# Logiciel numérique associé à une modélisation de systèmes informatiques

François Cachard

# THESE

*présentée à*

**l' Université Scientifique et Médicale de Grenoble**

*et à*

**l' Institut National Polytechnique de Grenoble**

*pour obtenir le grade de*

DOCTEUR INGENIEUR

Génie Informatique

*par*

François CACHARD

# LOGICIEL NUMERIQUE ASSOCIE A UNE MODELISATION DE SYSTEMES INFORMATIQUES.

Thèse soutenue le 24 septembre 1981 devant la Commission d'Examen :

| | | |
|---|---|---|
| Monsieur | G. VEILLON | Président |
| Madame | F. CHATELIN | Examinateurs |
| Messieurs | M. VERAN | |
| | F. ROBERT | |

## UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Monsieur Gabriel CAU : Président
Monsieur Joseph KLEIN : Vice-Président

---

## MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

### PROFESSEURS TITULAIRES

| | | |
|---|---|---|
| MM. | AMBLARD Pierre | Clinique de dermatologie |
| | ARNAUD Paul | Chimie |
| | ARVIEU Robert | I.S.N. |
| | AUBERT Guy | Physique |
| | AYANT Yves | Physique approfondie |
| Mme | BARBIER Marie-Jeanne | Electrochimie |
| MM. | BARBIER Jean-Claude | Physique expérimentale |
| | BARBIER Reynold | Géologie appliquée |
| | BARJON Robert | Physique nucléaire |
| | BARNOUD Fernand | Biosynthèse de la cellulose |
| | BARRA Jean-René | Statistiques |
| | BARRIE Joseph | Clinique chirurgicale A |
| | BEAUDOING André | Clinique de pédiatrie et puériculture |
| | BELORIZKY Elie | Physique |
| | BARNARD Alain | Mathématiques pures |
| Mme | BERTRANDIAS Françoise | Mathématiques pures |
| MM. | BERTRANDIAS Jean-Paul | Mathématiques pures |
| | BEZES Henri | Clinique chirurgicale et traumatologie |
| | BLAMBERT Maurice | Mathématiques pures |
| | BOLLIET Louis | Informatique (I.U.T. B) |
| | BONNET Jean-Louis | Clinique ophtalmologie |
| | BONNET-EYMARD Joseph | Clinique hépato-gastro-entérologie |
| Mme | BONNIER Marie-Jeanne | Chimie générale |
| MM. | BOUCHERLE André | Chimie et toxicologie |
| | BOUCHEZ Robert | Physique nucléaire |
| | BOUSSARD Jean-Claude | Mathématiques appliquées |
| | BOUTET DE MONVEL Louis | Mathématiques pures |
| | BRAVARD Yves | Géographie |
| | CABANEL Guy | Clinique rhumatologique et hydrologique |
| | CALAS François | Anatomie |
| | CARLIER Georges | Biologie végétale |
| | CARRAZ Gilbert | Biologie animale et pharmacodynamie |

.../...

| | | |
|---|---|---|
| MM. | CAU Gabriel | Médecine légale et toxicologie |
| | CAUQUIS Georges | Chimie organique |
| | CHABAUTY Claude | Mathématiques pures |
| | CHARACHON Robert | Clinique ot-rhino-laryngologique |
| | CHATEAU Robert | Clinique de neurologie |
| | CHIBON Pierre | Biologie animale |
| | COEUR André | Pharmacie chimique et chimie analytique |
| | COUDERC Pierre | Anatomie pathologique |
| | DEBELMAS Jacques | Géologie générale |
| | DEGRANGE Charles | Zoologie |
| | DELORMAS Pierre | Pneumophtisiologie |
| | DEPORTES Charles | Chimie minérale |
| | DESRE Pierre | Métallurgie |
| | DODU Jacques | Mécanique appliquée (I.U.T. I) |
| | DOLIQUE Jean-Michel | Physique des plasmas |
| | DREYFUS Bernard | Thermodynamique |
| | DUCROS Pierre | Cristallographie |
| | FONTAINE Jean-Marc | Mathématiques pures |
| | GAGNAIRE Didier | Chimie physique |
| | GALVANI Octave | Mathématiques pures |
| | GASTINEL Noël | Analyse numérique |
| | GAVEND Michel | Pharmacologie |
| | GEINDRE Michel | Electroradiologie |
| | GERBER Robert | Mathématiques pures |
| | GERMAIN Jean-Pierre | Mécanique |
| | GIRAUD Pierre | Géologie |
| | JANIN Bernard | Géographie |
| | KAHANE André | Physique générale |
| | KLEIN Joseph | Mathématiques pures |
| | KOSZUL Jean-Louis | Mathématiques pures |
| | KRAVTCHENKO Julien | Mécanique |
| | LACAZE Albert | Thermodynamique |
| | LACHARME Jean | Biologie végétale |
| Mme | LAJZEROWICZ Janine | Physique |
| MM. | LAJZEROWICZ Joseph | Physique |
| | LATREILLE René | Chirurgie générale |
| | LATURAZE Jean | Biochimie pharmaceutique |
| | LAURENT Pierre | Mathématiques appliquées |
| | LEDRU Jean | Clinique médicale B |
| | LE ROY Philippe | Mécanique (I.U.T. I) |

.../...

| | | |
|---|---|---|
| MM. | LLIBOUTRY Louis | Géophysique |
| | LOISEAUX Jean-Marie | Sciences nucléaires |
| | LONGEQUEUE Jean-Pierre | Physique nucléaire |
| | LOUP Jean | Géographie |
| Mlle | LUTZ Elisabeth | Mathématiques pures |
| MM. | MALINAS Yves | Clinique obstétricale |
| | MARTIN-NOEL Pierre | Clinique cardiologique |
| | MAYNARD Roger | Physique du solide |
| | MAZARE Yves | Clinique Médicale A |
| | MICHEL Robert | Minéralogie et pétrographie |
| | MICOUD Max | Clinique maladies infectieuses |
| | MOURIQUAND Claude | Histologie |
| | MOUSSA André | Chimie nucléaire |
| | NEGRE Robert | Mécanique |
| | NOZIERES Philippe | Spectrométrie physique |
| | OZENDA Paul | Botanique |
| | PAYAN Jean-Jacques | Mathématiques pures |
| | PEBAY-PEYROULA Jean-Claude | Physique |
| | PERRET Jean | Séméiologie médicale (neurologie) |
| | RASSAT André | Chimie systématique |
| | RENARD Michel | Thermodynamique |
| | REVOL Michel | Urologie |
| | RINALDI Renaud | Physique |
| | DE ROUGEMONT Jacques | Neuro-Chirurgie |
| | SARRAZIN Roger | Clinique chirurgicale B |
| | SEIGNEURIN Raymond | Microbiologie et hygiène |
| | SENGEL Philippe | Zoologie |
| | SIBILLE Robert | Construction mécanique (I.U.T. I) |
| | SOUTIF Michel | Physique générale |
| | TANCHE Maurice | Physiologie |
| | VAILLANT François | Zoologie |
| | VALENTIN Jacques | Physique nucléaire |
| Mme | VERAIN Alice | Pharmacie galénique |
| MM. | VERAIN André | Physique biophysique |
| | VEYRET Paul | Géographie |
| | VIGNAIS Pierre | Biochimie médicale |

## PROFESSEURS ASSOCIES

| | | |
|---|---|---|
| MM. | CRABBE Pierre | CERMO |
| | SUNIER Jules | Physique |

## PROFESSEURS SANS CHAIRE

| | | |
|---|---|---|
| Mlle | AGNIUS-DELORS Claudine | Physique pharmaceutique |
| | ALARY Josette | Chimie analytique |
| MM. | AMBROISE-THOMAS Pierre | Parasitologie |
| | ARMAND Gilbert | Géographie |
| | BENZAKEN Claude | Mathématiques appliquées |
| | BIAREZ Jean-Pierre | Mécanique |
| | BILLET Jean | Géographie |
| | BOUCHET Yves | Anatomie |
| | BRUGEL Lucien | Energétique (I.U.T. I) |
| | BUISSON René | Physique (I.U.T. I) |
| | BUTEL Jean | Orthopédie |
| | COHEN-ADDAD Jean-Pierre | Spectrométrie physique |
| | COLOMB Maurice | Biochimie médicale |
| | CONTE René | Physique (I.U.T. I) |
| | DELOBEL Claude | M.I.A.G. |
| | DEPASSEL Roger | Mécanique des fluides |
| | GAUTRON René | Chimie |
| | GIDON Paul | Géologie et minéralogie |
| | GLENAT René | Chimie organique |
| | GROULADE Joseph | Biochimie médicale |
| | HACQUES Gérard | Calcul numérique |
| | HOLLARD Daniel | Hématologie |
| | HUGONOT Robert | Hygiène et médecine préventive |
| | IDELMAN Simon | Physiologie animale |
| | JOLY Jean-René | Mathématiques pures |
| | JULLIEN Pierre | Mathématiques appliquées |
| Mme | KAHANE Josette | Physique |
| MM. | KRAKOWIACK Sacha | Mathématiques appliquées |
| | KUHN Gérard | Physique (I.U.T. I) |
| | LUU DUC Cuong | Chimie organique - pharmacie |
| | MICHOULIER Jean | Physique (I.U.T. I) |
| Mme | MINIER Colette | Physique (I.U.T. I) |

.../...

| | | |
|---|---|---|
| MM. | PELMONT Jean | Biochimie |
| | PERRIAUX Jean-Jacques | Géologie et minéralogie |
| | PFISTER Jean-Claude | Physique du solide |
| Mlle | PIERY Yvette | Physiologie animale |
| MM. | RAYNAUD Hervé | M.I.A.G. |
| | REBECQ Jacques | Biologie (CUS) |
| | REYMOND Jean-Charles | Chirurgie générale |
| | RICHARD Lucien | Biologie végétale |
| Mme | RINAUDO Marguerite | Chimie macromoléculaire |
| MM. | SARROT-REYNAULD Jean | Géologie |
| | SIROT Louis | Chirurgie générale |
| Mme | SOUTIF Jeanne | Physique générale |
| MM. | STIEGLITZ Paul | Anesthésiologie |
| | VIALON Pierre | Géologie |
| | VAN CUTSEM Bernard | Mathématiques appliquées |

## MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

| | | |
|---|---|---|
| MM. | ARMAND Yves | Chimie (I.U.T. I) |
| | BACHELOT Yvan | Endocrinologie |
| | BARGE Michel | Neuro-chirurgie |
| | BEGUIN Claude | Chimie organique |
| Mme | BERIEL Hélène | Pharmacodynamie |
| MM. | BOST Michel | Pédiatrie |
| | BOUCHARLAT Jacques | Psychiatrie adultes |
| Mme | BOUCHE Liane | Mathématiques (CUS) |
| MM. | BRODEAU François | Mathématiques (I.U.T. B) (Personne étrangère habilitée à être directeur de thèse) |
| | BERNARD Pierre | Gynécologie |
| | CHAMBAZ Edmond | Biochimie médicale |
| | CHAMPETIER Jean | Anatomie et organogénèse |
| | CHARDON Michel | Géographie |
| | CHERADAME Hervé | Chimie papetière |
| | CHIAVERINA Jean | Biologie appliquée (EFP) |
| | COLIN DE VERDIERE Yves | Mathématiques pures |
| | CONTAMIN Charles | Chirurgie thoracique et cardio-vasculaire |
| | CORDONNER Daniel | Néphrologie |
| | COULOMB Max | Radiologie |
| | CROUZET Guy | Radiologie |

.../...

| | | |
|---|---|---|
| MM. | CYROT Michel | Physique du solide |
| | DENIS Bernard | Cardiologie |
| | DOUCE Roland | Physiologie végétale |
| | DUSSAUD René | Mathématiques (CUS) |
| Mme | ETERRADOSSI Jacqueline | Physiologie |
| MM. | FAURE Jacques | Médecine légale |
| | FAURE Gilbert | Urologie |
| | GAUTIER Robert | Chirurgie générale |
| | GIDON Maurice | Géologie |
| | GROS Yves | Physique (I.U.T. I) |
| | GUIGNIER Michel | Thérapeutique |
| | GUITTON Jacques | Chimie |
| | HICTER Pierre | Chimie |
| | JALBERT Pierre | Histologie |
| | JUNIEN-LAVILLAVROY Claude | O.R.L. |
| | KOLODIE Lucien | Hématologie |
| | LE NOC Pierre | Bactériologie-virologie |
| | MACHE Régis | Physiologie végétale |
| | MAGNIN Robert | Hygiène et médecine préventive |
| | MALLION Jean-Michel | Médecine du travail |
| | MARECHAL Jean | Mécanique (I.U.T. I) |
| | MARTIN-BOUYER Michel | Chimie (CUS) |
| | MASSOT Christian | Médecine interne |
| | NEMOZ Alain | Thermodynamique |
| | NOUGARET Marcel | Automatique (I.U.T. I) |
| | PARAMELLE Bernard | Pneumologie |
| | PECCOUD François | Analyse (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse) |
| | PEFFEN René | Métallurgie (I.U.T. I) |
| | PERRIER Guy | Géophysique-glaciologie |
| | PHELIP Xavier | Rhumatologie |
| | RACHALL Michel | Médecine interne |
| | RACINET Claude | Gynécologie et obstétrique |
| | RAMBAUD Pierre | Pédiatrie |
| | RAPHAEL Bernard | Stomatologie |
| Mme | RENAUDET Jacqueline | Bactériologie (pharmacie) |
| MM. | ROBERT Jean-Bernard | Chimie-physique |
| | ROMIER Guy | Mathématiques (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse) |
| | SAKAROVITCH Michel | Mathématiques appliquées |

.../...

| | | |
|---|---|---|
| MM. | SCHAERER René | Cancérologie |
| Mme | SEIGLE-MURANDI Françoise | Crytogamie |
| MM. | STOEBNER Pierre | Anatomie pathologie |
| | STUTZ Pierre | Mécanique |
| | VROUSOS Constantin | Radiologie |

## MAITRES DE CONFERENCES ASSOCIES

| | | |
|---|---|---|
| MM. | DEVINE Roderick | Spectro Physique |
| | KANEKO Akira | Mathématiques pures |
| | JOHNSON Thomas | Mathématiques appliquées |
| | RAY Tuhina | Physique |

## MAITRE DE CONFERENCES DELEGUE

| | | |
|---|---|---|
| M. | ROCHAT Jacques | Hygiène et hydrologie (pharmacie) |

Fait à Saint Martin d'Hères, novembre 1977

# INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

**Année universitaire 1979-1980**

**Président** : M. Philippe TRAYNARD

**Vice-Présidents** : M. Georges LESPINARD
M. René PAUTHENET

---

## PROFESSEURS DES UNIVERSITES

| | | |
|---|---|---|
| MM. | ANCEAU François | Informatique fondamentale et appliquée |
| | BENOIT Jean | Radioélectricité |
| | BESSON Jean | Chimie Minérale |
| | BLIMAN Samuel | Electronique |
| | BLOCH Daniel | Physique du Solide - Cristallographie |
| | BOIS Philippe | Mécanique |
| | BONNETAIN Lucien | Génie Chimique |
| | BONNIER Etienne | Métallurgie |
| | BOUVARD Maurice | Génie Mécanique |
| | BRISSONNEAU Pierre | Physique des Matériaux |
| | BUYLE-BODIN Maurice | Electronique |
| | CHARTIER Germain | Electronique |
| | CHERADAME Hervé | Chimie Physique Macromoléculaires |
| Mme | CHERUY Arlette | Automatique |
| MM. | CHIAVERINA Jean | Biologie, Biochimie, Agronomie |
| | COHEN Joseph | Electronique |
| | COUMES André | Electronique |
| | DURAND Francis | Métallurgie |
| | DURAND Jean-Louis | Physique Nucléaire et Corpusculaire |
| | FELICI Noël | Electrotechnique |
| | FOULARD Claude | Automatique |
| | GUYOT Pierre | Métallurgie Physique |
| | IVANES Marcel | Electrotechnique |
| | JOUBERT Jean-Claude | Physique du Solide - Cristallographie |
| | LACOUME Jean-Louis | Géographie - Traitement du Signal |
| | LANCIA Roland | Electronique - Automatique |
| | LESIEUR Marcel | Mécanique |
| | LESPINARD Georges | Mécanique |
| | LONGEQUEUE Jean-Pierre | Physique Nucléaire Corpusculaire |
| | MOREAU René | Mécanique |
| | MORET Roger | Physique Nucléaire Corpusculaire |
| | PARIAUD Jean-Charles | Chimie - Physique |
| | PAUTHENET René | Physique du Solide - Cristallographie |
| | PERRET René | Automatique |

.../...

| | | |
|---|---|---|
| MM. | PERRET Robert | Electrotechnique |
| | PIAU Jean-Michel | Mécanique |
| | PIERRARD Jean-Marie | Mécanique |
| | POLOUJADOFF Michel | Electrotechnique |
| | POUPOT Christian | Electronique - Automatique |
| | RAMEAU Jean-Jacques | Chimie |
| | ROBERT André | Chimie Appliquée et des matériaux |
| | ROBERT François | Analyse numérique |
| | SABONNADIERE Jean-Claude | Electrotechnique |
| Mme | SAUCIER Gabrielle | Informatique fondamentale et appliquée |
| M. | SOHM Jean-Claude | Chimie - Physique |
| Mme | SCHLENKER Claire | Physique du Solide - Cristallographie |
| MM. | TRAYNARD Philippe | Chimie - Physique |
| | VEILLON Gérard | Informatique fondamentale et appliquée |
| | ZADWORNY François | Electronique |

**CHERCHEURS DU C.N.R.S.** (Directeur et Maître de Recherche)

| | | |
|---|---|---|
| M. | FRUCHART Robert | Directeur de Recherche |
| MM. | ANSARA Ibrahïm | Maître de Recherche |
| | BRONOEL Guy | Maître de Recherche |
| | CARRE René | Maître de Recherche |
| | DAVID René | Maître de Recherche |
| | DRIOLE Jean | Maître de Recherche |
| | KAMARINOS Georges | Maître de Recherche |
| | KLEITZ Michel | Maître de Recherche |
| | LANDAU Ioan-Doré | Maître de Recherche |
| | MERMET Jean | Maître de Recherche |
| | MUNIER Jacques | Maître de Recherche |

**Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)**

**E.N.S.E.E.G.**

| | | |
|---|---|---|
| MM. | ALLIBERT Michel | |
| | BERNARD Claude | |
| | CAILLET Marcel | |
| Mme | CHATILLON Catherine | |
| MM. | COULON Michel | |
| | HAMMOU Abdelkader | |
| | JOUD Jean-Charles | |
| | RAVAINE Denis | |
| | SAINFORT | C.E.N.G. |

.../...

MM. SARRAZIN Pierre
SOUQUET Jean-Louis
TOUZAIN Philippe
URBAIN Georges — Laboratoire des Ultra-Réfractaires ODEILLO

**E.N.S.M.E.E.**

MM. BISCONDI Michel
BOOS Jean-Yves
GUILHOT Bernard
KOBILANSKI André
LALAUZE René
LANCELOT Françis
LE COZE Jean
LESBATS Pierre
SOUSTELLE Michel
THEVENOT François
THOMAS Gérard
TRAN MINH Canh
DRIVER Julian
RIEU Jean

**E.N.S.E.R.G.**

MM. BOREL Joseph
CHEHIKIAN Alain
VIKTOROVITCH Pierre

**E.N.S.I.E.G.**

MM. BORNARD Guy
DESCHIZEAUX Pierre
GLANGEAUD François
JAUSSAUD Pierre
Mme JOURDAIN Geneviève
MM. LEJEUNE Gérard
PERARD Jacques

**E.N.S.H.G.**

M. DELHAYE Jean-Marc

**E.N.S.I.M.A.G.**

MM. COURTIN Jacques
LATOMBE Jean-Claude
LUCAS Michel
VERDILLON André

*Je tiens à exprimer ma reconnaissance à :*

*Madame F. CHATELIN pour sa bienveillance, sa disponibilité et ses encouragements au cours de l'élaboration de ce travail,*

*Monsieur G. VEILLON pour avoir accepté de présider ce jury,*

*Monsieur M.VERAN qui a constamment suivi la progression de ce travail, en particulier dans son aspect informatique,*

*Monsieur F. ROBERT, pour ses conseils.*

*Je remercie également les membres de l'Equipe d'Analyse Numérique pour la bonne ambiance qui y règne et plus particulièrement Monsieur M. MERCIER.*

*Je tiens à remercier Cl. Meyrieux qui a effectué avec compétence et amabilité la frappe de cette thèse, ainsi que le service de reprographie pour la qualité de leur travail.*

# SOMMAIRE

pages

# INTRODUCTION

## I - PRESENTATION GENERALE. [1]

Les méthodes quantitatives d'évaluation des performances des systèmes informatiques sont usuellement divisées en deux classes :

- les mesures sur le système réel :
  - matérielles
  - logicielles
- les modèles de systèmes, probabilistes ou déterministes, dont la résolution peut être obtenue par :
  - simulation
  - des méthodes analytiques
  - des méthodes statistiques.

Si des mesures sur un système réel en donnent une image très fine et permettent de diagnostiquer des erreurs ou des blocages, elles ne peuvent en général, ni expliquer ni a fortiori prédire son fonctionnement. Par contre, le but même d'une expérience de modélisation est de donner une image relativement simple et interprétable du système étudié et d'en déduire des résultats sur son comportement pour différentes charges ou dans de nouvelles configurations.

Cependant, ces distinctions ne doivent pas faire oublier les liens étroits qui existent entre ces deux approches. Nous pouvons noter en particulier les suivants :

- Les mesures sont nécessaires pour fournir des données à un modèle quel qu'il soit,
- Un modèle peut aider à préciser les mesures à effectuer sur le système réel en déterminant les paramètres clés de son fonctionnement.

Pour notre part, nous ne nous intéressons dans ce mémoire qu'à l'aspect modélisation des systèmes informatiques, et plus spécialement aux modèles mis sous forme de réseaux de files d'attente. Ce type de modèle correspond à une vue le plus souvent assez naturelle d'un système informatique où les ressources (unité centrale, canaux, périphériques) sont représentées par des stations (dans notre terminologie une station est constituée de files d'attente associées à des serveurs) et où les différents utilisateurs de ces ressources (programmes, tâches, requêtes) définissent des populations de clients transitant à travers un réseau dont les noeuds sont ces stations. Il est clair que ce mode de représentation introduit une simplification du système réel qu'il reste à apprécier ponctuellement, et que nous ne traiterons pas ici.

Les qualités essentielles des modèles à réseaux de files d'attente, et qui peuvent expliquer leur popularité actuelle, sont les suivantes :

- fournir une bonne représentation des phénomènes globaux régissant les systèmes informatiques,
- ceci à l'aide d'un modèle relativement intuitif et simple à définir,
- pour laquelle les données sont le plus souvent accessibles par des mesures assez peu sophistiquées (probabilités de transition, taux de service, etc...).

Par ailleurs, les progrès réalisés ces dernières années en théorie des files d'attente permettent de disposer d'un large éventail de techniques de résolution, approchées ou exactes, qui offrent des domaines d'application complémentaires. Il paraît donc intéressant de regrouper au sein d'un même outil l'ensemble de ces techniques afin d'en faciliter la mise en oeuvre. C'est cet objectif qui a conduit à la réalisation du produit logiciel QNAP (Queueing Network Analysis Package) réalisé en collaboration entre l'IRIA, l'IRISA et la CII-HB.

La structure de cet outil et les domaines d'intérêt qui justifient son existence sont présentés dans le paragraphe II.

Le paragraphe III est consacré à une brève présentation des méthodes de résolution dans QNAP. Ces méthodes permettent d'obtenir l'état stationnaire des modèles étudiés.

Enfin, le paragraphe IV s'attachera à montrer les améliorations que l'on peut apporter à l'un de ces types de résolution : la méthode numérique.

## II - PRESENTATION GENERALE DE L'OUTIL [1] [2]

### 1) Objectif

Le but du QNAP est de mettre à la disposition des utilisateurs un outil de description et de résolution de modèles formulés en termes de réseaux de files d'attente, les caractéristiques de ces réseaux pouvant être très générales. Parmi les buts recherchés pour un tel outil, on peut citer :

* Réduction du travail de programmation (langage adapté aux structures de files d'attente, méthodes de résolution intégrées).

* Transparence des méthodes de résolution (l'utilisateur ne doit connaître que les conditions d'application de chacune des méthodes proposées).

* Diminution des coûts de résolution (choix de la méthode de résolution la plus adaptée).

* Approfondissement de la connaissance des propriétés des réseaux de files d'attente (études comparatives entre diverses méthodes de résolution : domaines d'application respectifs, précision, rapidité).

* Description et résolution de modèles complexes (mise en oeuvre de modélisations hiérarchisées). [3]

2) Fonctions

QNAP est un ensemble de programmes écrits en FORTRAN IV pour lequel les données sont introduites par l'intermédiaire d'un fichier ou d'un périphérique interactif. Un langage de spécification simple, commun à toutes les méthodes de résolution, permet de constituer ces données. Ce langage est utilisé pour décrire :

- La configuration du réseau : un réseau est constitué d'un ensemble de stations (serveurs et file d'attente) dans lequel transitent des clients selon un certain routage (transitions). Ces clients peuvent être répartis dans plusieurs classes définissant des comportements différents.

- Le traitement effectué par chaque station (service) : ce traitement peut être décrit soit par une durée aléatoire de distribution donnée, soit par un algorithme pouvant mettre en jeu des mécanismes complexes de synchronisation.

- Le programme contrôlant la résolution du réseau : initialisation et modification des paramètres, activation des modules de résolution, etc...

Les méthodes de résolution peuvent être divisées en trois classes :

- simulation à évènements discrets [4]
- résolutions mathématiques exactes
  - méthodes numériques d'analyse d'un modèle markovien [5]
  - méthodes analytiques utilisant les théorèmes généraux de Baskett-Chandy-Muntz et Palacios. [6]
- résolutions mathématiques approchées :
  - méthode itérative [7]
  - méthode de diffusion. [8]

Toutes ces méthodes permettent le calcul des grandeurs (taux d'utilisation, longueur moyenne de file d'attente, etc...) qui caractérisent à l'état stationnaire les stations qui composent le réseau. Ce sont ces méthodes qui vont être décrites dans le paragraphe suivant.

## III - LES METHODES DE RESOLUTION

### 1) Simulation [4]

Il s'agit ici de simulation de type "évènements discrets". Tout modèle décrit dans le langage peut être résolu par cette méthode. Les possibilités ainsi offertes dépassent largement le domaine classique des files d'attente et atteignent un degré de

généralité proche de celui fourni par les langages de simulation généraux (manipulation d'attributs associés aux stations ou aux clients, opérations de synchronisation, création dynamique d'objets,...).

Le contrôle de la précision des résultats obtenus par simulation est assuré pour les taux d'utilisation des serveurs. Pour ces grandeurs un intervalle de confiance est calculé par la méthode des blocs, et édité conjointement avec la grandeur.

2) Méthode analytique [6] [9]

Cette méthode est basée sur les théorèmes de Baskett-Chandy-Muntz et Palacios, qui considèrent des réseaux ouverts, fermés ou mixtes de files d'attente où le routage est décrit par une matrice de transition $P = (P_{ir;js})$ qui définit une chaîne de Markov du premier ordre ($P_{ir;js}$ est la probabilité constante pour qu'un client de classe r ayant achevé son service à la station i , se rende à la station j en passant en classe s). Les stations composant le réseau peuvent être de quatre types : FIFO, LIFO préemptif, "processor sharing" ou serveur infini. Pour les trois derniers types, les taux de service peuvent être différents pour chaque classe de clients.

Ces théorèmes permettent d'exprimer explicitement les probabilités stationnaires exactes, qui ne dépendent que des probabilités de transition et des taux de service, sans autre référence aux distributions. Les algorithmes de calcul correspondants sont efficaces et rapides.

3) Méthode numérique [10]

Ce type de méthodes s'applique dans tous les cas où le modèle est markovien du premier ordre à nombre fini d'états. Le principe consiste à construire la matrice de transition de

chaîne de Markov associée au modèle. L'avantage de ces méthodes réside dans le fait qu'elles permettent d'obtenir les probabilités exactes du vecteur distribution de probabilité à l'état stationnaire.

Ces trois types de méthodes offrent des domaines d'application très différents liés aux hypothèses faites dans le modèle (discipline d'attente, distribution des durées de service,...). Le problème de la méthode numérique implémentée dans QNAP est une vitesse d'exécution trop lente;c'est pourquoi on s'est attaché à améliorer cette méthode.

## IV - AMELIORATION DE LA METHODE NUMERIQUE

Parmi les méthodes décrites précédemment la méthode numérique est intéressante puisqu'elle peut être appliquée pratiquement quelles que soient les hypothèses faites sur le modèle. Les limites inhérentes à ce type de résolution proviennent en pratique de la multiplication des états de la chaîne de Markov dès que le modèle atteint une certaine complexité et aussi de la difficulté de résolution dûeà de mauvais conditionnements(prise en compte dans le modèle de phénomènes ayant des constantes de temps très différentes).
L'espace mémoire et le temps de calcul limitent alors la taille des modèles que l'on peut traiter à l'aide de ces méthodes.
Pour améliorer le logiciel implémenté dans QNAP on a, dans le Chapitre 1,présenté le problème numérique à résoudre;puis nous avons présenté les méthodes susceptibles de nous intéresser dans ce type de résolution. On verra qu'elles peuvent être classées en deux familles :

- des méthodes avec résolution de gros système
- des méthodes avec projection.

La fin de ce premier chapitre est consacrée à la présentation plus approfondie des trois méthodes qui nous ont semblé être les plus intéressantes.

Puis dans les Chapitres 2, 3 et 4 nous avons décrit les caractéristiques des logiciels associés à ces méthodes.

La comparaison des trois méthodes retenues est décrite au Chapitre 5. Elle tient compte des caractéristiques des logiciels (taille du logiciel, espace mémoire demandé,...) aussi bien que la valeur des résultats et du temps d'exécution nécessaire pour les obtenir.

# CHAPITRE 1

## PRÉSENTATION DU PROBLÈME THÉORIQUE
## ET
## MÉTHODES DE RÉSOLUTION

## I - PRESENTATION DU PROBLEME

L'analyse des réseaux de files d'attente (générés ici dans la modélisation de systèmes informatiques), conduit à une modélisation par une chaîne de Markov en temps continu avec un ensemble d'états discret (en nombre fini).

On connait $s_{ki}$ le taux de transition de l'état k à l'état i.

On appelle $P_i(t)$ la probabilité que le système soit à l'état i à l'instant $t$ .

On a la relation :

$$P_i(t+\delta_t) = P_i(t)[1 - \sum_{j\neq i}^{n} s_{ij}\delta_t] + [\sum_{k\neq i} s_{ki} P_k(t)]\delta_t$$

On pose :

$$s_{ii} = - \sum_{j\neq i}^{n} s_{ij}$$

$$P_i(t+\delta_t) = [\sum_{k=1}^{n} s_{ki} P_k(t)]\delta_t + P_i(t)$$

Soit $\overset{o}{P}_i(t) = \lim_{\delta t \to 0} \frac{P_i(t+\delta_t) - P_i(t)}{\delta_t}$

$$\overset{o}{P}_i(t) = \sum_{k=1}^{n} s_{ki} P_k(t) .$$

Ce qui donne en notation matricielle

$$\overset{o}{P}(t) = S^t P(t) .$$

à l'état stationnaire

$$\overset{o}{P}(t) = 0 \qquad S^t \overset{o}{P}(t) = 0 \qquad (1)$$

En substituant à l'une de ces lignes $\Sigma P_i = 1$ on obtient un système de Cramer. Mais la résolution du système pose un gros problème d'instabilité numérique. [cf. paragraphe II 1) a)]. On peut aborder le problème différemment :

$$(1) \quad \Longleftrightarrow \quad S^t \Delta_t P + P = P$$

d'où $$[S^t \Delta_t + I] P = P$$

donc $A^t P = P$ où $A^t = S^t \Delta_t + P$

On choisit $\Delta_t = \inf (- \frac{1}{s_{ii}})$ .

On a donc un problème de valeur propre, P étant alors le vecteur propre à gauche de A associé à la valeur propre 1, avec A matrice stochastique et creuse.

A étant une matrice stochastique, on sait que l'on a une seule valeur propre de module 1, qui est effectivement 1 , valeur propre simple ici (A étant irréductible puisque associée aux réseaux de files d'attente).

Posons $S = A^t$ , le problème revient donc à calculer le vecteur propre à droite de S associé à la valeur propre 1 (S a les mêmes propriétés que A).

On est ainsi passé d'un problème de résolution d'un système mal conditionné $[S^t \overset{o}{P}(t) = 0]$ à un problème de calcul du vecteur propre à droite associé à la valeur propre dominante 1.

## II - DIFFERENTES METHODES DE CALCUL DU VECTEUR PROPRE CHERCHE

### 1) Méthodes avec résolution de gros systèmes

#### a) Résolution du système

Il s'agit de résoudre un système homogène (cf. chapitre 5)

$$SP = 0$$

la matrice S étant une grande matrice creuse telle que
- la valeur propre zéro est simple
- la somme des éléments de chaque colonne est nulle.

On a donc :

$$\sum_j s_{ij} = 0 \quad , \quad \forall\, i$$

Un vecteur propre à gauche associé à la valeur propre zéro est le vecteur $\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$ . La dépendance entre les équations est donc que leur somme est nulle. Cette dépendance faisant intervenir toutes les équations, pour avoir un système de Cramer il suffit de remplacer n'importe quelle équation par $\sum_{i=1}^{n} P_i = 1$. Cette méthode présente le gros inconvénient d'être numériquement instable.

#### b) Itération inverse

Comme l'on connait la valeur propre dominante $\lambda_1 = 1$ c'est la méthode de recherche de vecteur propre la plus logique.

C'est une méthode de la puissance portant sur la matrice $B = (S-\mu I)^{-1}$ (si S est la matrice dont on cherche le vecteur propre associé à la valeur propre proche de $\mu$ ).

Rappel : Si $\lambda_1$ est la valeur propre dominante de $S(|\lambda_1| > |\lambda_j| , \forall j \neq 1)$ et $\lambda_1$ valeur propre simple alors pour $X_0$ non orthogonal au vecteur propre à gauche de S associé à $\lambda_1$ , la suite $(Xn)_{n \geq 0}$ définie par

$$x_{n+1} = \frac{S x_n}{\alpha_n}$$

où $\alpha_n$ est la composante de module maximum de $S x_n$ converge vers un vecteur propre à droite de S associé à $\lambda_1$ et $\alpha_n$ converge vers $\lambda_1$ .

On a $\|x_n - u_1\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^n\right)$ où $u_1$ est le vrai vecteur propre à droite de S associé à $\lambda_1$ et $\lambda_2$ la deuxième plus grande valeur propre de S.

Les valeurs propres de B sont $\mu_i = \frac{1}{\lambda_i - \mu}$ si les $\lambda_i$ sont les valeurs propres de S.

Pour $\mu = 1 + \varepsilon$ , en appliquant la méthode de la puissance itérée à B , $x_n$ va converger vers le vecteur propre cherché et la vitesse de convergence est

$$V = \frac{\lambda_2 - 1 + \varepsilon}{\varepsilon}$$

donc pour $\varepsilon$ petit, la vitesse de convergence est très grande.

2) Méthodes avec projection de calcul d'éléments propres.

a) Principe de ces méthodes. [11]

L'idée des méthodes que l'on va décrire est d'approcher les éléments propres de S matrice creuse de taille n par ceux d'une matrice de taille beaucoup plus petite p. Cette matrice est obtenue par projection orthogonale de S sur un sous espace $E_p$ . Soit $\pi_p$ la matrice de projection de E sur $E_p$. Alors le couple d'éléments propres de S : $(u,\lambda)$ est approché par $(u_p,\lambda_p)$ vérifiant :

$$\pi_p \, S \, U_p = \lambda_p \, U_p \quad \text{avec} \quad U_p \neq 0 \quad , \; U_p \in E_p$$

$\lambda_p$ et $U_p$ sont les éléments propres de $S_p = \pi_p S$ , et $S_p$ est l'approximation de S associée à la méthode de Galerkin. Soit $\hat{S}_p$ l'application : $\pi_p \, S \restriction E_p$ , $\hat{S}_p$ et $S_p \, \pi_p$ ont les mêmes valeurs propres non nulles $\lambda_p$ , et les mêmes vecteurs propres associés $U_p$ . Il faut donc construire une base orthonormale de $E_p$, et la matrice $(p \times p)$, $H_p$ qui représente $\hat{S}_p$ dans cette base. Le problème est donc ramené à une recherche d'éléments propres dans $\mathbb{C}^p$ :

$$H_p \, \phi_p = \lambda_p \, \phi_p \quad , \quad \phi \in \mathbb{C}^p$$

$\phi_p$ représente les composantes de $U_p$ dans la base de $E_p$ . Etant donnés une valeur propre $\lambda$, et un vecteur propre associé u tel que $\|u\|_2 = 1$ , on voudrait savoir s'il existe une suite d'éléments propres $U_p,\lambda_p$ de $H_p$ , avec $\phi_p \in \mathbb{C}^p$ , qui converge rapidement vers U et $\lambda$ quand p croît. Si $\lambda$ est une valeur propre simple, comme dans notre cas, les valeurs $|\lambda-\lambda_p|$ et $\|U-U_p\|$ peuvent être bornées supérieurement par le résidu

$$r_p = S_pU - \lambda U = (S_p-S)U = (\pi_p-I)SU$$

on a donc $$\|r_p\|_2 \leq |\lambda| \; \|(I-\pi_p)U\|_2 \; .$$

b) Méthode de la puissance itérée

Ici les espaces de projection seront de dimension 1, la méthode consiste donc à projeter les vecteurs propres approchés sur l'espace engendré par eux-mêmes.

Cette méthode permet de calculer les premiers éléments propres dominants d'une matrice S.
A partir d'un vecteur initial $x_0$, on forme la suite de vecteurs $(x_k)$ , $k \geq 0$ définie par :

$$x_{k+1} = \frac{S\ x_k}{\alpha_k} \quad \text{où} \quad \alpha_k = \|S\ x_k\|_\infty$$

* conditions de convergence
  la valeur propre dominante $\lambda_1$ doit être simple et le vecteur initial $x_0$ ne doit pas être orthogonal au vecteur propre à gauche associé à $\lambda_1$ .

* La vitesse de convergence de la puissance itérée est

$$\|x_k - u_1\| = O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^k\right)$$

  où $u_1$ est le vecteur propre cherché et $\lambda_2$ la deuxième plus grande valeur propre de S.
  Cette vitesse peut être légèrement améliorée par une accélération de Tchebycheff.

c) Itérations simultanées [12] [11]

Cette méthode calcule les p premiers éléments propres dominants d'une matrice S, il s'agit d'une généralisation de la méthode de la puissance, où les espaces de projection sont de dimension p.
On génère une suite de matrices $Q_k$ dont les colonnes forment une base orthonormée de l'espace $E_k$ défini par

$$E_k = S^k E_0$$

($E_0$ étant engendré par les colonnes de la matrice initiale $Q_0$).

On procède de la manière suivante :

Pour $k \geq 1$ : 1) calcul de la matrice $(n \times p)$ $X_k = S\, Q_k$

2) orthogonalisation des vecteurs colonnes de $X_k$ :

$X_k = Q_k\, R_k$ c'est-à-dire $Q_k = X_k\, R_k$

où $R_k$ est une matrice triangulaire supérieure.

Si $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq \lambda_p | \ast | \lambda_{p+1}|$ alors la suite de matrices $Q_k$ converge vers $Q$ , dont les colonnes forment une base orthonormale de l'espace propre M engendré par les p premiers vecteurs propres, si et seulement si les p vecteurs initiaux vérifient $\{P\, x_i\}_{1 \leq i \leq p}$ indépendants : où $P$ est la matrice de projection spectrale sur l'espace invariant $M$ . (La projection spectrale étant la projection parallèle à l'espace orthogonale à celui engendré par les vecteurs propres et les vecteurs invariants de $S^T$ ). La vitesse de convergence de la ième colonne de $Q_k$ est de l'ordre de $(\frac{\lambda_{j+1}}{\lambda_j})$ si $|\lambda_{j+1}| \neq |\lambda_j|$

Ce n'est pas une amélioration par rapport à la méthode précédente, on peut toutefois remarquer que la suite $(E_k)$ $k \geq 0$ converge vers M. On considère alors la matrice $(p \times p)$ $H_k = Q_k^T\, S\, Q_k$ projection de S sur $E_k$ dans la base $Q_k$ (voir a)). Alors $H_k$ converge vers $H = Q^T\, S\, Q$ projection sur M dans la base Q.

D'où l'algorithme.

On part de $Q_0$ formé de p vecteur orthonormaux

$$X_{k+1} = S\, Q_k$$

$Q_{k+1}$ est obtenue à partir de $X_{k+1}$ par la décomposition de Schmidt puis périodiquement

$$H_{k+1} = Q_{k+1}^T\, S\, Q_{k+1}$$

On obtient $V_{k+1}$ en diagonalisant $H_{k+1}$

$$H_{k+1} = V_{k+1} \wedge_{k+1} V_{k+1}^T \qquad (\wedge_{k+1} \text{ diagonale})$$

puis

$$Q_{k+2} = Q_{k+1} V_{k+1}^T$$

$Q_{k+2}$ représente alors le système des p vecteurs propres de $H_{k+1}$ qui représente l'opérateur $\pi_{k+1} A \pi_{k+1}$ (où $\pi_k$ est l'opérateur de projection sur $E_k$ dans la base $Q_k$) dans la base $Q_{k+1}$ de $E_{k+1}$.

* Conditions de convergence

  Si $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_p| > |\lambda_{p+1}|$ l'algorithme converge si et seulement si les p vecteurs initiaux vérifient $\{Px_i\}_{1 \leq i \leq p}$ indépendants

* Vitesse de convergence
  Si $|\lambda_j| > |\lambda_{j+1}|$ la vitesse de convergence pour la jème colonne de $Q_k$ est $\|Q_k(.,j) - u_j\| = O(|\frac{\lambda_{p+1}}{\lambda_j}|^k)$ . Donc la vitesse de convergence a été nettement améliorée par rapport à la méthode précédente.

d) Méthode d'Arnoldi [2], [13], [14]

C'est une généralisation de l'algorithme de Lanczos au cas des matrices non symétriques. Elle permet de résoudre de manière efficace les problèmes de calcul des vecteurs propres associés à quelques valeurs propres "extrémales".

* Description de l'algorithme

  Soit S une matrice réelle quelconque de dimension n × n et $x_0$ un vecteur initial quelconque. On génère d'abord une base de vecteurs orthonormaux de l'espace de Krylov de dimension p ($E_p$ engendré par $x_0, Sx_0, \ldots, S^{p-1}x_0$ avec $p << n$), définie par :

$$v_1 = x_0 \,/\, \|x_0\|$$

pour $j=1,2,\ldots,p-1$ faire

$$\hat{v}_{j+1} = Sv_j - \sum_{i=1}^{j} h_{ij}\, V_i$$

avec $h_{ij} = \langle Sv_j, v_i \rangle$ (produit scalaire usuel de $\mathbb{R}^n$)

$$v_{j+1} = \hat{v}_{j+1} \,/\, \|\hat{v}_{j+1}\|$$

$$h_{j+1,j} = \|\hat{v}_{j+1}\|$$

L'algorithme est réalisable si $\|\hat{v}_{j+1}\| \neq 0$ pour $j=1,2,\ldots,n-1$. Notons maintenant $V_p$ la matrice composée de colonnes $v_1,\ldots,v_p$. Alors $H_p = V_p^t\, S\, V_p$ est une matrice de Hessenberg supérieure d'ordre p . Si on appelle $\pi_p$ la projection orthogonale sur $E_p$ alors $H_p$ est la matrice associée à l'opérateur $\pi_p\, S\, \pi_p$ (opérateur de $E_p$ dans $E_p$). On a donc restreint la matrice S au sous-espace $E_p$ , et on va approcher les éléments propres de S par ceux de $H_p$ (les valeurs propres de S par celles de $H_p$, les vecteurs propres de S par $V_p\, y_i^{(p)} = u_i^{(p)}$ , où $y_i^{(p)}$ est le vecteur propre correspondant de $H_p$).

* Convergence

Comme p prend un nombre fini de valeurs, on ne peut parler rigoureusement de convergence. Puisqu'on peut trouver des bornes supérieures pour les quantités $|\lambda_j - \lambda_j^p|$ et $\|u_j - u_j^p\|_2$ , nous allons définir une vitesse de convergence. Lorsque toutes les valeurs propres sont réelles, elle est analogue à celle de l'algorithme de Lanczos pour les matrices symétriques. Si l'on peut avoir des valeurs propres complexes comme dans le cas qui nous préoccupe, alors cette vitesse

de convergence est meilleure que $O(|\frac{\lambda_2}{\lambda_1}|^{p-1})$ si $|\lambda_2| < \lambda_1 = 1$, ce résultat est très faible et reflète mal la vrai vitesse de convergence.

## III - ALGORITHMES UTILISES POUR LES METHODES TESTEES

### 1) Algorithme d'Arnoldi

Cet algorithme permet de calculer le vecteur propre dominant approché associé à la valeur propre 1.

Procédure générale :

1. On initialise la procédure en calculant $S^{15} \frac{u}{\|u\|}$
   Le vecteur u a toutes ses composantes égales à 1. (L'intérêt de cette initialisation est que u aura des composantes dépendant fortement des vecteurs propres dominants).
2. A l'aide d'une orthogonalisation de Schmidt modifiée, on construit la matrice $H_p$ (matrice de Hessenberg) associée à l'opérateur de projection $\pi_p S \pi_p$ . A toutes les étapes de construction on teste si le vecteur courant n'est pas déjà une bonne approximation du vecteur propre cherché.
3. On calcule le vecteur propre associé à la valeur propre le plus proche de 1 de $H_n$ par la méthode du quotient de Rayleigh.
4. Test de convergence. S'il n'est pas satisfait on retourne en 2, avec comme vecteur initial le vecteur propre de $H_n$ obtenu au pas 3.

* Procédure de Schmidt modifiée.

C'est l'algorithme de Gram-Schmidt. [15]
Soit $Q = (q_1, q_2, \ldots, q_p) \in R^{m \times p}$ $(m \geq n)$ une matrice aux colonnes orthonormales

$$Q^T Q = I$$

soit $v \in R^m$ .
On cherche des vecteurs q de $R^m$ , r de $R^p$ et un scalaire $\rho$ tels que

$$(Q,v) = (Q,q)\begin{pmatrix} I & r \\ 0 & \rho \end{pmatrix}$$

et

$$Q^T q = 0$$

La dernière colonne donne : $v = Qr + q\rho$

$$Q^T v = r + 0$$

Si $v' = q\rho$ , on a $v' = v - Qr = (I - QQ^T)v$

Si $m > n$ comme $\|q\| = \|v'\| = \rho$ on a

$$q = \frac{v'}{\rho} \text{ si } \rho \neq 0$$

mais si $m = p$ , la matrice Q est orthogonale

$v' = 0$ et donc $\rho = 0$

Numériquement ce cas est impossible mais toutefois $\rho$ peut être très petit.

Si $\frac{\|v'\|}{\|v\|}$ est petit, la formation de v' produira une grosse erreur sur v' et donc sur q. On peut alors chercher à corriger v' en le réorthogonalisant d'où en appliquant le même processus à v' -qui remplace v- on a :

$$s = Q^T v' \text{ et } v'' = v' - Qs = v - Q(r+s)$$

donc v' est remplacé par v'' et r par r+s.

Si $\frac{\|v''\|}{\|v'\|}$ n'est pas trop petit, on retient v" sinon on réitère le procédé.

D'où l'algorithme :

Soit $\eta$ un paramètre tel que $0 \leq \eta < 1$ par exemple $\eta = \frac{1}{\sqrt{2}}$

$$r^\circ = 0 \quad , \quad v^\circ = v$$

pour k=1,2,3,..., jusqu'à ce que $\|u^k\| > \eta \|u^{k-1}\|$

$$s^k = Q^T v^{k-1}$$

$$r^k = r^{k-1} + s^k$$

$$u^k = Q s^k$$

$$v^k = v^{k-1} - u^k$$

$$r = r^k \quad , \quad \rho = \|v^k\| \quad \text{et} \quad q = \frac{v^k}{\rho}$$

J.W. Daniel a montré [5] qu'un bon test de convergence pour l'algorithme de Schmidt consiste à s'arrêter lorsque

$$\|v_{k-1}\| + \omega\|Q^T v_{k-1}\| < \theta \|v_k\|$$

avec $\omega = \frac{\beta}{\alpha}$ et les paramètres $\omega, \theta$ doivent être spécifiés par l'utilisateur.

J.W. Daniel démontre que si $\alpha$ est "suffisamment" grand et si $\begin{cases} \theta_\alpha > \xi^* \text{ où } \xi^* \to 1 \\ \theta > 1 \end{cases}$ alors la méthode converge comme $\omega = \frac{\beta}{\alpha}$ , si $\alpha \to \infty$ $\omega \to 0$ et donc en prenant $\omega = 0$ et $\theta > 1$ la méthode converge.

Un bon choix qui a été testé est : $\omega = 0$ et $\theta = \sqrt{2}$ ou $\theta = 10$ .

On peut encore améliorer dans notre cas ce test en prenant $\omega = 1$ et $\theta = 2 + 10(\frac{N}{IS})^2$ où IS représente la taille de Q que l'on veut obtenir par une suite d'orthogonalisation de Schmidt.

On montre que cet algorithme converge sous de très larges conditions.

RQIO [16]

Cette procédure calcule le vecteur propre associé à une valeur propre connue estimée (ici 1 au moment où l'on fait la première itération).
On cherche ce vecteur propre par des itérations du quotient de Rayleigh légèrement modifié.

(1) Initialisation

$\lambda_1 = 1$

u est choisi normalisé, avec des composantes de même poids

(2) Calcul et décomposition de $(H_p - \lambda_1 * I)$

(3) Résolution du système $(H_p - \lambda_1 * I)U_2 = U_1$ , normalisation de $U_2$, calcul de la norme de $U_2$ (ceci correspond à un pas d'itération inverse associée à la valeur propre $\lambda_1$)

(4) Nouvelle estimation de la valeur propre $\lambda_2 = \dfrac{U_2^T H_p U_2}{U_2^T U_2}$

Si convergence arrêt
Sinon si $|\lambda_2 - \lambda_1|$ grand, on repart en (3) avec $U_2 = U_1$
sinon on repart en (2) avec $\lambda_1 = \lambda_2$ et $U_1 = U_2$ .

La méthode du quotient de Rayleigh a donc été ici légèrement modifiée pour ne pas avoir à faire trop souvent de décomposition.

On précise ce que signifie au pas 4 les notions de convergence et de $|\lambda_2-\lambda_1|$ grand

$-|\lambda_2-\lambda_1|$ grand

Si $|\lambda_2-\lambda_1|$ est "grand" le pas 3 correspondant à un pas d'itération inverse n'a pas encore convergé, et donc un fait un autre pas d'itération inverse avec un nouveau vecteur départ (celui trouvé précédemment). Si l'itération (inverse a convergé on recommence une itération inverse avec la nouvelle estimation $\lambda_2$ de la valeur propre.
Le test est $|\lambda_2-\lambda_1| >$ DELTA. En entrée au premier appel DELTA vaut la précision demandée pour le calcul du vecteur propre, aux autres appels il vaut 0,1 × Res (où Res = $\|(H_p-\lambda_1 I)U_2\|_\infty$ résidu calculé à la sortie de l'appel précédent de RQIO). Plus le nombre de pas d'itération inverse est grand plus DELTA est important. (On a choisi ici un facteur multiplicatif de 1, 2, on aurait pu prendre tout nombre supérieur à 1).
La convergence dans RQIO est testée sous trois formes :

- on s'est d'abord fixé un nombre d'itérations maximum à ne pas dépasser
- puis on teste le résidu $\|(H_p-\lambda_1 I)U_2\|_\infty$
- on teste enfin la croissance de la norme du vecteur propre estimé

En effet :

Si $\alpha_i$ sont les valeurs propres de $H_n$ , alors les valeurs propres de $(H_p-\lambda_1 I)$ sont $\alpha_i-\lambda_1 = \mu_i$ et celles de $(H_p-\lambda_1 I)^{-1}$ sont $\beta_i = \frac{1}{\mu_i}$ . Si $\lambda_1$ approche à $\varepsilon$ près une des valeurs propres de $H_p$ par exemple $\alpha_1$ , alors $\beta_1 = \frac{1}{\alpha_1-\lambda_1} = \frac{1}{\varepsilon}$ est valeur propre de $(H_p-\ _1\ I)^{-1}$ ; et donc si $U_1$ est une bonne approximation du vecteur propre associé à $\alpha_1$ alors sa norme est multipliée par $\frac{1}{\varepsilon}$ .

Pour $\varepsilon$ petit la croissance de la norme est très forte. On s'arrête donc dans ce cas.

2) Algorithme pour l'itération inverse

Description de l'algorithme.

1. On initialise $u = \frac{v}{\|v\|}$ où v est le vecteur dont chaque composante est égale à 1 (vu la rapidité de la convergence, il est inutile d'appliquer un certain nombre de pas de méthode de la puissance.

2. Décomposition de $B = (S-I-\varepsilon I)$.

3. Résolution de $BU_2 = U_1$ ; Normalisation de $U_2$ .

4. Si convergence, arrêt sinon on repart en 3 avec $U_1 = U_2$ .

La partie délicate de cette méthode est la décomposition de la matrice B . En effet comme $\varepsilon$ est petit, la matrice est presque singulière et de plus d'un point de vue informatique il faut que la matrice reste creuse. Il faut donc trouver une stratégie pour choisir le pivot qui permette d'obtenir un compromis.

3) Algorithme de résolution directe du système

On résoud directement le système $(S-I)P = 0$ en remplaçant la dernière équation par $\Sigma P_i = 1$ .

Description de l'algorithme.

1. Calcul de la matrice du système.

2. Décomposition de cette matrice.

3. Résolution du système et calcul du résidu.

On va dans les chapitres suivants décrire plus en détail les méthodes mises en oeuvre.

# CHAPITRE 2

## DOCUMENTATION DE PROGRAMME D'ARNOLDI

Ce chapitre décrit plus en détail les sous programmes utilisés pour la mise en oeuvre de la méthode d'Arnoldi.

I - Sous programme IOMSP - Sous programme principal.

But : Cette méthode permet de calculer le vecteur propre dominant d'une matrice stochastique.

L'appel s'effectue ainsi :

Call IOMSP (A,NN,N,S,V,H,B,VECTP,VCP,Y,R,PERM,ITMAX,EPS).

On précise d'abord la signification de ces paramètres.

Description des paramètres d'IOMSP

A tableau de réels double précision de dimension (le nombre d'éléments non nuls de la matrice)
en_entrée A contient les éléments non nuls de la matrice creuse dont on cherche le vecteur propre dominant. Ces éléments sont rangés en ligne. Les colonnes et les lignes sont classées dans l'ordre naturel. C'est-à-dire $a_{ij}$ précède $a_{k\ell}$ si $i < k$ ou $i = k$ et $j < \ell$.
Inchangé_en_sortie

NN tableau d'entiers de dimension (le nombre d'éléments non nuls de la matrice A)
en_entrée pour chaque ligne de A, NN contient d'abord le nombre d'éléments non nuls de la ligne puis les indices de colonne de ces éléments non nuls.

Exemple :

Soit la matrice $(a_{ij})$ $1 \leq i \leq 3$ , $1 \leq j \leq 3$

avec $a_{11} = 0,3$ $a_{12} = 0$ $a_{13} = 0,7$ $a_{21} = 0$

$a_{22} = 0,2$ $a_{23} = 0,8$ $a_{31} = 0,1$ $a_{32} = 0,9$

$a_{33} = 0$ alors

NN = ($\underline{2}$ , 1 , 3 , $\underline{2}$ , 2 , 3 , $\underline{2}$ , 1 , 2 )

et

A = (0,3 ; 0,7 ; 0,2 ; 0,8 ; 0,1 ; 0,9)

Les éléments soulignés de NN correspondent au nombre d'éléments non nuls des lignes correspondantes.

Inchangé en sortie

N Entier

en entrée, ordre de la matrice A dont on cherche le vecteur propre à droite associé à la valeur propre 1 dominante.

Inchangé en sortie

S entier

en entrée (S-1) taille de l'espace de Krylov (espace engendré par $X_0, AX_0, \ldots, A^{S-2}X_0$), taille des tableaux H,B,V et des vecteurs VCP et PERM.

inchangé en sortie

V tableau de réels double précision de dimension (N,S) tableau de travail, rempli dans Schmid contient les nouveaux vecteurs de base de l'espace de Krylov.

H tableau de réels double précision de dimension (S,S) tableau de travail, rempli à partir de R, contient la projection de A dans l'espace de Krylov.

B tableau de réels double précision de dimension (S,S) tableau de travail, rempli dans RQIO, contient (H-XLAM*I) (avec XLAM=1 au départ puis XLAM valeur propre approchée par le quotient de Rayleigh).

VECTP tableau de réels double précision de dimension (N)
tableau de travail, en sortie contient le vecteur propre cherché de A.

VCP tableau de réels double précision de dimension (S)
tableau de travail, rempli dans RQIO, contient le vecteur propre exprimé dans la base de Krylov.

Y tableau de réels double précision de dimension (N)
tableau de travail, contient l'approximation en cours du vecteur propre.

R tableau de réels double précision de dimension (2S)
tableau de travail, rempli dans Schmid, contient à chaque pas les coefficients de Fourier (coefficients de réorthogonalisation $R(i) \quad = < A\, V_j, V_i >$ où $1 \leq i \leq j$
$[V_1, \ldots, V_{S-1}]$ est la nouvelle base orthonormale de l'espace de Krylov $R(j+1) = \|V_{j+1}\|$ ).

PERM tableau de logiques de dimension (S)
tableau de travail, rempli dans RQIO, mémorise les permutations effectuées sur B dans la triangularisation de B.

ITMAX entier
en entrée contient le nombre maximum autorisé d'itérations dans Schmid.
inchangé en sortie

EPS réels double précision
en entrée contient la précision que l'on demande au vecteur propre
en sortie EPS contient la valeur du résidu.

Les sous programmes utilisés dans IOMSP sont :
XTAX, SCHMID, et RQIO ; on décrit maintenant l'algorithme de IOMSP.

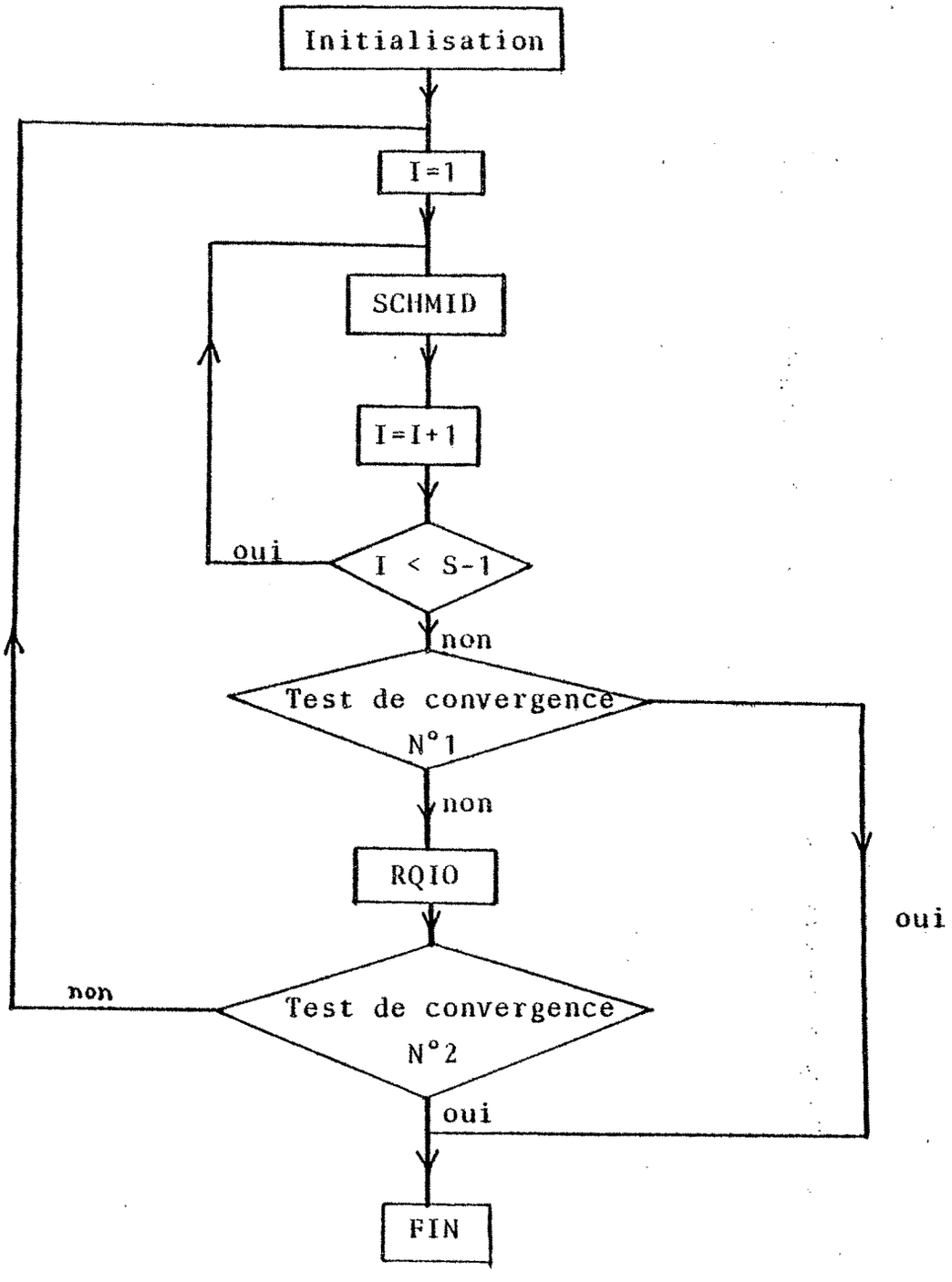
## II - DESCRIPTION DE L'ALGORITHME

On reprend l'algorithme décrit succintement chapitre 1 pragraphe III, en le détaillant, puis en le mettant sous forme d'organigramme.

(0) initialisation

(1) tant que Test de convergence n°2 non satisfait faire
début
I=1;
tant que I < S-1 faire SCHMID;
si test de convergence n°1 satisfait FIN;
RQIO;
fin;

ORGANIGRAMME DE IOMSP




L'initialisation est décrite Chapitre 1 paragraphe III. Les différentes étapes sont décrites dans les paragraphes suivants, et d'abord la construction de $H_n$ à l'aide d'une orthogonalisation de Schmidt modifiée.

## III - SOUS PROGRAMME DE SCHMID

But : Cette procédure orthogonalise, par une méthode de Gram-Schmidt iterative, le vecteur Y par rapport aux (I-1) vecteurs colonnes du tableau Q, qui forment une base orthonormale de l'espace de Krylov. Elle range le vecteur ainsi orthogonalisé dans Q(.,I).
(cf. chapitre 1 paragraphe III)
L'appel s'effectue ainsi :

```
call SCHMID(I,N,Q,Y,R,OMEG,TETA) .
```

1) Paramètres de SCHMID

I entier
en_entrée (I-1) est le nombre des vecteurs par rapport auxquels il faut orthogonaliser, I=2ème dimension de Q,
inchangé_en_sortie

N entier
en_entrée N est la dimension du problème, les vecteurs considérés sont de taille N, 1ère dimension de Q, taille de Y
inchangé_en_sortie

Q tableau de réels double précision de dimension (N,I)
en_entrée les (I-1) premières colonnes sont les vecteurs par rapport auxquels on orthogonalise,
en_sortie Q(.,I) contient le nouveau vecteur de la base orthonormale de l'espace de Krylov.

Y tableau de réels double précision de dimension (N)
en_entrée Y est le vecteur initial que l'on orthogonalise
en_sortie Y contient le vecteur orthogonal aux (I-1) précédents, le vecteur Y est non normalisé.

R tableau de réels double précision de dimension (2S)
tableau de travail,
en_sortie R(1:I) contient les coefficients d'orthogonalisation ($R(k) = \langle Y, Q(.,k) \rangle$ et $R(I+1) = \|Y\|$)

OMEG,TETA réels double précision
en_entrée paramètres du test d'arrêt fixés par l'utilisateur avec TETA > 1 , OMEG $\geq$ 0

inchangés_en_sortie

2) Description de l'algorithme.

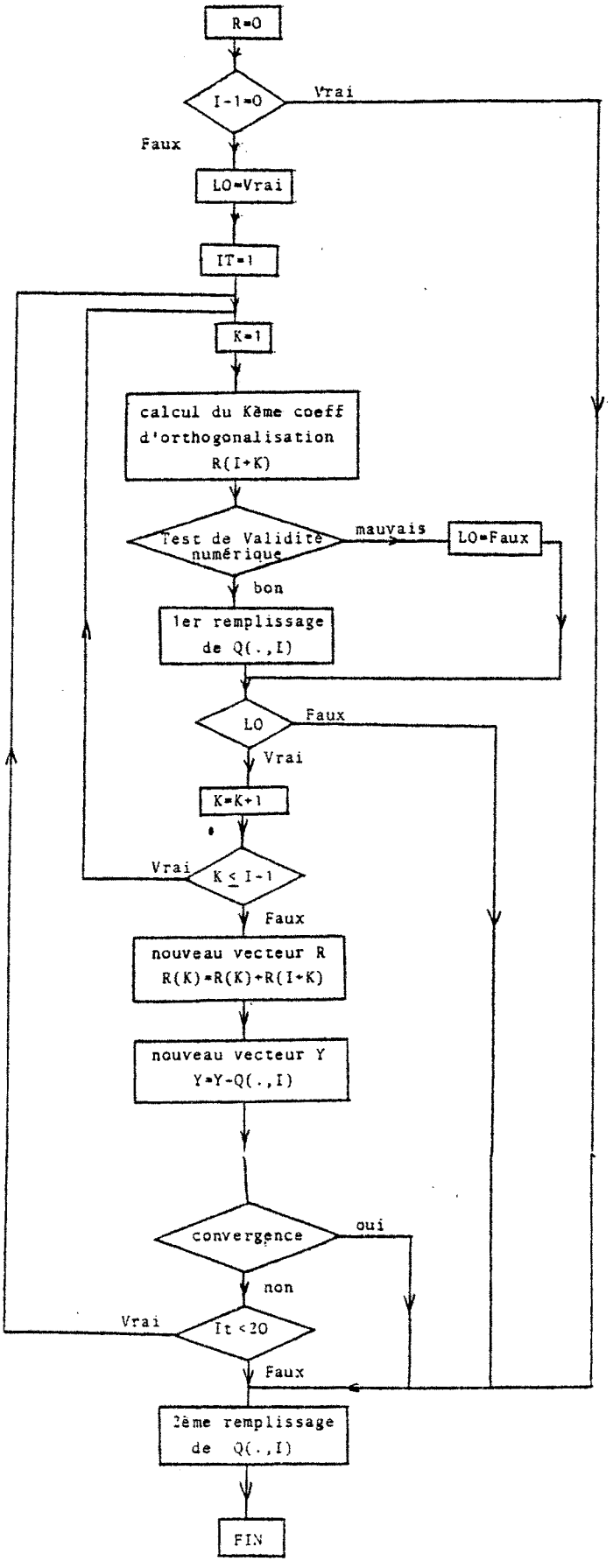
On détaille l'algorithme décrit au chapitre 1 paragraphe III puis on le met sous forme d'organigramme.

* algorithme

```
(0)  initialisation
(1)  si I≠1 faire
       début
       It=1;
       tant que (non convergence) et (It ≤ 20) faire
         début
         K=1;
         tant que K ≤ I-1 faire
           début
           R(I+K) = <Q(.,K),Y> ;
           si validité numérique alors Q(.,I)=Q(.,I)+R(I+K)*Q(.,
             sinon LO=Faux;
           si (non LO) alors
             début
             remplissage de Q(.,I);
             FIN;
             fin;
           fin;
         pour K=1 jusqu'à (I-1) faire R(K)=R(K)+R(I+K);
         Y=Y-Q(.,I);
         It=It+1;
         fin;
       fin;
(2)  Remplissage de Q(.,I);
     FIN;
```

Après l'organigramme on détaille quelques points de l'algorithme.

ORGANIGRAMME DE SCHMID



3) Etapes de l'algorithme

- Calcul du Kème coefficient d'orthogonalisation
  Celui-ci stocké dans R(I+K) correspond à < Q(.,K),Y >
  (Y étant le dernier vecteur orthogonalisé calculé)

- Test de validité numérique
  Il s'agit de tester si la projection de Y rangée dans R(I+.) est très voisine du vecteur nul. En effet, dans ce cas soit le vecteur est déjà orthogonal à l'espace de Krylov, soit le vecteur Y passé en paramètre est déjà dans l'espace de Krylov. On sort alors de Schmid et par le test de convergence numéro un de l'algorithme général on vérifie si le vecteur Y est déjà dans l'espace de Krylov.

- 1er remplissage de Q(.,I)
  Q(.,I) contient la projection de l'Y courant sur l'espace de Krylov

$$Q(.,I) = \sum_{K=1}^{I-1} R(I+K) * Q(.,K)$$

- Test de convergence
  Expliqué plus en détail au chapitre 1 paragraphe III, il consiste à tester si $\|Y\|_{\text{ancien}} + \omega\|R(I+K)\| < \theta\|Y\|_{\text{nouveau}}$
  avec $\omega = 1$ , $\theta = 2+10*(I/S-1)^2$

- 2ème remplissage de Q(.,I)
  Il contient l'Ième vecteur de la base orthonormale de l'espace de Krylov (en fait c'est Y normalisé).

Le test de convergence numéro un de l'algorithme général est expliqué au paragraphe IV ; on expose maintenant le sous programme RQIO .

## IV - SOUS PROGRAMME RQIO

Cette procédure calcule par la méthode du quotient de Rayleigh le vecteur propre d'une matrice Hessenberg supérieure associé à une valeur propre estimée.

L'appel s'effectue ainsi :

Call RQIO(N,A,B,WR,DELTA,RV,PERMUT)

but : Cette procédure calcule le vecteur RV, vecteur propre à droite de la matrice Hessenberg supérieure A de taille (N,N) associée à la valeur propre estimée WR. La méthode décrite au Chapitre 1, paragraphe III, consiste à faire une itération inverse sur A pour la valeur propre estimée WR, puis réestimer WR et recommencer.

### 1) Description des paramètres RQIO

N entier
en_entrée N est la taille de l'espace de Krylov N=S-1, de RV,PERMUT,A et B
inchangé_en_sortie

A tableau de réels double précision de dimension (N,N)
en_entrée A contient la matrice obtenue par projection dans l'espace de Krylov, de la matrice dont on cherche le vecteur propre à droite associé à la valeur propre la plus voisine de 1. A a été calculé par Schmid.
inchangé_en_sortie

B tableau de réels double précision de dimension (N,N) tableau de travail. B est rempli avec $(A-\lambda I)$ à trianguliariser suivant les différentes valeurs de $\lambda$ , puis B contient la partie triangulaire de $(A-\lambda I)$.

WR réel double précision
en_entrée WR contient la valeur propre 1. On cherche le vecteur propre associé à la valeur propre la plus proche de WR. WR contient ensuite les différentes valeurs propres données par la méthode du quotient de Rayleigh, en sortie WR contient une estimation de la valeur propre approchée.

DELTA réel double précision
en_entrée au premier appel DELTA=Eps, aux suivants c'est 0,1 fois le résidu de l'itération précédente (éloignement approché de l'estimation du vecteur propre, et du vrai vecteur propre) en_sortie il contient la distance de la valeur propre approchée au reste du spectre.

RV tableau de réels double précision de dimension (N)
en_sortie RV contient le vecteur propre associé à la valeur propre le plus proche de 1.

PERMUT tableau de logiques de dimension (N)
tableau de travail, PERMUT mémorise les permutations effectuées sur les lignes de B lors de la triangularisation de B.

2) Description de l'algorithme.

On détaille l'algorithme décrit au chapitre 1, paragraphe III, puis on le met sous forme d'organigramme.

Algorithme

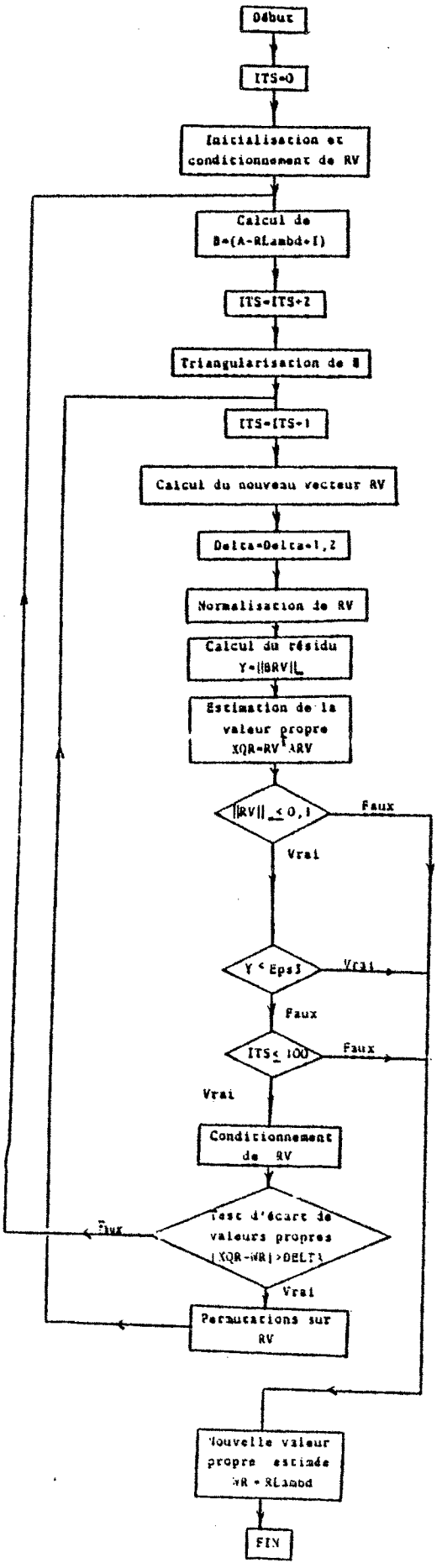
(0) Initialisation et conditionnement de RV;
(1) Tant que (non convergence) et (Its $\leq$ 100) faire
début
si ($|XQR-WR| > Delta$) faire
début
B=(A-RLambd*I) ;
Its=Its+2 ;
Triangularisation de B ;
fin;
Its=Its+1;
$RV=B^{-1}RV$;
Delta=Delta*1,2;

$$RV = \frac{RV}{||RV||};$$

$Y=||BRV||_{\infty}$;
$XQR=(RV)^{T}ARV$;
Permutation sur RV;
WR=RLambd;
fin;

Après l'organigramme on détaille quelques points de l'algorithme.

## ORGANIGRAMME DE RQIO



3) Etapes de l'algorithme

* Initialisation et conditionnement de RV

Dans la résolution du système, si RV est une bonne approximation d'un vecteur propre et RLambd une bonne approximation de la valeur propre associée, la norme du vecteur solution du système est très grande par rapport à la norme de RV de départ. Pour éviter les overflow le "conditionnement" de RV consiste à rendre sa norme petite.

Au premier appel de RQIO , RV est rempli ainsi

$$RV = \begin{matrix} Eps3 \\ Eps3 \\ \vdots \\ Eps3 \end{matrix} \qquad \text{de Eps3} = (\underset{i}{\text{Max}}\ \underset{j}{\text{Max}}\ A_{ij}) \times 10^{-10}$$

ensuite $RV = RV \times \dfrac{Eps3}{\|RV\|_\infty}$

* Triangularisation de B et calcul du nouveau vecteur RV

Pour la triangularisation on applique une méthode de Gauss à une matrice Hessenberg. Il suffit donc d'annuler la sous diagonale avec un choix du meilleur pivot (entre l'élément de la diagonale et celui de la sous-diagonale)

(1) Si $|B(i,i-1)| > |B(i,i)|$ échanger les lignes (i-1) et (i)
Permut (i-1) = vrai, permuter RV(i) et RV(i-1),
si B(i-1,i-1)=0 alors B(i-1,i-1)=Eps3.

(2) soustraire $\dfrac{B(i,i-1)}{B(i-1,i-1)}$ × ligne (i-1) de la ligne (i)

La résolution s'effectue ainsi :

$$(3)\quad RV(K) = \frac{RV(K)}{B(K,K)} - \sum_{j=K+1}^{N} B(K,j) * RV(j)$$

DELTA=DELTA × 1,2

Tests de convergence

Ils sont au nombre de quatre :

- un sur la norme de RV : $\|RV\|_\infty \leq 0,1$ expliqué au chapitre 1 paragraphe III,
- un sur la valeur du résidu : Y < Eps3
  avec Y = ||BRV|| et VRV = (A-RLambd*I)RV
- un sur le nombre d'itérations maximum : ITS $\leq$ 100
- le dernier sur l'écart de valeurs propres :
  |XQR-WR| > DELTA
  pour ce test voir chapitre 1 paragraphe III.

Si le nombre d'itérations est trop grand, on peut remplacer dans le test Y < Eps3 le nombre Eps3 par un nombre plus grand par exemple Eps;de même dans le conditionnement, Eps3 peut être remplacé par un nombre plus grand. Par contre il ne faut pas toucher à Eps3 au pas (1) de la triangularisation de B

* Conditionnement de RV

Avant de recalculer un nouveau vecteur RV, on rend de nouveau sa norme petite RV = Eps3*RV

* Permutations sur RV

Avant de se servir de la matrice triangulaire obtenue dans B, on fait les permutations mémorisées dans PERMUT sur le second membre du système c'est-à-dire sur RV .

On revient maintenant sur les tests de convergence de l'algorithme général.

## V - TESTS DE CONVERGENCE DE IOMSP

Ils sont au nombre de deux :

### 1) Test de convergence numéro un

On a vu dans la méthode de la puissance itérée que la suite des vecteurs $(X_n)$ $n \geq 0$ , définie par $X_{n+1} = \frac{AX_n}{\alpha_n}$ où $\alpha_n = \|AX_n\|$ convergeait vers le vecteur propre dominant de A, pour tout vecteur $X_o$ non orthogonal au vecteur propre à gauche associé à la valeur propre dominante.
Ici la valeur propre dominante étant 1 cela veut dire qu'à partir d'un certain rang on aura $AX_n \approx X_n$ .
Comme on forme l'espace de Krylov à partir de $(X_o, AX_o, \ldots, A^{s-2}X_o)$ si $X_o$ est suffisamment proche de la solution il est possible que $A^rX_o$ avec $r < s-2$ soit vecteur propre de A associé à la valeur propre 1.
Dans ce cas $A^jX_o = A^rX_o$ pour $j \geq r$ .
Le test $\|(A-I)Y\|_\infty \leq Eps$ permet donc avant l'appel de Schmid de vérifier que le vecteur $Y = A^rX_o$ à orthogonaliser n'est pas déjà le vecteur propre cherché. Ce test permet d'éviter que le paramètre de Schmid soit impossible à orthogonaliser. Ce test complète donc celui de validité numérique de Schmid.

### 2) Test de convergence numéro deux

Il s'agit comme dans le précédent de tester la valeur du résidu, ceci après l'appel de RQIO.

# CHAPITRE 3

## DOCUMENTATION DE PROGRAMME DE L'ITÉRATION INVERSE

Ce chapitre décrit plus en détail les sous programmes utilisés pour la mise en oeuvre de la méthode d'itération inverse. Cette méthode permet de calculer le vecteur propre associé à la valeur propre 1 d'une matrice creuse.
L'appel s'effectue ainsi

```
Call ITIN (A,IND,ZEPS,IND2,IW,N,IA,NW,IIW,INDW,
          IWW,IND1,IW1,A1,B1,B,B2,W)
```

## I - DESCRIPTION DES PARAMETRES D'ITIN

A tableau de réels double précision de dimension (IA)
en entrée A contient les éléments non nuls de la matrice creuse (A-I) et tous les éléments de la diagonale principale. Les éléments de A sont rangés par colonne. Les lignes sont classées dans l'ordre naturel ainsi que les colonnes. C'est-à-dire $a_{ij}$ précède $a_{k\ell}$ si $j < \ell$ ou $j = \ell$ , et $i < k$
en sortie A contient les éléments de la matrice triangulaire associée à (A-(1+XMU)I), cf. remplissage de A.

IND tableau d'entiers de dimension (IA)
en entrée IND contient le numéro de la ligne des éléments de A. C'est-à-dire si $(a_{ij})$ est rangé dans A(K), alors IND(K) contient i
en sortie IND contient les numéros de ligne de la matrice triangulaire rangée dans A.

ZEPS réel double précision
en_entrée ZEPS contient la précision que l'on demande au vecteur propre (borne du résidu)
inchangé_en_sortie

IND2 tableau d'entiers de dimension (IA)
tableau de travail rempli dans FO3AJF

IW tableau d'entiers de dimension (IIW,R) avec R $\geq$ 13
On a pris R = 13 ,
en_entrée IW(I,1) contient l'indice dans le tableau du premier élément de la colonne I, et IW(N+1,L) contient l'adresse du premier élément de A non utilisé. Cette colonne est inchangée en sortie. Le reste du tableau est utilisé comme espace de travail dans FO3AJF et FO4APF.

N entier
en_entrée N contient l'ordre de la matrice
inchange_en_sortie

IA entier
en_entrée IA contient la dimension des tableaux A, IND, IND2 . Le nombre d'éléments non nuls dans la forme décomposée est limité à IA, et donc une estimation très large est recommandée,
en_sortie IA est inchangé

NW entier
en_entrée NW contient la dimension des tableaux IWW et INDW.
Une valeur de N/10 est souhaitable.
inchangé_en_sortie.

IIW entier
en_entrée contient la première dimension du tableau IW
IIW $\geq$ N+1
inchangé_en_sortie

Tableaux de travail

INDW Tableau d'entiersde dimension (NW)
utilisé dans FO3AJF.

IWW Tableau d'entiersde dimension (NW)
utilisé dans FO3AJF.

IND1 Tableau d'entiersde dimension (IW(N+1,1))
Après remplissage contient les numéros de lignes des éléments non nuls de (A-I) recopiés dans A1.
A1, IND1 et IW1 permettent de calculer le résidu.

IW1 Tableau d'entiersde dimension (N)
Après remplissage IW1(I) contient l'indice dans A1 du premier élément de la colonne I.

A1 Tableau de réels double précision de dimension (IW(N+1,1))
Après remplissage contient les éléments non nuls et ceux de la diagonale principale de (A-I).

B1 Tableau de réels double précision de dimension (N) contient les estimations successives du vecteur propre cherché en sortie B contient le vecteur propre cherché.

B2 Tableau de réels double précision de dimension (N)
contient la dernière valeur de B, sert dans le deuxième test d'arrêt.

W Tableau de réels double précision de dimension (N)
rempli dans FO4APF, pour un gain de place on peut faire EQUIVALENCE(IW(1,6)W(1)).

Sous programmes utilisés :

Les programmes FO3AJF et FO4APF de la bibliothèque NAG.

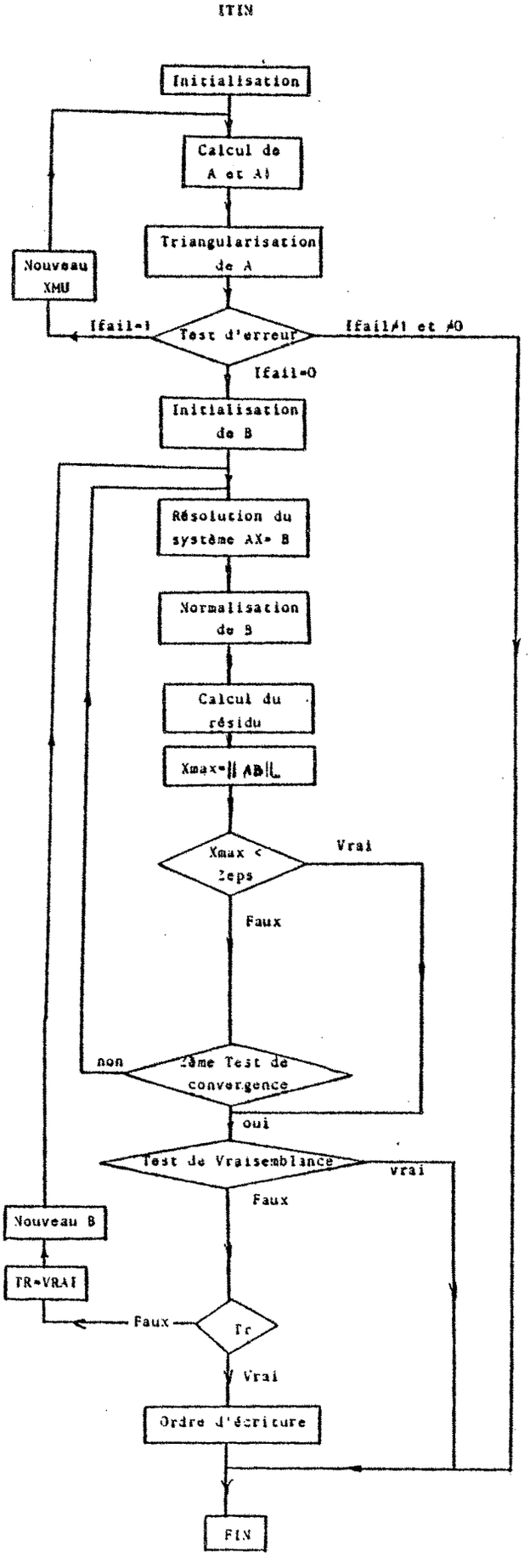
## II - DESCRIPTION DE L'ALGORITHME

On reprend en le détaillant l'algorithme décrit succintement chapitre 1, paragraphe IV.

(0) initialisation ;
(1) calcul de A et A1;
Triangularisation de A;
si erreur faire
début
nouveau XMU ;
calcul de A et A1;
triangularisation de A1;
fin;
(2) Tant que (non convergence) faire
début
$B = A^{-1}B$;
$B = B/||B||$;
$XMAX = ||AB||_\infty$;
fin CV;
si (non vraisemblance) faire
début
nouveau B ;
tant que (non convergence) faire
début
$B = A^{-1}B$;
$B = B/||B||$;
$XMAX = ||AB||_\infty$ ;
fin;
fin vraisemblance;

Les différentes étapes de cet algorithme sont décrites après l'organigramme. On commence par la triangularisation de A qui nécessite le plus de temps d'exécution.

## ORGANIGRAMME GENERAL



## III - TRIANGULARISATION DE A

Pour ce faire on appelle la procédure de NAG : FO3AJF

Cette procédure décompose une matrice réelle et creuse A en deux factures triangulairesLU et elle calcule le déterminant

L'appel s'effectue ainsi :

```
Call FO3AJF(A,IND,IW,IIW,IND2,N,IWW,INDW,NW,6,U,
                IA,D1,ID,JSCALE,IFAIL)
```

But : Etant donné une matrice creuse A, le sous programme décompose A en deux matricestriangulaires,A = LU (L matrice triangulaire unitaire, U matrice triangulaire supérieure), en utilisant une stratégie de pivotage pour trouver un compromis entre maintenir la creusité et garder une bonne précision.

LISTE DE PARAMETRES

A,IND,IW,IIW,IND2,N,IWW,NW,IA
Cf. programme principal.

G réel double précision
en_sortie G est une estimation de la perturbation relative des éléments de A.

U réel double précision
en_entrée U doit avoir une valeur tel que $0 < U \leq 1$
U permet de contrôler le choix des pivots. Quand on cherche une ligne ou une colonne comme pivot tout élément plus petit que U fois le plus grand est exclu.

D1 réel double précision
en_sortie D1 contient une partie du déterminant, lorsque le déterminant s'écrit D1*(2,0**10)

ID entier
en_sortie donne l'autre partie du déterminant.

JSCALE entier

en entrée égal à 1, sert d'indication d'erreur

en sortie vaut

0 s'il n'y a pas d'erreur

1 si la matrice est singulière

si JSCALE > 0 la ligne de numéro JSCALE dépend des autres

si JSCALE < 0 la colonne de numéro (JSCALE) dépend des autres

2 si l'élément A(JSCALE) est overflow

3 si la taille de A est trop petite

si JSCALE << N alors il faut beaucoup plus de place mémoire, sinon juste un peu plus suffit

On expose maintenant la résolution du système qui découle de cette triangularisation.

## IV - RESOLUTION DU SYSTEME AX=b

Pour ce faire on appelle la procédure de NAG qui est complémentaire avec F03AJF : F04APF.

F04APF calcule la solution approchée d'un système d'équations, associé à une matrice creuse, avec un seul deuxième membre. Le système s'écrit AX=b ou $A^TX=b$ , où A a déjà été mis sous forme LU par la procédure F03AJF.

L'appel s'effectue ainsi :

Call F04APF(A,IND,IW,IIW,IA,W,B,MTYPE,IFAIL)

But : le sous programme prend les facteurs triangulaires : A=LU, où L est une matrice triangulaire inférieure unitaire et U une matrice triangulaire supérieure et suivant les valeurs de MTYPE fait :

- pour MTYPE = 1 on calcule $b = A^{-1}b$ c'est-à-dire on résout $AX=b$ en deux temps d'abord $Ly=b$ puis $Ux=y$ ;
- pour MTYPE = 2 on calcule $b = (A^T)^{-1}b$ c'est-à-dire on résout $A^TX=b$ en deux temps d'abord $U^Ty=b$ puis $Lx=y$ ;
- pour MTYPE = 3 on calcule $b = Ab$ c'est-à-dire on fait d'abord la multiplication par U puis celle par L ;
- pour MTYPE = 4 on calcule $b = A^Tb$ c'est-à-dire on fait d'abord la multiplication par $L^T$ puis celle par $U^T$ .

REMARQUE :

Si l'algorithme général s'avère performant, on pourra toujours pour cette procédure alléger le programme source, en enlevant les lignes de code spécifiques des problèmes associés à MTYPE = 2, 3, 4 .

LISTE DES PARAMETRES

A,IND,IW,IIW,N,IA,W voir programme principal.

B tableau de réels double précision, inchangé en sortie de dimension (N)
en_entrée B contient le second membre de l'équation à résoudre
en_sortie il contient la solution approchée du système.

MTYPE entier
permet de faire le choix du système à résoudre (cf. plus haut).
inchangé en sortie

IFAIL entier
en_entrée IFAIL vaut 1, sert d'indicateur d'erreur
en_sortie vaut :

0 s'il n'y a pas d'erreurs
1 si FO3AJF a engendré des erreurs
2 si on n'est pas passé dans FO3AJF
3 si FO3AKF sous programme de FO3AJF a engendré des erreurs
4 si MTYPE $\neq$ 1, 2, 3, 4.

On revient maintenant sur les autres étapes de l'algorithme général.

## V - AUTRES ETAPES D'ITIN

On les reprend dans l'ordre de leurs apparitions dans l'organigramme du paragraphe II.

- Calcul de A et A1

On a vu chapitre 1 paragraphe II d) que la méthode d'itération inverse consistait en une méthode de la puissance itérée sur la matrice $(A-\mu I)^{-1}$, où $\mu$ est une valeur proche de la valeur propre dont on veut connaitre un vecteur propre associé. Puis, chapitre 1 paragraphe 4 on a posé, puisqu'ici cette valeur propre était 1, la valeur égale à $1+\varepsilon$. Dans ITIN cet $\varepsilon$ est XMU. XMU peut être choisi par l'utilisateur sachant que :

- $\varepsilon$ doit être positif, car 1 étant valeur propre dominante et le spectre ayant beaucoup de valeurs propres proches de 1, on ne risque pas alors d'approcher plus près d' une autre valeur propre que 1 .
- plus $\varepsilon$ est petit plus la vitesse de convergence : $v = \frac{\lambda_1 - 1 + \varepsilon}{\varepsilon}$ (cf. chapitre 1 paragraphe 2) est bonne .
- si $\varepsilon$ est trop petit le système peut devenir un système singulier à cause des arrondis machine.

Dans A on range la matrice (A-I-XMU*I) et dans A1 on range la matrice transmise en paramètre c'est-à-dire (A-I). Dans le programme XMU a été pris au départ égal à $10^{-10}$ .

- Test d'erreur, et nouveau XMU

Il s'agit de tester s'il y a eu des erreurs dans la décomposition de la matrice faite par F03AJF. On a vu paragraphe II de ce chapitre que cette information était transmise par l'intermédiaire de IFAIL. Ce test revient donc à tester la valeur de IFAIL.

• Si IFAIL = 0 pas d'erreur on continue.

• Si IFAIL = 1 on a une matrice singulière, d'après ce qui a été dit au paragraphe précédent, cela signifie que XMU est trop petit, donc on recommencera avec un "nouveau XMU" c'est-à-dire on pose XMU = $10^{-3}$ , et on remet le tableau A à sa valeur de départ c'est-à-dire (A-I).

• Si IFAIL = 3 on arrête le programme de manière à ce que l'utilisateur augmente la taille de A si IFAIL = 3 , et regarde ce qui se passe dans le tableau A pour IFAIL = 2.

- Initialisation de B

Cf. chapitre 1 paragraphe IV. B = $\frac{V}{\|V\|}$ où V est le vecteur dont chaque composante est égale à 1.

- Tests de convergence et tests de vraisemblance.

Les tests de convergence sont au nombre de deux.

• Test de convergence numéro un : Xmax < Zeps

Il s'agit comme dans la méthode d'Arnoldi de tester la valeur du résidu. Pour des raisons de cohérence ZEPS est égal à EPS d'Arnoldi.

• Test de convergence numéro deux :

$$\|B_{nouveau} - B_{ancien}\|_\infty < 10^{-5}$$

On a mémorisé dans B2 l'ancienne valeur de B, et on calcule la norme du maximum du vecteur B2-B.

Le test de vraisemblance permet de tester la cohérence du vecteur résultat.

- Test de vraisemblance

Comme la matrice A est stochastique on sait qu'un vecteur propre de A associé à la valeur propre 1 a toutes ses composantes de même signe. C'est une information très forte pour N grand.

On calcule XMO = Min b(i) et XPL = Max b(i) et on regarde
i

s'ils sont de même signe, en acceptant que leurs signes soient différents si $|XMO| < 2 * 10^{-3}$ ou $|XPL| < 2 * 10^{-3}$ (ceci à cause de petites erreurs de calcul).

Si le test de vraisemblance est faux pour la première fois (première fois donnée par TR), on refait une résolution de Ax = B avec un nouveau vecteur B de départ sinon on s'arrête.

- Nouveau B

Toutes les composantes de B supérieures à $2*10^{-3}$ sont mises égales à 1, les autres à 0 . Ceci pour ne prendre que les composantes de B suffisamment positives.

Remarque : B ne peut alors être orthogonal au vecteur propre à gauche de A associé à la valeur propre 1 puisque un de ces vecteurs propres à gauche est de la forme :

$$\begin{vmatrix} \alpha \\ \alpha \\ \vdots \\ \alpha \end{vmatrix} \quad \text{avec} \quad \alpha \neq 0$$

# CHAPITRE 4

## DOCUMENTATION DE PROGRAMME DE LA RÉSOLUTION DIRECTE

Ce chapitre décrit plus en détail les sous-programmes utilisés pour la mise en oeuvre de la méthode de résolution directe.
Cette méthode permet de résoudre le système :

$$\begin{cases} (S-I)P = 0 \\ \Sigma P_i = 1 \end{cases}$$

L'appel s'effectue ainsi :

```
Call RESOL (A,IND,IND2,IW,N,IA,NW,IIW,
            INDW,IWW,IND1,A1,B,W,B1)
```

## I - DESCRIPTION DES PARAMETRES DE RESOL

A Tableau de réels double précision de dimension (IA)
en_entrée : A contient les éléments non nuls de la matrice creuse (A-I). Les éléments de A sont rangés par colonne. Les lignes sont classées dans l'ordre naturel ainsi que les colonnes.
C'est-à-dire $a_{ij}$ précède $a_{k\ell}$ si $j < \ell$ ou $j = \ell$, et $i < k$
en_sortie : A contient les éléments de la matrice triangulaire associée à la matrice (S-I) ou la dernière ligne a été remplacée par une ligne ne contenant que des 1 .

IND Tableau d'entiers de dimension (IA)

en_entrée IND contient le numéro de la ligne des éléments de A. C'est-à-dire $(a_{ij})$ est rangé dans A(K) alors IND(K) contient i

en_sortie : IND contient les numéros de ligne de la matrice triangulaire rangée dans A .

IND2 Tableau d'entiers de dimension (IA)
tableau de travail rempli dans FO3AJF

IW Tableau d'entiers de dimension (IIW,R) avec $R \geq 13$
On a pris R = 13 ,

en_entrée IW(I,1) contient l'indice dans le tableau du premier élément de la colonne I , et IW(N+1,L) contient l'adresse du premier élément de A non utilisé. Cette colonne est inchangée en sortie. Le reste du tableau est utilisé comme espace de travail dans FO3AJF et FO4APF.

N entier

en_entrée N contient l'ordre de la matrice
inchangé_en_sortie

IA entier

en_entrée IA contient la dimension des tableaux, A, IND, IND2. Le nombre d'éléments non nuls dans la forme décomposée est limité à IA, et donc une estimation très large est recommandée,
en_sortie IA est inchangée

NW entier

en_sortie NW contient la dimension des tableaux IWW et INDW.
Une valeur de N/10 est souhaitable.
inchangé_en_sortie.

IIW entier

en_entrée contient la première dimension du tableau IW
$IIW \geq N+1$
inchangé_en_sortie

Tableaux de travail

INDW Tableau d'entiers de dimension (NW)
utilisé dans F03AJF.

IWW Tableau d'entiers de dimension (NW)
utilisé dans F03AJF.

IND1 Tableau d'entiers de dimension (IW(N+1,1))
Après remplissage contient les numéros de ligne des éléments non nuls de (S-I) recopiés dans A1.
A1, IND1 et IW1 permettent de calculer le résidu.

IW1 Tableau d'entiers de dimension (N)
Après remplissage IW1(I) contient l'indice dans A1 du premier élément de la colonne 1.

A1 Tableau de réels double précision de dimension (IW(N+1,1)).
Après remplissage contient les éléments non nuls de (S-I).

W Tableau de réels double précision de dimension (N) rempli dans F04APF, pour un gain de place on peut faire EQUIVALENCE(IW(1,6)W(1)).

B1 Tableau de réels double précision de dimension (N) contient après la résolution du système le vecteur résidu : (S-I)B .

Sous programmes utilisés :

Les programmes F03AJF et F04APF de la bibliothèque NAG.

## II - DESCRIPTION DE L'ALGORITHME

On reprend en le détaillant l'algorithme décrit succintement chapitre 1, paragraphe III 3).

(0) Calcul de la matrice A associée au système;
Calcul de B;
(1) Triangularisation de A;
si erreur fin;
(2) Résolution de AX=B;
si erreur fin;
calcul du vecteur résidu;
fin;

Cet algorithme étant très simple, on n'a pas fait d'organigramme ni détaillé les étapes de l'algorithme. Pour la triangularisation et la résolution il faut se reporter au chapitre 3 paragraphe III et IV.

# CHAPITRE 5

# RÉSULTATS ET CONCLUSION

Dans ce chapitre, après un rappel des contraintes dans lesquelles se sont effectuées les travaux, on présente les résultats obtenus.

## I - CONTRAINTES IMPOSEES

### 1) Taille du programme

Le logiciel de résolution du problème proposé devant s'inscrire à l'intérieur d'un logiciel beaucoup plus important (QNAP), il était nécessaire que le logiciel numérique ne soit pas trop important.

Ceci est d'une grande importance, car dans le cadre du choix de la méthode de résolution du système, on a été amené à prendre une méthode peu performante non seulement sur le plan temps de calcul mais aussi sur celui de propagation d'erreur.

### 2) Précision demandée

La solution du problème correspondant au vecteur des probabilités à l'équilibre d'une chaine de Markov irréductible, il semble nécessaire d'avoir sur chaque composante de ce vecteur une précision de l'ordre du millième. Le fait que la précision ne soit pas globale mais porte sur chaque terme a imposé le choix de la norme du vecteur résidu : $\| \ \|_\infty$ .

### 3) Temps de calcul

Pour les contraintes de temps, le fait que ce logiciel soit intégré dans un logiciel plus gros a deux conséquences :

- il ne faut pas pénaliser le programme total par un temps de calcul trop important,
- dans le logiciel, des méthodes de résolution approchée sont déjà implémentées. Il faut donc que notre méthode ait des temps de calcul analogues : la connaissance de la solution exacte ne doit pas entraîner une perte de temps trop importante.

4) Taille des matrices à traiter

Le logiciel QNAP est capable d'analyser des réseaux de toutes tailles. La taille des matrices à traiter est limitée uniquement par l'espace mémoire disponible, et par les répercussions que l'augmentation de taille entraine sur les contraintes de temps et de précision.

Il est sûr que du point de vue de la modélisation de systèmes informatiques il est intéressant de pouvoir multiplier les états de la chaine de Markov générée et donc d'augmenter la taille de la matrice des probabilités de transition.

De plus, il est nécessaire pour le logiciel de savoir traiter des matrices dont le spectre admet beaucoup de valeurs propres de module proche de 1. Dans le paragraphe suivant on donne des exemples de spectre de matrice à traiter.

5) "Conditionnement" des matrices

Dans le tableau suivant nous avons calculé les dix valeurs propres de plus grand module pour trois matrices de taille moyenne qui sont des exemples type de ce que le logiciel va avoir à traiter. Les valeurs propres de ces matrices ont été obtenues par la méthode des itérations simultanées (cf. chapitre 1 paragraphe II 2) c)).

On remarque dans les trois cas que ces dix valeurs propres ont un module très proche de 1 . La recherche du vecteur propre associé à la valeur propre 1 sera donc difficile.

*voir tableau.*

| N | 66 | 66 | 132 | 24 |
|---|---|---|---|---|
| $\tau$ | 353 | 353 | 589 | 97 |
| V.P. | 1.000 | 1.000 | 1.000 | 1.000 |
| | 0.9943 | 0.9964 | 0.9744 | 0.9464 |
| | 0.9892+i0.0062 | 0.9930 | 0.9707 | 0.9106+i0.0445 |
| | 0.9892+i0.0062 | 0.9896 | 0.9541 | 0.9106-i0.0445 |
| | 0.9878-i0.0058 | 0.9857 | 0.9460+i0.0411 | 0.8527 |
| | 0.9878-i0.0058 | 0.9818 | 0.9460+i0.0411 | 0.7948+i0.2782 |
| | 0.9818+i0.0070 | 0.9774 | 0.9393+i0.0449 | 0.7948-i0.2782 |
| | 0.9818-i0.0070 | 0.9729 | 0.9393-i0.0449 | 0.5901+i0.3260 |
| | 0.9763+i0.006 | 0.9675 | 0.9261+i0.0091 | 0.5901-i0.3260 |
| | 0.9763-i0.006 | 0.9629 | 0.9261-i0.0091 | 0.6724 |

Valeurs propres dominantes de matrices types

N : nombre d'états du système, taille de la matrice

$\tau$ : nombre de terme non nuls dans la matrice stochastique

VP : valeurs propres de la matrice.

## II - RESULTATS

Les méthodes présentées précédemment ont été testées sur le HB68, ordinateur du Centre Interuniversitaire de Calcul de Grenoble (CICG); les temps de calcul obtenus sont supérieurs à ceux obtenus par des programmes similaires sur un IRIS 80 ou sur un IBM/360. Notamment pour le logiciel de résolution de système on ne retrouve pas des temps comparables à ceux de référence des sous programmes utilisés.

Dans les tableaux présentés ci-après, les comparaisons des résultats ont été faites par rapport à ceux obtenus par la méthode d'itération inverse. En effet, celle-ci, dans les cas difficiles, obtient les meilleurs résidus et, dans le cas facile, les résidus sont inférieurs à $10^{-10}$ et les résultats peuvent alors être considérés comme exacts. Ceci a d'ailleurs été vérifié à l'aide de la résolution directe lorsque le résidu obtenu était nul.

Dans les paragraphes suivants, les résultats sont d'abord étudiés pour chaque méthode individuellement, puis on fait la comparaison entre les méthodes. Nous commençons par la méthode dont les résultats sont les plus précis.

On trouvera en annexe le vecteur de probabilités à l'équilibre pour l'exemple $n = 66$ , $\tau = 86$ (ex. n°11 ) obtenu par les trois méthodes.

### 1) Résultats de l'itération inverse

Le tableau suivant répertorie les résultats obtenus par itération inverse pour les différents exemples avec leurs caractéristiques :

N = nombre d'états du système, taille de la matrice

$\tau$ = nombre d'éléments non nuls de la matrice.

| Exemple N° | Données N | Données $\tau$ | Itérations | Précision de la décomposition | Résidu | Temps de calcul (s) |
|---|---|---|---|---|---|---|
| 1 | 4 | 7 | 1 | $2,2\ 10^{-19}$ | $1,3\ 10^{-12}$ | 0.385 |
| 2 | 6 | 16 | 1 | $3,5\ 10^{-18}$ | $3,6\ 10^{-20}$ | 0.285 |
| 3 | 8 | 26 | 1 | $2,2\ 10^{-19}$ | $1,25\ 10^{-12}$ | 0.45 |
| 4 | 8 | 26 | 1 | $2,2\ 10^{-19}$ | $1,4\ 10^{-12}$ | 0.36 |
| 5 | 16 | 58 | 1 | $2,2\ 10^{-18}$ | $1,4\ 10^{-12}$ | 0.60 |
| 6 | 23 | 71 | 1 | $3,5\ 10^{-18}$ | $1,2\ 10^{-12}$ | 0.56 |
| 7 | 31 | 91 | 1 | $3,5\ 10^{-18}$ | $8,3\ 10^{-7}$ | 1.78 |
| 8 | 31 | 105 | | matrice non inversible | | |
| | | | 13 | $5,6\ 10^{-17}$ | $8\ 10^{-7}$ | 8.28 |
| 9 | 37 | 131 | 1 | $3,5\ 10^{-18}$ | $9,2\ 10^{-7}$ | 2.56 |
| 10 | 54 | 250 | 1 | $3,5\ 10^{-18}$ | $1,9\ 10^{-6}$ | 2.02 |
| 11 | 66 | 286 | 1 | $3,5\ 10^{-18}$ | $4,8\ 10^{-9}$ | 4.74 |
| 12 | 66 | 451 | 1 | $5,6\ 10^{-17}$ | $4,1\ 10^{-9}$ | 4.70 |
| 13 | 67 | 227 | 12 | $4,1\ 10^{-17}$ | $1,1\ 10^{-5}$ | 7.98 |
| 14 | 115 | 421 | 1 | $3,5\ 10^{-18}$ | $4,6\ 10^{-10}$ | 5.06 |
| 15 | 185 | 1051 | 1 | $3,5\ 10^{-18}$ | $3\ 10^{-8}$ | 41.19 |
| 16 | 216 | 971 | 1 | $3,5\ 10^{-18}$ | $1,5\ 10^{-13}$ | 30.3 |
| 17 | 216 | 971 | 1 | $5,6\ 10^{-17}$ | $1,3\ 10^{-7}$ | 25.82 |
| 18 | 343 | 1567 | 1 | $3,5\ 10^{-18}$ | $1,3\ 10^{-13}$ | 93.8 |
| 19 | 512 | 2367 | 1 | $3,5\ 10^{-18}$ | $8,6\ 10^{-14}$ | 372.6 |
| 20 | 1107 | 5664 | Matrice trop importante | | | |

Dans ce tableau on peut tout de suite noter que dans le test numéro huit correspondant à une matrice de taille 31 , la méthode sans préconditionnement de la matrice nous avait conduit à une matrice non inversible pour $(S-\varepsilon I)$, pour $\varepsilon = 10^{-11}$; le programme est alors reparti au départ avec $\varepsilon = 10^{-3}$ ce qui a donné 13 itérations et un temps de calcul plus important. La méthode avec préconditionnement initial de la matrice donne les résultats suivants :
1 itération , un résidu de $8{,}2\,10^{-9}$ et une erreur de décomposition de $5{,}6\ 10^{-17}$ .

Le tableau qui est présenté est celui correspondant à la méthode sans préconditionnement , pour montrer que ce préconditionnement n'est nécessaire que dans un cas.

En effet, le préconditionnement utilisé (multiplication de toutes les colonnes de la matrice par $1/\varepsilon$) n'influe que sur l'inversibilité de la matrice. A ce cas près le tableau avec préconditionnement est identique.

Le dernier cas n'a pu être traité par cette méthode parce qu'elle coûtait trop cher (un temps supérieur à 2000 s). D'une manière générale on voit que la progession très rapide du temps de calcul sur les derniers exemples du tableau montre que la méthode est limitée à des matrices de taille moyenne.

On peut aussi noter d'après les exemples 11, 12 et 13, ou 7, 8 et 9 qui sont des matrices de tailles voisines mais avec des valeurs propres réparties différemment à l'intérieur du spectre, que la méthode n'est pas trop sensible à la qualité de celui-ci sauf dans un cas très difficile comme le 13ème. C'est d'ailleurs le seul qui ait fait appel au phénomène itératif de la méthode.

On va voir dans le paragraphe suivant qu'au contraire la méthode d'Arnoldi est plus sensible à la répartition des valeurs propres.

2) Résultats de la méthode d'Arnoldi

Le tableau suivant répertorie les résultats obtenus par la méthode d'Arnoldi pour les exemples déjà décrits pour la méthode d'itération inverse.

Caractéristiques de ces exemples :

N = nombre d'états du système, taille de la matrice

$\tau$ = nombre d'éléments non nuls de la matrice.

*Voir tableau page suivante.*

| Exemple N° | Données N | Données $\tau$ | Itérations | Résidu | Temps de calcul (s) | Résultat |
|---|---|---|---|---|---|---|
| 1 | 4 | 7 | 1 | $2,1\ 10^{-10}$ | 0.46 | exact |
| 2 | 6 | 16 | 1 | $6,2\ 10^{-10}$ | 0.26 | exact |
| 3 | 8 | 26 | 10 | $1,1\ 10^{-5}$ | 2.68 | exact |
| 4 | 8 | 26 | 21 | $6,6\ 10^{-5}$ | 5.98 | $5\ 10^{-5}$ |
| 5 | 16 | 58 | 8 | $1,5\ 10^{-5}$ | 8.35 | $5\ 10^{-5}$ |
| 6 | 23 | 71 | 9 | $2,8\ 10^{-8}$ | 2.71 | $5\ 10^{-5}$ |
| 7 | 31 | 91 | 22 | $9,9\ 10^{-6}$ | 13.25 | $4\ 10^{-2}$ |
| 8 | 31 | 105 | 25 | $1,8\ 10^{-6}$ | 13.84 | $3\ 10^{-3}$ |
| 9 | 37 | 131 | non convergence | | | |
| 10 | 54 | 250 | 7 | $2,5\ 10^{-5}$ | 4.36 | $5\ 10^{-4}$ |
| 11 | 66 | 286 | 17 | $4,3\ 10^{-6}$ | 10.65 | $3\ 10^{-5}$ |
| 12 | 66 | 451 | 5 | $6,3\ 10^{-5}$ | 4.23 | $6\ 10^{-4}$ |
| 13 | 67 | 227 | 2 | $1,6\ 10^{-4}$ | 1.57 | Faux |
| 14 | 115 | 421 | 17 | $4,3\ 10^{-8}$ | 15.37 | $4\ 10^{-5}$ |
| 15 | 185 | 1051 | 8 | $1,4\ 10^{-6}$ | 12.64 | $5\ 10^{-5}$ |
| 16 | 216 | 971 | 8 | $9,5\ 10^{-10}$ | 13.07 | exact |
| 17 | 216 | 971 | 29 | $1,3\ 10^{-6}$ | 63.35 | $7\ 10^{-5}$ |
| 18 | 343 | 1567 | 10 | $9\ 10^{-9}$ | 25.71 | exact |
| 19 | 512 | 2367 | 10 | $9\ 10^{-9}$ | 37 | exact |
| 20 | 1107 | 5664 | 8 | $9\ 10^{-9}$ | 342 | |

Dans ce tableau la colonne "résultat" provient de deux sortes d'analyses.

Une première qui est une comparaison avec les résultats de l'itération inverse pour des raisons explicitées au début de ce paragraphe. Une deuxième qui se sert de l'information suivante : toutes les composantes du vecteur résultat sont de même signe. Cette analyse est en fait le test de vraisemblance du sous programme ITIN.

Cette deuxième analyse intervient lorsque l'on n'est pas sûr du résultat donné par l'itération inverse.

Cette méthode a buté sur trois exemples avec un espace de projection de dimension Min(9,N-1) (N taille de la matrice). Mais en augmentant la taille de l'espace de projection deux de ces cas peuvent être résolus ; l'exemple 7 facilement (projection sur un espace de dimension 15) et l'exemple 9 difficilement (projection sur un espace de dimension 20 et mauvais résidu), l'exemple 13 semble trop difficile. Cette méthode est donc très sensible à la répartition des valeurs propres mais on va voir dans le paragraphe suivant que la résolution directe l'est infiniment plus.

3) Résultats de la résolution directe

Le tableau suivant répertorie les résultats par la méthode de résolution directe pour les examples déjà décrits pour la précédente méthode.

Caractéristiques de ces exemples :

N : nombre d'états du système, taille de la matrice

$\tau$ : nombre d'éléments non nuls de la matrice.

*Voir tableau page suivante.*

Dans ce tableau la colonne résidu ne devrait théoriquement ne comporter que des zéros, or les problèmes numériques prévus ont fait que cela n'est vrai que dans sept cas sur dix-sept.

La méthode par le même type d'analyse des résultats que dans le paragraphe précédent s'avère donner des résultats faux dans quatre cas et des résultats inacceptables pour la contrainte de précision dans trois cas. On voit donc que c'est une méthode peu fiable.

On peut remarquer aussi que nous avons deux cas de taille de matrices trop importante alors que pour l'itération inverse nous n'en avions qu'un; en fait, pour l'avant dernier exemple le sous programme de transformation de la matrice est responsable de ce fait. Il aurait été en fait facile de remedier à ce problème mais dans la mesure où la matrice correspondant à cet exemple ne présentait pas de difficulté, cet exemple n'a pas été traité.

De même l'exemple 2 n'a pu être traité, il correspondait à une matrice d'essai non issue d'une chaine de Markov irréductible (dont le graphe associé est connexe). Le remplacement de la dernière équation du système se trouvait alors injustifiable.

Le paragraphe suivant compare les trois méthodes utilisées, plus particulièrement le temps de calcul qu'elles nécessitent.

| Exemple N° | Données N | Données $\tau$ | Précision de la décomposition | Temps de calcul (s) | Résidu | Résultat |
|---|---|---|---|---|---|---|
| 1 | 4 | 7 | $2,2\ 10^{-19}$ | 0.50 | 0 | exact |
| 2 | 6 | 16 | Matrice non issue d'une chaîne de Markov | | | |
| 3 | 8 | 26 | $2,2\ 10^{-19}$ | 0.48 | 0 | exact |
| 4 | 8 | 26 | $2,2\ 10^{-19}$ | 0.32 | 0 | exact |
| 5 | 16 | 58 | $3,5\ 10^{-18}$ | 0.51 | 0 | exact |
| 6 | 23 | 71 | $3,5\ 10^{-19}$ | 0.67 | 0 | exact |
| 7 | 31 | 91 | $2,2\ 10^{-19}$ | 0.86 | $3,2\ 10^{-5}$ | faux |
| 8 | 31 | 105 | $3,5\ 10^{-18}$ | 0.97 | $2,4\ 10^{-6}$ | $4\ 10^{-3}$ |
| 9 | 37 | 131 | $3,5\ 10^{-18}$ | 1.19 | $3,7\ 10^{-5}$ | faux |
| 10 | 54 | 250 | $3,5\ 10^{-18}$ | 2.32 | $10^{-4}$ | $2\ 10^{-4}$ |
| 11 | 66 | 286 | $3,5\ 10^{-18}$ | 3.7 | $3,2\ 10^{-7}$ | $3\ 10^{-4}$ |
| 12 | 66 | 451 | $5,6\ 10^{-17}$ | 3.49 | $2,7\ 10^{-7}$ | exact |
| 13 | 67 | 227 | $3,5\ 10^{-18}$ | 2.69 | $6\ 10^{-5}$ | faux |
| 14 | 115 | 421 | $3,5\ 10^{-18}$ | 6 | $5\ 10^{-8}$ | exact |
| 15 | 185 | 1051 | $3,5\ 10^{-18}$ | 47 | $5\ 10^{-6}$ | $5\ 10^{-5}$ |
| 16 | 216 | 971 | $3,5\ 10^{-18}$ | 43.5 | 0 | exact |
| 17 | 216 | 971 | $3,5\ 10^{-18}$ | 40.2 | 1,003 | faux |
| 18 | 343 | 1567 | $5,6\ 10^{-17}$ | 93.9 | 0 | exact |
| 19 | 512 | 2367 | Matrice trop importante | | | |
| | 1107 | 5664 | Matrice trop importante | | | |

## III - COMPARAISON DES METHODES

En ce qui concerne la place mémoire réclamée par chaque méthode, on peut dire qu'elle est identique dans les trois logiciels ; pour la taille du code il en va très différemment: en effet le sous programme IOMSP (Méthode d'Arnoldi) est plus "petit" que les deux autres (de l'ordre de deux fois).

### 1) Comparaison des résidus

A la lecture des tableaux de résultats on remarque tout de suite deux phénomènes concernant les résidus :

- les résidus de la méthode de résolution directe sont ou très bons ou mauvais
- les résidus de la méthode d'itération inverse sont toujours meilleurs que ceux obtenus par Arnoldi.

La première remarque permet de conclure à la très mauvaise fiabilité de la méthode de résolution directe. La deuxième doit être nuancée par les faits suivants : l'itération inverse converge presque toujours en une seule itération; aussi, pour obtenir des résidus équivalents par la méthode d'Arnoldi, il aurait fallu notamment dans les exemples "petits" un trop grand nombre d'itérations. On aurait pu aussi augmenter la taille de l'espace de projection pour améliorer les résidus obtenus. Mais tout ceci n'aurait eu pour effet que d'augmenter le temps de calcul de cette méthode pour un gain de résidu non nécessaire, vu la contrainte de précision.

Il faut tout de même remarquer avant de comparer la vitesse de ces trois méthodes que c'est l'itération inverse qui est le programme le plus fiable du point de vue résultat, mais que quelle que soit la difficulté des exemples à traiter, seule la méthode d'Arnoldi permet d'aller vers les grandes tailles.

2) Comparaison des temps de calcul

Cette comparaison est faite à l'aide des graphiques suivants, qui représentent les temps d'exécution des trois méthodes en fonction de la taille de la matrice, puis du nombre d'éléments non nuls de celle-ci.

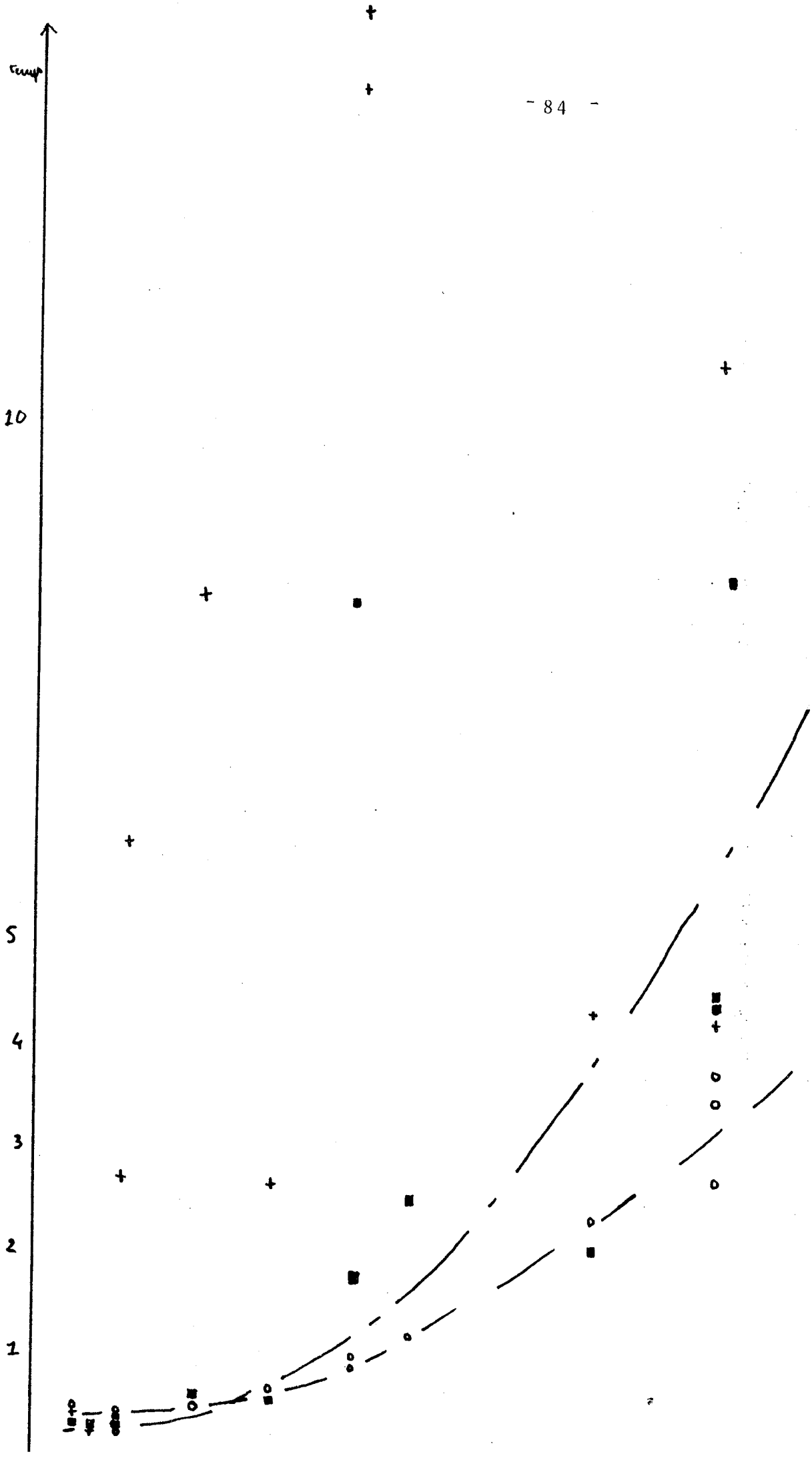
Temps d'exécution des méthodes
de résolution directe , d'Arnoldi, d'itération
inverse en fonction de la taille de la matrice.

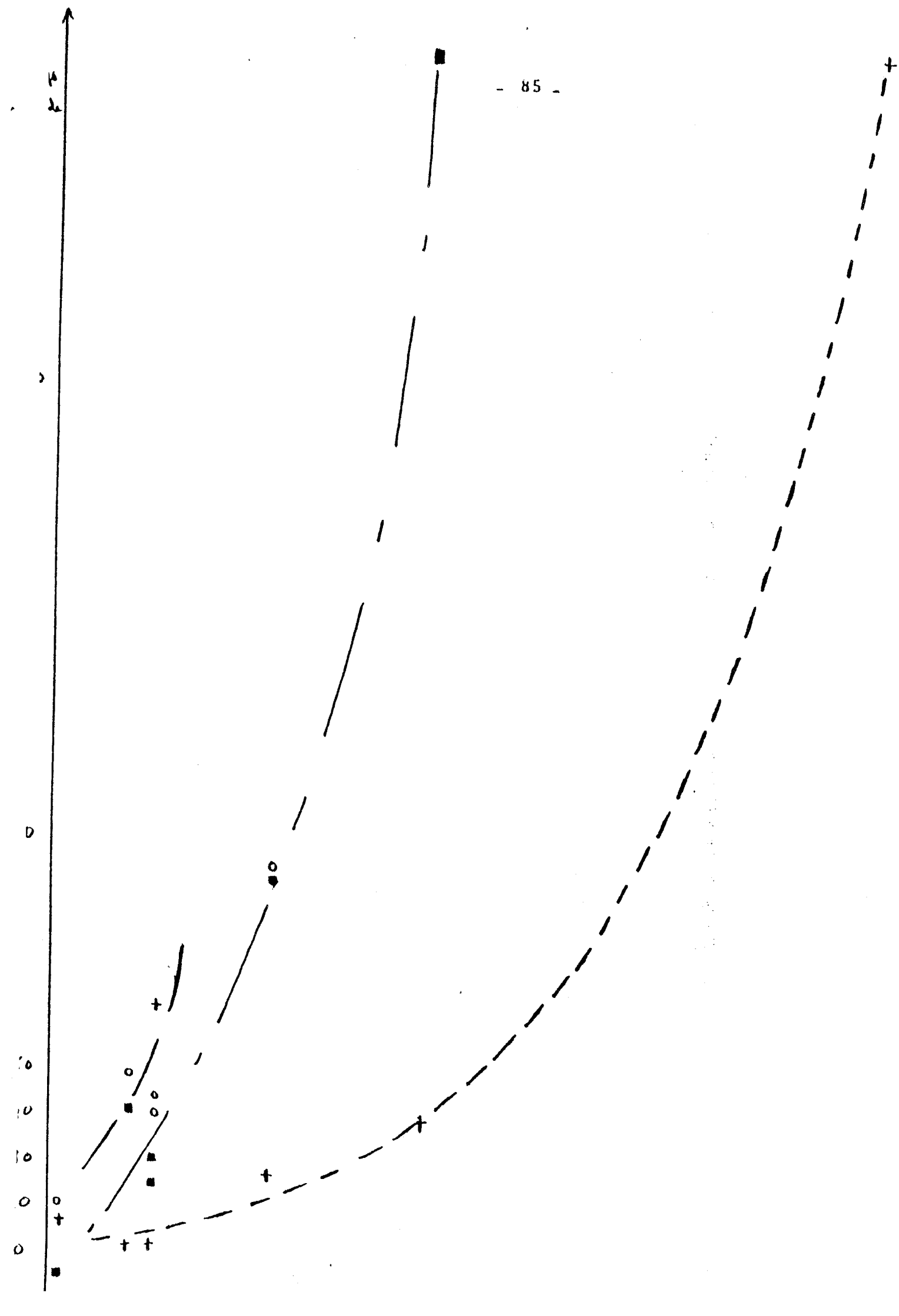
N : taille de la matrice

o ——— ——— ——— méthode directe

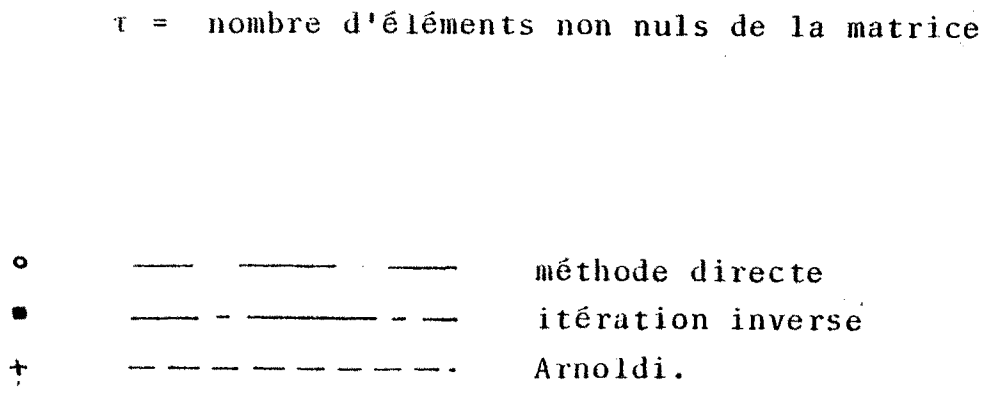
■ ——— . ——— . — itération inverse

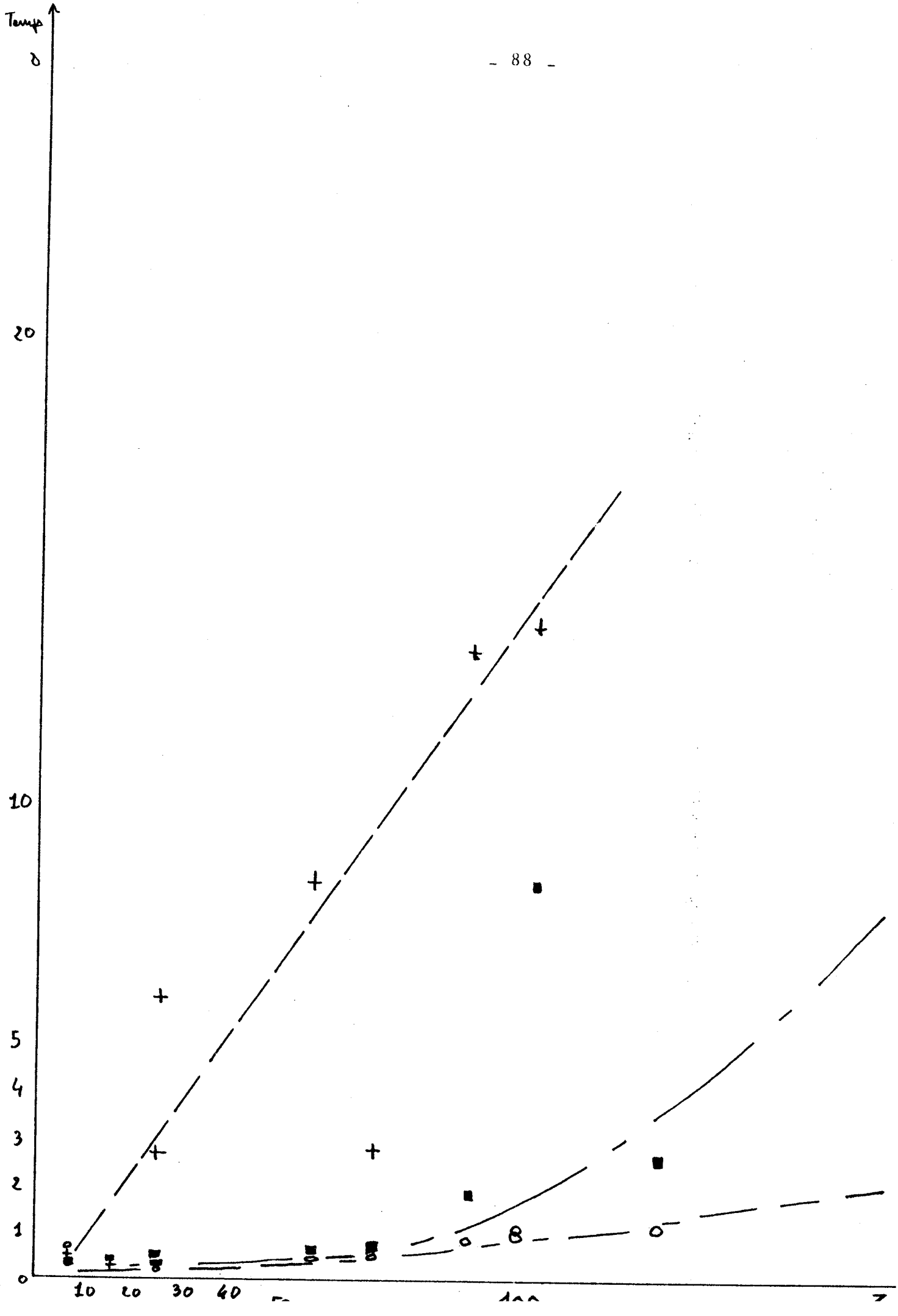
\+ _ — — — — — — — — Arnoldi

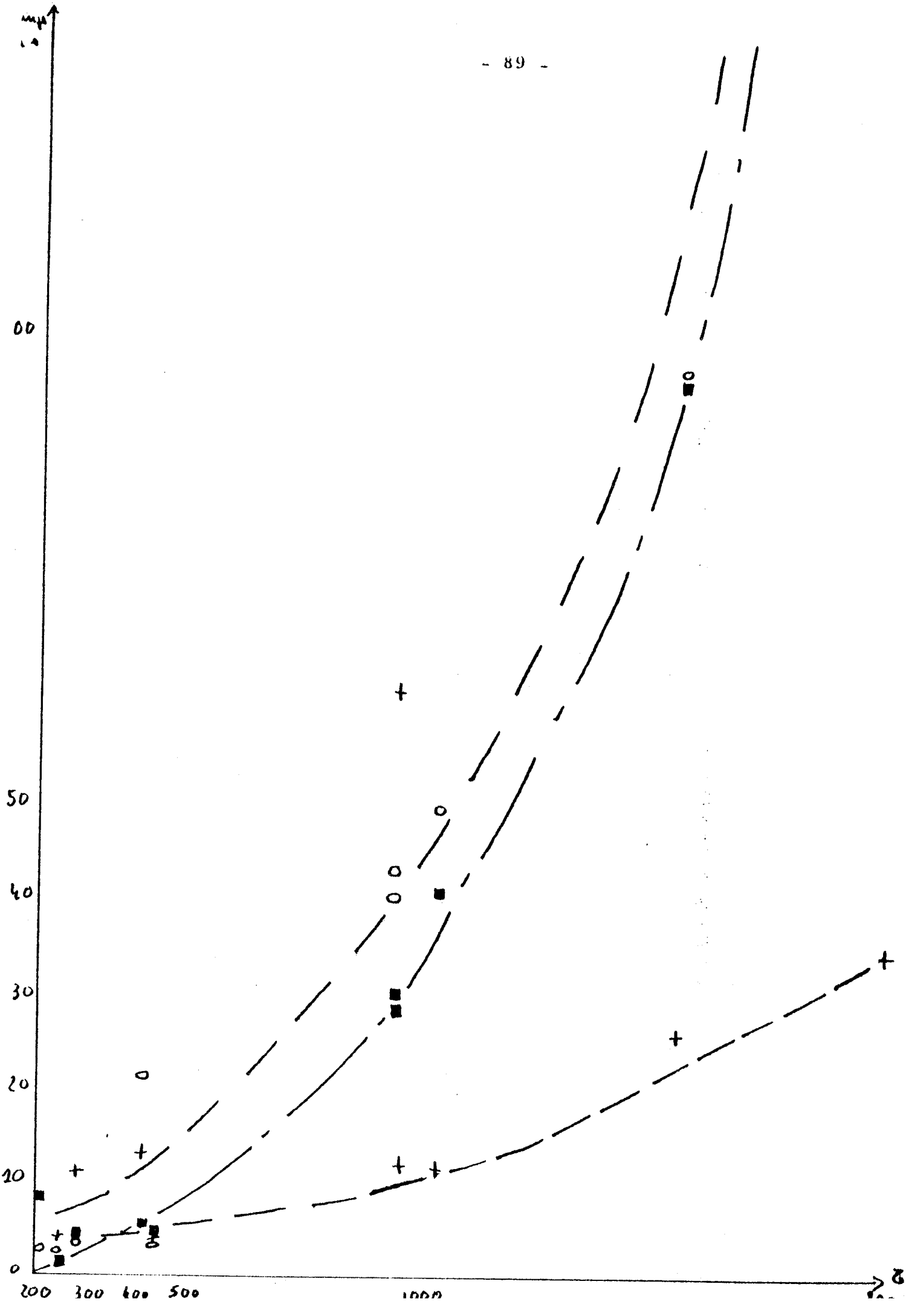

Temp
10
5
4
3
2
1


μ
λ

Temps d'exécution des
méthodes de résolution directe, d'Arnoldi et
d'itération inverse en fonction du nombre
d'éléments non nuls de la matrice




Temp
δ
20
10
5
4
3
2
1
0
10
20
30
40


00
50
40
30
20
10
0
200
300
400
500

De ces graphiques, il ressort que, quelle que soit la méthode, le temps d'exécution dépend plus du nombre d'éléments non nuls que de la taille de la matrice.

D'autre part, pour les méthodes de résolutions directes, les courbes obtenues sont voisines. C'est fort logique puisque le temps d'exécution de la méthode d'itération inverse est pratiquement celui de la résolution d'un système ayant pour taille celle du problème considéré (on a un seul système à résoudre, sauf dans un cas). La méthode de résolution directe semble plus rapide que celle d'itération inverse pour les matrices de petite taille, légèrement plus lente pour les matrices plus grandes, mais cette différence reste faible et ne permet pas de conclure en faveur de l'une ou l'autre des deux méthodes.

Cela s'explique peut-être par le fait que dans la méthode d'itération, on a plus d'opérations initiales à effectuer et par la suite le système étant moins difficile à résoudre, pour les grandes tailles la méthode devient plus rapide que celle de résolution directe.

La loi du temps de calcul de la méthode d'Arnoldi par rapport à la taille de la matrice semble difficile à discerner pour les petites tailles, pour devenir facile pour des tailles plus importantes. Cela provient essentiellement du jeu d'essais que nous avons choisi, il n'y a de cas difficiles que dans les tailles relativement peu importantes, sauf pour l'exemple 17.

Dans le cas des exemples difficiles comportant un petit nombre d'éléments non nuls, le temps de calcul d'Arnoldi semble dépendre linéairement de ce nombre, et surtout est beaucoup plus important que le temps de calcul des deux autres méthodes. Ceci est encore vérifié pour l'exemple 17, où le point correspondant est très proche de la droite tracée.

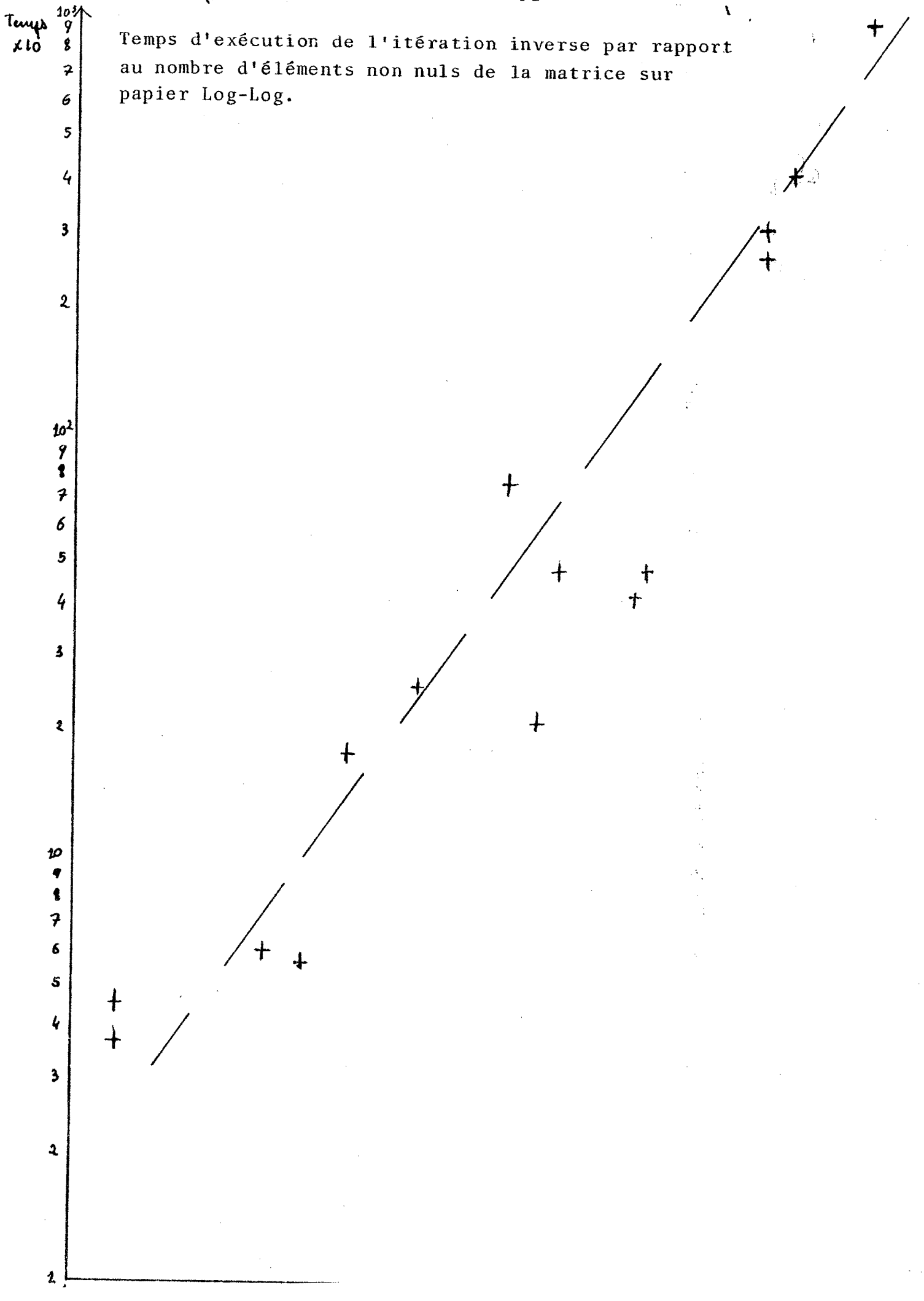
Dans les exemples faciles la méthode d'Arnoldi va beaucoup plus vite que les deux autres, d'autant plus que la taille ou le nombre d'éléments non nuls augmente.

Si la vitesse d'exécution de la méthode d'Arnoldi est au pire linéaire en fonction de ce nombre d'éléments non nuls, comme le prouvent les deux graphiques suivants, pour les deux autres méthodes ces temps sont presque des fonctions puissance de ce nombre.

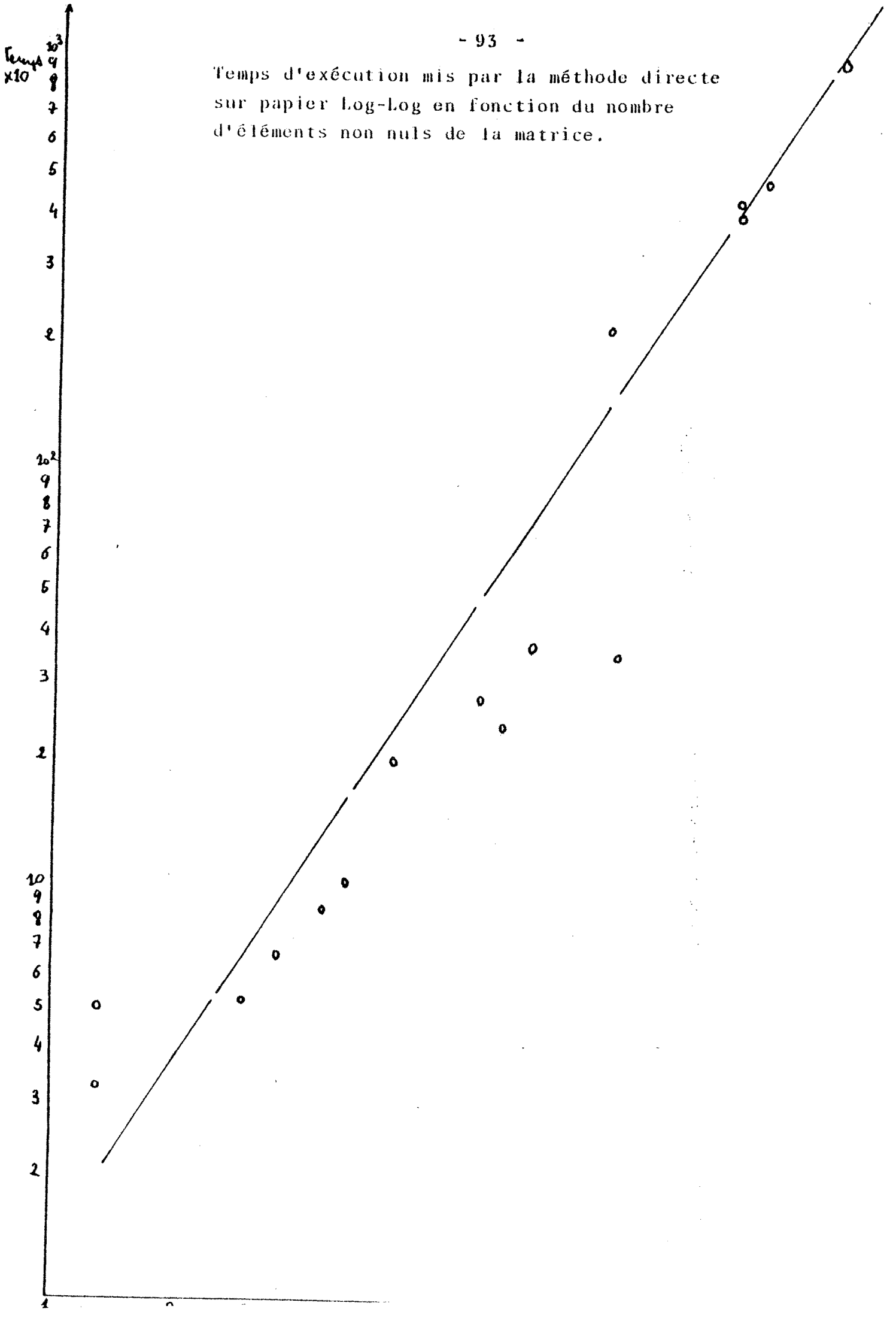
Ce qui vérifie partiellement la théorie puisque la résolution de système linéaire est de l'ordre de $k_1 * \tau^2 / n$ où $\tau$ est le nombre d'éléments non nuls et $k_1$ une constante "petite". [17] .

Donc, de toutes manières, même pour les cas difficiles la méthode d'Arnoldi pour une taille suffisamment grande (pour une taille < 500) serait plus rapide que celle d'itération inverse.

Temps d'exécution de l'itération inverse par rapport au nombre d'éléments non nuls de la matrice sur papier Log-Log.



Temps d'exécution mis par la méthode directe sur papier Log-Log en fonction du nombre d'éléments non nuls de la matrice.



## IV - CONCLUSION

Toutes ces comparaisons ont permis de mettre en valeur les qualités et les défauts de la méthode d'Arnoldi. En effet, la conclusion à tirer de toutes ces informations est que la méthode d'Arnoldi est face à ce problème surement la meilleure méthode, tout juste pourrait-on suggérer qu'une méthode de résolution de système itérative à l'intérieur de l'itération inverse serait à tester. Mais en fait le temps d'exécution de ce type de méthode est trop grand pour pouvoir espérer véritablement un gain.

Dans le cadre d'un problème très difficile, la méthode d'Arnoldi peut être mise en échec (exemple 13) mais dans les autres cas difficiles en testant la valeur du résidu en sortie on peut facilement déterminer s'il est nécessaire d'augmenter la taille de projection. En fait tout le problème de cette méthode réside dans le fait que la non connaissance du spectre de la matrice ne permet pas d'avoir une idée de la taille optimale de l'espace de projection. On peut tout de même retenir qu'une taille de l'ordre de la dizaine semble bonne.

Il aurait été intéressant de tester une meilleure méthode de résolution de système (avec un logiciel plus gros) mais de toute manière il sera difficile pour cette méthode de diviser le temps de calcul de l'itération inverse par 10 (exemple 19).

Les limites de la méthode d'Arnoldi pour un temps de calcul raisonnable semblent de toute manière être un nombre d'éléments non nuls de la matrice de l'ordre de 7000 à 8000 .

# ANNEXE I

## RÉSULTATS POUR L'EXEMPLE 11

Résultats obtenus par les
méthodes de résolution directe, d'Arnoldi et
d'itération inverse pour l'exemple 11 .

Pour chaque méthode le terme le moins vraisemblable (à cause du signe) a été souligné on donne maintenant l'écart maximal sur chaque composante du vecteur entre les trois méthodes

Arnoldi - itération inverse : $5\ 10^{-4}$

Résolution directe -
itération inverse : $2\ 10^{-4}$

Arnoldi - Résolution directe : $7\ 10^{-4}$

```
**********************************************************************
*                                                                    *
*                  RESULTATS DE RESOL                                *
*                  pour le fichier11                                 *
*                                                                    *
**********************************************************************

 erreur dans la decomposition:    0.3469446951953614188D-17
 ------------------------------------------------------------

   0.9880469581     1    0.0041399746     2    0.0000140309     3    0.0078850450     4   -0.0000013960     5
   0.0000265444     6   -0.0000006904     7   -0.0000027325     8    0.0000500244     9   -0.0000003360    10
  -0.0000013486    11   -0.0000054216    12   -0.0000001659    13   -0.0000006561    14   -0.0000026676    15
  -0.0000108282    16   -0.0000000825    17   -0.0000003240    18   -0.0000012972    19   -0.0000053094    20
  -0.0000000409    21   -0.0000001610    22   -0.0000006403    23   -0.0000025802    24   -0.0000105992    25
  -0.0000000196    26   -0.0000000799    27   -0.0000003182    28   -0.0000012734    29   -0.0000051474    30
  -0.0000000075    31   -0.0000000385    32   -0.0000001582    33   -0.0000006329    34   -0.0000025396    35
  -0.0000102836    36   -0.0000000151    37   -0.0000000767    38   -0.0000003149    39   -0.0000012622    40
  -0.0000050722    41   -0.0000000308    42   -0.0000001538    43   -0.0000006286    44   -0.0000025207    45
  -0.0000101370    46   -0.0000000637    47   -0.0000003091    48   -0.0000012564    49   -0.0000050373    50
  -0.0000001324    51   -0.0000006217    52   -0.0000025121    53   -0.0000100691    54   -0.0000002758    55
  -0.0000012499    56   -0.0000050232    57   -0.0000005748    58   -0.0000025101    59   -0.0000100434    60
  -0.0000011987    61   -0.0000050345    62   -0.0000025001    63   -0.0000100822    64   -0.0000052151    65
  -0.0000108793    66

   residu =0.0000003152
   --------------------

 temps d'execution =3,661589 s
 -----------------------------
```

```
**************************************************************************
*                                                                        *
*                    RESULTATS DE IOMSP                                  *
*                    pour le fichier11                                   *
*                                                                        *
**************************************************************************
```

```
 * * * * * * * * * *ITER= 17 * * * * * * * * * *

  0.987316D+00     0.427378D-02     0.182247D-04     0.814862D-02   -0.355819D-05
  0.339385D-04     0.136203D-05    -0.676295D-05     0.659171D-04    0.347213D-06
  0.121930D-05    -0.140742D-04     0.606271D-06     0.114366D-05    0.184503D-05
 -0.279398D-04     0.106972D-05     0.817278D-06     0.318279D-06    0.195721D-05
  0.562993D-06     0.930346D-06     0.129346D-05     0.704627D-06    0.843474D-06
  0.603923D-06     0.265316D-05     0.276605D-05     0.838852D-06   -0.563830D-07
 -0.171270D-07     0.548769D-06     0.694567D-06     0.174788D-05    0.662291D-06
 -0.174353D-05     0.146378D-06     0.157506D-05     0.376023D-05    0.266588D-05
 -0.147910D-05     0.335558D-06     0.154643D-05     0.278462D-05    0.498344D-06
 -0.415288D-05     0.855458D-06     0.228211D-05     0.176649D-05   -0.226643D-05
  0.648689D-06     0.881777D-06     0.665388D-06    -0.821324D-05    0.723010D-06
  0.280137D-05    -0.302339D-05    -0.633830D-07    -0.134697D-05   -0.923266D-05
  0.631526D-07    -0.163142D-05    -0.659518D-06    -0.580985D-05    0.799508D-07
 -0.145775D-05

res     0.00000434404765
-------------------------

temps d' execution =10,652644 s
-------------------------------
```

```
**********************************************************************
*                                                                    *
*                    RESULTATS DE ITIN                               *
*                    pour le fichier11                               *
*                                                                    *
**********************************************************************

 xmu:0.100000E-10 zeps:0.100000E-04

 erreur dans la decomposition:     0.34694469519536141888D-17
 ---------------------------------------------------

       0.98779    1        0.00414    2        0.00001    3        0.00788    4        -0.00000    5
       0.00003    6       -0.00000    7       -0.00000    8        0.00005    9        -0.00000   10
      -0.00000   11       -0.00000   12       -0.00000   13       -0.00000   14        -0.00000   15
      -0.00001   16       -0.00000   17       -0.00000   18       -0.00000   19        -0.00000   20
      -0.00000   21       -0.00000   22       -0.00000   23       -0.00000   24        -0.00001   25
      -0.00000   26       -0.00000   27       -0.00000   28       -0.00000   29        -0.00000   30
      -0.00000   31       -0.00000   32       -0.00000   33       -0.00000   34        -0.00000   35
      -0.00001   36       -0.00000   37       -0.00000   38       -0.00000   39        -0.00000   40
      -0.00000   41       -0.00000   42       -0.00000   43       -0.00000   44        -0.00000   45
      -0.00001   46       -0.00000   47       -0.00000   48       -0.00000   49        -0.00000   50
      -0.00000   51       -0.00000   52       -0.00000   53       -0.00001   54        -0.00000   55
      -0.00000   56       -0.00000   57       -0.00000   58       -0.00000   59        -0.00000   60
      -0.00000   61       -0.00000   62       -0.00000   63       -0.00000   64        -0.00000   65
       0.00000   66

  residu test1 = 0.478420E-08
  ---------------------------

 temps d' execution = 4,742476 s
  ------------------------------
```

# ANNEXE II

## MÉTHODE D'ARNOLDI

```
      SUBROUTINE IOMSP(N,S,V,H,B,VECTP,VCP,Y,R,PERM,ITMAX,EPS)
C*****************************************************************
C                                                                *
C RESOLUTION MATRICIELLE ASSOCIEE A L 'ANAKYSE MARKOVIENNE       *
C                     METHODE D' ARNOLDI                         *
C                                                                *
C*****************************************************************
C
      IMPLICIT DOUBLE PRECISION (B-H,O-Z)
      INTEGER N,S
      DIMENSION V(N,S),H(S,S),B(S,S)
      DIMENSION Y(1),R(1),VECTP(1),VCP(1)
       REAL EPS
      LOGICAL PERM(1)
      COMMON/DO/A(7000),NN(7000)
       COMMON L(6),TRACE
       LOGICAL TRACE
C
C*****************************************************************************
C A TABLEAU DE REELS DOUBLES  DE DIMENSION LE NOMBRE D'ELEMENTS NON NULS
C          DE LA MATRICE
C A CONTIENT LES ELEMENTS DE LA MATRICE CREUSE
C
C NN TABLEAU D'ENTIERS DE MEME DIMENSION QUE A
C NN CONTIENT  LES INDICES DE COLONNES DES ELEMENTS  NON NULS
C      DE LA MATRICE CREUSE STOCKEE DANS A
C
C N ORDRE DE LA MATRICE A
C
C S TAILLE DE L'ESPACE DE KRYLOV
C
C V TABLEAU DE TRAVAIL DE DIMENSION (N,S)
C
C H TABLEAU DE TRAVAIL  DE DIMENSION (S,S)
C
C B TABLEAU DE TRAVAIL DE DIMENSION (S,S)
C
C VECTP TABLEAU DE DIMENSION (N),EN SORTIE CONTIENT LE VECTEUR PROPRE CHERCHE
C
C Y VECTEUR DE TRAVAIL DE DIMENSION (N)
C
C R VECTEUR DE TRAVAIL DE DIMENSION (2S)
C
C PERM TABLEAU DE LOGIQUES DE DIMENSION (2S),TABLEAU DE TRAVAIL
C
C ITMAX ENTIER CONTIENT LE NOMBRE ITERATIONS MAXIMUM DE SCHMID
C
C EPS REELS DOUBLE CONTIENT LA PRECISION DEMANDEE AU VECTEUR PROPRE
C
C*****************************************************************************
C
      WRITE(6,104)N,S,ITMAX
      TRACE=.TRUE.
      IF(TRACE) WRITE(6,104) N,S
      IF(TRACE) WRITE(21,104) N,S
 104  FORMAT (214)
      T=DSQRT(1.DO/DBLE(FLOAT(N)))
      DO 5 I=1,N
 5    Y(I)=T
C
```

```
C  INITIALISATION DE Y ..........
C
C
       ITER=0
40     ITER=ITER+1
       IF(ITER.GT.10) GOTO 52
   99 FORMAT( 6E13.6)
       DO 50 K=1,5
       CALL XATX(N,Y,VECTP)
        CALL XATX(N,VECTP,Y)
50     CONTINUE
       T=DSQRT(PS(N,Y,Y))
       DO 51 K=1,N
51     Y(K)=Y(K)/T
       CALL XATX(N,Y,VECTP)
       T=PS(N,VECTP,Y)
       IF(DABS(T-1.D0).GE.0.1) GOTO 40
52     CONTINUE
C
C FIN INITIALISATION DE Y ....
C
       ITER=0
       RES=EPS*10.D0
 2     CALL SCHMID(N,1,V,Y,R,1.D0,3.D0)
       ITER=ITER+1
       IF(TRACE) WRITE(6,108) ITER
       IF(TRACE) WRITE(21,108) ITER
 108   FORMAT(1H ,///,10(2H *),'ITER=',I3,10(2H *)//)
C
C DEBUT DE GRANDE BOUCLE ............
C
       IS=S-1
       DO 310 I=1,S
 310   VCP(I)=0.D0
       TOL=1.D-15
       I=0
C
C BOUCLE D'APPEL DE SCHMID   ............
C
 4     I=I+1
       CALL XATX(N,V(1,I),Y)
C
C     TEST DE CONVERGENCE NUMERO1
C
       XYY=0.D0
       DO 78 ILE=1,N
       XYW=DABS(Y(ILE)-V(ILE,I))
   78  IF(XYY.LT.XYW) XYY=XYW
       IF (XYY.GT.EPS) GOTO 74
       DO 75 ILE=1,N
   75  VECTP(ILE)=Y(ILE)
       WRITE(6,76) XYY
  76   FORMAT( 14H NOUVEAU TEST ,E13.6,6H =RES )
       WRITE(21,76) XYY
       RETURN
C
C FIN DU 1ER TEST
C
   74  I1=I+1
       OMEGA=1.D0
```

```
      TETA=2.DO + 10.D0*(DBLE(FLOAT(I))/DBLE(FLOAT(IS)))**2
      CALL SCHMID(N,I1,V,Y,R,OMEGA,TETA)
      DO 6 K=1,I1
    6 H(K,I)=R(K)
      IF (I .LT. IS) GOTO 4
C
C FIN DE LA BOUCLE DE SCHMID     ..........
C
      XLAM=1.DO
      DELTA=DABS(RES)*0.1DO
      CALL RQID(I,S,H,B,XLAM,DELTA,VCP,IFAIL,PERM)
C
C TEST DE CONVERGENCE NUMERO2
C
      RES=DABS(H(I+1,I)*VCP(I))
      IF(TRACE) WRITE(6,133) ITER,RES,XLAM
      IF(TRACE) WRITE(21,133) ITER,RES,XLAM
  133 FORMAT (' ITER=',I3,'RESIDU=',D24.14,'   V.P. = ',D25.15,//)
C     ***   CALCUL DES VECTEURS DE RITZ  ***
      DO 340 I=1,N
      T=0.DO
      DO 350 K=1,IS
  350 T=T+V(I,K)*VCP(K)
      VECTP(I)=T
340   CONTINUE
      IF ( RES .LT. EPS) GOTO 433
      IF (ITER .GT. ITMAX) GOTO 432
C
C FIN TEST2
C
C ** DEBUT REINITIALISATION
      T=DSQRT(PS(N,VECTP,VECTP))
      DO 66 I=1,N
   66 Y(I)=VECTP(I)/T
      GOTO  2
C **  FIN  REINITIALISATION
C
C FIN GRANDE BOUCLE ..............
C
  432 WRITE(6,109) ITMAX
109   FORMAT(' *** WARNING: NO CONVERGENCE AFTER ',I3,' ITERATIONS')
      RETURN
  433  EPS=RES
      T=0.DO
      DO 434 K=1,N
  434 IF( DABS(VECTP(K)) .GT. DABS(T)) T=VECTP(K)
      DO 435 K=1,N
  435 VECTP(K)=VECTP(K)/T
      WRITE(6,1000) ITER
1000  FORMAT(1H ,I4,' ITERATIONS ')
      RETURN
      END
      SUBROUTINE XATX(N,X,Y)
      IMPLICIT DOUBLE PRECISION (B-H,O-Z)
      DIMENSION X(1),Y(1)
      COMMON/DO/A(7000),NN(7000)
C                                    t
C XATX  RANGE DANS Y LE PRODUIT A *X
C
      DO 1 K=1,N
```

```
 1      Y(K)=0.DO
        K=1
        II=0
        DO 3 I=1,N
        T=X(I)
        JMAX=NN(K)
        DO 2 J=1,JMAX
        II=II+1
        K=K+1
        JJ=NN(K)
 2      Y(JJ)=Y(JJ)+T*A(II)
 3      K=K+1
        RETURN
        END
        SUBROUTINE SCHMID(N,I,Q,Y,R,OMEG,TETA)
        IMPLICIT DOUBLE PRECISION (B-H,O-Z)
        DIMENSION Q(N,I)
        DIMENSION  Y(1),R(1)
        LOGICAL LO,LS
C
C  CETTE PROCEDURE  ORTHOGONALISE LE VECTEUR Y  PAR RAPPORT
C     AUX (I-1) VECTEURS COLONNES DU TABLEAU Q QUI FORMENT UNE
C      BASE ORTHONORMALE DE L'ESPACE DE KRYLOV.ELLE RANGE CE
C      VECTEUR  AINSI ORTHOGONALISE DANS Q(.,I)
C
C  PARAMETRES
C
C  N ENTIER DIMENSION DU PROBLEME
C
C  I ENTIER (I-1) NOMBRE  DE VECTEURS PAR
C   RAPPORT AUXQUELS IL FAUT ORTHOGONALISER
C
C  Q TABLEAU DE DIMENSION (N,I),LES (I-1) PREMIERES COLONNES  SONT
C   LES VECTEURS PAR RAPPORT AUXQUELS ON ORTHOGONALISE
C
C  Y VECTEUR DE DIMENSION (N) ,VECTEUR A ORTHOGONALISER
C
C  R TABLEAU DE TRAVAIL DE DIMENSION (2S),CONTIENT LES
C   COEFF D' ORTHOGONALISATION
C
C  OMEG,TETA  TETA>1,OMEG>0
C
        I1=I-1
        DO 1 K=1,I1
      1 R(K)=0.DO
        RO=PS(N,Y,Y)
        ROO=RO
        IF (I1.NE.0) GOTO  11
        RO1= RO
        GOTO 9
     11 IT=0
C  **BOUCLE ITERATIVE D'ORTHOGONALISATION
     10 DO 2 J=1,N
        LO=.TRUE.
      2 Q(J,I)=0.
        DO 3 K=1,I1
        SS=0.DO
        DO 4 J=1,N
      4 SS=SS+Q(J,K)*Y(J)
C LO TESTE LA VALIDITE NUMERIQUE
```

```
      IF (SS.GT.(0.1E-12)) LO=.FALSE.
C     IF (SS.LT.(0.1E-18)) SS=0.E+00
      R(I+K)= SS
      DO 5 J=1,N
    5 Q(J,I)= Q(J,I) + SS* Q(J,K)
    3 CONTINUE
      DO 6 K=1,I1
    6 R(K)=R(K)+R(I+K)
      XX=0.D0
      DO 7 J=1,N
      Y(J)=Y(J)-Q(J,I)
      XY=DABS(Y(J))
    7 IF (XX.LT.XY) XX=XY
      IF (XX.GT.0.1D-10) GOTO 13
      DO 20 ILE=1,N
   20 Y(ILE)=Y(ILE)/XX
      TETA=TETA*XX
   13 RO1=PS(N,Y,Y)
      T=0.
      IF(LO) GOTO 9
      DO 8 K=1,I1
    8 T=T+ R(I+K)**2
      IT=IT+1
C    ** TEST POUR LE RETOUR  EN 10 **
      IF((ROO +OMEG*T).LT.(TETA*RO1)) GOTO 9
      ROO=RO1
      IF (IT .LT.20) GOTO 10
      WRITE(6,100)
  100 FORMAT('  .. PLUS DE 20 ITS DANS SCHMIDT ..',/)
    9 RO1=DSQRT(RO1)
      DO 12 J=1,N
   12 Q(J,I)= Y(J)/RO1
      R(I)=RO1
   14 RETURN
      END
      FUNCTION PS(N,X,Y)
      IMPLICIT DOUBLE PRECISION (B-H,O-Z)
      DIMENSION X(N),Y(N)
C
C PRODUIT SCALAIRE DE X PAR Y
C
      T=0.0D0
      DO 1 K=1,N
    1 T=T+X(K)*Y(K)
      PS=T
      RETURN
      END
      SUBROUTINE RQIO(N,NA,A,B,WR,DELTA,RV,IFAIL,PERMUT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(NA,N),B(NA,N),RV(N)
      LOGICAL PERMUT(1)
       COMMON L(6),TRACE
       LOGICAL TRACE
      DOUBLE PRECISION NORM,NORMV
C
C     CETTE PROCEDURE CALCULE PAR LA METHODE DU QUOTIENT DE RAYLEIGH
C      LE VECTEUR PROPRE RV  ASSOCIE A LA VALEUR PROPRE LA PLUS PROCHE
C      DE WR DE LA MATRICE HESSENBERG SUPERIEURE A
C
C     A MATRICE  HESSENBERG SUPERIEURE
```

```
C
C     B TABLEAU DE TRAVAIL DE DIMENSION (N,NA)
C
C   WR REEL ESTIMATION DE LA VALEUR PROPRE
C
C     DELTA ESTIMATION DE L'ECART DE LA VALEUR PRPRE AVEC LE RESTE DU SPEC
C
C  RV VECTEUR DE DIMENSION (N) CHERCHE
C
C  PERMUT TABLEAU DE LOGIQUES DE DIMENSION (N) ,TABLEAU DE TRAVAIL
C
C     COMPUTE NORM OF MATRIX A
      NORM=0.D0
      MP=1
      DO 1 I=1,N
      X=0.D0
      DO 2 J=MP,N
 2    X=X+DABS(A(I,J))
      IF (X .GT. NORM) NORM=X
      MP=I
 1    CONTINUE
C     INITIALISE CONSTANTS
C     DELTA=DELTA*0.5D0
      EPS3=NORM*0.1D-09
      GROWTO=0.1D0
      XQR=WR
      ITS=0
      RLAMBD=WR
      EPS4=(DELTA/WR)**2
C   ***   INITIALIZE STARTING VECTOR RV  ***
      NORM=0.D0
      DO 11 I=1,N
      IF (NORM .LT. DABS(RV(I))) NORM=DABS(RV(I))
 11   CONTINUE
      IF (NORM .NE. 0.D0) GOTO 110
      DO 111 I=1,N
 111  RV(I)=EPS3
      GOTO 240
 110  NORMV=EPS3/NORM
      DO 12 I=1,N
 12   RV(I)=RV(I)*NORMV
 240  MP=1
C   ***    COMPUTE B = A - RLAMBD * ID  BEFORE DECOMPOSITION
      DO 320 I=1,N
      DO 300 J=MP,N
 300  B(I,J)=A(I,J)
      B(I,I)=B(I,I)-RLAMBD
      MP=I
 320  CONTINUE
C    ***    DECOMPOSITION OF B ***
C
      ITS=ITS+2
      DO 400 I=2,N
      MP=I-1
C    ***    TEST FOR INTERCHANGE  ***
      PERMUT(MP)=(DABS(B(I,MP)) .GT. DABS(B(MP,MP)))
      IF (.NOT. PERMUT(MP)) GOTO 360
      Y=RV(I)
      RV(I)=RV(MP)
      RV(MP)=Y
```

```
      DO 340 J=MP,N
      Y=B(I,J)
      B(I,J)=B(MP,J)
 340  B(MP,J) = Y
 360  IF(B(MP,MP) .EQ. 0.D0) B(MP,MP)=EPS3
      X=B(I,MP)/B(MP,MP)
      B(I,MP)=X
      IF (X .EQ. 0.D0) GOTO 400
      DO 380 J=I,N
 380  B(I,J)=B(I,J)-X*B(MP,J)
      RV(I)=RV(I)-X*RV(MP)
 400  CONTINUE
C     ***   END DECOMPOSITION NOW  BACK- SUBSTITUTE
      IF(B(N,N) .EQ. 0.D0) B(N,N)=EPS3
 401  ITS=ITS+1
      DELTA=DELTA*1.2D0
      DO 500 II=1,N
      I=N+1-II
      Y=RV(I)
      IF (I.EQ.N) GOTO 480
      IP1=I+1
      DO 460 J=IP1,N
 460  Y=Y-B(I,J)*RV(J)
 480  RV(I)=Y/B(I,I)
 500  CONTINUE
C     ***   COMPUTE INF. NORM  AND EUCL. NORM  OF RV
      NORM=0.D0
      NORMV=0.D0
      DO 780 I=1,N
      X=DABS(RV(I))
      IF (NORMV .GE. X ) GOTO 760
      NORMV=X
 760  NORM=NORM+X*X
 780  CONTINUE
      NORM=DSQRT(NORM)
      DO 820 I=1,N
 820  RV(I)=RV(I)/NORM
      X1=XQR
      XQR=0.D0
      Y=0.D0
      MP=1
      DO 842 I=1,N
      S=0.D0
      DO 841 J=MP,N
 841  S=S+A(I,J)*RV(J)
      MP=I
      T=DABS(X1*RV(I)-S)
      IF (T .GT. Y) Y=T
      XQR=XQR+S*RV(I)
 842  CONTINUE
      RLAMBD=XQR
      IF (NORM .GE. GROWTO) GOTO 1000
      IF (Y .LT. EPS3) GOTO 1000
      IF (ITS .GT. 100) GOTO 1001
      DO 843 I=1,N
 843  RV(I)=RV(I)*EPS3
      IF(DABS(XQR-WR) .GE. DELTA) GOTO 850
      GOTO 240
 850  DO 844 I=2,N
      MP=I-1
```

```
      IF (.NOT. PERMUT(MP)) GOTO 844
      Y=RV(I)
      RV(I)=RV(MP)
      RV(MP)=Y
844   CONTINUE
      MP=1
      DO 851 I=2,N
      RV(I)=RV(I) - B(I,MP)*RV(MP)
851   MP=I
      GOTO 401
1001  WRITE(6,100)
100   FORMAT(15H FAILED IN RQIO)
      IFAIL=ITS
1000  IFAIL=ITS
      WR=RLAMBD
      IF(TRACE) WRITE(6,999)IFAIL,WR,Y
      IF(TRACE) WRITE(21,999)IFAIL,WR,Y
999   FORMAT('  ((CTR. RQIO ITS=',I3,'V.P.. & RES.=',2D25.15,'))'/)
      RETURN
      END
```

# ANNEXE III

## MÉTHODE D'ITÉRATION INVERSE

```
      subroutine itin
      c
      c*****************************************************************
      c                                                                *
      c RESOLUTION MATRICIELLE ASSOCIEE A L 'ANAKYSE MARKOVIENNE       *
      c            METHODE D* ITERATION INVERSE                        *
      c                                                                *
      c*****************************************************************
      c
      implicit real*8(a-a)
      real*8 d1,g,u
      dimension iw1(1500)
      real*8 xz
      logical tr
       common /mime/ a(50200),ind(50200)
       common/moy/zeps,ind2(50200),iw(1500,13),n,ia,nw,iiw
      data u /0.05/,ifail /1/,xmu /0.1d-10/
      data val /0.2/
      data jscale/1/
      integer indw(150),iww(150),ind1(7000)
      real*8 a1(7000),b1(1500),b(1500),yx,b2(1500),w(1500)
      data b2/1500*0.d0/
      c
      c a tableau de reels doubles de dimension(ia)
      c  a contient les elements non nuls de la matrice
      c
      c  ind tableau d* entiers de dimension (ia)
      c   ind contient les indices de ligne des elements non nuls de la matrice
      c
      c zeps reel double,contient la precision demandee au vecteur propre
      c
      c ind2 tableau de travail ,tableau d*entiers de dimension (ia)
      c
      c iw tableau d*entiers,iw(i,1) contient l*indice dans a du premier element
      c    de la colonne i,iw de dimension (iiw,13)
      c
      c n entier,ordre de la matrice
      c
      c ia entier,nombre d*elements non nuls de a
      c
      c nw entier,taille de iww et indw
      c
      c iiw entier ,premiere dimension de iw
      c
      c liste des tableaux de travail avec leurs dimensions
      c indw(nw),iww(nw),ind1(iw(n+1,1)),iw1(n),a1(iw(n+1,1)),b1(n),b2(n),w(n)
      c
      c retour en cas d*invraisemblance
      c
      9 continue
      write(21,1003) xmu,zeps
      write(6,1003) xmu,zeps
      tr=.false.
      1003 format( 5h xmu:,e12.6,6h zeps:,e12.6)
      c
      c l/u decomposition
      c
      c calcul de a
      c
      do 15 i=1,n
```

```
      ii=iw(i,1)
      ij=iw(i+1,1)-1
      iw1(i)=iw(i,1)
      do 16 j=ii,ij
      a1(j)=a(j)
      a(j)=a(j)/xmu
      ind1(j)=ind(j)
      if (ind(j).ne.i) goto 16
      a(j)=a(j)-1
   16 continue
   15 continue
      iw1(n+1)=iw(n+1,1)
      ik=iw(n+1,1)-1
      c
      c fin du calcul de a
      c
      c triangularisation de a
      c
       call f03ajf(a,ind,iw,iiw,ind2,n,iww,indw,nw,g,u,ia,d1,id,jscale,ifail
      c
      c test d'erreur de decomposition
      c
      write(6,1002) g
      write(21,1002) g
 1002 format( 30h erreur dans la decomposition:,e30.20)
      if (ifail.eq.0) goto 10
      if (ifail.eq.3) write(6,1) jscale
      write(6,1) ifail
      write(21,1) ifail
    1 format( 8h"erreur",i4)
      if (ifail.ne.1) goto 101
      xmu=0.1d-02
      jscale=1
      ia=50200
      do 102 kil=1,iw1(n+1)
      a(kil)=a1(kil)
  102 ind(kil)=ind1(kil)
      do 103 kil=1,n+1
  103 iw(kil,1)=iw1(kil)
      goto 9
  101 continue
      return
   10 continue
      c
      c initialisation de b
      c
      do 2 i=1,n
    2 b(i)=1.
      ifail=0
      niter=0
      c
      c resolution du systeme
      c boucle de recherche du vecteur propre
      c
   20 call f04apf(a,ind,iw,iiw,n,ia,w,b,1,ifail)
      niter=niter+1
      write(6,555) niter
  555 format( 10h iteration,i10)
      c
      c test d'erreur de resolution
```

```
c
if (ifail.eq.0) goto 40
write(6,3) ifail
3 format( 9h"erreur1",i4)
return
c
c calcul du residu
c
40 continue
xz=0.
do 33 i=1,n
33 xz=xz+dabs(b(i))
do 34 i=1,n
b(i)=b(i)/xz
34 b1(i)=0.d0
write(21,11)(b(i),i,i=1,n)
write(6,11)(b(i),i,i=1,8)
11 format(5(f15.10,3x,i2))
do 18 i=1,n
ii=iw1(i)
ij=iw1(i+1)-1
do 41 j=ii,ij
iz=ind1(j)
41 b1(iz)=b1(iz)+a1(j)*b(i)
18 continue
write(21,11) (b1(i),i,i=1,n)
write(6,11) (b1(i),i,i=1,8)
c
c test de vraisemblance et test d'arret
c
xmax=0.
xmm=0.
do 19 i=1,n
ax=dabs(b(i))-dabs(b2(i))
b2(i)=b(i)
if (xmm.lt.ax) xmm=ax
x=dabs(b1(i))
19 if (xmax.lt.x) xmax=x
write(6,11) xmax
write(21,5)niter,xmax
5 format( 11h iteration ,i4, 10h 1er test ,e13.6)
if (xmax.gt.zeps) goto 1206
goto 1207
1206 continue
write(6,1205) xmm
write(21,1205) xmm
1205 format( 11h 2eme test ,f12.9)
if (xmm.gt.0.00001) goto 20
write(6,1008)
write(21,1008)
1008 format( 22h arret au 2eme test   )
goto 1201
1207 continue
c test de vraisemblance
xpl=0.
xmo=0.
do 1200 i=1,n
if (b(i).lt.xmo) xmo=b(i)
if (b(i).gt.xpl) xpl=b(i)
1200 continue
```

```
if ((xpl.lt.0.002).or.(xmo.gt.-0.002)) goto1201
write(6,1202)
1202 format( 16h invraisemblance )
if (tr) goto 1201
tr=.true.
do 1203 i=1,n
if (b(i).gt.0.002) b(i)=1.
1203 continue
goto 20
1201 continue
return
end
```

# ANNEXE IV

## MÉTHODE DE RÉSOLUTION DIRECTE

```
      subroutine resol
      c*****************************************************************
      c                                                                *
      c RESOLUTION MATRICIELLE ASSOCIEE A L 'ANAKYSE MARKOVIENNE       *
      c                  METHODE DE RESOLUTION DIRECTE                 *
      c                                                                *
      c*****************************************************************
      c
      implicit real*8(a-a)
      real*8 d1,g,u
       common/pume1/a1(7000),ind1(7000),iw1(1500)
      real*8 xz
       common/pume/ a(50200),ind(50200)
       common/puy/ind2(50200),iw(1500,13),n,ia,nw,iiw
      data u /0.5/,ifail /0/,xmu /0.0d-10/
      data val /0.2/
      data jscale/1/
      integer indw(150),iww(150)
      real*8 b1(1500),b(1500),yx,w(1500)
      c

      c a tableau de reels de dimension (ia),contient les elements non nuls
      c    de la matrice (s-i) ou la matrice s est celle fournie par QNAP
      c   on a aussi remplace la derniere ligne par une ligne de 1
      c
      c ind tableau d'entiers de dimension (ia)
      c  ind contient les indices de ligne des
      c    elements non nuls contenus dans a
      c
      c ind2 tableau d'entiers de dimension (ia),tableau de travail
      c
      c iw tableau d'entiers de dimension (iiw,13)
      c   iw(i,1) contient l'indice dans le tableau du premier
      c   element de la colonne i, iw(n+1,1) contient le premier
      c   element de a non utilise
      c
      c n entier, ordre de la matrice
      c
      c ia entier, le nombre d'elements non nuls dans la forme decomposee
      c    est limite a ia
      c
      c nw  entier  taille des tableaux iww et indw, valeur n/10
      c     souhaitable
      c
      c iiw  entier  premiere dimension du tableau iw
      c      iiw >= n+1
      c
      c          tableaux de travail et leurs dimensions
      c
      c indw(nw),iww(nw), ind1(iw(n+1,1)), iw1(n), a1(iw(n+1,1)),w(n),b1(n)
      c
      c  a1, ind1, iw1 stockent (s-i)
      c
      c
      c triangularisation de a
      c
       call f03ajf(a,ind,iw,iiw,ind2,n,iww,indw,nw,g,u,ia,d1,id,jscale,ifail)
      c
      c test d'erreur de decomposition
      c
```

```
      write(6,1002) g
      write(21,1002) g
 1002 format( 30h erreur dans la decomposition:,e30.20)
      if (ifail.eq.0) goto 10
      if (ifail.eq.3) write(6,1) jscale
      write(6,1) ifail
      write(21,1) ifail
    1 format( 8h"erreur",i4)
      return
c ** initialisation de b **
   10 continue
c calcul de b
      do 2 i=1,n-1
    2 b(i)=0.
      b(n)=1.
c
c ** resolution du systeme **
c
      ifail=0
       call f04apf(a,ind,iw,iiw,n,ia,w,b,1,ifail)
c
c test d'erreur de resolution
c
      if (ifail.eq.0) goto 40
      write(6,3) ifail
    3 format( 9h"erreur1",i4)
      return
c
c calcul du residu
c
   40 continue
      do 34 i=1,n
   34 b1(i)=0.d0
      write(21,11)(b(i),i,i=1,n)
      write(6,11)(b(i),i,i=1,8)
   11 format(5(f15.10,3x,i2))
      do 18 i=1,n
      ii=iw1(i)
      ij=iw1(i+1)-1
      do 41 j=ii,ij
      iz=ind1(j)
   41 b1(iz)=b1(iz)+a1(j)*b(i)
   18 continue
      write(21,11) (b1(i),i,i=1,n)
      write(6,11) (b1(i),i,i=1,8)
      axmax=0.
      do 19 i=1,n
      ax=dabs(b1(i))
   19 if (axmax.lt.ax) axmax=ax
      write(21,11) axmax
      write(6,11) axmax
      return
      end
```

# B I B L I O G R A P H I E

[1] VERAN, M. : "QNAP : manuel d'utilisation".
Rapport technique, Centre Scientifique CII-HB, Grenoble, Novembre 1977.

[2] MERLE, D. ; POTIER, D. ; VERAN, M. :
"Un outil d'analyse des performances des systèmes informatiques". Congrès AFCET 1978.

[3] COURTOIS, P.J. : "Decomposability, instabilities and saturation in multiprogramming system."
C.ACM 18, 7, July 1975.

[4] LEROUDIER, J. ; PARENT, M. : "Discrete event simulation modelling of computer system for performance evaluation".
IRIA/LABORIA, Rapport de Recherche n° 177, juin 1976.

[5] STEWART, W.S. ; " MARCA : Markov chain analyser".
IRISA, Rapport n° 45, juin 1976.

[6] BASKETT, F. ; CHANDY, K.M. ; MUNTZ, R.R. ; PALACIOS, F.G. :
"Open, closed and mixed networks of queues with different classes of customers."
J. ACM 22, 2, april 1975.

[7] MARIE, R. : "Approximations et application de réseaux de files d'attente".
IRISA, Rapport n° 65, 1976.

[8] GELENBE, E. ; PUJOLLE, G. : "Probabilistic models of computer systems."
Part II, IRIA/LABORIA, Rapport de recherche n° 147, décembre 1975.

[9] MERLE, D. : "Algorithmes de calcul des probabilités stationnaires d'un réseau de files d'attente".
IRIA/LABORIA, décembre 1977.

[10] STEWART, W.J. : "Markov analysis of operating system techniqs ".
Ph.D. Thesis. Queen's University of Belfast, july 1974.

[11] CHATELIN, F. : "Spectral approximation of linear operators". Academic Press. New-York, février 1981.

[12] STEWART, W. : "Simultaneous iteration for computing invariant subspaces of non Hermitian matrices".
Num. Math. 25, 123-136 (1976).

[13] ARNOLDI, W.E. : "The principle of minimized iterations in the solutions of the matrix eigenvalue problem".
Quart. Appl. Math. 9, 17-29 (1951).

[14] SAAD, Y. : "Etude de la convergence du procédé d'Arnoldi pour le calcul des éléments propres de grandes matrices non symétriques".
Séminaire IMAG, Grenoble (17 mai 1979), Rapport n° 321.

[15] DANIEL, J.W. ; GRAGG, W.B. ; KAUFMAN, L. ; STEWART, G.W. "Reorthogonalisation and stable algorithms for updating the Gram-Schmidt QR-factorization.
Mathematics of computation, V-30, N.136 (octobre 1976).

[16] WILKINSON, J.M. ; REINSCH, C. : "Linear algebra".

[17] DUFF, I.S. : "Some current approches to the solution of large sparse systems of linear systems".
Decembre 1978.

dernière page de la thèse

# AUTORISATION DE SOUTENANCE

Vu les dispositions de l'article 3 de l'arrêté du 16 avril 1974

Vu le rapport de présentation de :

Madame F. CHATELIN

Monsieur M. VERAN

Monsieur C A C H A R D François

est autorisé à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR INGENIEUR, Spécialité "Génie Informatique".

Fait à Grenoble, le 11 septembre 1981

Le Président de l'I.N.P.G.

**D. BLOCH**
Président
de l'Institut National Polytechnique
de Grenoble