(Zi + \frac{1}{Zi}\right) \qquad \text{et} \qquad \widetilde{Z} = \frac{1}{2}\left(\overset{*}{Z}i + \frac{1}{\overset{*}{Z}i}\right)$$

on a donc

$$\rho'^2_i = |Zi|^2 = \frac{1}{4}\left(\rho i + \frac{1}{\rho i}\right)^2 - \sin^2 \theta i$$

$$\widetilde{\rho}^2_i = |\widetilde{Z}i|^2 = \frac{1}{4}\left(\alpha \rho i + \frac{1}{\alpha \rho i}\right)^2 - \sin^2 \theta_i$$

$$tg\widetilde{\theta} i = \frac{\alpha^2 \rho^2_{i-1}}{\alpha^2 \rho^2_{i+1}} . tg\theta i$$

$$tg\theta' i = \frac{\rho^2_{i-1}}{\rho^2_{i+1}} . tg\theta i$$

On voit clairement(en général)qu'il n'existe pas de α fonction de $\rho i, \theta i$ tel que $\widetilde{\theta}_1 = \widetilde{\theta}_2$ et $\widetilde{\rho}_1 = \widetilde{\rho}_2$ soient verifiés simultanément.
Une premiere approximation consiste par extention du cas réel à faire abstraction des arguments θ_1 et θ_2 et ne raisonner que sur les modules ρ_1 et ρ_2.
Le mieux qu'on puisse faire est d'égaliser la contribution de ρ_1 et ρ_2 dans les modules $\widetilde{\rho}_{1;2}$ d'où l'on prend:

$$\alpha = \frac{1}{\sqrt{\rho_1 \rho_2}} \tag{23}$$

d'où l'on a

$$|\widetilde{\rho}_1 - \widetilde{\rho}_2| < |\rho'_1 - \rho'_2|$$

$$0 < |\widetilde{\rho}^2_1 - \widetilde{\rho}^2_2| < 1 \tag{24}$$

et une égalisation du facteur de reduction de tg θ qui découle des relations

$$\rho_1^* \rho_2^* = \alpha\rho_1 \cdot \alpha \rho_2 = 1$$

$$\Psi(\rho) = \frac{\rho^2 - 1}{\rho^2 + 1} = -\Psi\left(\frac{1}{\rho}\right)$$

$$\left.\begin{aligned} tg\widetilde{\theta}_1 &= \Psi(\rho_1^*)\, tg\theta_1 \\ tg\widetilde{\theta}_2 &= -\Psi(\rho_1^*)\, tg\theta_2 \end{aligned}\right\} \tag{25}$$

$$\overset{*}{Z}i = \alpha Z \qquad i=1,2$$

puis

$$Z' = \frac{1}{2}\left(Zi + \frac{1}{Zi}\right) \qquad \text{et} \qquad \widetilde{Z} = \frac{1}{2}\left(\overset{*}{Z}i + \frac{1}{\overset{*}{Z}i}\right)$$

on a donc

$$\rho'^2_i = |Zi|^2 = \frac{1}{4}\left(\rho i + \frac{1}{\rho i}\right)^2 - \sin^2 \theta i$$

$$\widetilde{\rho}^2_i = |\widetilde{Zi}|^2 = \frac{1}{4}\left(\alpha \rho i + \frac{1}{\alpha \rho i}\right)^2 - \sin^2 \theta_i$$

$$\operatorname{tg} \widetilde{\theta} i = \frac{\alpha^2 \rho^2_{i-1}}{\alpha^2 \rho^2_{i+1}} \operatorname{tg} \theta i$$

$$\operatorname{tg} \theta' i = \frac{\rho^2_{i-1}}{\rho^2_{i+1}} \operatorname{tg} \theta i$$

On voit clairement (en général) qu'il n'existe pas de α fonction de $\rho i, \theta i$ tel que $\widetilde{\theta}_1 = \widetilde{\theta}_2$ et $\widetilde{\rho}_1 = \widetilde{\rho}_2$ soient verifiés simultanément.

Une premiere approximation consiste par extention du cas réel à faire abstraction des arguments θ_1 et θ_2 et ne raisonner que sur les modules ρ_1 et ρ_2.

Le mieux qu'on puisse faire est d'égaliser la contribution de ρ_1 et ρ_2 dans les modules $\widetilde{\rho}_{1;2}$ d'où l'on prend:

$$\alpha = \frac{1}{\sqrt{\rho_1 \rho_2}} \tag{23}$$

d'où l'on a

$$\left|\widetilde{\rho}_1 - \widetilde{\rho}_2\right| < \left|\rho'_1 - \rho'_2\right|$$

$$0 < \left|\widetilde{\rho}^2_1 - \widetilde{\rho}^2_2\right| < 1 \tag{24}$$

et une égalisation du facteur de reduction de tg θ qui découle des relations

$$\rho_1^* \rho_2^* = \alpha \rho_1 \cdot \alpha \rho_2 = 1$$

$$\Psi(\rho) = \frac{\rho^2 - 1}{\rho^2 + 1} = -\Psi\left(\frac{1}{\rho}\right)$$

$$\left.\begin{aligned} \operatorname{tg} \widetilde{\theta}_1 &= \Psi(\rho_1^*) \operatorname{tg} \theta_1 \\ \operatorname{tg} \widetilde{\theta}_2 &= -\Psi(\rho_1^*) \operatorname{tg} \theta_2 \end{aligned}\right\} \tag{25}$$

CONCLUSION

On peut dire que la transformation α n'annule plus l'ecart $\tilde{Z}_1 - \tilde{Z}_2$, mais contribue à réduire l'ecart en norme de $\tilde{Z}_1$ et $\tilde{Z}_2$ par rapport à une itération de Newton(Z'_1 et Z'_2) et égalise en valeur absolue le facteur de réduction de la tangente des arguments φ_1 et φ_2 .
Donc si on fait jouer à $\lambda max(A^*_K)$ et $\lambda min(A^*_K)$ le rôle de Z_1 et Z_2 dans l'algorithme précedent.En raison des proprietes d'invariance de la fonction signe on a:

$$\forall K \text{ signe } (A^*_K) = \text{signe}(A_K) = \text{signe } (A)$$

c'est à dire

$$\lim_{K \to \infty} A^*_K = \lim_{K \to \infty} A_K = S$$

Ceci étant,on peut dire que l'algorithme accélérè se comporte dans son stade ultime exactement comme l'algorithme de Newton,l'accélération de la convergence au sens de la reduction du nombre d'itéré pour atteindre S avec une precision donnée,intervient donc essentiellement dans l'evolution de A^*_K correspendant aux phases 1et2 du cas scalaire.

Pour éviter le calcul des valeurs propres on est amené à faire l'approximation suivante:

$$\alpha_K = \sqrt{\frac{\| A^{*-1}_K \|}{\| A^*_K \|}} \qquad (27)$$

ceci est légitime car:

$$|\lambda max(A)| \leqslant \|A\| \qquad \text{et} \qquad |\lambda min(A)| \geqslant 1/\|A^{-1}\|$$

L'algorithme de Newton accéléré est donc en définitive,défini par la procédure suivante.

$$A^*_0 = A$$

$$A^*_{K+1} = \frac{1}{2}\left[\alpha_K A^*_K + \frac{1}{\alpha_K} A^{*-1}_K\right] \qquad (28)$$

$$\alpha_K = \sqrt{\frac{\|A_K^{-1}\|}{\|A_K\|}}$$

CONCLUSION

On peut dire que la transformation α n'annule plus l'ecart $\tilde{Z}_1 - \tilde{Z}_2$, mais contribue à réduire l'ecart en norme de $\tilde{Z}_1$ et $\tilde{Z}_2$ par rapport à une itération de Newton(Z'_1 et Z'_2) et égalise en valeur absolue le facteur de réduction de la tangente des arguments φ_1 et φ_2 .
Donc si on fait jouer à $\lambda max(A_K^*)$ et $\lambda min(A_K^*)$ le rôle de Z_1 et Z_2 dans l'algorithme précedent. En raison des proprietes d'invariance de la fonction signe on a:

$$\forall K \text{ signe } (A_K^*) = \text{signe}(A_K) = \text{signe } (A)$$

c'est à dire

$$\lim_{K \longrightarrow \infty} A_K^* = \lim_{K \longrightarrow \infty} A_K = S$$

Ceci étant, on peut dire que l'algorithme accéléré se comporte dans son stade ultime exactement comme l'algorithme de Newton, l'accélération de la convergence au sens de la reduction du nombre d'itéré pour atteindre S avec une precision donnée, intervient donc essentiellement dans l'evolution de A_K^* correspendant aux phases 1et2 du cas scalaire.

Pour éviter le calcul des valeurs propres on est amené à faire l'approximation suivante:

$$\alpha_K = \sqrt{\frac{\| A_K^{*-1} \|}{\| A_K^* \|}} \qquad (27)$$

ceci est légitime car:

$$|\lambda max(A)| \leq \|A\| \qquad \text{et} \qquad |\lambda min(A)| \geq 1/\|A^{-1}\|$$

L'algorithme de Newton accéléré est donc en définitive, défini par la procédure suivante.

$$A_0^* = A$$

$$A_{K+1}^* = \frac{1}{2}\left[\alpha_K A_K^* + \frac{1}{\alpha_K} A_K^{*-1}\right] \qquad (28)$$

$$\alpha_K = \sqrt{\frac{\|A_K^{-1}\|}{\|A_K\|}}$$

3-3-IMPLEMENTATION

On a vu précédemment que l'algorithme de Newton accéléré ,était défini par

$$A_0^* = A$$

$$A_{K+1}^* = \frac{1}{2}\left[\alpha_K A_K^* + \frac{1}{\alpha_K} A_K^{*-1}\right]$$

$$\alpha_K = \sqrt{\frac{\|A_K^{*-1}\|}{\|A_K^*\|}}$$

Nous remarquons que le calcul de α_K passe par celui des normes de matrices; ainsi s'impose un choix judicieux de ces dernières.

3-3-1-Choix de la norme

On a par définition

$$\|A\|_p = \max_{\|x\|_p=1} \|Ax\|_p$$

ce qui donne

$p=1$ $\quad \|A\|_1 = \max_j \left(\sum_{i=1}^{n} |a_{ij}|\right)$

$p=2$ $\quad \|A\|_2 = \sqrt{\max(A^T A)}$

$p=\infty$ $\quad \|A\|_\infty = \max_i \left(\sum_{j=1}^{n} |a_{ij}|\right)$

$p=F$ $\quad \|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$

on en déduit que:

$$\|A_1\| = \|A^T\|_\infty$$

$$\|A_2\| \leq \|A\|_F \leq \begin{cases} \sqrt{n}\cdot\|A\|_1 \\ \sqrt{n}\cdot\|A\|_\infty \end{cases}$$

$$\frac{1}{\sqrt{n}}\cdot\|A\|_p \leq \|A\|_F \leq \sqrt{n}\cdot\|A\|_p$$

notre choix doit ainsi satisfaire deux critères:

- le coût calcul
- la recherche du plus petit majorant de $|\lambda_{max}|$

On remarque que le plus petit majorant de $|\lambda_{max}|$ est $\|A\|_2$, tandis que le critère coût calcul est satisfait par $\|A\|_1$ ou $\|A\|_\infty$.
Un compromis sera ainsi donné par :

$$\|A\| = \min(\|A\|_1, \|A\|_\infty) \qquad (29)$$

3-3-IMPLEMENTATION

On a vu précédement que l'algorithme de Newton accéléré était défini par

$$A^*_0 = A$$

$$A^*_{K+1} = \frac{1}{2}\left[\alpha_K A^*_K + \frac{1}{\alpha_K} A^{-1}_K\right]$$

$$\alpha_K = \sqrt{\frac{\|A^{-1}_K\|}{\|A^*_K\|}}$$

Nous remarquons que le calcul de α_K passe par celui des normes de matrices; ainsi s'impose un choix judicieux de ces dernières.

3-3-1-Choix de la norme

On a par définition

$$\|A\|_p = \max_{\|x\|_p = 1} \|Ax\|_p$$

ce qui donne

p=1	$\|A\|_1 = \max_j \left(\sum_{i=1}^{n} \|a_{ij}\|\right)$
p=2	$\|A\|_2 = \sqrt{\max(A^T A)}$
p=∞	$\|A\|_\infty = \max_i \left(\sum_{j=1}^{n} \|a_{ij}\|\right)$
p=F	$\|A\|_F = \sqrt{\sum_{i,j} a^2_{ij}}$

on en déduit que:

$$\|A_1\| = \|A^T\|_\infty$$

$$\|A_2\| \leq \|A\|_F \leq \begin{cases} \sqrt{n} \cdot \|A\|_1 \\ \sqrt{n} \cdot \|A\|_\infty \end{cases}$$

$$\frac{1}{\sqrt{n}} \cdot \|A\|_p \leq \|A\|_F \leq \sqrt{n} \cdot \|A\|_p$$

notre choix doit ainsi satisfaire deux critères:

- le coût calcul
- la recherche du plus petit majorant de $|\lambda_{max}|$

On remarque que le plus petit majorant de $|\lambda_{max}|$ est $\|A\|_2$, tandis que le critère coût calcul est satisfait par $\|A\|_1$ ou $\|A\|_\infty$.

Un compromis sera ainsi donné par :

$$\|A\| = \min(\|A\|_1, \|A\|_\infty) \qquad (29)$$

3-3-2-Critère d'arrêt

Il convient de définir un critère d'arrêt tel que

$$\lim_{K\to\infty} \Lambda_K^* = S \qquad \begin{cases} \alpha_K \longrightarrow 1 \\ \| \Lambda_{K+1}^* - \Lambda_K^* \| \longrightarrow 0 \\ \frac{1}{n}.\mathrm{trace}(\Lambda_K^*)^2 \longrightarrow 1 \end{cases}$$

On en déduit les conditions suivantes classées par coût croissant dont chacune peut constituer un test de convergence :

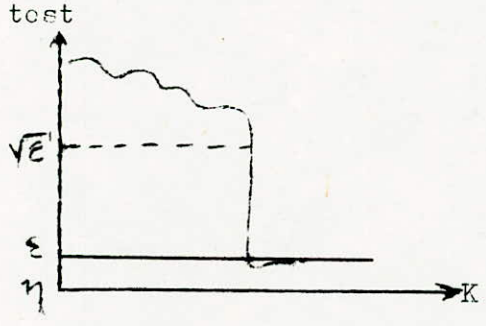
$$1-|\alpha_K - 1| \leqslant \varepsilon$$
$$2-\ \| \Lambda_{K+1}^* - \Lambda_K^* \| \leqslant \varepsilon \| \Lambda_K^* \|$$
$$3-\left| \frac{1}{n}.\mathrm{trace}(\Lambda_K^*)^2 - 1 \right| < \varepsilon \qquad \varepsilon < \eta$$

η étant la précision machine

Remarque: $\alpha_0 = 1$, pour une matrice orthogonale ; donc le 1º test ne s'applique pas à la première itération car $\| \Lambda_1^* - \Lambda_0^* \| \neq 0$.

Le 1º et 3º tests sont équivalents. En raison du coût calcul du 3º test(n^3 multiplications) en comparaison avec le 1º(n^2), on a ainsi éliminé le 3º. On aura le test suivant

$$\text{"test"} = \max\left(|\alpha_K - 1|, \frac{\| \Lambda_{K+1}^* - \Lambda_K^* \|}{\| \Lambda_K^* \|} \right) \qquad \text{fin} \qquad (30)$$



La variable test évolue d'une façon chaotique avant de décroître jusqu'à une valeur de l'ordre de grandeur de η. La difficulté réside ainsi dans la fixation d'un seuil ε. Or la convergence finale a une nature quadratique ; donc la variable test décroît très rapidement à partir d'un certain moment pour arriver sur une asymptote basse ε dépendant de η et de A.

3-3-2-Critère d'arrêt

Il convient de définir un critère d'arrêt tel que

$$\lim_{K\to\infty} \Lambda_K^* = S \qquad \begin{cases} \alpha_K \longrightarrow 1 \\ \left\| \Lambda_{K+1}^* - \Lambda_K^* \right\| \longrightarrow 0 \\ \frac{1}{n}.\mathrm{trace}(\Lambda_K^*)^2 \longrightarrow 1 \end{cases}$$

On en déduit les conditions suivantes classées par coût croissant dont chacune peut constituer un test de convergence :

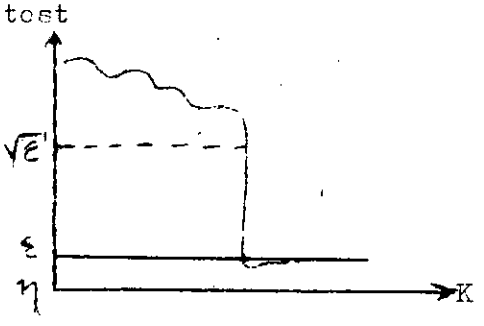
$$1-\left|\alpha_K - 1\right| \leqslant \varepsilon'$$
$$2-\ \left\|\Lambda_{K+1}^* - \Lambda_K^*\right\| \leqslant \varepsilon \left\|\Lambda_K^*\right\| \qquad \varepsilon < \eta$$
$$3-\left|\frac{1}{n}.\mathrm{trace}(\Lambda_K^*)^2 - 1\right| < \varepsilon$$

η étant la précision machine

Remarque : $\alpha_0 = 1$, pour une matrice orthogonale ; donc le 1º test ne s'applique pas à la première itération car $\left\| \Lambda_1^* - \Lambda_0^* \right\| \neq 0$.

Le 1º et 3º tests sont équivalents. En raison du coût calcul du 3º test(n^3 multiplications) en comparaison avec le 1º(n^2), on a ainsi éliminé le 3º. On aura le test suivant

$$\text{"test"} = \max\left(\left|\alpha_K - 1\right|, \frac{\left\| \Lambda_{K+1}^* - \Lambda_K^* \right\|}{\left\| \Lambda_K^* \right\|}\right) \qquad \text{fin} \qquad (30)$$



La variable test évolue d'une façon chaotique avant de décroître jusqu'à une valeur de l'ordre de grandeur de η. La difficulté réside ainsi dans la fixation d'un seuil ε. Or la convergence finale a une nature quadratique ; donc la variable test décroît très rapidement à partir d'un certain moment pour arriver sur une asymptote basse ε dépendant de η et de A.

Il faut donc s'assurer d'abord qu'on ait atteind la zone de convergence quadratique. Pour cela on introduit un seuil suffisament grand vis à vis de η pour qu'il soit atteint, par exemple $\sqrt{\eta}$.

La détection de l'asymptote se fait sur la base de deux valeurs consécutives du "test":

$$test_{K+1} \simeq test_K^2 \qquad \text{et} \qquad test_{K+1} \simeq test_K$$

d'où l'on aura le critère d'arrêt suivant:

$$\text{Si}\left[test_{K+1} < \sqrt{\eta} \quad \text{et} \quad test_{K+1} > \frac{1}{2} test_K\right] \text{ fin} \tag{31}$$

Enfin pour prévenir tout bouclage intempestif de l'algorithme, le nombre d'itérations doit être limité sur la base de la valeur maximum possible de N:

$$n(\varphi) = \left\lfloor \log_2 \max(\|A\|, \|A^{-1}\|) \right\rfloor$$

et

$$q(\varepsilon_0) < 1 + \left\lfloor \log_2 \left(\frac{c,64}{\eta}\right) \right\rfloor \qquad \text{avec} \quad \left(\tan\varepsilon_0 < \frac{2}{\pi.\eta} < \frac{c,64}{\eta}\right)$$

En conséquence on a implémenté le test supplémentaire:

$$N = \left\lfloor \log_2 \max(\|A\|, \|A^{-1}\|) \right\rfloor + q_m(\eta) + p(\eta) \tag{32}$$

où $q_m(\eta)$ et $p(\eta)$ sont des constantes machines introduites une fois pour toutes.

Il faut donc s'assurer d'abord qu'on ait atteind la zone de convergence quadratique. Pour cela on introduit un seuil suffisament grand vis à vis de η pour qu'il soit atteint, par exemple $\sqrt{\eta}$.
La détection de l'asymptote se fait sur la base de deux valeurs consécutives du "test":

$$test_{K+1} \simeq test_K^2 \qquad \text{et} \qquad test_{K+1} \simeq test_K$$

d'où l'on aura le critère d'arrêt suivant:

$$\text{Si}\left[test_{K+1} < \sqrt{\eta} \text{ et } test_{K+1} > \frac{1}{2} test_K\right] \text{ fin} \tag{31}$$

Enfin pour prévenir tout bouclage intempestif de l'algorithme, le nombre d'itérations doit être limité sur la base de la valeur maximum possible de N:

$$n(\varphi_0) = \left\lfloor \log_2 \max(\|A\|, \|A^{-1}\|) \right\rfloor$$

et

$$g(\varphi_0) < 1 + \left\lfloor \log_2 \left(\frac{c,64}{\eta}\right) \right\rfloor \qquad \text{avec} \quad \left(\operatorname{tg}\varphi_0 < \frac{2}{\pi.\eta} < \frac{c,64}{\eta}\right)$$

En conséquence on a implémenté le test supplémentaire:

$$N = \left\lfloor \log_2 \max(\|A\|, \|A^{-1}\|) \right\rfloor + q_m(\eta) + p(\eta) \tag{32}$$

où $q_m(\eta)$ et $p(\eta)$ sont des constantes machines introduites une fois pour toutes.

CHAPITRE II/

APPLICATION A LA RESOLUTION DES EQUATIONS DES EQUATIONS DE RICCATI

Partie A/

Equations de Riccati dans les problèmes de commande optimale avec critère quadratique.

Partie B/

Résolution de l'équation de Riccati dans le cas continu

Partie C/

Résolution de l'équation de Riccati dans le cas discret

CHAPITRE II/

APPLICATION A LA RESOLUTION DES EQUATIONS DES EQUATIONS DE RICCATI

Partie A/

Equations de Riccati dans les problèmes de commande optimale avec critère quadratique.

Partie B/

Résolution de l'équation de Riccati dans le cas continu

Partie C/

Résolution de l'équation de Riccati dans le cas discret

I N T R O D U C T I O N

Après avoir étudié le concept de fonction de matrice, nous procédons dans ce chapitre à son application pour la résolution des équations de Riccati apparaissant dans les problèmes d'optimalisation déterministe de la commande.

Pour cela, il nous a paru utile de diviser ce chapitre en trois parties:

- la première traîtera de la commande optimale des systèmes linéaires avec critère quadratique; ceci pour montrer l'origine des équations de Riccati.
- la deuxième aura pour but de donner un algorithme(utilisant la fonction signe) résolvant l'équation de Riccati dans le cas continu et donner l'organigramme.
- quand à la dernière partie, elle fera l'objet d'un algorithme (utilisant la fonction signe) résolvant l'équation de Riccati dans le cas discret et présentation de l'organigramme.

I N T R O D U C T I O N

Après avoir étudié le concept de fonction de matrice, nous procédons dans ce chapitre à son application pour la résolution des équations de Riccati apparaissant dans les problèmes d'optimalisation déterministe de la commande.

Pour cela, il nous a paru utile de diviser ce chapitre en trois parties:

- la première traîtera de la commande optimale des systèmes linéaires avec critère quadratique; ceci pour montrer l'origine des équations de Riccati.
- la deuxième aura pour but de donner un algorithme(utilisant la fonction signe) résolvant l'équation de Riccati dans le cas continu et donner l'organigramme.
- quand à la dernière partie, elle fera l'objet d'un algorithme (utilisant la fonction signe) résolvant l'équation de Riccati dans le cas discret et présentation de l'organigramme.

EQUATIONS DE RICCATI DANS LES PROBLEMES DE COMMANDE OPTIMALE AVEC CRITERE QUADRATIQUE

1-INTRODUCTION

La minimisation d'un critère quadratique constitue l'un des moyens de parvenir à la détermination d'une structure de commande par retour d'état pour les systèmes multidimensionnels.

En effet, un critère quadratique permet d'exprimer d'une manière convenable les qualités globales recherchées par la commande tant en assurant le meilleur compromis entre certaines performances, représentées par des termes de pondération faisant intervenir les sorties ou les variables d'état, et une économie d'énergie.

Un autre avantage, non moins négligeable de la méthode quadratique est de conduire à des développements mathématiques nombreux et puissants.

2-PRESENTATION DU PROBLEME

Soient le système linéaire invariant et gouvernable

$$\dot{X}(t) = A.X(t) + B.U(t)$$
$$Y(t) = C.X(t) \qquad (33)$$

X: état , U: commande , Y: sortie

et le coût (en supposant l'instant final infini)

$$J = \frac{1}{2} \int_0^{\infty} \left[\langle X(t) , Q.X(t) \rangle + \langle U(t) , R.U(t) \rangle \right] dt \qquad (34)$$

3-MINIMISATION DE L'HAMILTONIEN

L'hamiltonien associé à (33) et (34) est

$$H = \frac{1}{2} \langle X(t) , Q.X(t) \rangle + \frac{1}{2} \langle U(t) , R.U(t) \rangle$$
$$+ \langle p(t) , A.X(t) + B.U(t) \rangle$$

EQUATIONS DE RICCATI DANS LES PROBLEMES DE COMMANDE OPTIMALE AVEC CRITERE QUADRATIQUE

1-INTRODUCTION

La minimisation d'un critère quadratique constitue l'un des moyens de parvenir à la détermination d'une structure de commande par retour d'état pour les systèmes multidimensionnels.

En effet, un critère quadratique permet d'exprimer d'une manière convenable les qualités globales recherchées par la commande tant en assurant le meilleur compromis entre certaines performances, représentées par des termes de pondération faisant intervenir les sorties ou les variables d'état, et une économie d'énergie.

Un autre avantage, non moins négligeable de la méthode quadratique est de conduire à des développements mathématiques nombreux et puissants.

2-PRESENTATION DU PROBLEME

Soient le système linéaire invariant et gouvernable

$$\dot{X}(t) = A.X(t) + B.U(t)$$
$$Y(t) = C.X(t) \qquad (33)$$

X: état , U: commande , Y: sortie

et le coût (en supposant l'instant final infini)

$$J = \frac{1}{2} \int_0^{\infty} \left[\langle X(t) , Q.X(t) \rangle + \langle U(t) , R.U(t) \rangle \right] dt \qquad (34)$$

3-MINIMISATION DE L'HAMILTONIEN

L'hamiltonien associé à (33) et (34) est

$$H = \frac{1}{2} \langle X(t) , Q.X(t) \rangle + \frac{1}{2} \langle U(t) , R.U(t) \rangle$$
$$+ \langle p(t) , A.X(t) + B.U(t) \rangle$$

$$H = \frac{1}{2} \langle X(t) , Q.X(t) \rangle + \frac{1}{2} \langle U(t) , R.U(t) \rangle$$

$$+ \langle p(t) , A.X(t) \rangle + \langle p(t) , B.U(t) \rangle$$

où p est le Vecteur adjoint solution de l'équation :

$$\dot{p}(t) = -\nabla x \, H$$

d'où

$$\dot{p}(t) = -Q.X(t) - A^T p(t)$$

Le long de la trajectoire optimale, nous devons avoir

$$\nabla u H = 0 \quad \Longrightarrow \quad R.U + B^T.p(t) = 0$$

d'où

$$U(t) = -R^{-1} B^T p(t) \tag{35}$$

R^{-1} existe du moment que R est définie positive.

Remarque

$\nabla_u H =$ implique seulement l'éxistence d'un extrémum.Si de plus on a

$\frac{\partial^2 H}{\partial u^2}$ défini positif, alors cet extrémum s'identifie à un minimum.

Or $\frac{\partial^2 H}{\partial u^2} = R$ qui est définie positive

alors U(t) minimalise bien l'hamiltonien

4-EQUATIONS CANONIQUES REDUITES

En remplaçant U(t) par sa valeur dans (33), on aura

$$\dot{X}(t) = A.X(t) - BR^{-1} B^T p(t)$$

$$\dot{p}(t) = -Q.X(t) - A^T.p(t)$$

qui sont les équations canoniques réduites.

Soit $V(t) = BR^{-1} B^T$

on aura donc

$$H = \frac{1}{2} \langle X(t) , Q.X(t) \rangle + \frac{1}{2} \langle U(t) , R.U(t) \rangle$$

$$+ \langle p(t) , A.X(t) \rangle + \langle p(t) , B.U(t) \rangle$$

où p est le Vecteur adjoint solution de l'équation :

$$\dot{p}(t) = -\nabla x \; H$$

d'où

$$\dot{p}(t) = -Q.X(t) - A^T p(t)$$

Le long de la trajectoire optimale, nous devons avoir

$$\nabla u H = 0 \quad \Longrightarrow \quad R.U + B^T.p(t) = 0$$

d'où

$$U(t) = -R^{-1} B^T p(t) \qquad (35)$$

R^{-1} existe du moment que R est définie positive.

Remarque

$\nabla_u H =$ implique seulement l'éxistence d'un extremum.Si de plus on a

$\frac{\partial^2 H}{\partial u^2}$ défini positif, alors cet extremum s'identifie à un minimum.

Or

$\frac{\partial^2 H}{\partial u^2} = R$ qui est définie positive

alors U(t) minimalise bien l'hamiltonien

4-EQUATIONS CANONIQUES REDUITES

En remplaçant U(t) par sa valeur dans (33), on aura

$$X(t) = A.X(t) - BR^{-1} B^T p(t)$$

$$p(t) = -Q.X(t) - A^T.p(t)$$

qui sont les équations canoniques réduites.

Soit $V(t) = BR^{-1} B^T$

on aura donc

$$\begin{bmatrix} \dot{X}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} A & -V \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} X(t) \\ p(t) \end{bmatrix} \qquad (36)$$

On a donc un système de 2n équations différentielles homogènes. Il admet une solution unique lorsque sont fixées les 2n conditions aux limites. Or n conditions sont fournies par X(0), les autres sont données par $p(\infty) = 0 \quad (t_1 = \infty)$.

5-détermination de P

On voit aisément que p(t) et X(t) sont reliées par la relation suivante:

$$p(t) = P.X(t)$$

d'où

$$\frac{\partial}{\partial t} p(t) = \frac{\partial}{\partial t}(P.X(t)) = P.\dot{X}(t) \qquad (37)$$

D'après (36), on peut écrire

$$\dot{X}(t) = \left[A - VP \right] X(t)$$

$$\dot{p}(t) = \left[-Q - A^T P \right] X(t)$$

en remplaçant $\dot{X}(t)$ par sa valeur dans (37), on aboutit à:

$$\dot{p}(t) = P \left[A - VP \right] X(t)$$

$$\left[-Q - A^T P \right] X(t) = P \left[A - VP \right] X(t)$$

d'où finalement

$$PA + A^T P - PVP + Q = 0 \qquad (38)$$

c'est l'équation de Riccati matricièlle.

Ainsi pour avoir la commande optimale d'un système linéaire avec critère quadratique, il sffit de calculer la solution P de l'équation de Riccati pour avoir

$$U(t) = -R^{-1} B^T P.X(t) \qquad (39)$$

$$\begin{bmatrix} \dot{X}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} A & -V \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} X(t) \\ p(t) \end{bmatrix} \qquad (36)$$

On a donc un système de 2n équations différentielles homogènes. Il admet une solution unique lorsque sont fixées les 2n conditions aux limites. Or n conditions sont fournies par X(0), les autres sont données par $p(\infty) = 0 \quad (t_1 = \infty)$.

5-détérmination de P

On voit aisément que p(t) et X(t) sont reliées par la relation suivante:

$$p(t) = P.X(t)$$

d'où

$$\frac{\partial}{\partial t} p(t) = \frac{\partial}{\partial t}(P.X(t)) = P.\dot{X}(t) \qquad (37)$$

D'après (36), on peut écrire

$$\dot{X}(t) = \left[A - VP \right] X(t)$$

$$\dot{p}(t) = \left[-Q - A^T P \right] X(t)$$

en remplaçant $\dot{X}(t)$ par sa valeur dans (37), on aboutit à:

$$\dot{p}(t) = P \left[A - VP \right] X(t)$$

$$\left[-Q - A^T P \right] X(t) = P \left[A - VP \right] X(t)$$

d'où finalement

$$PA + A^T P - PVP + Q = 0 \qquad (38)$$

c'est l'équation de Riccati matricielle.

Ainsi pour avoir la commande optimale d'un système linéaire avec critère quadratique, il sffit de calculer la solution P de l'équation de Riccati pour avoir

$$U(t) = -R^{-1} B^T P.X(t) \qquad (39)$$

RESOLUTION DE L'EQUATION DE RICCATI (CAS CONTINU) A L'AIDE DE LA FONCTION SIGNE DE MATRICE

1-INTRODUCTION

Après avoir introduit le concept de la fonction signe de matrice et le problème de commande optimale avec critère quadratique, nous allons maintenant aborder l'application de ce concept à la résolution de l'équation de Riccati introduite en termes de commande optimale des systèmes continus.

2-DEFINITION DU PROBLEME

Soit le système invariant à instant final infini:

$$\dot{X}(t)=A.X(t) + B.U(t)$$

et le critère

$$J=\frac{1}{2}\int_0^{\infty}(U^TRU + X^TQX)dt. \qquad (40)$$

La commande par retour d'état linéaire minimisant J est

$$\hat{U}= -R^{-1}B^TP\,X(t)$$

avec P solution de l'équation de Riccati

$$PA+A^TP-PBR^{-1}B^TP+Q=0 \qquad (41)$$

satisfaisant les hypothèses suivantes

$$\left.\begin{array}{l} R^T=R \quad R>0 \qquad Q^T=Q \qquad Q\geqslant 0 \\ (A,B) \text{ stabilisable} \\ (C,A) \text{ détectable} \qquad C:\ C^TC=Q \end{array}\right\} \qquad (42)$$

On sait alors que P^T est aussi solution de (41) et que

$$P^T=P \qquad P\geqslant 0 \qquad P \text{ unique} \qquad (43)$$

$$\text{réel } \lambda_i[\tilde{A}]<0 \qquad \text{avec} \qquad \tilde{A}=A-BR^{-1}B^TP \qquad (44)$$

L'équation de Riccati peut alors s'écrire

$$P\tilde{A} + A^TP + Q = 0 \qquad (45)$$

RESOLUTION DE L'EQUATION DE RICCATI (CAS CONTINU) A L'AIDE DE LA FONCTION SIGNE DE MATRICE

1-INTRODUCTION

Après avoir introduit le concept de la fonction signe de matrice et le problème de commande optimale avec critère quadratique, nous allons maintenant aborder l'application de ce concept à la résolution de l'équation de Riccati introduite en termes de commande optimale des systèmes continus.

2-DEFINITION DU PROBLEME

Soit le système invariant à instant final infini:

$$\dot{X}(t)=A.X(t) + B.U(t)$$

et le critère

$$J=\frac{1}{2}\int_0^{\infty}(U^TRU + X^TQX)dt. \qquad (40)$$

La commande par retour d'état linéaire minimisant J est

$$\hat{U}= -R^{-1}B^TP\,X(t) \qquad (41)$$

avec P solution de l'équation de Riccati

$$PA+A^TP-PBR^{-1}B^TP+Q=0$$

satisfaisant les hypothèses suivantes

$$\left.\begin{array}{l} R^T=R \quad R>0 \qquad Q^T=Q \qquad Q\geqslant 0 \\ (A,B) \text{ stabilisable} \\ (C,A) \text{ détectable} \qquad C:\ C^TC=Q \end{array}\right\} \qquad (42)$$

On sait alors que P^T est aussi solution de (41) et que

$$P^T=P \qquad P\geqslant 0 \qquad P \text{ unique} \qquad (43)$$

$$\text{réel } \lambda_i\left[\tilde{A}\right]<0 \qquad \text{avec} \qquad \tilde{A}=A-BR^{-1}B^TP \qquad (44)$$

L'équation de Riccati peut alors s'écrire

$$P\tilde{A} + A^TP + Q = 0 \qquad (45)$$

Cela exprime donc que le système bouclé est asymptotiquement stable. Si on remplace l'hypothèse,(C, A)détéctable,par(C, A) reconstructible (observable), on aura P> 0

3-RESOLUTION

On sait que la résolution de (40) par la methode des variations,nous conduit à introduire un état adjoint λ tel que

$$\begin{bmatrix} \dot{X} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} X \\ \lambda \end{bmatrix}$$

et la relation

$$\lambda = P.X$$

conduit alors à l'équation de Riccati.

Soit donc la matrice

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \quad \begin{matrix} \uparrow \\ 2n \\ \downarrow \end{matrix} \qquad \begin{matrix} \\ \\ \end{matrix}$$
$$\leftarrow 2n \rightarrow \qquad \rightarrow \qquad (46)$$

et la transformation

$$Z = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \quad \begin{matrix} \uparrow \\ 2n \\ \downarrow \end{matrix}$$
$$\leftarrow 2n \rightarrow \qquad (47)$$

La matrice H(2n.2n) est hamiltonienne si et seulement si

$$H = ZH^TZ \qquad (48)$$

c'est à dire,si on partitionne H comme suit

$$H = \begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}$$

on aura

$$H_1 = -H_2^T \quad , \quad H_{12} = H_{12}^T \quad , \quad H_{21} = H_{21}^T$$

Il est ainsi aisé de verifier que H est une matrice hamiltonienne;de plus on a d'après(47):

Cela exprime donc que le système bouclé est asymptotiquement stable. Si on remplace l'hypothèse,(C, A)détéctable,par(C, A) reconstructible (observable), on aura $P \rangle 0$

3-RESOLUTION

On sait que la résolution de (40) par la methode des variations,nous conduit à introduire un état adjoint λ tel que

$$\begin{bmatrix} \dot{X} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} X \\ \lambda \end{bmatrix}$$

et la relation

$$\lambda = P.X$$

conduit alors à l'équation de Riccati.

Soit donc la matrice

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \quad \begin{matrix} \uparrow \\ 2n \\ \downarrow \end{matrix} \qquad \begin{matrix} \\ \\ \\ \end{matrix}$$
$$\leftarrow 2n \rightarrow \qquad\qquad (46)$$

et la transformation

$$Z = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \quad \begin{matrix} \uparrow \\ 2n \\ \downarrow \end{matrix}$$
$$\leftarrow 2n \rightarrow \qquad\qquad (47)$$

La matrice H(2n.2n) est hamiltonienne si et seulement si

$$H = ZH^TZ \qquad (48)$$

c'est à dire,si on partitionne H comme suit

$$H = \begin{bmatrix} H_1 & H_{12} \\ H_{21} & H_2 \end{bmatrix}$$

on aura

$$H_1 = -H_2^T \quad , \quad H_{12} = H_{12}^T \quad , \quad H_{21} = H_{21}^T$$

Il est ainsi aisé de verifier que H est une matrice hamiltonienne;de plus on a d'après(47):

$$Z^{-1} = Z^T = -Z$$

Remarques

-les valeurs propres d'une matrice hamiltonienne apparaissent par paires opposées.

-si $(v_1, v_2)^T$ est un vecteur propre droit, $(-v_2, v_1)^T$ est un vecteur propre gauche.

Maintenant on est en mesure d'énoncer le théorême suivant:

THEOREME: étant donné le systeme invariant à temps final infini défini par les equations(40), et les hypothèses(42), alors la matrice

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}$$

est telle que

1-réel $\lambda[H] \neq 0$

2-$\lambda i[\tilde{A}] = \lambda i[A - BR^{-1}B^T\Gamma] = \lambda i[H]$ i:réel $\lambda i < 0$

3-

$$H = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} J & 0 \\ 0 & -J \end{bmatrix} \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix}$$

4- $P = W_2 W_{12}^{-1} = -V_{12}^{-1} V_1$

5- $\tilde{A} = W_{12}(-J)W_{12}^{-1}$

où la 3troisième relation exprime la décomposition de H sous forme de Jordan, et la cinquième relation celle qui en découle pour $\tilde{A}$ (système bouclé).

Remarque: la démonstration se trouve en annexe III

Ainsi d'après ce théorême on a une solution explicite pour

$$\Gamma = W_2 W_{12}^{-1} = -V_{12}^{-1} V_1$$

Mais si H n'est pas diagonalisable le problème est pratiquement insoluble. Pour contourner cette difficulté, on va aborder le problème comme suit:

Soit

$$H = U \begin{bmatrix} \tilde{A} & 0 \\ 0 & -\tilde{A} \end{bmatrix} U^{-1}$$

$$Z^{-1} = Z^T = -Z$$

Remarques

-les valeurs propres d'une matrice hamiltonienne apparaissent par paires opposées.

-si $(v_1, v_2)^T$ est un vecteur propre droit, $(-v_2, v_1)^T$ est un vecteur propre gauche.

Maintenant on est en mesure d'énoncer le théorême suivant:

THEOREME: étant donné le systeme invariant à temps final infini défini par les equations(40), et les hypothèses(42), alors la matrice

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}$$

est telle que

1-réel $\lambda[H] \neq 0$

2-$\lambda i[\tilde{A}] = \lambda i[A - BR^{-1}B^T P] = \lambda i[H]$ i:réel $\lambda i < 0$

3-

$$H = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} J & 0 \\ 0 & -J \end{bmatrix} \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix}$$

4- $P = W_2 W_{12}^{-1} = -V_{12}^{-1} V_1$

5- $\tilde{A} = W_{12}(-J) W_{12}^{-1}$

où la 3troisième relation exprime la décomposition de H sous forme de Jordan, et la cinquième relation celle qui en découle pour $\tilde{A}$ (systèmo bouclé).

Remarque: la démonstration se trouve en annexe III

Ainsi d'après ce théorême on a une solution explicite pour

$$P = W_2 W_{12}^{-1} = -V_{12}^{-1} V_1$$

Mais si H n'est pas diagonalisable, le problème est pratiquement insoluble. Pour contourner cette difficulté, on va aborder le problème comme suit:

Soit

$$H = U \begin{bmatrix} \tilde{A} & 0 \\ 0 & -\tilde{A} \end{bmatrix} U^{-1}$$

d'où

$$S = \text{signe}(H) = U \begin{bmatrix} \text{signe}(\tilde{A}) & 0 \\ 0 & \text{signe}(-\tilde{A}) \end{bmatrix} U^{-1} \qquad (49)$$

$\tilde{A}$ etant asymptotiquement stable, donc

réel $\lambda i[\tilde{A}] < 0$

d'où

$$\text{signe}(\tilde{A}) = -1 \qquad S = U \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} U^{-1} \qquad (50)$$

Introduisons alors la matrice

$$F = \frac{1}{2}(1 + S) \qquad (51)$$

c'est à dire

$$F = U \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} U^{-1}$$

en remplaçant U par sa valeur, il vient

$$F = \begin{bmatrix} VP & -V \\ -(1-PV)P & 1-PV \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}$$

d'où finalement par identification, on aboutit à

$$P = -F_{12}^{-1} \cdot F_1 \qquad (52)$$

On obtient ainsi une expression explicite pour P comparable à celle du théorême avec l'avantage d'éliminer la factorisation de Jordan.
Toutefois ce résultat n'est valable que si V est régulière.
D'après l'equation de Lyapunov(annexe), V est à priori symétrique non négative. La matrice $\tilde{A}$ etant stable, V est donnée par

$$V = \int_0^\infty e^{\tilde{A}^T t} B R^{-1} B^T e^{\tilde{A} t} . dt$$

Or V doit être régulière, donc définie positive. Ceci n'est vrai que si la paire $(\tilde{A}, B)$ est commandable, c'est à dire la paire (A, B) commandable

d'où

$$S = \text{signe}(H) = U \begin{bmatrix} \text{signe}(\tilde{A}) & 0 \\ 0 & \text{signe}(-\tilde{A}) \end{bmatrix} U^{-1} \qquad (49)$$

$\tilde{A}$ etant asymptotiquement stable, donc

réel $\lambda i[\tilde{A}] < 0$

d'où

$$\text{signe}(\tilde{A}) = -1 \qquad S = U \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} U^{-1} \qquad (50)$$

Introduisons alors la matrice

$$F = \frac{1}{2}(1 + S) \qquad (51)$$

c'est à dire

$$F = U \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} U^{-1}$$

en remplaçant U par sa valeur, il vient

$$F = \begin{bmatrix} VP & -V \\ -(1-PV)P & 1-PV \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}$$

d'où finalement par identification, on aboutit à

$$P = -F_{12}^{-1} \cdot F_1 \qquad (52)$$

On obtient ainsi une expression explicite pour P comparable à celle du théorême avec l'avantage d'éliminer la factorisation de Jordan.
Toutefois ce résultat n'est valable que si V est régulière.
D'après l'equation de Lyapunov(annexe), V est à priori symétrique non négative. La matrice $\tilde{A}$ etant stable, V est donnée par

$$V = \int_0^\infty e^{\tilde{A}^T t} B R^{-1} B^T e^{\tilde{A} t} . dt$$

Or V doit être régulière, donc définie positive. Ceci n'est vrai que si la paire $(\tilde{A}, B)$ est commandable, c'est à dire la paire (A, B) commandable

En résumé on aura l'algorithme de calcul suivant

$$\text{Calculer} \quad S = \text{signe}(H)$$

$$\text{Poser} \quad F = \frac{1}{2}(I + S) = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix} \tag{53}$$

$$\text{Calculer} \quad P = -F_{12}^{-1}.F_1$$

Il ne nous reste plus maintenant qu'à évaluer le coût calcul de cet algorithme et le comparer à l'approche d'Anderson, qui jusqu'ici s'avère être la plus efficace.

4-COUT CALCUL

Le coût calcul de cet algorithme s'évalue comme suit:

-n^3 multiplications pour le calcul de l'inverse d'une matrice (n'étant sa dimension).

-K: nombre d'iterations nécessaires pour calculer S= signe de (H)

Donc le coût calcul est de l'ordre de $K(2n)^3 = 8Kn^3$. D'autre part le calcul de P nécessite approximativement $2n^3$ multiplications.

D'où le coût total de l'algorithme:

$$Cm = (8K + 2)n^3 \tag{54}$$

Compte tenu de la nature du problème de la commande que l'on traite, les pôles du systéme bouclé (valeurs propres de H) sont situés dans le plan complexe au voisinage de l'axe des réels; on peut donc admettre que

$$|tg\Theta| < t \qquad t \simeq 1$$

Θ argument des pôles complexes.

Ce qui permettera un bon conditionnement de la matrice H vis à vis du calcul de sa matrice signe.

En résumé on aura l'algorithme de calcul suivant

$$\text{Calculer} \quad S = \text{signe}(H)$$

$$\text{Poser} \quad F = \frac{1}{2}(I + S) = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix} \tag{53}$$

$$\text{Calculer} \quad P = -F_{12}^{-1}.F_1$$

Il ne nous reste plus maintenant qu'à évaluer le coût calcul de cet algorithme et le comparer à l'approche d'Anderson, qui jusqu'ici s'avère être la plus efficace.

4-COUT CALCUL

Le coût calcul de cet algorithme s'évalue comme suit:

-n^3 multiplications pour le calcul de l'inverse d'une matrice (n'étant sa dimension).

-K: nombre d'iterations nécessaires pour calculer S= signe de (H)

Donc le coût calcul est de l'ordre de $K(2n)^3 = 8Kn^3$.D'autre part le calcul de P nécessite approximativement $2n^3$ multiplications

D'où le coût total de l'algorithme:

$$Cm = (8K + 2)n^3 \tag{54}$$

Compte tenu de la nature du problème de la commande que l'on traite,les pôles du systéme bouclé(valeurs propres de H)sont situés dans le plan complexe au voisinage de l'axe des réels;on peut donc admettre que

$$|tg\Theta| < t \qquad t \simeq 1$$

Θ argument des pôles complexes.

Ce qui permettera un bon conditionnement de la matrice H vis à vis du calcul de sa matrice signe.

En fait dans la plupart des situations, le nombre d'itérations K varie de 5 à 20 en fonction du spectre du système bouclé indépendamment de la dimension n du problème et du spectre du système en boucle ouverte.

5-COMPARAISON AVEC L'APPROCHE D'ANDERSON:

Le principe de l'algorithme d'ANDERSON consiste à poser:

$$H_K = \begin{bmatrix} A_K & B_K \\ C_K & -A_K^T \end{bmatrix} \tag{55}$$

d'où

$$A_{K+1} = \frac{1}{2}(A_K + A_K^{-1})$$

$$B_{K+1} = \frac{1}{2}(B_K + B_K^{-1})$$

$$C_{K+1} = \frac{1}{2}(C_K + C_K^{-1})$$

Calculons A_K^{-1}, B_K^{-1} et C_K^{-1}

Soit

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

donc

$$E = (A-BD^{-1}C)^{-1}$$

$$G = -D^{-1}C(A-BD^{-1}C)^{-1}$$

$$F = -A^{-1}B(D-CA^{-1}B)^{-1}$$

$$H = (D-CA^{-1}B)^{-1}$$

d'où

$$A_{K+1} = \frac{1}{2}\left[A_K + \left[A_K + B_K(A_K^T)^{-1}C_K\right]^{-1}\right]$$

$$B_{K+1} = \frac{1}{2}\left[B_K + \left[A_K + B_K(A_K^T)^{-1}C_K\right]^{-1}B_K(A_K^T)^{-1}\right]$$

En fait dans la plupart des situations, le nombre d'itérations K varie de 5 à 20 en fonction du spectre du système bouclé indépendamment de la dimension n du problème et du spectre du système en boucle ouverte.

5-COMPARAISON AVEC L'APPROCHE D'ANDERSON:

Le principe de l'algorithme d'ANDERSON consiste à poser:

$$H_K = \begin{bmatrix} A_K & B_K \\ C_K & -A_K^T \end{bmatrix} \tag{55}$$

d'où

$$A_{K+1} = \frac{1}{2}(A_K + A_K^{-1})$$

$$B_{K+1} = \frac{1}{2}(B_K + B_K^{-1})$$

$$C_{K+1} = \frac{1}{2}(C_K + C_K^{-1})$$

Calculons A_K^{-1}, B_K^{-1} et C_K^{-1}

Soit

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

donc

$$E = (A-BD^{-1}C)^{-1}$$

$$G = -D^{-1}C(A-BD^{-1}C)^{-1}$$

$$F = -A^{-1}B(D-CA^{-1}B)^{-1}$$

$$H = (D-CA^{-1}B)^{-1}$$

d'où

$$A_{K+1} = \frac{1}{2}\left[A_K + \left[A_K + B_K(A_K^T)^{-1}C_K\right]^{-1}\right]$$

$$B_{K+1} = \frac{1}{2}\left[B_K + \left[A_K + B_K(A_K^T)^{-1}C_K\right]^{-1}B_K(A_K^T)^{-1}\right]$$

$$C_{K+1} = \frac{1}{2}\left[C_K + (A_K^T)^{-1} C_K \left[A_K + B_K (A_K^T)^{-1} C_K\right]^{-1}\right]$$

Comparativement à cette approche d'ANDERSON [4 ; 5], notre algorithme ne tient pas compte de la structure de l'hamiltonien pour calculer sa matrice signe.

L'avantage de cette approche (d'ANDERSON) pourait être le gain en place mémoire et en coût calcul.

Cependant cette économie de mémoires est inférieure à $2n^2$ ($6n^2+n$ au lieu de $8n^2$) et celle en multiplications de n^3 ($7n^3$ au lieu de $8n^3$).

Mais vu que l'implimentation et beaucoup moins simple, cette économie est relativement faible. De plus cette solution a recourt à 2 inversions de matrices $(A_K^T)^{-1}$ et $\left[A_K + B_K (A_K^T)^{-1} C_K\right]^{-1}$ dont il faut connaitre le condi-tionnement au cours des itératons, en particulier la matrice du système en boucle ouverte.

$$C_{K+1} = \frac{1}{2}\left[C_K + (A_K^T)^{-1} C_K \left[A_K + B_K (A_K^T)^{-1} C_K\right]^{-1}\right]$$

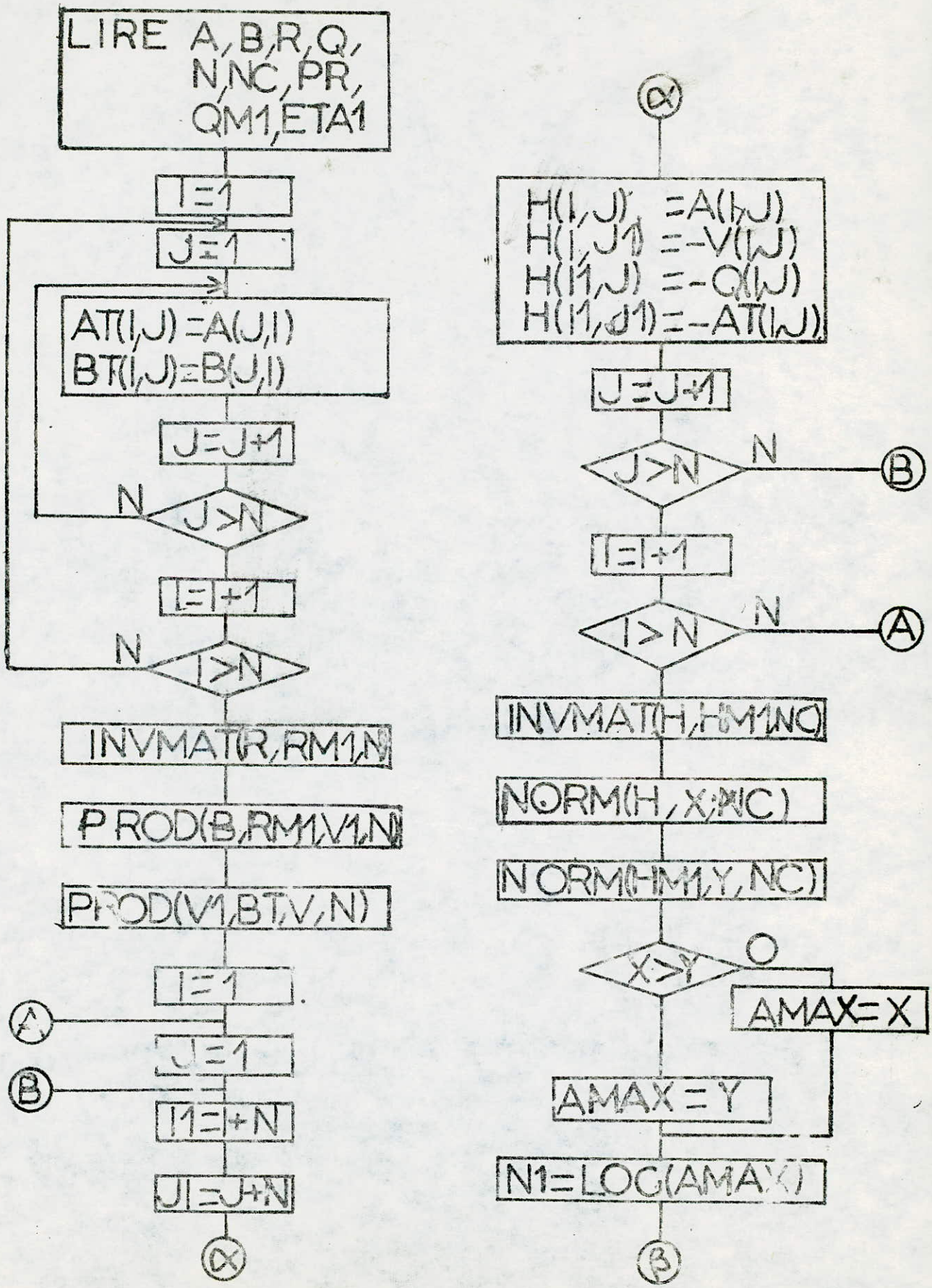
Comparativement à cette approche d'ANDERSON [4 ; 5], notre algorithme ne tient pas compte de la structure de l'hamiltonien pour calculer sa matrice signe.

L'avantage de cette approche (d'ANDERSON) pourait être le gain en place mémoire et en coût calcul.

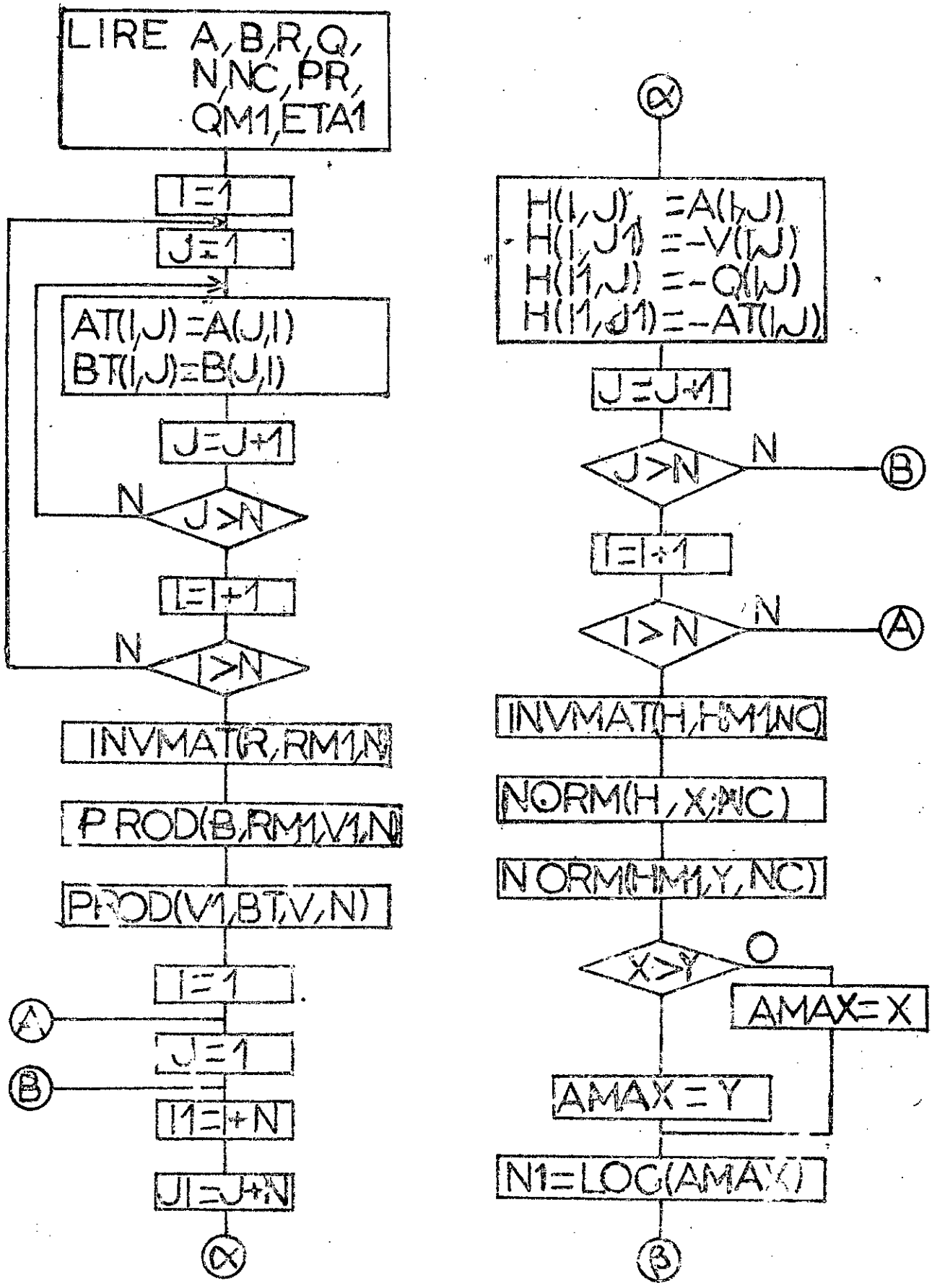
Cependant cette économie de mémoires est inférieure à $2n^2$ ($6n^2+n$ au lieu de $8n^2$) et celle en multiplications de n^3 ($7n^3$ au lieu de $8n^3$).

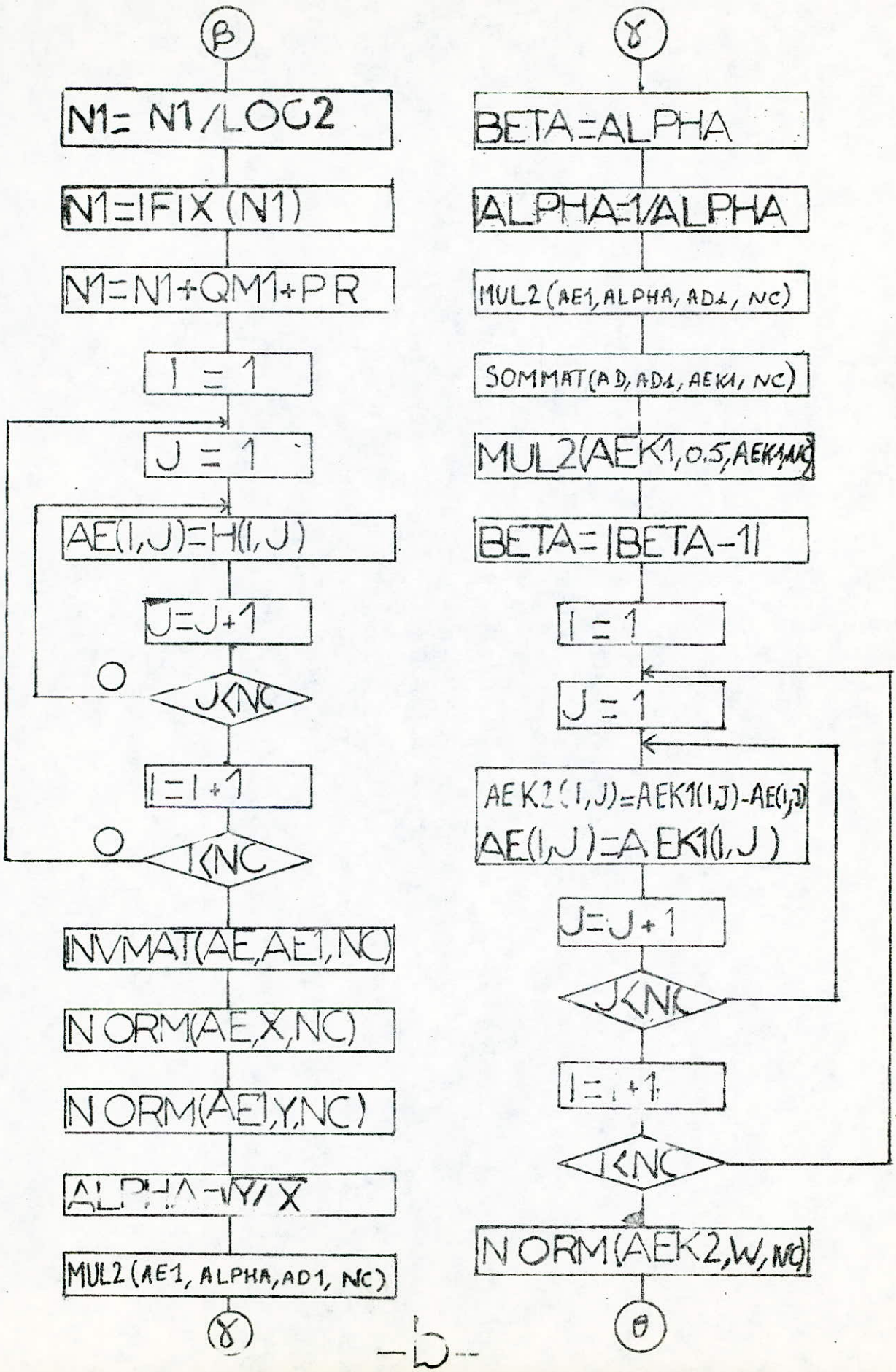
Mais vu que l'implimentation et beaucoup moins simple, cette économie est relativement faible. De plus cette solution a recourt à 2 inversions de matrices $(A_K^T)^{-1}$ et $\left[A_K + B_K (A_K^T)^{-1} C_K\right]^{-1}$ dont il faut connaitre le condi-tionnement au cours des itératons, en particulier la matrice du système en boucle ouverte.

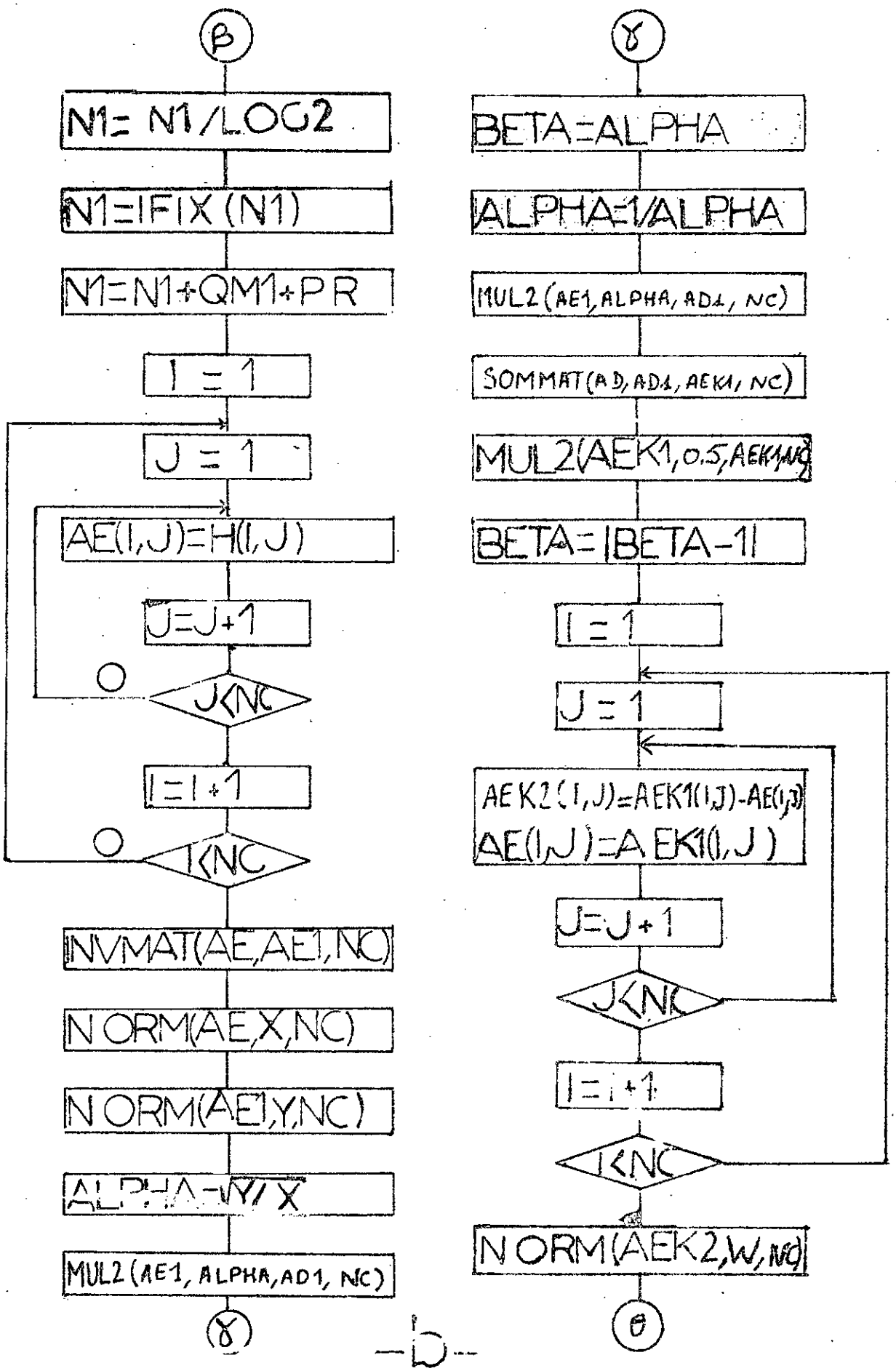
ORGANIGRAMME PRINCIPAL

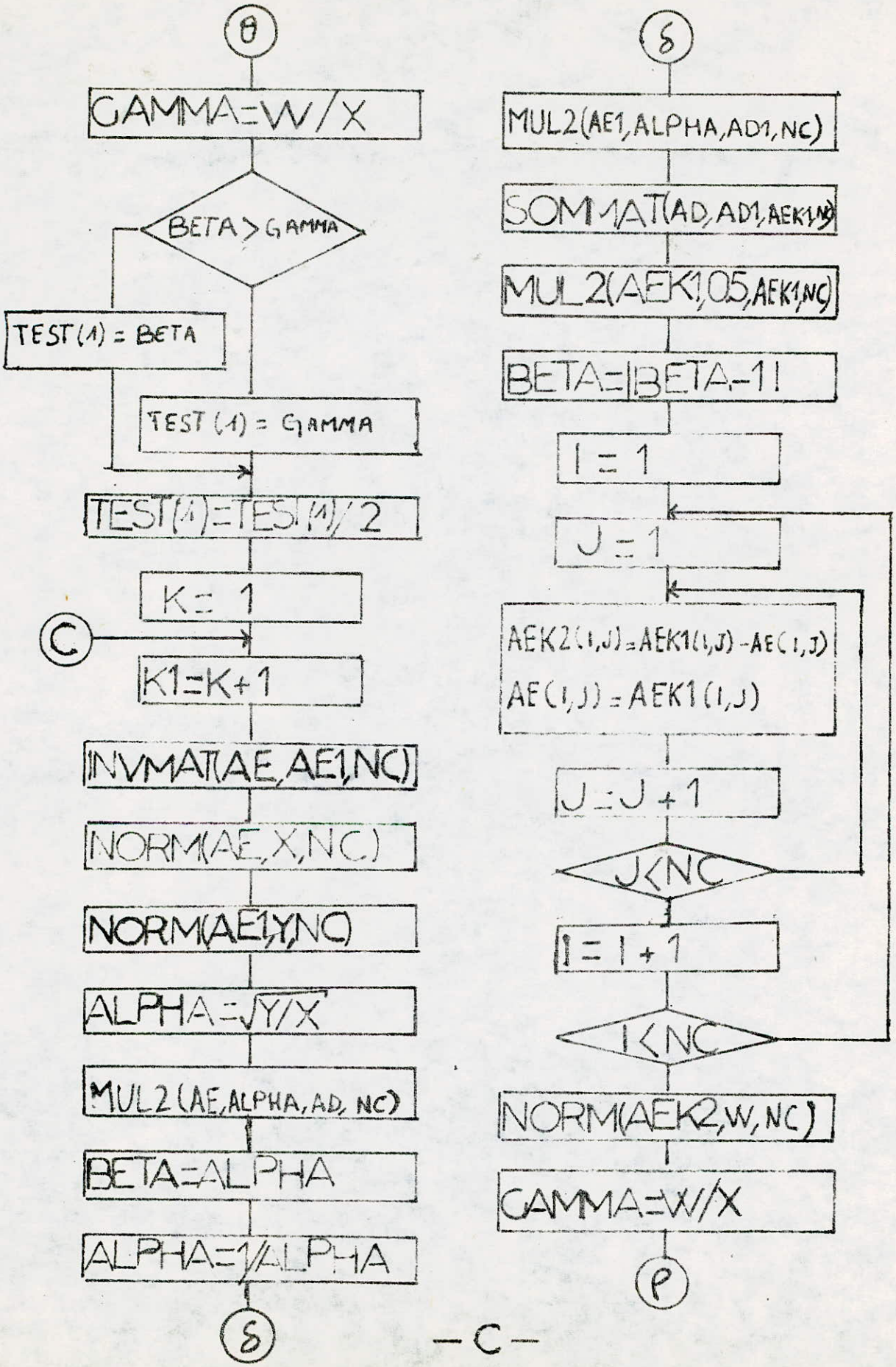


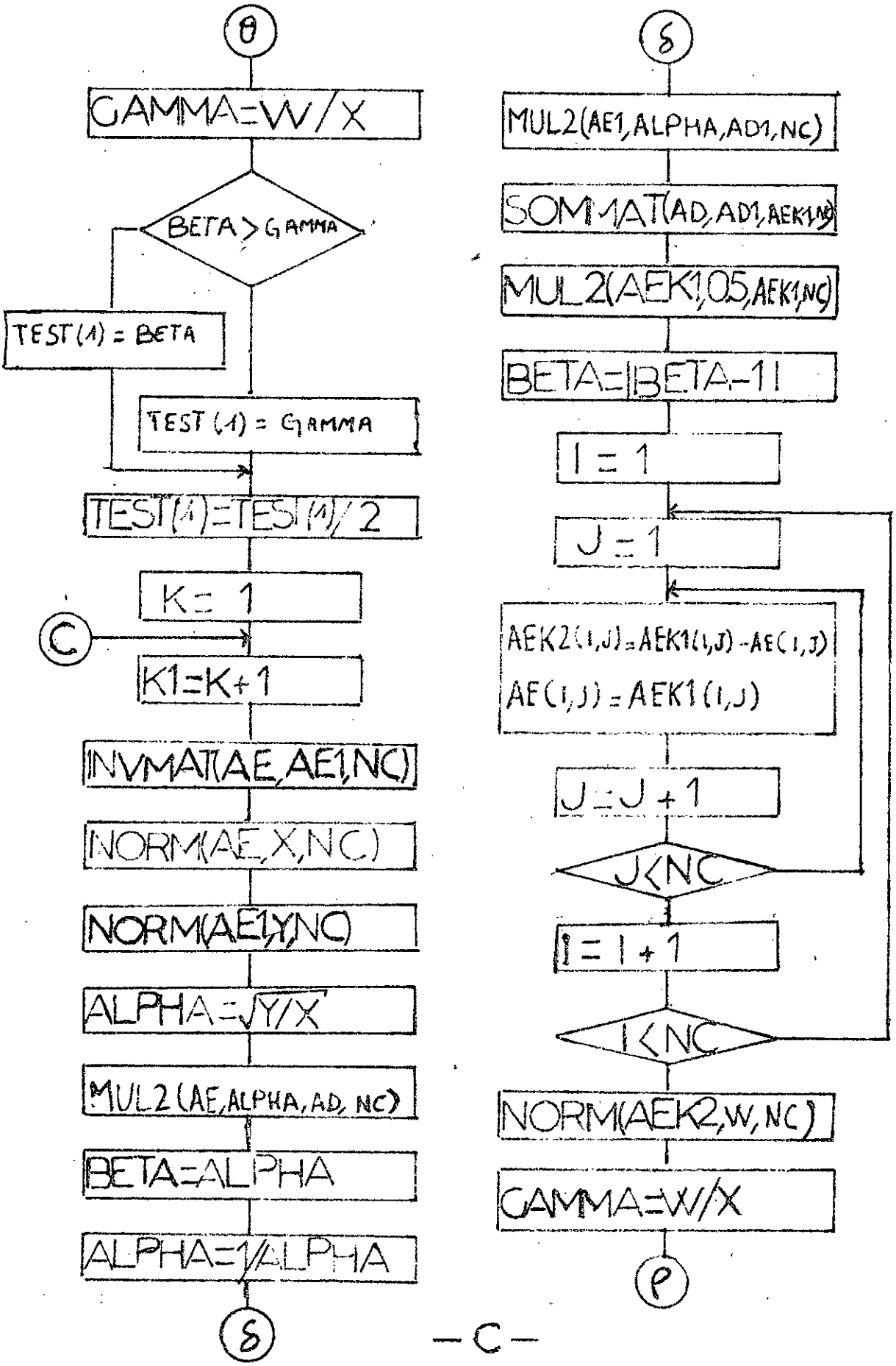
ORGANIGRAMME PRINCIPAL

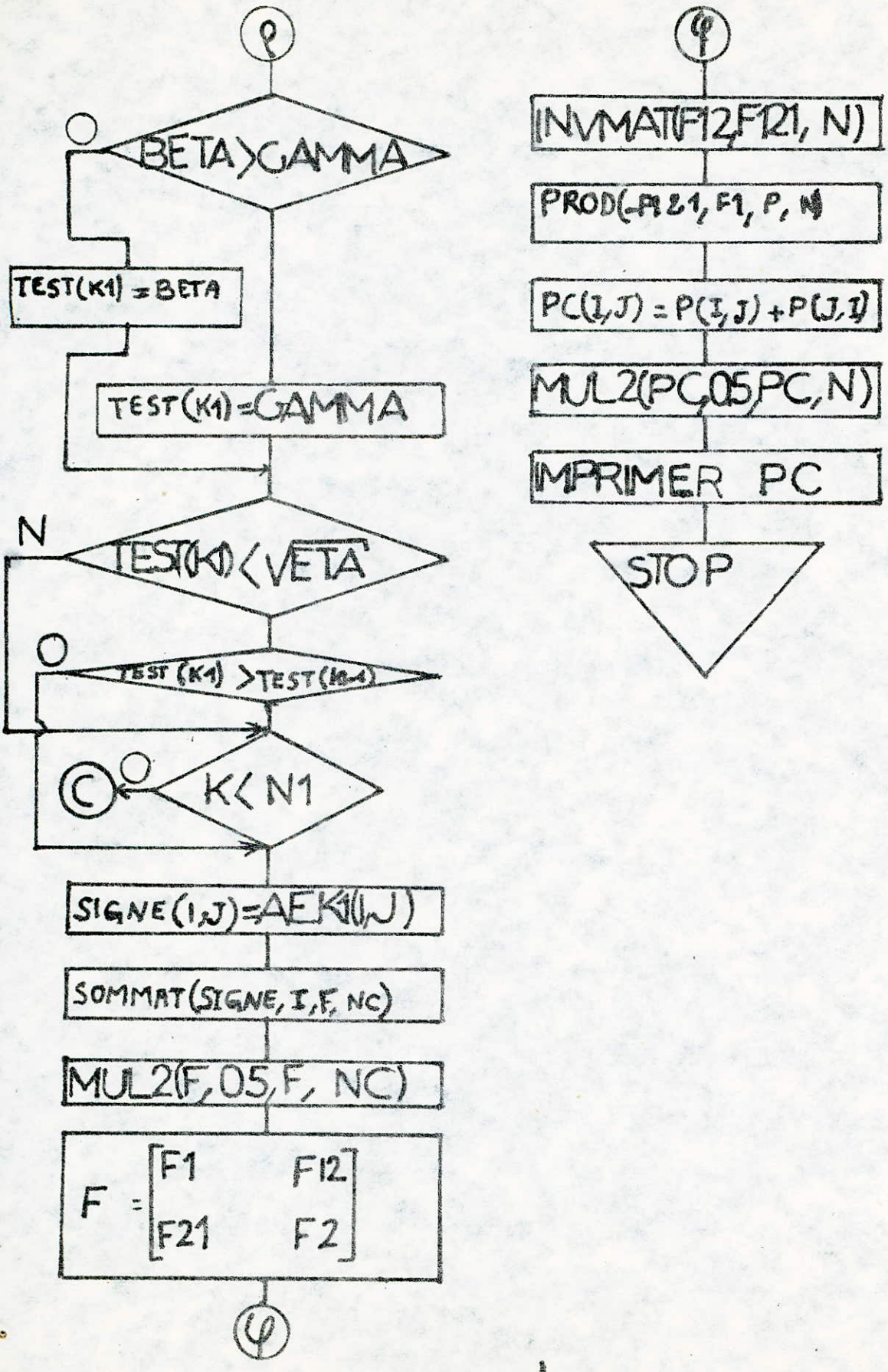


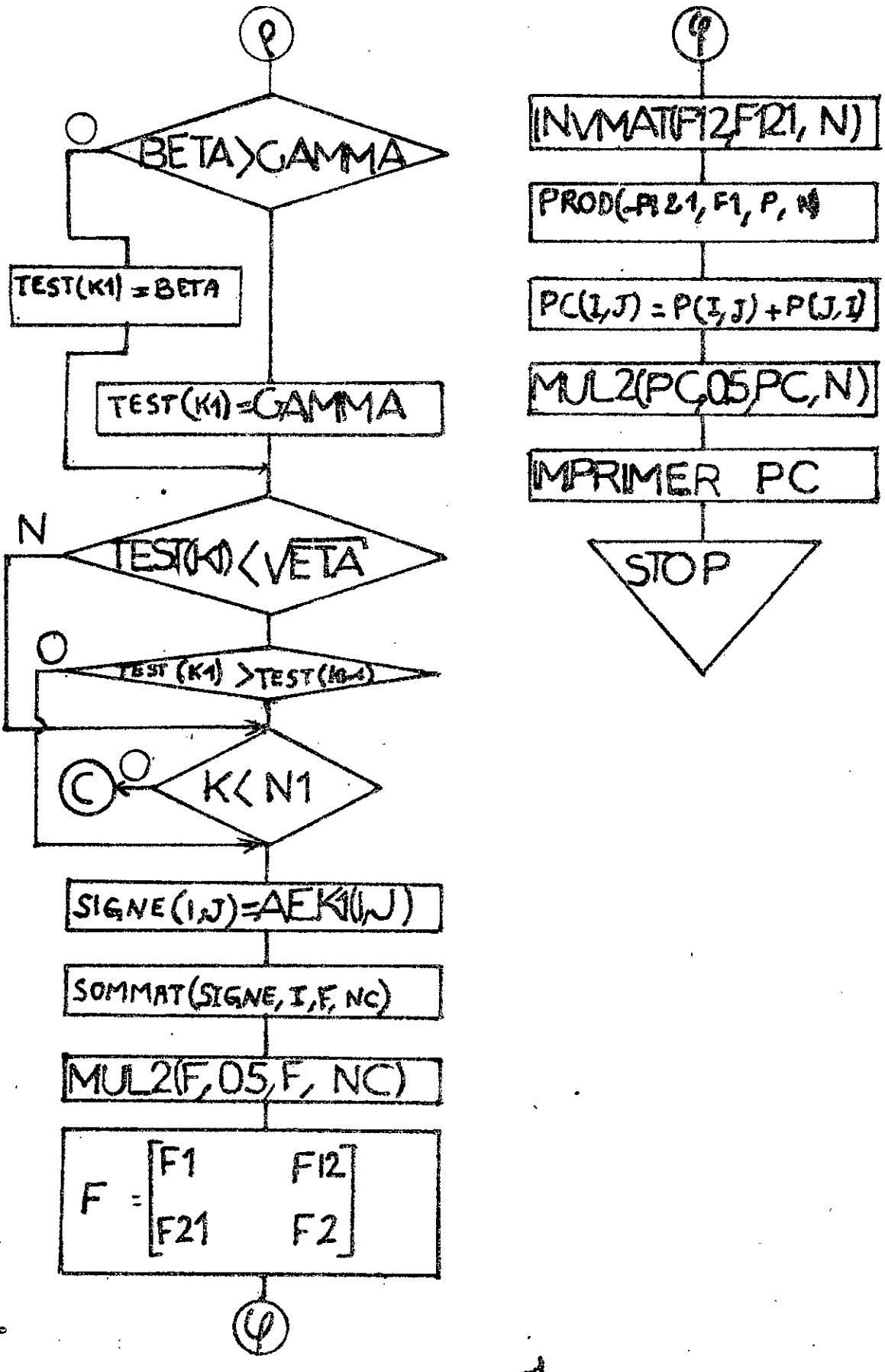

β
N1= N1/LOG2
N1=IFIX (N1)
N1=N1+QM1+PR
I = 1
J = 1
AE(I,J)=H(I,J)
J=J+1
O
J<NC
I=I+1
O
I<NC
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=√Y/X
MUL2 (AE1, ALPHA, AD1, NC)
δ
γ
BETA=ALPHA
ALPHA=1/ALPHA
MUL2 (AE1, ALPHA, AD1, NC)
SOMMAT(AD, AD1, AEK1, NC)
MUL2(AEK1, 0.5, AEK1, NC)
BETA=|BETA-1|
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J)=AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
θ


β
N1= N1/LOG2
N1=IFIX (N1)
N1=N1+QM1+PR
I = 1
J = 1
AE(I,J)=H(I,J)
J=J+1
J<NC
I=I+1
I<NC
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=Y/X
MUL2 (AE1, ALPHA, AD1, NC)
δ
γ
BETA=ALPHA
ALPHA=1/ALPHA
MUL2 (AE1, ALPHA, AD1, NC)
SOMMAT(AD, AD1, AEK1, NC)
MUL2(AEK1, 0.5, AEK1,NC)
BETA=|BETA-1|
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J)=AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
θ

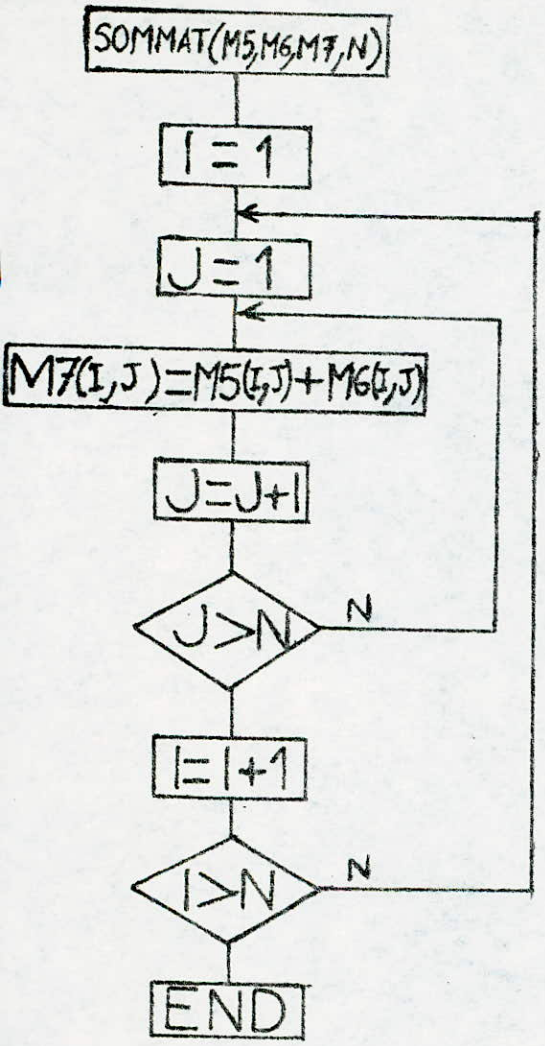

θ
GAMMA=W/X
BETA > GAMMA
TEST(1) = BETA
TEST(1) = GAMMA
TEST(1)=TEST(1)/2
K = 1
C
K1=K+1
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=√Y/X
MUL2(AE,ALPHA,AD,NC)
BETA=ALPHA
ALPHA=1/ALPHA
δ
δ
MUL2(AE1,ALPHA,AD1,NC)
SOMMAT(AD,AD1,AEK1,NC)
MUL2(AEK1,0.5,AEK1,NC)
BETA=|BETA-1|
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J)=AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
GAMMA=W/X
ρ

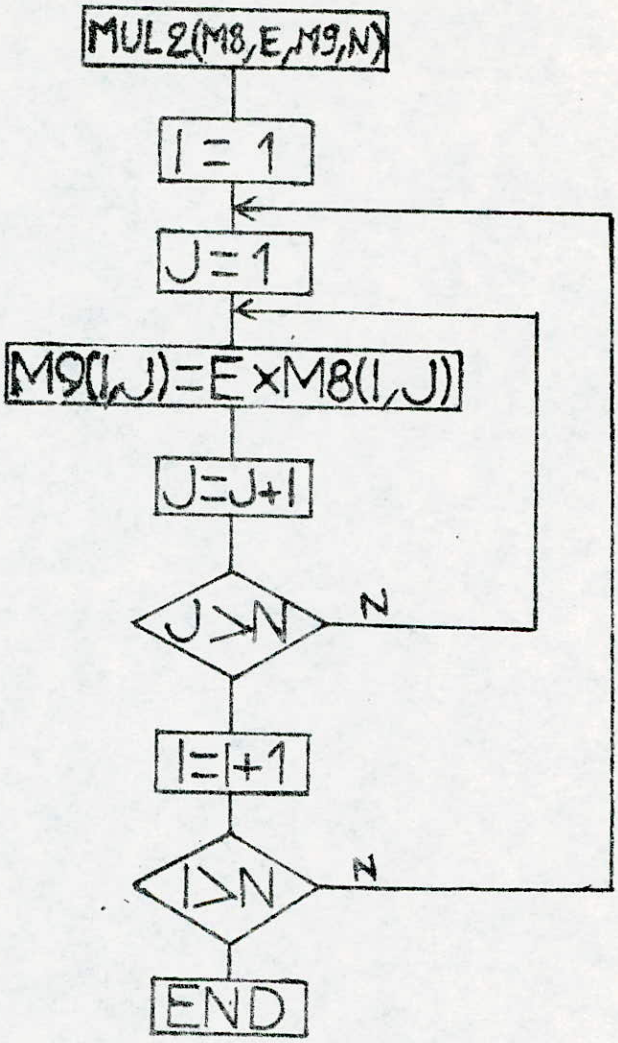

θ
GAMMA=W/X
BETA > GAMMA
TEST(1) = BETA
TEST(1) = GAMMA
TEST(1)=TEST(1)/2
K = 1
C
K1=K+1
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=√Y/X
MUL2(AE,ALPHA,AD,NC)
BETA=ALPHA
ALPHA=1/ALPHA
δ
δ
MUL2(AE1,ALPHA,AD1,NC)
SOMMAT(AD,AD1,AEK1,NC)
MUL2(AEK1,0.5,AEK1,NC)
BETA=|BETA-1|
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J) = AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
GAMMA=W/X
ρ


BETA>GAMMA
TEST(K1) = BETA
TEST(K1) = GAMMA
N
TEST(K1) < VETA
TEST (K1) > TEST (K-1)
K < N1
SIGNE(I,J) = AEK1(I,J)
SOMMAT(SIGNE, I, F, NC)
MUL2(F, 05, F, NC)
F = [F1 F12; F21 F2]
INVMAT(F12, F121, N)
PROD(-F1&1, F1, P, N
PC(I,J) = P(I,J) + P(J,I)
MUL2(PC, 05, PC, N)
IMPRIMER PC
STOP

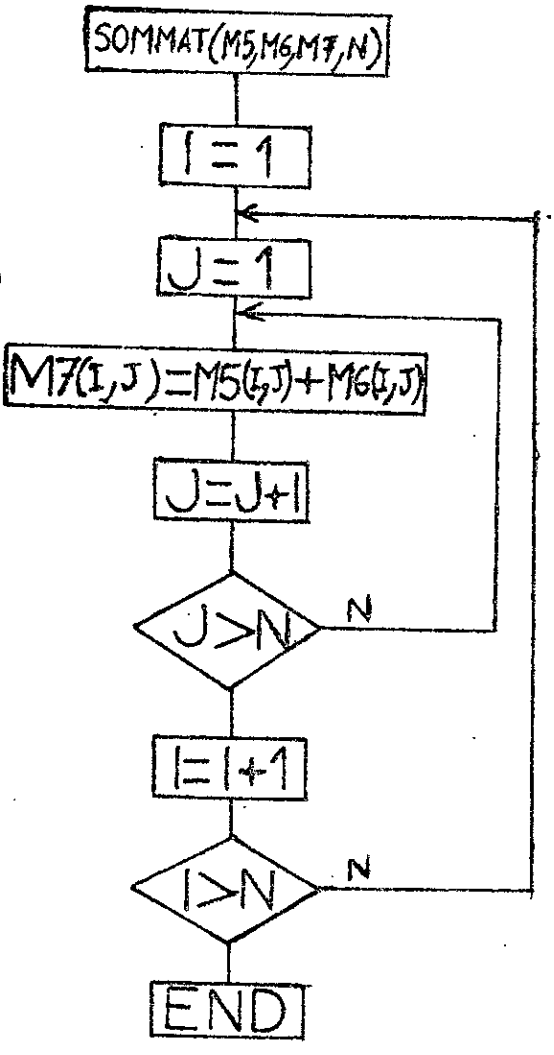

BETA>GAMMA
TEST(K1) = BETA
TEST(K1) = GAMMA
N
TEST(K1) < VETA
TEST (K1) > TEST (K-1)
K < N1
SIGNE (I,J) = AEK1(I,J)
SOMMAT(SIGNE, I, F, NC)
MUL2(F, 05, F, NC)
F = [F1 F12; F21 F2]
INVMAT(F12, F121, N)
PROD(-F12 21, F1, P, N
PC(I,J) = P(I,J) + P(J,I)
MUL2(PC, 05, PC, N)
IMPRIMER PC
STOP

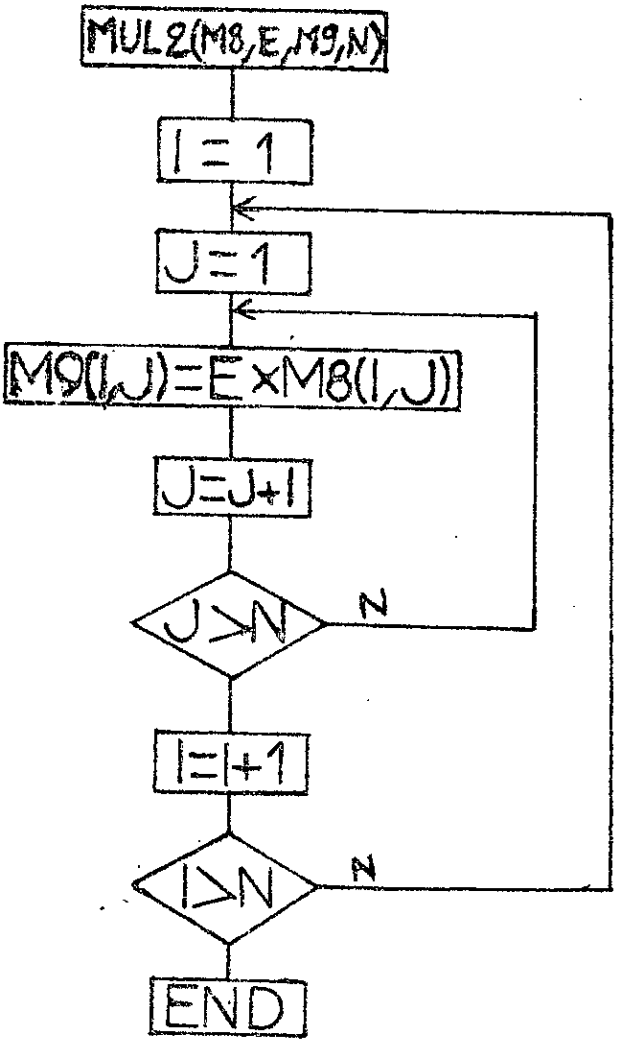
SOUS PROGRAMMES

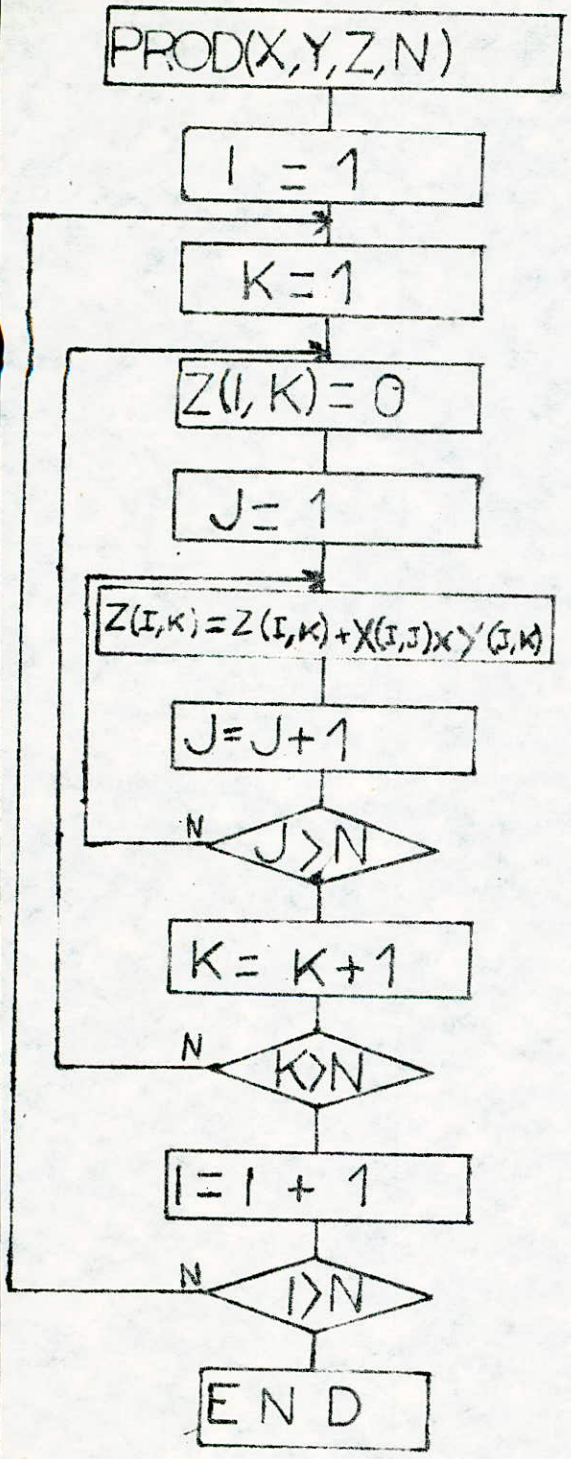


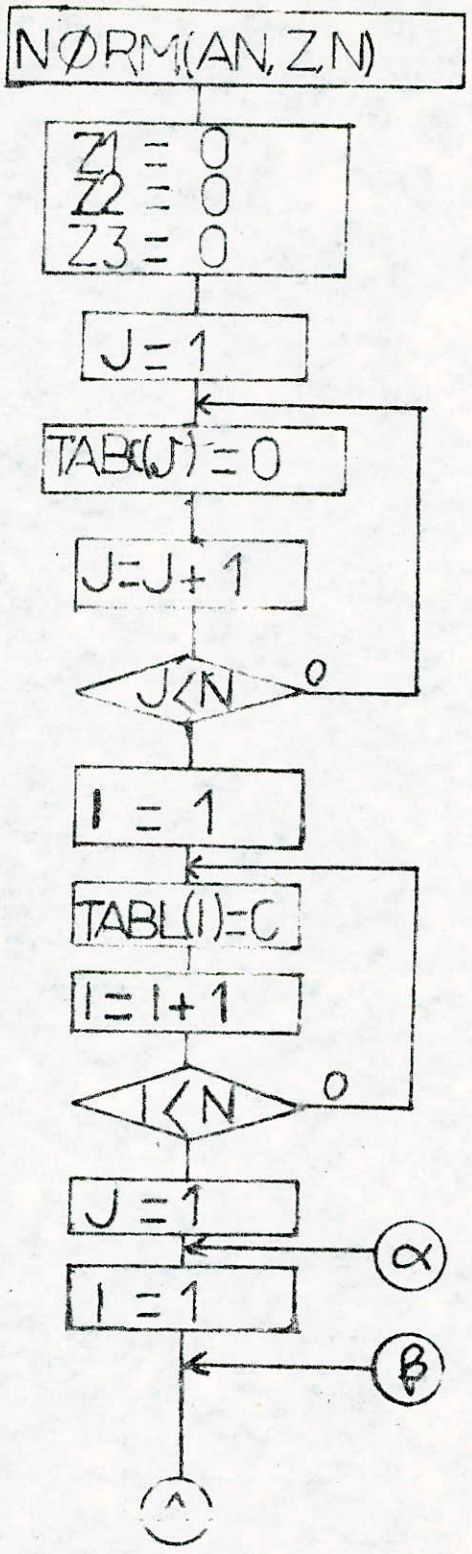


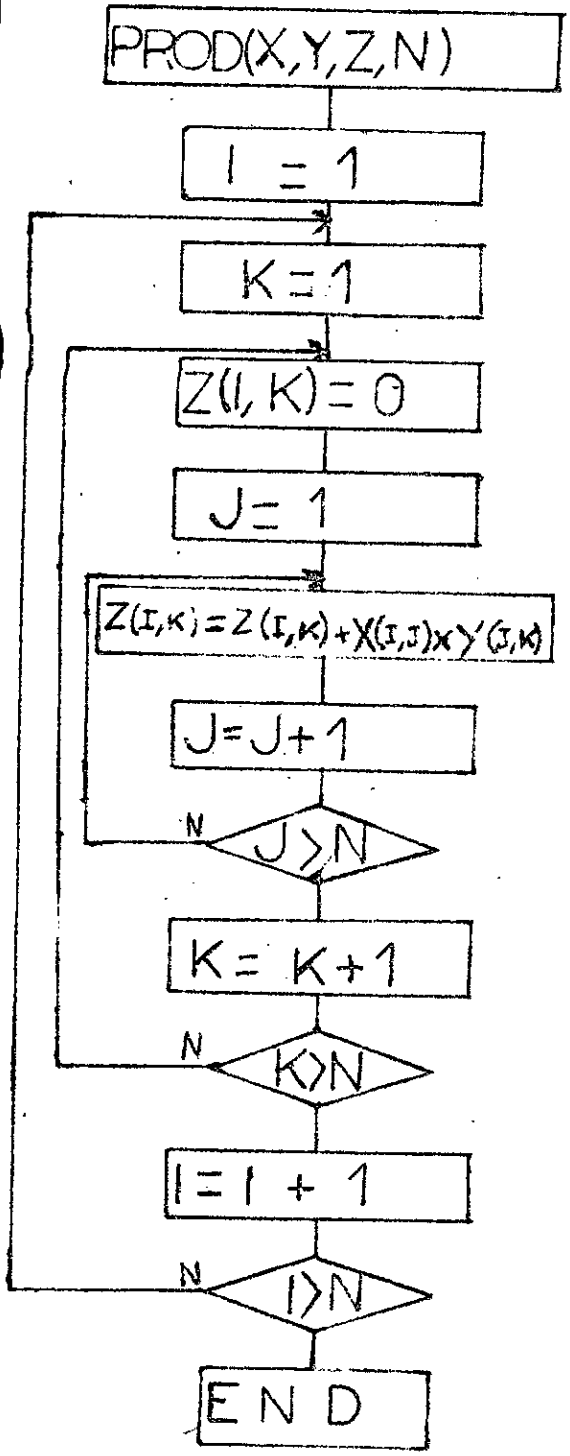
SOUS PROGRAMMES

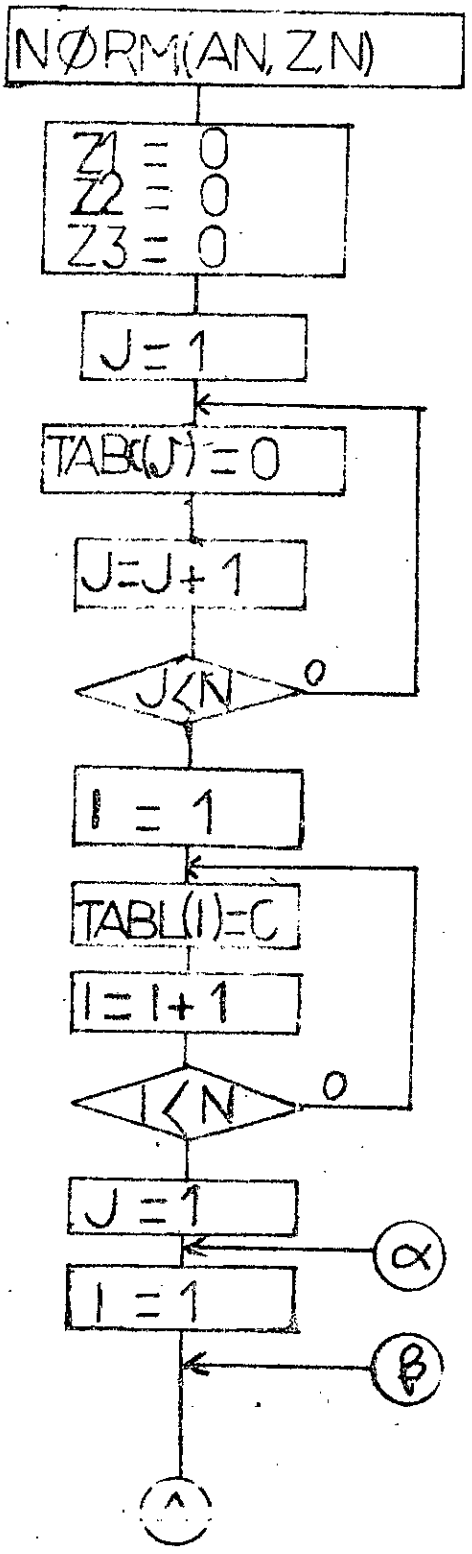


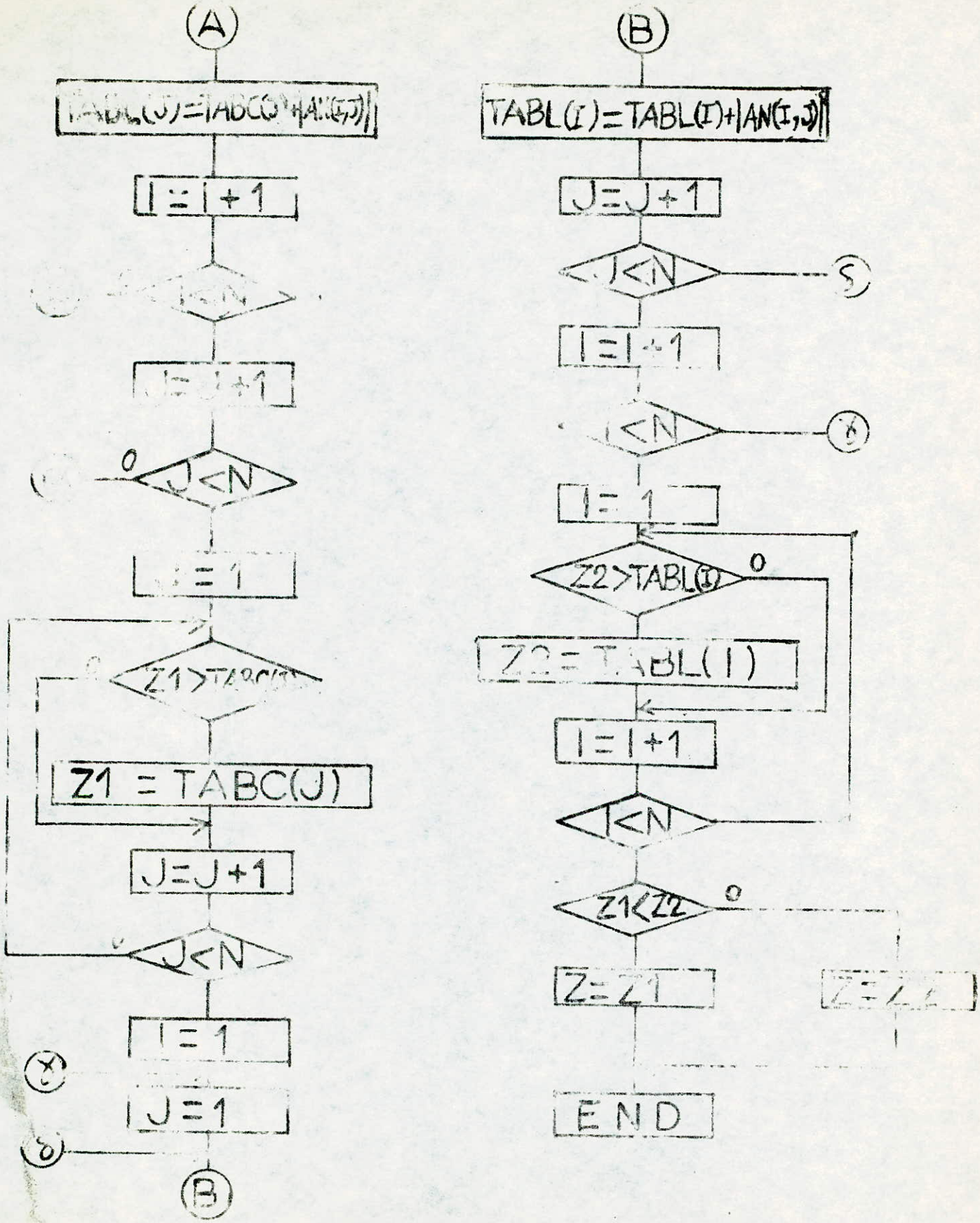


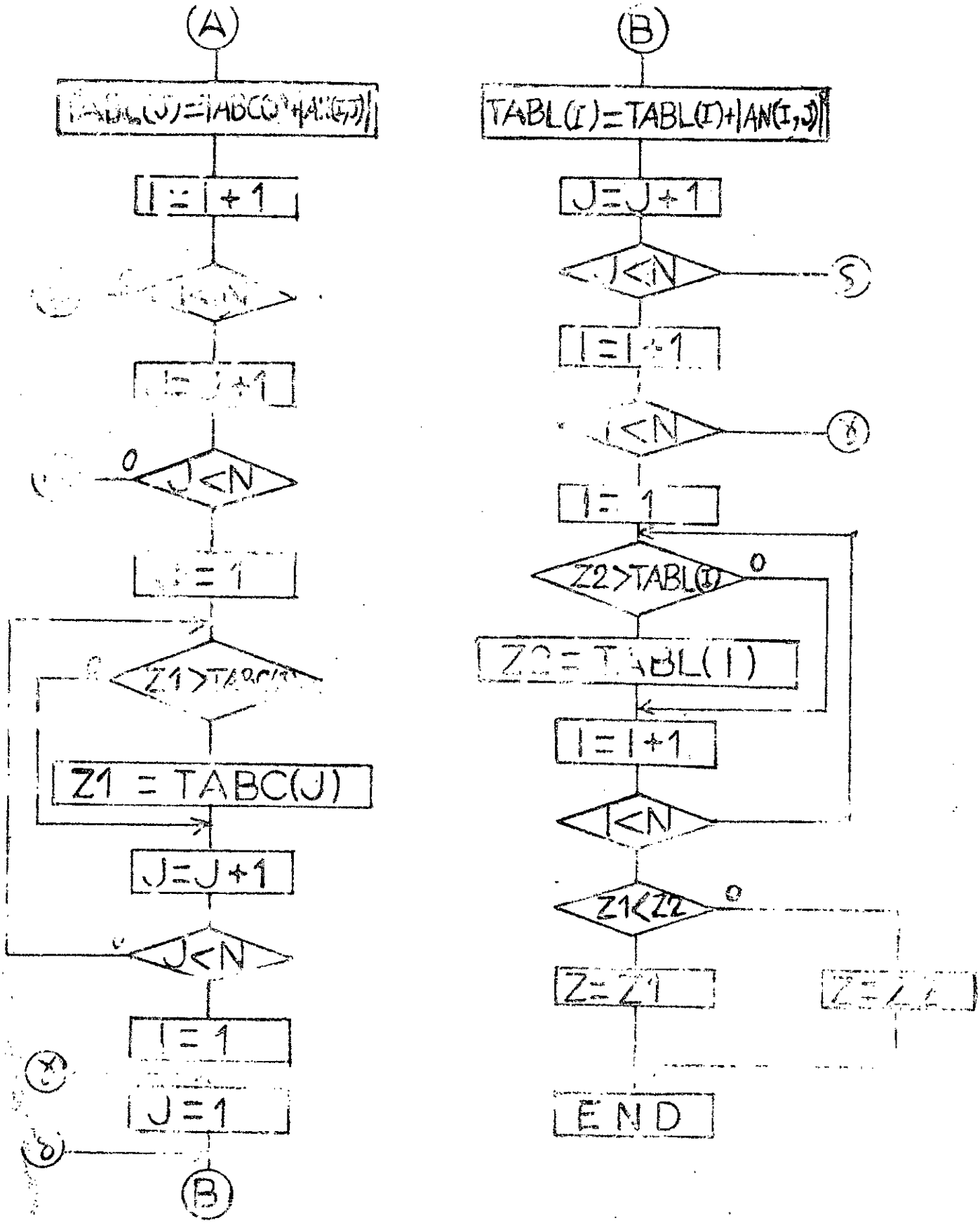

PROD(X,Y,Z,N)
I = 1
K = 1
Z(I,K) = 0
J = 1
Z(I,K) = Z(I,K) + X(I,J) × Y(J,K)
J = J + 1
N
J > N
K = K + 1
N
K > N
I = I + 1
N
I > N
END


NØRM(AN,ZN)
Z1 = 0
Z2 = 0
Z3 = 0
J = 1
TAB(J) = 0
J = J + 1
J < N
O
I = 1
TABL(I) = C
I = I + 1
I < N
O
J = 1
α
I = 1
β
^


PROD(X,Y,Z,N)
I = 1
K = 1
Z(I, K) = 0
J = 1
Z(I,K) = Z(I,K) + X(I,J) x Y(J,K)
J = J + 1
N
J > N
K = K + 1
N
K > N
I = I + 1
N
I > N
END


NØRM(AN, ZN)
Z1 = 0
Z2 = 0
Z3 = 0
J = 1
TAB(J) = 0
J = J + 1
J < N
O
I = 1
TABL(I) = C
I = I + 1
I < N
O
J = 1
α
I = 1
β
^


(A)
TABL(J)=TABL(J)+AN(I,J)
I=I+1
I<N
J=J+1
J<N
I=1
Z1>TABC(J)
Z1 = TABC(J)
J=J+1
J<N
I=1
J=1
(B)
(B)
TABL(I)=TABL(I)+AN(I,J)
J=J+1
J<N
(5)
I=I+1
I<N
(6)
I=1
Z2>TABL(I)
Z2=TABL(I)
I=I+1
I<N
Z1<Z2
Z=Z1
END


A
TABC(J)=TABC(J)+AN(I,J)
I=I+1
I<N
J=J+1
J<N
I=1
Z1>TABC(J)
Z1 = TABC(J)
J=J+1
J<N
I=1
J=1
B
B
TABL(I)=TABL(I)+AN(I,J)
J=J+1
J<N
I=I+1
I<N
I=1
Z2>TABL(I)
Z2=TABL(I)
I=I+1
I<N
Z1<Z2
Z=Z1
END

RESOLUTION DE L'EQUATION DE RICCATI

DANS LE CAS DISCRET

I-INTRODUCTION

On considère ici la résolution numérique de l'équation algébrique de Riccati,

$$A^TPA - P - A^TPB(B^TPB + R)B^TPA + Q = 0$$

introduite en termes de commande des systèmes discrets; on developpe, pour cela, le concept de produit étoile et les propriétés qui s'y rattachent.

L'application de ces résultats conduit à une méthode déduite de la solution apportée au problème continu ainsi qu'à un algorithme itératif du type "Square root"($P = L.L^T$) présentant une convergence d'ordre 2.

La première méthode qui est en fait une nouvelle voie constituant un traitement unifié des équations de Riccati à l'aide de la fonction signe de matrice, est alors comparée à cet algorithme ainsi qu'à d'autres méthodes, les unes considérées déjà comme classiques, les autres un peu plus récentes.

2-DEFINITION DU PROBLEME

On se propose de résoudre l'équation algébrique de Riccati

$$P = A^TPA - A^TPB(B^TPB + R)^{-1}B^TPA + Q \qquad (56)$$

introduite ici dans le cadre de la commande optimale des systèmes discrets linéaires invariants.

Soit donc le système

$$X_{t+1} = A.X_t + B.U_t \qquad (57)$$

et le critère quadratique à minimiser $J = \frac{1}{2}\sum_{t=0}^{\infty}(U_t^TRU_t + X_t^TQX_t)$ (58)

moyennant les hypothèses suivantes:

(A,B) stabilisable (59a)

(C,A) détéctable $C^TC = Q$, C (59b)

RESOLUTION DE L'EQUATION DE RICCATI

DANS LE CAS DISCRET

I-INTRODUCTION

On considère ici la résolution numérique de l'équation algébrique de Riccati,

$$A^TPA - P - A^TPB(B^TPB + R)B^TPA + Q = 0$$

introduite en termes de commande des systèmes discrets; on developpe, pour cela, le concept de produit étoile et les propriétés qui s'y rattachent.

L'application de ces résultats conduit à une méthode déduite de la solution apportée au problème continu ainsi qu'à un algorithme itératif du type "Square root"($P = L.L^T$) présentant une convergence d'ordre 2.

La première méthode qui est en fait une nouvelle voie constituant un traitement unifié des équations de Riccati à l'aide de la fonction signe de matrice, est alors comparée à cet algorithme ainsi qu'à d'autres méthodes, les unes considérées déjà comme classiques, les autres un peu plus récentes.

2-DEFINITION DU PROBLEME

On se propose de résoudre l'équation algébrique de Riccati

$$P = A^TPA - A^TPB(B^TPB + R)^{-1}B^TPA + Q \qquad (56)$$

introduite ici dans le cadre de la commande optimale des systèmes discrets linéaires invariants.

Soit donc le système

$$X_{t+1} = A.X_t + B.U_t \qquad (57)$$

et le critère quadratique à minimiser $J = \frac{1}{2}\sum_{t=0}^{\infty}(U_t^TRU_t + X_t^TQX_t)$ (58)

moyennant les hypothèses suivantes:

(A,B) stabilisable (59a)

(C,A) détéctable $C^TC = Q$ (59b)

$$B^T Q B + R = (B^T Q B + R)^T \tag{59c}$$

On sait [I] que

$$P = P^T \geq 0 \qquad P \text{ unique} \tag{60}$$

$$|\lambda_i(\tilde{A})| < 1 \qquad \tilde{A} = A - B (B^T P B + R)^{-1} B^T P A \tag{61}$$

Cette dernière conclusion exprimant que le système bouclé par la commande

$$\hat{U}_t = - [(B^T P B + R)^{-1} B^T P A] X_t \tag{62}$$

est asymtotiquement stable

enfin le coût optimal J (Ū). est donné par :

$$J = \frac{I}{2} X_o^T . P . X_o \tag{63}$$

Considérons maintenant l'état adjoint λ_t que l'on peut introduire pour résoudre le problème de minimisation avec contraintes défini par (57) et (58).

D'une part on a

$$\lambda t = P t . X t \tag{64}$$

et d'autre part l'ensemble des équations hamiltoniennes discrètes s'écrivent :

$$\begin{bmatrix} X_{t+1} \\ \lambda t \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix} \cdot \begin{bmatrix} X_t \\ \lambda t + I \end{bmatrix} \tag{65}$$

Notons

$$M = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix} \tag{66}$$

$$B^T Q B + R = (B^T Q B + R)^T \tag{59c}$$

On sait [I] que

$$P = P^T \geq 0 \qquad P \text{ unique} \tag{60}$$

$$|\lambda_i(\tilde{A})| < 1 \qquad \tilde{A} = A - B (B^T P B + R)^{-1} B^T P A \tag{61}$$

Cette dernière conclusion exprimant que le système bouclé par la commande

$$\hat{U}_t = - [(B^T P B + R)^{-1} B^T P A] X_t \tag{62}$$

est asymtotiquement stable

enfin le coût optimal $J(\hat{U})$ est donné par :

$$J = \frac{I}{2} X_o^T . P . X_o \tag{63}$$

Considérons maintenant l'état adjoint λ_t que l'on peut introduire pour résoudre le problème de minimisation avec contraintes défini par (57) et (58).

D'une part on a

$$\lambda t = P t . X t \tag{64}$$

et d'autre part l'ensemble des équations hamiltoniennes discrètes s'écrivent :

$$\begin{bmatrix} X_{t+1} \\ \lambda t \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix} \cdot \begin{bmatrix} X_t \\ \lambda t + I \end{bmatrix} \tag{65}$$

Notons

$$M = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix} \tag{66}$$

La matrice associée à ces équations, précisons le, n'est pas elle même hamiltonienne, comme dans le cas continu, ni symplectique(annexe)bien que l'on puisse en déduire une forme sympléctique à l'aide d'une transformation notée $\Psi(.)$, que l'on introduira au paragraphe suivant.

3- PRODUIT SIMPLE ET PRODUIT " étoile " DE MATRICES

Tout un chacun connaît l'illustration classique du produit de matrices 2X2 à travèrs la combinaison en serie de quadripoles (Fig I) :

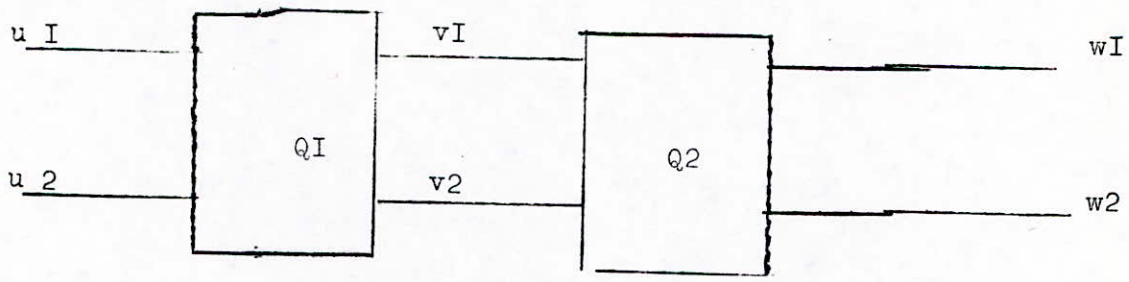


Figure I

que l'on formalise comme suit :

$$\begin{bmatrix} vI \\ v2 \end{bmatrix} = \begin{bmatrix} aI & bI \\ cI & dI \end{bmatrix} X \begin{bmatrix} uI \\ u2 \end{bmatrix} \Longrightarrow V = Q_1 u \qquad (67)$$

$$\begin{bmatrix} wI \\ w2 \end{bmatrix} = \begin{bmatrix} a2 & b2 \\ c2 & d2 \end{bmatrix} X \begin{bmatrix} vI \\ v2 \end{bmatrix} \Longrightarrow W = Q_2 v \qquad (68)$$

Le quadripôle équivalent Q est tel que

$$W = Q u \qquad (69)$$

$$Q = Q_2 Q_1$$

d'ou

$$Q = \begin{bmatrix} a2\,aI + b2\,cI & a2\,bI + b2\,dI \\ c2\,QI + d2\,cI & c2\,bI + d2\,dI \end{bmatrix} \qquad (70)$$

La matrice associée à ces équations, précisons le, n'est pas elle même hamiltonienne, comme dans le cas continu, ni symplectique(annexe)bien que l'on puisse en déduire une forme sympléctique à l'aide d'une transformation notée $\Psi(\ .\ )$, que l'on introduira au paragraphe suivant.

3- PRODUIT SIMPLE ET PRODUIT " étoile " DE MATRICES

Tout un chacun connaît l'illustration classique du produit de matrices 2X2 à travèrs la combinaison en serie de quadripoles (Fig I) :

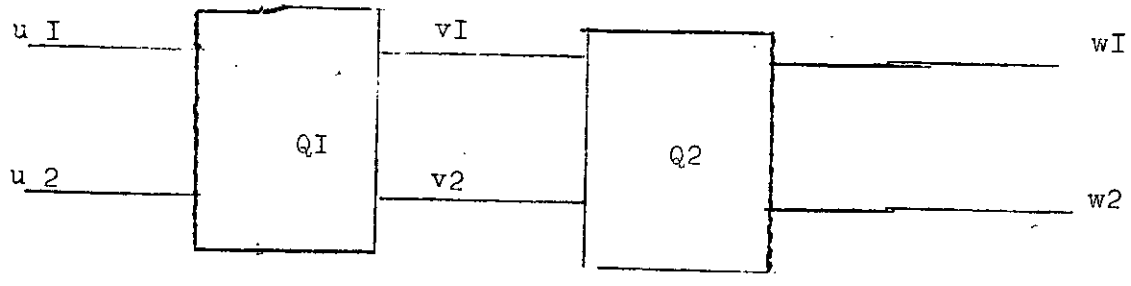


Figure I

que l'on formalise comme suit :

$$\begin{bmatrix} vI \\ v2 \end{bmatrix} = \begin{bmatrix} aI & bI \\ cI & dI \end{bmatrix} X \begin{bmatrix} uI \\ u2 \end{bmatrix} \Longrightarrow V = Q_1\, u \qquad (67)$$

$$\begin{bmatrix} wI \\ w2 \end{bmatrix} = \begin{bmatrix} a2 & b2 \\ c2 & d2 \end{bmatrix} X \begin{bmatrix} vI \\ v2 \end{bmatrix} \Longrightarrow W = Q_2\, v \qquad (68)$$

Le quadripôle équivalent Q est tel que

$$\begin{aligned} W &= Q\, u \\ Q &= Q_2\, Q_1 \end{aligned} \qquad (69)$$

d'ou

$$Q = \begin{bmatrix} a2\,aI + b2\,cI & a2\,bI + b2\,dI \\ c2\,QI + d2\,cI & c2\,bI + d2\,dI \end{bmatrix} \qquad (70)$$

Ce produit simple se généralise comme on le sait aux matrices de dimension quelconque.

IL est distributif par rapport à l'addition, associatif, possede un element nul et un element neutre.

Ainsi l'enssemble des matrices (n.n) muni de la loi d'addition et de ce produit forme un anneau et constitue une algèbre.

On va introduire maintenant un second type de produit matriciel basé lui aussi sur la composition en serie de quadripôles et montre qu'il possède les mêmes proprietés que le produit simple.

Considerons donc le problème de propagation d'ondes suivant:

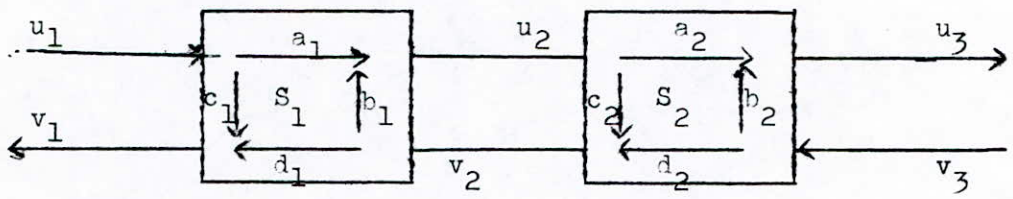


Figure 2

u_I,u_2 et u_3 sont des rayons incidents

v_I,v_2 et v_3 sont des rayons emergents

les milieux séparant les phases I,2 et 3 sont caractérisès par des Coefficients de reflexion de transmission désignés par a,b,c,et d et qui constituent les matrices S_I,et S_2(2.2).Les relations "entrées" "sorties" pour chacune des transitions s'écrivent:

$$\begin{bmatrix} u_2 \\ v_1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \begin{bmatrix} u_1 \\ v_2 \end{bmatrix} = S_1 \begin{bmatrix} u_1 \\ v_2 \end{bmatrix} \qquad (71)$$

$$\begin{bmatrix} u_3 \\ v_2 \end{bmatrix} = \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} \begin{bmatrix} u_2 \\ v_3 \end{bmatrix} = S_2 \begin{bmatrix} u_2 \\ v_3 \end{bmatrix} \qquad (72)$$

Les matrices S sont ce qu'on appelle des matrices d'eparpillement. La matrice globale est définie implicitement par:

$$\begin{bmatrix} u_3 \\ v_1 \end{bmatrix} = S \begin{bmatrix} u_I \\ v_3 \end{bmatrix} \qquad (73)$$

Ce produit simple se généralise comme on le sait aux matrices de dimension quelconque.

IL est distributif par rapport à l'addition, associatif, possede un element nul et un element neutre.

Ainsi l'enssemble des matrices (n.n) muni de la loi d'addition et de ce produit forme un anneau et constitue une algèbre.

On va introduire maintenant un second type de produit matriciel basé lui aussi sur la composition en serie de quadripôles et montre qu'il possède les mêmes propriétés que le produit simple.

Considerons donc le problème de propagation d'ondes suivant:

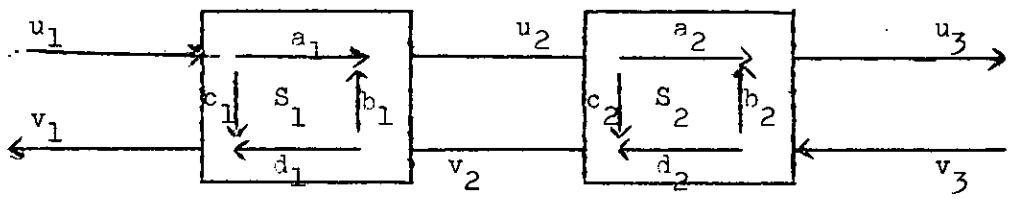


Figure 2

u_I,u_2 et u_3 sont des rayons incidents

v_I,v_2 et v_3 sont des rayons emergents

les milieux séparant les phases I,2 et 3 sont caractérisès par des Coefficients de reflexion de transmission désignés par a,b,c,et d et qui constituent les matrices S_I,et S_2(2.2).Les relations "entrées" "sorties" pour chacune des transitions s'écrivent:

$$\begin{bmatrix} u_2 \\ v_1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \begin{bmatrix} u_1 \\ v_2 \end{bmatrix} = S_1 \begin{bmatrix} u_1 \\ v_2 \end{bmatrix} \qquad (71)$$

$$\begin{bmatrix} u_3 \\ v_2 \end{bmatrix} = \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} \begin{bmatrix} u_2 \\ v_3 \end{bmatrix} = S_2 \begin{bmatrix} u_2 \\ v_3 \end{bmatrix} \qquad (72)$$

Les matrices S sont ce qu'on appelle des matrices d'eparpillement. La matrice globale est définie implicitement par:

$$\begin{bmatrix} u_3 \\ v_1 \end{bmatrix} = S \begin{bmatrix} u_I \\ v_3 \end{bmatrix} \qquad (73)$$

ce qui donne

$$\mathbf{S} = \begin{bmatrix} \dfrac{a_1 \cdot a_2}{1-c_2 b_1} & b_2 + \dfrac{a_2 b_1 d_2}{1-c_2 b_1} \\ c_1 + \dfrac{a_1 c_2 d_1}{1-c_2 b_1} & \dfrac{d_1 d_2}{1-c_2 b_1} \end{bmatrix} \tag{74}$$

Ainsi le produit étoile est défini éxplicitement par:

$$\mathbf{S} = \mathbf{S}_1 \mathbf{*} \mathbf{S}_2 \tag{75}$$

Remarque:

S n'a de sens que si $c_2 b_1 \neq 1$, ce qui est évident en terme de refléxion et transmission.

Cas où les quantités ui,vi deviennent des n- vecteurs,Q et S étant alors des matrices(2n.2n) constituées de 4(n.n)matrices A,B,C,D.

Le produit simple donné par(30)se généralise par éxtension par:

$$Q = Q_2 Q_1 = \begin{bmatrix} A_2 A_1 + B_2 C_1 & A_2 B_1 + B_2 D_1 \\ C_2 A_1 + D_2 C_1 & C_2 B_1 + D_2 D_1 \end{bmatrix} \tag{76}$$

Pour le produit étoilé on procède de la même façon que pour le cas scalaire

Posons:

$$S = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{77}$$

ce qui donne

$$S = \begin{bmatrix} \dfrac{a_1 \cdot a_2}{1-c_2 b_1} & b_2 + \dfrac{a_2 b_1 d_2}{1-c_2 b_1} \\ c_1 + \dfrac{a_1 c_2 d_1}{1-c_2 b_1} & \dfrac{d_1 d_2}{1-c_2 b_1} \end{bmatrix} \qquad (74)$$

Ainsi le produit étoile est défini explicitement par:

$$S = S_1 \ast S_2 \qquad (75)$$

Remarque:

S n'a de sens que si $c_2 b_1 \neq 1$, ce qui est évident en terme de refléxion et transmission.

Cas où les quantités ui,vi deviennent des n- vecteurs,Q et S étant alors des matrices(2n.2n) constituées de 4(n.n)matrices A,B,C,D.

Le produit simple donné par (6) se généralise par éxtension par:

$$Q = Q_2 Q_1 = \begin{bmatrix} A_2 A_1 + B_2 C_1 & A_2 B_1 + B_2 D_1 \\ C_2 A_1 + D_2 C_1 & C_2 B_1 + D_2 D_1 \end{bmatrix} \qquad (76)$$

Pour le produit étoilé on procède de la même façon que pour le cas scalaire

Posons:

$$S = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad (77)$$

D'après (71) et (72) on a:

$$V_1 = C_1 U_1 + D_1 V_2$$

$$V_2 = C_2 U_2 + D_2 V_3$$

$$U_2 = A_1 U_1 + B_1 V_2$$

$$U_3 = A_2 U_2 + B_2 V_3$$

Ce qui donne:

$$V_1 = \left[C_1 + D_1(1-C_2B_1)^{-1}C_2A_1\right] U_1 + \left[D_1(1-C_2B_1)^{-1}D_2\right] V_3$$

$$U_3 = A_2\left[1+B_1(1-C_2B_1)^{-1}C_2\right]A_1 \; U_1 + \left[B_2+A_2B_1(1-C_2B_1)^{-1}D_2\right] V_3$$

Soit finalement:

$$\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} * \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} =$$

$$\begin{bmatrix} A_2\left[1+B_1(1-C_2B_1)^{-1}C_2\right]A_1 & B_2+A_2B_1(1-C_2B_1)^{-1}D_2 \\ C_1+D_1(1-C_2B_1)^{-1}C_2A_1 & D_1(1-C_2B_1)^{-1}D_2 \end{bmatrix} \qquad (78)$$

Résultat qui s'identifie au cas scalaire sauf pour le terme (1,1). On peut néanmoins y parvenir en opérant la transformation suivante.

a l'aide de l'identité matricielle:

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \qquad (79)$$

on peut vérifier que

$$I + B(I-CB)^{-1}C = (I-bc)^{-1} \qquad (80)$$

En appliquant (80) il vient une seconde expression du produit étoile:

D'après (71) et (72) on a:

$$V_1 = C_1 U_1 + D_1 V_2$$

$$V_2 = C_2 U_2 + D_2 V_3$$

$$U_2 = A_1 U_1 + B_1 V_2$$

$$U_3 = A_2 U_2 + B_2 V_3$$

Ce qui donne:

$$V_1 = \left[C_1 + D_1(1-C_2B_1)^{-1}C_2A_1\right] U_1 + \left[D_1(1-C_2B_1)^{-1}D_2\right] V_3$$

$$U_3 = A_2\left[1+B_1(1-C_2B_1)^{-1}C_2\right]A_1 \; U_1 + \left[B_2+A_2B_1(1-C_2B_1)^{-1}D_2\right] V_3$$

Soit finalement:

$$\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} * \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} =$$

$$\begin{bmatrix} A_2\left[1+B_1(1-C_2B_1)^{-1}C_2\right]A_1 & B_2+A_2B_1(1-C_2B_1)^{-1}D_2 \\ C_1+D_1(1-C_2B_1)^{-1}C_2A_1 & D_1(1-C_2B_1)^{-1}D_2 \end{bmatrix} \qquad (78)$$

Résultat qui s'identifie au cas scalaire sauf pour le terme (1,1). On peut néanmoins y parvenir en opérant la transformation suivante.

à l'aide de l'identité matricielle:

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \qquad (79)$$

on peut vérifier que

$$I + B(I-CB)^{-1}C = (I-bc)^{-1} \qquad (80)$$

En appliquant (80) il vient une seconde expression du produit étoile:

$$\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \ast \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} = \begin{bmatrix} A_2(I-B_1C_2)^{-1}A_1 & B_2+A_2(I-B_1C_2)^{-1}B_1D_2 \\ C_1+D_1C_2(I-B_1C_2)^{-1}A_1 & D_1(I-C_2B_1)^{-1}D_2 \end{bmatrix} \quad (81)$$

qui n'est définie que si les matrices $(I-B_1C_2)$ et $(I-C_2B_1)$ sont régulières. Ceci étant admis, on rappellera que l'ensemble des matrices(2n.2n) muni de la loi d'addition classique et du produit $Q = Q_2Q_1$ ou du produit étoile $S=S_1 \ast S_2$ forme un anneau et constitue une algèbre.

On va maintenant aborder une seconde voie dont l'interêt est de mettre en évidence un isomorphisme entre le produit $\ast$ et . appelé transformation Ψ.

4-CONSTRUCTION DE L'ISOMORPHISME

Soit le problème élémentaire suivant de propagation:

$$\begin{bmatrix} U_2 \\ V_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U_1 \\ V_2 \end{bmatrix} = S \begin{bmatrix} U_1 \\ V_2 \end{bmatrix} \quad (82)$$

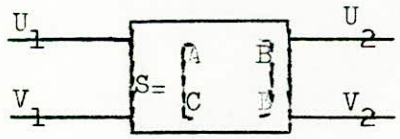


fig 3

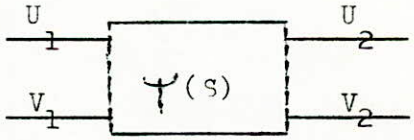


fig 4

Introduisons une transformation Ψ définie par

$$\begin{bmatrix} U_2 \\ V_2 \end{bmatrix} = \Psi(S) \begin{bmatrix} U_1 \\ V_1 \end{bmatrix} \quad (83)$$

Ce résultat est illustré par la figure 4

D'après (82) on a:

$$V_2 = -D^{-1}CU_1 + D^{-1}V_2$$

$$U_2 = (A-BD^{-1}C)U_1 + BD^{-1}V_2$$

$$\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \overset{*}{} \begin{bmatrix} A_2 & B_2 \\ C_2 & D_2 \end{bmatrix} = \begin{bmatrix} A_2(I-B_1C_2)^{-1}A_1 & B_2+A_2(I-B_1C_2)^{-1}B_1D_2 \\ C_1+D_1C_2(I-B_1C_2)^{-1}A_1 & D_1(I-C_2B_1)^{-1}D_2 \end{bmatrix} \quad (81)$$

qui n'est définie que si les matrices $(I-B_1C_2)$ et $(I-C_2B_1)$ sont régulières. Ceci étant admis, on rappellera que l'ensemble des matrices(2n.2n) muni de la loi d'addition classique et du produit $Q = Q_2Q_1$ ou du produit étoile $S=S_1 \overset{*}{} S_2$ forme un anneau et constitue une algèbre.

On va maintenant aborder une seconde voie dont l'intérêt est de mettre en évidence un isomorphisme entre le produit $\overset{*}{}$ et . appelé transformation Ψ.

4-CONSTRUCTION DE L'ISOMORPHISME

Soit le problème élémentaire suivant de propagation:

$$\begin{bmatrix} U_2 \\ V_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U_1 \\ V_2 \end{bmatrix} = S \begin{bmatrix} U_1 \\ V_2 \end{bmatrix} \quad (82)$$

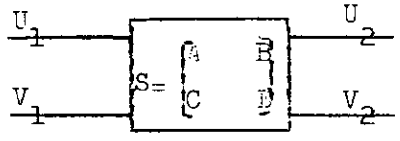


fig 3

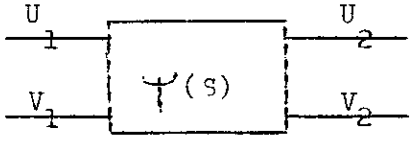


fig 4

Introduisons une transformation Ψ définie par

$$\begin{bmatrix} U_2 \\ V_2 \end{bmatrix} = \Psi(S) \begin{bmatrix} U_1 \\ V_1 \end{bmatrix} \quad (83)$$

Ce résultat est illustré par la figure 4

D'après (82) on a:

$$V_2 = -D^{-1}CU_1 + D^{-1}V_2$$

$$U_2 = (A-BD^{-1}C)U_1 + BD^{-1}V_2$$

d'où

$$\Psi(S)=\begin{bmatrix} A-BD^{-1}C & BD^{-1} \\ -D^{-1}C & D^{-1} \end{bmatrix} = \tilde{S} \qquad (84)$$

Il est aisé de vérifier que

$$\Psi(\tilde{S}) = S \qquad (85)$$

donc pourvu que le bloc D soit régulier, la transformation Ψ vérifie

$$\Psi\left[\Psi(S)\right] = S \Longrightarrow \qquad \Psi^{-1}(S) = \Psi(S) \qquad (86)$$

où Ψ^{-1} est la transformation inverse.

En appliquant ce résultat au problème défini à la figure 2, il vient:

$$\Psi(S) = \Psi(S_2)\cdot\Psi(S_1) \qquad \Longrightarrow \quad S=\Psi\left[\Psi(S_2)\cdot\Psi(S_1)\right] = S_1 * S_2 \qquad (87)$$

où encore d'après (86):

$$\Psi(S_1 * S_2) = \Psi(S_2)\cdot\Psi(S_1) \qquad (88)$$

Inversement en raisonnant à partir de la figure 1 et en passant par l'étape de la figure 2 on aura:

$$\Psi(Q_1) * \Psi(Q_2) = \Psi(Q_2\cdot Q_1) \qquad (89)$$

En résumé, la transformation involutive Ψ définit donc un isomorphisme entre le produit étoile et le produit classique de matrice.

Il faut toutefois noter que cette voie requiert l'hypothèse que D_1 et D_2 soient réguliers.

d'où

$$\Psi(S)=\begin{bmatrix} A-BD^{-1}C & BD^{-1} \\ -D^{-1}C & D^{-1} \end{bmatrix} = \tilde{S} \qquad (84)$$

Il est aisé de vérifier que

$$\Psi(\tilde{S}) = S \qquad (85)$$

donc pourvu que le bloc D soit régulier, la transformation Ψ vérifie

$$\Psi\left[\Psi(S)\right] = S \Longrightarrow \qquad \Psi^{-1}(S) = \Psi(S) \qquad (86)$$

où Ψ^{-1} est la transformation inverse.

En appliquant ce résultat au problème défini à la figure 2, il vient:

$$\Psi(S) = \Psi(S_2)\cdot\Psi(S_1) \qquad \Longrightarrow \quad S=\Psi\left[\Psi(S_2)\cdot\Psi(S_1)\right] = S_1 * S_2 \qquad (87)$$

où encore d'après (86):

$$\Psi(S_1 * S_2) = \Psi(S_2)\cdot\Psi(S_1) \qquad (88)$$

Inversement en raisonnant à partir de la figure 1 et en passant par l'étape de la figure 2 on aura:

$$\Psi(Q_1) * \Psi(Q_2) = \Psi(Q_2\cdot Q_1) \qquad (89)$$

En résumé, la transformation involutive Ψ définit donc un isomorphisme entre le produit étoile et le produit classique de matrice.

Il faut toutefois noter que cette voie requiert l'hypothèse que D_1 et D_2 soient réguliers.

5-APPLICATION A L'EQUATION DE RICCATI DISCRETE

Considérons de nouveau les équations hamiltoniènnes qui s'écrivent:

$$\begin{bmatrix} X_{t+1} \\ \lambda t \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix} \begin{bmatrix} X_t \\ \lambda t+1 \end{bmatrix}$$

Appliquant la transformation Ψ tout en supposant pour l'instant que A est régulière, il vient d'après (84):

$$\begin{bmatrix} X_{t+1} \\ \lambda t+1 \end{bmatrix} = \begin{bmatrix} A + VA^{-T}Q & -VA^{-T} \\ -A^{-T}Q & A^{-T} \end{bmatrix} \begin{bmatrix} X_t \\ \lambda t \end{bmatrix} = S \begin{bmatrix} X_t \\ \lambda_t \end{bmatrix} \qquad (90)$$

où $V = BR^{-1}B^T$ et A^{-T} désigne la transposée de l'inverse

D'après VAUGHAN /IC/, il vient:

$$S = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} J^{-1} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix}^{-1} \qquad (91)$$

J:blocs de Jordan

avec

$$P = W_{21} \cdot W_1^{-1} \qquad (92)$$

Sétant une matrice symplectique(annexe), on peut alors lui associer une matrice hamiltonienne M par transformation bilinéaire(annexe);

il vient d'après (91)

$$M = (S-1)(S+1) = W \begin{bmatrix} (J^{-1}-1)^{-1}(J^{-1}+1) & 0 \\ 0 & (J-1)^{-1}(J+1) \end{bmatrix} W^{-1}$$

5-APPLICATION A L'EQUATION DE RICCATI DISCRETE

Considérons de nouveau les équations hamiltoniènnes qui s'écrivent:

$$\begin{bmatrix} X_{t+1} \\ \lambda_t \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix} \begin{bmatrix} X_t \\ \lambda_{t+1} \end{bmatrix}$$

Appliquant la transformation Ψ tout en supposant pour l'instant que A est régulière, il vient d'après (84):

$$\begin{bmatrix} X_{t+1} \\ \lambda_{t+1} \end{bmatrix} = \begin{bmatrix} A + VA^{-T}Q & -VA^{-T} \\ -A^{-T}Q & A^{-T} \end{bmatrix} \begin{bmatrix} X_t \\ \lambda_t \end{bmatrix} = S \begin{bmatrix} X_t \\ \lambda_t \end{bmatrix} \qquad (90)$$

où $V = BR^{-1}B^T$ et A^{-T} désigne la transposée de l'inverse

D'après VAUGHAN [IC], il vient:

$$S = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} J^{-1} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix}^{-1} \qquad (91)$$

J:blocs de Jordan

avec

$$P = W_{21} \cdot W_1^{-1} \qquad (92)$$

Sétant une matrice symplectique(annexe), on peut alors lui associer une matrice hamiltonienne M par transformation bilinéaire(annexe);

il vient d'après (91)

$$M = (S-1)(S+1) = W \begin{bmatrix} (J^{-1}-1)^{-1}(J^{-1}+1) & 0 \\ 0 & (J-1)^{-1}(J+1) \end{bmatrix} W^{-1}$$

$$= W \begin{bmatrix} -\Lambda & 0 \\ 0 & \Lambda \end{bmatrix} W^{-1} \qquad (93)$$

avec $\Lambda = (J-1)^{-1}(J+1)$, réel $[\lambda(\Lambda)] > 0$ (94)

La matrice signe de H s'écrit alors:

$$\text{signe}(H) = Z \begin{bmatrix} \cdot & & 0 \\ & \text{signe}(\lambda i) & \\ 0 & & \cdot \end{bmatrix} Z^{-1} \; ; \quad M = ZJZ^{-1} \qquad (95)$$

J étant la forme de Jordan

$$\text{signe}(\lambda) = \begin{cases} +1 & \text{si réel}(\lambda) > 0 \\ -1 & \text{si réel}(\lambda) < 0 \end{cases} \qquad (96)$$

tenant compte des relations (93) et (94), on a alors:

$$\text{signe}(H) = W \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} W^{-1} \qquad (97)$$

Introduisons alors la matrice:

$$F = \text{signe}(H) + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = W \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} W^{-1} + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

d'où

$$F = 2 \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} -V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix} \quad \text{avec} \quad W^{-1} = \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix}$$

soit finalement d'après (92)

$$F_{21} F_1^{-1} = W_{21} W_1^{-1} = P$$

Un point important est qu'en définitive, on peut s'affranchir de l'hypothèse que l'on avait faite: Λ régulière. En effet la matrice symplectique S peut s'écrire:

$$= W \begin{bmatrix} -\Lambda & 0 \\ 0 & \Lambda \end{bmatrix} W^{-1} \qquad (93)$$

avec $\Lambda = (J-1)^{-1}(J+1)$, réel $[\lambda(\Lambda)] > 0$ (94)

La matrice signe de M s'écrit alors:

$$\text{signe}(M) = Z \begin{bmatrix} \cdot & & 0 \\ & \text{signe}(\lambda i) & \\ 0 & & \cdot \end{bmatrix} Z^{-1} \; ; \quad M = ZJZ^{-1} \qquad (95)$$

J étant la forme de Jordan

$$\text{signe}(\lambda) = \begin{cases} +1 & \text{si réel}(\lambda) > 0 \\ -1 & \text{si réel}(\lambda) < 0 \end{cases} \qquad (96)$$

tenant compte des relations (93) et (94), on a alors:

$$\text{signe}(M) = W \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} W^{-1} \qquad (97)$$

Introduisons alors la matrice:

$$F = \text{signe}(M) + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = W \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} W^{-1} + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

d'où

$$F = 2 \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} -V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix} \quad \text{avec} \quad W^{-1} = \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix}$$

soit finalement d'après (92)

$$F_{21} F_1^{-1} = W_{21} W_1^{-1} = P$$

Un point important est qu'en définitive, on peut s'affranchir de l'hypothèse que l'on avait faite: A régulière. En effet la matrice symplectique S peut s'écrire:

$$S = \begin{bmatrix} I & BR^{-1}B^T \\ 0 & A^T \end{bmatrix} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} = U^{-1}.L$$

d'où

$$H = (S-1)^{-1}(s+1) = (U^{-1}L-1)^{-1}(U^{-1}L+1) = (L-U)^{-1}(L+U)$$

On remarque dans cette expression que A^{-1} n'apparait plus. Par ailleurs H étant hamiltonienne sans valeurs propres sur l'axe imaginaire, donc L-U est régulière ainsi que L+U.

En résumé, l'algorithme que l'on se propose est défini par:

$$F = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix} = \left[\text{signe}(L-U)^{-1}(L+U) \right] + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (98)$$

6-COMPARAISON AVEC D'AUTRES METHODES

L'étude comparative que l'on développe maintenant est décomposée en deux parties respectivement consacrées aux méthodes à caractère implicite et celles à formulation itérative explicite.

6-1-Methodes à caractère implicite

Les deux méthodes qui font l'objet de ce paragraphe reposent, tout comme la méthode de la fonction signe de matrice, sur les propriétés de la matrice symplectique(90)associée à l'équation de Riccati. Cependant les deux méthodes supposent que la matrice A est régulière. De plus, comme pour la méthode de la fonction signe de matrice, aucune ne permet de garantire le caractère symétrique non négatif de P, par une approche du type "square root".

6-1-1-Méthode de VAUGHAN

Nous considérons d'abord la méthode de VAUGHAN [10]; sans doute la plus classique, qui peut se résumer dans la relation:

$$S = \begin{bmatrix} I & BR^{-1}B^T \\ C & A^T \end{bmatrix} \begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix} = U^{-1}.L$$

d'où

$$H = (S-1)^{-1}(s+1) = (U^{-1}L-1)^{-1}(U^{-1}L+1) = (L-U)^{-1}(L+U)$$

On remarque dans cette expression que A^{-1} n'apparait plus. Par ailleurs H étant hamiltonienne sans valeurs propres sur l'axe imaginaire, donc L-U est régulière ainsi que L+U.

En résumé, l'algorithme que l'on se propose est défini par:

$$F = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix} = \left[\text{signe}(L-U)^{-1}(L+U) \right] + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (98)$$

6-COMPARAISON AVEC D'AUTRES METHODES

L'étude comparative que l'on développe maintenant est décomposée en deux parties respectivement consacrées aux méthodes à caractère implicite et celles à formulation itérative explicite.

6-1-Methodes à caractère implicite

Les deux méthodes qui font l'objet de ce paragraphe reposent, tout comme la méthode de la fonction signe de matrice, sur les propriétés de la matrice symplectique(90)associée à l'équation de Riccati. Cependant les deux méthodes supposent que la matrice A est régulière. De plus, comme pour la méthode de la fonction signe de matrice, aucune ne permet de garantire le caractère symétrique non négatif de P, par une approche du type "square root".

6-1-1-Méthode de VAUGHAN

Nous considérons d'abord la méthode de VAUGHAN [10]; sans doute la plus classique, qui peut se résumer dans la relation:

$$P = W_{21} W_1^{-1} \qquad (99)$$

avec (90) on a:

$$S = \begin{bmatrix} A+VA^{-T}Q & -VA^{-T} \\ -A^{-T}Q & A^{-T} \end{bmatrix} = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} J^{-1} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix}^{-1} \qquad (100)$$

et

$|\lambda(J)| > 1$ J blocs de Jordan

En fait, dans la mesure où la décomposition sous forme de Jordan est impossible, ce résultat ne peut constituer la base d'un algorithme que si S est strictement diagonalisble.

Il n'en reste pas moins vrai que peuvent toujours subsister les problèmes liés au mauvais conditionnement (nombre de condition de W élevé) ou encore la grande sensibilité de certaines valeurs propres voisines entre elles ou/et de 1 (en module).

6-1-2-Méthode de LAUB

L'alternative proposée récement par Laub [11], constitue un progrès considérable par rapport aux résultats précédents dans la mesure où elle supprime l'essentielles difficultés numériques.

Sa méthode repose sur le fait qu'une relation du type (99) peut être obtenue sans imposer que S soit mis sous forme de Jordan, mais seulement sous forme de Schur réelle:

$$S = \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & \Lambda_{12} \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix}^T \qquad (101a)$$

$$|\lambda(\Lambda_1)| < 1 \qquad |\lambda(\Lambda_2)| > 1$$

Λ_1 et Λ_2 ne sont plus des blocs de Jordan inverses l'un de l'autre, mais

$$P = W_{21} W_1^{-1} \qquad (99)$$

avec (90) on a:

$$S = \begin{bmatrix} A+VA^{-T}Q & -VA^{-T} \\ -A^{-T}Q & A^{-T} \end{bmatrix} = \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix} \begin{bmatrix} J^{-1} & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} W_1 & W_{12} \\ W_{21} & W_2 \end{bmatrix}^{-1} \qquad (100)$$

et

$|\lambda(J)| > 1$ J blocs de Jordan

En fait, dans la mesure où la décomposition sous forme de Jordan est impossible, ce résultat ne peut constituer la base d'un algorithme que si S est strictement diagonalisble.

Il n'en reste pas moins vrai que peuvent toujours subsister les problèmes liés au mauvais conditionnement (nombre de condition de W élevé) ou encore la grande sensibilité de certaines valeurs propres voisines entre elles ou/et de 1 (en module).

6-1-2-Méthode de LAUB

L'alternative proposée récement par Laub [11], constitue un progrès considérable par rapport aux résultats précédents dans la mesure où elle supprime l'essentielles difficultés numériques.

Sa méthode repose sur le fait qu'une relation du type (99) peut être obtenue sans imposer que S soit mis sous forme de Jordan, mais seulement sous forme de Schur réelle:

$$S = \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & \Lambda_{12} \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} V_1 & V_{12} \\ V_{21} & V_2 \end{bmatrix}^T \qquad (101a)$$

$$|\lambda(\Lambda_1)| < 1 \qquad |\lambda(\Lambda_2)| > 1$$

Λ_1 et Λ_2 ne sont plus des blocs de Jordan inverses l'un de l'autre, mais

ont leur spectre réparti comme J^{-1} et J. De plus la matrice V est orthogonale donc parfaitement conditionnée. Dans ces conditions le calcul de P comme suit ne pose plus de problème particulier

$$P = V_{21} V_1^{-1} \qquad (101b)$$

Quand à la forme de Schur , elle est obtenue à l'aide du package Eispack /12/ et des modules de Stewart /13/.

6-2-ALGORITHMES ITERATIFS

6-2-1-Méthode du produit etoile

La première méthode que l'on se propose d'aborder est en fait une application du produit étoile.

Compte tenu de (90), on a le résultat suivant:

Théorème:

$$P = \lim_{K \to \infty} \left[\Psi(Z_K) \right]_{2,1} \qquad (102)$$

avec $Z_{K+1} = Z_K . S \qquad Z_0 = \begin{bmatrix} 1 & 0 \\ P_0 & 1 \end{bmatrix}$ (103)

$$\forall P_0 = P_0^T \geqslant 0$$

où P est solution de l'équation de Riccati et où $\left[. \right]_{21}$ désigne le bloc (2,1) de dimension n.n des matrices partitionnées 2n.2n.

Ceci étant, on peut maintenant en déduire une seconde relation en terme de produit étoile. Rappelons que (90) s'écrit

$$S = \Psi(M)$$

Posons alors

$$Y_K = \Psi(Z_K) \Longrightarrow Z_K = \Psi(Y_K)$$

$$\Psi(Y_{K+1}) = Z_{K+1} = \Psi(Y_K) . \Psi(M)$$

ont leur spectre réparti comme J^{-1} et J. De plus la matrice V est orthogonale donc parfaitement conditionnée. Dans ces conditions le calcul de P comme suit ne pose plus de problème particulier

$$P = V_{21} V_1^{-1} \tag{101b}$$

Quand à la forme de Schur , elle est obtenue à l'aide du package Eispack /12/ et des modules de Stewart /13/.

6-2-ALGORITHMES ITERATIFS

6-2-1-Méthode du produit etoile

La première méthode que l'on se propose d'aborder est en fait une application du produit étoile.

Compte tenu de (90), on a le résultat suivant:

Théorème:

$$P = \lim_{K \to \infty} \left[\Psi(Z_K) \right]_{2,1} \tag{102}$$

avec $Z_{K+1} = Z_K . S$ $\qquad Z_0 = \begin{bmatrix} 1 & 0 \\ -P_0 & 1 \end{bmatrix}$ (103)

$$\forall P_0 = P_0^T \geq 0$$

où P est solution de l'équation de Riccati et où $\left[\cdot \right]_{21}$ désigne le bloc (2,1) de dimension n.n des matrices partitionnées 2n.2n.

Ceci étant, on peut maintenant en déduire une seconde relation en terme de produit étoile. Rappelons que (90) s'écrit

$$S = \Psi(M)$$

Posons alors

$$Y_K = \Psi(Z_K) \Longrightarrow Z_K = \Psi(Y_K)$$

$$\Psi(Y_{K+1}) = Z_{K+1} = \Psi(Y_K) . \Psi(M)$$

soit

$$Y_{K+1} = M \circledast Y_K \qquad Yo = \begin{bmatrix} 1 & 0 \\ Po & 1 \end{bmatrix}$$

de plus

$$Y_K = \Psi(Z_K) = \begin{bmatrix} * & * \\ \Gamma_K & * \end{bmatrix}$$

donc

$$\Gamma = \lim_{K \longrightarrow \infty} \left[Y_K \right]_{2,1}$$

$$Y_{K+1} = M \circledast Y_K \qquad Yo = \begin{bmatrix} 1 & 0 \\ Po & 1 \end{bmatrix} \quad \forall Po = Po^T \geqslant 0$$

Ce formalisme équivalent à (102)-(103) présente l'avantage de lever la contrainte A régulière.

Un algorithme dit de doublement, dont la convergence est d'ordre 2 s'obtient alors en posant:

$$M_1 = M = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix}$$

$$M_2 = M_1 \circledast M_1$$

$$M_{2^K} = M_{2^{K-1}} \circledast M_{2^{K-1}} \qquad \text{avec} \quad M_{2^K} = \begin{bmatrix} U_K & -V_K \\ W_K & U_K^T \end{bmatrix}$$

d'où tout en simplifiant les notations

$$\left.\begin{aligned} U_{K+1} &= U_+ = U(I+VW)^{-1}U & Uo &= A & (a)\\ V_{K+1} &= V_+ = V + U(I+VW)^{-1}VU^T & Vo &= BR^{-1}B^T & (b)\\ W_{K+1} &= W_+ = W + U^T W(I+VW)^{-1}U & Wo &= Q & (c)\\ \Gamma &= \lim_{K \longrightarrow \infty} W_K = P_{2^K} & & & (d) \end{aligned}\right\} \quad (104)$$

soit

$$Y_{K+1} = M \ast Y_K \qquad Yo = \begin{bmatrix} 1 & 0 \\ Po & 1 \end{bmatrix}$$

de plus

$$Y_K = \Psi(Z_K) = \begin{bmatrix} \ast & \ast \\ \Gamma_K & \ast \end{bmatrix}$$

donc

$$P = \lim_{K \longrightarrow \infty} \left[Y_K \right]_{2,1}$$

$$Y_{K+1} = M \ast Y_K \qquad Yo = \begin{bmatrix} 1 & 0 \\ Po & 1 \end{bmatrix} \quad \forall Po = Po^T \geqslant 0$$

Ce formalisme équivalent à (102)-(103) présente l'avantage de lever la contrainte A régulière.

Un algorithme dit de doublement, dont la convergence est d'ordre 2 s'obtient alors en posant:

$$M_1 = M = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix}$$

$$M_2 = M_1 \ast M_1$$

$$M_{2^K} = M_{2^{K-1}} \ast M_{2^{K-1}} \qquad \text{avec} \quad M_{2^K} = \begin{bmatrix} U_K & -V_K \\ W_K & U_K^T \end{bmatrix}$$

d'où tout en simplifiant les notations

$$\left.\begin{aligned} U_{K+1} &= U_+ = U(I+VW)^{-1}U & Uo &= A & (a) \\ V_{K+1} &= V_+ = V + U(I+VW)^{-1}VU^T & Vo &= BR^{-1}B^T & (b) \\ W_{K+1} &= W_+ = W + U^T W(I+VW)^{-1}U & Wo &= Q & (c) \\ P &= \lim_{K \longrightarrow \infty} W_K = P_{2^K} & & & (d) \end{aligned}\right\} \quad (104)$$

(b) et (c) sont des équations de Riccati écrites sous leur forme équivalente (annexe). On pose alors

$$V = X^T.X \quad \text{et} \quad W = Y^T.Y$$

Algorithme: calculer $D = X.U^T$, $E = X.Y^T$, $F = Y.U$

1º/ réaliser la triangularisation

$$\begin{bmatrix} I & O \\ E & D \\ O & X \end{bmatrix} \begin{bmatrix} K & L \\ O & X_+ \\ O & O \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ O & \cdot \end{bmatrix} \tag{105}$$

au moyen de la matrice orthogonale τ [15], d'où X_+ triangulaire supérieure

2º/ résoudre le système triangulaire inférieur

$$K^T.M = F$$

réaliser la triangularisation "orthogonale":

$$\tau' \begin{bmatrix} Y \\ M \end{bmatrix} = \begin{bmatrix} Y_+ \\ O \end{bmatrix} = \begin{bmatrix} \cdot \\ O \end{bmatrix}$$

dont on tire Y_+ triangulaire supérieur.

3º/ (a) Y de rang maximal (W régulier)

-résoudre les systèmes triangulaires inférieurs:

$$Y^T.H = U^T \quad \text{puis} \quad K^T.G = H$$

-calculer

$$U_+ = G^T.M$$

(b) Y quelconque (W singulier):

-calculer $G = I + X^T E Y$

-résoudre le système linéaire $G.H = U$

-calculer $U_+ = U.H$ (106)

Le coût total (en multiplications) de cet algorithme est de $7n^3$ approximativement.

(b) et (c) sont des équations de Riccati écrites sous leur forme équivalente (annexe). On pose alors

$$V = X^T.X \quad \text{et} \quad W = Y^T.Y$$

Algorithme: calculer $D = X.U^T$, $E = X.Y^T$, $F = Y.U$

1º/ réaliser la triangularisation

$$\begin{bmatrix} I & O \\ E & D \\ O & X \end{bmatrix} \begin{bmatrix} K & L \\ O & X_+ \\ O & O \end{bmatrix} = \begin{bmatrix} \times & \times \\ & \times \\ O & \end{bmatrix} \qquad (105)$$

au moyen de la matrice orthogonale Σ [15], d'où X_+ triangulaire supérieure

2º/ résoudre le système triangulaire inférieur

$$K^T.M = F$$

réaliser la triangularisation "orthogonale":

$$\Sigma' \begin{bmatrix} Y \\ M \end{bmatrix} = \begin{bmatrix} Y_+ \\ O \end{bmatrix} = \begin{bmatrix} \times & \times \\ & \times \\ O & \end{bmatrix}$$

dont on tire Y_+ triangulaire supérieur.

3º/ (a) Y de rang maximal (W régulier)

-résoudre les systèmes triangulaires inférieurs:

$$Y^T.H = U^T \quad \text{puis} \quad K^T.G = H$$

-calculer

$$U_+ = G^T.M$$

(b) Y quelconque (W singulier):

-calculer $G = I + X^T E Y$

-résoudre le système linéaire $G.H = U$

-calculer $U_+ = U.H$ (106)

Le coût total (en multiplications) de cet algorithme est de $7n^3$ approximativement.

6-2-2-Méthode de HEWER

La deuxième alternative que l'on se propose de rappeler est dûe à Hewer [7].

Le principe de l'algorithme consiste à résoudre l'équation de Riccati par la méthode de Newton; ce qui conduit au processus itératif suivant:

$$M_j - \tilde{A}_j^T M_j \tilde{A}_j = G_j^T R G_j + Q \qquad (107a)$$

$$G_j = (B^T M_{j-1} B + R)^{-1} B^T M_{j-1} A \qquad (107b)$$

$$\tilde{A}_j = A - B G_j \qquad (107c)$$

où apparait donc à chaque pas la résolution d'une équation de Lyapunov (107a). On sait alors que la suite M_j présente une convergence monotone décroissante d'ordre 2 vers la solution P cherchée quelque soit l'initialisation Go tel que

$$|\lambda(\tilde{A}o = A - BGo)| < 1$$

avec

$$P_{K+1} = A^T P_K A - A^T P_K B (B^T P_K B + R)^{-1} B^T P_K A + Q \quad Po=0 \qquad (108)$$

Sur le plan coûtcalcul, chaque itération est dominée par la résolution de l'équation de Lyapunov (107a). Le coût global de (107) s'élève en moyenne à environ [8] : $20n^3$ ($m < n$, B: m.n).

En conséquence, cet algorithme est à priori coûteux, chose à laquelle s'ajoute la non garantie sur le caractère symètrique non négatif de la solution obtenue et un logiciel beaucoup plus lourd (éq de Lyapunov).

6-2-3-Méthode classique stabilisée

Une seconde formulation itérative est directement fournie par(108) ou sa version dite stabilisée (annexe):

$$P_{K+1} = \tilde{A}_K^T P_K A + G_K^T R G_K + Q$$

6-2-2-Méthode de HEWER

La deuxième alternative que l'on se propose de rappeler est dûe à Hewer [7].

Le principe de l'algorithme consiste à résoudre l'équation de Riccati par la méthode de Newton; ce qui conduit au processus itératif suivant:

$$M_j - \tilde{A}_j^T M_j \tilde{A}_j = G_j^T R G_j + Q \qquad (107a)$$

$$G_j = (B^T M_{j-1} B + R)^{-1} B^T M_{j-1} A \qquad (107b)$$

$$\tilde{A}_j = A - B G_j \qquad (107c)$$

où apparait donc à chaque pas la résolution d'une équation de Lyapunov (107a). On sait alors que la suite M_j présente une convergence monotone décroissante d'ordre 2 vers la solution P cherchée quelque soit l'initialisation Go tel que

$$\left| \lambda(\tilde{A}o = A - BGo) \right| < 1$$

avec

$$P_{K+1} = A^T P_K A - A^T P_K B (B^T P_K B + R)^{-1} B^T P_K A + Q \quad Po=0 \qquad (108)$$

Sur le plan coûtcalcul, chaque itération est dominée par la résolution de l'équation de Lyapunov (107a). Le coût global de (107) s'élève en moyenne à environ [8] : $20n^3 (m < n,\ B: m.n)$.

En conséquence, cet algorithme est à priori coûteux, chose à laquelle s'ajoute la non garantie sur le caractère symètrique non négatif de la solution obtenue et un logiciel beaucoup plus lourd (éq de Lyapunov).

6-2-3-Méthode classique stabilisée

Une seconde formulation itérative est directement fournie par(108) ou sa version dite stabilisée (annexe):

$$P_{K+1} = \tilde{A}_K^T P_K A + G_K^T R G_K + Q$$

$$\left.\begin{aligned} G_K &= (B^T \Gamma_K P + R)^{-1} B^T \Gamma_K A \\ \tilde{A}_K &= A - BG_K \end{aligned}\right\} \qquad (109)$$

qu'il ne faut pas confondre, malgré sa ressemblance avec le processus itératif de Hewer(107).

6-2-4-Square root d'ordre 1

A la différence de la méthode de Hewer, on peut réaliser une formulation du type square root de (108) garantissant le caractère symétrique non négatif de la solution chérchée.

$$\left.\begin{aligned} R &= X^T.X \qquad Q = V^T.V \\ \Gamma_K &= S_K^T.S_K \qquad S_K \text{ triangulaire supérieure} \end{aligned}\right\} \qquad (110)$$

itération K:

-calculer $\quad D = S_K.B \qquad E = S_K.A \qquad (111)$

-réaliser la triangularisation orthogonale

$$T \begin{bmatrix} X & Q \\ D & E \\ O & V \end{bmatrix} = \begin{bmatrix} L & K \\ O & S_{K+1} \\ O & O \end{bmatrix} \qquad (112)$$

Le coût calcul pour $m < n$ (B:n.m), reste inférieur à $3n^3$. Neanmoins, il apparaît clairement dans le tableau suivant, que le coût relatif à cette méthode est assez grand lorsque le régime pérmanant est suffisament long à atteindre, c'est à dire lorsque le temps de réponse du système bouclé est important.

Tableau :II

$N = 2^K$	$C = N.3n^3$
2	$6n^3$
4	$12n^3$
8	$24n^3$
16	$48n^3$
32	$96n^3$
64	$192n^3$

N:nombre d'itérations

C:coût calcul

$$\left.\begin{aligned} G_K &= (B^T \Gamma_K P + R)^{-1} B^T \Gamma_K A \\ \tilde{A}_K &= A - BG_K \end{aligned}\right\} \qquad (109)$$

qu'il ne faut pas confondre, malgré sa ressemblance avec le processus itératif de Hewer(107).

6-2-4-Square root d'ordre 1

A la différence de la méthode de Hewer, on peut réaliser une formulation du type square root de (108) garantissant le caractère symétrique non négatif de la solution chérchée.

$$\left.\begin{aligned} R = X^T.X \qquad & Q = V^T.V \\ \Gamma_K = S_K^T.S_K \qquad & S_K \text{ triangulaire supérieure} \end{aligned}\right\} \qquad (110)$$

itération K:

-calculer $\quad D = S_K.B \qquad E = S_K.A. \qquad (111)$

-réaliser la triangularisation orthogonale

$$T \begin{bmatrix} X & Q \\ D & E \\ O & V \end{bmatrix} = \begin{bmatrix} L & K \\ O & S_{K+1} \\ O & O \end{bmatrix} \qquad (112)$$

Le coût calcul pour $m < n$ (B:n.m), reste inférieur à $3n^3$. Neanmoins, il apparaît clairement dans le tableau suivant, que le coût relatif à cette méthode est assez grand lorsque le régime pérmanant est suffisament long à atteindre, c'est à dire lorsque le temps de réponse du système bouclé est important.

Tableau :II

$N = 2^K$	$C = N.3n^3$
2	$6n^3$
4	$12n^3$
8	$24n^3$
16	$48n^3$
32	$96n^3$
64	$192n^3$

N:nombre d'itérations

C:coût calcul

7-CONCLUSIONS

Il est clair que quels qu'en soient les fondements théoriques , toutes les techniques numériques permettant de calculer P sont essentiellement itératives, dès lors on peut les classer par exemple, suivant leur complexité de mise en oeuvre, leur stabilité numérique, leur coût calcul par itération leur vitesse de convergence...

A ceci on peut ajouter leur domaine d'application: A régulière par éxemple ou encore leur garantie concernant le caractère symétrique non négatif de P. C'est dans cet ésprit que l'on va dégager parmi les huit méthodes, celles qui sont succeptibles de constituer les éléments de base d'un logiciel pour la résolution des équations de Riccati.

1-Méthode classique: MC équation (108)

2-Méthode classique stabilisée: MCS équation (109)

3-Square root d'ordre 1: SR1 équations (111)-(112)

4-Square root d'ordre 2:(produit $*$): SR2 équatios (105)-(106)

5-Newton: N équation (107)

6-Vaughan: V équations (99)-(100)

7-Laub: L équations (101a)-(101b)

8-Fonction signe: FS équation (98)

On peut déjà noter que les cinq dernières présentent une convergence d'ordre 2 dont seule SR2 garantit le caractère symétrique non négatif de P.
Par ailleurs V et L supposent A régulière.
Sur le plan de la stabilité numérique, V peut être éliminée d'emblée (mise sous forme de Jordan, conditionnement) comparativement à L et FS qui oppèrent sur les matrices symplectiques.Il en va de même des méthodes d'ordre 1: MC et MCS malgré son appélation stabilisée.
Deux points restent à considirer la mise en oeuvre et le coût calcul

7-CONCLUSIONS

Il est clair que quels qu'en soient les fondements théoriques , toutes les techniques numériques permettant de calculer P sont essentiellement itératives, dès lors on peut les classer par exemple, suivant leur complexité de mise en oeuvre, leur stabilité numérique, leur coût calcul par itération leur vitesse de convergence...

A ceci on peut ajouter leur domaine d'application: A régulière par exemple ou encore leur garantie concernant le caractère symétrique non négatif de P. C'est dans cet éspirit que l'on va dégager parmi les huit méthodes, celles qui sont succeptibles de constituer les éléments de base d'un logiciel pour la résolution des équations de Riccati.

1-Méthode classique: MC équation (108)

2-Méthode classique stabilisée: MCS équation (109)

3-Square root d'ordre 1: SR1 équations (111)-(112)

4-Square root d'ordre 2:(produit *): SR2 équations (105)-(106)

5-Newton: N équation (107)

6-Vaughan: V équations (99)-(100)

7-Laub: L équations (101a)-(101b)

8-Fonction signe: FS équation (98)

On peut déjà noter que les cinq dernières présentent une convergence d'ordre 2 dont seule SR2 garantit le caractère symétrique non négatif de P.
Par ailleurs V et L supposent A régulière.
Sur le plan de la stabilité numérique, V peut être éliminée d'emblée (mise sous forme de Jordan, conditionnement) comparativement à L et FS qui oppèrent sur les matrices symplectiques.Il en va de même des méthodes d'ordre 1: MC et MCS malgré son appélation stabilisée.
Deux points restent à considirer la mise en oeuvre et le coût calcul

qu'on peut apprécier à l'aide du tableau 3 et ce pour chaque méthode. Ainsi deux méthodes nous semblent particulièrement interessantes. Tout d'abord la version SR2 qui est peut être potentièllement la plus séduisante, car pour un codage de complexité moyenne, elle offre une convergence d'ordre 2 tout en garantissant numériquement le caractère symetrique non négatif de la solution cherchée. Et la méthode F.S que l'on a développée et qui semble être très compététive sur le plan rapidité, est également la plus générale.

En effet le même code peut servir pour résoudre les équations de Riccati relatives aux systèmes discrets que celles relatives aux systèmes continus. Seuls les codages de l'hamiltonien et de F changent.

RICCATI

continu: $PA + A^T P - PVP + Q = 0$

discret: $A^T P(I + VP)^{-1} A - P + Q = 0$

avec $V = BR^{-1}B^T$

Enfin si l'on pose V=0, le même algorithme permet de résoudre les équations de Lyapunov correspondantes

LYAPUNOV

continu: $PA + A^T P + Q = 0$

discret: $A^T PA - P + Q = 0$

qu'on peut apprécier à l'aide du tableau 3 et ce pour chaque méthode. Ainsi deux méthodes nous semblent particulièrement interessantes. Tout d'abord la version SR2 qui est peut être potentièllement la plus séduisante, car pour un codage de complexité moyenne, elle offre une convergence d'ordre 2 tout en garantissant numériquement le caractère symetrique non négatif de la solution cherchée. Et la méthode F.S que l'on a développée et qui semble être très compététive sur le plan rapidité, est également la plus générale.

En effet le même code peut servir pour résoudre les équations de Riccati relatives aux systèmes discrets que celles relatives aux systèmes continus. Seuls les codages de l'hamiltonien et de Γ changent.

RICCATI

continu: $PA + A^T P - PVP + Q = 0$

discret: $A^T P(I + V\Gamma)^{-1} A - P + Q = 0$

avec $V = BR^{-1}B^T$

Enfin si l'on pose V=0, le même algorithme permet de résoudre les équations de Lyapunov correspondantes

LYAPUNOV

continu: $PA + A^T P + Q = 0$

discret: $A^T PA - P + Q = 0$

TABLEAU 3

	MC	MCS	SR1	SR2	N	V	L	F.S
Ordre de convergence	1	1	**1**	2	2	2	2	2
A régulière	NON	NON	NON	NON	NON	OUI	OUI	NON
$P \geqslant 0$	NON	NON	OUI	OUI	NON	NON	NON	NON
stabilité numérique	NON	NON	OUI	OUI	OUI	NON	OUI	OUI
mise en oeuvre	S	S	S	M	I	M	I	S
coût CALCUL	# 5 n^3 ($m = n/2$)	# 5 n^3 ($m = n/2$)	# 2 n^3 ($m = n/2$)	7 n^3 $\forall m$	3 n^3 ($m = n/2$) + 20 n^3 (eq. Lyap)	70 n^3 (*)	75 n^3 (*)	5 $n^3 \forall m$ (20 à 30 n^3) (*)

* coût total pour obtenir P

S simple

M moyenne

I importante

TABLEAU 3

	MC	MCS	SR1	SR2	N	V	L	F.S
Ordre de convergence	1	1	1	2	2	2	2	2
A régulière	NON	NON	NON	NON	NON	OUI	OUI	NON
$P \geqslant 0$	NON	NON	OUI	OUI	NON	NON	NON	NON
stabilité numérique	NON	NON	OUI	OUI	OUI	NON	OUI	OUI
mise en oeuvre	S	S	S	M	I	M	I	S
coût calcul	$\#5\ n^3$ $(m = n/2)$	$\#5n^3$ $(m = n/2)$	$\#2\ n^3$ $(m = n/2)$	$7\ n^3$ $\forall m$	$3\ n^3$ $(m = n/2)$ $+20\ n^3$ (eq. Lyap)	$70\ n^3$ (*)	$75\ n^3$ (*)	$5\ n^3 \forall m$ (20 à 30 n^3) (*)

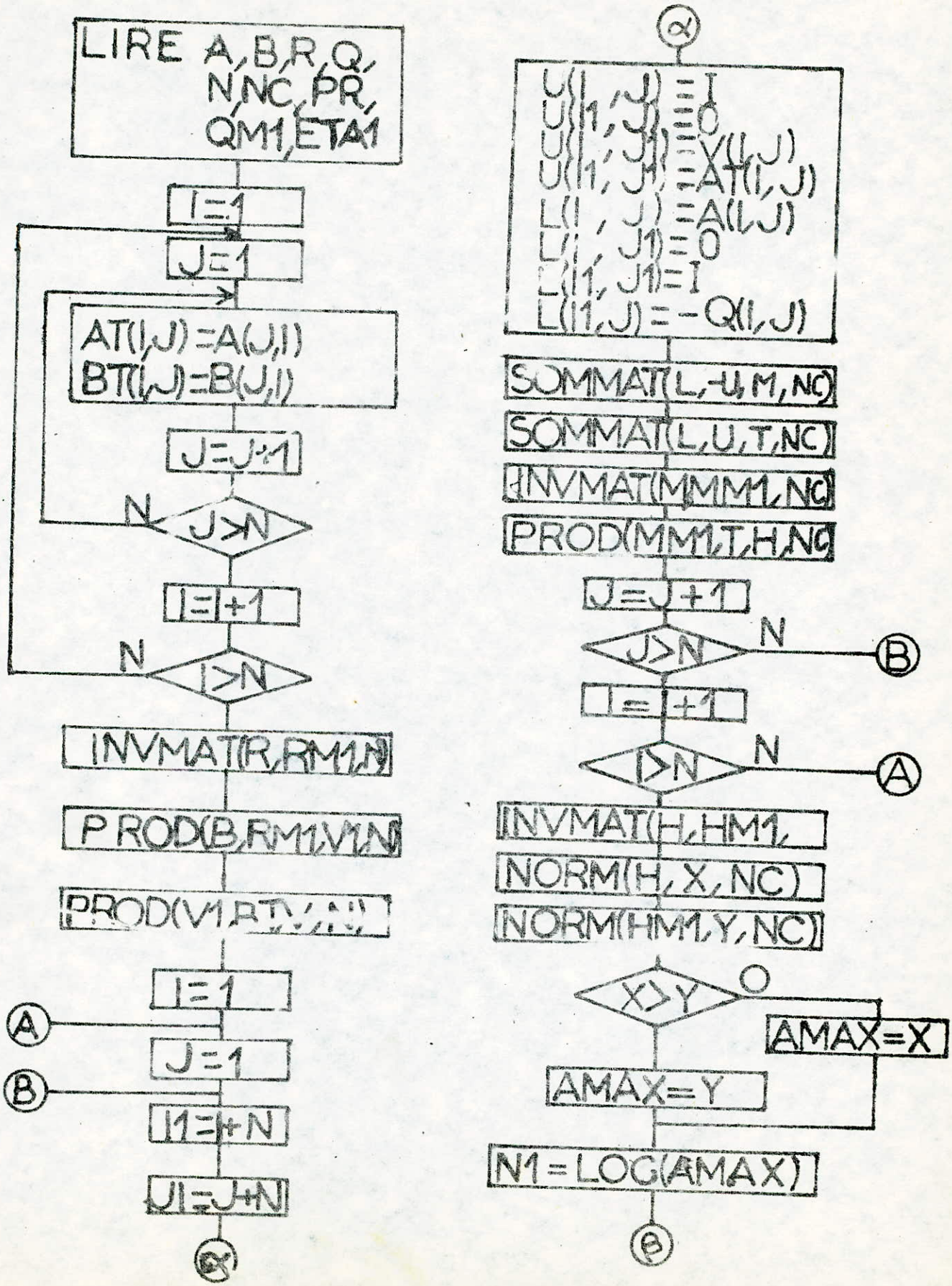
* coût total pour obtenir P

S simple

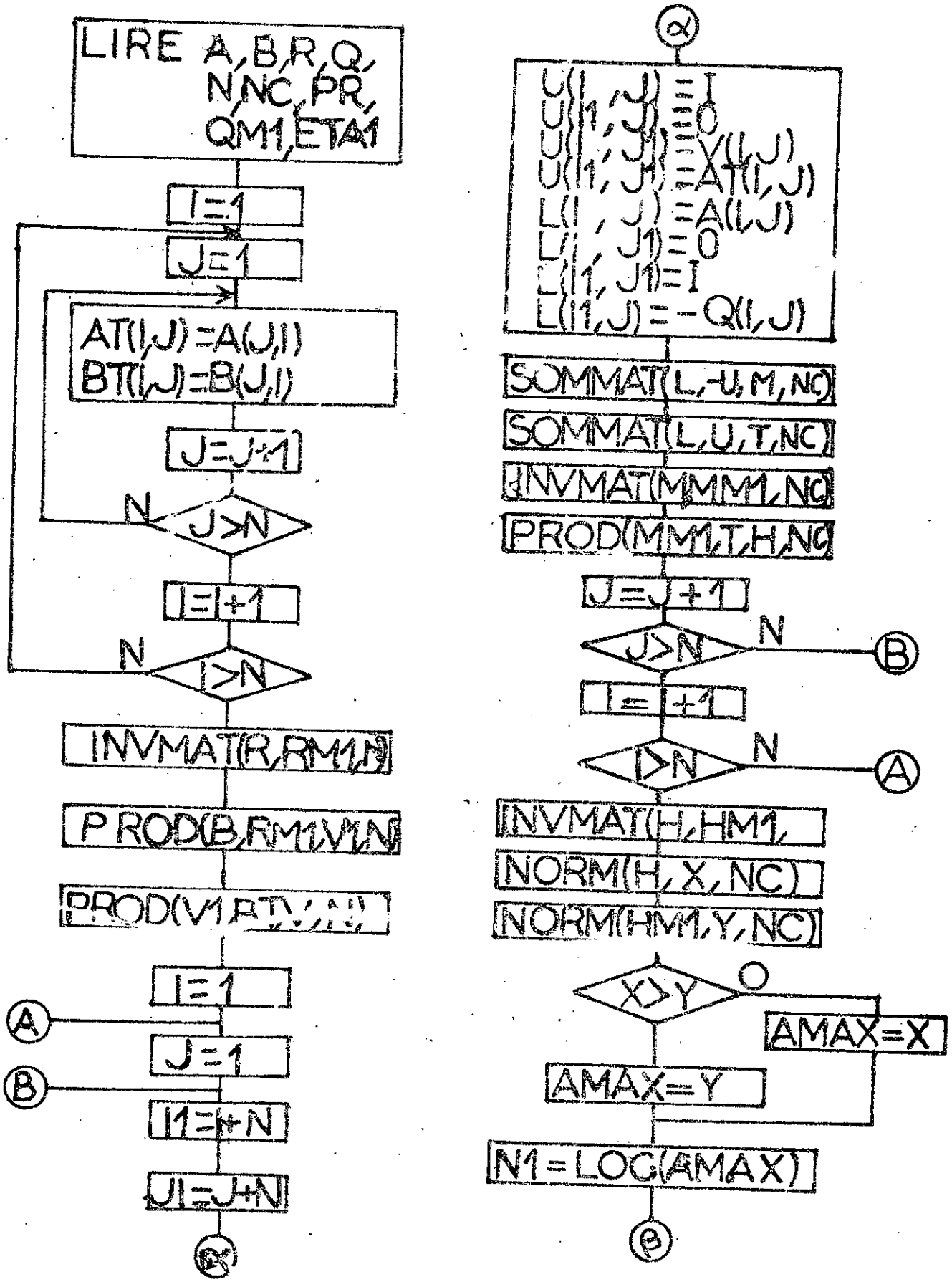
M moyenne

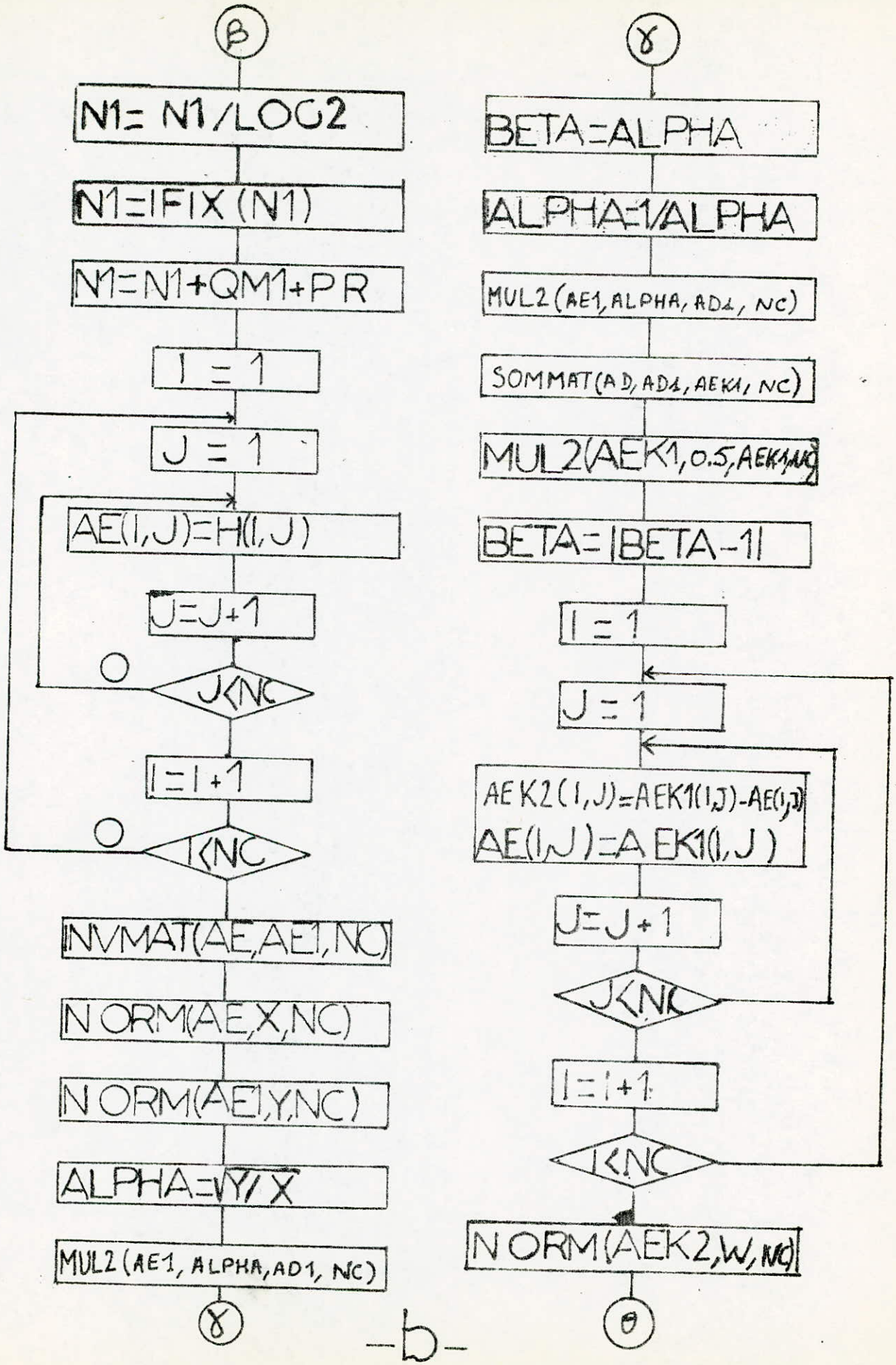
I importante

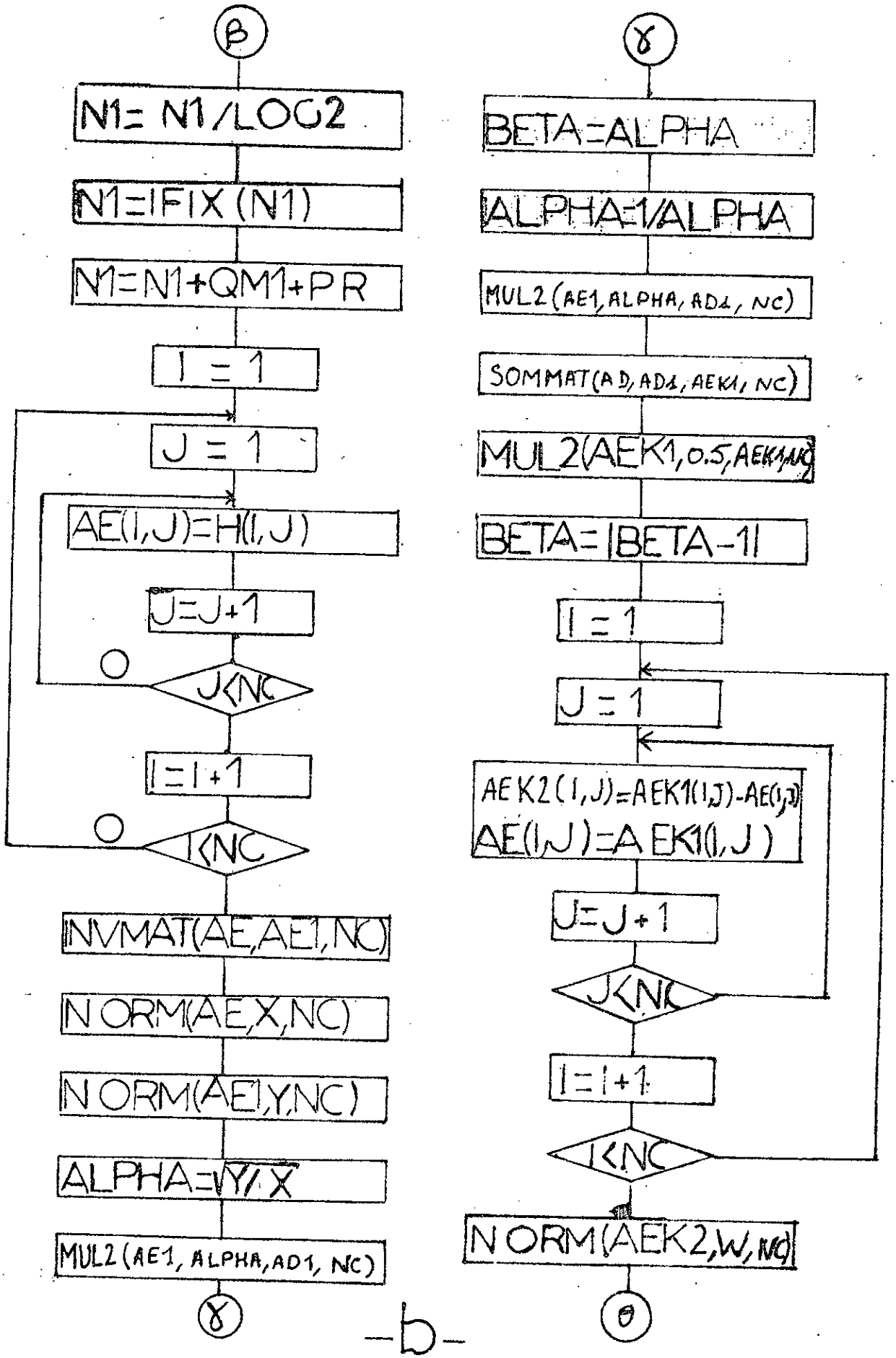
ORGANIGRAMME PRINCIPAL

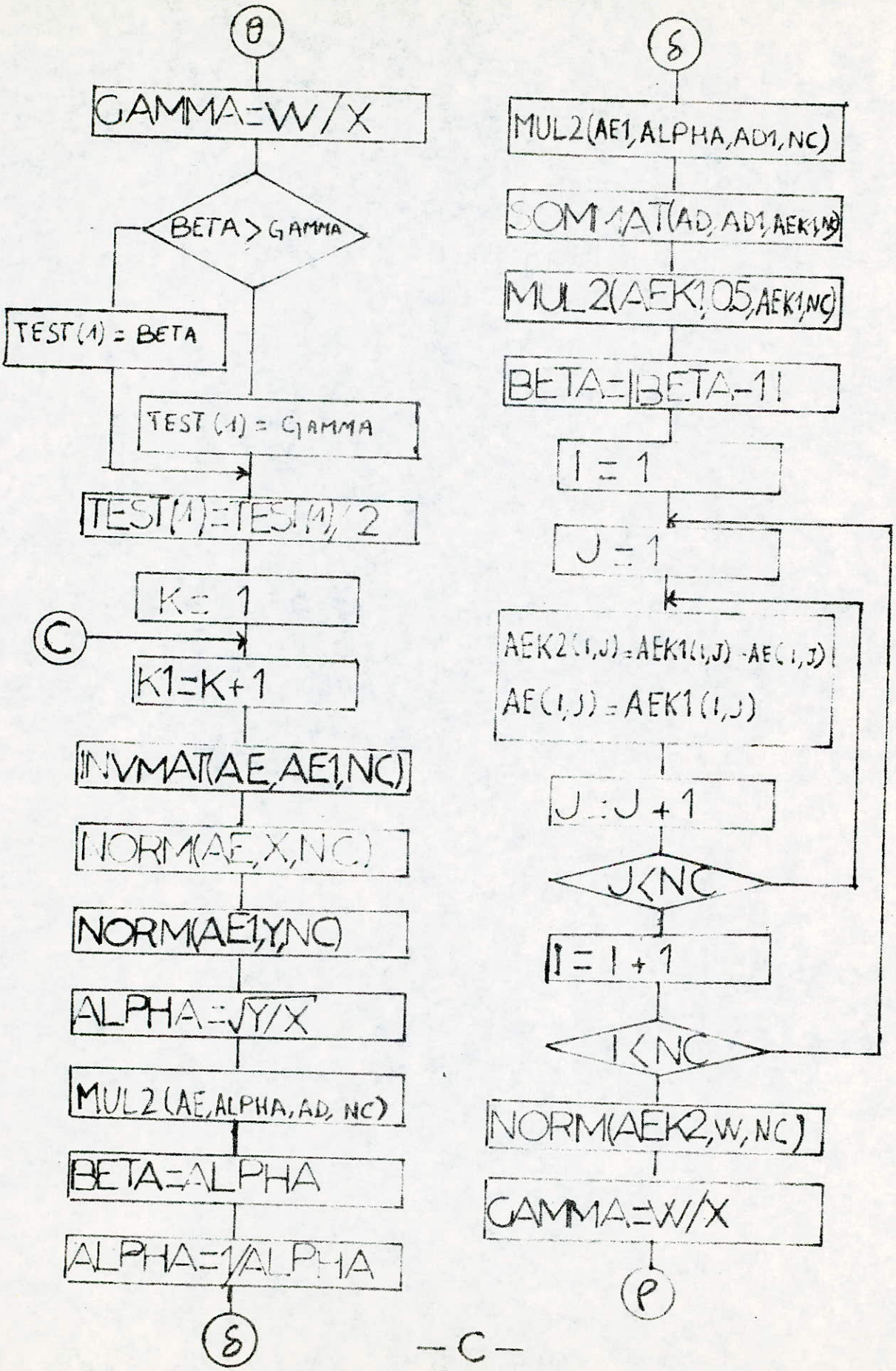


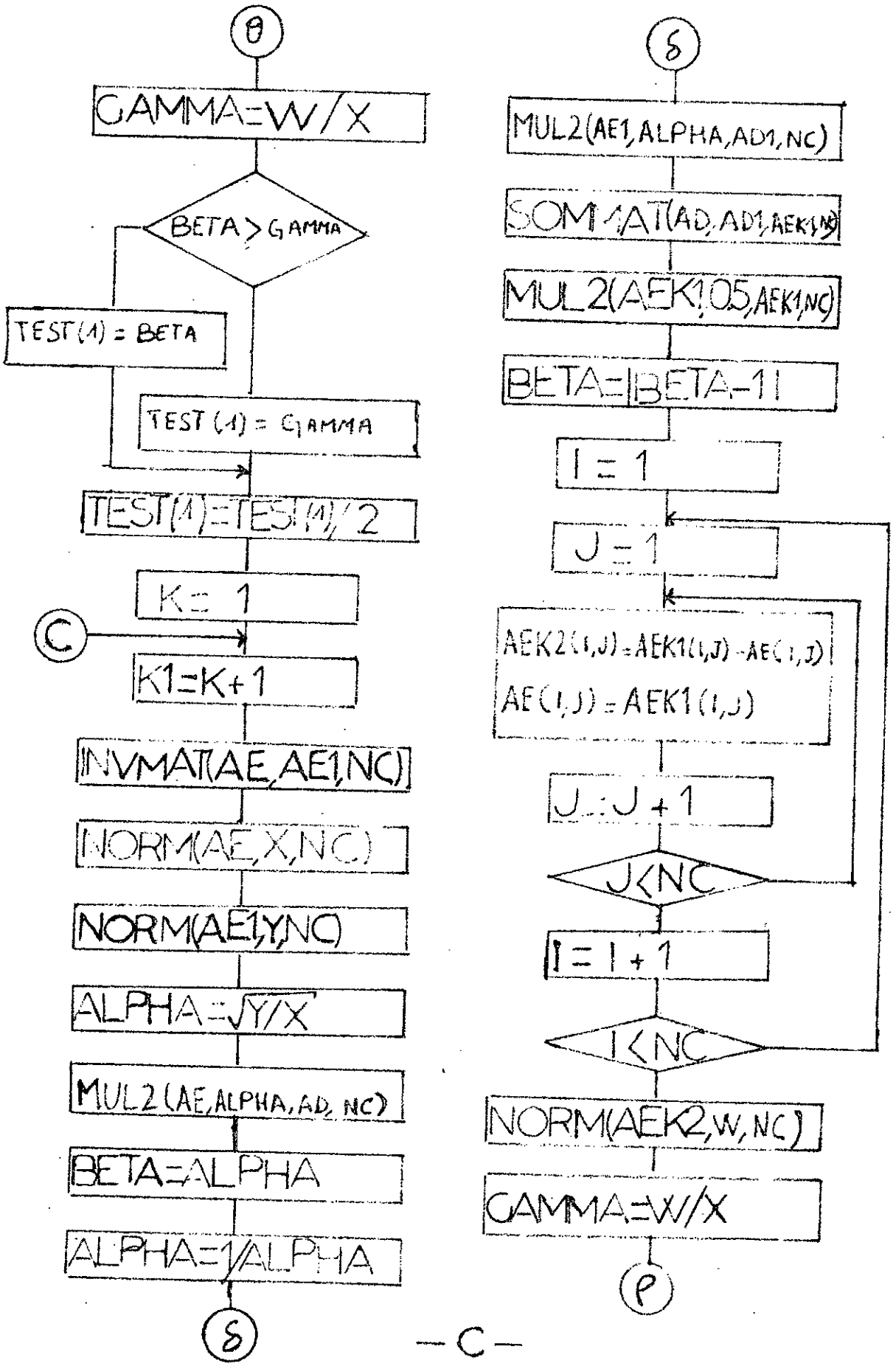
ORGANIGRAMME PRINCIPAL

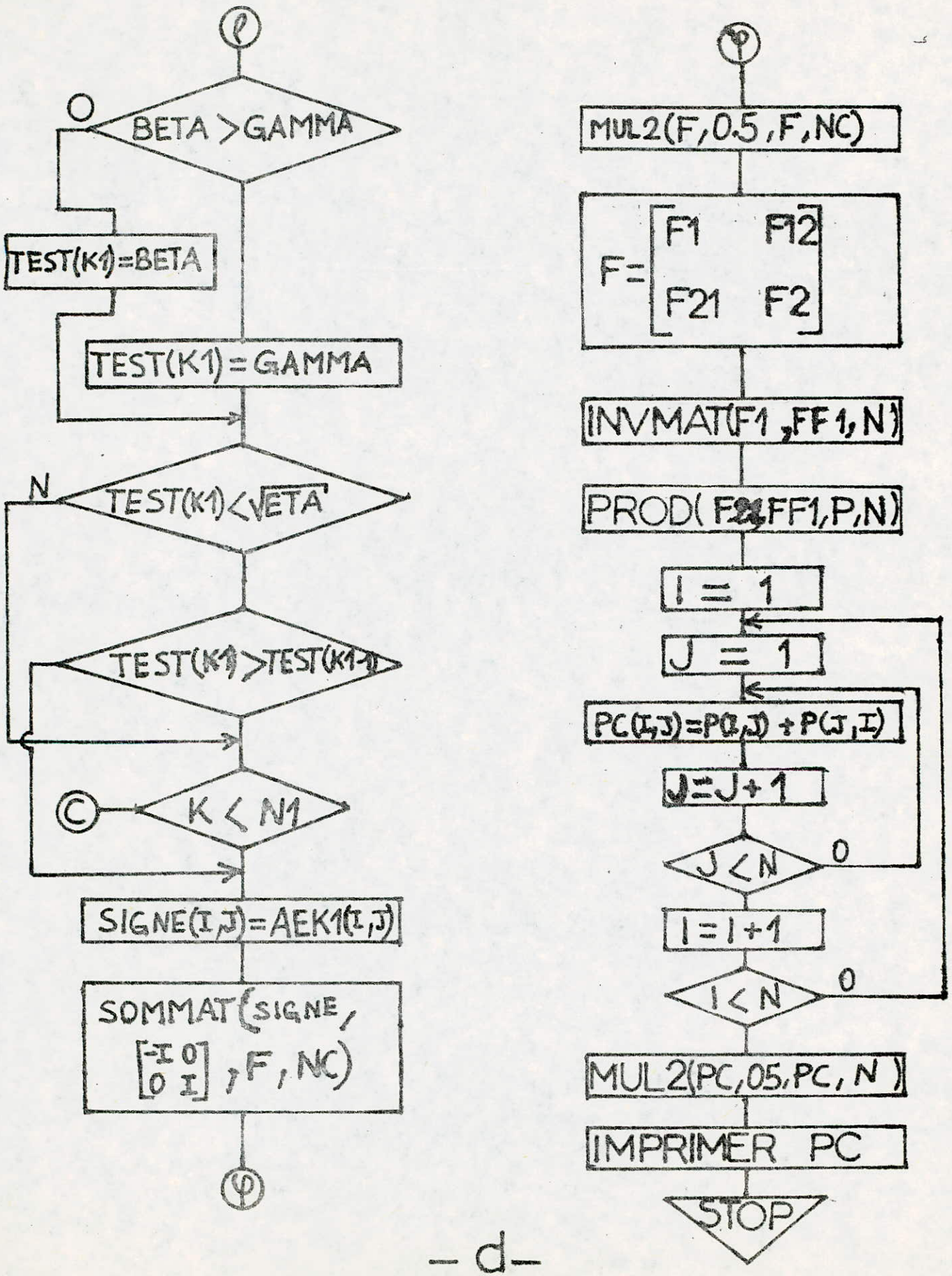


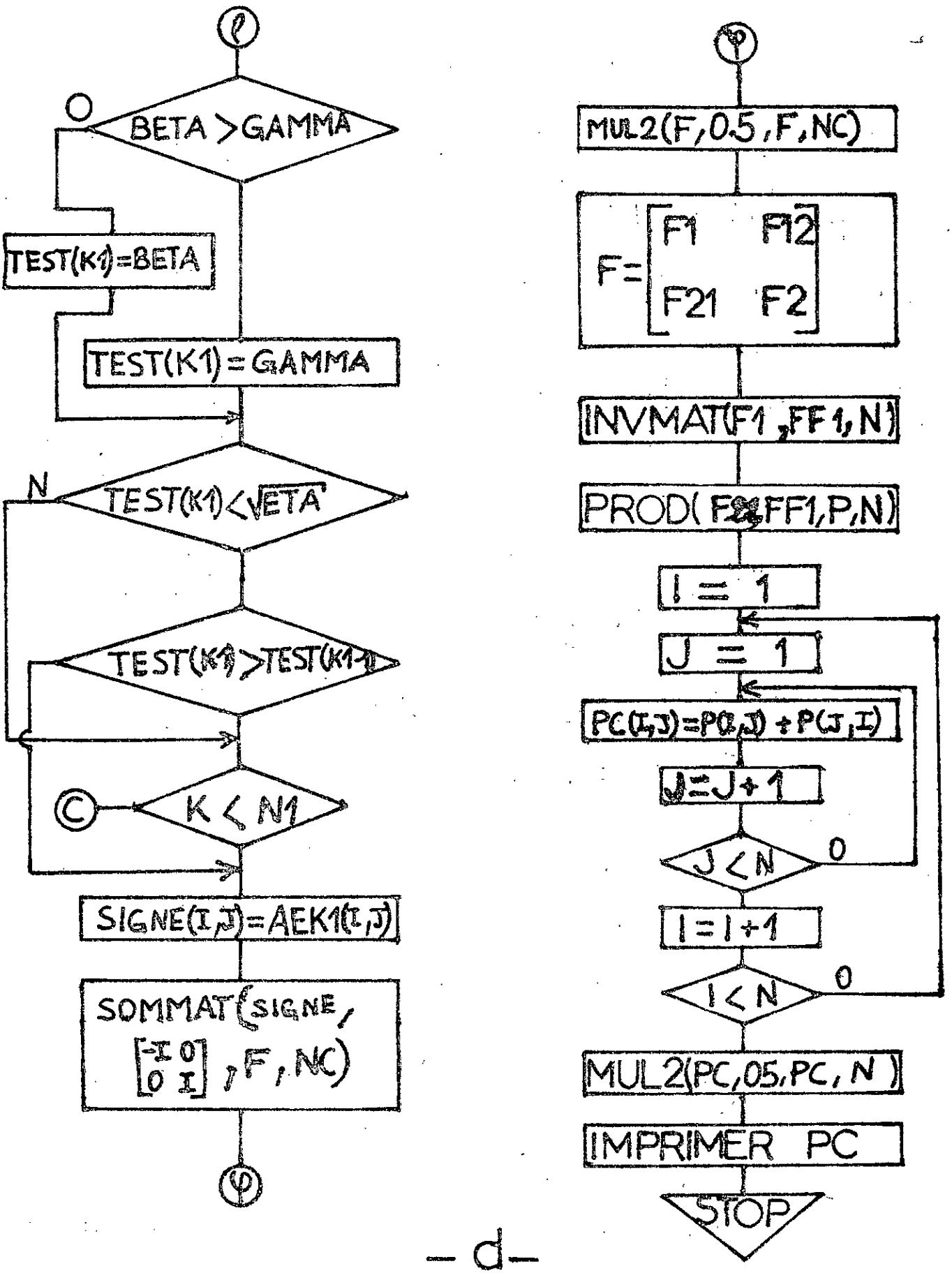

β
N1= N1/LOG2
N1=IFIX(N1)
N1=N1+QM1+PR
I = 1
J = 1
AE(I,J)=H(I,J)
J=J+1
J<NC
I=I+1
I<NC
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=√Y/X
MUL2(AE1, ALPHA,AD1, NC)
γ
γ
BETA=ALPHA
ALPHA=1/ALPHA
MUL2(AE1,ALPHA, AD1, NC)
SOMMAT(AD, AD1, AEK1, NC)
MUL2(AEK1,0.5,AEK1,NC)
BETA=|BETA-1|
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J)=AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
θ


β
N1= N1/LOG2
N1=IFIX(N1)
NM=N1+QM1+PR
I = 1
J = 1
AE(I,J)=H(I,J)
J=J+1
J<NC
I=I+1
I<NC
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=√Y/X
MUL2(AE1, ALPHA,AD1, NC)
γ
γ
BETA=ALPHA
ALPHA=1/ALPHA
MUL2(AE1,ALPHA, AD1, NC)
SOMMAT(AD,AD1,AEK1, NC)
MUL2(AEK1,0.5,AEK1,NC)
BETA=|BETA-1|
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J)=AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
θ


θ
GAMMA=W/X
BETA > GAMMA
TEST(1) = BETA
TEST(1) = GAMMA
TEST(1)=TEST(1)/2
K= 1
C
K1=K+1
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=√Y/X
MUL2(AE,ALPHA,AD,NC)
BETA=ALPHA
ALPHA=1/ALPHA
δ
δ
MUL2(AE1,ALPHA,AD1,NC)
SOMMAT(AD,AD1,AEK1,NC)
MUL2(AEK1,.05,AEK1,NC)
BETA=BETA-1
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J)=AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
GAMMA=W/X
P


θ
GAMMA=W/X
BETA > GAMMA
TEST(1) = BETA
TEST(1) = GAMMA
TEST(1)=TEST(1)/2
K= 1
C
K1=K+1
INVMAT(AE,AE1,NC)
NORM(AE,X,NC)
NORM(AE1,Y,NC)
ALPHA=√Y/X
MUL2(AE,ALPHA,AD,NC)
BETA=ALPHA
ALPHA=1/ALPHA
δ
δ
MUL2(AE1,ALPHA,AD1,NC)
SOMMAT(AD,AD1,AEK1,NC)
MUL2(AEK1,.05,AEK1,NC)
BETA=BETA-1
I = 1
J = 1
AEK2(I,J)=AEK1(I,J)-AE(I,J)
AE(I,J)=AEK1(I,J)
J=J+1
J<NC
I=I+1
I<NC
NORM(AEK2,W,NC)
GAMMA=W/X
P


BETA > GAMMA
O
TEST(K1)=BETA
TEST(K1) = GAMMA
N
TEST(K1)<√ETA
TEST(K1)>TEST(K1-1)
K < N1
SIGNE(I,J)=AEK1(I,J)
SOMMAT(SIGNE, [-I 0 / 0 I], F, NC)
MUL2(F,0.5,F,NC)
F= F1 F12 / F21 F2
INVMAT(F1, FF1, N)
PROD(F2, FF1, P, N)
I = 1
J = 1
PC(I,J)=P(I,J)+P(J,I)
J=J+1
J < N
O
I=I+1
I < N
O
MUL2(PC,05,PC,N)
IMPRIMER PC
STOP


BETA > GAMMA
TEST(K1)=BETA
TEST(K1) = GAMMA
N
TEST(K1) < BETA
TEST(K1) > TEST(K1-1)
K < N1
SIGNE(I,J)=AEK1(I,J)
SOMMAT(SIGNE, [-I 0; 0 I], F, NC)
MUL2(F,0.5,F,NC)
F= [F1 F12; F21 F2]
INVMAT(F1, FF1, N)
PROD(F2, FF1, P, N)
I = 1
J = 1
PC(I,J)=P(I,J) + P(J,I)
J=J+1
J < N
0
I=I+1
I < N
0
MUL2(PC,05,PC,N)
IMPRIMER PC
STOP

CHAPITRE III/

APPLICATIONS NUMERIQUES

CHAPITRE III/

APPLICATIONS NUMERIQUES

APPLICATIONS

1-DESCRIPTION DES PROGRAMMES

1-1-Objet des programmes

On considere ici la résolution des équations de Riccti sous la forme:

$PA+A^TP-PVP+Q=0$ cas continu

$A^TP(I+VP)^{-1}A-P+Q=0$ cas discret

Ce programme peut être étendu à la resolution des équations de Lyapounov(V=0)

$PA+A^TP+Q=0$ Cas continu

$A^TP-P+Q=0$ Cas discret

I-2-Algorithme

IL consiste à calculer la matrice signe de H,(2n.2n),associée aux equations hamiltoniènnes système. Cette matrice doit être elle-même hamiltoniènne,pour les systèmes continus,la matrice H est par nature hamiltoniènne tandis que pour le cas discret,elle doit subir deux transformations;une premiere pour la rendre symplectique,une seconde transformation bilinéaire pour la rendre hamiltonienne.

Les principales etapes du programme se résument de la manière suivante

A/ Calculs preliminaires:

- Déterminer la matrice V telle que $V=BR^{-1}B^T$
- calcul de H

pour le cas continu.

$$H=\begin{bmatrix} A & V \\ -Q & -A \end{bmatrix}$$

Pour le cas discret,on doit calculer d'abord Uet L telles que

$$U=\begin{bmatrix} I & V \\ 0 & A^T \end{bmatrix} \qquad L=\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix}$$

APPLICATIONS

1-DESCRIPTION DES PROGRAMMES

1-1-Objet des programmes

On considere ici la résolution des équations de Riccati sous la forme:

$PA+A^TP-PVP+Q=0$ cas continu

$A^TP(I+VP)^{-1}A-P+Q=0$ cas discret

Ce programme peut être étendu à la resolution des équations de Lyapounov(V=0)

$PA+A^TP+Q=0$ Cas continu

$A^TP-P+Q=0$ Cas discret

I-2-Algorithme

IL consiste à calculer la matrice signe de H,(2n.2n),associée aux equations hamiltoniènnes système. Cette matrice doit être elle-même hamiltoniènne,pour les systèmes continus,la matrice H est par nature hamiltoniènne tandis que pour le cas discret,elle doit subir deux transformations;une premiere pour la rendre symplectique,une seconde transformation bilinéaire pour la rendre hamiltonienne.

Les principales etapes du programme se résument de la manière suivante

A/ Calculs preliminaires:

- Déterminer la matrice V telle que $V=BR^{-1}B^T$
- calcul de H

pour le cas continu.

$$H=\begin{bmatrix} A & V \\ -Q & -A \end{bmatrix}$$

Pour le cas discret,on doit calculer d'abord Uet L telles que

$$U=\begin{bmatrix} I & V \\ 0 & A^T \end{bmatrix} \qquad L=\begin{bmatrix} A & 0 \\ -Q & I \end{bmatrix}$$

Poser H telle que

$$H=(L-U)^{-1}(L+U)$$

- calculer le nombre d'itérations total tel que

$$N1= N+ QM1+PR$$

avec $N= \lfloor \log_2 \max(\|H\| , \|H^{-1}\|) \rfloor$

$QMI(\eta)= 1+ \lfloor \log_2 (0,64/\eta) \rfloor$

$PR(\eta)$=nombre d'itérations supplementaires en fonction de la precision demandée

B/ <u>calculs des itérés</u>

On a posé à priori $AE=A_0^*$

puis on a calculé α comme étant $\alpha = \sqrt{\frac{\|AE^{-1}\|}{\|AE\|}}$

Puis on a calculé le premier itéré à savoir

$$AEI= \frac{1}{2} (\alpha AE + \frac{1}{\alpha} AE^{-1})$$

Apres cela on réalise la boucle suivante:

I- Calculer

$$\alpha_{k-1} = \sqrt{\frac{\|AE_{k-1}^{-1}\|}{\|AE_{k-1}\|}}$$

$$AE_k = \frac{1}{2}(\alpha_{k-1} . AE_{k-1} + \frac{1}{\alpha_{k-1}} . AE_{k-1}^{-1})$$

si $TEST(k) < \sqrt{\eta} = ETA1$ on passe à l'étape 3

2- On passe à l'étape 4

3- Si $test(k) > (1/2).test(k-1)$ on passe à l'étape 5

4- Si $k < N+1$ on retourne vers 1

5- La matrice signe est donnée par

$$SIGNE (I,J) = AE_k(I,J)$$

Poser H telle que

$$H=(L-U)^{-1}(L+U)$$

- calculer le nombre d'itérations total tel que

$$N1= N+ QM1+PR$$

avec $N= \lfloor \log_2 \max(\|H\| , \|H^{-1}\|) \rfloor$

$QMI(\eta)= 1+ \lfloor \log_2 (0,64/\eta) \rfloor$

$PR(\eta)$=nombre d'itérations supplementaires en fonction de la precision demandée

B/ <u>calculs des itérés</u>

On a posé à priori $AE=A_0^*$

puis on a calculé α comme étant $\alpha = \sqrt{\frac{\|AE^{-1}\|}{\|AE\|}}$

Puis on a calculé le premier itéré à savoir

$$AEI= \frac{1}{2} (\alpha AE + \frac{1}{\alpha} AE^{-1})$$

Apres cela on réalise la boucle suivante:

I- Calculer

$$\alpha_{k-1} = \sqrt{\frac{\|AE_{k-1}^{-1}\|}{\|AE_{k-1}\|}}$$

$$AE_k = \frac{1}{2}(\alpha_{k-1} . AE_{k-1} + \frac{1}{\alpha_{k-1}} . AE_{k-1}^{-1})$$

si $TEST(k) < \sqrt{\eta} = ETA1$ on passe à l'étape 3

2- On passe à l'étape 4

3- Si $test(k) > (1/2).test(k-1)$ on passe à l'étape 5

4- Si $k < N+1$ on retourne vers 1

5- La matrice signe est donnée par

$$SIGNE\ (I,J) = AE_k(I,J)$$

C/Calcul de la matrice P: solution de l'équation de Riccati

1-cas continu:

On calcule F comme étant

$$F = \frac{1}{2}(I + SIGNE) = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}$$

La matrice P est alors donnée par

$$P = -F_{12}^{-1}.F_1$$

2-cas discret

La matrice F est calculée comme suit

$$F = SIGNE + \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}$$

d'où

$$P = F_{21}.F_1^{-1}$$

1-3-Définition des arguments

Soit le système

$$\dot{X} = A.X + B.U$$

$$Y = C.X$$

et le coût associé

$$J = \int_0^{\infty} (X^T QX + U^T RU)dt$$

On définit alors les arguments suivants:

A: matrice réelle(n.n)

B: matrice réelle(n.n)

Q: matrice réelle(n.n)

R: matrice réelle(n.n)

N: entier égal à n(dimension de A,B,Q,R)

NC:entier égal à 2n(dimension de F)

C/Calcul de la matrice P: solution de l'équation de Riccati

1-cas continu:

On calcule F comme étant

$$F = \frac{1}{2}(I + SIGNE) = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}$$

La matrice P est alors donnée par

$$P = -F_{12}^{-1}.F_1$$

2-cas discret

La matrice F est calculée comme suit

$$F = SIGNE + \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} F_1 & F_{12} \\ F_{21} & F_2 \end{bmatrix}$$

d'où

$$P = F_{21}.F_1^{-1}$$

1-3-Définition des arguments

Soit le système

$$\dot{X} = A.X + B.U$$

$$Y = C.X$$

et le coût associé

$$J = \int_0^{\infty} (X^T QX + U^T RU)dt$$

On définit alors les arguments suivants:

A: matrice réelle(n.n)

B: matrice réelle(n.n)

Q: matrice réelle(n.n)

R: matrice réelle(n.n)

N: entier égal à n(dimension de A,B,Q,R)

NC:entier égal à 2n(dimension de F)

η: précision machine(simple précision $\eta = 10^{-16}$)

ETA1: $\sqrt{\eta}$

PR: nombre d'itérations supplémentaires pour obtenir une précision donnée(on a pris PR=5)

QM1: nombre d'itérations relatif à la convergence de l'argument θ_k. Pour le cas réel QM1=0

N1: nombre d'itérations total

TEST: variable logique

SIGNE: matrice signe de H (2n.2n)

P: solution de l'équation de Riccati

PC: matrice P corrigée

1-4-Structure du programme

Le programme fait appel à 5 sous programmes et utilise les fonctions mathématiques et utilitaires suivantes:

SQRT(X): racine carrée de X

ALOG(X): logarithme népérien de X

ABS(X): valeur absolue de X

IFIX : conversion réel- entier

sous programmes

PROD(M1,M2,M3,N): fait le produit de la matrice M1 parM2 toute deux de dimension N et met le resultat dans M3

M1.M2=M3

MUL2(M, ξ ,MP,N): fait le produit d'une matrice(N.N) par un scalaire ξ et met le résultat dans MP

ξ.M=MP

SOMMAT(D,E,F,N): fait la somme de deux matrice DetE de dimension (N.N) et met le resultat dans F: (D+E)=F

η: précision machine(simple précision $\eta = 10^{-16}$)

ETA1: $\sqrt{\eta}$

PR: nombre d'itérations supplémentaires pour obtenir une précision donnée(on a pris PR=5)

QM1: nombre d'itérations relatif à la convergence de l'argument θ_k. Pour le cas réel QM1=0

N1: nombre d'itérations total

TEST: variable logique

SIGNE: matrice signe de H (2n.2n)

P: solution de l'équation de Riccati

PC: matrice P corrigée

1-4-Structure du programme

Le programme fait appel à 5 sous programmes et utilise les fonctions mathématiques et utilitaires suivantes:

SQRT(X): racine carrée de X

ALOG(X): logarithme népérien de X

ABS(X): valeur absolue de X

IFIX : conversion réel- entier

sous programmes

PROD(M1,M2,M3,N): fait le produit de la matrice M1 parM2 toute deux de dimension N et met le resultat dans M3

M1.M2=M3

MUL2(M,E,MP,N): fait le produit d'une matrice(N.N) par un scalaire E et met le résultat dans MP

E.M=MP

SOMMAT(D,E,F,N): fait la somme de deux matrice DetE de dimension (N,N) et met le resultat dans F: (D+E)=F

NORM(A,X,N): calcule la norme de deux façons et choisit la plus petite

$$X = \min(\|A\|_1, \|A\|_\infty)$$

MRINV(A,B,KOD,DET,EPS,IL,IC): inverse la matrice A et la met dans B

$$B = A^{-1}$$

(on a fait appèl à ce sous programme dirèctement

de la bibliothèque du M I T R A 125)

2-APPLICATIONS NUMERIQUES

Notre but dans ce chapitre est d'illustrer notre étude avec des exemples précis tout en méttant en évidence quelques caracteristiques de nos programmes

2-1-RESOLUTION DE L'EQUATION DE RICCATI

2-1-1 Cas continu

Dans ce paragraphe nous allons considérer deux cas de variations du paramètre N1 (nombre d'itérations total) par rapport:

- aux pôles
- à la dimension de notre système

a/Variation de N1 par rapport aux pôles

Pour cela nous allons considérer deux systèmes de même dimension l'un stable l'autre instable.

- Soit un système dynamique continu dont la matrice de gain en boucle ouverte est:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

et le coût quadratique: $J = \int_0^\infty (x^T Q x + u^T R u) dt$

avec $Q = \begin{bmatrix} 4 & 0 \\ 0 & L \end{bmatrix}$ et $V = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$

NORM(A,X,N): calcule la norme de deux façons et choisit la plus petite

$$X = \min(\|A\|_1, \|A\|_\infty)$$

MRINV(A,B,KCD,DET,EPS,IL,IC): inverse la matrice A et la met dans B

$$B = A^{-1}$$

(on a fait appèl à ce sous programme directement

de la bibliothèque du M I T R A 125)

2-APPLICATIONS NUMERIQUES

Notre but dans ce chapitre est d'illustrer notre étude avec des exemples précis tout en méttant en évidence quelques caracteristiques de nos programmes

2-1-RESOLUTION DE L'EQUATION DE RICCATI

2-1-1 Cas continu

Dans ce paragraphe nous allons considérer deux cas de variations du paramètre N1 (nombre d'itérations total) par rapport:

- aux pôles
- à la dimension de notre système

a/Variation de N1 par rapport aux pôles

Pour cela nous allons considérer deux systèmes de même dimension l'un stable l'autre instable.

- Soit un système dynamique continu dont la matrice de main en boucle ouverte est:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

et le coût quadratique: $J = \int_0^\infty (x^T Q x + u^T R u) dt$

avec $Q = \begin{bmatrix} 4 & 0 \\ 0 & L \end{bmatrix}$ et $V = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}$

L'équation de Riccati à résoudre sera donc:

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} P + P \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} + P \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} P + \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

résultats:

-nombre d'itérations N1 = 7

-matrice P:

$$P = \begin{bmatrix} 5,999988 & 1,999993 \\ 2,0 & 4,99994 \end{bmatrix}$$

la disymétrie est de l'ordre de 7.10^{-6}

Après correction la matrice P devient:

$$P = \begin{bmatrix} 5,999988 & 1,99996 \\ 1,99996 & 4,99994 \end{bmatrix} \simeq \begin{bmatrix} 6 & 2 \\ 2 & 5 \end{bmatrix}$$

Le temps d'éxécution était de 3" 1

- le deuxième système est tel que le gain en boucle ouverte soit:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}$$

Q et V sont les mêmes que pour le cas précédent

l'équation de Riccati à résoudre serait donc

$$\begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} P + P \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} + P \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} P + \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

- nombre d'itérations: N1 = 7

-matrice P(solution):

$$P = \begin{bmatrix} 5,999986 & 1,999987 \\ 2,000002 & 0,999999 \end{bmatrix}$$

L'équation de Riccati à résoudre sera donc:

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} P + P \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} + P \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} P + \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

résultats:

-nombre d'itérations N1 = 7

-matrice P:

$$P = \begin{bmatrix} 5,999988 & 1,999993 \\ 2,0 & 4,99994 \end{bmatrix}$$

la disymétrie est de l'ordre de 7.10^{-6}

Après correction la matrice P devient:

$$P = \begin{bmatrix} 5,999988 & 1,99996 \\ 1,99996 & 4,99994 \end{bmatrix} \simeq \begin{bmatrix} 6 & 2 \\ 2 & 5 \end{bmatrix}$$

Le temps d'éxécution était de 3" 1

- le deuxième système est tel que le gain en boucle ouverte soit:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}$$

Q et V sont les mêmes que pour le cas précédent

l'équation de Riccati à résoudre serait donc

$$\begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} P + P \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} + P \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} P + \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

- nombre d'itérations: N1 = 7

-matrice P(solution):

$$P = \begin{bmatrix} 5,999986 & 1,999987 \\ 2,000002 & 0,999999 \end{bmatrix}$$

-dissymétrie est d'environ 10^{-5}

-après correction la matrice P devient

$$P = \begin{bmatrix} 5,999986 & 1,999995 \\ 1,999995 & 0,999999 \end{bmatrix} \simeq \begin{bmatrix} 6 & 2 \\ 2 & 1 \end{bmatrix}$$

- temps d'éxécution 2" 2

b/ Variation de N1 par rapport à la dimension

Soit le système dynamique continu dont la matrice gain en boucle ouverte est:

$$A = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & -2 \\ 0 & 1 & -1 \end{bmatrix}$$

le coût quaratique J est tel que

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad \text{et} \quad V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

L'équation de Riccati à résoudre serait donc:

$$\begin{bmatrix} -1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -1 \end{bmatrix} P + P \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & -2 \\ 0 & 1 & -1 \end{bmatrix} + P \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} P + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} = 0$$

Résultats

-nombre d'itérations: N1 = 7

-matrice P(solution):

$$\begin{bmatrix} 0,5944349 & -0,3234051 & 0,3047468 \\ -0,3234048 & 1,2125367 & -0,2125883 \\ 0,3047490 & -0,2126026 & 1,8561363 \end{bmatrix}$$

-dissymétrie est d'environ 10^{-5}

-après correction la matrice P devient

$$P = \begin{bmatrix} 5,999986 & 1,999995 \\ 1,999995 & 0,999999 \end{bmatrix} \simeq \begin{bmatrix} 6 & 2 \\ 2 & 1 \end{bmatrix}$$

- temps d'éxécution 2" 2

b/ Variation de N1 par rapport à la dimension

Soit le système dynamique continu dont la matrice main en boucle ouverte est:

$$A = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & -2 \\ 0 & 1 & -1 \end{bmatrix}$$

le coût quaratique J est tel que

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad \text{et} \quad V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

L'équation de Riccati à résoudre serait donc:

$$\begin{bmatrix} -1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -1 \end{bmatrix} P + P \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & -2 \\ 0 & 1 & -1 \end{bmatrix} + P \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} P + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} = 0$$

Résultats

-nombre d'itérations: N1 = 7

-matrice P(solution):

$$\begin{bmatrix} 0,5944349 & -0,3234051 & 0,3047468 \\ -0,3234048 & 1,2125367 & -0,2125883 \\ 0,3047490 & -0,2126026 & 1,8561363 \end{bmatrix}$$

- la dissymétrie est de l'ordre de 10^{-5} à 10^{-6}

-la matrice P corrigée est

$$P = \begin{bmatrix} 0,5944347 & -0,3234050 & 0,3047479 \\ -0,3234050 & 1,2125367 & -0,2125955 \\ 0,3047479 & -0,2125955 & 1,8561363 \end{bmatrix}$$

- le temps d'éxécution est de 3" 8

Conclusions

D'après les résultats obtenus, on vient de confirmer quelques caractéristiques de cette méthode (que l'on avait citées dans la partie théorique) à savoir que:

- l'on peut trouver la commande optimale d'un système qu'il soit stable ou instable, de plus l'instabilité n'influe en rien sur le nombre d'itérations total, c'est à dire, sur la rapidité de la convergence de notre algorithme.
- la dimension de notre système n'influe pas ou du moins très peu sur le nombre d'itérations que doit éffectuer notre algorithme pour converger.

Remarque

En ce qui concerne la variation du nombre d'itérations par rapport au spectre de la matrice, cela sera mis en évidence dans la résolution de l'équation de Lyapunov et les conclusions s'y rapportant s'appliquent intégralement au cas de la résolution de l'équation de Riccati.

2-1-2/ Cas discret

Pour le cas discret les conclusions sont identiques à celles du cas continu vu que l'algorithme est presque le même.

Néanmoins, on a pris un exemple particulier où l'on doit faire d'abord des calculs préliminaires (calcul de V et de Q),c'est à dire en partant uniquement du système d'équations d'état.

- la dissymétrie est de l'ordre de 10^{-5} à 10^{-6}

-la matrice P corrigée est

$$P = \begin{bmatrix} 0,5944347 & -0,3234050 & 0,3047479 \\ -0,3234050 & 1,2125367 & -0,2125955 \\ 0,3047479 & -0,2125955 & 1,8561363 \end{bmatrix}$$

- le temps d'éxécution est de 3" 8

Conclusions

D'après les résultats obtenus, on vient de confirmer quelques caractéristiques de cette méthode (que l'on avait citées dans la partie théorique) à savoir que:

- l'on peut trouver la commande optimale d'un système qu'il soit stable ou instable, de plus l'instabilité n'influe en rien sur le nombre d'itérations total, c'est à dire, sur la rapidité de la convergence de notre algorithme.
- la dimension de notre système n'influe pas ou du moins très peu sur le nombre d'itérations que doit éffectuer notre algorithme pour converger.

Remarque

En ce qui concerne la variation du nombre d'itérations par rapport au spectre de la matrice, cela sera mis en évidence dans la résolution de l'équation de Lyapunov et les conclusions s'y rapportant s'appliquent intégralement au cas de la résolution de l'équation de Riccati.

2-1-2/ Cas discret

Pour le cas discret les conclusions sont identiques à celles du cas continu vu que l'algorithme est presque le même.

Néanmoins, on a pris un exemple particulier où l'on doit faire d'abord des calculs préliminaires (calcul de V et de Q),c'est à dire en partant uniquement du système d'équations d'état.

Soit donc un système discret de dimension 5 dont l'équation d'état est

$$X_{K+1} = A.X_K + B.U_K$$

et le coût

$$J = \sum_{K=0}^{\infty} \left(X_K^T.Q.X_K + U_K^T.R.U_K \right)$$

avec

$$A = \begin{bmatrix} 0,75 & 0,009 & 0 & 0 & 0 \\ -1,74 & 0,91 & 0 & 0 & 0 \\ -0,3 & -0,0015 & 0,95 & 0 & 0 \\ 0 & 0 & 0 & 0,55 & 0 \\ -0,15 & -0,008 & 0 & 0 & 0,905 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} 0 & 0 & 24,64 & 0 & 0 \\ 0 & 0 & 0 & 0,835 & 0 \\ 0 & 0 & 0 & 0 & 1,83 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

les matrices Q et V sont calculées comme suit

$$Q = H^T H \qquad V = BR^{-1}B^T$$

L'équation de Riccati étant:

$$A^T P(I + VP)^{-1} A - P + Q = 0$$

on a aboutit aux résultats suivants:

- matrice P (solution):

$$P = \begin{bmatrix} 72,30322 & 2,65930 & -200,53321 & -0.37338 & -9,98702 \\ 2,65924 & 1,14270 & -6,12289 & -0,31483 & -2,33572 \\ -200,54036 & -6,12382 & 1212,88950 & 1,29670 & 13,77488 \\ -0,37376 & -0,31487 & -1,29774 & -0,93206 & 0,46769 \\ -9,98926 & -2,33575 & 13,77612 & -0,46759 & 11,74435 \end{bmatrix}$$

Soit donc un système discret de dimension 5 dont l'équation d'état est

$$X_{K+1} = A.X_K + B.U_K$$

et le coût

$$J = \sum_{K=0}^{\infty} \left(X_K^T.Q.X_K + U_K^T.R.U_K \right)$$

avec

$$A = \begin{bmatrix} 0,75 & 0,009 & 0 & 0 & 0 \\ -1,74 & 0,91 & 0 & 0 & 0 \\ -0,3 & -0,0015 & 0,95 & 0 & 0 \\ 0 & 0 & 0 & 0,55 & 0 \\ -0,15 & -0,008 & 0 & 0 & 0,905 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} 0 & 0 & 24,64 & 0 & 0 \\ 0 & 0 & 0 & 0,835 & 0 \\ 0 & 0 & 0 & 0 & 1,83 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

les matrices Q et V sont calculées comme suit

$$Q = H^T H \qquad V = BR^{-1}B^T$$

L'équation de Riccati étant:

$$A^T P(I + VP)^{-1} A - P + Q = 0$$

on a aboutit aux résultats suivants:

- matrice P (solution):

$$P = \begin{bmatrix} 72,30322 & 2,65930 & -200,53321 & -0.37338 & -9,98702 \\ 2,65924 & 1,14270 & -6,12289 & -0,31483 & -2,33572 \\ -200,54036 & -6,12382 & 1212,88950 & 1,29670 & 13,77488 \\ -0,37376 & -0,31487 & -1,29774 & -0,93206 & 0,46769 \\ -9,98926 & -2,33575 & 13,77612 & -0,46759 & 11,74435 \end{bmatrix}$$

- nombre d'itérations N1 = 19
- la dissymétrie est de l'ordre de 10^{-3} à 10^{-5}
- après correction la matrice P devient

$$P = \begin{bmatrix} 72,30322 & 2,65927 & -200,53685 & -0,37357 & -9,98814 \\ 2,65927 & 1,14270 & -6,12336 & -0,31485 & -2,33573 \\ -200,53685 & -6,12336 & 1212,88950 & 1,29722 & 13,77550 \\ -0,37357 & -0,31485 & 1,29722 & 0,93206 & 0,46764 \\ -9,98814 & -2,33573 & 13,77550 & 0,46764 & 11,74434 \end{bmatrix}$$

- le temps d'éxécution était de 17" 6

- nombre d'itérations N1 = 19
- la dissymétrie est de l'ordre de 10^{-3} à 10^{-5}
- après correction la matrice P devient

$$P = \begin{bmatrix} 72,30322 & 2,65927 & -200,53685 & -0,37357 & -9,98814 \\ 2,65927 & 1,14270 & -6,12336 & -0,31485 & -2,33573 \\ -200,53685 & -6,12336 & 1212,88950 & 1,29722 & 13,77550 \\ -0,37357 & -0,31485 & 1,29722 & 0,93206 & 0,46764 \\ -9,98814 & -2,33573 & 13,77550 & 0,46764 & 11,74434 \end{bmatrix}$$

- le temps d'éxécution était de 17" 6

3- APPLICATION A LA STABILITE DES SYSTEMES

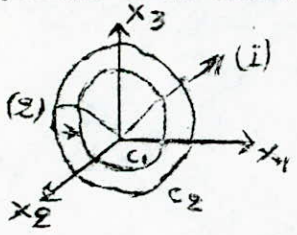
3-1- Rappel sur la deuxième methode de Lyapunov

Nous allons étudier le cas d'un système à trois variables dans la representation d'état.Ce systeme étant linéaire et son point d'équilibre se trouve à l'origine.

$$\dot{x}_1 = X_1(x_1,x_2,x_3)$$
$$\dot{x}_2 = X_2(x_1,x_2,x_3)$$
$$\dot{x}_3 = X_3(x_1,x_2,x_3)$$

Supposons que l'on puisse mettre en évidence dans l'éspace (x_1,x_2,x_3) une famille de surfaces fermées entourant l'origine,telles que par chaque point de l'éspace passe une surface unique de la famille.Si dans toute une région entourant l'origine le comportement du système est tel que la vitesse du point(x_1,x_2,x_3)doit toujours être dirigée vers l'intérieur de la surface de la famille passant par ce point;le point representatif finira par arriver à l'origine et par conséquent le système sera stable.

Inversement on conçoit que si la vitesse est dirigée vers l'éxterieur le système sera instable.



(I) instable
(2) stable

V(x)=Ste

Les équations des surfaces fermées sont du type

$$V(x_1,x_2,x_3) = \text{Cste}$$

D'où plus précisement pour l'application de la méthode directe de Lyapunov,on considère des fonctions auxilliaires $V(x_1,.....,x_n)$

3- APPLICATION A LA STABILITE DES SYSTEMES

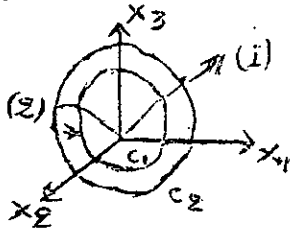
3-1- Rappel sur la deuxième methode de Lyapunov

Nous allons étudier le cas d'un système à trois variables dans la representation d'état.Ce systeme étant linéaire et son point d'équilibre se trouve à l'origine.

$$\dot{x}_1 = X_1(x_1,x_2,x_3)$$
$$\dot{x}_2 = X_2(x_1,x_2,x_3)$$
$$\dot{x}_3 = X_3(x_1,x_2,x_3)$$

Supposons que l'on puisse mettre en évidence dans l'éspace (x_1,x_2,x_3) une famille de surfaces fermées entourant l'origine,telles que par chaque point de l'éspace passe une surface unique de la famille.Si dans toute une région entourant l'origine le comportement du système est tel que la vitesse du point(x_1,x_2,x_3)doit toujours être dirigée vers l'intérieur de la surface de la famille passant par ce point;le point representatif finira par arriver à l'origine et par conséquent le système sera stable.

Inversement on conçoit que si la vitesse est dirigée vers l'éxterieur le système sera instable.



(I) instable
(2) stable

V(x)=Ste

Les équations des surfaces fermées sont du type

$$V(x_1,x_2,x_3) = \text{Cste}$$

D'où plus précisement pour l'application de la méthode directe de Lyapunov,on considère des fonctions auxilliaires $V(x_1,.....,x_n)$

définies positives vérifiant les conditions suivantes:

a/V est nulle à l'origine $V(0,.....,0)=0$

b/V est infinie pour des variables infinies $V(x_1,.....,x_n) \longrightarrow \infty$

$x_i \longrightarrow \infty$

c/V reste finie lorsque les variables sont finies

Les fonctions V sont dites fonction de Lyapunov.

Nous sommes maintenant en mesure d'énnoncer les deux théorèmes fondamentaux de Lyapunov relatifs respectivement à la stabilité assymptotique et la stabilité simple.

théorème I

S'il est possible de trouver une fonction V de signe défini (dans un domaine G comprenant la position d'équilibre) et dont la derivée totale par rapport au temps dV/dt soit définie et de signe opposé dans le méme domaine, l'équilibre sera asymptotiquement stable dans ce domaine G.

théorème 2

S'il est possible de trouver une fonction V de signe defini(dans un domaine G comprenant la position d'équilibre)et dont la dérivée totale par rapport au temps dV/dt soit semi définie et de signe opposé dans le même domaine l'équilibre est stable dans ce domaine

Rappels

1-Une fonction sera dite semi definie dans tout un domaine G si elle conserve le même signe en tout point de ce domaine mais s'annule aussi en d'autres points que l'origine.

2-Une fonction sera dite indéfinie dans le domaine G si elle prend des signes opposés en différents points de ce domaine.

définies positives vérifiant les conditions suivantes:

a/V est nulle à l'origine $V(0,.....,0)=0$

b/V est infinie pour des variables infinies $V(x_1,.....,x_n) \longrightarrow \infty$

$x_i \longrightarrow \infty$

c/V reste finie lorsque les variables sont finies

Les fonctions V sont dites fonction de Lyapunov.

Nous sommes maintenant en mesure d'énnoncer les deux théorèmes fondamentaux de Lyapunov relatifs respectivement à la stabilité assymptotique et la stabilité simple.

théorème I

S'il est possible de trouver une fonction V de signe défini (dans un domaine G comprenant la position d'équilibre) et dont la derivée totale par rapport au temps dV/dt soit définie et de signe opposé dans le méme domaine, l'équilibre sera asymptotiquement stable dans ce domaine G.

théorème 2

S'il est possible de trouver une fonction V de signe defini(dans un domaine G comprenant la position d'équilibre)et dont la dérivée totale par rapport au temps dV/dt soit semi définie et de signe opposé dans le même domaine l'équilibre est stable dans ce domaine

Rappels

1-Une fonction sera dite semi definie dans tout un domaine G si elle conserve le même signe en tout point de ce domaine mais s'annule aussi en d'autres points que l'origine.

2-Une fonction sera dite indéfinie dans le domaine G si elle prend des signes opposés en différents points de ce domaine.

Remarque importante

Lorsque le système étudié est linéaire on peut montrer que les théorèmes 1 et 2 deviennent des conditions nécéssaires et suffisantes de stabilité.En effet le signe de dV/dt est lié au signe du produit de la vitèsse du point sur la trajectoire et de ∇V

3-2- Etablissement de l'équation de Lyapunov.

Soit le systéme linéaire autonome représenté à l'aide de la représentation d'état par l'équation:

$\dot{X}=A\,X$ et V(X)une forme quadratique du type $V(X)= X^T\,P\,X$

où P est une matrice symétrique,définie positive. Donc V(X) est définie positive au sens de Lyapunov.

pour que le systéme soit stable il faut et il suffit que:

$(dV/dt)<0 \Longrightarrow \dot{X}^T P X + X^T P \dot{X} = (AX)^T P X + X^T P A X.$

$= X^T A^T P X + \dot{X}^T P A X = X^T (A^T P + PA)\, X < 0$

Ce qui revient à résoudre l'équation :

$A^T P + P A = - Q$ où Q est définie positive

Conclusion

Aisi des mêmes algorithmes résolvant les équations de Riccati nous permettent de résoudre celles de Lyapunov relatives à la stabilité.En éffet l'équation de Riccati étant:

$$A^T P + P A + P D P + Q = 0$$

D'où en méttant D=0,on retrouve l'équation de Lyapunov.

3-3-Applications numériques

On a vu ultérieurement que, pratiquement le même algorithme ayant servi pour la résolution de l'équation de Riccati, peut servir à résoudre l'équation de Lyapunov (en prenant V=0).

Remarque importante

Lorsque le système étudié est linéaire on peut montrer que les théorèmes 1 et 2 deviennent des conditions nécéssaires et suffisantes de stabilité.En effet le signe de dV/dt est lié au signe du produit de la vitèsse du point sur la trajectoire et de ∇V

3-2- Etablissement de l'équation de Lyapunov.

Soit le systéme linéaire autonome représenté à l'aide de la représentation d'état par l'équation:

$\dot{X}=A\,X$ et V(X)une forme quadratique du type $V(X)= X^T\,P\,X$

où P est une matrice symétrique,définie positive. Donc V(X) est définie positive au sens de Lyapunov.

pour que le systéme soit stable il faut et il suffit que:

$$(dV/dt)<0 \Longrightarrow \dot{X}^T P X + X^T P \dot{X} = (AX)^T P X + X^T P A X.$$

$$= X^T A^T P X + \dot{X}^T P A X = X^T (A^T P + PA)\, X < 0$$

Ce qui revient à résoudre l'équation :

$A^T P + P A = - Q$ où Q est définie positive

Conclusion

Ainsi des mêmes algorithmes résolvant les équations de Riccati nous permettent de résoudre celles de Lyapunov relatives à la stabilité.En éffet l'équation de Riccati étant:

$$A^T P + P A + P D P + Q = 0$$

D'où en méttant D=0,on retrouve l'équation de Lyapunov.

3-3-Applications numériques

On a vu ultérieurement que, pratiquement le même algorithme ayant servi pour la résolution de l'équation de Riccati, peut servir à résoudre l'équation de Lyapunov (en prenant V=0).

Il nous a paru utile d'étudier dans ce cas, la variation du nombre d'itérations total en fonction du spectre.

L'équation de Lyapunov dans le cas continu étant

$$A^T P + PA + Q = 0$$

on va considérer dans ce qui suit trois cas d'applications dans lesquels la matrice A reste la même.

Quand à la matrice Q, elle est mise sous forme diagonale

$$Q = M.D.M^{-1}$$

où

M est la matrice des vecteurs propres, fixe pour les trois exemples.

D est la matrice diagonale de Q. Elle varie d'un exemple à l'autre.

$$A = \begin{bmatrix} 4 & 1 & -1 \\ 1 & 3 & 0 \\ -1 & 0 & 5 \end{bmatrix} \qquad M = \begin{bmatrix} 1 & -1 & 1 \\ 2 & 0 & 1 \\ 1 & 2 & -1 \end{bmatrix}$$

Premier cas

$$Q = M.D_1.M^{-1}$$

avec

$$D_1 = \text{diag}(-1,1 \ ; \ -1,01 \ ; \ -0,95)$$

résultats

- nombre d'itérations total N1 = 13
- la matrice P après correction est

Il nous a paru utile d'étudier dans ce cas, la variation du nombre d'itérations total en fonction du spectre.

L'équation de Lyapunov dans le cas continu étant

$$A^T P + PA + Q = 0$$

on va considérer dans ce qui suit trois cas d'applications dans lesquels la matrice A reste la même.

Quand à la matrice Q, elle est mise sous forme diagonale

$$Q = M.D.M^{-1}$$

où

M est la matrice des vecteurs propres, fixe pour les trois exemples.

D est la matrice diagonale de Q. Elle varie d'un exemple à l'autre.

$$A = \begin{bmatrix} 4 & 1 & -1 \\ 1 & 3 & 0 \\ -1 & 0 & 5 \end{bmatrix} \qquad M = \begin{bmatrix} 1 & -1 & 1 \\ 2 & 0 & 1 \\ 1 & 2 & -1 \end{bmatrix}$$

Premier cas

$$Q = M.D_1.M^{-1}$$

avec

$$D_1 = \text{diag}(-1,1 \ ; \ -1,01 \ ; \ -0,95)$$

résultats

- nombre d'itérations total N1 = 13
- la matrice P après correction est

$$P = \begin{bmatrix} -9,0765685 & 2,9470474 & -2,0365685 \\ 2,9470474 & -2,6469392 & 3,8338310 \\ -2,0365685 & 3,8338310 & -4,4555342 \end{bmatrix}$$

- le temps d'éxécution était de 4" 4

Deuxième cas

$Q = M.D_2.M^{-1}$ avec D_2= diag(-1 ; -2 ; -3)

résultats

- nombre d'itérations total N1 = 16
- la matrice P après corréction est

$$P = \begin{bmatrix} -0,8173513 & 1,4871061 & 0,6029091 \\ 1,4871061 & -0,7418242 & 1,6044044 \\ 0,6029091 & 1,6044044 & -1,5638620 \end{bmatrix}$$

- le temps d'éxécution étant 5" 6

Troisième cas

$Q = M.D_3.M^{-1}$ avec D_3 = diag(-10 ; -1 ; -1000)

résultats

- nombre d'itérations total N1 = 33
- la matrice P après corréction est

$$\begin{bmatrix} -0,3189621 & -0,0882767 & -0,2126414 \\ -0,0882767 & -0,0295951 & -0,0537453 \\ -0,2126414 & -0,0537453 & 0,0140174 \end{bmatrix}$$

- le temps d'éxécution est de 8" 3

Conclusion

On constate que, d'après ces résultats, le nombre d'itérations varie en fonction du spectre. Ceci était prévisible car à chaque itération, il fallait équilibrer la matrice H dont on calcule la matrice signe jusqu'à avoir toutes les valeurs propres égales ou très proches les unes des autres. Cet équilibrage est donc éssentiellement fonction du spectre de H. Ce qui augmente N1 lorsque les valeurs propres sont fortement distinctes.

$$P = \begin{bmatrix} -9,0765685 & 2,9470474 & -2,0365685 \\ 2,9470474 & -2,6469392 & 3,8338310 \\ -2,0365685 & 3,8338310 & -4,4555342 \end{bmatrix}$$

- le temps d'éxécution était de 4" 4

Deuxième cas

$Q = M.D_2.M^{-1}$ avec $D_2 = \text{diag}(-1 ; -2 ; -3)$

résultats

- nombre d'itérations total $N1 = 16$
- la matrice P après corréction est

$$P = \begin{bmatrix} -0,8173513 & 1,4871061 & 0,6029091 \\ 1,4871061 & -0,7418242 & 1,6044044 \\ 0,6029091 & 1,6044044 & -1,5638620 \end{bmatrix}$$

- le temps d'éxécution étant 5" 6

Troisième cas

$Q = M.D_3.M^{-1}$ avec $D_3 = \text{diag}(-10 ; -1 ; -1000)$

résultats

- nombre d'itérations total $N1 = 33$
- la matrice P après corréction est

$$\begin{bmatrix} -0,3189621 & -0,0882767 & -0,2126414 \\ -0,0882767 & -0,0295951 & -0,0537453 \\ -0,2126414 & -0,0537453 & 0,0140174 \end{bmatrix}$$

- le temps d'éxécution est de 8" 3

Conclusion

On constate que, d'après ces résultats, le nombre d'itérations varie en fonction du spectre. Ceci était prévisible car à chaque itération, il fallait équilibrer la matrice H dont on calcule la matrice signe jusqu'à avoir toutes les valeurs propres égales ou très proches les unes des autres. Cet équilibrage est donc éssentiellement fonction du spectre de H. Ce qui augmente N1 lorsque les valeurs propres sont fortement distinctes.

-oOo- C O N C L U S I O N -oOo-

On a développé dans cette étude un concept (la fonction signe de matrice), et ses applications à la commande optimale et à la stabilité des systèmes.D'où l'on a été amené à établir des algorithmes nécéssaires à la résolution des équations de RICCATI et de LYAPUNOV dans les cas continu et discret.Ces algorithmes tiénnent compte des contraintes sur le temps calcul,la précision, et les performances téchniques du calculateur MITRA 125 se trouvant au centre de calcul du C E N.

On a montré sur le plan de la convergence comment évoluaient éxactement les itérés engendrés par l'algorithme de Newton, introduit pour le calcul de la fonction signe . Les résultats obtenus ont débouché sur deux nouveaux algorithmes,l'un fini pour les matrices à spectre réel,l'autre relatif au cas général (spectre complexe),constituant ce qu'on a appelé un algorithme de Newton accéléré.

Du point de vue compléxité de mise en oeuvre,stabilité numérique,coût calcul par itération et vitesse de convergence,la méthode de la fonction signe que l'on avait développée(au niveau de ses applications) nous semble être très performante.De plus elle est plus générale.

En éffet le même code peut servir pour résoudre les équations de Riccati relatives aux systèmes continus et discrets.Seule la détérmination de l'hamiltonien et le calcul de P changent.

-oOo- C O N C L U S I O N -oOo-

On a développé dans cette étude un concept (la fonction signe de matrice), et ses applications à la commande optimale et à la stabilité des systèmes.D'où l'on a été amené à établir des algorithmes nécéssaires à la résolution des équations de RICCATI et de LYAPUNOV dans les cas continu et discret.Ces algorithmes tiénnent compte des contraintes sur le temps calcul,la précision, et les performances téchniques du calculateur MITRA 125 se trouvant au centre de calcul du C E N.

On a montré sur le plan de la convergence comment évoluaient éxactement les itérés engendrés par l'algorithme de Newton, introduit pour le calcul de la fonction signe . Les résultats obtenus ont débouché sur deux nouveaux algorithmes,l'un fini pour les matrices à spectre réel,l'autre relatif au cas général (spectre complexe),constituant ce qu'on a appelé un algorithme de Newton accéléré.

Du point de vue complexité de mise en oeuvre,stabilité numérique,coût calcul par itération et vitesse de convergence,la méthode de la fonction signe que l'on avait développée(au niveau de ses applications) nous semble être très performante.De plus elle est plus générale.

En éffet le même code peut servir pour résoudre les équations de Riccati relatives aux systèmes continus et discrets.Seule la détérmination de l'hamiltonien et le calcul de P changent.

Enfin si l'on pose V=0, le même algorithme permet de résoudre les équations de Lyapunov.

Notre expérience dans le domaine numérique montre que les résultats obtenus(sauf mauvais conditionnement) comparés à ceux de Barraud et Beavers, sont accéptables.

Du point de vue symétrie, les résultats obtenus(avant corréction) ont une dissymétrie de l'ordre de 10^{-5} à 10^{-6}.

Quand à la convergence, elle est du même ordre que la méthode SR2 seulement notre algorithme ne demande aucun calcul préliminaire, d'où une facilité de mise en oeuvre,une diminution du coût calcul et de l'occupation mémoire.

Comme continuité à notre travail nous pensons que les remarques suivantes sont utiles:

- on pourrait développer d'autres algorithmes de la forme

$$A_{K+1} = \alpha A_K + \beta A_K^{-1}$$

où il s'agirait de calculer α et β de telle sorte que l'algorithme converge plus rapidement que celui dit de Newton accéléré(où $\alpha\beta = \frac{1}{4}$)

- suivant la nature des problèmes de côntrole auxquels on s'interèsse,on pourrait penser que si des calculs relativement longs du type conditionnement de la matrice H étaient faits hors ligne, on pourrait améliorer la rapidité de convergence des algorithmes pour une éventuelle application en ligne.

Enfin si l'on pose V=0, le même algorithme permet de résoudre les équations de Lyapunov.

Notre expérience dans le domaine numérique montre que les résultats obtenus(sauf mauvais conditionnement) comparés à ceux de Barraud et Beavers, sont accéptables.

Du point de vue symétrie, les résultats obtenus(avant corréction) ont une dissymétrie de l'ordre de 10^{-5} à 10^{-6}.

Quand à la convergence, elle est du même ordre que la méthode SR2 seulement notre algorithme ne demande aucun calcul préliminaire, d'où une facilité de mise en oeuvre,une diminution du coût calcul et de l'occupation mémoire.

Comme continuité à notre travail nous pensons que les remarques suivantes sont utiles:

- on pourrait développer d'autres algorithmes de la forme

$$A_{K+1} = \alpha A_K + \beta A_K^{-1}$$

où il s'agirait de calculer α et β de telle sorte que l'algorithme converge plus rapidement que celui dit de Newton accéléré(où $\alpha\beta = \frac{1}{4}$)

- suivant la nature des problèmes de côntrole auxquels on s'interèsse,on pourrait penser que si des calculs relativement longs du type conditionnement de la matrice H étaient faits hors ligne, on pourrait améliorer la rapidité de convergence des algorithmes pour une éventuelle application en ligne.

[illegible]

ANNEXE I

RAPPELS MATHEMATIQUES

1-Valeurs propres d'une matrice

On appelle valeur propre d'une matrice toute racine λ de son équation caractéristique

$$\det(\lambda I - A) = 0$$

Si A(n.n), on aura n valeurs propres réelles ou complexes, distinctes ou multiples.

2-Vecteurs propres d'une matrice

On appelle vecteur propre m associé à une valeur propre, tout vecteur solution de l'équation

$$[A - \lambda I]\, m = 0$$

donc à une valeur propre simple correspond un vecteur propre unique (à un coefficient de proportionalité près). A une valeur propre multiple d'ordre μ, peut correspondre 1 à μ vecteurs propres selon le nombre de solutions indépendantes du système d'équations.

3-Forme diagonale d'une matrice

Si toutes les racines de l'équation caractéristique sont distinctes, il est possible de mettre la matrice A sous forme diagonale

$$\tilde{A} = \text{diag}\,[\lambda_1, \lambda_2, \ldots\ldots \lambda_n] \qquad \lambda i \neq \lambda j \text{ si } i \neq j$$

La matrice de transformation M est la matrice des vecteurs propres

$$M = [m_1 m_2 \ldots\ldots m_n] \qquad \text{avec} \quad A m_1 = \lambda_1 m_1$$

$$\tilde{A} = M^{-1} A M \quad \Longleftrightarrow \quad MA = AM \quad \Longleftrightarrow \quad M A M^{-1} = A$$

ANNEXE I

RAPPELS MATHEMATIQUES

1-Valeurs propres d'une matrice

On appelle valeur propre d'une matrice toute racine λ de son équation caractéristique

$$\det(\lambda I - A) = 0$$

Si A(n.n), on aura n valeurs propres réelles ou complexes, distinctes ou multiples.

2-Vecteurs propres d'une matrice

On appelle vecteur propre r associé à une valeur propre, tout vecteur solution de l'équation

$$[A - \lambda I]\, m = 0$$

donc à une valeur propre simple correspond un vecteur propre unique (à un coefficient de proportionalité près). A une valeur propre multiple d'ordre μ, peut correspondre 1 à μ vecteurs propres selon le nombre de solutions indépendantes du système d'équations.

3-Forme diagonale d'une matrice

Si toutes les racines de l'équation caractéristique sont distinctes, il est possible de mettre la matrice A sous forme diagonale

$$\tilde{A} = \text{diag}\,[\lambda_1, \lambda_2, \ldots\ldots \lambda_n] \qquad \lambda_i \neq \lambda_j \text{ si } i \neq j$$

La matrice de transformation M est la matrice des vecteurs propres

$$M = [m_1 m_2 \ldots\ldots m_n] \qquad \text{avec} \quad A m_1 = \lambda_1 m_1$$

$$\tilde{A} = M^{-1} A M \quad \Longleftrightarrow \quad MA = AM \quad \Longleftrightarrow \quad MAM^{-1} = A$$

donc, on peut dire que si on a une matrice A telle que $\lambda_i \neq \lambda_j$ si $i \neq j$

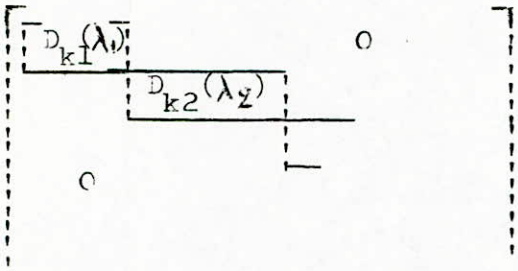
$$\Longrightarrow \exists\ \tilde{A} = \text{diag}\left[\lambda_1 \ldots\ldots \lambda_n\right] \quad \text{telle que} \quad A = M\tilde{A}M^{-1}$$

4- Forme quasi diagonale de Jordan

Appelons matrice élémentaire, une matrice $D_k(\lambda)$ (k.k), dont la diagonale principale est formée des k même nombres complexes λ ,et dont les éléments situés immédiatement à droite des éléments diagonaux sont égaux à 1 et tous les autre éléments nuls:

$$D_k(\lambda) = \begin{bmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{bmatrix}$$

Toute matrice A(n.n) est semblable à une matrice de la forme



dans laquelle les éléments situés au voisinage de la diagonale principale se distribuent dans une suite de matrices élémentaires de rang $k_1, k_2, \ldots k_n$ avec $k_1+k_2+\ldots\ldots k_n = n$; tous les autres éléments sont nuls.
Les éléments diagonaux $\lambda_1, \lambda_2, \ldots\ldots$ correspondant sont manifestement les valeurs propres de la matrice A.
Si $k_i > 1$, λ_i est valeur propre multiple d'ordre au moins égal à k_i.
Cet ordre vaut éxactement k_i si λ_i sont distinctes .Mais il peut arriver par exemple que $\lambda_2 = \lambda_1$, λ_1 est alors multiple d'ordre k_1+k_2 même si $k_1=1$.
Si tous les k_i sont egaux à 1,A est semblable à une matrice diagonale;

donc, on peut dire que si on a une matrice A telle que $\lambda_i \neq \lambda_j$ si $i \neq j$

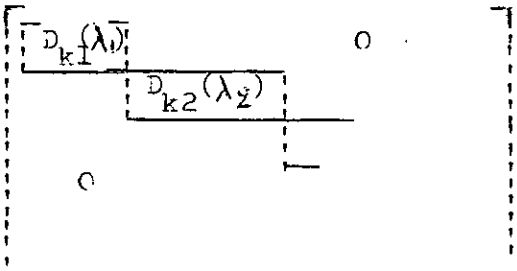
$$\Longrightarrow \exists\ \tilde{A} = \text{diag}\,[\lambda_1 \ldots\ldots \lambda_n] \quad \text{telle que} \quad A = M\tilde{A}M^{-1}$$

4- Forme quasi diagonale de Jordan

Appelons matrice élémentaire, une matrice $D_k(\lambda)$ (k.k), dont la diagonale principale est formée des k même nombres complexes λ ,et dont les éléments situés immédiatement à droite des éléments diagonaux sont égaux à 1 et tous les autre éléments nuls:

$$D_k(\lambda) = \begin{bmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{bmatrix}$$

Toute matrice A(n.n) est semblable à une matrice de la forme



dans laquelle les éléments situés au voisinage de la diagonale principale se distribuent dans une suite de matrices élémentaires de rang $k_1, k_2, \ldots k_n$ avec $k_1+k_2+\ldots\ldots k_n = n$; tous les autres éléments sont nuls.
Les éléments diagonaux $\lambda_1, \lambda_2, \ldots\ldots$ correspondant sont manifestement les valeurs propres de la matrice A.
Si $k_i > 1$, λ_i est valeur propre multiple d'ordre au moins égal à k_i.
Cet ordre vaut éxactement k_i si λ_i sont distinctes .Mais il peut arriver par exemple que $\lambda_2 = \lambda_1$, λ_1 est alors multiple d'ordre k_1+k_2 même si $k_1=1$.
Si tous les k_i sont egaux à 1,A est semblable à une matrice diagonale;

chacune des matrices élémentaires est alors de rang 1.

Donc en conclusion on peut dire que toute matrice A peut se mettre sous la forme de Jordan $A = M J M^{-1}$

J étant la matrice formée de blocs de Jordan, M étant la matrice formée des vecteurs propres

5-Théorème concernant l'algorithme de Newton

Soit f(x) une fonction de la variable réelle x, de classe c2 tel que $f(\hat{x}) = 0$ et $f'(\hat{x}) \neq 0$ ($\hat{x}$racine simple). Alors il existe un intervalle ouvert $N(\hat{x})$ contenant $\hat{x}$tel que si $x_1 \in N(\hat{x})$ l'algorithme de Newton

$$x_{K+1} = x_K + f(x_k)/f'(x_k)$$

engendre une suite x_k qui converge vers $\hat{x}$, de plus

$$\lim_{k \to \infty} \frac{x_{k+1} - \hat{x}}{(x_k - \hat{x})^2} = \frac{f''(\hat{x})}{2f'(\hat{x})}$$

résultat qui traduit une convergence finale d'ordre 2.

Remarque

Ce théorème peut s'appliquer aux fonctions de variables complexes analytiques.

Démonstration de la formule (9)

considérons la relation suivante

$$\frac{x_k}{2} < x_{k+1} < \frac{1}{2}(1+x_k)$$

soit

$$\frac{x_{k-1}}{2} < x_k \quad \Longrightarrow \quad \frac{x_{k-1}}{2^2} < x_{k+1}$$

chacune des matrices élémentaires est alors de rang 1.

Donc en conclusion on peut dire que toute matrice A peut se mettre sous la forme de Jordan $A = M J M^{-1}$

J étant la matrice formée de blocs de Jordan, M étant la matrice formée des vecteurs propres

5-Théorème concernant l'algorithme de Newton

Soit f(x) une fonction de la variable réelle x, de classe c2 tel que $f(\hat{x}) = 0$ et $f'(\hat{x}) \neq 0$ ($\hat{x}$racine simple). Alors il existe un intervalle ouvert $N(\hat{x})$ contenant $\hat{x}$tel que si $x_1 \in N(\hat{x})$ l'algorithme de Newton

$$x_{K+1} = x_K + f(x_k)/f'(x_k)$$

engendre une suite x_k qui converge vers $\hat{x}$, de plus

$$\lim_{k \to \infty} \frac{x_{k+1} - \hat{x}}{(x_k - \hat{x})^2} = \frac{f''(\hat{x})}{2f'(\hat{x})}$$

résultat qui traduit une convergence finale d'ordre 2.

Remarque

Ce théorème peut s'appliquer aux fonctions de variables complexes analytiques.

Démonstration de la formule (9)

considérons la relation suivante

$$\frac{x_k}{2} < x_{k+1} < \frac{1}{2}(1+x_k)$$

soit

$$\frac{x_{k-1}}{2} < x_k \implies \frac{x_{k-1}}{2^2} < x_{k+1}$$

$$\frac{x_{k-2}}{2} < x_{k-1} \Longrightarrow \frac{x_{k-2}}{2^2} < x_{k+1}$$

$$\vdots$$

$$\frac{x_c}{2} < x_1 \Longrightarrow \frac{x_o}{2^n} < x_n \quad \text{en prenant } k+1=n$$

Prenons maintenant l'inégalité de droite

$$x_k < \frac{1}{2}(1+x_{k-1}) \Longrightarrow x_{k+1} < \frac{1}{2}\left[1 + \frac{1}{2}(1+x_{k-1})\right]$$

$$x_{k-1} < \frac{1}{2}(1+x_{k-2}) \Longrightarrow x_{k+1} < \frac{1}{2}\left[1+ \frac{1}{2}(1+ \frac{1}{2}(1+x_{k-2}))\right]$$

$$\vdots$$

$$x_1 < \frac{1}{2}(1 + x_o) \Longrightarrow x_{k+1} < \frac{1}{2}\left[1+\frac{1}{2}(1+\frac{1}{2}(\ldots.(1+x_o).)\right]$$

d'où

$$x_{k+1} < \frac{1}{2} + \frac{1}{2^2} + \ldots\ldots + \frac{1}{2^n} + \frac{1}{2^n} x_o$$

le deuxième terme de l'inégalité est une suite géomètrique de raison 1/2 dont la somme est égale à

$$S = (1 - 1/2^n)$$

d'où l'on peut écrire

$$\frac{x_o}{2^n} < x_n < (1 - 1/2^n) + x_o/2^n$$

Démonstration de la formule( )

d'après (11) on a $\rho^2_{k+1} = \frac{1}{4}\rho^2_k + \frac{1}{4}\frac{1}{\rho^2_k} - 1/2 + \cos^2\theta_k$

comme $\rho_k > 1 \Longrightarrow 1/4.\rho^2_k \ll 1 \Longrightarrow \rho^2_{k+1} > \frac{1}{4}\rho^2_k - \frac{1}{2}$

maintenant si $1/\rho^2_k \simeq 1$ et $\cos^2\theta_k \simeq 1$ on aura $\rho^2_{k+1} < \frac{1}{4}\rho^2_k + 3/4$

$$\frac{x_{k-2}}{2} < x_{k-1} \Longrightarrow \frac{x_{k-2}}{2^2} < x_{k+1}$$

$$\vdots$$

$$\frac{x_o}{2} < x_1 \Longrightarrow \frac{x_o}{2^n} < x_n \quad \text{en prenant } k+1=n$$

Prenons maintenant l'inégalité de droite

$$x_k < \frac{1}{2}(1+x_{k-1}) \Longrightarrow x_{k+1} < \frac{1}{2}\left[1 + \frac{1}{2}(1+x_{k-1})\right]$$

$$x_{k-1} < \frac{1}{2}(1+x_{k-2}) \Longrightarrow x_{k+1} < \frac{1}{2}\left[1+\frac{1}{2}(1+\frac{1}{2}(1+x_{k-2}))\right]$$

$$\vdots$$

$$x_1 < \frac{1}{2}(1+x_o) \Longrightarrow x_{k+1} < \frac{1}{2}\left[1+\frac{1}{2}(1+\frac{1}{2}(\ldots.(1+x_o).)\right]$$

d'où

$$x_{k+1} < \frac{1}{2} + \frac{1}{2^2} + \ldots\ldots + \frac{1}{2^n} + \frac{1}{2^n} x_o$$

le deuxième terme de l'inégalité est une suite géomètrique de raison 1/2

dont la somme est égale à

$$S = (1 - 1/2^n)$$

d'où l'on peut écrire

$$\frac{x_o}{2^n} < x_n < (1 - 1/2^n) + x_o/2^n$$

Démonstration de la formule( )

d'après (11) on a $\rho^2_{k+1} = \frac{1}{4}\rho^2_k + \frac{1}{4}\frac{1}{\rho^2_k} - 1/2 + \cos^2\theta_k$

comme $\rho_k > 1 \Longrightarrow 1/4.\rho^2_k \ll 1 \Longrightarrow \rho^2_{k+1} > \frac{1}{4}\rho^2_k - \frac{1}{2}$

maintenant si $1/\rho^2_k \simeq 1$ et $\cos^2\theta_k \simeq 1$ on aura $\rho^2_{k+1} < \frac{1}{4}\rho^2_k + 3/4$

ANNEXE II

DEFINITIONS ET PROPRIETES IMPORTANTES

COMMANDABILITE

la paire (A,B) est dite commande si et seulement si tout état $x_{to} \neq 0$ peut être ramené à zero en un temps fini.

(A,B)commandable $\Longleftrightarrow$ rang$(B, AB, \ldots\ldots, A^{n-1} B) = n = \dim(A)$

RECONSTRUCTIBILITE

Etant donné une sortie $y = cx$,et $\dot{x} = Ax$, la paire (C,A) est reconstructible si et seulement si tout etat $x_{t1} \neq 0$ peut être déterminé de façon unique à partir de la sortie passée $y(t), t \in [t0, t1]$ sur un horizon (t1-t0) fini:

(C,A) reconstructible $\Longrightarrow$ rang $\left[C^T, A^T C^T, \ldots\ldots, (A^T)^{n-1} C^T\right] = n$

STABILISABILITE

La paire (A,B) est stabilisable si et seulement si:

$$\exists L: \text{réel } \lambda i \left[A - BL\right] < 0$$

DETACTIBILITE

La paire (C,A) est détéctable si et seulement si:

$$\exists M: \text{réel } \lambda i \left[A + MC\right] < 0$$

PROPRIETES

-étant donnée une paire (A,B) non commandable, la paire(A,B) est stabilisable si et seulement si les états non commandables sont asymptotiquement stables.

-étant donnée une paire (C,A) non reconstructible,la paire(C,A) est détéctable si et seulement si les états non reconstructibles sont asymptotiquement stables.

ANNEXE II

DEFINITIONS ET PROPRIETES IMPORTANTES

COMMANDABILITE

la paire (A,B) est dite commande si et seulement si tout état $x_{to} \neq 0$ peut être ramené à zero en un temps fini.

(A,B)commandable $\Longleftrightarrow$ rang$(B, AB, \ldots\ldots, A^{n-1} B) = n = \dim(A)$

RECONSTRUCTIBILITE

Etant donné une sortie $y = cx$,et $\dot{x} = Ax$, la paire (C,A) est reconstructible si et seulement si tout etat $x_{t1} \neq 0$ peut être déterminé de façon unique à partir de la sortie passée $y(t), t \in [tC, t1]$ sur un horizon (t1-t0) fini:

(C,A) reconstructible $\Longrightarrow$ rang $\left[C^T, A^T C^T, \ldots\ldots, (A^T)^{n-1} C^T\right] = n$

STABILISABILITE

La paire (A,B) est stabilisable si et seulement si:

$$\exists L: \text{réel } \lambda i \left[A - BL\right] < 0$$

DETACTIBILITE

La paire (C,A) est détéctable si et seulement si:

$$\exists M: \text{réel } \lambda i \left[A + MC\right] < 0$$

PROPRIETES

-étant donnée une paire (A,B) non commandable, la paire(A,B) est stabilisable si et seulement si les états non commandables sont asymptotiquement stables.

-étant donnée une paire (C,A) non reconstructible,la paire(C,A) est détéctable si et seulement si les états non réconstructibles sont asymptotiquement stables.

ANNEXE III

Démonstration du théorème concernant l'équation(continue) de Riccati.

Etant donné le problème de commande (40)-(40a) et les hypothèses (42),on sait que la matrice du système bouclé $\tilde{A}$(44) est asymptotiquement stable, la matrice P étant solution de l'équation de Riccati.Ceci étant,on peut alors vérifier que

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} = \begin{bmatrix} I & -V \\ P & I-PV \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & -A \end{bmatrix} \begin{bmatrix} I-VP & V \\ -P & I \end{bmatrix}$$

Il est clair que cette relation peut encore s'écrire

$$H = U \begin{bmatrix} A & 0 \\ 0 & -A \end{bmatrix} U^{-1}$$

ce qui démontre les propriétés 1 et 2 du théorème

Définissons maintenant la factorisation de Jordan de A par

$$A = W_{12}(-J)W_{12}^{-1}$$

et reportons ce résultat dans la première relation ci-dessus, il vient

$$H = \begin{bmatrix} W_{12} & -VW_{12} \\ PW_{12} & (I-PV)W_{12} \end{bmatrix} \begin{bmatrix} -J & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} W_{12}^{-1}(I-VP) & W_{12}^{-1}V \\ -W_{12}^{-1} & W_{12}^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} -VW_{12} & W_{12} \\ (1-PV)W_{12} & PW_{12} \end{bmatrix} \begin{bmatrix} J & 0 \\ 0 & -J \end{bmatrix} \begin{bmatrix} -W_{12}^{-1}P & W_{12}^{-1} \\ W_{12}^{-1}(I-VP) & W_{12}^{-1}V \end{bmatrix}$$

En identifiant cette relation avec la troisième relation du théorème,on obtient alors la relation (4) du théorème.

ANNEXE III

Démonstration du théorème concernant l'équation(continue) de Riccati.

Etant donné le problème de commande (40)-(40a) et les hypothèses (42),on sait que la matrice du système bouclé $\tilde{A}$(44) est asymptotiquement stable, la matrice P étant solution de l'équation de Riccati.Ceci étant,on peut alors vérifier que

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} = \begin{bmatrix} I & -V \\ P & I-PV \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & -A \end{bmatrix} \begin{bmatrix} I-VP & V \\ -P & I \end{bmatrix}$$

Il est clair que cette relation peut encore s'écrire

$$H = U \begin{bmatrix} A & 0 \\ 0 & -A \end{bmatrix} U^{-1}$$

ce qui démontre les propriétés 1 et 2 du théorème

Définissons maintenant la factorisation de Jordan de A par

$$A = W_{12}(-J)W_{12}^{-1}$$

et reportons ce résultat dans la première relation ci-dessus, il vient

$$H = \begin{bmatrix} W_{12} & -VW_{12} \\ PW_{12} & (I-PV)W_{12} \end{bmatrix} \begin{bmatrix} -J & Q \\ 0 & J \end{bmatrix} \begin{bmatrix} W_{12}^{-1}(I-VP) & W_{12}^{-1}V \\ -W_{12}^{-1} & W_{12}^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} -VW_{12} & W_{12} \\ (1-PV)W_{12} & PW_{12} \end{bmatrix} \begin{bmatrix} J & 0 \\ 0 & -J \end{bmatrix} \begin{bmatrix} -W_{12}^{-1}P & W_{12}^{-1} \\ W_{12}^{-1}(I-VP) & W_{12}^{-1}V \end{bmatrix}$$

En identifiant cette relation avec la troisième relation du théorème,on obtient alors la relation (4) du théorème.

Quelques formes équivalentes à: $P = A^TPA - A^TPB(B^TPB+R)^{-1}B^TPA + Q$ (1)

Soit le gain G introduit implicitemert:

$$G = (B^TPB + R)^{-1}B^TPA$$

de sorte que le système bouclé $\tilde{A}$ s'écrit

$$\tilde{A} = A - BG$$

Dans ces conditions, on peut vérifier que (1) est équivalente à

$$P = \tilde{A}^TP\tilde{A} + G^TRG + Q$$

De plus si $R > 0$, on a

$$\begin{aligned} P &= A^TPA - A^TPB(B^TPB + R)^{-1}B^TPA + Q \\ &= A^TP\left[I - B(B^TPB + R)^{-1}B^TP\right]A + Q \\ &= A^TP(I + BR^{-1}B^TP)^{-1}A + Q \end{aligned}$$

u encore

$$\begin{aligned} P &= A^T\left[I - PB(B^TPB+R)^{-1}B^T\right]PA + Q \\ &= A^T(I+PBR^{-1}B^T)^{-1}PA + Q \end{aligned}$$

l'on pose $P = K^TK$ (forme factorisée), on aura

$$\begin{aligned} K^TK &= A^TK^TKA - A^TK^TKB(B^TK^TKB + R)^{-1}B^TK^TKA + Q \\ K^TK &= A^TK^T\left[I - KB(B^TK^TKB + R)^{-1}B^TK^T\right]KA + Q \\ K^TK &= A^TK^T(I + KBR^{-1}B^TK^T)^{-1}KA + Q \end{aligned}$$

arque

s différentes transformations ont été opérées en utilisant l'identité

tricielle suivante:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Quelques formes équivalentes à: $P = A^T PA - A^T PB(B^T PB+R)^{-1}B^T PA + Q$ (1)

Soit le gain G introduit implicitement:

$$G = (B^T PB + R)^{-1}B^T PA$$

de sorte que le système bouclé $\tilde{A}$ s'écrit

$$\tilde{A} = A - BG$$

Dans ces conditions, on peut vérifier que (1) est équivalente à

$$P = \tilde{A}^T P\tilde{A} + G^T RG + Q$$

De plus si $R > 0$, on a

$$\begin{aligned} P &= A^T PA - A^T PB(B^T PB + R)^{-1}B^T PA + Q \\ &= A^T P\left[I - B(B^T PB + R)^{-1}B^T P\right]A + Q \\ &= A^T P(I + BR^{-1}B^T P)^{-1}A + Q \end{aligned}$$

ou encore

$$\begin{aligned} P &= A^T\left[I - PB(B^T PB+R)^{-1}B^T\right]PA + Q \\ &= A^T(I+PBR^{-1}B^T)^{-1}PA + Q \end{aligned}$$

si l'on pose $P = K^T K$ (forme factorisée), on aura

$$\begin{aligned} K^T K &= A^T K^T KA - A^T K^T KB(B^T K^T KB + R)^{-1}B^T K^T KA + Q \\ K^T K &= A^T K^T\left[I - KB(B^T K^T KB + R)^{-1}B^T K^T\right]KA + Q \\ K^T K &= A^T K^T(I + KBR^{-1}B^T K^T)^{-1}KA + Q \end{aligned}$$

Remarque

Les différentes transformations ont été obtenues en utilisant l'identité matricielle suivante:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Matrice symplectique

Soit une matrice réelle 2n.2n.Introduisons l'opérateur 2n?2n;

$$g = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \iff g^{-1} = -g = g^T$$

Définition; La matrice M est symplectique si et seulement si:

$$M^{-1} = g^{-1} M^T g$$

Propriété 1:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} D^T & -B^T \\ -C^T & A^T \end{bmatrix}$$

Propriété 2:

$$M^{-1}M = I \quad \begin{cases} A^T D - C^T B = D^T A - B^T C = I \\ B^T D \text{ symétrique; si } \exists\, D^{-1} : BD^{-1} \text{ symétrique} \\ A^T C \text{ symétrique; si } \exists\, A^{-1} : CA^{-1} \text{ symétrique} \end{cases}$$

Propriété 3:

$$MM^{-1} = I \quad \begin{cases} AD^T - BC^T = DA^T - CB^T = I \\ AB^T \text{ symétrique; si } \exists\, A^{-1} : A^{-1}B \text{ symétrique} \\ CD^T \text{ symétrique; si } \exists\, D^{-1} : D^{-1}C \text{ symétrique} \end{cases}$$

Propriété 4:

Si λ est valeur propre de M alors $1/\lambda$ l'est aussi, d'où detM=1

Propriété 5:

Soit $V = \begin{bmatrix} x \\ y \end{bmatrix}$ le vecteur propre associé à λ , alors$(y^T, -x^T)$ est le vecteur propre gauche associé à $1/\lambda$.

Matrice symplectique

Soit une matrice réelle 2n.2n.Introduisons l'opérateur 2n?2n;

$$g = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \iff g^{-1} = -g = g^T$$

Définition; La matrice M est symplectique si et seulement si:

$$M^{-1} = g^{-1} M^T g$$

Propriété 1:

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} D^T & -B^T \\ -C^T & A^T \end{bmatrix}$$

Propriété 2:

$$M^{-1}M = I \quad \begin{cases} A^T D - C^T B = D^T A - B^T C = I \\ B^T D \text{ symétrique; si } \exists\, D^{-1} : BD^{-1} \text{ symétrique} \\ A^T C \text{ symétrique; si } \exists\, A^{-1} : CA^{-1} \text{ symétrique} \end{cases}$$

Propriété 3:

$$MM^{-1} = I \quad \begin{cases} AD^T - BC^T = DA^T - CB^T = I \\ AB^T \text{ symétrique; si } \exists\, A^{-1} : A^{-1}B \text{ symétrique} \\ CD^T \text{ symétrique; si } \exists\, D^{-1} : D^{-1}C \text{ symétrique} \end{cases}$$

Propriété 4:

Si λ est valeur propre de M alors $1/\lambda$ l'est aussi, d'où detM=1

Propriété 5:

Soit $V = \begin{bmatrix} x \\ y \end{bmatrix}$ le vecteur propre associé à λ , alors$(y^T, -x^T)$ est le vecteur propre gauche associé à $1/\lambda$.

Matrice hamiltonienne

Soit H une matrice réelle 2n.2n

Définition: la matrice H est hamiltonienne si et seulement si:

$$-H = \mathcal{J}^{-1} H^T \mathcal{J}$$

Propriété 1: forme générique

$$H = \begin{bmatrix} A & B^T B \\ C^T C & -A^T \end{bmatrix}$$

Propriété 2:

Si λ est valeur propre de H, alors $-\lambda$ est aussi valeur propre de H.

Propriété 3;

Soit $V = \begin{bmatrix} x \\ y \end{bmatrix}$ le vecteur propre associé à λ, alors $(y^T, -x^T)$ est le vecteur propre gauche associé à $-\lambda$.

Relation entre matrice hamiltonienne et sympléctique

Soit S une matrice sympléctique telle que $|\lambda(S)| \neq 1$. On peut alors lui associer une matrice hamiltonienne H, telle que réel $[\mu(H)] \neq 0$ à l'aide de la transformation bilinéaire

$$\mu = \frac{\lambda + 1}{\lambda - 1}$$

d'où $$H = (S-1)^{-1}(S+1) = (S+1)(S-1)^{-1}$$

ou inversement

$$S = (H-1)^{-1}(H+1) = (H+1)(H-1)^{-1}$$

Matrice hamiltonienne

Soit H une matrice réelle 2n.2n

Définition: la matrice H est hamiltonienne si et seulement si:

$$-H = \mathcal{E}^{-1} H^T \mathcal{E}$$

Propriété 1: forme générique

$$H = \begin{bmatrix} A & B^T B \\ C^T C & -A^T \end{bmatrix}$$

Propriété 2:

Si λ est valeur propre de H, alors $-\lambda$ est aussi valeur propre de H.

Propriété 3;

Soit $V = \begin{bmatrix} x \\ y \end{bmatrix}$ le vecteur propre associé à λ, alors $(y^T, -x^T)$ est le vecteur propre gauche associé à $-\lambda$.

Relation entre matrice hamiltonienne et sympléctique

Soit S une matrice sympléctique telle que $|\lambda(S)| \neq 1$. On peut alors lui associer une matrice hamiltonienne H, telle que réel $[\mu(H)] \neq 0$ à l'aide de la transformation bilinéaire

$$\mu = \frac{\lambda + 1}{\lambda - 1}$$

d'où $$H = (S-1)^{-1}(S+1) = (S+1)(S-1)^{-1}$$

ou inversement

$$S = (H-1)^{-1}(H+1) = (H+1)(H-1)^{-1}$$

```
 1          C     PROGRAMME PRINCIPAL
 2          C     RESOLUTION DE L'EQUATION DE RICCATI DANS LE CAS CONTINU
 3          C     VARIATIONS DU NOMBRE D'ITERATIONS/A LA DIMENSION
 4                DIMENSION IL(36),IC(36),TEST(20)
 5                DIMENSION A(3,3),Q(3,3),AT(3,3)
 6                DIMENSION V(3,3),H(6,6),HM1(6,6)
 7                DIMENSION F121(3,3),F1(3,3),P(3,3)
 8                DIMENSION AE(6,6),SIGNE(6,6),F(6,6),F12(3,3)
 9                REAL ID(6,6)
10                DIMENSION AE1(6,6),AD(6,6),AD1(6,6)
11                DIMENSION AEK1(6,6),AEK2(6,6),AES(6,6)
12                DIMENSION PT(3,3),PC(3,3)
13                N=3
14   006          NC=6
15   00A          PR=5.
16   00E          QM1=0.
17   012          ETA1=0.00000001
18   012     C       MISE A ZERO
19   016          DO 1 I=1,N
20   01A          DO 1 J=1,N
21   01E          A(I,J)=0.
22   02E          V(I,J)=0.
23   03E     1    Q(I,J)=0.
24   062          A(1,1)=-1.
25   06C          A(2,1)=-1.
26   078          A(2,3)=-2.
27   084          A(3,2)=1.
28   08A          A(3,3)=-1.
29   096          V(1,1)=1.
30   09A          V(2,2)=1.
31   0A0          Q(1,1)=1.
32   0A4          Q(2,2)=2.
33   0AA          Q(3,3)=3.
34   0AA     C       CALCUL DE LA TRANSPOSEE DE A
35   0B0          DO 21 I=1,N
36   0B4          DO 5 J=1,N
37   0B8          AT(I,J)=A(J,I)
38   0D8     5    CONTINUE
39   0E2     21   CONTINUE
40   0E2     C       REMPLISSAGE DE LA MATRICE H
41   0EC          DO 23 I=1,N
42   0F0          DO 6  J=1,N
43   0F4          H(I,J)=A(I,J)
44   114     6    CONTINUE
45   11E     23   CONTINUE
46   128          DO 25 I=1,N
47   12C          DO 14 J=1,N
48   130          J1=J+N
49   136          H(I,J1)=-V(I,J)
50   15C     14   CONTINUE
51   166     25   CONTINUE
52   170          DO 26 I=1,N
53   174          DO 13 J=1,N
54   178          J1=J+N
55   17E          I1=I+N
```

```
                C     PROGRAMME PRINCIPAL
                C     RESOLUTION DE L'EQUATION DE RICCATI DANS LE CAS CONTINU
                C     VARIATIONS DU NOMBRE D'ITERATIONS/A LA DIMENSION
                      DIMENSION IL(36),IC(36),TEST(20)
                      DIMENSION A(3,3),Q(3,3),AT(3,3)
                      DIMENSION V(3,3),H(6,6),HM1(6,6)
                      DIMENSION F121(3,3),F1(3,3),P(3,3)
                      DIMENSION AE(6,6),SIGNE(6,6),F(6,6),F12(3,3)
                      REAL ID(6,6)
                      DIMENSION AE1(6,6),AD(6,6),AD1(6,6)
                      DIMENSION AEK1(6,6),AEK2(6,6),AES(6,6)
                      DIMENSION PT(3,3),PC(3,3)
                      N=3
  006                 NC=6
  00A                 PR=5.
  00E                 QM1=0.
  012                 ETA1=0.00000001
  012           C        MISE A ZERO
  016                 DO 1 I=1,N
  01A                 DO 1 J=1,N
  01E                 A(I,J)=0.
  02E                 V(I,J)=0.
  03E           1     Q(I,J)=0.
  062                 A(1,1)=-1.
  06C                 A(2,1)=-1.
  078                 A(2,3)=-2.
  084                 A(3,2)=1.
  08A                 A(3,3)=-1.
  096                 V(1,1)=1.
  09A                 V(2,2)=1.
  0A0                 Q(1,1)=1.
  0A4                 Q(2,2)=2.
  0AA                 Q(3,3)=3.
  0AA           C        CALCUL DE LA TRANSPOSEE DE A
  0B0                 DO 21 I=1,N
  0B4                 DO 5 J=1,N
  0B8                 AT(I,J)=A(J,I)
  0D8           5     CONTINUE
  0E2           21    CONTINUE
  0E2           C        REMPLISSAGE DE LA MATRICE H
  0EC                 DO 23 I=1,N
  0F0                 DO 6  J=1,N
  0F4                 H(I,J)=A(I,J)
  114           6     CONTINUE
  11E           23    CONTINUE
  128                 DO 25 I=1,N
  12C                 DO 14 J=1,N
  130                 J1=J+N
  136                 H(I,J1)=-V(I,J)
  15C           14    CONTINUE
  166           25    CONTINUE
  170                 DO 26 I=1,N
  174                 DO 13 J=1,N
  178                 J1=J+N
  17E                 I1=I+N
```

```
 56    184            H(I1,J1)=-AT(I,J)
 57    1AA    13      CONTINUE
 58    1B4    26      CONTINUE
 59    1BE            DO 27 I=1,N
 60    1CE            DO 17 J=1,N
 61    1C6            I1=I+N
 62    1CC            H(I1,J)=-Q(I,J)
 63    1F2    17      CONTINUE
 64    1FC    27      CONTINUE
 65    206            PRINT 90
 66    210            PRINT 71,((H(I,J),J=1,NC),I=1,NC)
 67    24A            CALL MRINV(H,HM1,NC,KOD,DET,EPS,IL,IC)
 68    24E            PRINT 60
 69    258            PRINT 71,((HM1(I,J),J=1,NC),I=1,NC)
 70    292            CALL NORM(H,X,NC)
 71    296            PRINT 89,X
 72    2A6            CALL NORM(HM1,Y,NC)
 73    2AA            PRINT 99,Y
 74    2BA            AMAX=Y
 75    2BE            IF(X.GT.Y) AMAX=X
 76    2D6            GO TO 79
 77    2D8    79      CONTINUE
 78    2D8            PRINT 550,AMAX
 79    2D8    C          CALCUL DU NOMBRE D'ITERATIONS TOTAL
 80    2E8            N1=ALOG(AMAX)
 81    2F0            N1=N1/ALOG(2)
 82    302            N1=IFIX(N1)
 83    30A            PRINT 80,N1
 84    31A            N1=N1+QM1+PR
 85    326            N1=IFIX(N1)
 86    32E            PRINT 110,N1
 87    32E    C          CALCUL DE LA MATRICE SIGNE
 88    33E            DO 9 I=1,NC
 89    342            DO 9 J=1,NC
 90    346    9       AE(I,J)=H(I,J)
 91    372            CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
 92    376            CALL NORM(AE,X,NC)
 93    37A            CALL NORM(AE1,Y,NC)
 94    37E            ALPHA=SQRT(Y/X)
 95    386            PRINT 82,ALPHA
 96    396            BETA=ALPHA
 97    39A            CALL MUL2(AE,ALPHA,AD,NC)
 98    39E            ALPHA=1/ALPHA
 99    3A6            CALL MUL2(AE1,ALPHA,AD1,NC)
100    3AA            CALL SOMMAT(AD,AD1,AES,NC)
101    3AE            CALL MUL2(AES,0.5,AEK1,NC)
102    3B4            BETA=BETA-1
103    3C2            BETA=ABS(BETA)
104    3C8            DO 83 I=1,NC
105    3CE            DO 84 J=1,NC
106    3D6            AEK2(I,J)=AEK1(I,J)-AE(I,J)
107    3F8            AE(I,J)=AEK1(I,J)
108    410    84      CONTINUE
109    420    83      CONTINUE
110    42E            CALL NORM(AEK2,W,NC)
```

```
 56    184            H(I1,J1)=-AT(I,J)
 57    1AA    13      CONTINUE
 58    1B4    26      CONTINUE
 59    1BE            DO 27 I=1,N
 60    1CE            DO 17 J=1,N
 61    1C6            I1=I+N
 62    1CC            H(I1,J)=-Q(I,J)
 63    1F2    17      CONTINUE
 64    1FC    27      CONTINUE
 65    206            PRINT 90
 66    210            PRINT 71,((H(I,J),J=1,NC),I=1,NC)
 67    24A            CALL MRINV(H,HM1,NC,KOD,DET,EPS,IL,IC)
 68    24E            PRINT 60
 69    258            PRINT 71,((HM1(I,J),J=1,NC),I=1,NC)
 70    292            CALL NORM(H,X,NC)
 71    296            PRINT 89,X
 72    2A6            CALL NORM(HM1,Y,NC)
 73    2AA            PRINT 99,Y
 74    2BA            AMAX=Y
 75    2BE            IF(X.GT.Y) AMAX=X
 76    2D6            GO TO 79
 77    2D8    79      CONTINUE
 78    2D8            PRINT 550,AMAX
 79    2D8    C          CALCUL DU NOMBRE D'ITERATIONS TOTAL
 80    2E8            N1=ALOG(AMAX)
 81    2F0            N1=N1/ALOG(2)
 82    302            N1=IFIX(N1)
 83    30A            PRINT 80,N1
 84    31A            N1=N1+QM1+PR
 85    326            N1=IFIX(N1)
 86    32E            PRINT 110,N1
 87    32E    C          CALCUL DE LA MATRICE SIGNE
 88    33E            DO 9 I=1,NC
 89    342            DO 9 J=1,NC
 90    346    9       AE(I,J)=H(I,J)
 91    372            CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
 92    376            CALL NORM(AE,X,NC)
 93    37A            CALL NORM(AE1,Y,NC)
 94    37E            ALPHA=SQRT(Y/X)
 95    386            PRINT 82,ALPHA
 96    396            BETA=ALPHA
 97    39A            CALL MUL2(AE,ALPHA,AD,NC)
 98    39E            ALPHA=1/ALPHA
 99    3A6            CALL MUL2(AE1,ALPHA,AD1,NC)
100    3AA            CALL SOMMAT(AD,AD1,AES,NC)
101    3AE            CALL MUL2(AES,0.5,AEK1,NC)
102    3B4            BETA=BETA-1
103    3C2            BETA=ABS(BETA)
104    3C8            DO 83 I=1,NC
105    3CE            DO 84 J=1,NC
106    3D6            AEK2(I,J)=AEK1(I,J)-AE(I,J)
107    3F8            AE(I,J)=AEK1(I,J)
108    410    84      CONTINUE
109    420    83      CONTINUE
110    42E            CALL NORM(AEK2,W,NC)
```

```
111   432          GAMMA=W/X
112   43C          IF(BETA.GT.GAMMA) TEST(1)=BETA
113   45C          TEST(1)=GAMMA
114   462          TEST(1)=TEST(1)/2
115   472          DO 10 K=1,N1
116   478          K1=K+1
117   47E          CALL NORM(AE,X,NC)
118   482          CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
119   486          CALL NORM(AE1,Y,NC)
120   48A          ALPHA=SQRT(Y/X)
121   494          PRINT 82,ALPHA
122   4A4          BETA=ALPHA
123   4A8          CALL MUL2(AE,ALPHA  ),NC)
124   4AE          ALPHA=1/ALPHA
125   4B8          CALL MUL2(AE1,ALPHA,AD1,NC)
126   4BE          CALL SOMMAT(AD,AD1,AES,NC)
127   4C2          CALL MUL2(AES,0.5,AEK1,NC)
128   4C6          BETA=BETA-1
129   4D4          BETA=ABS(BETA)
130   4DA          DO 85 I=1,NC
131   4E0          DO 86 J=1,NC
132   4E8          AEK2(I,J)=AEK1(I,J)-AE(I,J)
133   50A          AE(I,J)=AEK1(I,J)
134   522    86    CONTINUE
135   532    85    CONTINUE
136   540          CALL NORM(AEK2,W,NC)
137   544          GAMMA=W/X
138   54E          IF(BETA GT.GAMMA) TEST(K1)=BETA
139   576          TEST(K1)=GAMMA
140   584          IF(TEST(K1).LT.ETA1) GOTO 33
141   5A4          GOTO 34
142   5A6    33    IF(TEST(K1).GT.TEST(K1-1)) GOTO 35
143   5D6          GOTO 34
144   5D8    34    TEST(K1)=TEST(K1)/2
145   5FA    10    CONTINUE
146   60A    35    CONTINUE
147   60A          DO 11 I=1,NC
148   612          DO 11 J=1,NC
149   61A    11    SIGNE(I,J)=AEK1(I,J)
150   64E          PRINT 100
151   658          PRINT 101,((SIGNE(I,J),J=1,NC),I=1,NC)
152   6A2          DO 7 I=1,NC
153   6AA          DO 7 J=1,NC
154   6B2          ID(I,J)=0.
155   6C4          IF(I.EQ.J) ID(I,J)=1
156   6E8    7     CONTINUE
157   704          PRINT 150
158   70E          PRINT 160,((ID(I,J),J=1,NC),I=1,NC)
159   70E    C        CALCUL DE LA MATRICE F
160   758          CALL SOMMAT(SIGNE,ID,F,NC)
161   75C          PRINT 71,((F(I,J),J=1,NC),I=1,NC)
162   7A6          CALL MUL2(F,0.5,F,NC)
163   7AA          DO 12 I=1,N
164   7B2          DO 12 J=1,N
165   7BA          J1=J+N
```

```
111   432          GAMMA=W/X
112   43C          IF(BETA.GT.GAMMA) TEST(1)=BETA
113   45C          TEST(1)=GAMMA
114   462          TEST(1)=TEST(1)/2
115   472          DO 10 K=1,N1
116   478          K1=K+1
117   47E          CALL NORM(AE,X,NC)
118   482          CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
119   486          CALL NORM(AE1,Y,NC)
120   48A          ALPHA=SQRT(Y/X)
121   494          PRINT 82,ALPHA
122   4A4          BETA=ALPHA
123   4A8          CALL MUL2(AE,ALPHA  ),NC)
124   4AE          ALPHA=1/ALPHA
125   4B8          CALL MUL2(AE1,ALPHA,AD1,NC)
126   4BE          CALL SOMMAT(AD,AD1,AES,NC)
127   4C2          CALL MUL2(AES,0.5,AEK1,NC)
128   4C6          BETA=BETA-1
129   4D4          BETA=ABS(BETA)
130   4DA          DO 85 I=1,NC
131   4E0          DO 86 J=1,NC
132   4E8          AEK2(I,J)=AEK1(I,J)-AE(I,J)
133   50A          AE(I,J)=AEK1(I,J)
134   522    86    CONTINUE
135   532    85    CONTINUE
136   540          CALL NORM(AEK2,W,NC)
137   544          GAMMA=W/X
138   54E          IF(BETA .GT.GAMMA) TEST(K1)=BETA
139   576          TEST(K1)=GAMMA
140   584          IF(TEST(K1).LT.ETA1) GOTO 33
141   5A4          GOTO 34
142   5A6    33    IF(TEST(K1).GT.TEST(K1-1)) GOTO 35
143   5D6          GOTO 34
144   5D8    34    TEST(K1)=TEST(K1)/2
145   5FA    10    CONTINUE
146   60A    35    CONTINUE
147   60A          DO 11 I=1,NC
148   612          DO 11 J=1,NC
149   61A    11    SIGNE(I,J)=AEK1(I,J)
150   64E          PRINT 100
151   658          PRINT 101,((SIGNE(I,J),J=1,NC),I=1,NC)
152   6A2          DO 7 I=1,NC
153   6AA          DO 7 J=1,NC
154   6B2          ID(I,J)=0.
155   6C4          IF(I.EQ.J) ID(I,J)=1
156   6E8    7     CONTINUE
157   704          PRINT 150
158   70E          PRINT 160,((ID(I,J),J=1,NC),I=1,NC)
159   70E    C        CALCUL DE LA MATRICE F
160   758          CALL SOMMAT(SIGNE,ID,F,NC)
161   75C          PRINT 71,((F(I,J),J=1,NC),I=1,NC)
162   7A6          CALL MUL2(F,0.5,F,NC)
163   7AA          DO 12 I=1,N
164   7B2          DO 12 J=1,N
165   7BA          J1=J+N
```

```
166   7C2          F12(I,J)=F(I,J1)
167   7F2    12    F1(I,J)=F(I,J)
168   820          CALL MRINV(F12,F121,N,KOD,DET,EPS,IL,IC)
169   824          DO 8 I=1,N
170   82C          DO 8 J=1,N
171   834    8     F121(I,J)=-F12[illegible](I,J)
172   834    C        CALCUL DE LA MATRICE P
173   86E          CALL PROD(F121,F1,P,N)
174   872          PRINT 200
175   87C          PRINT 201,((P(I,[illegible]),J=1,N),I=1,N)
176   87C    C        CALCUL DE LA MATRICE P CORRIGEE
177   8C6          DO 56 I=1,N
178   8CE          DO 57 J=,N
179   8D6          PT(I,J)=P(J,I)
180   8F8    57    CONTINUE
181   908    56    CONTINUE
182   916          CALL SOMMAT(P,PT,PC,N)
183   91A          CAMM MUL2(PC,0.5,PC,N)
184   91E          PRINT 300
185   928          PRINT 201,((PC(I,J),J=1,N),I=1,N)
186   972    90    FORMAT(////,10X,'MATRICE H')
187   972    150   FORMAT(////,10X,'MATRICE ID')
188   972    300   FORMAT(////,10X,'MATRICE PC')
189   972    1[illegible]  FORMAT(////,20X,'NOMBRE D'ITERATIONS=',I2)
190   972    101   FORMAT(//// 6(5X,F9.6))
191   972    100   FORMAT(////,10X,'MATRICE SIGNE.')
192   972    200   FORMAT(////,10X,'MATRICE P')
193   972    82    FORMAT(////,10X,'ALPHA=',F12.5,/)
194   972    60    FORMAT(////,10X,'MATRICE HM1')
195   972    80    FORMAT(////,20X,'N1=',I2)
196   972    89    FORMAT(1H,////,20X,'NORM DE H=',F12.6)
197   972    99    FORMAT(////,10X,'NORM[illegible] DE HM1=',F12.6)
198   972    550   FORMAT(////,10X,'AMAX=',F6.3,/)
199   972    201   FORMAT(////,3(4X,F10.7))
200   972    71    FORMAT(////,6(3X,F10.6))
201   972    160   FORMAT(////,6(3X,[illegible]2.0))
202   972          STOP
203   974          END
```

```
166   7C2           F12(I,J)=F(I,J1)
167   7F2    12     F1(I,J)=F(I,J)
168   820           CALL MRINV(F12,F121,N,KOD,DET,EPS,IL,IC)
169   824           DO 8 I=1,N
170   82C           DO 8 J=1,N
171   834    8      F121(I,J)=-F12?(I,J)
172   834    C         CALCUL DE LA MATRICE P
173   86E           CALL PROD(F121,F1,P,N)
174   872           PRINT 200
175   87C           PRINT 201,((P(I,J),J=1,N),I=1,N)
176   87C    C         CALCUL DE LA MATRICE P CORRIGEE
177   8C6           DO 56 I=1,N
178   8CE           DO 57 J=1,N
179   8D6           PT(I,J)=P(J,I)
180   8F8    57     CONTINUE
181   908    56     CONTINUE
182   916           CALL SOMMAT(P,PT,PC,N)
183   91A           CALL MUL2(PC,0.5,PC,N)
184   91E           PRINT 300
185   928           PRINT 201,((PC(I,J),J=1,N),I=1,N)
186   972    90     FORMAT(////,10X,'MATRICE H')
187   972    150    FORMAT(////,10X,'MATRICE ID')
188   972    300    FORMAT(////,10X,'MATRICE PC')
189   972    1?     FORMAT(////,20X,'NOMBRE D'ITERATIONS=',I2)
190   972    101    FORMAT(////,6(5X,F9.6))
191   972    100    FORMAT(////,10X,'MATRICE SIGNE.')
192   972    200    FORMAT(////,10X,'MATRICE P')
193   972    82     FORMAT(////,10X,'ALPHA=',F12.5,/)
194   972    60     FORMAT(////,10X,'MATRICE HM1')
195   972    80     FORMAT(////,20X,'N1=',I2)
196   972    89     FORMAT(1H,////,20X,'NORM DE H=',F12.6)
197   972    99     FORMAT(////,10X,'NORME DE HM1=',F12.6)
198   972    550    FORMAT(////,10X,'AMAX=',F6.3,/)
199   972    201    FORMAT(////,3(4X,F10.7))
200   972    71     FORMAT(////,6(3X,F10.6))
201   972    160    FORMAT(////,6(3X,?2.0))
202   972           STOP
203   974           END
```

```
1                       C     PROGRAMME PRINCIPAL
2                       C     RESOLUTION DE L'EQUATION DE RICCATI DANS LE CAS DISCRET
3                             DIMENSION IL (100), IC (100)
4                             DIMENSION AEK1 (10,10), AEK2(10,10), AE1(10,10)
5                             DIMENSION HT (5,5), SIGNE (10,10), F(10,10), AE(10,10)
6                             DIMENSION A(5,5), B (5,5), R(5,5), H(5,5)
7                             DIMENSION V(5,5), AT(5,5), BT (5,5), U (10,10)
8                             DIMENSION VI (5,5), FT (5,5), PC (5,5)
9                             DIMENSION Q (5,5), T (10,10), RM1 (5,5)
10                            DIMENSION ID (5,5), U1 (10,10), S (10,10), SS1 (10,10)
11                            DIMENSION F1 (5,5), P21 (5,5), PC(5,5)
12                            DIMENSION FF1 (10,10), AD (10,10) AD1 (10,10)
13                            REAL  M (10,10), MM1 (10,10), L (10,10), IE (10,10)
14                            DIMENSION TEST (20)
15                            N = 5
16      006                   NC =10
17      00A                   QM1 = 0.
18      00E                   PR = 5.
19      012                   ETA1 = 0.00000001
20      016                   DO 1  I=1,N
21      01A                   DO 1J = 1,N
22      01E                   A (I,J) = 0.
23      02E                   R (I, J) = 0.
24      03E                   Q (I,J)  = 0.
25      04E       1           L (I,J)  =0.
26      072                   A (1,1)  =0.75
27      076                   A (1,2)  =0.09
28      07C                   A (2,1)  =1.74
29      088                   A (2,2)  =0.91
30      08E                   A( 3,1)  =0.3
31      09A                   A (3,2)  =0.0015
32      0A6                   A (3,3)  =0.95
33      0AC                   A (4,4)  =0.55
34      0B2                   A (5,1)  =0.15
35      0BE                   A (5,2)  =0.008
36      0CA                   A (5,5)  =0.905
37      0D0                   PRINT  30
38      0DA                   PRINT 31, ( (A(I,J)J=1,N), I=1,N)
39      114                   H (1,3) = 24.64
40      11A                   H (2,4) = 0.835
41      120                   H (3,5) = 1.83
42      126                   H (1,1) = 1.
43      12A                   B (2,2) = 2.
44      130                   B (4,2) = 1.
45      136                   B (5,1) = 1.
46      13C                   PRINT  40
47      146                   PRINT  41, ( (B (I,J),J =1,N), I=1,N)
48      180                   DO 55 I =1, N
49      184                   DO 4 J = 1, N
50      188                   BT (I,J) = B (J,I)
51      1A8       4           CONTINUE
52      1B2      55           CONTINUE
53      1BC                   DO  221 = 1, N
54      100                   DO  15 J = 1,N
55      1C4      15           HT (I,J) =P (J,I)
56      1EE      22           CONTINUE
```

```
1                       C       PROGRAMME PRINCIPAL
2                       C       RESOLUTION DE L'EQUATION DE RICCATI DANS LE CAS DISCRET
3                               DIMENSION IL (100), IC (100)
4                               DIMENSION AEK1 (10,10), AEK2(10,10), AE1(10,10)
5                               DIMENSION HT (5,5), SIGNE (10,10), F(10,10), AE(10,10)
6                               DIMENSION A(5,5), B (5,5), R(5,5), H(5,5)
7                               DIMENSION V(5,5), AT(5,5), BT (5,5), U (10,10)
8                               DIMENSION VI (5,5), FT (5,5), PC (5,5)
9                               DIMENSION Q (5,5), T (10,10), RM1 (5,5)
10                              DIMENSION ID (5,5), U1 (10,10), S (10,10), SS1 (10,10)
11                              DIMENSION F1 (5,5), P21 (5,5), PC(5,5)
12                              DIMENSION FF1 (10,10), AD (10,10) AD1 (10,10)
13                              REAL  M (10,10), MM1 (10,10), L (10,10), IE (10,10)
14                              DIMENSION TEST (20)
15                              N = 5
16      006                     NC =10
17      00A                     QM1 = 0.
18      00E                     PR = 5.
19      012                     ETA1 = 0.00000001
20      016                     DO 1  I=1,N
21      01A                     DO 1J = 1,N
22      01E                     A (I,J) = 0.
23      02E                     R (I, J) = 0.
24      03E                     Q (I,J)  = 0.
25      04E     1               L (I,J)  =0.
26      072                     A (1,1)  =0.75
27      076                     A (1,2)  =0.09
28      07C                     A (2,1)  =1.74
29      088                     A (2,2)  =0.91
30      08E                     A( 3,1)  =0.3
31      09A                     A (3,2)  =0.0015
32      0A6                     A (3,3)  =0.95
33      0AC                     A (4,4)  =0.55
34      0B2                     A (5,1)  =0.15
35      0BE                     A (5,2)  =0.008
36      0CA                     A (5,5)  =0.905
37      0D0                     PRINT  30
38      0DA                     PRINT 31, ( (A(I,J)J=1,N), I=1,N)
39      114                     H (1,3) = 24.64
40      11A                     H (2,4) = 0.835
41      120                     H (3,5) = 1.83
42      126                     H (1,1) = 1.
43      12A                     B (2,2) = 2.
44      130                     B (4,2) = 1.
45      136                     B (5,1) = 1.
46      13C                     PRINT  40
47      146                     PRINT  41, ( (B (I,J),J =1,N), I=1,N)
48      180                     DO 55 I =1, N
49      184                     DO 4 J = 1, N
50      188                     BT (I,J) = B (J,I)
51      1A8     4               CONTINUE
52      1B2     55              CONTINUE
53      1BC                     DO  221 = 1, N
54      100                     DO  15 J = 1,N
55      1C4     15              HT (I,J) =P (J,I)
56      1EE     22              CONTINUE
```

```
57          1F8                 DO  21 I =1,N
58          1FC                 DO 5 J = 1,N
59          200       5         AT (I,J)= A(J,I)
60          22A      21         CONTINUE
61          234                 CALL PROD (HT,H,Q,N)
62          238                 PRINT 51, ((Q(I,J), J = 1,N), I = 1,N)
63          272                 R (1,1) = 1.
64          276                 R (2,2) = 1.
65          27C                 RR(3,3) = 1.
66          282                 R (4,4) = 1.
67          288                 R (5,5) = 1.
68          28E                 PRINT 60
69          298                 PRINT 41, ((R(I,J), = 1,N) , I = 1,N)
70          2D2                 CALL MRINV (R, RM1,N,KOD,DET,EPS,IL,1C)
71          2D8                 PRINT 41, ((RM1 (I,J), J =1,N), I = 1,N)
72          322                 CALL PROD (B,RM1, V1, N)
73          326                 CALL PROD (V1,BT,V?N)
74          32A                 DO 23 1= 1,N
75          332                 DO  6J = 1,N
76          33A                 U(I,J) =0
77          34C                 IF (I.EQ.J)L (I,J) = 1.
78!         370                 GOTO 6
79          372       6         CONTINUE
80          380      23         CONTINUE
81          38E                 DO 25 I = 1,N
82          396                 DO 14 J = 1,N
83          39E                 J1 = J+N
84          3A8                 U(I,J1) = V (I,J)
85          3CE      14         CONTINUE
86          3DE      25         CONBINUE
87          3EC                 DO 26 I= 1,N
88          3F4                 DO 13 J = 1,N
89          3FC                 J1 = J * N
90          406                 I1 = I + N
91          410                 U (I1, J1) = AT (I,J)
92          436      13         CONTINUE
93          446      26         CONTINUE
94          454                 DO 27 I = 1,N
95          45C                 DO 17 J =1,N
96          464                 I 1 = I + 5
97          46E                 U (I1,J) = 0.
98          484      17         CONTINUE
99          494      27         CONTINUE
100         4A2                 PRINT 160
101         4AC                 PRINT 71, (●U(I,J)J=1;NC,I=1,NC)
102         4F6                 DO 7 I=1,N
103         4FE                 DO 7 J=1,N
104         506                 I1 = 1,N
105         510                 J1 = J+N
106         51A                 L (I!,J1)=0
107         536                 IF (I1.EQ●J1)L(I1,J1)=1
108         564                 GOTO 7
109         566       7         CONTINUE
110         582                 DO 28 J = 1,N
```

```
57          1F8                 DO  21 I =1,N
58          1FC                 DO 5 J = 1,N
59          200       5         AT (I,J)= A(J,I)
60          22A      21         CONTINUE
61          234                 CALL PROD (HT,H,Q,N)
62          238                 PRINT 51, ((Q(I,J), J = 1,N), I = 1,N)
63          272                 R (1,1) = 1.
64          276                 R (2,2) = 1.
65          27C                 RR(3,3) = 1.
66          282                 R (4,4) = 1.
67          288                 R (5,5) = 1.
68          28E                 PRINT 60
69          298                 PRINT 41, ((R(I,J), = 1,N) , I = 1,N)
70          2D2                 CALL MRINV (R, RM1,N,KOD,DET,EPS,IL,1C)
71          2D8                 PRINT 41, ((RM1 (I,J), J =1,N), I = 1,N)
72          322                 CALL PROD (B,RM1, V1, N)
73          326                 CALL PROD (V1,BT,V?N)
74          32A                 DO 23 1= 1,N
75          332                 DO  6J = 1,N
76          33A                 U(I,J) =0
77          34C                 IF (I.EQ.J)L (I,J) = 1.
78          370                 GOTO 6
79          372       6         CONTINUE
80          380      23         CONTINUE
81          38E                 DO 25 I = 1,N
82          396                 DO 14 J = 1,N
83          39E                 J1 = J+N
84          3A8                 U(I,J1) = V (I,J)
85          3CE      14         CONTINUE
86          3DE      25         CONTINUE
87          3EC                 DO 26 I= 1,N
88          3F4                 DO 13 J = 1,N
89          3FC                 J1 = J * N
90          406                 I1 = I + N
91          410                 U (I1, J1) = AT (I,J)
92          436      13         CONTINUE
93          446      26         CONTINUE
94          454                 DO 27 I = 1,N
95          45C                 DO 17 J =1,N
96          464                 I 1 = I + 5
97          46E                 U (I1,J) = 0.
98          484      17         CONTINUE
99          494      27         CONTINUE
100         4A2                 PRINT 160
101         4AC                 PRINT 71, (U(I,J)J=1;NC,I=1,NC)
102         4F6                 DO 7 I=1,N
103         4FE                 DO 7 J=1,N
104         506                 I1 = 1,N
105         510                 J1 = J+N
106         51A                 L (I!,J1)=0
107         536                 IF (I1.EQ.J1)L(I1,J1)=1
108         564                 GOTO 7
109         566       7         CONTINUE
110         582                 DO 28 J = 1,N
```

```
111     58A             DO  16 J = 1,N
112     592             L (I,J)= A (I,J)
113     5B4     16      CONTINUE
114     5C4     28      CONTINUE
115     5D2             DO 24 J = 1,N
116     5DA             DO 18 J = 1,N
117     5E2             I1 = I+N
118     5EC             L(I1,J) = -Q(I,J)
119     618     18      CONTINUE
120     628     24      CONTINUE
121     636             DO  19 I = 1,N
122     63E             DO 19 J = 1,N
123     646             J1 = J+N
124     650             L(I,J1)=0
125     66C     19      CONTINUE
126     68A             PRINT 150
127     694             PRINT 71, ((L (I,J), J =1,NC), I=1,NC)
128     6DE             CALL SOMMAT (L,U,T,NC)
129     6E2             DO 81 = 1,NC
130     6EA             DO8 J= 1,NC
131     6F2             U1 (I,J)=-U(I,J)
132     734             CALL SOMMAT (L,U1,M,NC)
133     738             CALL MRINV (MM1,NC,KOD,DET,EPS,I1,IC)
134     73C             CALL PROD(MM1,T,S,NC)
135     740             PRINT 90
136     74A             PRINT 71,((S(I,J),J=1,NC),I=1,NC)
137     794             CALL NORM(S,X,NC)
138     798             PRINT 100,X
139     7A8             CALL MRINV(S,SS1,NC,KOD,DET,EPS,IL,IC)
140     7AC             CALL NORM(SS1,Y,NC)
141     780             AMAX=Y
142     7B4             IF(X.GT.Y) AMAX=X
143     7CC             GOTO 79
144     7CE      79     CONTINUE
145     7CE             N1=ALOG(AMAX)
146     7D6             N1=N1/ALOG(2)
147     7F0             N1=IFIX(N1)
148     7F8             N1=N1+QM1+PR
149     808             PRINT 110,N1
```

```
111     58A             DO  16 J = 1,N
112     592             L (I,J)= A (I,J)
113     5B4     16      CONTINUE
114     5C4     28      CONTINUE
115     5D2             DO 24 J = 1,N
116     5DA             DO 18 J = 1,N
117     5E2             I1 = I+N
118     5EC             L(I1,J) = -Q(I,J)
119     618     18      CONTINUE
120     628     24      CONTINUE
121     636             DO  19 I = 1,N
122     63E             DO 19 J = 1,N
123     646             J1 = J+N
124     650             L(I,J1)=0
125     66C     19      CONTINUE
126     68A             PRINT 150
127     694             PRINT 71, ((L (I,J), J =1,NC), I=1,NC)
128     6DE             CALL SOMMAT (L,U,T,NC)
129     6E2             DO 81 = 1,NC
130     6EA             DO8 J= 1,NC
131     6F2             U1 (I,J)=-U(I,J)
132     734             CALL SOMMAT (L,U1,M,NC)
133     738             CALL MRINV (MM1,NC,KOD,DET,EPS,I1,IC)
134     73C             CALL PROD(MM1,T,S,NC)
135     740             PRINT 90
136     74A             PRINT 71,((S(I,J),J=1,NC),I=1,NC)
137     794             CALL NORM(S,X,NC)
138     798             PRINT 100,X
139     7A8             CALL MRINV(S,SS1,NC,KOD,DET,EPS,IL,IC)
140     7AC             CALL NORM(SS1,Y,NC)
141     780             AMAX=Y
142     7B4             IF(X.GT.Y) AMAX=X
143     7CC             GOTO 79
144     7CE      79     CONTINUE
145     7CE             N1=ALOG(AMAX)
146     7D6             N1=N1/ALOG(2)
147     7F0             N1=IFIX(N1)
148     7F8             N1=N1+QM1+PR
149     808             PRINT 110,N1
```

```
150     818          DO       9I=1,NC
151     820          DO       9J=1,NC
152     828    9     AE(I,J)    =S(I,J)
153     864          CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
154     868          CALL NORM(AE,X,NC)
155     86C          CALL NORM(AE1,Y,NC)
156     870          ALPHA=SORT(Y/X)
157     878          PRINT 82,ALPHA
158     888          BETA=ALPHA
159     88C          CALL MUL2(AE,ALPHA,AD,NC)
160     890          ALPHA=1/ALPHA
161     898          CALL MUL2(AE1,ALPHA,AD1,NC)
162     89C          CALL SOMMAT4AD,AD1,AEK1,NC)
163     8A0          CALL MUL2(AEK1,0.5,AEK1,NC)
164     8A4          BETA=BETA-1
165     8B8          BETA=ABS(BETA)
166     8BE          DO 84 I=1,NC
167     8C6          DO 83 J=1 NC
168     8CE          AEK2(I,J)=AEK1(I,J)-AE(I,J)
169     8FC          AE(I,J)=AEK1(I,J)
170     91C    83    CONTINUE
171     92C    84    CONTINUE
172     93A          CALL NORM(AEK2,W,NC)
173     93E          GAMMA=W/X
174     944          IF(BETA.GT.GAMMA) TEST(1)=BETA
175     960          TEST(1)=GAMMA
176     966          TEST(1)=TEST(1)/2
177     976          DO 10 K=1,N1
178     97C          K1=K+1
179     982          CALL NORM(AE,X,NC)
180     986          CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
181     98A          CALL NORM(AE1,Y ,NC)
182     98E          ALPHA=SQRT(Y/X)
183     996          PRINT 82,ALPHA
184     9A6          BETA=ALPHA
185     9AA          CALL MUL2(AE,ALPHA,AD,NC)
186     9AE          ALPHA=1/ALPHA
```

```
150     818          DO       9I=1,NC
151     820          DO       9J=1,NC
152     828     9    AE(I,J)    =S(I,J)
153     864          CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
154     868          CALL NORM(AE,X,NC)
155     86C          CALL NORM(AE1,Y,NC)
156     870          ALPHA=SORT(Y/X)
157     878          PRINT 82,ALPHA
158     888          BETA=ALPHA
159     88C          CALL MUL2(AE,ALPHA,AD,NC)
160     890          ALPHA=1/ALPHA
161     898          CALL MUL2(AE1,ALPHA,AD1,NC)
162     89C          CALL SOMMAT4AD,AD1,AEK1,NC)
163     8A0          CALL MUL2(AEK1,0.5,AEK1,NC)
164     8A4          BETA=BETA-1
165     8B8          BETA=ABS(BETA)
166     8BE          DO 84 I=1,NC
167     8C6          DO 83 J=1 NC
158     8CE          AEK2(I,J)=AEK1(I,J)-AE(I,J)
169     8FC          AE(I,J)=AEK1(I,J)
170     91C     83   CONTINUE
171     92C     84   CONTINUE
172     93A          CALL NORM(AEK2,W,NC)
173     93E          GAMMA=W/X
174     944          IF(BETA.GT.GAMMA) TEST(1)=BETA
175     960          TEST(1)=GAMMA
176     966          TEST(1)=TEST(1)/2
177     976          DO 10 K=1,N1
178     97C          K1=K+1
179     982          CALL NORM(AE,X,NC)
180     986          CALL MRINV(AE,AE1,NC,KOD,DET,EPS,IL,IC)
181     98A          CALL NORM(AE1,Y ,NC)
182     98E          ALPHA=SQRT(Y/X)
183     996          PRINT 82,ALPHA
184     9A6          BETA=ALPHA
185     9AA          CALL MUL2(AE,ALPHA,AD,NC)
186     9AE          ALPHA=1/ALPHA
```

```
187     9B6         CALL MUL2(AE1,ALPHA,AD1,NC)
188     9BA         CALL SOMNAT(AD,AD1,AEK1,NC)
189     9BE         CALL MUL2(AEK1,0.5,AEK1,NC)
190     9C2         BETA=BETA-1
191     9D6         BETA=ABS(BETA)
192     9DC         DO 85 I=1,NC
193     9E4         DO 86 J=1,NC
194     9EC         AEK2(I,J)=AEK1(I,J)-AE(I,J)
195     A1A         AE(I,J)=AEK1(I,J)
196     A3A   86    CONTINUE
197     A4A   85    CONTINUE
198     A58         CALL NORM(AEK2,W,NC)
199     A           GAMMA=W/X
200     A62         IF(BETA.GT.GAMMA)TEST(K1)=BETA
201     A86         TEST(K1)=GAMMA
202     A94         IF(TEST(K1).LT.ETA1)GOTO 33
203     AB4
204     AB6   33    IF(TEST(K1).GT.TEST(K1-1)) GOTE 35
205     AF6         GOTO 34
206     AEA   34    TEST(K1)=TEST(K1)/2
207     B10   10    CONTINUE
208     B1C   35    CONTINUE
209     B1E         DO 11 I=1,NC
210     B26         DO 11 J=1,NC
211     B2E   11    SIGNE(I,J)=AEK1(I,J)
212     B6A         PRINT 1101
213     B74         PRINT 71,((SIGNE(I,J),J=1,NC),I=1,NC)
214     BC0         DO 3 I=1,N
215     BC8         DO 3 J=1,N
216     BD0         I1=I+N
217     BDA         J1=J+N
218     BE4         IE(I1,J1)=0.
219     BF6         IF(I1.EQ.J1)IE(I1,J1)=1.
220     C1F    3    CONTINUE
221     C3A         DO 2 I=1,N
222     C42         DO 2 J=1,N
```

```
187     9B6         CALL MUL2(AE1,ALPHA,AD1,NC)
188     9BA         CALL SOMNAT(AD,AD1,AEK1,NC)
189     9BE         CALL MUL2(AEK1,0.5,AEK1,NC)
190     9C2         BETA=BETA-1
191     9D6         BETA=ABS(BETA)
192     9DC         DO 85 I=1,NC
193     9E4         DO 86 J=1,NC
194     9EC         AEK2(I,J)=AEK1(I,J)-AE(I,J)
195     A1A         AE(I,J)=AEK1(I,J)
196     A3A   86.   CONTINUE
197     A4A   85    CONTINUE
198     A58         CALL NORM(AEK2,W,NC)
199     A.          GAMMA=W/X
200     A62         IF(BETA.GT.GAMMA)TEST(K1)=BETA
201     A86         TEST(K1)=GAMMA
202     A94         IF(TEST(K1).LT.ETA1)GOTO 33
203     AB4
204     AB6   33    IF(TEST(K1).GT.TEST(K1-1)) GOTE 35
205     AF6         GOTO 34
206     AEA   34    TEST(K1)=TEST(K1)/2
207     B10   10    CONTINUE
208     B1C   35    CONTINUE
209     B1E         DO 11 I=1,NC
210     B26         DO 11 J=1,NC
211     B2E   11    SIGNE(I,J)=AEK1(I,J)
212     B6A         PRINT 1101
213     B74         PRINT 71,((SIGNE(I,J),J=1,NC),I=1,NC)
214     BC0         DO 3 I=1,N
215     BC8         DO 3 J=1,N
216     BD0         I1=I+N
217     BDA         J1=J+N
218     BE4         IE(I1,J1)=0.
219     BF6         IF(I1.EQ.J1)IE(I1,J1)=1.
220     C1F    3    CONTINUE
221     C3A         DO 2 I=1,N
222     C42         DO 2 J=1,N
```

```
223     C4A          IE(I,J)=0.
224     C5C          IF(I.EQ.J)IE(I,J)=-1.
225     C86    2     CONTINUE
226     CA2          DO 20 I=1,N
227     CAA          DO 29 J=1,N
228     CB2          I1=I+N
229     CBC          IE(I1,J)=0.
230     CD2    29    CONTINUE
231     CE2    20    CONTINUE
232     CF0          DO 1500 I=1,N
233     CF8          DO 1501 J=1,N
234     D00          J1=J+N
235     D0A          IF(I,J1)=0.
236     D1C    1501  CONTINUE
237     D2C    1500  CONTINUE
238     D3A          CALL SOMMAT(SIGNE,IE,F,NC)
239     D3E          PRINT 120
240     D48          PRINT 71,((F(I,J),J=1,NC),I=1,NC)
241     D94          DO 12 I=1,N
242     D9C          DO 12 J=1,N
243     DA4          I1=I+N
244     DAE          F1(I,J)=F(I,J)
245     DD0    12    F21(I,J)=F(I1,I)
246     E12          CALL MRINV(F1,FF1,N,KOD,DET,EPS,IL,IC)
247     E16          CALL PROD(F21,FF1,F,N)
248     E1A          PRINT 140
249     E24          RINT 81,((P(I,J),J=1,N),I=1,N)
250     E6E          DO 56 I=1,N
251     E76          DO 57 J=1,N
252     E7E          PT(I,J)=F(J,I)
253     EA0    57    CONTINUE
254     EB0    56    CONTINUE
255     EBE          CALL SOMMAT P,PT,PC,N)
256     EC2          CALL MUL 2 (PC,0.5,PC,N)
257     EC6          PRINT 700
258     ED0          PRINT 81, ((PC(I,J), j=1,N),I=1,N)
259     F1A    71    FORMAT ( ///,10(1x,F12.5))
260     F1A    81    FORMAT (///,5(4xF12.5))
261     F1A    90    FORMAT  ////,10x,MATRICE S ';//)
```

```
223     C4A          IE(I,J)=0.
224     C5C          IF(I.EQ.J)IE(I,J)=-1.
225     C86    2     CONTINUE
226     CA2          DO 20 I=1,N
227     CAA          DO 29 J=1,N
228     CB2          I1=I+N
229     CBC          IE(I1,J)=0.
230     CD2    29    CONTINUE
231     CE2    20    CONTINUE
232     CF0          DO 1500 I=1,N
233     CF8          DO 1501 J=1,N
234     D00          J1=J+N
235     D0A          IF(I,J1)=0.
236     D1C    1501  CONTINUE
237     D2C    1500  CONTINUE
238     D3A          CALL SOMMAT(SIGNE,IE,F,NC)
239     D3E          PRINT 120
240     D48          PRINT 71,((F(I,J),J=1,NC),I=1,NC)
241     D94          DO 12 I=1,N
242     D9C          DO 12 J=1,N
243     DA4          I1=I+N
244     DAE          F1(I,J)=F(I,J)
245     DD0    12    F21(I,J)=F(I1,I)
246     E12          CALL MRINV(F1,FF1,N,KOD,DET,EPS,IL,IC)
247     E16          CALL PROD(F21,FF1,F,N)
248     E1A          PRINT 140
249     E24          RINT 81,((P(I,J),J=1,N),I=1,N)
250     E6E          DO 56 I=1,N
251     E76          DO 57 J=1,N
252     E7E          PT(I,J)=F(J,I)
253     E80    57    CONTINUE
254     E80    56    CONTINUE
255     E8E          CALL SOMMAT(P,PT,PC,N)
256     EC2          CALL MUL 2 (PC,0.5,PC,N)
257     EC6          PRINT 700
258     ED0          PRINT 81, ((PC(I,J), j=1,N),I=1,N)
259     F1A    71    FORMAT (///,10(1x,F12.5))
260     F1A    81    FORMAT (///,5(4xF12.5))
261     F1A    90    FORMAT  ////,10x,MATRICE S ';//)
```

```
262     F1A     1101    FORMAT (////,10x, MATRICE SIGNE.'//)
263     F1A      150    FORMAT (////,10x, MATRICE L')
264     F1A      160    FORMAT (////,10x , MATRICE L')
265     F1A       41    FORMAT (////,5 (10x,F2.0)
266     F1A       60    FORMAT (////,10x' MATRICE R. ' //)
267     F1A       51    FORMAT (////, 5 (10x,F6.3))
268     F1AA      40    FORMAT (////, 10x, MATRICE B. '  //)
269     F1A       30    FORMAT (////,10x'MATRICE A.'//)
270     F1A       31    FORMAT (////,5 (10x,F7.4)
271     F1A      700    FORMAT (1H,////,10x,' MATRICE PC.' //)
272     F1A      140    FORMAT (1H,////,10x,' MATRICE P.' //)
273     F1A      120    FORMAT (1H,////,10, MATRICE F.' //)
274     F1A      110    FORMAT (////,20 x ' NOMBRE D'ITERATIONS = ' ,12)
275     F1A       82    FORMAT (///,10x ' ALPHA = ', F.12.5,/)
276     F1        100   FORMAT (1 H, ////,20 x, NORME DE S = ', F12.5)
277     F1A             STOP
278     F1C             END
```

```
262     F1A     1101    FORMAT (////,10x, MATRICE SIGNE.'//)
263     F1A      150    FORMAT (////,10x, MATRICE L')
264     F1A      160    FORMAT (////10x , MATRICE L')
265     F1A       41    FORMAT (////,5 (10x,F2.0)
266     F1A       60    FORMAT (////,10x' MATRICE R. ' //)
267     F1A       51    FORMAT (////, 5 (10x,F6.3))
268     F1AA      40    FORMAT (////, 10x, MATRICE B. '  //)
269     F1A       30    FORMAT (////,10x'MATRICE A.'//)
270     F1A       31    FORMAT (////,5 (10x,F7.4)
271     F1A      700    FORMAT (1H,////,10x,' MATRICE PC.' //)
272     F1A      140    FORMAT (1H,////,10x,' MATRICE P.' //)
273     F1A      120    FORMAT (1H,////,10, MATRICE F.' //)
274     F1A      110    FORMAT (////,20 x ' NOMBRE D'ITERATIONS = ' ,12)
275     F1A       82    FORMAT (///,10x ' ALPHA = ', F.12.5,/)
276     F1A      100    FORMAT (1 H, ////,20 x, NORME DE S = ', F12.5)
277     F1A             STOP
278     F1C             END
```

BIBLIOGRAPHIE

Cours d'automatique avancée :Optimalisation déterministe de la commande PAUL LEFEVRE(école nationale supérieure des techniques avancées)

The matrix sign function and computations in systems(EUGENE DENMAN and ALEX N.BEAVERS,JR) Applied mathematics and computations 2,63-94(1976)

3 J.L.CASTI,Dynamical systems and their applications: linear théory, Académic Press 1977.

4 B.D.O.ANDERSON,Second ordr convergent algorithms for the steadystate Riccati equation,Paper FA6-11:00,1977 I.E.E.E

5 B.D.O.ANDERSON,Second ordr convergent algorithms for the steadystate Riccati equation,Int.J.Contr.,1978,

A.BARRAUD,Investigation autour de la fonction signe d'une matrice,application à l'équation de Riccati,RAIRO automatique,n°4 1979,p335-368

7 G.A.HEWER,An Iterative Technique for the computations of the steadystate Gain for the DiscreteOptimal Regulator,I.E.E.E vol.AC-16,n°4,1971

8 A.BARRAUD,Produit étoile et fonction signe de matrice,application à l'équation de Riccati dans le cas discret,RAIRO automatique/Systems Analysis and Control(vol.14,n°1,1980.p.55 à 85)

10 D.R.VAUGHAN.A Non Recursive Algebraic Solution for the Discrete Riccati Equation I.E.E.E,vol,AC-15,n°5 1970.

11 A.J.LAUB,A Schur Method for Solving Algebraix Riccati Equations,I.E.E.E 1978

12 B.T.SMITH, Matrix Eigen-System Routines.EISPACK Guide,vol 6, 1976

13 G.W.STEWART,HQR and Exchng Fortran Subroutines vol.2,1976.

15 C.L.LAWSON et R.J.HANSON,Solving Least Squares Problems,Prentice Hall 1974.

GILES.P , Systèmes asservis linéaires

BIBLIOGRAPHIE

Cours d'automatique avancée :Optimalisation déterministe de la commande PAUL LEFEVRE(école nationale supérieure des techniques avancées)

The matrix sign function and computations in systems(EUGENE DENMAN and ALEX N.BEAVERS,JR) Applied mathematics and computations 2,63-94(1976)

3 J.L.CASTI,Dynamical systems and their applications: linear théory, Académic Press 1977.

4 B.D.O.ANDERSON,Second ordr convergent algorithms for the steadystate Riccati equation,Paper FA6-11:00,1977 I.E.E.E

5 B.D.O.ANDERSON,Second ordr convergent algorithms for the steadystate Riccati equation,Int.J.Contr.,1978,

A.BARRAUD,Investigation autour de la fonction signe d'une matrice,application à l'équation de Riccati,RAIRO automatique,n°4 1979,p335-368

7 G.A.HEWER,An Iterative Technique for the computations of the steadystate Gain for the DiscreteOptimal Regulator,I.E.E.E vol.AC-16,n°4,1971

8 A.BARRAUD,Produit étoile et fonction signe de matrice,application à l'équation de Riccati dans le cas discret,RAIRO automatique/Systems Analysis and Control(vol.14,n°1,1980.p.55 à 85)

10 D.R.VAUGHAN.A Non.Recursive Algebraic Solution for the Discrete Riccati Equation I.E.E.E,vol,AC-15,n°5 1970.

11 A.J.LAUB,A Schur Method for Solving Algebraix Riccati Equations,I.E.E.E 1978

12 B.T.SMITH, Matrix Eigen-System Routines.EISPACK Guide,vol 6, 1976

13 G.W.STEWART,HQR and Exchng Fortran Subroutines vol.2,1976.

15 C.L.LAWSON et R.J.HANSON,Solving Least Squares Problems,Prentice Hall 1974.

GILES.P , Systèmes asservis linéaires